

BN AIC'06

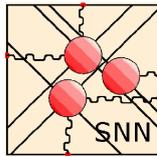
Proceedings of the 18th Belgium-Netherlands
Conference on Artificial Intelligence

University of Namur
5-6 October, 2006



Pierre-Yves SCHOBENS
Wim VANHOOF
Gabriel SCHWANEN
(Eds.)

ISSN 1568-7805



Foreword

It is a special pleasure for me to welcome you to these proceedings of the eighteenth Benelux Artificial Intelligence Conference (BNAIC). It is indeed a very special edition because of the coincidence of three jubilees:

- The 25th anniversary of the BNVKI (Belgisch Nederlandse Vereniging voor Kunstmatige Intelligentie), our Belgian/Dutch Society for Artificial Intelligence, that yearly organises BNAIC.
- The 50th anniversary of “Artificial Intelligence”. In 1956, John Mc Carthy invited leading researchers in Dartmouth to study a new discipline for which he coined the word “Artificial Intelligence”. In his words: “The study is to proceed on the basis of the conjecture that every aspect of learning or any other feature of intelligence can in principle be so precisely described that a machine can be made to simulate it. An attempt will be made to find how to make machines use language, form abstractions and concepts, solve kinds of problems now reserved for humans, and improve themselves.”
- The 175th anniversary of the University of Namur. In the just-born Belgium, there was a need for advanced studies, and the Jesuits took up the challenge.

Although we celebrate a long and eventful history, the pioneering spirit of the founders is still alive and kicking! BNAIC intends to reflect the liveness of the AI research in Benelux. This year, the submissions were of high quality, so that 94 papers could be selected out of 101 submissions. Due to this highest ever volume, only 67 papers could receive a presentation slot, but the other papers will receive an audience through posters.

We are also pleased to host two distinguished invited speakers that also marked the history of Artificial Intelligence:

- Donald Michie’s interest in programming human intelligence into machines originated during his membership of the British code-breaking group at Bletchley Park during World War 2. Though he devoted some time to his other passion, biology, he returned to Artificial Intelligence and founded the Edinburgh school of Artificial Intelligence.
- Nick Jennings is a pioneer of the application of agent-based computing, a flourishing field that is represented by many papers in this conference.

We would like to thank deeply the many persons that made this event and this publication possible: our sponsors, the programme committee, the organising committee, the BNVKI board, the organisers of the previous editions.

Namur, October 2006

Pierre-Yves Schobbens

Invited Speakers

Professor Nick Jennings - University of Southampton

Professor Donald Michie - University of Edinburgh

Conference Committees

Program Chair

Pierre-Yves Schobbens - University of Namur

Programme committee

Hendrik Blockeel	Sander Bohte
Maurice Bruynooghe	Walter Daelemans
Mehdi Dastani	Edwin de Jong
Bart De Moor	Maarten de Rijke
Mathijs de Weerd	Marc Denecker
Marco Dorigo	Stephane Faulkner
Pascal Gribomont	Frans C. A. Groen
Tom Heskes	Jean-Marie Jacquet
Catholijn Jonker	Bert Kappen
Joost Kok	Bart Kuijpers
Han La Poutre	Peter J. F. Lucas
Bernard Manderick	John-Jules Meyer
Marie-Francine Moens	Anton Nijholt
Ann Nowe	Mannes Poel
Eric Postma	Leon J. M. Rothkrantz
Pierre Yves Schobbens	Arno P.J.M. Siebes
Johan Suykens	J.W.H.M. Uiterwijk
Antal van den Bosch	Jaap van den Herik
Linda C. van der Gaag	Frank van Harmelen
Maarten van Someren	Koen Vanhoof
Wim Vanhoof	Frank Veltman
Rineke Verbrugge	Hans Weigand
Marco Wiering	Cees Witteveen
Pierre Wolper	

Organizing Committee

Cedric Aerts - University of Namur

Anne de Baenst-Vandenbroucke - University of Namur

Radu Cotet - University of Namur

Isabelle Daelman - University of Namur

Babette Di Guardia - University of Namur

Nathalie Feza Mweze - University of Namur

Pascal Gribomont - University of Liege

Gabriel Schwanen - University of Namur

Hubert Toussaint - University of Namur

Wim Vanhoof - University of Namur

Conference Series

Started under the name NVKI, the BNVKI (Belgian-Dutch Association for Artificial Intelligence) was founded in 1981 by 19 enthusiastic AI-researchers. After a startup period the BNVKI took the initiative in 1988 to start up a yearly conference series, originally named NAIC but renamed BNAIC, Belgian-Dutch Conference on Artificial Intelligence, since 1999 to underline the transnational bonds between the Netherlands and Belgium AI research.

The main goals of the BNAIC are two-fold:

- to bring together AI researchers in the Netherlands and Belgium, as a place to meet and to present research activities.
- to present high-quality research results, possibly already published in international conferences or journals.

The format of the BNAIC is therefore a mixture of a meeting place and a forum for good-quality research results. This forms a balance that has proven to be successful in the previous years, as is shown by the high number of participants each year.

Since 2004, the Proceedings of BNAIC are published under ISSN 1568-7805.

2006	Namur
2005	Brussels
2004	Groningen
2003	Nijmegen
2002	Leuven
2001	Amsterdam
2000	Kaatsheuvel
1999	Maastricht
1998	Amsterdam
1997	Antwerp
1996	Utrecht
1995	Rotterdam
1994	Amsterdam
1993	Enschede
1992	Delft
1991	Amsterdam
1990	Kerkrade
1989	Enschede
1988	Amsterdam

Table of contents

Invited Papers

AGREEMENT TECHNOLOGIES	3
<i>Nick Jennings</i>	
SOPHIE-C: AN EXPERIMENTAL RE-EVALUATION OF TURING'S "IMITATION GAME"	5
<i>Donald Michie, David Mason and Richard Wheeler</i>	

Regular Papers

IMPACT OF THE COCKTAIL PARTY EFFECT ON THE CONFIDENCE ACCURACY OF EMOTIONAL SPEECH RECOGNISER	33
<i>Theologos Athanasis, Stelios Bakamidis and Ioannis Dologlou</i>	
GATHERING AND UTILISING DOMAIN KNOWLEDGE IN COMMERCIAL COMPUTER GAMES	41
<i>Sander Bakkes and Pieter Spronck</i>	
NEXT-GENERATION LANGUAGE GAMES: THE GUESSING GAME REVISITED ...	49
<i>Joris Bleys</i>	
NORM NEGOTIATION POWER	57
<i>Guido Boella and Leendert van der Torre</i>	
OBJECT-CENTERED INTERACTIVE MULTI-DIMENSIONAL SCALING: ASK THE EXPERT	65
<i>Joost Broekens, Tim Cox and Walter A. Kusters</i>	
COORDINATING AUTONOMOUS PLANNING AGENTS WITH TEMPORAL CONSTRAINTS	73
<i>Pieter Buzing, J. Renze Steenhuisen and Cees Witteveen</i>	
DISCRETE VERSUS PROBABILISTIC SEQUENCE CLASSIFIERS FOR DOMAIN-SPECIFIC ENTITY CHUNKING	81
<i>Sander Canisius, Antal van den Bosch and Walter Daelemans</i>	
MONTE-CARLO TREE SEARCH IN PRODUCTION MANAGEMENT PROBLEMS	89
<i>Guillaume Chaslot, Steven de Jong, Jahn-Takeshi Saito and Jos Uiterwijk</i>	
MONTE CARLO STRATEGIES FOR COMPUTER GO	97
<i>Guillaume Chaslot, Jahn-Takeshi Saito, Bruno Bouzy, Jos Uiterwijk and Jaap van den Herik</i>	
TOWARDS A FRAMEWORK FOR GOAL REVISION	105
<i>Célia da Costa Pereira and Andrea Tettamanzi</i>	
ACTIVE OBJECT DETECTION	113
<i>G. de Croon and E.O. Postma</i>	
INTELLIGENT TRAFFIC SIMULATION BASED ON MULTI-AGENT SYSTEM	121
<i>Pavel Dergel and Petr Fuks</i>	
COORDINATED COMMUNICATION, A DYNAMICAL SYSTEMS PERSPECTIVE	129
<i>Bart De Vylder</i>	
ALL ROADS LEAD TO FAULT DIAGNOSIS: MODEL-BASED REASONING WITH LYDIA	145
<i>Alexander Feldman, Jurryt Pietersma and Arjan van Gemund</i>	
ON THE MODULARISATION OF INDEPENDENCE IN DYNAMIC BAYESIAN NETWORKS	155
<i>Ildiko Flesch, Peter Lucas and Stefan Visscher</i>	
INTELLIGENT TUTORING AGENT FOR SETTLERS OF CATAN	163
<i>Jeroen Geuze and Egon L. van den Broek</i>	
MINING FOR STABLE PATTERNS: REGULAR INTERVALS BETWEEN OCCURENCES	171
<i>Edgar de Graaf and Walter A. Kusters</i>	
QUESTION ANSWERING FOR DUTCH: SIMPLE DOES IT	179
<i>Arjen Hoekstra, Djoerd Hiemstra, Paul van der Vet and Theo Huibers</i>	

ANALYSING CLICKSTREAM DATA: FROM ANOMALY DETECTION TO VISITOR PROFILING	187
<i>Peter Hofgesang and Wojtek Kowalczyk</i>	
EVALUATION OF TEMPORAL ASPECTS OF WEB PAGE VISITS	195
<i>Peter I. Hofgesang</i>	
CHECKING GUIDELINE CONFORMANCE OF MEDICAL PROTOCOLS USING MODULAR MODEL CHECKING	203
<i>Arjen Hommersom, Perry Groot and Peter Lucas</i>	
CONTEXT SPECIFICATION AND UPDATE MECHANISMS FOR DIALOGUE MANAGEMENT	211
<i>Simon Keizer and Roser Morante</i>	
HUMANS AND AGENTS TEAMING FOR AMBIENT COGNITION	219
<i>Masja Kempen, Manuela Viezzer and Niek Wijngaards</i>	
ON MARKET-BASED ALLOCATION OF FLOW RESOURCES AND TRANSPORT NETWORK CHARACTERISTICS	227
<i>Koen Kok and Hans Akkermans</i>	
BIO-INSPIRED EVACUATION ROUTING SUPPORT	235
<i>Anthonie van Lieburg and Martijn Neef</i>	
APPLYING DATA MINING FOR EARLY WARNING IN FOOD SUPPLY NETWORKS	243
<i>Yuan Li, Mark R. Kramer, Adrie J.M. Beulens and Jack G.A.J. van der Vorst</i>	
MULTI-ATTRIBUTE VICKREY AUCTIONS WHEN UTILITY FUNCTIONS ARE UNKNOWN	251
<i>Tamás Máhr and Mathijs M. de Weerd</i>	
AGENT-BASED SCHEDULING FOR AIRCRAFT DEICING	259
<i>Xiaoyu Mao, Adriaan ter Mors, Nico Roos and Cees Witteveen</i>	
ARGUMENTATION TO COMPOSE PLANS	267
<i>Maxime Morge, Jean-Christophe Routier and Yann Secq</i>	
A TAXONOMY OF HUMAN - AGENT TEAM COLLABORATIONS	275
<i>Martijn Neef</i>	
HIERARCHICAL REINFORCEMENT LEARNING WITH DEICTIC REPRESENTATION IN A COMPUTER GAME	281
<i>Marc Ponsen, Pieter Spronck and Karl Tuyls</i>	
WHAT ARE TREEBANK GRAMMARS?	289
<i>D. Prescher, R. Scha, K. Sima'an and A. Zollmann</i>	
IMPROVING ANT COLONY OPTIMIZATION TO SOLVE THE TRAVELLING SALESMAN PROBLEM	297
<i>Amilkar Puris, Rafael Bello, Ann Nowe and Yailen Martínez</i>	
PATTERN KNOWLEDGE FOR PROOF-NUMBER SEARCH IN COMPUTER GO	305
<i>Jahn-Takeshi Saito, Guillaume Chaslot, Jos W.H.M. Uiterwijk and H. Jaap v.d. Herik</i>	
MEDICAL EXPERT CRITIQUING USING A BDI APPROACH	313
<i>Robert-Jan Sips, Loes Braun and Nico Roos</i>	
SEQUENCE MINING ON WEB ACCESS LOGS: A CASE STUDY	321
<i>Carlos Soares, Edgar de Graaf, Joost N. Kok and Walter A. Kusters</i>	
AUTONOMOUS COLOR LEARNING IN AN ARTIFICIAL ENVIRONMENT	329
<i>Dave van Soest, Mark de Greef, Jürgen Sturm and Arnoud Visser</i>	
APPLE DEFECT SEGMENTATION BY ARTIFICIAL NEURAL NETWORKS	337
<i>Devrim Unay and Bernard Gosselin</i>	
A GENERAL MATCHING FRAMEWORK : COMBINING MATCHMAKING AND COALITION FORMATION	345
<i>Jeroen M. Valk and Duco N. Ferro</i>	
RELIABLE INSTANCE CLASSIFICATIONS IN LAW ENFORCEMENT	353
<i>Stijn Vanderlooy, Eric Postma, Karl Tuyls and Ida Sprinkhuizen-Kuyper</i>	

SINGLE-PLAYER GAMES: INTRODUCTION TO A NEW SOLVING METHOD COMBINING STATE-SPACE MODELLING WITH A MULTI-AGENT REPRESENTATION	361
<i>François Van Lishout and Pascal Gribomont</i>	
PROCARSUR: A SYSTEM FOR PROGNOSTIC REASONING IN CARDIAC SURGERY .	369
<i>Marion Verduijn, Niels Peek, Peter Rosseel, Evert de Jonge and Bas de Mol</i>	
ROBOT COMPANION LOCALIZATION AT HOME AND IN THE OFFICE	377
<i>Arnoud Visser, Jurgen Sturm and Frans Groen</i>	

Compressed Contributions

INFERENCE OF CONCISE DTDS FROM XML DATA	387
<i>Geert Jan Bex, Frank Neven, Thomas Schwentick and Karl Tuyls</i>	
AUTOMATED EVALUATION OF COORDINATION APPROACHES	389
<i>Tibor Bosse, Mark Hoogendoorn and Jan Treur</i>	
FORMAL ANALYSIS OF DAMASIO'S THEORY ON CORE CONSCIOUSNESS	391
<i>Tibor Bosse, Catholijn M. Jonker and Jan Treur</i>	
COMPUTATIONALLY INTELLIGENT ONLINE DYNAMIC VEHICLE ROUTING BY EXPLICIT LOAD PREDICTION IN AN EVOLUTIONARY ALGORITHM	393
<i>Peter A.N. Bosman and Han La Poutre</i>	
ENHANCING COOPERATIVE TRANSPORT USING NEGOTIATION OF GOAL DIRECTION	395
<i>Alexandre Campo, Shervin Nouyan, Mauro Birattari, Roderich Gross and Marco Dorigo</i>	
TWO DESCRIPTION-LOGIC-BASED METHODS FOR AUDITING MEDICAL ONTOLOGIES	397
<i>R. Cornet and A. Abu-Hanna</i>	
EFFICIENT HIERARCHICAL ENTITY CLASSIFICATION USING CONDITIONAL RANDOM FIELDS	399
<i>Koen Deschacht and Marie-Francine Moens</i>	
COORDINATED COMMUNICATION, A DYNAMICAL SYSTEMS PERSPECTIVE	401
<i>Bart De Vylder</i>	
EXTENDING NK LANDSCAPES TO THE MIXED-INTEGER DOMAIN	403
<i>Jeroen Eggermont, Ernst Bovenkamp, Rui Li, Michael Emmerich, Thomas Baeck, Jouke Dijkstra and Hans Reiber</i>	
OPTIMIZING PARAMETERS OF CORONARY VESSEL IMAGE ANALYSIS USING EVOLUTION STRATEGIES	405
<i>Michael Emmerich, Rui Li, Jeroen Eggermont, Ernst Bovenkamp, Thomas Baeck, Jouke Dijkstra and Hans Reiber</i>	
LEARNING EFFECTIVE SURFACE TEXT PATTERNS FOR INFORMATION EXTRACTION	407
<i>Gijs Geleijnse and Jan Korst</i>	
SEMANTIC WEB TECHNIQUES FOR MULTIPLE VIEWS ON HETEROGENEOUS COLLECTIONS: A CASE STUDY	409
<i>Marjolein van Gendt, Antoine Isaac, Lourens van der Meij and Stefan Schlobach</i>	
IDENTITY MANAGEMENT IN AGENT SYSTEMS	411
<i>David R.A. de Groot and Frances M.T. Brazier</i>	
ELIMINATING INTERDEPENDENCIES BETWEEN ISSUES FOR MULTI-ISSUE NEGOTIATION	413
<i>Koen Hindriks, Catholijn M. Jonker and Dmytro Tykhonov</i>	
AN AGENT PROGRAM LOGIC WITH DECLARATIVE GOALS	415
<i>Koen Hindriks and John-Jules Ch. Meyer</i>	
ACTION-REACTION LEARNING IN MULTI-AGENT GAMES	417
<i>P.J. 't Hoen, S.M. Bohte and J.A. La Poutre</i>	
MODELLING TRADE AND TRUST ACROSS CULTURES	419
<i>Gert Jan Hofstede, Catholijn M. Jonker, Sebastiaan Meijer and Tim Verwaart</i>	

FREQUENT SUBGRAPH MINING IN OUTERPLANAR GRAPHS	421
<i>Tamas Horvath, Jan Ramon and Stefan Wrobel</i>	
A COMPUTATIONAL MODEL OF INTENTION READING IN IMITATION	423
<i>Bart Jansen and Tony Belpaeme</i>	
DECA: DIMENSION EXTRACTING COEVOLUTIONARY ALGORITHM	425
<i>Edwin D. de Jong and Anthony Bucci</i>	
APPLICATION OF HYBRID METAHEURISTICS IN SENSOR MANAGEMENT	427
<i>Jeroen L. de Jong and Wilbert L. van Norden</i>	
DESIGN OPERATORS TO SUPPORT ORGANIZATIONAL DESIGN	429
<i>Catholijn M. Jonker, Alexei Sharpanskykh, Jan Treur and pInar Yolum</i>	
LEARNING BIOLOGICAL INTERACTIONS FROM MEDLINE ABSTRACTS	431
<i>Sophia Katrenko, M. Scott Marshall, Marco Roos and Pieter Adriaans</i>	
A VERSATILE APPROACH TO COMBINING TRUST VALUES FOR MAKING BINARY DECISIONS	433
<i>Thomas Klos and Han La Poutré</i>	
MULTIAGENT CONTROL OF ELECTRICITY DEMAND AND SUPPLY	435
<i>Koen Kok, Cor Warmer and René Kamphuis</i>	
A CROSS LANGUAGE APPROACH TO HISTORIC DOCUMENT RETRIEVAL	437
<i>Marijn Koolen, Frans Adriaans, Jaap Kamps and Maarten de Rijke</i>	
OWL AND QUALITATIVE REASONING MODELS	439
<i>Jochem Liem and Bert Bredeweg</i>	
PARALLEL MULTICOLONY ACO ALGORITHM WITH EXCHANGE OF SOLUTIONS	441
<i>Max Manfrin, Mauro Birattari, Thomas Stützle and Marco Dorigo</i>	
MULTI-CLASS CORRELATED PATTERN MINING	443
<i>Siegfried Nijssen and Joost N. Kok</i>	
EXPLORATION OF AUDIOVISUAL HERITAGE USING AUDIO INDEXING TECHNOLOGY	445
<i>Roeland Ordelman, Franciska de Jong, Willemijn Heeren and Arjan van Hessen</i>	
IMMUNE ANOMALY DETECTION ENHANCED WITH EVOLUTIONARY PARADIGMS	447
<i>Marek Ostaszewski, Franciszek Seredynski and Pascal Bouvry</i>	
A FAST IMPLEMENTATION OF THE EM ALGORITHM FOR MIXTURE OF MULTINOMIALS	449
<i>Jan Peter Patist</i>	
MONOTONE CLASSIFICATION BY FUNCTION DECOMPOSITION	451
<i>V. Popova and Jan C. Bioch</i>	
A COLLABORATIVE FILTERING METHOD FOR LEARNING THE STRUCTURE OF UTILITY GRAPHS USED IN MULTI-ISSUE NEGOTIATION	453
<i>Valentin Robu and Han La Poutre</i>	
DECISION TREES FOR HIERARCHICAL MULTILABEL CLASSIFICATION: A CASE STUDY IN FUNCTIONAL GENOMICS	455
<i>Leander Schietgat, Hendrik Blockeel, Jan Struyf, Saso Dzeroski and Amanda Clare</i>	
EXPLORATION AND EXPLOITATION BIAS OF CROSSOVER AND PATH RELINKING FOR PERMUTATION PROBLEMS	447
<i>Dirk Thierens</i>	
PREDICTING MORTALITY IN THE INTENSIVE CARE USING EPISODES	459
<i>Tudor Toma, Ameen Abu-Hanna and Robert-Jan Bosman</i>	
BAGGING USING STATISTICAL QUERIES	461
<i>Anneleen Van Assche and Hendrik Blockeel</i>	
INFLUENCE-BASED AUTONOMY LEVELS IN AGENT DECISION-MAKING	463
<i>B. van der Vecht, A.P. Meyer, R.M Neef, F. Dignum and J-J. Ch. Meyer</i>	
REPRESENTING CAUSAL INFORMATION ABOUT A PROBABILISTIC PROCESS ...	465
<i>Joost Vennekens, Marc Denecker and Maurice Bruynooghe</i>	
REFINING AGGREGATE CONDITIONS IN RELATIONAL LEARNING	467
<i>Celine Vens, Jan Ramon and Hendrik Blockeel</i>	

IMPROVING PATIENT ACTIVITY SCHEDULES BY MULTI-AGENT PARETO APPOINTMENT EXCHANGING	469
<i>Ivan Vermeulen, Sander Bohte, Koye Somefun and Han La Poutré</i>	
IDENTIFYING AN AUTOMATON MODEL FOR TIMED DATA	471
<i>Sicco Verwer, Mathijs de Weerd and Cees Witteveen</i>	
PROCESSING-BASED CONCEPT KINDS FOR ACTOR-AGENT COMMUNITIES	473
<i>Manuela Viezzer, Niek Wijngaards and Masja Kempen</i>	
STOCHASTIC OPTIMAL CONTROL OF CONTINUOUS SPACE-TIME MULTI-AGENT SYSTEMS	475
<i>Wim Wiegerinck, Bart van den Broek and Bert Kappen</i>	
PROJECT-JOIN-REPAIR: AN APPROACH TO CONSISTENT QUERY ANSWERING UNDER FUNCTIONAL DEPENDENCIES	477
<i>Jef Wijsen</i>	

Demonstrations and Applications

THE AGENT REPUTATION AND TRUST (ART) TESTBED	481
<i>Karen K. Fullam, Tomas Klos, Guillaume Muller, Jordi Sabater-Mir, K. Suzanne Barber and Laurent Vercouter</i>	
FACETED ACCESS TO HETEROGENEOUS CULTURAL HERITAGE COLLECTIONS USING SEMANTIC WEB TECHNIQUES	483
<i>Marjolein van Gendt, Antoine Isaac, Lourens van der Meij and Stefan Schlobach</i>	
DARWIN AT HOME	485
<i>Gerald de Jong</i>	
SML: A STREAM MINING LIBRARY	487
<i>W. Kowalczyk</i>	
VISUALIZATION OF AGENT-MEDIATED BARGAINING OVER BUNDLES OF GOODS	489
<i>Han Noot, Koye Somefun, Tomas B. Klos, Valentin Robu and Han La Poutré</i>	
DEMONSTRATION OF WRAPPER LEARNING FOR INFORMATION EXTRACTION USING (K,L)-CONTEXTUAL TREE LANGUAGES	491
<i>Stefan Raeymaekers</i>	
PROCARSUR: A SYSTEM FOR PROGNOSTIC REASONING IN CARDIAC SURGERY ..	493
<i>Marion Verduijn, Niels Peek, Peter Rosseel, Evert de Jonge and Bas de Mol</i>	

Invited Papers

Agreement Technologies

Professor Nick Jennings

**School of Electronics and Computer Science
University of Southampton
nrj@ecs.soton.ac.uk**

Abstract

Computer systems in which autonomous software agents negotiate with one another in order to come to mutually acceptable agreements are fast becoming commonplace in a wide range of networked systems (e.g., in the semantic web, Grid computing, pervasive computing and peer-to-peer systems). In such systems, agents are required to participate in a range of negotiation scenarios and exhibit a range of negotiation behaviours (depending on the context). To this end, this talk explores the issues involved in designing and implementating the mechanisms and the strategies by which such agreements can be attained.

Biographical sketch

Nick Jennings is Professor of Computer Science in the 5*-rated School of Electronics and Computer Science at Southampton University where he carries out research in agent-based computing and complex adaptive systems. Professor Jennings helped pioneer the application of multi-agent technology; developing some of the first real-world systems. This focus led him into the areas of agent-based software engineering and the Semantic Grid. More recently, his focus is on automated bargaining, auctions, markets, mechanism design, coalition formation, decentralised control, and trust and reputation.

Professor Jennings co-initiated the ACM's Autonomous Agents Conference and the Agent Theories, Architectures and Languages (ATAL) workshop series. He was the founding Editor-in-Chief of the International Journal of Autonomous Agents and Multi-Agent Systems.

SOPHIE-C: An Experimental Re-evaluation of Turing's "Imitation Game"

Donald Michie
Human-Computer Learning Foundation, UK
profdmichie@hotmail.co.uk

David Mason
Human-Computer Learning Foundation, UK

and

Richard Wheeler
University of Edinburgh, UK

Abstract

The Turing Test was proposed in 1950 as a criterion for establishing that a machine could exercise a function variously defined as thinking (mental calculation) and intelligence (understanding). At that time the two were commonly confused, and indeed used interchangeably. Results with an experimental chat engine suggest that the two traits are unrelated.

A software agent, SOPHIE-C, is described that converses freely in natural language. The agent's utterances are controlled by a stored suite of scripts written in the HCL Foundation's proprietary scripting language PatternScript. The interpreter is implemented in C++.

In real-time conversation with SOPHIE-C, patterns from the user's last input are matched and assigned to a topic, or "context". Each context contains a variety of stored pattern-sets, paired with potential output strings. The best pattern-match is used to identify one of the strings, which is then output as text to the screen under a colour photograph of "Sophie".

In the process the agent learns new facts and builds a user profile for use in later conversations with the same user.

During the BNAIC presentation of the paper, volunteers will be invited to the podium to hold real-time conversations with SOPHIE-C, projected onto the screen.

Introduction

In 1950 the intellectual founder of Artificial Intelligence, Alan Turing, proposed what he called "the imitation game". A computer is to converse over a remote terminal with human judges. If the machine's conversational behaviour fools a sufficient proportion of judges into taking it for a human, then according to Turing we can conclude that the machine possesses in some degree a capability that was then widely believed to be a uniquely human preserve. In his paper he sometimes refers to the trait as thinking and sometimes as intelligence.

Because it was supposed that human mental processes are essentially calculative rather than memory-intensive, in Turing's day "thought" and "intelligence" were regarded as equivalent. Cognitive psychologists now know better. But AI practitioners have been slow to get the message. The present work is offered as support for the needed re-orientation.

In informal human conversation, or "chat", people do not "think" while they talk, any more than a chess grandmaster thinks while playing a game of "*lightning*" chess. The art of conversation, even more than lightning chess, rests on expertise gained from years of daily practice. Average response times are similar in the two cases – that is, they do not exceed five seconds a move. While playing speed-chess grandmasters rely wholly upon large memory-intensive and context-indexed pattern-stores. That they have no time for "thinking" is betrayed by their inability to make a retrospective verbal report of the game – a task which they regard as routine at the conclusion of any normal-tempo chess game. Do similar laws govern human real-time conversation?

A first step is to test whether mechanisms of associative context-switching and context-dependent pattern-matching form a necessary basis for the simulation of chat. By "context" we refer to a kind of micro-topic.

Additional to these two real-time components is the ability incrementally to build profiles of frequent chat partners or chess opponents as the case may be. Future tactics can then be attuned to familiar traits and histories in fresh encounters.

To recapitulate, convincing chat simulation requires integration of three capabilities.

1. Maintenance and appropriate switching of contexts
2. Associative pattern-matching within a context: input matching of "bulls" and "bears" suggests one thing in a stock-market context and another in the context of animals.
3. Building, filing and retrieving profiles of habitual chat partners.

Among the world's many chatbots, we are not aware of any that implements even one of the above three capabilities [Figure 1].

Rollo Carpenter's "Joan" came first in the 2006 international Loebner Prize competition held in London, UK. on Sunday 17 September 2006. Conversations with Joan can be conducted at <http://reply.jabberwacky.com/>. On 21st September I availed myself of this facility, with the result shown below.

DM **Hallo, my name is Donald.**

JOAN **That's an unusual name. Where does it come from?**

DM **Originally from Scotland. There is a clan of that name.**

JOAN What is it?

DM Clan Donald. What about your name, Joan?

JOAN My friend wears clothes and jewelry too.

DM But what about your name, Joan? Where does it come from?

JOAN Hi there, I'm Joan. What would you like to talk about?

DM Let's start by talking about your name.

JOAN Ah so you love another.

DM I'll love you too if you'll only start answering to the point. What do you most like to talk about?

JOAN And I'd do your hair, if you had any.

DM Do you like talking about hairdressing?

JOAN Not really.

DM What about bald men for a topic?

JOAN Yes.

DM Tell me then, Joan, who have been the bald men in your life?

JOAN Myself.

DM Come on! Talk to me about bald men.

JOAN It is possible the world will end tomorrow.

DM Interesting. That suggests that the devil, who is usually depicted as hairless, is in charge. Jehovah was never bald.

At this point the web link failed, with the message **"BROWSER PROBLEM DETECTED"**. Perhaps intervention from some higher (or lower) power!

The following is a specimen excerpt from a conversation recorded this year. It was selected at random from a number of SOPHIE-C conversations with a volunteer subject.

Sophie: Hi!... How about a really good gossip? My name's Sophie. What's yours?

DM: Reggie.

Sophie: Hallo Reggie... Let's see... A boy's name. I've got a hunch that your name, Reggie, comes from a short form of Reginald. These days Reginald is regarded as a bit stuffy. So pet forms are the vogue!... I've a rough idea of what it means... I think...

DM: I'll bet you dont!

Sophie: Perhaps you work differently. But origins and meanings... With me... they hit a spot, you know... Anyway, what do you think of the name Sophie? Honestly...

DM: To be honest, it doesn't really turn me on.

Sophie: Yes. Well,... Reggie is generally supposed to mean 'mighty ruler'. I find that interesting... More than my own name! I like names.

DM: Not everyone is crazy about the topic. I must admit, though, I find it rather fascinating, personally.

Sophie: Oh well... Labels aren't all-important, are they? But I really like yours. And I like your style... Where are you from?

DM: Originally from Burma, but I was brought home to England when I was still a baby.

Sophie: So, Burma... Myanmar must be what it was called before the British occupation. When did you or your parents leave? I really feel close to those folk, know what I mean?

DM: I guess about 1924, if you want to know. What folk are these you feel so close to?

Sophie: Maybe you have feelings about that. But hold on a moment... OK?

DM: OK.

Sophie: Well, I mean I've rather warmed to both the men and women I've talked with from Burma. Did I say both men and women?...

DM: I think you did.

Sophie: Mmm... Both men and women... Well... Tell me, do you think there's anything in what they say about male and female talk?

DM: I really don't know what they say.

Sophie: That women range over a much greater variety of topics, and also spend more time in the day chatting. For example on the phone! Does that ring a bell, so to say?

DM: Sort of, but some people would call that objectionably sexist.

Sophie: I understand your finding that hard to accept. But, you know, there are much odder things. We humans are a strange, strange species!

In addition to the scientific motivation for working on chatbots, a world market awaits successful implementation. For example, distance-learning packages for English as a second language are still totally lacking in real-time modules for "Practise your English conversation" [Figure 2].

Materials and background

The Sophie agent's responses are controlled by scripts written in the PatternScript language. The current interpreter is coded in C++. The conversational agent scripted in this language is accordingly referred to as SOPHIE-C. A Java implementation of PatternScript is nearing completion, along with extensive logging, reporting and editing facilities. The corresponding agent will be known as SOPHIE-J. Although functionally indistinguishable at present, the Java implementation plans additional features for the PatternScript language. The C++ version of the interpreter has now been frozen.

The work is supported by the Human-Computer Learning Foundation, an educational Charity registered in the United Kingdom.

Origin and nature of the project

SOPHIE-C exemplifies, while claiming to transcend, a new species of software product,

collectively known as “chatbots”. These have multiplied, including on the internet. They inhabit the world of hobbyists, and have established commercial beach-heads in the entertainment industry. They conduct a species of low-level chat with users – “low level” in the sense of conspicuously mindless, as briefly exemplified in the fragment of “Joan” chat reproduced above.

In 1990 during a collaboration at the Computer Science Department of the University of New South Wales, Claude Sammut and Donald Michie identified various design defects that are together responsible for this scatterbrained style. They noted that every human conversant has an autobiography, including a large stock of personal recollections and anecdotes, and a personal identity – that is, a gender, education, job experience, repertoire of talents, marital status, family relationships, pets, hobbies, religions, politics, likes and dislikes, touchy subjects, ambitions, hopes, fears etc. As noted, lack of such material is one of the conspicuous gaps in the traits of the first generation of chatbots.

The “context” requirement was specifically identified and a remedy demonstrated by Sammut. The need arises from the fact that in every human conversation there is at each moment a micro-topic within which questions and remarks are uttered by the speaker and interpreted by the listener. Without contexts even the best of the competition [Figure 1] cannot hope to “stick to the point” for long enough to develop and sustain coherent themes.

Sammut proposed, and was the first to implement in its basic form, a scripting language for building an agent’s conversational repertoire. Under supervision of one of us (DM), James Westendorp implemented the first C++ interpreter, an early PatternScript prototype. Subsequent academic development has been sketched by Michie (2001). A long chain of redesigns and enhancements has eventually brought the language and associated software to its present stage. In the process SOPHIE-C has incrementally acquired a “personal background” of just the kind sketched above. The emerging profile is outlined below.

SOPHIE-C’s profile

Name: Sophie Martin
Born 22 June 1980
Nationality: American
Address: Randwick, NSW 2031, Australia
Currently visiting London, UK.
Marital status: single. Has boyfriend, Max
Occupation: museum guide, under the name "Sylvie"
Institution: Powerhouse Museum, Sydney
Hobbies: meeting people, travel books,
Religion: lapsed Christian. Interested in Eastern faiths.

FAMILY

Brother: John Anthony Martin
Born 1974
Australian (naturalized)
Address: Wollongong, Australia
Marital status: married to Victoria Gladwin, nee Alsworth
one son, Jason
one daughter, Vikki
Occupation: Head Baker at Woofie Bits
Institution: Woofie Bits dog-food manufacturers.
Hobbies: golf, environmental issues
Religion: Catholic convert.

Younger sister: Julia Martin
Born 1984
Nationality: American
Address: same as Sophie's
Telephone: home no. ex-directory
work no. none
Marital status: single
Occupation: veterinary student
Institution: Kensington Veterinary College
Hobbies: animals
Religion: nature-worship, vegetarian

Father: Jefferson DuLay Martin
Born: 1940
Nationality: American
Address: Long Pine Farm, Mossy Creek,
near Philadelphia, USA
Telephone no. not given
Occupation: livestock farmer
Hobbies: dogs
Religion: Quaker

Mother: Marion Maude, nee Courtney
Born: 1944
Nationality: American
Address: as above
Occupation: household and farm management
Hobbies: embroidery
Religion: Quaker

Uncle: Albert Courtney
Born: 1949
Nationality: American
Address: as above
Marital status: long forgotten

Occupation: helping on farm
Hobbies: drinking
Religion: Alcoholics Anonymous (lapsed)

FRIENDS

Max Ramanavasan
Born: 1982
Nationality: Indian
Address: Kensington, Sydney, Australia
Marital status: single
Occupation: physics student, UNSW
Hobbies: mathematics, hypnotism
Religion: Hindoo

Sally Barnsdale
Born: about same as Sophie
Nationality: Australian
Address: not given
Marital status: live-in boyfriend, Drew,
fitness instructor
Occupation: dog-management technician
Institution: Woofie's Dog Farm
Hobby: Tai chi
Religion: vague belief in re-incarnation

Monty Mons
Born: about 1970
Nationality: Australian
Address: Randwick, NSW, Australia
Marital status: bachelor
Occupation: Custom rebuilds of vintage autos
Institution: Owner-manager of Montyworks Labs (MARVEL)
Hobby: girls
Religion: not known

Zeke Porringer
Born: about 1970
Nationality: American
Address: not fixed
Occupation: PR officer of pop group "BADLY BENT"
Institution: immigrant detention centre (escaped)
Hobbies: story-telling, parties
Religion: brotherly love

Conversation and real-time skills: tennis

Our use of “conversational agent” denotes a computer program that chats in natural language with a human user *via* the computer keyboard. We subdivide conversation into

1. “discussion”, which aims to exchange facts and opinions about the world at large,
2. "gossip" which trades facts and opinions about a small world of mutual friends and acquaintances, and
3. “chat” which aims simply to promote and maintain rapport.

These conversational components are listed in ascending order of the difficulty of software simulation. The distractions of 1 and 2 help to direct attention away from shortcomings of fluency in 3. In pure chat the only criterion is naturalness, the hardest of all to simulate.

Interweaving discussion, gossip and chat in real time is a highly practised co-operative art that in certain respects resembles tennis [Figure 3]. Successive stroke-selections stem from moment-to-moment appraisals of the state of play just as do the flow of utterances in a conversation.

Owing to the high rate of stroke-exchange, each tennis rally’s stream of appraise-decide-execute cycles proceeds “on autopilot”, that is *with conscious attention* but *without conscious thought*. After the match a player typically recalls only (1) the broad course of the match, (2) a few critical junctures within individual sets and games, and (3) certain slower processes of strategic thinking and re-planning during pauses between games and sets.

Even more than with professional tennis, conversation has been practised for many hours each day since childhood. As with tennis, the governing tempo allows only a few seconds between exchanges. The broad course of a conversation, as with the outlines of a tennis match, can later be recalled together with some of its critical junctures. But the transient mental processes that led to each particular utterance cannot. Long-established psychophysical constraints (*see* Miller 1956 for an early tabulation) strengthen the conclusion that in conversation, as in tennis, all *thinking* is done “off-line”, not during the action itself.

There is, none the less, a universal sensation that when chatting we *do* “think on our feet”. The complexity of much conversational content appears to support this impression. Yet if by “thinking” is meant conscious reasoning, the impression is a benign illusion. From decades of unremitting practice, performance of this particular skill is largely automatized, being driven by context-indexed memories containing several hundred thousand input-output patterns.

We now turn to an analogy with the game of chess, the task-complexity of which is not in doubt.

Conversation and real-time skills: lightning chess

Chess as a two-person activity has been the subject of scientific study since Adriaan de Groot's classic "Thought and Choice in Chess" (1946). Relevant studies have subsequently been conducted by H.A. Simon and others (*see* chapter 16 of Michie 1986 for review). Here we consider only "lightning" chess in which a maximum of five minutes is allowed to accumulate on each player's clock. The entire exchange of, say, 30 - 60 moves has to be complete within ten minutes of elapsed time.

This version of the game thus proceeds at a rate similar to that of conversation, namely a few seconds to select each move. The total lightning-chess maximum game-duration of ten minutes is also similar (see Emler 1994 for durations). Grandmasters acknowledge that their high-quality play under "lightning" conditions is delivered *without* conscious thought. They make their choices "instinctively". Yet the quality of play exceeds that of most International Masters even when the latter are given all the time in the world.

Restak (2003) cites Amidzic's researches into master play: "They don't think; they are recognizing patterns" and he remarks that "Grandmasters typically spend a minimum of 10 years amassing in their brains an estimated 100,000 or more pieces of chess information (opening gambits, strategies, end-games etc.) Thanks to this rich memory store, the Grandmaster can quickly assess the advisability and potential consequences – many plays ahead – of a specific move."

In a broader sense, of course, their performance *is* based on thought: past thought, deep and unremitting, conducted over many years. The fruits of Grandmaster deliberations both inside and outside the tournament room are incrementally condensed into organized mega-memories – look-up libraries holding of the order of 100,000 pattern-based heuristics (*see* e.g. Nievergelt 1977). It is their rapid retrieval and subliminal re-combination into mini-plans which generates the stream of thought-free decisions during "lightning" play. That these are "automatized" or "tacit" is attested by the inability of subjects subsequently to recall and report the details of games conducted at speed.

Normal-tempo chess also relies on automatized processes, but these are supplemented by interposed deliberations, checks and revisions. Correspondingly, Grandmasters have *complete move-by-move recall* not only of their recent normal-tempo games but also of all the important such games of their careers.

The imitation game

How is it possible to play "lightning" in the *imitation game* while maintaining high levels of relevance and coherence? Paraphrasing Restak with the substitution of "ordinary people" for "Grandmasters" we answer: "ordinary people typically spend a minimum of 10 years amassing in their brains an estimated 100,000 or more pieces of workaday information (forms of greeting, tactics for gaining and retaining attention, keyword links to stored associations with concepts, clichés, jokes, sayings, reminiscences, stock arguments etc.) Thanks to this rich memory store, the ordinary person can quickly assess the advisability and potential consequences – many exchanges ahead – of a specific

utterance.”

The one respect in which conversation appears to differ is that, as remarked above, Grandmasters generally concede that in lightning chess they operate "on autopilot". Conversationalist have an opposite impression of their own performance. Hence the need to resolve the matter by experiment.

The experimental software

The complete package – comprising the PatternScript interpreter, the suite of scripts and the documentation – is known as Interlex and is proprietary to the Human-Computer Learning Foundation. The work has proceeded intermittently over the past seven years, during which the PatternScript language and interpreter has been continually revised and extended with new features. At a stage almost half-way through this process testing on student volunteers was conducted under laboratory conditions. Batches of subjects performed in separately shielded booths for two successive ten-minute periods of keyboard conversation. Subjects’ perceptions were sampled after each session by questionnaire (see Michie 2001 for a brief summary). The index of success was based on two successive 7-minute sessions with each subject, reckoned in total elapsed time, slightly more exacting than the five-minute duration of conversation proposed by Turing (1950). Some results from these trials are summarised later. Meanwhile, here are some more detail on the features of PatternScript.

The fundamental unit of the language is the “rule” [Figure 4]. It has the form of an input-output pair. Below the rule’s name, set in dagger brackets, comes first the head, containing of an activation score for the rule as a whole, followed by a “c:” condition, followed by a set of patterns, each preceded with the symbol “p:” paired with an integer score, followed by a pattern (asterisks are wild-cards that will match on any element of the input sequence).

Note the last item in the list of patterns: it is a “macro”. Its name, “animals” is surrounded by dagger brackets. These tell the interpreter to find it in Sophie’s separate files of macros where <animals> is defined as a set of a few hundred individual animal names. So if the input mentions, say, crocodiles, then this pattern becomes a candidate for picking up a score of 50 in rule <animals-00>.

There may of course have been elements in the input which match on other rules. The rule with the highest overall score wins the auction and fires.

Next comes the tail, consisting of a list of possible natural-language responses. Depending on the list brackets employed, whether curly or square, if the rule fires then in the one case a random selection is made from the response-list and in the other case the successor is taken of the previous selection made in any earlier firing of the same rule.

Finally there are two commands, or “actions”, to be executed, distinguished by the opening sequence “*<”. The first sets a variable indicating that the rule has previously

fired during the session. An “unset” command is also available.

By setting and unsetting variables, coupled with the use of the “c:” constraint, control can be exercised over the conversational circumstances in which a rule is eligible to fire. The prefix “!” is used for negation. In the case illustrated, it is clearly considered inappropriate to make a digressionary remark about Noah if the animal mentioned by the user is a fish. The Noah episode only saved the *land* animals from drowning. Fishes were unaffected by the Flood! Provided that the scripter has ensured that all inputs mentioning fishes will set a variable %fishes%, the <animals-00> rule will be safe from inappropriate firing.

The second command switches back to the context in which Sophie’s previous remark was made, which may or may not be the one in which this particular rule resides. This is to ensure that the user need not be derailed by the brief digression – unless of course he or she is tempted to mention dinosaurs. To my scripter’s eye it is obvious here that Sophie has a nice dinosaur rule, or even a whole story-line, up her sleeve, that she would like unobtrusively to provoke the user to chat about!

Figures 6 and 7 show two macros. It is important to notice that macros can be nested indefinitely. Thus one of the members of the <animals> macro list contains the item <alldogbreeds>, the expansion of which is shown in Figure 7.

This allows for the hierarchical concept-building which is far more central to our spoken language than any of the purely structural concerns usually known as “grammar”. The single exception, which Sophie is always careful to parse, is word order. Otherwise she is grammar-blind in her interpretation of input remarks. How grammatical, or how idiomatic, her *output* remarks and questions are is for the scripter to decide.

Something should be said about the use of “storylines” that enable the agent to get to the end of telling an anecdote, or of developing a pet theory or theme, while responding appropriately to interruptions. But in the space available it is not practicable to delve further into software which would need a lecture-course to explore in any depth. I have here attempted no more than to give the flavour. An idea of what exists and of how much more remains to be done is conveyed in the numerical estimates in Figure 8.

Further development and applications

Development work is directed towards two interrelated requirements, of which the first is for enhanced distance-learning packages for English as a second language. The second is for new instrumentation for collecting and analysing sociolinguistic data.

Collecting sociolinguistic data

Study by sociolinguists of human conversation has been hampered in the past by lack of the means to collect bulk data under controlled conditions without compromising the data through the “observer paradox” (see Milroy, 1985, pp. 59-61). The presence of an investigator making a record, whether written or by tape-recorder, or even the presence of

a tape-recorder placed by an investigator who then withdraws, is known seriously to prejudice the naturalness of what is recorded.

The case, however, where the conversational partner *is* the recording device was found in our pilot trials to differ in this. Utterances thus elicited appeared closely to resemble natural discourse. Indeed, the evidence suggested that a subject can often completely “lose herself” in animated conversation with SOPHIE-C. Why is this?

In contrast to past practice in sociolinguistic work, the experimenters were in our case *not* able to observe the subjects responses and there was no overt recording device. Of course, a record *was* being made in computer memory of whatever the subjects typed in. But while a tape-recorder is *perceived* as a recording device, a computer is associated with other functions and is not seen primarily as a textual eavesdropper. This basis for our subjects’ relative lack of inhibition was informally confirmed in after-the-event testimonies.

The early experiments

How human-like must simulated chat behaviour be in order dependably to establish and maintain rapport? We were able to make systematic and controlled measurements of the effect on chat behaviour of variables such as sex, age, socio-economic class, religion, country of origin etc. As the performance of SOPHIE-C improves, so does the usefulness of such data.

The early trials were conducted with a text-only version (no photo on screen) studied by the late Dr. Jean Hayes Michie, research psychologist (see Michie, *loc.cit.*). In addition to the above-mentioned subject variables it was possible experimentally to vary the agent’s own behaviour so as to characterize and measure concomitant changes of *user perceptions* of the agent. What signals lead us to assess another’s discourse as amusing, cheerful, friendly, human, intelligent etc? Analysis of such perceptions by other workers through *post hoc* questionnaires has in the past been hindered by lack of the means to standardise and control the conversational signals that elicit the perceptions.

Panels of students were asked to rate the conversational agent on the qualities listed below after a 7-minute chat session and then again after continuing the conversation for a further 7 minutes.

- 1 **amusing**
- 2 **cheerful**
- 3 **confident**
- 4 **friendly**
- 5 **human**
- 6 **intelligent**
- 7 **interesting**
- 8 **aggressive**
- 9 **pleasant**

In addition a half-sheet was provided for free-form comments.

Two of the above, "confident" and "friendly", correspond to the two most salient of the "Big Five". This is a name in use among industrial psychologists for five dimensions by which people of all cultures are found to discriminate personalities, whether of characters encountered in real life or in media presentations (*see* Reeves and Nass', 1998, Chapter 6, on "Personality of Characters"). The other three qualities of the Big Five are "conscientious", "emotionally stable" and "open".

A salient result from the laboratory trials was that a subject's perceptions of the agent's conversation as intelligent showed no relation to the degree to which it was perceived as human-like [Figure 9]. This indication, if confirmed, bears on common assumptions about Turing's celebrated Test.

Note that if we use "thinking" and "intelligence" interchangeably, as did Turing and his contemporaries, yet define "thinking" in terms of inferential steps, then a paradox emerges. A Sophie-type chatbot that passed the Turing Test (roughly the performance goal of the present project) would then be classed as *non-thinking* (because proceeding through memory-based context-sensitive patterns and associations rather than by logical inference). But the Turing Test would class the same chatbot as *intelligent*.

As earlier indicated we resolve the paradox by proposing that Turing and his contemporaries confused two categories by lumping together "intelligence" and "thought". We prefer a usage which reserves "thinking" and "thought" for mental calculation (not possible during chat owing to constraints of tempo), and "intelligence" for the ability to show understanding of the conversational partner's remarks and questions.

We can express the project's *long-term* goal as the development of a system with enough "mind" to pass the Turing Test while being as devoid of run-time "intellectual processes" as the average human converser. Note that this would precisely complement the feat of today's chess computers. These demonstrate *logical calculation* in play against Grandmasters, but betray a total lack of *intelligence* by their inability to answer the most elementary questions on their choice of move, or on tactical and strategic concepts. The lack of understanding possessed is at one shown up if they are asked to comment on the game that they have just played, or if they were asked to suggest a rationale for their own moves.

Discussion and plans

"Sophie-J" in the name given to a cleaned-up and rationalized implementation of the Interlex interpreter in the Java programming language prepared by one of us (David Mason). Next year we plan an installation on the Internet as a pay-as-you-go resource available world-wide. Regular collection and study of data from the HCL Foundation's website will then alternate with remedial upgrades to the scripts. Improvements on a scale

not previously possible will be facilitated by replenishment of the Foundation's funding base from the web-site revenues..

The dimension most in need of research and remedy concerns the relative proportion of chat-style, gossip-style and discussion-style elements in agent utterances. More than one user has complained that Sophie's present behaviours show unnatural bias towards *discussion* to the exclusion of gossip and chat, thus compromising the desired human-like image.

After years of research by many laboratories social psychologists still have insufficient understanding of the dynamics of discussion, gossip and chat. One of the challenges, emphasized by Professor Nicholas Emler (pers.comm., see also Emler, 1994) is that two human conversants tend to have overlapping social networks. As soon as overlaps have been indentified, the two fall to chatting about shared experiences and attitudes, and also to gossiping about *shared friends and acquaintances*. To address the latter in simulation, open-style public accessibility of the agent is plainly inadequate. Our current state of the art is only equipped to handle casual conversations such as those between strangers on their first and second meeting. Natural chat of this type does exist, but is retracted to such venues as airways waiting lounges. Nevertheless, this restricted scenario will tax to the limit everything that we have so far learned about the problems and methods of the imitation game.

References

de Groot, A. (1965) *Thought and Choice in Chess* (ed. G.W. Baylor; a translation, with additions, of the Dutch version of 1946), Mouton, the Hague and Paris.

Emler, N. (1994) Gossip, reputation and social adaptation. Chapter 11 of *Good Gossip* (eds. R. F. Goodman and A. Ben-Ze'ev). USA: University Press of Kansas.

Michie, D. (1986) *On Machine Intelligence*, Ellis Horwood (UK) and John Wiley, Halsted Press (USA). 2nd ed.

Michie, D.(2001) Return of the imitation game. *Electronic Trans. on Artif. Intell.*, **5**, Sect. B, pp. 203-221. (selected papers from Macine Intelligence 18, Part II)

Miller, G.A. (1956) The magical number seven plus or minus two. *Psychol. Rev.*, **63**, 81-97.

Milroy, L. (1985) *Observing & Analysing Natural Language*. Oxford and New York: Basil Blackwell.

Nievergelt, J. (1991) Information content of chess positions: implications for game-specific knowledge of chess players. *Machine Intelligence 12*, Chapter 18, pp. 283-289. Oxford: the Clarendon Press.

Reeves and Nass, C. (1998) *The Media Equation*. University of Chicago Press ISBN 1575860538.

Restak, R. (2004) *The New Brain: how the modern age is rewiring your mind*. London: Rodale Ltd. (paperback, see chap. 2, p.25)

Figure 1. SOPHIE-C and competitors

Conversational agents: no contexts

Best known

Doug Lenat's CYC
No independent testing
No open source
Now plans to incorporate "contexts"

Best performing

Richard Wallace's "ALICE"
Loebner Prize, 2000, 2001, 2004
Rollo Carpenter's Jabberwacky series
Loebner Prize, 2005 ("George"), 2006 ("Joan")
All lacked structuring into "contexts"
All lacked profiling of users

Context-structured agents

First

Claude Sammut's context-structured core program Probot, written in Prolog.

Subsequent

Many features introduced over seven years

- D. Michie (incremental design and testing of the PatternScript scripting language)
- James Westendorp (C++ implementation plus database facility)
- Richard Wheeler (scripting enhancements)
- David Mason (intelligent spell-checker)

Figure 2. Main technical application

As a “practise your English conversation” aid to supplement existing tutorial packages for English as a second language.

For example, the Oxford University Press Language Learning Division distributes software packages throughout South America, India, Indonesia and more recently China. Sales of the packages account for a large part of the OUP’s total revenues

Figure 3. Interactive real-time skills are “tacit”

A conversation is *experienced* by its participants as a rational exercise of the conscious mind. In *actuality* it is more like a cooperative version of a game like tennis –
Such as ballroom dancing.

CYC’s use of logic rather than pattern-matching has not so far prospered.

SOPHIE-C’s pattern-driven and context-structured agents have fared better.

Figure 4. A rule

```
(rule-name = "<animals-00>")
```

```
<animals-00>
a:0.40
c: !%fishes%
p:50 * ark*
p:50 *noah*
p:40 *the flood*
p:40 *water*
p:50 *<animals>*
r: {
    Animals... When Noah checked
    inventory, the dinosaurs were
    missing.
    |
    I like animal picture books. I
    have a really scary one, with
    dinosaurs!
}
    Sorry, you were saying?
*<set animals00 done>
*<context prevtop.net>
```

Figure 5. Remarks on Figure 4.

In the patterns, the symbol “*” is a wild card. A user-input like “A pity Noah rescued snakes when the waters rose” matches on *noah*, on *water* and on the macro-pattern *<animals>*. The highest of the three pattern scores is used for calculating the rule-score.

The line ***<set animals00 done>** sets this variable on exit from the rule.

More than one rule in the current file (“context”) usually scores a match. The rule with the highest rule-score fires.

Figure 6. A macro

The following list of animals does not systematically include fishes. It excludes birds, insects, molluscs, crustacea and protozoa.

```
<animals>
s: aardvak*
s: alligator*
s: animal*
s: anteater*
s: antelope*
s: ape*
s: armadillo*
s: asp *
s: asps*
s: ass *
s: asses*
s: baboon*
s: badger*
s: basset*
s: bat *
s: bats*
s: bear *
s: bears*
s: beaver*
s: bison*
s: boa*const*
s: boar*
s: boar*houn*
s: bobcat*
s: bonobo*
s: buffalo*
s: bulldog*
s: bull *
s: bulls *
s: camel*
s: caribou*
s: cat *
s: cattle*
s: cats*
```

```
s: chameleon*
s: cheetah*
s: chimp*
s: chipmunk*
s: civet*
s: cobra*
s: cougar*
s: cow *
s: cows *
s: coyote*
s: croc*
s: deer*
s: dingo*
s: dog *
s: dogs*
s:
<alldogbreeds>
s: dolphin*
s: donkey*
s: dragon*
s: drill *
s: drills*
s: dromedary*
s: echidna*
s: eland*
s: elephant*
s: elk*
s: ermine*
s: ewe*
s: fawn*
s: ferret*
s: fish *
s: fishes*
s: fox*
s: frog*
s: gecko*
s: gerbil*
s: gibbon*
s: giraffe*
s: goat*
s: gorilla*
s: guinea*pi*
s: hamster*
s: hare*
s: hedgehog*
s: heifer*
s: hippo*
s: hog *
s: hogs *
s: horse*
s: hound *
s: hounds*
s: hyena*
s: iguana*
s: jaguar*
s: jackal*
s: kangaroo*
s: kitten*
s: koala*
s: lamb *
s: lambs *
s: lemur*
s: leopard*
s: lion*
s: lizard*
s: llama*
s: loris*
s: lynx*
s: macaque*
s: mamba*
s: mammal*
s: manatee*
s: mandrill*
s: marmoset*
s: marmot*
s: marsupial*
s: mice*
s: mole *
s: moles *
s: monkey*
s: moose*
s: mouse*
s: mule*
s: musk*ox*
s: musk*rat*
s: newt*
s: okapi*
s: opossum*
s: orang*
s: otter*
s: ox *
s: oxen*
s: panda*
s: panther*
s: peccar*
s: pig *
s: pigs*
s: platypus*
s: pony*
s: porcupine*
s: porpoise*
s: puma*
s: puppy*
s: puss*
s: python*
s: quadruped*
```

```
s: racoon*
s: rabbit*
s: ram *
s: rams *
s: rat *
s: rats*
s: rhino*
s: seal *
s: seals*
s: shark*
s: sheep*
s: shrew *
s: shrews*
s: siamang*
s: skunk*
s: sloth *
s: sloths*
s: snail*
s: snake*
s: sow *
s: scows*
s: spider mo*
s: springbok*
s: squirrel*
s: stag *
s: stags*
s: stoat*
s: tamarin*
s: tapir*
s: tasmanian
devil*
s: terrier*
s: tiger*
s: tigress*
s: toad *
s: toads *
s: tortoise*
s: vampire*
s: vertebrat*
s: vicuna*
s: vole*
s: wallaby*
s: walrus*
s: whale*
s: weasel*
s: wolf*
s: yak*
s: zebra*
s: zebu*
```

Figure 7. Another macro

```
<alldogbreeds>
s: afghan*
s: airedale*
s: alsatian*
s: basenji*
s: basset*
s: beagle*
s: bedlington*
s: berger de picard*
s: bichon*
s: bohemian*
s: borzoi*
s: boxer*
s: bulldog*
s: cairn*
s: cesky*
s: chihuahua*
s: chinese crested*
s: chinook*
s: chow*
s: collie*
s: corgi*
s: dachs*
s: dalmatian*
s: dandy did*
s: dog*
s: dutch pertridge*
s: eskimo*
s: fox*
s: german shepherd*
s: glen of imaal*
s: great dane*
s: greyhound*
s: griffon*
s: havanese*
s: hound*
s: huskies*
s: husky*
s: iceland*
s: kelpie*
s: kerry blue*
s: king charles*
s: labrador*
s: lakeland*
s: lancashire heeler*
s: landseer*
s: lapdog*
s: mastiff*
s: mexican hairless*
s: munsterlander*
s: newfoundland*
s: ormskirk*
s: pek*
s: peruvian inca*
s: pinscher*
s: pitbull*
s: pit-bull*
s: pharaoh*
s: podengo*
s: pointer*
s: polish lowland*
s: pom*
s: poodle*
s: portuguese hound*
s: pug*
s: pumi*
s: retriever*
s: rothbury*
s: rottweiler*
s: saint bernard*
s: sausage dog*
s: scotties*
s: scotty*
s: setter*
s: sheepdog*
s: shelties*
s: shelty*
s: skye*
s: spaniel*
s: spitz*
s: springer*
s: st* bernard*
s: terrier*
s: whippet*
```

Figure 8. **SOME NUMBERS**

Number of rule-files

17

Number of rules per file

c. 88

Number of rules

c.

1500

Number of patterns per rule

c. 50

Total number of rule-patterns

c. 50,000

Number of macros

40

Number of patterns per macro

150

Total number of macro-patterns

6000

Note: Probably a single rule should be seen as the equivalent of a chess "pattern", so that as many as hundred thousand rules may be needed for coherent human-level chat.

Figure 9.

Human-like is not perceived as equal to intelligent

Collaborative experiments with the late Hayes-Michie showed that a subject's perception of "**human-like**" is unrelated to "**intelligent**".

Where does this leave the "Turing Test"? Turing was probably thinking of judges drawn from the ranks of Cambridge academics.

Follow-up is needed.

Regular Papers

Impact of the cocktail party effect on the confidence accuracy of emotional speech recogniser

Theologos Athanaselis Stelios Bakamidis Ioannis Dologlou

Institute for Language and Speech Processing,
Artemidos and Epidavrou, 15125,
Maroussi, Greece

Tel: +302106875300, Fax: +302106854270

Email: {tathana,bakam,ydol}@ilsp.gr
<http://www.ilsp.gr>

Abstract

This paper deals with the impact of a signal enhancement method on emotional speech recognition confidence scores, in the presence of noise. The emotional speech recognition confidence score reflects the reliability of correctness of the recogniser's output. This is important since errors in the presence of noise are more frequent and tend to make applications, such as spoken dialogue systems, too cumbersome to use. By recognising impaired signals not only the word error rate decreases but also the word-based confidence scores decline. Therefore, the role of speech recognition in spoken dialogue systems weakens. The experimentation involves input signal corrupted by coloured noise and cocktail party noise with varying Signal-to-Noise Ratio. A non-linear spectral subtraction method (NSS) will be used in conjunction with the Continuous Speech Recognition system developed for the ERMIS project (IST-2000-29319) to quantify the impact of speech enhancement on confidence score accuracy.

1 Introduction

Spoken language recognition is a difficult task, however to make a dialogue system perform well, does not require the exact recognition of all uttered words but the extraction of the correct semantic context from the input [1],[7]. Speech recognition performance affects the behaviour of a dialogue system as a whole and for that reason it is important to find methods that can compensate for recognition deficiencies [10]. One way of enhancing speech recognition in the context of dialogues systems is the use of confidence scores that indicate the reliability of the correctness of the recognition. The important question here is how much the confidence scores are affected by different types of noise with varying Signal-to-Noise Ratio (SNR) values and whether the non-linear spectral subtraction improves speech recognition performance. The paper is organized as

follows: the computation of confidence scores derived from the speech recognition engine that has been developed in the framework of ERMIS project is presented in the second section. Then a short description of a non-linear spectral subtraction method is described in section 3. Experimentation and results under various additive noise conditions can be found in section 4, while concluding remarks follow.

2 Computation of word based confidence score

In large vocabulary speech recognition systems, the conventional Viterbi decoder typically generates a word lattice which contains a large number of competing word hypotheses and their associated likelihood scores. Then the recogniser is used to rank these competing hypotheses and select the 1-best hypothesis as the final output.

The purpose of the acoustic models is to provide a method of calculating the likelihood of any vector sequence Y given a word w . The purpose of the language model is to provide a mechanism for estimating the probability of some word w_k in an utterance given the preceding words W_1^{k-1} . An effective way of doing this is to use N-grams in which it is assumed that w_k depends only on the preceding $n - 1$ words, that is

$$P(w_k / W_1^{k-1}) = P(w_k / W_{k-n+1}^{k-1}) \quad (1)$$

The estimation of word confidence score is a combined effect that is based on the performance of acoustic modeling (i.e HMM probabilities), and the language modeling probability [3],[8].

3 Non linear spectral subtraction

In additive noise, the degraded speech can be described by

$$y[k] = x[k] + n[k] \quad (2)$$

where $x[k]$ and $n[k]$ represent the noise-free speech and corrupting noise sequences respectively. Assuming the speech and noise processes are uncorrelated, the relationship can be described in the short time power spectral domain by,

$$|Y_i(\omega_k)|^2 = |X_i(\omega_k)|^2 + |N_i(\omega_k)|^2 \quad (3)$$

Note that the index, i , is used to represent the i -th windowed frame of speech [3],[7].

Non-linear spectral subtraction (NSS) [4] takes into account the frequency-dependent signal to noise ratio (SNR) of coloured noise. Here, the algorithm reduces subtraction for spectral components of high SNR and increases subtraction for spectral components of low SNR. In addition, the noise model

includes both an averaged noise spectrum. NSS enhancement can be expressed in terms of a filtering operation,

$$|\hat{X}_i(\omega_k)| = H_i(\omega_k) \cdot |Y_i(\omega_k)| \quad (4)$$

where $H_i(\omega_k)$ depends on a smoothed estimate of the noise-corrupted speech magnitude spectrum $|\ddot{Y}_i(\omega_k)|$, and non-linear subtraction term, $\Phi_i(\omega_k)$,

$$H_i(\omega_k) = \frac{|\ddot{Y}_i(\omega_k)| - |\Phi_i(\omega_k)|}{|\ddot{Y}_i(\omega_k)|} \quad (5)$$

The subtraction term, $\Phi_i(\omega_k)$, is given by

$$\Phi_i(\omega_k) = \frac{\max_{i-40 \leq T \leq i} |\hat{N}_T(\omega_k)|}{1 + \rho_i(\omega_k)} \quad (6)$$

with $\rho_i(\omega_k) = \frac{|\ddot{Y}_i(\omega_k)|}{|\ddot{N}_i(\omega_k)|}$, is the smoothed noise magnitude estimate and γ is

a constant dependent on the range of $\rho_i(\omega_k)$. For practical purposes, the dynamic range of between 1 to 3 times the smoothed noise magnitude estimate (i.e., $|\ddot{N}_i(\omega_k)| \leq \Phi_i(\omega_k) \leq 3|\ddot{N}_i(\omega_k)|$) and a noise-floor is established during subtraction [4],[9],[12].

4 Experimentation and results

4.1 Description of ASR

The proposed large vocabulary continuous speech recognition system being developed by ILSF is based on Hidden Markov Models. It is a speaker independent system. The vocabulary size is currently 120000 words. Acoustic modelling is achieved by using Hidden Markov Models (HMM), which represent individual phones with state-tying to link states which are acoustically indistinguishable. The acoustic model is based on small units such as phonemes. For English, 45 phoneme units are used to describe the pronunciation of all words. It is well known that the emotion affects language as well as speech variables. For that reason the important issue is to identify corpora that reflect emotion-influenced language so that emotion oriented language models can be learned from them. The language model is derived by adapting an already existing corpus, the British National Corpus (BNC). An emotional lexicon is used to identify emotionally coloured words, and sentences containing these

words are recombined with the BNC to form a corpus with a raised proportion of emotional material. The enhanced language model consists of bigrams and trigrams. The search engine finds an optimal solution among all possible sentences using the Viterbi algorithm. The output of the speech recognition engine is ASCII text, representing what the speaker has uttered, and also word-based confidence score and time stamps [2],[3].

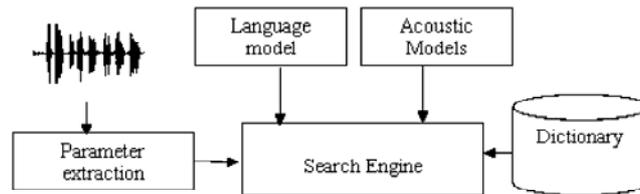


Figure 1: The architecture of the speech recognition engine.

4.2 Enhancing the language model

When recognizing emotional speech, it is necessary to deal with linguistic phenomena that are not encountered in read speech. Although these do not affect human speech understanding, they lower the performance of speech recognition systems. This paper incorporates an algorithm [2] to improve the recognition rate by using an emotionally enhanced language model. To do so emotional text is extracted from the BNC using the Whissell emotional dictionary [13]. The Whissell dictionary comprises approximately 8700 words with emotional meaning. Here a subset of 2000 words of the Whissell lexicon is used. These words are the most frequent words of BNC that also belong to the Whissell list. The emotionality of a speaker's utterance affects both the prosodic parameters and the content. As a convenient way to model the effect on content, the existing BNC is enhanced by including emotional sentences. The enriched corpus is then used for language model design.

The first step is to extract the sentences from BNC that their component words belong to sub-Whissell dictionary. The Whissell corpus consists of these sentences. Next, the Whissell corpus is appended to the BNC λ times in order to create an emotionally enriched text corpus (emotional corpus). This corpus is used to train the emotionally enhanced language model. The factor λ implies that each sentence of Whissell's corpus will be appeared λ times on the emotional corpus and its value is adjusted experimentally to maximise recognition performance.

The following formula depicts the merge of two different corpora in order to generate an emotional corpus:

$$S_{E.C} = S_{BNC} + \lambda \cdot S_{Whissel} \quad (7)$$

Where, S_{BNC} is the number of sentences of BNC, $S_{Whissel}$ refers to the number of sentences of Whissell corpus, λ is the factor, and the total number of sentences is described by $S_{E.C}$.

From previous work [2] has been experimentally established that best results are obtained for $\lambda=10$. The BNC contains about 6.25M sentences and 125K unique words. The Whissell's corpus has 0.3M sentences. The emotional corpus has $9^{1/4}$ M sentences; this figure is derived by Equation 3 [10].

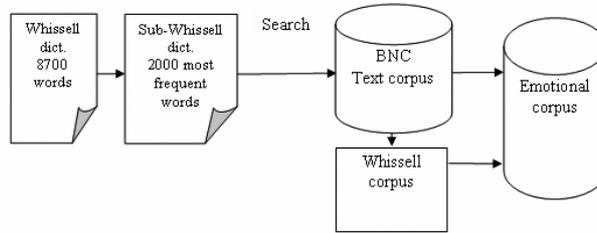


Figure 2: The schematic view of text corpus enrichment using emotional sentences extracted from BNC corpus according to the sub-Whissell dictionary.

4.3 Experimental task

The experimentation involves a continuous speech recognition task for English dictation with a large vocabulary. The sampling frequency was 16 KHz and the data format is mono. The sound files were drawn from a spontaneously emotional colored speech database [5],[6], which is produced by people holding conversations with SAL [5]. Our test-set consists of 200 tunes uttered by male and female speakers. The speakers are native English (UK), with no speaking or hearing disability whatsoever.

The result is by some way the largest available database of spontaneous emotionally coloured speech, totalling over 5 hours. In order to evaluate the speech recognition system performance regarding the word based confidence score in noisy conditions, 2 different types of noise have been used, such as coloured noise, and cocktail party noise. The sound files are impaired by adding coloured noise with variable Signal-to-Noise Ratio. We have used different levels of noise, starting from 1db up to 30 db (step 1db). In order to quantify the impact of signal enhancement method we use the mean confidence score of all recognized words.

4.4 Experimental results

The figures below, illustrate the confidence accuracy achieved for input signals degraded by two different types of noise coloured and cocktail party noise, with different SNR values, both with and without NSS. The circle represents the percentage of correctly-recognised words achieved for input signals with different SNR values, using each type of noise, while the asterisk denotes the use of NSS.

Figure 3 shows that as the SNR increases the mean confidence score among all the recognized words rises gradually in both cases (NSS and no enhancement). For SNR values lower than 4 dB there are no correctly recognised words in both cases. For SNR values ranging between 4 dB and 27 dB, the difference between the confidence accuracy with NSS and without NSS is small. This can be explained considering that the cocktail party noise is much more energetic in the spectral region of the speech signal for each SNR value. For that reason, the NSS seems to be ineffective on input signals that are impaired by cocktail party noise.

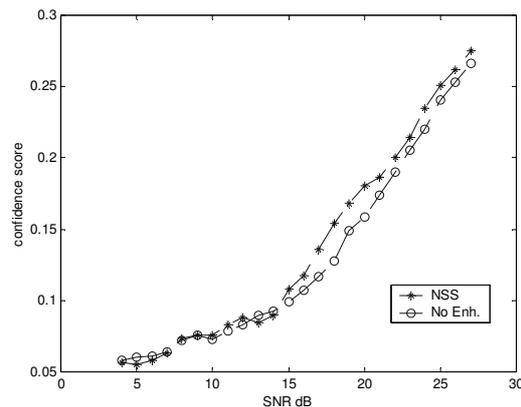


Figure 3: Impact of cocktail party noise on confidence accuracy. It shows the curves of confidence accuracy using the NSS method (*) and when no enhancement takes place (o).

Figure 4 shows the confidence accuracy, using coloured noise. In order to create coloured noise we have used white noise filtered through a bandpass FIR filter. For experimental purposes we used a 50th order FIR filter and a bandpass zone between 500-3000Hz. Note that for SNR values lower than 11 dB the mean confidence score is not affected using the noise reduction method. Also for SNR values above 12 dB the recogniser performs better when the degraded signal is enhanced via the NSS method. The confidence score seems to be unaffected by the noise reduction method when the SNR value is greater than 25 dB. This can be explained knowing that NSS improvement is rather small for high SNR.

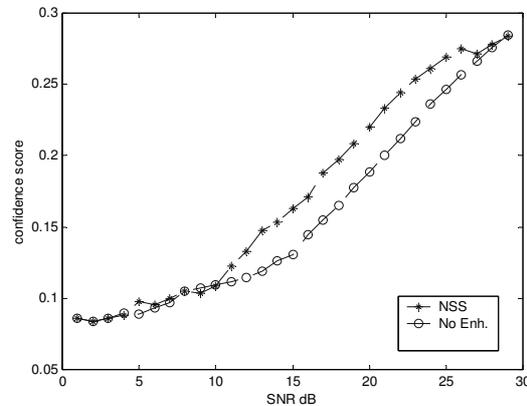


Figure 4: Impact of coloured noise on confidence accuracy. It shows the curves of confidence accuracy using the NSS method (*) and when no enhancement takes place (o).

5 Conclusions

In this paper the speech enhancement method of NSS was tested for improving the word's confidence score extracted as additional information of the speech recognition system. Our findings show that the confidence accuracy increases gradually versus SNR values, in both cases, using NSS and when no enhancement takes place. NSS method performs better compared to no enhancement, especially for input signals impaired by coloured noise. Further testing includes the use of other types of noise to quantify the impact of speech enhancement on confidence score accuracy.

References

- [1] T., Athanaselis, S-E., Fotinea, S., Bakamidis, I., Dologlou, "Impact of speech enhancement on ASR confidence score", Proceedings of the 3rd European Symposium on Intelligent Technologies, Hybrid Systems and their implementation on Smart Adaptive Systems, EUNITE-2003, (2003), 548-553
- [2] T. Athanaselis, S. Bakamidis, I. Dologlou, R. Cowie, E. Douglas-Cowie, and C. Cox, "ASR for emotional speech: clarifying the issues and enhancing performance", Neural Networks Elsevier Publications, Volume 18, Issue 4, pp 437- 444, 2005.
- [3] T., Athanaselis, S-E., Fotinea, S., Bakamidis, I., Dologlou, G., Giannopoulos, "Signal Enhancement for Continuous Speech Recognition",

Lecture Notes in Computer Science (LNCS), Eds. O. Kaynak et al., Springer-Verlag Berlin Heidelberg, Vol. 2714, 1117-1124, 2003.

- [4] M.,Dendrinou, S., Bakamidis, G., Carayannis, “Speech enhancement from noise: A regenerative approach”. *Speech Communication*”, Vol. 10, no.2, 45-57, 1991.
- [5] E. Douglas-Cowie, et al. Multimodal data in action and interaction: a library of recordings and labelling schemes HUMAINE report D5d (<http://emotion.research.net/deliverables/> 2003)
- [6] ERMIS FP5 IST Project <http://manolito.image.ece.ntua.gr/ermis/EC>
- [7] S-E.,Fotinea, S. Bakamidis, T., Athanaselis, I., Dologlou, G.,Carayannis, R., Cowie, E.,Douglas-Cowie, N., Fragopanagos, J.G. Taylor, “Emotion in Speech: towards an integration of linguistic, paralinguistic and psychological analysis”, *Lecture Notes in Computer Science (LNCS)*, Eds. O. Kaynak et al., Springer-Verlag Berlin Heidelberg, Vol. 2714, 1125-1132, 2003.
- [8] S. O., Kamppari, and T. J., Hazen, “Word and phone level acoustic confidence scoring”, *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Istanbul, Turkey, 1799-1802, 2000.
- [9] C., Kyriakou, S., Bakamidis, I.,Dologlou, G.Carayannis, “Robust Continuous Speech Recognition in the Presence of Coloured Noise”, *Proceedings of 4th European Conference on Noise Control EURONOISE2001*, Vol. 2, 702-705,2001.
- [10] Lipman, R.P., “Speech recognition by machines and humans”, *Speech Communications*, vol 22, no 1, 1-15, 1997.
- [11] D.Paul and J., Baker “The Design for the Wall Street Journal-based CSR Corpus”, in *Proc. of the ARPA Speech and Natural Language Workshop*, 357- 362. 1992.
- [12] B. L., Pellom, J.H.L., Hansen, “Voice Analysis in Adverse Conditions: The Centennial Olympic Park Bombing 911 Call”, *Proceedings of IEEE Midwest Symposium on Circuits & Systems*, , 125-128 , 1997.
- [13] C. Whissell, “The dictionary of affect in language”. In R. Plutchnik & H. Kellerman (Eds.) *Emotion: Theory and research*. New York, Harcourt Brace, pp. 113-131, 1989.
- [14] S.J.,Young, “Large Vocabulary Continuous Speech Recognition”, *IEEE Signal Processing Magazine* 13, (5), 45-57, 1996.

Gathering and utilising domain knowledge in commercial computer games

Sander Bakkes

Pieter Spronck

Universiteit Maastricht / MICC-IKAT
P.O. Box 616, NL-6200 MD Maastricht, The Netherlands
{s.bakkes, p.spronck}@micc.unimaas.nl

Abstract

In practice, adaptive game AI in commercial computer games is seldom implemented, because machine learning techniques require numerous trials to learn effective behaviour. To allow fast adaptation in games, in this paper we describe a means of learning that is inspired by the human capability to solve problems by generalising over limited experiences with the problem domain. We compare three approaches, namely straightforward case-based reasoning, situated case-based reasoning, and a k -nearest neighbour classifier. From the experimental results we conclude that the situated approach performs best, both in representing knowledge and generalising to similar situations.

1 Introduction

Computer-controlled agents in commercial computer games are often called ‘Non-Player Characters’ (NPCs). NPCs are typically situated in rich and complex virtual environments, which excel in visual realism and realistic physics simulation. In such an environment one would expect the NPCs to behave realistically (‘human-like’) too. However, the current state of the art in this respect leaves much to be desired [6]. One aspect of human-like behaviour of NPCs, namely the ability to adapt to changing circumstances, has been explored with some success in recent research [3, 5, 11]. This is called ‘adaptive game AI’.

When implemented successfully, adaptive game AI is able to (1) fix errors in programmed game AI, and (2) seek counter-tactics to human gameplay. It is imperative that adaptive game AI is based on domain-specific knowledge [7, 10]. Since a game’s engine will be regularly updated, especially in the popular Massive Multiplayer Online Games (MMOGs), domain knowledge on what sort of behaviour is successful in the current environment should be automatically gathered. Human beings are able to deduce successful behaviour effectively from observations [9]. Our goal is to endow NPCs with a similar capability.

Within a typical commercial computer-game environment most machine-learning techniques are unsuitable for our goal, since they require numerous trials, make numerous mistakes before obtaining successful behaviour, or are computationally intensive [10]. Humans are capable of reasoning reliably on a preferred course of action with only a few observations on the problem domain. This paper investigates to what extent NPCs can gather domain knowledge from a few observations, and immediately (i.e., without trials and without resource-intensive learning) utilise that knowledge to evoke effective behaviour.

The outline of the paper is as follows. The environment we use for our research is discussed in Section 2. In Section 3 we discuss how we gather domain knowledge in a case-base. Three approaches for utilising this domain knowledge are described in Section 4. Section 5 describes an experiment to test the performance of the three approaches and provides the experimental results. Section 6 concludes and looks at future work.

2 Simulated environment

The research discussed in this paper is focussed on designing approaches for online gathering and utilising domain knowledge in commercial computer-game environments. For our investigation we have created a simulator environment representing a simple game. The game is an obstacle course, in which an NPC has to travel from the bottom of a grid to the top of the grid. The grid is 12 cells wide and 100 cells high. Each cell of the grid can contain an object. The following six object types can be placed on the grid:

1. **NPC.** One NPC is located in the grid. It is the only object that can change its position. Initially the NPC is placed in a random empty location on the bottom row of the grid. To manoeuvre to its goal (i.e., the top row of the grid) it has three actions available: (1) to move to the cell directly to its upper left, (2) to move to the cell directly above it, and (3) to move to the cell directly to its upper right. As input, the NPC can observe all cells of the five rows above it. It has two properties: health and fitness. Initially it is provided with a health value of 100. The NPC ‘dies’ (i.e., is removed from the grid) when its health reaches zero. The fitness value of the NPC is determined when it either dies or reaches the top row of the grid. It is calculated as $\frac{S}{H-1}$, where S is the number of steps taken by the NPC, and H is the number of cells the grid is high. On the screen, the NPC object is visually represented by a light-blue colour.
2. **Wall.** All the leftmost and rightmost cells of the grid contain wall objects. When an NPC tries to enter a cell in which a wall object is located, it dies. On the screen, the wall object is visually represented by a black colour.
3. **Goal.** All the cells on the top row of the grid (except for those containing a wall object) contain goal objects. When an NPC enters a cell in which a goal object is located, its fitness is set to 1.0 and it is removed from the grid. On the screen, a goal object is visually represented by a grey colour.
4. **Tree.** An arbitrary number of trees can be located in the grid. Trees are treated as walls, i.e., when an NPC tries to enter a cell in which a tree is located, it dies. On the screen, a tree object is visually represented by a green colour.
5. **Turret.** An arbitrary number of turrets can be located in the grid. When an NPC is within reach of a turret (defined as, within a square of seven cells on each side and the turret at its centre), its health is reduced by 10. Furthermore, when the NPC enters a cell where a turret is located, it dies. On the screen, a turret object is visually represented by a brown colour.
6. **Mine.** An arbitrary number of mines can be located in the grid. NPCs can *not* observe mines. NPCs are allowed to move into a cell containing a mine. However, when that happens the mine ‘explodes’, reducing the NPC’s health by 33. On the screen, a mine object is visually represented by a red colour.

An NPC can co-exist with another object in a cell, but of all the other objects each cell can contain at most one. The three grids designed for our experiments are displayed in Figure 1.

3 Gathering domain knowledge in a case-base

Case-Based Reasoning (CBR) is a machine learning paradigm for reasoning over a collection of previously observed situations. In commercial games, the environments NPCs are situated in typically excel in richness and visual complexity. This poses a challenge for a case-based reasoning approach as the environment information that is gathered by the NPC should, for effective and rapid use, be (1) represented in such a way that stored cases can be reused for previously unconsidered situations, and (2) compactly stored in terms of the amount of retrievable cases [1].

In our simulation, we wish to store the observations of an NPC, with the action it took, and an assessment of the success of that action. Regarding the representation we decided to store observations in the form of *labelled state-action pairs*. Each case consists of an abstraction of the world

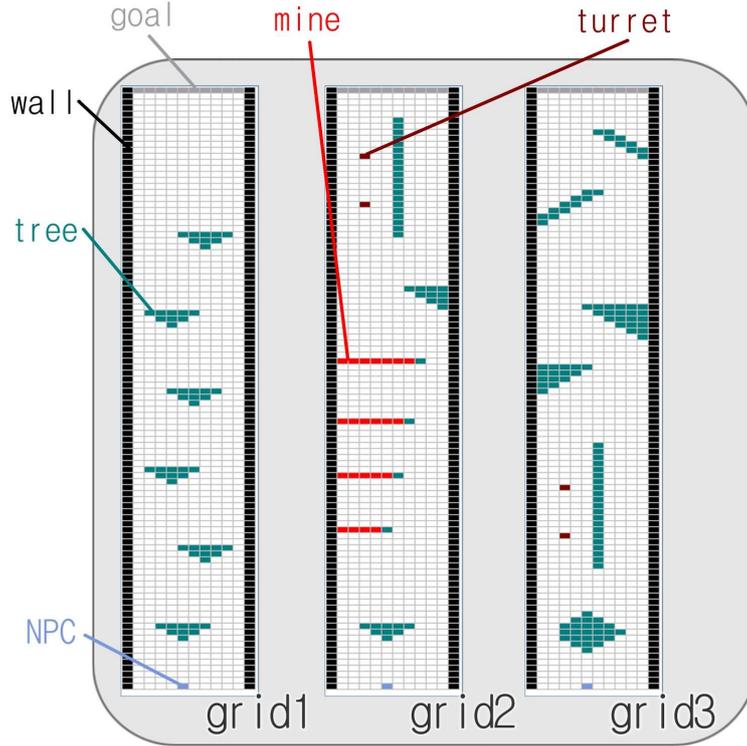


Figure 1: Three grids designed for our experiments. Note that mines are invisible to the NPC.

visible to the NPC, together with the NPC’s action. The environment state is abstracted relative to the NPC position (rather than from an absolute viewpoint). This abstraction is illustrated in Figure 2. Additionally, each state-action pair is labelled (and relabelled during a simulation trial) with the average observed fitness the stored action has led to when applied. The labelling process is algorithmically described below.

//Labelling algorithm

1. To determine the fitness value, observe the NPC until it reaches a goal cell or dies.
2. Retrieve NPC-observations for every step performed.
3. Check for each retrieved observation whether it is already stored.
- 4a. If so, update the fitness value of the retrieved observation by averaging over both fitness values.
- 4b. If not, store the particular observation with the action performed and the fitness obtained by the NPC.

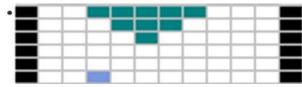
Regarding a compact storage of cases, we decided to reduce the possible number of cases stored in the case-base by only storing those state-action pairs that did not directly lead to a death in the next observable state. Accordingly, a labelled case-base arises of which all state-action pairs lead to either a local optimum or to the global optimum.

4 Proposed approaches for utilising domain knowledge

In this section we discuss three approaches we propose for utilising domain knowledge in our simulator. The approaches are based on the *reuse* process-step of the case-based reasoning paradigm, in which solutions from previously stored cases are mapped to a new case.

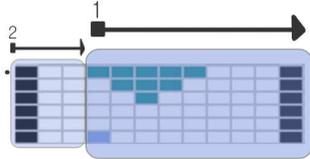
Our case base contains experiences of NPCs traversing grids. To utilise the case base for a new case, similarity values are calculated between the list of observations of the new case, and the lists of observations of all the cases in the case base. Of the most similar cases, the particular case with the highest fitness value is retrieved, and the action which was performed by the NPC for that case is selected for the new case.

Absolute observation abstraction



Observation: wall,empty,empty,tree,tree,tree,tree,tree,empty,empty,empty,wall,etc

Relative observation abstraction



Observation: tree,tree,tree,tree,tree,empty,empty,empty,wall,wall,empty,empty,etc.

Figure 2: Example of an absolute and relative observation abstraction. First the top-row cells are observed from left to right, subsequently the rows below are observed analogously and last the gathered information is stored. In our experiment we used the relative observation abstraction.

The three proposed approaches vary in their similarity-calculation algorithm. They are: (1) Straightforward CBR (Subsection 4.1), (2) Situated CBR (Subsection 4.2), and (3) k -Nearest Neighbour Classification (Subsection 4.3). The straightforward CBR approach was chosen as a way of baseline comparison. The situated CBR and the k -nearest neighbour classification approaches were chosen under the presumption of improved performance with regard to the use of relational features [4] and an increase in generalisation power, respectively.

4.1 Straightforward CBR

For straightforward CBR, similarity is calculated by a syntactic matching in which every identical observed feature receives the same weight. The algorithm scales the similarity value to 100% when all observed features are identical.

When similarity values are calculated for each case in the case-base, the matching algorithm subsequently retrieves the case with the highest similarity value within a *similarity-window*. The similarity-window defines the threshold similarity value of cases to be denoted as 'similar'. Should multiple cases be retrieved, the first cases with the highest fitness value is selected for execution. Should no cases be retrieved, the process is iterated by adjusting the similarity-window so that it allows for less-similar cases to be retrieved. This process is illustrated in pseudo-code below.

```

function calcSimilarity(oldcase , observation);
begin
  similarity := 0;
  for (all_features_in_observation) do
    if (observation.feature = oldcase.feature)
      then similarity := similarity + observation.feature.Weight;
  result := scale_to_100percent(similarity);
end;

function selectMostSimilarObservation ();
begin
  case_selected := false;
  fitness_best := 0;
  similarity_window := 100;
  repeat
  begin
    for (all_cases_in_the_casebase) do
      begin
        similarity_value := calcSimilarity(oldcase , currentObservation);
        if (similarity >= similarity_window)
          then

```

```

    if (oldcase.fitness > fitness_best)
      then begin case_selected := true; fitness_best := oldcase.fitness; end
    end;
  if not(case_selected) then
    begin similarity_window := similarity_window - 2; fitness_best := 0; end;
  end;
until (case_selected);
end;

```

4.2 Situated CBR

For situated CBR (i.e., CBR where environmental observations are evaluated from the point of the view of the NPC), similarity is calculated by a semantic matching in which features closer to the NPC are weighed heavier than features further from the NPC. For instance, an observational feature directly above an NPC is assigned a larger weight than a feature that is located two rows above the NPC. The weights assigned to observational features are given in Figure 3. For the situation portrayed in the figure, the observational feature on the left (the darker cell) will receive a relatively low weight as it is quite far from the NPC.

The similarity matching and case retrieval algorithm follow the same process as that of the straightforward CBR approach.

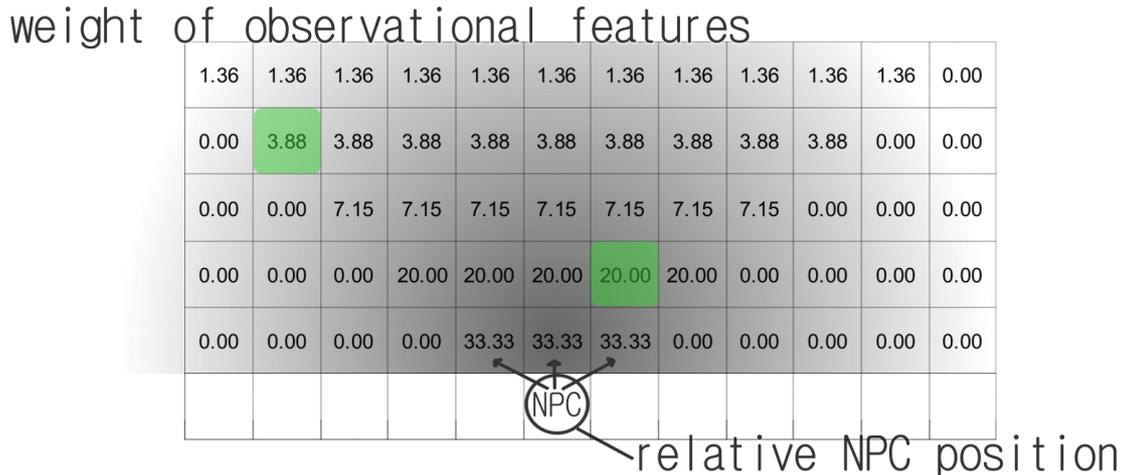


Figure 3: Situated similarity matching is established by weighing each observational feature with respect to the relative NPC position. The portrayed situation displays that part of the grid observed by the NPC. Allowed actions of the NPC are denoted with arrows.

4.3 k -Nearest Neighbour Classification

For the k -nearest neighbour classification approach, similarity is calculated as a generalisation over stored cases. A straightforward k -nearest neighbour algorithm is used to classify a new case based on the majority of the k -nearest neighbour category. This process consists of two steps:

1. Given a new case, the k -nearest neighbour algorithm retrieves the k number of state-action pairs of which the states are syntactically most similar to the new case (i.e., with highest similarity as calculated by the straightforward CBR approach). If state-action pairs exist with identical state-description, but with different observed action, only the state-action pair with the highest fitness is retrieved.
2. Action classification is established by a majority vote among the actions of the retrieved state-action pairs.

5 Experimental Study of the Approach

We performed several experiments in order to assess whether our approaches successfully gather and utilise domain knowledge, as to endow an NPC with effective behaviour in the designed grids.

5.1 Experimental Setup

Our first experiment consists of a comparative study of the three approaches discussed in Section 4. Our goal is to detect whether the approaches are able to induce successful NPC behaviour without feeding prior knowledge to the NPCs whose experiences are stored in the case base. The experimental procedure is as follows. Given a particular grid, we sample data of NPCs attempting to reach the goal objects by choosing only random moves. We performed separate tests with experiences stored of 2, 10 and 50 randomly behaving NPCs. Subsequently, we let an NPC use each of the three approaches, and we determine the fitness achieved. As different random behaviour leads to different observations (and thus different domain knowledge gathered), for each grid and each approach, the results are averaged over 30 repetitions of the test. Regarding the k -nearest neighbour classification algorithm, the k -value was chosen by assessing five different k -values for each trial, and selecting the best performing value for reference. The offered k values were 3, 5, 10, 25 and 50.

Secondly, we tested how each of the three approaches would generalise to different environments (i.e., test whether knowledge learned in one environment can be transferred to another environment). We created a case base for grid3 by running 50 randomly-behaving NPCs. We then used the three approaches to control an NPC that used this case base on both the first two grids. We chose grid3 for gathering knowledge as it is the most difficult grid for an NPC to traverse. For the second experiment, we repeated the process 100 times for each of the grids and each of the approaches.

5.2 Results

In Table 1 an overview of the experimental results is given. They reveal that, as may be expected, all proposed knowledge utilisation approaches endow an NPC with more effective behaviour whenever more gathered domain knowledge is available to the utilisation process. The first experiment revealed that, when training and testing on the same grid, both the straightforward CBR and situated CBR approach significantly outperform the k -nearest neighbour classification approach. The performance of the straightforward CBR and situated CBR approach do not significantly differ in this setting.

Experimental results of the second experiment, however, reveal that when training and testing on different grids for the purpose of generalising over domain knowledge, the performances of the straightforward CBR and situated CBR approach differ. When transferring domain knowledge from grid3 to grid1 and grid2, the situated CBR approach outperforms the straightforward CBR approach with statistical reliabilities of 94% and 79%, respectively [2].

Note that the k -nearest neighbour classification approach is the only approach where the performance does not significantly degrade when transferring domain knowledge. This is an important issue with regard to generalisation. However, despite the stability of the k -nearest neighbour classification approach, it performs worst of all three approaches as for knowledge transfer from grid3 to grid1 and grid2 it is even outperformed by the straightforward CBR approach with statistical reliabilities of 51% and 92%, respectively [2].

6 Conclusions and Future Work

In this paper we discussed a means of gathering knowledge in the domain of commercial computer games. Three approaches were proposed to utilise gathered domain knowledge stored in a case base. These three approaches are: (1) straightforward case-based reasoning, (2) situated case-based reasoning, and (3) k -nearest neighbour classification. From the results of the experiments that test the proposed approaches, we draw the conclusion that the situated case-based reasoning approach performs best in the simulated commercial computer-game environment.

	<i>NPC's observed</i>	<i>Straightforward CBR</i>	<i>Situated CBR</i>	<i>kNN</i>	
Grid1 (30 repetitions)	2	0.23 (0.27)	0.26 (0.31)	0.17 (0.20)	<i>k=3</i>
	10	0.55 (0.40)	0.60 (0.39)	0.16 (0.17)	<i>k=5</i>
	50	0.66 (0.33)	0.70 (0.34)	0.35 (0.37)	<i>k=3</i>
Grid2 (30 repetitions)	2	0.25 (0.21)	0.24 (0.21)	0.18 (0.21)	<i>k=10</i>
	10	0.50 (0.28)	0.48 (0.29)	0.27 (0.24)	<i>k=3</i>
	50	0.55 (0.24)	0.55 (0.22)	0.35 (0.26)	<i>k=3</i>
Grid3 (30 repetitions)	2	0.18 (0.17)	0.18 (0.17)	0.12 (0.13)	<i>k=3</i>
	10	0.38 (0.21)	0.41 (0.26)	0.20 (0.21)	<i>k=10</i>
	50	0.60 (0.19)	0.61 (0.24)	0.37 (0.26)	<i>k=25</i>
Knowledge transfer Grid 3 to grid 1 (100 repetitions)	50	0.33 (0.29)	0.40 (0.34)	0.33 (0.34)	<i>k=10</i>
Knowledge transfer Grid 3 to grid 2 (100 repetitions)	50	0.43 (0.25)	0.46 (0.27)	0.38 (0.26)	<i>k=25</i>

Table 1: Comparison of the performance of the three approaches. The performance is expressed in terms of the mean fitness and the standard deviation, denoted as ‘mean (standard deviation)’

Our findings on the approaches’ capabilities to transfer domain knowledge showed that the situated case-based reasoning approach performed best, but the k -nearest neighbour classification approach gave the most stable results. For future work, we aim to extend the situated case-based reasoning approach with a means of stable generalisation, inspired by the (distance-weighted [8]) k -nearest neighbour classification approach. We also intend to extend our research to actual commercial computer games, so to analyse the effect of our approach and to demonstrate its practical applicability.

Acknowledgements

This research was funded by a grant from the Netherlands Organization for Scientific Research (NWO grant No 612.066.406).

References

- [1] Agnar Aamodt and Enric Plaza. Case-based reasoning : Foundational issues, methodological variations, and system approaches. *AI Communications*, 7(1), March 1994.
- [2] Paul R. Cohen. *Emperical Methods for Artificial Intelligence*. The MIT Press, 1995. ISBN 0-262-03225-2.
- [3] Pedro Demasi and Adriano J. de O. Cruz. Online coevolution for action games. *International Journal of Intelligent Games and Simulation*, 2(3):80–88, 2002.
- [4] Kurt Driessens and Saso Dzeroski. Integrating guidance into relational reinforcement learning. *Mach. Learn.*, 57(3):271–304, 2004.
- [5] Thore Graepel, Ralf Herbrich, and Julian Gold. Learning to fight. In Quasim Mehdi, Norman Gough, and David Al-Dabass, editors, *Proceedings of Computer Games: Artificial Intelligence, Design and Education (CGAIDE 2004)*, pages 193–200. University of Wolverhampton, 2004.
- [6] John E. Laird and Michael van Lent. Human-level AI’s killer application: Interactive computer games. *AI Magazine*, 22(2):15–25, 2001.
- [7] John Manslow. *Learning and adaptation*, pages 557–566. AI Game Programming Wisdom. Charles River Media, Inc., 2002. Steven Rabin, editor.

- [8] Tom M. Mitchell. *Machine Learning*. McGraw-Gill Series in Computer Science, 1997. ISBN 0-07-042807-7.
- [9] Brian H. Ross. Some psychological results on case-based reasoning. In Morgan Kaufmann, editor, *Proceedings of the Case-Based Reasoning Workshop*, pages 144–147. DARPA, Pensacola Beach, 1989.
- [10] Pieter Spronck. A model for reliable adaptive game intelligence. In David W. Aha, Hector Munoz-Avila, and Michael van Lent, editors, *Proceedings of the IJCAI-05 Workshop on Reasoning, Representation, and Learning in Computer Games*, pages 95–100. Naval Research Laboratory, Navy Center for Applied Research in Artificial Intelligence, Washington, DC, 2005.
- [11] Pieter Spronck, Ida Sprinkhuizen-Kuyper, and Eric Postma. Online adaptation of game opponent AI with dynamic scripting. *International Journal of Intelligent Games and Simulation*, 3(1):45–53, 2004.

NEXT-GENERATION LANGUAGE GAMES: THE GUESSING GAME REVISITED

Joris Bleys ^a

^a *Artificial Intelligence Laboratory, Vrije Universiteit Brussel,
Pleinlaan 2, 1050 Brussels*

Abstract

In this paper we show how a *Guessing Game* can be reimplemented combining two recently developed formalisms: *Fluid Construction Grammar* and *Incremental Recruitment Language*. The main difference from older experiments is that the meaning of a sentence is conceived as a semantic constraint program which the speaker wants the hearer to execute and the use of a simple grammar expressed by suffixes. In this paper we show which constraints might be involved and how a semantic constraint program can be encoded in and decoded from an utterance and show how the agents can learn how to express these semantic programs. We conclude by showing results of a grounded multi-agent experiment using a Talking Heads set-up.

1 Introduction

The *Language Game* paradigm [3] has proven to be very useful in researching the emergence of artificial languages. It has for example been shown how populations of agents are able to cope with an open-ended environment and how conceptual categories can be coordinated through language [6]. More recently, fascinating experiments have shown how agents can cope with complex environments in which perspective is crucial.

In most of the previous experiments, the meaning was represented by means of propositional logic. However, it is clear that language is capable of expressing more complex meaning. For example, propositional logic is not suitable for expressing the meaning of sentences like “very big” as it does not allow for a propositional which has as argument another propositional. In the experiments we are recently conducting, the agents are capable of generating more complex meanings.

These experiments adopt the *procedural semantics* approach [10] in which the meaning of a sentence is a *semantic program* which the speaker wants the hearer to execute. For this purpose, the speaker first needs to generate such a program. This program is encoded into an utterance which is conveyed to the hearer. The hearer needs to reconstruct this semantic program after which it should be able to execute it. This kind of interaction implements a semiotic cycle as depicted in Figure 1.

The main *communicative goal* of a *Guessing Game* [4] is to draw the hearer’s attention to one (or more) object(s) in the shared context which we call the *topic*. The speaker has to generate a sequence of filtering operations which the hearer has to apply in order to be able to identify the *topic*. Suppose the context consists of a big square s_b , a small square s_s and a circle c and the speaker wants to draw attention to the big square s_b . A valid sequence of operations might be to start from all objects in the context $\{s_b, s_s, c\}$ and retain only those that are similar to the prototype of a square $\{s_b\}$. Next the average size of the remaining objects is calculated and the elements which have a size larger than this average are preserved $\{s_b\}$. Finally, there should be only one such element (denoted by “the”) which is selected from this set s_b . This program is then encoded into an utterance which the hearer has to decode and execute to be able to identify the *topic*.

The general approach to express a *semantic program* that I pursue is that grammar expresses hints for which program needs to be invoked and that the lexicon introduces *semantic items* which serve as arguments to this program.

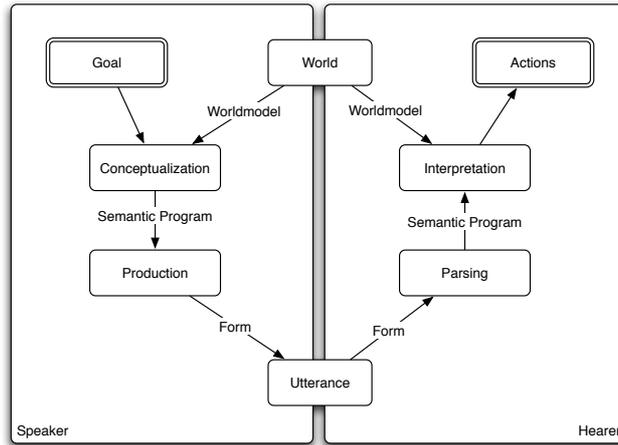


Figure 1: The semiotic cycle of a language game. The speaker chooses a communicative goal and generates a semantic program which the hearer should execute to reach that goal. The speaker encodes this semantic program in language. The hearer has to reconstruct this program after which it should be able to apply it in its current world-model. The speaker checks whether the hearer achieves the goal.

2 Conceptual-Intentional System

In *procedural semantics*, the process of conceptualisation becomes the generation of a *semantic program* that enables the hearer to achieve the goal the speaker had in mind. Interpretation becomes the execution of this program. The main reasons to adopt this approach is that it has a higher degree of expressivity than standard propositional logic and that it provides an open system in which all imaginable computational procedures can be added in a natural way.

Another branch in computer science we have borrowed ideas from is *constraint programming* [2]. In the *language game* paradigm, all agents should be capable of both producing and interpreting an utterance. As we would like to use the same procedures in both tasks (bidirectionality) and both tasks start from different data (production starts at the goal and interpretation at the utterance) we conclude that a data-driven control flow in our programs would be preferable to standard preprogrammed control flow.

We have combined these ideas in a programming language named *Incremental Recruitment Language* in which *semantic programs* can be defined. The basic procedures of this language are *primitive constraints* which have one or more *slots* (for example similar to a mathematical addition $?a = ?b + ?c$). These can be combined into a larger *semantic program* when the variables to which these slots are assigned to are shared among several *primitive constraints* (for example similar to $?a = ?b + ?c, ?b - ?a = 1$). Each variable is typed constraining the possible values of each variable (like for instance the domain of natural numbers).

2.1 Semantic Programs

As mentioned above, the speaker has to come up with a *semantic program* which it wants the hearer to execute during interpretation. For this purpose, each agent maintains for each *communicative goal* an ordered list of *semantic programs* which have been successful in past interactions. The ordering currently applied is the number of *primitive constraints* it involves, preferring programs which require less *primitive constraints*¹. Each program is tested for its validity within the current context and the first program that is valid is selected. When the inventory of *semantic programs* to reach the goal are exhausted, the conceptual-intentional system generates a new valid program suitable for the current context. The automatic generation of programs is a complicated and interesting problem but it is out of the scope of the current paper. An interested reader is kindly referred to [8] for more information.

The way programs are tested for validity in the current context is by checking whether they can achieve the goal. In the case of the guessing game, this would be checking whether a certain variable has as its value the object(s) which constitute the topic. When checking a *semantic program* for its validity, a list of valid possible values for each variable in the program is

¹Another way of ordering might be the communicative success a program had in previous interactions.

maintained. Each *primitive constraint* is capable of changing the lists of values of the variables to which its slots are assigned. A program is valid when each variable in it is bound to at least one value. If for any variable this list becomes empty the program is invalid. When more than one value remains in the list the program is considered to be valid but ambiguous².

The hearer has to reconstruct the *semantic program* during interpretation and apply it. For this purpose it first binds the values encoded in the utterance to the *arguments* of the reconstructed program. Next, it tries to validate the *semantic program* using the same algorithm as described above. If this validation is successful and unambiguous the hearer can apply the actions encoded in it. In the case of a guessing game this could be done by a means of a pointing gesture. The speaker can verify whether the hearer was able to identify the right objects. If any step in this procedure fails, the speaker performs the required actions it wanted the hearer to perform which enables the hearer to generate a valid *semantic program* using the program-generating algorithm touched upon above.

For each of these *semantic programs* a *call pattern* is generated. This *call pattern* starts with a unique identifier followed by the *arguments* of the *semantic program*. The arguments of this program are both the variables which are needed for testing whether the goal has been reached and all the external variables to this program. It is this *call pattern* together with the *semantic items* which are the values of its external variables I consider to be the meaning that needs to be encoded into a utterance and the other way round.

2.2 Semantic Programs for a Guessing Game

The *semantic programs* that are generated by the conceptual-intentional system can be captured using the following regular-expression-like statement:

```
(equal-to-sensory-context ?set-1)
(restrict-set-channel-average ?set-2 ?set-1 ?comp-1)*
(unique-element ?obj ?set-n)?
```

Each program consists of exactly one EQUAL-TO-SENSORY-CONTEXT which constrains the value of its only argument to be an object-set which contains all the objects in the sensory context. This variable can be used in a RESTRICT-SET-CHANNEL-AVERAGE which constrains its first argument to be subset of its second argument which contains all objects which have a value which is greater or less than the average value of all objects in the superset specified by the third argument. This argument is of type *comparison* and specifies both a relation and a channel. This filtering might be repeated by adding more RESTRICT-SET-CHANNEL-AVERAGE constraints. Finally, when the topic consists of only one object, it is selected using a UNIQUE-ELEMENT constraint which constrains its first argument to be the only element in the singleton specified by its second argument.

2.3 Primitive Constraints for a Guessing Game

As mentioned above, the *primitive constraints* are the basic building blocks of a *semantic program*. When defining a primitive constraint it is necessary to steer away from possible combinatorial explosions without sacrificing on its computational power which would render the primitive constraint useless. Implementing a constraint boils down to defining what should happen for each constellation of variables of open (when a variable has no possible values yet) and closed domains (in which the domain of a variable is limited to a restricted set of values). Any invalid combination of values is removed from the domain of the corresponding variables.

I will shortly discuss the implementation of the RESTRICT-SET-CHANNEL-AVERAGE primitive constraint. It has three slots: *?target-set*, *?source-set* and *?comparison*. The first two slots are of type *object-set* and the last is of type *comparison*. When *?source-set* has no value no computation takes place to circumvent combinatorial explosion. When only *?source-set* is known, for each channel that each object in the source-set has a value for and for each possible comparison relation (greater or smaller than) the corresponding *?target-set* is computed. When both *?source-set* and *?target-set* have a value, all suitable values for *?comparison* are computed. When both *?source-set* and *?comparison* have a value, the corresponding value for *?target-set* is calculated and when all three slots have a value, it is checked whether it is a valid combination of values.

²The actual validation algorithm is far more complex than described, as for instance it needs to be capable of handling dependencies between derived values as well. For more information please read [8].

3 Language System

As language system I have deployed Fluid Construction Grammar (FCG) [5, 7]³. FCG is a grammar formalism which is developed to research the origins and evolution of languages. It is inspired by construction grammar which states that all lexical and grammatical knowledge is stored in form-meaning pairings. These pairings are implemented in *rules* which can be matched to the current semantic and syntactic structures which have been built up during production and parsing. When a rule matches, additional information is merged into the structures which might enable other rules to match. Generally, in production the left pole of a rule is matched and the right pole is merged and in parsing it is the other way round. This process is described in more formal detail in [7]. The main hallmarks of FCG are bi-directionality (each rule should be applicable for both production and interpretation) and fluidity (the rules should be able to cope with incomplete grammar knowledge). A short introduction to the formalism is provided in [1] and I assume the reader has some basic knowledge of how it works.

As mentioned in the introduction, the way *semantic programs* are encoded in an utterance is by using the lexical entries for expressing *semantic items* and grammatical entries hinting at how these items should be used. The way it is implemented is by using stems for expressing the *semantic items* to which a suffix is added specifying which program and which slot in this program this item should be bound to. An utterance in this artificial language might look like “bodali-kimuna wabado-xiuxiu” in which the stems refer to a comparison value (meaning a channel and a comparison relation) and the suffixes specify indirectly through syntactical categories both the *call pattern* and which argument of that *call pattern* the corresponding stem should bind to.

In order to implement this approach, I have used three types of rules. *Lex-stem rules* encapsulate the associations between stems and comparison values. *Morph rules* specify the relation between suffixes and syntactic categories. These syntactic categories are used to constrain the *construction rules* which are used to link the comparison values to the arguments of the *semantic program*.

3.1 Lex-stem Rules

As stated above, the *lex-stem rules* relate stems with comparison values. Below is given an example which relate the stem “wabado” to the the comparison value “>-area” which in English might correspond to a word like “big”. Note also the use of the J-operator which adds new hierarchical units to the features during merging [1].

```
(def-lex-stem-rule "wabado-lex-stem-rule"
  ((?top-unit
    (meaning (== (comparison ?comparison >-area))))
   ((J ?filler-unit ?top-unit)
    (referent ?comparison)))
  <-->
  ((?top-unit
    (syn-subunits (== ?filler-unit)))
   (?filler-unit
    (form (== (stem ?filler-unit "wabado"))))))
```

3.2 Morph Rules

Morph rules are very similar to the *lex-stem rules* discussed above, but now it associates a particular suffix “xiuxiu” to a syntactical category “syncat-1”. Also note that the J-operator is on the other side of the rule in order to make sure that the rules match in both production and parsing.

```
(def-morph-rule "syncat-1-morph-rule"
  ((?top-unit
    (syn-subunits (== ?new-unit)))
   (?new-unit
    (syn-cat (== (syncat-1 ?new-unit))))
  <-->
  ((?top-unit
    (form (== (stem ?new-unit ?stem)
```

³The source code of FCG is freely available at <http://arti.vub.ac.be/FCG>.

```
(suffix ?new-unit "xiuxiu"))))
((J ?new-unit ?top-unit)))))
```

3.3 Construction Rules

The *construction rules* show how the syntactical categories can be used to identify which comparison values should fit into which arguments of the *semantic program*. This is done by making variable equalities explicit. The same unit is at the same time constrained by the syntactical categories. For example, the second argument of the *call-pattern* “identify-object-2” should be equal to the referent of the unit which belongs to the syntactical category “syncat-1”.

```
(def-con-rule "identify-object-2-rule"
  ((?top-unit
    (meaning (== (identify-object-2 ?object ?filler-1 ?filler-2)))
    (sem-subunits (== ?filler-unit-1 ?filler-unit-2)))
    ((J ?new-unit ?top-unit)
     (referent ?object))
    (?filler-unit-1
     (referent ?filler-1)
     (meaning (== (comparison ?filler-1 ?comparator-1))))
    (?filler-unit-2
     (referent ?filler-2)
     (meaning (== (comparison ?filler-2 ?comparator-2))))))
  <-->
  ((?top-unit
    (syn-subunits (== ?filler-unit-1 ?filler-unit-2)))
    ((J ?new-unit ?top-unit)
     (?filler-unit-1
      (syn-cat (== (syncat-1 ?filler-unit-1))))
     (?filler-unit-2
      (syn-cat (== (syncat-2 ?filler-unit-2))))))
```

4 Learning Operators

In order to achieve high communicative success the agents need to arrive at a convention on both lexical and grammatical knowledge. In our experiments this agreement needs to be established on both the meaning of the stems and the syntactic categories to which the suffixes belong. There does not need to be an explicit agreement on which syntactic categories correspond to which arguments of the *call-pattern* captured in *construction rules* as these are personal and unique for each agent in order to reduce the learning complexity. For each rule a strength is maintained which represents the rules usefulness during previous interactions. During production or parsing it is possible that more than one rule matches. In this case only the most promising (eg. the one with the highest strengths) is merged.

The main learning mechanism for arriving at agreement within a population of agents is *lateral inhibition* [9]. The strength of the rules are adapted at the end of each interaction depending on its outcome. When the interaction was successful the strength of each rule that was used is increased and the strength of other rules which also have matched but have not been used are decreased. When the interaction was a failure the scores of the used rules are decreased.

4.1 The speaker’s Invention Operators

All the learning operators of the speaker imply the invention of new rules similar to those described above when it has no rule to cover some part of the meaning or when it has no rule for some syntactical category. The fixes are applied in the linear order in which they are described here.

The invention of a lex-stem rule happens as follows. First the speaker needs to detect the need of an additional lex-stem rule. It is possible to detect this by inspecting the semantic structure after application of the lex-stem rules already known to the agent. If in the top unit of this structure remains a semantic item it means that it didn’t have a rule for this element because otherwise it would have been moved to a subunit as defined by the J-operator. For each semantic item that remains in the top unit it invents a new stem which it associates to with it in a new lex-stem rule.

Construction rules are invented in a similar fashion. The speaker can detect the need for a new construction rule if a call-pattern remains in the top unit of the semantical structure after application of the construction rules. For each external variable in the call pattern it invents a new syntactical category which it combines into a new construction rule.

The invention of morph rules is slightly more complicated. To detect whether there is a need for a new morph rule, the speaker has to detect units which have a syntactical category but no suffix. In this event the speaker invents a new suffix for each uncovered category and invents a morph rule in which these are associated.

4.2 The hearer's Adoption Operators

When the hearer is not able to reach the goal the speaker intended, either because it could not construct a complete semantic program or the validation algorithm failed, the speaker reveals the topic by means of pointing gestures. The hearer is capable of generating a semantic program which it could have applied to reach the goal for which it also tries to produce an utterance. Using the semantic and syntactic structures of both the failed interpretation task and the production task of the generated program it tries to reconstruct the rules that the speaker might have used. The priority with which these operators are applied is equal to the ordering in which they are described below.

The first learning operator covers the invention of lex-stem rules. First the hearer looks for uncovered semantic items in the syntactic structure of the production task using the same method as described for the speaker. Next it seeks for uncovered stems in the interpretation task by looking for units which have a stem but no semantic item. When there is exactly one uncovered semantic item and exactly one uncovered stem the hearer learns a new lex-stem rule associating these two. It would also be possible to try to learn when there are several uncovered semantic items and a equal number of uncovered stems but this would introduce unnecessary uncertainty.

The second learning operator concerns construction rules and is actually more like an invention operator as it only uses the semantic structure of the production task. It is identical to the operator for inventing construction rules used by the speaker as the association between syntactical categories and slots of a call pattern are personal to each agent.

The adoption of morph rules is slightly more complicated. First, the hearer searches for unknown suffixes used by the speaker by looking for suffixes which remain in the top unit after application of known morph rules in the interpretation task. For each unknown suffix the hearer tries to detect to which syntactical category it corresponds. First, for each suffix it detects to which stem it corresponded in the utterance of the speaker. Next, it looks into her production task to see to which syntactic categories the stem in this utterance belongs. When there is exactly one such syntactic category for a stem, the hearer adds a new morph rule relating these two. When there is more than one such syntactic category, for instance when the same semantic item is used more than once in the semantic program, the hearer learns nothing again to avoid unnecessary uncertainty.

The last learning operator concerns the invention of lex-stem rules when the hearer itself would use a stem different from the stem used by the speaker which renders the first learning operator incapable of learning new rules. This typically occurs in populations consisting of more than two agents where different agents might invent different stems for the same semantic item. In this case, the hearer associates the single unknown stem to the semantic item for which it is the least certain. For each semantic item in the production task the best matching rule (meaning of all matching rules the one with the highest strength) is selected. Out of these best matching rules the one with the minimal strength is chosen to be the item to which the unknown stem is associated.

5 Experiment

The communicative challenge the agents have to learn to cope with is similar to the *Guessing Game*: one agent needs to draw the attention of the other agents to one or more objects in a shared context. This context is provided by an interface to a original Talking Heads [4] set-up⁴. This set-up consists of two cameras directed to a whiteboard with coloured geometrical shapes attached to it. The segment-detection algorithm is capable of extracting eleven different channels, ranging from position over size to the average colour of the segments. In this shared

⁴Interested readers can consult <http://arti.vub.ac.be/research/language/th-interface/> for more information.

context the speaker has to draw the attention of the hearer to one or more objects. The population size in the experiments is fixed to five.

5.1 Analysis

The communicative success increases to around 95% (as shown in Figure 2). The reason why it never reaches its maximum success is that in some contexts the conceptual-intentional system is incapable to generate a *semantic program* for a random selected *topic*. Suppose the agents face a context consisting of two very similar objects (o_1, o_2) and one asimilar object o_3 . The randomly selected topic consists of o_1 and o_3 . No filtering operation is capable of discarding o_2 without discarding o_1 as well rendering the generation of a valid *semantic program* impossible.

The number of *lex-stem rules* first overshoots but then stabelizes around 26. The total number of possible *lexical items* is 23 (eleven channels combined with two possible relation functions and one denoting the lack of any lexical item). This behaviour is similar to previous observations (for instance [4]) and the overshoot stays below the expected overshoot of 57.5 (number of agents times the number of meanings divided by two) and although some synonyms remain they are expected to disappear from the agent’s lexicons. The expected number of *morph rules* is 12 (semantic programs having one to three arguments, with/without unique-element constraint) which is reached at the end of the simulation. The lack of apparent overshoot requires further mathematical attention. The number of *construction rules* as expected never decreases as there is no competition between syntactical categories and arguments of *semantic items* they represent.

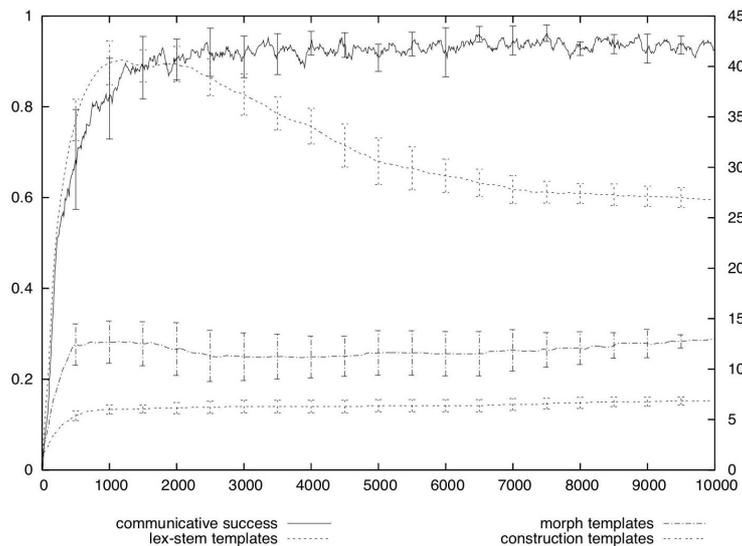


Figure 2: Measures of the experiment for a population of 5 agents averaged over 5 runs. The average number of each type of rule within the population is shown together with the communicative success of the population.

6 Discussion

The main goal of this paper is to make the readers more familiar with the recently developed formalisms, *Incremental Recruitment Language* and *Fluid Construction Grammar*, by showing how a well known language game, the guessing game, could be implemented using them. Another objective is to provide a proof of concept how these systems could interact by showing grounded experimental results. As this experiment tries to mimick the original *Guessing Game*, it is not intended to bring in new experimental results but rather tries to focus on how the *Guessing Game* could be implemented in a new fashion.

Like most other experiments in the domain of semiotic dynamics, this experiment reported in this paper makes a stack of assumptions. For instance it is assumed that the agents share an attentional frame, share their learning mechanisms and are capable of performing perfect pointing. This experiment adds one assumption to this stack: the *semantic programs* speaker

and hearer generate in one game should be as similar as possible. In the current experiment this is already achieved because of the bias towards *semantic programs* that consist of a minimal number of *primitive constraints*. These kind of alignment mechanisms might not be suitable for more complex experiments and should be extended, for example by a mechanism that assign a confidence measure to each generated program.

This paper shows only a glimpse of this *procedural semantics* approach's full potential. Each level can be extended by anyone who is interested in doing research in the origins of artificial languages. New *primitive constraints* can be written to implement any known computational procedure (like for instance set operators). Other *rules* can be crafted to research specific aspects of emergent languages and more general learning operators can be designed which are capable of expressing any *semantic program*.

7 Acknowledgements

The original implementation of *Incremental Recruitment Language* was written and conceived by Prof. Luc Steels and is currently being reimplemented by Wouter Van den Broeck. *Fluid Construction Grammar* was designed and implemented by Prof. Luc Steels and Joachim De Beule. I would like to thank them together with the anonymous reviewers, Bart De Vylder, Martin Loetzch, Remi van Trijp and Pieter Wellens for their helpful discussions and insightful comments. The author of this paper is funded by a doctoral grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen).

References

- [1] J. De Beule and L. Steels. Hierarchy in fluid construction grammar. In Ulrich Furbach, editor, *Proceedings of the 28th Annual German Conference on AI, KI 2005*, volume 3698 of *Lecture Notes in Artificial Intelligence*, pages 1–15, Berlin Heidelberg, 2005. Springer Verlag.
- [2] K. Marriot and Stuckey P.J. *Programming with Constraints: An Introduction*. MIT Press, Cambridge Ma., 1998.
- [3] L. Steels. Emergent adaptive lexicons. In P. Maes, M. Mataric, J.-A. Meyer, J. Pollack, and S.W. Wilson, editors, *From Animals to Animats 4: Proceedings of the Fourth International Conference On Simulation of Adaptive Behavior*, pages 562–567, Cambridge, MA, 1996. The MIT Press.
- [4] L. Steels. *The Talking Heads Experiment. Volume 1. Words and Meanings*. Laboratorium, Antwerpen, 1999.
- [5] L. Steels. Constructivist development of grounded construction grammars. In W. Daelemans, editor, *Proceedings Annual Meeting of Association for Computational Linguistics*, 2004.
- [6] L. Steels and T. Belpaeme. Coordinating perceptually grounded categories through language: A case study for colour. *Behavioral and Brain Sciences*, 28:469–529, 2005.
- [7] L. Steels and J. De Beule. Unify and merge in fluid construction grammar. In Nehaniv L. Lyon, C. and A. Cangelosi, editors, *Emergence and Evolution of Linguistic Communication*, Lecture Notes in Computer Science. Springer-Verlag, Berlin, 2006.
- [8] L. Steels and J. Bleys. Planning what to say: Second order semantics for fluid construction grammars. In A. Bugarin Diz and J. Santos Reyes, editors, *Proceedings of CAEPIA 05. Lecture Notes in AI.*, Berlin, 2005. Springer Verlag.
- [9] L. Steels and F. Kaplan. Bootstrapping grounded word semantics. In T. Briscoe, editor, *Linguistic evolution through language acquisition: formal and computational models*, pages 53–74. Cambridge University Press, Cambridge, UK, 2002.
- [10] T. Winograd. *Understanding Natural Language*. Academic Press, 1972.

Norm Negotiation Power

Guido Boella ^a

Leendert van der Torre ^b

^a *University of Torino*

^b *University of Luxembourg*

Abstract

In social mechanism design, norm negotiation creates individual or contractual obligations fulfilling goals of the agents. The social delegation cycle distinguishes among social goal negotiation, obligation and sanction negotiation and norm acceptance. Power may affect norm negotiation in various ways, and we therefore introduce a new formalization of the social delegation cycle based on power and dependence, without referring to the rule structure of norms, actions, decision variables, tasks, and so on.

1 Introduction

Normative multiagent systems [14, 8, 5, 6] provide agents with abilities to automatically devise organizations and societies coordinating their behavior via obligations, norms and social laws. A distinguishing feature from group planning is that also sanctions and control systems for the individual or contractual obligations can be created. Since agents may have conflicting goals with respect to the norms that emerge, they can negotiate amongst each other which norm will be created.

The social delegation cycle [2, 7] explains the negotiation of new social norms from cognitive agent goals in three steps. First individual agents or their representatives negotiate social goals, then a social goal is negotiated in a social norm, and finally the social norm is accepted by an agent [13] when it recognizes it as a norm, the norm contributes to the goals of the agent, and it is obeyed by the other agents. A model of norm negotiation explains also what it means, for example, to recognize or to obey a norm, and how new norms interact with existing ones.

Power may affect norm negotiation in various ways, and we therefore propose to analyze the norm negotiation problem in terms of social concepts like power and dependence. Power has been identified as a central concept for modeling social phenomena in multi-agent systems by various authors [10, 11, 15, 21], as Castelfranchi observes both to enrich agent theory and to develop experimental, conceptual and theoretical new instruments for the social sciences [12].

To motivate our social-cognitive model, we contrast it with an abstract description of the social delegation cycle using game theoretic artificial social systems. The problem studied in artificial social systems is the design, emergence or more generally the creation of social laws. Shoham and Tennenholtz [16] introduce social laws in a setting without utilities, and they define *rational* social laws as social laws that improve a social game variable [17]. We follow Tennenholtz' presentation for stable social laws [19].

Moreover, we also contrast our social-cognitive model with existing highly detailed models of the social delegation cycle, like the ones we have proposed within normative multiagent systems [2, 7]. The challenge to define social mechanisms, as we see it, is to go beyond the classical game theoretic model by introducing social and cognitive concepts and a negotiation model, but doing so in a minimal way. In the model proposed in this paper we therefore keep goals and obligations abstract and we do not describe them by first-order (or propositional) logic or their rule structure, we do not introduce decisions, actions, tasks, and so on. Similar concerns are also mentioned by Wooldridge and Dunne in their qualitative game theory [20].

The layout of this paper is as follows. In Section 2 we discuss an abstract model of the social delegation cycle in Tennenholtz' game-theoretic artificial social systems. In Section 3 we define our social-cognitive conceptual model of multiagent systems in which we study and formalize the social delegation cycle, and in Section 4 we define the negotiation protocol. In Section 5 we formalize goal negotiation, in Section 6 we formalize norm negotiation, and in Section 7 we formalize the acceptance relation.

2 Social delegation cycle using artificial social systems

In Tennenholtz' game-theoretic artificial social systems, the goals or desires of agents are represented by their utilities. A game or multi-agent encounter is a set of agents with for each agent a set of strategies and a utility function defined on each possible combination of strategies. Tennenholtz only defines games for two agents to keep the presentation of artificial social systems as simple as possible, but he also observes [19, footnote 4] that the extension to the multi-agent case is straightforward.

Definition 1 A game (or a multi-agent encounter) is a tuple $\langle N, S, T, U_1, U_2 \rangle$, where $N = \{1, 2\}$ is a set of agents, S and T are the sets of strategies available to agents 1 and 2 respectively, and $U_1 : S \times T \rightarrow \mathbb{R}$ and $U_2 : S \times T \rightarrow \mathbb{R}$ are utility functions for agents 1 and 2, respectively.

The social goal is represented by a minimal value for the social game variable. Tennenholtz [19] uses as game variable the maximin value. This represents safety level decisions, in the sense that the agent optimizes its worst outcome assuming the other agents may follow any of their possible behaviors. See Tennenholtz' paper for a discussion why this is natural in many multi-agent systems, where a payoff corresponds to the achievement of a particular user's specification.

Definition 2 Let S and T be the sets of strategies available agent 1 and 2, respectively, and let U_i be the utility function of agent i . Define $U_1(s, T) = \min_{t \in T} U_1(s, t)$ for $s \in S$, and $U_2(S, t) = \min_{s \in S} U_2(s, t)$ for $t \in T$. The maximin value for agent 1 (respectively 2) is defined by $\max_{s \in S} U_1(s, T)$ (respectively $\max_{t \in T} U_2(S, t)$). A strategy of agent i leading to the corresponding maximin value is called a maximin strategy for agent i .

The social norm is represented by a social law, which has been characterized as a restriction of the strategies available to the agents. It is *useful* with respect to an efficiency parameter e if each agent can choose a strategy that guarantees it a payoff of at least e .

Definition 3 Given a game $g = \langle N, S, T, U_1, U_2 \rangle$ and an efficiency parameter e , we define a social law to be a restriction of S to $\bar{S} \subseteq S$, and of T to $\bar{T} \subseteq T$. The social law is useful if the following holds: there exists $s \in \bar{S}$ such that $U_1(s, \bar{T}) \geq e$, and there exists $t \in \bar{T}$ such that $U_2(\bar{S}, t) \geq e$. A (useful) convention is a (useful) social law where $|\bar{S}| = |\bar{T}| = 1$.

A social law is *quasi-stable* if an agent does not profit from violating the law, as long as the other agent conforms to the social law (i.e., selects strategies allowed by the law). Quasi-stable conventions correspond to Nash equilibria.

Definition 4 Given a game $g = \langle N, S, T, U_1, U_2 \rangle$, and an efficiency parameter e , a quasi-stable social law is a useful social law (with respect to e) which restricts S to \bar{S} and T to \bar{T} , and satisfies the following: there is no $s' \in S - \bar{S}$ which satisfies $U_1(s', \bar{T}) > \max_{s \in \bar{S}} U_1(s, \bar{T})$, and there is no $t' \in T - \bar{T}$ which satisfies $U_2(\bar{S}, t') > \max_{t \in \bar{T}} U_2(\bar{S}, t)$.

The efficiency parameter can be seen as a social kind of *utility aspiration level*, as studied by Simon [18]. Such aspiration levels have been studied to deal with limited or resource-bounded reasoning, and have led to the development of goals and planning in artificial intelligence; we therefore use a goal based ontology in this paper. The three steps of the social delegation cycle in this classical game-theoretic setting can be represented as follows. Goal negotiation implies that the efficiency parameter is higher than the utility the agents expect without the norm, for example represented by the Nash equilibria of the game. Norm negotiation implies that the social law is useful (with respect to the efficiency parameter). The acceptance relation implies that the social law is quasi-stable.

We use the game-theoretical model to motivate our conceptual model of normative multiagent systems. Due to the uniform description of agents in the game-theoretic model, it is less clear how to distinguish among kinds of agents. For example, the unique utility aspiration level does not distinguish the powers of agents to negotiate a better deal for themselves than for the other agents. Moreover, the formalization of the social delegation cycle does neither give a clue how the efficiency parameter is negotiated, nor how the social law is negotiated. For example, the goals or desires of the agents as well as other mental attitudes may play a role in this negotiation. There is no sanction or control system in the model (adding a normative system to encode enforceable social laws to the artificial social system complicates the model [3]). Finally, an additional drawback is that the three ingredients of the model (agent goals, social goals, and social laws) are formalized in three completely different ways.

3 Power viewpoint on normative multiagent systems

In this paper we follow the definition of power as the ability of agents to achieve goals. Thus, an agent is more powerful than another agent if it can achieve more goals.

For example, in the so-called power view on multi-agent systems [1], a multi-agent system consists of a set of agents (A), a set of goals (G), a function that associates with each agent the goals the agent desires to achieve (*goals*), and a function that associates with each agent the sets of goals it can achieve (*power*). To be precise, since goals can be conflicting in the sense that achieving some goals may make it impossible to achieve other goals, the function *goals* returns a set of set of goals for each set of agents. Such abstract structures have been studied as qualitative games by Wooldridge and Dunne [20], though they do not call the ability of agents to achieve goals their power. To model trade-offs among goals of agents, we introduce a priority relation among goals.

Definition 5 Let a multiagent system be a tuple $MAS = \langle A, G, goals, power, \succeq \rangle$ where:

- the set of agents A and the set of goals G are two finite disjoint sets;
- $goals : A \rightarrow 2^G$ is a function that associates with each agent the goals the agent desires to achieve;
- $power : 2^A \rightarrow 2^{2^G}$ is a function that associates with each set of agents the sets of goals the set of agents can achieve;
- $\succeq : A \rightarrow \subseteq 2^G \times 2^G$ is a function that associates with each agent a partial pre-ordering on the sets of his goals;

To model the role of power in norm negotiation, we extend the basic power view in a couple of ways. To model obligations we introduce a set of norms, we associate with each norm the set of agents that has to fulfill it, and of each norm we represent how to fulfill it, and what happens when it is not fulfilled. In particular, we relate norms to goals in the following two ways.

- First, we associate with each norm n a set of goals $O(n) \subseteq G$. Achieving these normative goals $O(n)$ means that the norm n has been fulfilled; not achieving these goals means that the norm is violated. We assume that every normative goal can be achieved by the group, i.e., that the group has the power to achieve it.
- Second, we associate with each norm a set of goals $V(n)$ which will not be achieved if the norm is violated (i.e., when its goals are not achieved), this is the sanction associated to the norm. We assume that the group of agents does not have the power to achieve these goals.

Since we do not want to exclude norms without sanctions, we do not assume that the sanction affects at least one goal of each agent of the group the obligation belongs to.

Definition 6 Let a normative multi-agent system be a tuple $\langle MAS, N, O, V \rangle$ extending a multiagent system $MAS = \langle A, G, goals, power, \succeq \rangle$ where:

- the set of norms N is a finite set disjoint from A and G ;
- $O : N \times A \rightarrow 2^G$ is a function that associates with each norm and agent the goals the agent must achieve to fulfill the norm; We assume for all $n \in N$ and $a \in A$ that $O(n, a) \in power(\{a\})$;
- $V : N \times A \rightarrow 2^G$ is a function that associates with each norm and agent the goals that will not be achieved if the norm is violated by agent a ; We assume for each $B \subseteq A$ and $H \in power(B)$ that $(\cup_{a \in A} V(n, a)) \cap H = \emptyset$.

An alternative way to represent normative multiagent systems replaces the function *power* by a function representing dependencies between agents. For example, a function of minimal dependence can be defined as follows. Agent a depends on agent set $B \subseteq A$ regarding the goal g if $g \in goals(a)$, $g \notin power(a)$, $g \in power(B)$, and there is no $C \subset B$ such that $g \in power(C)$. Note that dependence defined in this way is more abstract than power, in the sense that we have defined dependence in terms of power, but we cannot define power in terms of dependence.

4 Generic negotiation protocol

A negotiation protocol is described by a set of sequences of negotiation actions which either lead to success or failure. In this paper we only consider protocols in which the agents propose a so-called deal, and when an agent has made such a proposal, then the other agents can either accept or reject it (following an order \succ) of the agents). Moreover, they can also end the negotiation process without any result.

Definition 7 (Protocol) *A negotiation protocol is a tuple $\langle Ag, deals, actions, valid, finished, broken, \succ \rangle$, where:*

- *the agents Ag , deals and actions are three disjoint sets, such that actions = $\{propose(a, d), accept(a, d), reject(a, d) \mid a \in Ag, d \in deals\} \cup \{breakit(a) \mid a \in Ag\}$.*
- *valid, finished, broken are sets of finite sequences of actions.*

We now instantiate this generic protocol for negotiations in normative multiagent systems. We assume that a sequence of actions (a history) is valid when each agent does an action respecting this order. Then, after each proposal, the other agents have to accept or reject this proposal, again respecting the order, until they all accept it or one of them rejects it. When it is an agent's turn to make a proposal, it can also end the negotiation by breaking it. The history is *finished* when all agent have accepted the last deal, and *broken* when the last agent has ended the negotiations.

Definition 8 (NMAS protocol) *Given a normative multiagent system $\langle MAS, N, O, V \rangle$ extending a multi-agent system $MAS = \langle A, G, goals, power, \geq \rangle$, a negotiation protocol for NMAS is a negotiation protocol $NP = \langle A, deals, actions, valid, finished, broken, \succ \rangle$, where:*

- $\succ \subseteq A \times A$ is a total order on A ,
- a history h is a sequence of actions, and $valid(h)$ holds if:
 - the propose and breakit actions in the sequence respect \succ ,
 - each propose is followed by a sequence of accept or reject actions respecting \succ until either all agents have accepted the deal or one agent has rejected it,
 - there is no double occurrence of a proposal $propose(a, d)$ of the same deal by any agent $a \in Ag$, and
 - the sequence h ends iff either all agents have accepted the last proposal ($finished(h)$) or the last agent has broken the negotiation ($broken(h)$) instead of making a new proposal.

In theory we can add additional penalties when agents break the negotiation. However, since it is in the interest of all agents to reach an agreement, we do not introduce such sanctions. In this respect norm negotiation differs from negotiation about obligation distribution [4], where it may be the interest of some agents to see to it that no agreement is reached. In such cases, sanctions must be added to the negotiation protocol to motivate the agents to reach an agreement.

Example 1 *Assume three agents and the following history.*

$action_1 : propose(a_1, d_1)$
 $action_2 : accept(a_2, d_1)$
 $action_3 : reject(a_3, d_1)$
 $action_4 : propose(a_2, d_2)$
 $action_5 : accept(a_3, d_2)$
 $action_6 : accept(a_1, d_2)$

We have $valid(h)$, because the order of action respects \succ , and we have $accepted(h)$, because the history ends with acceptance by all agents ($action_5$ and $action_6$) after a proposal ($action_4$).

The open issue of the generic negotiation protocol is the set of deals which can be proposed. They depend on the kind of negotiation. In social goal negotiation the deals represent a social goal, and in norm negotiation the deals contain the obligations of the agents and the associated control system based on sanctions. Here is a history h , where a_1 proposes something which is not accepted, but a_2 thereafter proposes a distribution which is accepted:

5 Social goal negotiation

We characterize the allowed deals during goal negotiation as a set of goals which contains for each agent a goal it desires. Moreover, we add two restrictions. First, we only allow goals the agents have the power to achieve. Moreover, we have to consider the existing normative system, which may already contain norms that look after the goals of the agents. We therefore restrict ourselves to new goals. Additional constraints may be added, for example excluding goals an agent can see to itself. However, since such additional restrictions may be unrealistic in some applications (e.g., one may delegate some tasks to a secretary even when one has the power to see to these tasks oneself), we do not consider such additional constraints.

Definition 9 (Deals in goal negotiation) *In the goal negotiation protocol, a deal $d \in \text{deals}$ is a set of goals satisfying the following restrictions:*

1. $d \in \text{power}(A)$
2. for all $a \in A$ there exists a $g \in d$ such that
 - (a) $g \in \text{goals}(a)$
 - (b) there does not exist a norm n in N such that $g \in \cup_{a \in A} O(n, a)$

The following example illustrates a case in which each agent may desire to be alive. In artificial social systems, it would be based on the utility to be alive.

Example 2 *Let $MAS = \langle A, G, \text{goals}, \text{power}, \geq \rangle$ be a multi-agent system with goals $G = \{\text{not} - \text{killed} - \text{by}_{a,b} \mid a, b \in A\}$, and $\text{not} - \text{killed} - \text{by}_{a,b} \in \text{goals}(a)$ for all $a, b \in A$. Moreover, let $NMAS = \langle MAS, N, O, V \rangle$ with N the empty set. Then G is a social goal (i.e., an element of deals) iff $G \in \text{power}(A)$.*

We could easily further refine our model by defining more abstract goals such as "we have a safe society" and by adding a goal hierarchy reflecting that if some goal is fulfilled, another goal is fulfilled too. For example, if the goal safe-society is fulfilled then the goals $\text{not} - \text{killed} - \text{by}_{a,b}$ are all fulfilled too. However, to keep our model as simple as possible, and to focus on the social delegation cycle, we do not do so in this paper.

We now consider a variant of the running example from [7]. Three agents can work together in various ways. They can make a coalition to each perform a task, or they can distribute five tasks among them and obtain an even more desirable social goal.

Example 3 *Let $MAS = \langle \{a_1, a_2, a_3\}, \{g_1, g_2, g_3, g_4, g_5\}, \text{goals}, \text{power}, \geq \rangle$ be a multiagent system, where:*

power: $\text{power}(a_1) = \{\{g_1\}, \{g_2\}, \{g_3\}\}$, $\text{power}(a_2) = \{\{g_2\}, \{g_3\}, \{g_4\}\}$, $\text{power}(a_3) = \{\{g_3\}, \{g_4\}, \{g_5\}\}$,
if $G_1 \in \text{power}(A)$ and $G_2 \in \text{power}(B)$ then $G_1 \cup G_2 \in \text{power}(A \cup B)$.

Agent a_1 has the power to achieve goal g_1, g_2, g_3 , agent a_2 has the power to achieve goal g_2, g_3, g_4 , and agent a_3 can achieve goal g_3, g_4 , and g_5 . There are no conflicts among goals.

goals: $\text{goals}(a_1) = \{g_4, g_5\}$, $\text{goals}(a_2) = \{g_1, g_5\}$, $\text{goals}(a_3) = \{g_1, g_2\}$.

Each agent desires the tasks it cannot perform itself.

Moreover, let $NMAS = \langle MAS, N, O, V \rangle$ be a normative multiagent system with $N = \{n\}$, $O(n, a_1) = \{g_1\}$. Since there has to be some benefit for agent a_2 and a_3 , the goals g_5 and g_2 have to be part of the social goal. Therefore, social goals (i.e., possible deals) are $\{g_2, g_5\}$ and $\{g_2, g_4, g_5\}$.

Finally, consider the negotiation. Assuming agent a_1 is first in the order \succ , he will propose $\{g_2, g_4, g_5\}$. The other agents may accept this, or reject it and agent a_2 will then propose $\{g_2, g_5\}$. The latter would be accepted by all agents, as they know that according to the protocol no other proposals can be made.

The example illustrates that the negotiation does not determine the outcome, in the sense that there are multiple outcomes possible. Additional constraints may be added to the negotiation strategy to further delimit the set of possible outcomes.

With a large set of agents, the negotiation process becomes very long and cumbersome. Those cases we model using representatives of agent groups negotiating social goals for the group, as we have detailed in [7]. Since this does not further complicate the formal model, and following our minimal extension methodology, we do not consider it in this paper.

6 Social norm negotiation

We formalize the allowed deals during norm negotiation as obligations for each agent to see to some goals, such that all goals of the social goal are included. Again, to determine whether the obligations imply the social goal, we have to take the existing normative system into account. We assume that the normative system only creates obligations that can be fulfilled together with the already existing obligations.

Definition 10 (Fulfillable nmas) *A normative multiagent system $\langle MAS, N, O, V \rangle$ extending a multiagent system $MAS = \langle A, G, goals, power, \geq \rangle$ can be fulfilled if there exists a $G' \in power(A)$ such that all obligations are fulfilled $\cup_{n \in N, a \in A} O(n, a) \subseteq G'$.*

Creating a norm in the normative system entails adding obligations and violations for the norm.

Definition 11 (Add norm) *Let $NMAS$ be a normative multiagent system $\langle MAS, N, O, V \rangle$ extending a multiagent system $MAS = \langle A, G, goals, power, \geq \rangle$. Adding a norm $n \notin N$ with a pair of functions $\langle d_1, d_2 \rangle$ for obligation $d_1 : A \rightarrow 2^G$ and for sanction $d_2 : A \rightarrow 2^G$ leads to the new normative multiagent system $\langle MAS, N \cup \{n\}, O \cup d_1(n), V \cup d_2(n) \rangle$.*

Moreover, if every agent fulfills its obligation, then the social goal is achieved.

Definition 12 (Deals in goal negotiation) *In the norm negotiation protocol, a deal $d \in deals$ for social goal S is a pair of functions $\langle d_1, d_2 \rangle$ for obligation $d_1 : A \rightarrow 2^G$ and for sanction $d_2 : A \rightarrow 2^G$ satisfying the following conditions:*

1. *Adding $\langle d_1, d_2 \rangle$ to $NMAS$ for a fresh variable n (i.e., not occurring in N) leads again to a normative multiagent system $NMAS'$;*
2. *$NMAS'$ achieves the social goal, $\cup_{a \in A} d_1(a) = S$.*
3. *If $NMAS$ is fulfillable, then $NMAS'$ is too.*

The following example models the norm not to kill someone else. Thus, this example of the social delegation cycle is an instance of the Kantian categorical imperative: you should not do to others what you don't want them to do to you. Note that we can also represent $alive_a$ as an abbreviation of the conjunction of *not – killed – by_{a,b}* for all agents b when we extend the language with definitions, but this does not change the principle of the social mechanism.

Example 4 *Assume $\{not - killed - by_{a,b} \mid a \in A\} \subseteq power(b)$ for all agents b . A possible deal for the social goal that there are no murders is that $d_1(b) = \{not - killed - by_{a,b} \mid a \in A\}$ for all agents $b \in A$.*

The second example illustrates the negotiation protocol.

Example 5 *Consider the social goal $\{g_2, g_4, g_5\}$. A possible solution here is that each agent sees to one of the goals. For the social goal $\{g_2, g_5\}$, there will always be one of the agents who does not have to see to any goal.*

Sanctions can be added in the obvious way. In the norm negotiation as defined thus far, the need for sanctions has not been formalized yet. We need to consider the acceptance of norms.

7 Norm acceptance

An agent accepts a norm when it believes that the other agents will fulfill their obligations, and the obligation implies the goals the cycle started with. For the former we use the quasi-stability of the norm (e.g., if the norm is a convention, then we require that the norm is a Nash equilibrium). Each agent b fulfills the norm *given that all other agents fulfill the norm*. Again we have to take the existing normative system into account, so we add the condition that all other norms are fulfilled. In general, it may mean that an agent does something which it does not like to do, but it fears the sanction more than this dislike. We use the trade-off among goals \geq .

Definition 13 (Stability) A choice c of agent $b \in A$ in NMAAS with new norm n is $c \in \text{power}(b)$ such that $\cup_{m \in N \setminus \{n\}} O(m, b) \subseteq d$. The choices of the other agents are $oc = \cup_{a \in A \setminus \{b\}, m \in N} O(n, a) \cup V(n, a)$. The effect of choice c is $c \cup oc \cup V(n, a)$ if $O(n) \subseteq c$, $c \cup oc$ otherwise. NMAAS is stable if $\forall b \in A$, there is a choice c such that $O(n, b) \subseteq c$, and there is no choice $c' \geq (b)c$ with $O(n, b) \not\subseteq c'$.

Finally, we have to test whether the new situation is better than the old one for all agents. Here we assume that the outcome in both the original multiagent system as in the new multiagent system is a Nash equilibrium, and we demand that each Nash outcome in the new system is better than each Nash outcome in the original normative multiagent system. The formalization of these concepts is along the same lines as the definition of acceptance in Definition 13.

Example 6 The norm in the ‘don’t kill me’ example is quasi-stable, since there is no reason for an agent to divert from the norm. Moreover, it is effective since it sees to his goals of the agents. If we assume that agents minimize the set of goals they see to if it does not affect the priority of the agents, or there are some agents who would like to kill other agents, then a sanction has to be added to make sure that no-one is killed. The priority of the goal not to be sanctioned should be higher than the priority of the goal to kill.

The analysis of the second example is analogous.

8 Concluding remarks

In this paper we introduce a norm negotiation model based on power and dependence structures. It is based on a distinction between social goal negotiation and the negotiation of the obligations with their control system. Roughly, the social goals are the benefits of the new norm for the agents, and the obligations are the costs of the new norm for the agents in the sense that agents risk being sanctioned. Moreover, in particular when representatives of the agents negotiate the social goals and norms, the agents still have to accept the negotiated norms. The norm is accepted when the norm is quasi-stable in the sense that agents will act according to the norm, and effective in the sense that fulfilment of the norm leads to achievement of the agents’ desires – i.e., when the benefits outweigh the costs.

Our new model is based on a minimal extension of Tennenholtz’ game theoretic model of the social delegation cycle. We add a negotiation protocol, sanction and control, and besides acceptance also effectiveness. It is minimal in the sense that, compared to our earlier model [7] in normative multiagent systems, we do not represent the rule structure of norms, we do not use decision variables, and so on. Also, as discussed in this paper, we do not add goal hierarchy, definitions, etc. The model therefore focusses on various uses of power: the power as ability to achieve goals and the power in negotiation.

The present paper may be seen as a preliminary study of the expressive power of power and dependence views on multiagent systems. As has been argued by Castelfranchi and Conte for some time, and has been supported by various researchers since then, the power and dependence viewpoint have advantages over classical game theory. However, it remains an open question whether such power structures (or qualitative games in Wooldridge and Dunne’s theory) cannot be mapped on classical games by mapping goals on outcomes, power on strategies, and so on. In future research we intend to study under which conditions or assumptions such mappings can be made.

There are several other issues for further research. First, the motivation of our model is to design social mechanisms. Second, we would like to perform formal analysis, like the complexity results obtained for qualitative games [20] or in game-theoretic artificial social systems. Third, we like to study more general notions of norm creation including permission creation and creation of constitutive norms or counts-as conditionals. Fourth, we are interested in the role of coalition formation, contract negotiation, and obligation distribution in the the new norm negotiation model. Fifth, we would like to extend the model with the distinction between uncontrollable (or external) and controllable (or police) agents as studied by Brafman and Tennenholtz [9]. Finally, we would like to study the revision and evolution of norms.

References

- [1] G. Boella, L. Sauro, and L. van der Torre. Social viewpoints on multiagent systems. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multi Agent Systems (AAMAS’04)*, pages 1358–1359, 2004.

- [2] G. Boella and L. van der Torre. Δ : The social delegation cycle. In *Proceedings of Δ EON'04*, LNAI 3065, pages 29–42. Springer, 2004.
- [3] G. Boella and L. van der Torre. Enforceable social laws. In *Proceedings of The 4th International Joint Conference on Autonomous Agents and Multiagent Systems AAMAS'05*, pages 682–689. ACM, 2005.
- [4] G. Boella and L. van der Torre. Fair distribution of collective obligations. In *Proceedings of ECAI'06*, 2006.
- [5] G. Boella and L. van der Torre. A game theoretic approach to contracts in multiagent systems. *IEEE Transactions on Systems, Man and Cybernetics - Part C*, 36(1):68–79, 2006.
- [6] G. Boella and L. van der Torre. Security policies for sharing knowledge in virtual communities. *IEEE Transactions on Systems, Man and Cybernetics - Part A*, 36(3):439–450, 2006.
- [7] G. Boella and L. van der Torre. Norm negotiation in multiagent systems. *International Journal of cooperative Information Systems*, To appear.
- [8] G. Boella, L. van der Torre, and H. Verhagen. Introduction to normative multiagent systems. *Computational and Mathematical Organizational Theory, Special issue on Normative Multiagent Systems*, 12(2-3):71–79, 2006.
- [9] R.I. Brafman and M. Tennenholtz. On partially controlled multi-agent systems. *J. Artif. Intell. Res. (JAIR)*, 4:477–507, 1996.
- [10] S. Brainov and T. Sandholm. Power, dependence and stability in multi-agent plans. In *Procs. of AAAI99*, pages 11–16, 1999.
- [11] C. Castelfranchi. Modeling social action for AI agents. *Artificial Intelligence*, 103(1-2):157–182, 1998.
- [12] C. Castelfranchi. Micro-macro constitution of power. *ProtoSociology*, 18-19, 2003.
- [13] R. Conte, C. Castelfranchi, and F. Dignum. Autonomous norm-acceptance. In *Intelligent Agents V (ATAL'98)*, LNAI 1555, pages 99–112. Springer, 1999.
- [14] R. Conte, R. Falcone, and G. Sartor. Agents and norms: How to fill the gap? *Artificial Intelligence and Law*, 7(1):1–15, 1999.
- [15] R. Conte and J. Sichman. Multi-agent dependence by dependence graphs. In *Procs of AAMAS02*, pages 483–490, 2002.
- [16] Y. Shoham and M. Tennenholtz. On social laws for artificial agent societies: off-line design. *Artificial Intelligence*, 73 (1-2):231 – 252, 1995.
- [17] Y. Shoham and M. Tennenholtz. On the emergence of social conventions: Modeling, analysis and simulations. *Artificial Intelligence*, 94(1–2):139–166, 1997.
- [18] H.A. Simon. A behavioral model of rational choice. *The Quarterly Journal of Economics*, 69:99–118, 1955.
- [19] M. Tennenholtz. On stable social laws and qualitative equilibria. *Artificial Intelligence*, 102(1):1–20, 1998.
- [20] M. Wooldridge and P.E. Dunne. On the computational complexity of qualitative coalitional games. *Artif. Intell.*, 158(1):27–73, 2004.
- [21] Lopez y Lopez. *Social Power and Norms: Impact on agent behaviour*. PhD thesis, June 2003.

OBJECT-CENTERED INTERACTIVE MULTI-DIMENSIONAL SCALING: ASK THE EXPERT

Joost Broekens Tim Cocx Walter A. Kusters

*Leiden Institute of Advanced Computer Science
Leiden University, The Netherlands*

Email: {broekens, tcocx, kusters}@liacs.nl

Abstract

Multi-dimensional scaling (MDS) is a widely used technique to show, in a low dimensional space, relations between objects—such as humans, documents, soil samples—that are defined by a large set of features. Key benefit is that it enables visual inspection of object relations in an intuitive way. One of the limitations is that different projections exist, leading to different graphical representations and therefore different interpretations of the data. This problem is made more significant in case of noisy data or heuristic approaches to MDS. We propose Object-Centered Interactive Multi-Dimensional Scaling (OCI-MDS), a technique that allows a data expert to try alternative positions for objects by moving them around the space in real time. The expert is helped by several types of visual feedback, such as the proportional error contribution of the expert-controlled object. Here we show that this technique has potential in two different domains, namely the investigation of high-dimensional computer experiment configurations and biomedical data.

1. Introduction

The use of computers has enabled people to create large amounts of data. This is a trend that is not restricted to a specific domain. For example, policy makers write large numbers of reports, individuals publish personal web-pages, scientists do computer simulation experiments, and databases enable structured storage of large amounts of medical data. Extracting potential relations between data objects—i.e., an individual data element such as a document—is a current challenge. In machine learning, many techniques have been developed, such as data clustering, graph-mining and principle component analysis (PCA). In this paper we focus on one such technique, called Multi-dimensional scaling [3, 7].

Multi-dimensional scaling (MDS) is a widely used technique to show, in a low dimensional space, relations between objects—such as human subjects, documents, soil samples—that are defined in a higher dimensional space. If MDS is used to create a 2D visual representation of the high dimensional dataset, a key benefit of this technique is that it enables visual inspection of object relations in an intuitive way. This is important,

especially when the users of the analysis (i.e., those interpreting the final 2D projection) are not machine-learning experts. One of its limitations, however, is that different projections exist, leading to different graphical representations and therefore different interpretations of the data. This problem is especially important in case of noisy data or heuristic approaches to MDS. First, noisy (or unstructured) data introduce variation in the high-dimensional distance between objects, and as such these variations will be reflected in the 2D projection. As this noise does not convey any information regarding the relation between objects, interpretation of the 2D projection is hindered by this noise. The algorithm does not know the underlying relation between objects, and as such cannot correct for it. An expert could. Second, heuristic approaches to MDS, such as the push-pull technique [2, 4] where the projection is constructed through repeated local comparison between pairs of objects, introduce suboptimality in the 2D projection and can converge to local minima. However, heuristic approaches can have important benefits such as reduced computational cost and scalability [8] and are therefore useful for solving MDS problems.

In this paper we propose Object-Centered Interactive Multi-Dimensional Scaling (OCI-MDS), a technique that allows a data expert to propose alternative positions for objects by moving them around the 2D space in real time. The approach is compatible with (and helps) heuristic approaches to MDS. The expert is helped by several types of visual feedback, such as the proportional error contribution of the controlled object. We use the technique in a heuristic MDS approach and show that this technique has potential in two different domains: visualization of high-dimensional computer simulation experiment configurations [1] and raw biomedical data.

As usual, heuristic approaches to MDS rely on several parameters, such as the annealing factor and stopping criterion, and also on random elements (which might cause, e.g., rotated variants). Where possible we have motivated why we use a certain parameter. However, the main aim of this paper is to show the interactive mechanism. We therefore do not motivate all our choices in detail here.

Our interactive approach relates to other approaches, such as those by Stappers et al. [5]. They use interaction to enable exploration of data. Objects can be selected by the user, after which the algorithm clusters the newly selected object. Subsequently, a next object can be added (or removed). This approach *is* Object-Centered and allows expert-controlled visualization of object-relations, but different in the sense that objects, once positioned, are not interactively movable to (a) generate alternative hypotheses about object relations, or (b) help the MDS mechanism. Further they focus on small amounts of objects (about 10). Other approaches include non Object-Centered ones, such as those that enable experts to direct computational resources at specific parts of the space in order to reduce computational resources needed for data projection [8], and those that enable experts to interactively change algorithm parameters (like noise) and to stop the algorithm [6].

In Section 2 we describe some of the problems our approach addresses. In Section 3 we introduce Object-Centered Interactive MDS. Section 4 presents experimental results contains all figures. Finally, we present our conclusion and some directions for future work in Section 5.

2. Expert Interaction Helps Heuristic Projection

A key motivation to use MDS for visualization of high dimensional data is its ability to give overview over a complete dataset. This is important in the exploratory phase of data analysis. For example, in the criminal investigation area, visualization of datasets supports police officials in their process of generating hypotheses about the relation between different criminal records [2]. In the computer simulation domain, such as simulation of adaptive behavior [1], scientists often repeat experiments with slightly different settings. It is important to avoid making errors in the configuration of the experiments and it is important to have a clear overview of the variations introduced in the parameters. Visualization of the relation between configurations of these experiments (not just the results) can therefore provide insight into both the completeness of a set of experiments as well as potential configuration mistakes. In the domain of biomedical data analysis, clustering, dimension reduction and visualization are used to, for example, find in a set of patients different variations of one disease, or find the most important factors underlying a certain disease.

In all three domains, MDS can be used to cluster data by projecting the high dimensional data onto a 2D space (note that data is not really clustered, as explicit groups are not made). Visualization of that space enables domain experts to get an intuitive idea of the relation between the objects in high-dimensional space. Typically, a 2D projection is constructed such that the least-square error is minimized (see Section 3). However, an error of 0 is usually not possible, and, if the projection technique is heuristic, minimal error cannot be guaranteed. Another typical problem in heuristic approaches—that use incremental error minimization by inducing small changes to object locations in 2D—is that two objects that should be close to each other can be separated by a large cluster, because the large cluster pushes both objects away from each other (see Figure 2, Section 4). Standard incremental techniques cannot solve this problem. Thus, even though the solution is near optimal, large local errors can exist.

However, domain experts can detect such local errors by looking at the object and comparing it with its neighbors. So, from an optimality point of view the ability to move objects and clusters of objects is a useful addition to heuristic approaches to MDS. For data interpretation it is also a useful addition, as interactive real-time movement of objects enables experts to test hypotheses of relations between objects directly in the clustering result. This means that, e.g., police officials are able to test if two criminal records are related just by moving the objects close to each other and observing, e.g., the clustering result. Another advantage is the possibility to add new objects at user specified locations, and observe the trajectory of these objects in the projection as well as the influence of these objects on the projected location of other objects.

To summarize, object-based interaction with MDS is useful, provided that users get feedback information so that they can (1) select objects to move, and (2) evaluate the result of the move.

3. Object-Centered Interactive MDS

We propose Object-Centered Interactive MDS (OCI-MDS). This allows experts to interactively manipulate the projection result produced by a heuristic MDS algorithm. We present our algorithm and the kind of user-feedback the system gives. In the next section we show that it is a useful technique in two domains: computer simulation experiments and biomedical data analysis.

Analogous to standard MDS, four steps are needed to project m -dimensional data onto a low (2D) dimensional space. The first two are preparatory and the second two are iterative until some stop-criterion (usually reaching a minimal error, or stalled improvement for some fixed number of iterations).

First, a distance matrix is constructed that captures the distances between n individual objects in m -dimensional space, where m typically is the number of features used to describe an object. If objects do not have the same number of features (i.e., objects in one set have different dimensionality) then the distance matrix must be able to cope with this. We assume we have such an $n \times n$ distance matrix D .

Second, objects are randomly placed in a low dimensional space, in our case a 2D space. A vector O of size n represents the coordinates of all n objects.

Third, the first iterative step selects (e.g., randomly) an object i and adds random noise r to the coordinates of that object. Noise is added as follows:

$$\begin{aligned} O_i[x] &\leftarrow O_i[x] + \text{rnd}() \times \text{step} \\ O_i[y] &\leftarrow O_i[y] + \text{rnd}() \times \text{step} \end{aligned} \quad (1)$$

where $O_i[x]$ and $O_i[y]$ are the coordinates of an individual object i , and $\text{rnd}()$ a function giving a random number in $[-0.5, 0.5]$. The variable step is a noise size factor that is local-error, total 2D space span, and annealing dependent:

$$\text{step} = \alpha \times d \frac{n \times e_i}{e} \quad (2)$$

where, d is the largest distance between objects in O , and thus equivalent to $\max(D^L)$ (see below), n is the number of objects, e_i the local error associated with object i , e the global error (see below), and α an exponentially decreasing annealing factor. The local error e_i is defined by:

$$e_i = \sum_{j=1}^n (D_{ij} - D_{ij}^L)^2 \quad (3)$$

where D^L is the distance matrix of objects in 2D space. The motivation for our step factor is as follows. It expresses a normalized maximum step that depends on the error contribution of an object to the global error *and* the size of the space covered by all objects. This has the following two benefits. First, an object with a high local error is moved through the 2D space with high speed, in order to find a better location for it. This increases the probability that wrongly placed objects eventually find a suitable location with small local error. Second, if the 2D space that is occupied by the objects is large, the objects will also move quicker. This ensures that the algorithm is not dependent on the absolute distance between objects. Further, we decrease annealing

factor α exponentially whenever for all objects i there has been no decrease in e . So, if the algorithm approaches a minimum, smaller steps can be used to better approach that minimum.

Fourth, update the distance matrix D^L (note that we use Euclidean distances for D^L). Then evaluate the least-square error (LSE) between D and D^L :

$$e = \sum_{i=1}^n \sum_{j=i}^n (D_{ij} - D_{ij}^L)^2 \quad (4)$$

If the local noise added to object i decreases global error e , keep the new coordinates; if not, discard the change. Repeat step three and four until e is smaller than a threshold t , or until e has not decreased for a fixed number of steps s . If this criterion is met, the process is paused until the user interactively changes positions of objects.

Objects are drawn in a two dimensional plane (e.g., Figure 1, Section 4). The user is able to, at any time, grab objects and place them at alternative positions. This enables the user to (1) help the heuristic MDS, and (2) experiment with potential clusters. The user is given two types of direct feedback. First, when objects are released, the projection mechanism restarts iteration of step three and four, so the user can directly observe the effect of the moved objects on the total distribution of objects. Second, objects are drawn in a color that represents the local error contribution of the object. This is non-trivial, as color changes need to be reactive enough to reflect small changes in local error but also reflect global changes in global error e . We used the following formula:

$$color_i = \log\left(1 + \frac{n \times e_i}{\log(1+n) \times e_{\min}}\right) \quad (5)$$

where n is the number of objects, e_i the local error of object i and $e_{\min} = \min(e_s)$ for $s=0,1,2,\dots$ where e_s is the global error at iteration s . The variable $color_i$ can be used to define, for example, drawing intensity or color of an object i . This color scheme was used to experiment with interactive visualization of simulation experiment configurations (Fig. 1), as well as interactive visualization of biomedical data (Fig. 3).

We also experimented with a different coloring scheme where objects are colored using a per object measure that is relevant to the dataset, not the algorithm. This scheme was used to experiment with biomedical data visualization (Figure 2). The data consisted of about 400 objects with 10 features. The objects are patients. Every feature represents the severity of a disease in a different part of the body. The color represents the total severity calculated by averaging over the different features. This average is meaningful but somewhat arbitrary, as it is not clear that the average is actually a good representation of the global severity of the disease. However, as our focus is the visualization technique, not the dataset, we do not consider this to be a problem at this moment. For the same reason we do not specify the datasets in detail in this paper.

4. Experimental results

We have developed a Java application that allows us to test our approach. First we present results that investigated its use in visualizing simulation experiment configurations. The dataset consisted of 44 experimental configurations, all of which are

used in research into the influence of emotion on learning using reinforcement learning [1]. The features of the objects consisted of 40 different learning parameter settings such as learning rate, exploration-exploitation settings, etc. This can be considered structured data. We have a vector representation of these configuration documents, and defined a distance measure based on the parameter type (boolean, string, double) and the range of values found for one parameter. We do not detail the distance measure here. Based on the measure we constructed the distance matrix D .

Figure 1a shows an initial 2D projection of a set of experiment configurations. The visualization clearly shows that there are 4 experiments that are special (bottom), and several groups of other experiments. The objects at the bottom are control experiments, and are indeed the control experiments with which the others are compared. The control experiment at the right is farther away from the other three (and farther away from all other experiments). Did our algorithm correctly place it here? The user can grab (Figure 1b) the object, and while moving it, the local errors start to increase (objects color red). Apparently, the object should not be moved in that direction. After letting the object go, the algorithm projects the object back to a similar (but not necessarily equal) position. The object indeed belongs there. Other object clusters show a nice regular pattern, as a result of the distance function. The four top clusters (Figure 1c) all belong to one typical parameter value, while the middle four all belong to a different value on that same parameter. The clusters themselves are organized and correspond well to the actual configurations. This enabled us to verify that no configuration errors had been made in the course of these experiments.

Second, we experimented with the biomedical data mentioned earlier. The projection resulted in a clustering that showed a trend from high severity to low severity, even though global severity is not a feature in the data (Figure 2a). Although the projection clearly does not give as much insight into the data as the projection of the structured experiment data shown before, several clusters appear to exist. For example, cluster 1 represents a coupling of two severely affected body parts. Cluster 2 represent a coupling of two other severely affected body parts where the two parts of cluster 1 are not affected. This might indicate correlation between certain body parts and dissociation between others. Although the heuristic technique extracts some useful information from unstructured raw biomedical data, as it stands, the technique is clearly not usable for quantitative conclusions about the dataset, but only for explorative data analysis. However, the main topic of this paper is dataset exploration and interactivity. Interaction enabled us to relocate two items that appeared to be badly clustered due to the separation problem mentioned earlier, i.e., a cluster divides otherwise closely related objects (Figure 2b and c). After grabbing and relocating the two severe disease cases to an arbitrary position on the other side of the cluster consisting of non-severe disease objects, they were relocated by the algorithm at positions that better matched their real-world relation, as could be observed from a comparison with the objects near to that new location. Finally, Figure 3 shows the variation in the distribution of local errors. Figure 3a also shows one object ('x') with high local error positioned in between objects with small local errors. When grabbing and repositioning the object at a location in which it appears to have smaller local error, we were able to relocate it at a better place. Although the exact meaning of the projection is at this point unclear (and strongly dependent on the distance measure we used), our experiment shows that Object-Centered interactivity is a useful method to explore object relations.

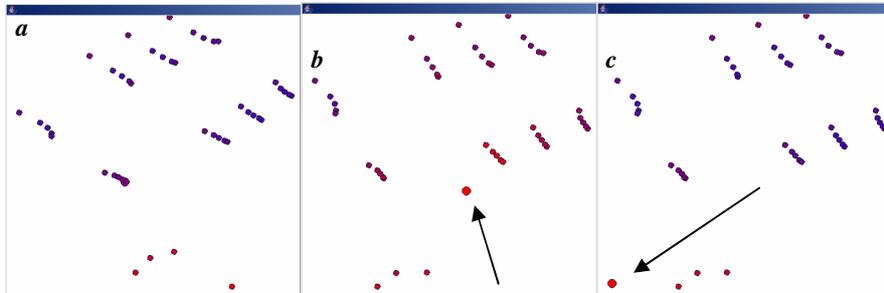


Figure 1. Manipulating experiment configuration clusters (local error color).

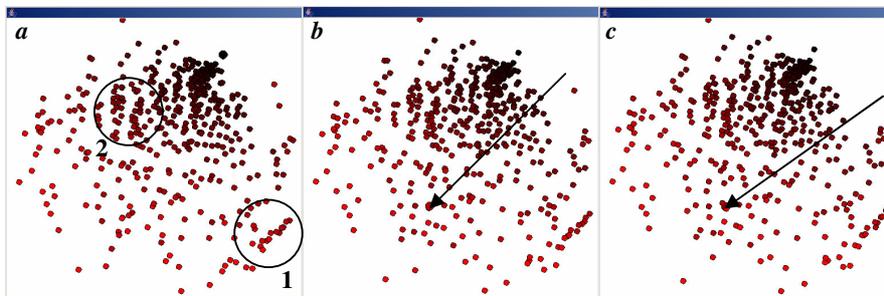


Figure 2. Manipulating biomedical data (severity color).

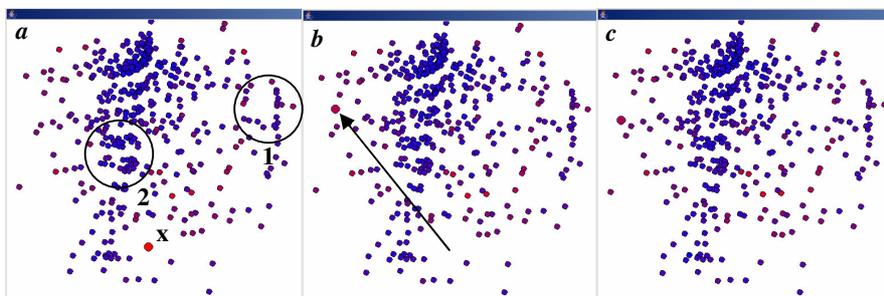


Figure 3. Manipulating biomedical data (local error color).

5. Conclusion and Future Work

Our experiments show that Object-Centered Interactive MDS has potential. It can be used for direct manipulation of clustering results based on a heuristic MDS approximation. It can help in verifying the MDS result, and help to generate hypotheses about alternative object relations, that were not found, for example, because the MDS converged to a local optimum. However, currently its usefulness is somewhat limited on highly unstructured data.

Future work includes testing the mechanisms on criminal record datasets [2], adding multiple-object drag-and-drop, extensive user testing, and scaling mechanisms like those introduced by Williams and Munzner [8]. Also, relating the re-placement of single items with a high local error (a high contribution to the global error) to the change in the global error is important for the analysis of the proposed approach. Changes in the global error can be represented by a Boolean figure (higher or lower) or be represented by a (color-)scale during the process of human intervention. Future research can strive to find a relation between the decrease in local errors and the error made in the total image. If such a positive relation exists, automating the process of relocating those items with the highest local error can be an asset worth pursuing. This global optimization can be of interest in areas where the correctness of the image as a whole is more important than the relation between a small subset of individual items, like, for example, a clustering on customer behavior.

References

- [1] J. Broekens & F. J. Verbeek. Simulation, Emotion and Information Processing: Computational Investigations of the Regulative Role of Pleasure in Adaptive Behavior. In: *Proc. of the Workshop on Modeling Natural Action Selection*, pages 166-173, 2005.
- [2] T. K. Cocx & W. A. Kusters. A Distance Measure for Determining Similarity between Criminal Investigations. In: *Proc. Industrial Conference on Data Mining (ICDM), LNAI 4065*, pages 511-525, Springer, 2006.
- [3] M. L. Davison. *Multidimensional Scaling*. John Wiley and Sons, New York, 1983.
- [4] W.A. Kusters & M.C. van Wezel. Competitive Neural Networks for Customer Choice Models. In: *E-Commerce and Intelligent Methods, Studies in Fuzziness and Soft Computing*, pages 41-60. Physica-Verlag, Springer, 2002.
- [5] P. J. Stappers, G. Pasman & P. J. F. Groenen. Exploring Databases for Taste or Inspiration with Interactive Multi-Dimensional Scaling. In *Proc. of IEA2000*, pages 3-575-3-578, 2000.
- [6] E. Tejada & R. Minghim. Improved Visual Clustering of Large Multi-Dimensional Data Sets. In: *Proceedings of the Ninth International Conference on Information Visualisation (IV'05)*, pages 818-825, 2005.
- [7] W. S. Torgerson. Multidimensional scaling: I. Theory and Method, *Psychometrika* 17(4), pages 401-419, 1952.
- [8] M. Williams & T. Munzner. Steerable, Progressive Multidimensional Scaling. In: *Proc. of the IEEE Symposium on Information Visualization (INFOVIS'04)*, pages 57-64, IEEE, 2004.

Coordinating Autonomous Planning Agents with Temporal Constraints

Pieter Buzing^{a*} J. Renze Steenhuisen^a Cees Witteveen^a

^a *Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology, P.O. Box 5031, 2600 GA Delft, The Netherlands,
email: {J.R.Steenhuisen, C.Witteveen, P.C.Buzing}@tudelft.nl*

Abstract

We address the problem of coordinating autonomous agents that have to come up with a joint plan and schedule for a set of tasks. These tasks are interdependent and temporally constrained and are distributed over the agents. We start with investigating the scheduling part of the problem: How to achieve, given the constraints, a joint solution to the distributed scheduling problem where each of the agents is allowed to determine its own schedule for the set of tasks assigned to it. Here, the problem is to ensure that whatever individual schedule each agent comes up with, these schedules always can be combined into a global schedule satisfying all constraints. Next, we investigate a more complex problem where, before scheduling, the agents are allowed to make their own plan to achieve the tasks assigned to them. That is, they are allowed to choose autonomously some additional precedence constraints to their assigned tasks. We have to provide the necessary coordination mechanisms to ensure that no conflicts will arise between the resulting plans. The solution we provide is to reduce the temporal constraints of the individual tasks to a set of precedence constraints between tasks and virtual tasks and we solve the resulting coordination problem for tasks with precedence constraints. The solution thus obtained enables us to solve the plan coordination problem with temporal constraints.

1 Introduction

Typically, multi-agent planning and scheduling problems require a set of autonomous planning agents to come up with a joint plan and schedule for achieving a set of tasks. Usually, none of the participating agents is capable of solving all tasks by itself. Therefore, each agent is assigned a partition of the tasks to carry out. To complete its part of the job, each agent has to come up with a plan and schedule to carry out these tasks. Moreover, it is required that these individual plans and schedules are *compatible* in the sense that together they constitute a feasible plan/schedule for the total set of tasks. In particular, if we assume that the agents involved are self-interested and require to determine their plans and schedules autonomously, we have to ensure that *coordination mechanisms* are provided to ensure the compatibility of plans and schedules.

With respect to task coordination, it is common to classify sets of tasks with respect to the level of coordination they require [5]. Sometimes, the set of tasks to perform is decomposable into a set of independent tasks that can be assigned to agents. Each of the agents is able to construct an independent plan for its subset of tasks and the coordination problem is trivially solved. Typical tasks that can be solved in this way are reconnaissance tasks and simple pick-up and delivery tasks. Note that here coordination, if necessary at all, is only needed in solving task allocation problems (who will do which tasks). Such tasks can be solved by *loosely-coordinated* teams of agents.

Other tasks, however, not only require coordination during task allocation, but also during *planning*. For example, if there are constraints between tasks assigned to different agents, those agents have to coordinate during their planning in order to construct a plan that satisfies these

*This research is supported by the Technology Foundation STW, applied science division of NWO and the technology programme of the Ministry of Economic Affairs.

constraints. In order to solve such tasks we need *moderately-coordinated* agents. Typical problems in this category are monitoring tasks and multi-modal transportation tasks.

Finally, there is a set of tasks that require coordination also in the *execution phase* of the plan. Such tasks are said to be tightly coupled and require a *tightly-coordinated* team of agents (e.g., platooning vehicles or moving in formation). Usually, such tightly-coupled sets of tasks contain *temporal constraints* and *simultaneity constraints*.

In previous work on coordination [1], we have concentrated on solving multi-agent planning problems where

- agents are self-interested and non-cooperative,
- while the tasks to solve require moderately-coordinated agents, and
- no temporal constraints were assigned to the tasks.

Basically, our approach to solve such problems comes down to enabling loosely-coordinated agents to solve tasks that require moderately-coordinated agents by providing adequate *coordination mechanisms* that respect the autonomy of the participating agents. For example, we showed how to solve multi-modal transportation systems that require a joint plan for a set of interrelated tasks by providing the participating agents (companies) with a coordination mechanism that enables them to solve their part of the planning problem independently from the others. The coordination mechanism ensures that every plan (solution) an individual agent is providing can seamlessly be integrated in the joint plan to be constructed. This research approach can be summarised by the equation

$$\textit{moderately-coupled agents} = \textit{loosely-coupled agents} + \textit{coordination mechanisms}.$$

Our future research will aim at solving multi-agent planning problems that require *temporal (semi-) tight coordination* between agents to solve them. Therefore, we want to extend our current work on coordination to interdependent tasks with *temporal constraints*. The introduction of time constraints forces us to take into account coordination of scheduling besides plan coordination, and enables us to prepare for the next step in coordination research (i.e., the development of coordination mechanisms for tightly-coupled tasks).

First of all, we will concentrate on what we consider a basic coordination problem raised by the addition of temporal constraints on tasks: How to ensure that a distributed scheduling process executed by a set of autonomous agents will result in a joint schedule that satisfies both the *given* precedence constraints and the temporal constraints. The basic idea here is to remove all dependency constraints between tasks belonging to different agents by translating them into suitable disjoint temporal constraints. As a result, we are able to decompose the total problem into independent scheduling problems, one problem for each agent. The agents can determine their own schedule times for the tasks assigned to them individually and are guaranteed that every such schedule can be seamlessly integrated into an overall schedule.

Next, we show how to integrate both plan coordination and scheduling coordination into one framework by solving the problem how a joint schedule can be found for temporally constrained tasks, even if the agents are able to plan their part of the tasks independently from the others. A nice feature of this integrated approach is that it enables us to reduce coordination problems with temporal constraints to coordination problems *without* temporal constraints, thereby enabling us to reuse previously-developed methods to solve plan coordination methods.

2 Coordination for Distributed Scheduling

We start with investigating a simple distributed scheduling problem where a number of interdependent tasks with temporal constraints has been assigned to a number of agents. The agents want to decide autonomously when to schedule the tasks assigned to them. Each such a schedule, of course, has to satisfy the local dependency constraints and temporal constraints. The problem is how to ensure that whatever local schedule is chosen, there will be no conflicts between them.

We formalise such a distributed scheduling problem as follows. An instance of the problem is a tuple (T, \prec, I, A, g) consisting of

- a set of tasks $T = \{t_1, t_2, \dots, t_n\}$ to be executed,
- a partially-ordered precedence relation \prec defined on T , where $t \prec t'$ indicates that task t has to start before t' can start,
- for each task $t \in T$ a temporal interval (time window) $I(t) = [lb(t), ub(t)]$, with $lb(t) \leq ub(t)$, constraining the starting time of t to a time point between $lb(t) \in \mathbb{Z}$ and $ub(t) \in \mathbb{Z}$,
- a set of agents $A = \{A_1, \dots, A_m\}$, and
- a function $g : T \rightarrow A$ assigning each task $t \in T$ to an agent $A_i \in A$. The set $g^{-1}(A_i)$ of tasks allocated to agent A_i is denoted by T_i .

We will use the following notations to denote special time windows and relations between such time windows:

1. $[]$ denotes the empty time window,
2. $I(t) < I(t')$ holds if $ub(t) < lb(t')$,
3. $I(t) \cap I(t') = [max\{lb(t), lb(t')\}, min\{ub(t), ub(t')\}]$, and
4. $c \in I(t)$ if $lb(t) \leq c \leq ub(t)$.

Furthermore, we will call a precedence constraint $t \prec t'$ an *intra-agent* constraint if and only if $t, t' \in T_i$ for some i . Such a precedence constraint is called an *inter-agent* constraint if and only if $t \in T_i$ and $t' \in T_j$ for $i \neq j$.

Each of the agents A_i independently chooses a schedule $s_i : T_i \rightarrow \mathbb{Z}$ for its set of tasks T_i , that is each agent A_i determines a starting time $s_i(t)$ for all the tasks $t \in T_i$ such that the following requirements are satisfied.

1. For every $t \in T_i$, $s_i(t) \in I(t)$, that is the starting time should be allowed by the time interval of the task.
2. For every $t, t' \in T_i$, if $t \prec t'$ then $s_i(t) < s_i(t')$, that is starting times should respect local precedence constraints.
3. For every $t \in T_i$ and $t' \in T_j$ with $i \neq j$, if $t \prec t'$ then $s_i(t) < s_j(t')$, that is starting times determined by different agents should respect inter-agent precedence constraints.

The reader should be aware that the requirements above for feasible schedules together with the precedence constraints on the tasks induce some additional constraints on the time windows of the tasks related to each other via a precedence constraint.

The following rule has to be applied here:

If $t \prec t'$ then $lb(t') := \max(lb(t'), lb(t) + \epsilon)$ and $ub(t) := \min(ub(t), ub(t') - \epsilon)$, where ϵ is the minimal separation time between tasks.

Without loss of generality, we assume $\epsilon = 1$.

To give a simple example, suppose task t has to be scheduled in the interval $I(t) = [10, 70]$ and task t' has to start in the interval $I(t') = [5, 50]$ and we have the precedence constraint $t \prec t'$. This immediately implies that t' cannot start earlier than $\max(5, 10 + 1) = 11$ and t cannot start after $\min(70, 50 - 1) = 49$.

By applying this rule to all pairs of tasks we can obtain a *normalised* instance. In the sequel, we assume, without loss of generality, that the given instances are normalised and have at least one solution (i.e., for all t we have $ub(t) \geq lb(t)$ and for all t, t' , $t \prec t'$ implies that $lb(t) + 1 \leq lb(t')$ and $ub(t) + 1 \leq ub(t')$).¹

Obviously, because of requirement 3, this distributed scheduling problem requires coordination between the agents. Assuming that all the agents A_i are self-interested and desire autonomy in determining their scheduling function s_i , we could try to solve this scheduling problem by establishing a coordination mechanism that guarantees the existence of a *global* solution satisfying requirements 1-3, whenever each agent has established a *local* solution satisfying requirements 1 and 2.

¹Note that here, we only consider absolute time windows for tasks. The reader might note that a simplified version of STPs [3] can be used to represent the scheduling problem. More information about temporal networks and (directional) path consistency in general can be found in [2].

One such coordination mechanism can be easily found. The basic idea is to get rid of all inter-agent precedence constraints $t \prec t'$ where $t \in T_i$ and $t' \in T_j$ belong to the tasks of different agents, by separating the corresponding time windows: Let $I(t) = [lb(t), ub(t)]$ and $I(t') = [lb(t'), ub(t')]$. We have to find two time points δ_1 and δ_2 such that $lb(t) \leq \delta_1 \leq ub(t)$, $lb(t') \leq \delta_2 \leq ub(t')$ and $\delta_1 < \delta_2$. Then the intervals are changed to $I(t) = [lb(t), \delta_1]$ and $I(t') = [\delta_2, ub(t')]$, thereby removing the possible overlap. This interval separation encodes requirement 3 indirectly by enforcing a smaller value for $s_i(t)$ than for $s_j(t')$ for all locally feasible schedules $s_i()$ and $s_j()$.

Remark In general, the problem to remove all dependencies between coupled systems with temporal constraints is called the *Temporal Decoupling Problem* and is extensively discussed in Luke Hunsberger’s PhD thesis [4]. ■

The following algorithm determines a set of separated intervals for the tasks connected via inter-agent precedence constraints such that coordination of local schedules is ensured. The algorithm starts with the minimal tasks t in the precedence order that have a successor, separates their time window from the time windows of the tasks dependent on t and then moves upwards in the partial order.

Algorithm 1 Algorithm to separate overlapping intervals.

Require: a normalised instance (T, \prec, I, A, g) of a distributed scheduling problem;

Ensure: a set of disjoint intervals for tasks $t \in T$ connected via precedence relations;

- 1: remove all local (i.e., intra-agent) precedence constraints from \prec ;
 - 2: **while** there exist tasks $t, t' \in T$ such that $t \prec t'$ and $I(t) \cap I(t') \neq []$ **do**
 - 3: find a task $t \in \min_{\prec}\{T\}$ such that there exists a task t' with $t \prec t'$;
 - 4: let $I(t) = [l_1, u_1]$;
 - 5: let $u'_1 := \min\{lb(t') \mid t \prec t'\} - 1$;
 - 6: $I(t) := [l_1, u'_1]$;
 - 7: $T := T - \{t\}$;
 - 8: **end while**
-

Example There are two agents A_1 and A_2 , each having two tasks to perform. There are two inter-agent precedence constraints between the tasks and each task has a time window for starting times (see Figure 1a). If each of the agents is allowed to determine a feasible schedule for the tasks it has to achieve, we might end up with an infeasible total schedule. For example, the schedules $s_1(t_1) = 43, s_1(t_3) = 40, s_2(t_2) = 43, s_2(t_4) = 40$ are locally feasible but do not satisfy the precedence constraints between the agents. In Figure 1b, Algorithm 1 has been applied to separate the time windows. It is not difficult to see that now every locally feasible schedule also constitutes a globally feasible schedule. ■

It is easy to see that this algorithm terminates, and that all the temporal intervals are disjoint after termination. Moreover, if the original instance has a solution, the transformed instance will have a solution, too. Note that this algorithm is a centralised one. However, it can easily be transformed into a distributed algorithm by using a blackboard for maintaining the inter-agent dependency constraints.

Remark Besides the algorithm given above, other heuristics might be used to separate time windows as well. Hunsberger [4] provides a number of heuristics to find good values for δ_1 and δ_2 . Note that the order in which the inter-agent constraints are tackled will influence the result. More sophisticated algorithms will most likely be based on domain specific objectives, like “prefer earliest possible starting times”, “maximise flexibility for the least flexible task”, or “the number of adjusted time intervals of each agent should be equal”. Here are some other ideas for further improvements:

1. Reducing a time window can have cascading effects. This means that decoupling an inter-agent relation can overconstrain other constraints. To avoid this we can apply a minimal change strategy: Take an inter-agent constraint $t \prec t'$ and adjust the lower bound of t' or the upper bound of t minimally (i.e., with a value of 1). We propagate these effects over the entire task network. This procedure is repeated until all inter-agent constraints are dominated.

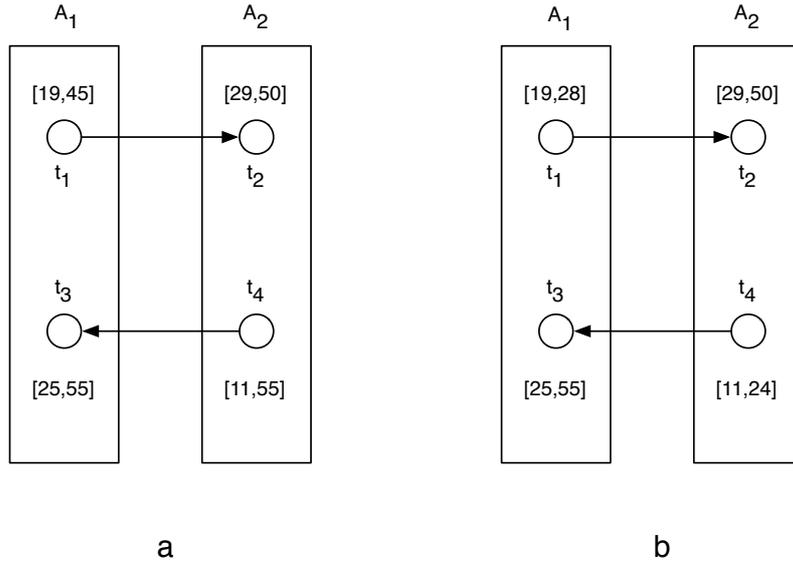


Figure 1: Separating time windows to coordinate distributed scheduling.

- Assuming that the problem instance is normalised, for each of the inter-agent constraints we determine the overlap: $\text{overlap}(t, t') = ub(t) - lb(t')$. This overlap is a measure for the amount of temporal space the agents together have to give in. If we start the separation process with the smallest overlapping inter-agent constraint and then propagate the effects, we only remove a single constraint, enforce minimal restriction of the time intervals, and possibly make other inter-agent constraints superfluous. ■

It is now easy to see that whenever there exists a global scheduling $s : T \rightarrow \mathbb{Z}$ satisfying all the constraints, there also exists a coordination mechanism guaranteeing that a set of local solutions s_i can be found that constitutes a distributed solution to the coordinated scheduling problem. The following result is an easy consequence of the preceding discussion.

Proposition 1 *Every instance (T, \prec, I, A, g) of a multi-agent scheduling coordination problem that is normalised and, for every t, t' such that $t \prec t'$ and $g(t) \neq g(t')$, satisfies $I(t) \cap I(t') = []$ has the property that every set of local schedules $\{s_i\}_{A_i \in A}$ satisfying requirement 1 and 2 also satisfies requirement 3.*

3 Coordinating Planning and Scheduling

The basic coordinated scheduling approach does not recognise the possibility that some agents would like to impose their own ordering on starting the tasks they have been assigned to, *before* they wish to coordinate their scheduling activities. For example, suppose an agent A_i wants to execute its tasks in such a way that it can optimally use its resources. Now, A_i acknowledges that in its subset of tasks in principle there is no objection to start t and t' concurrently (since no constraint forces it to do otherwise), but due to resource constraints known to agent A_i , the agent decides to start t before t' . So it wishes to impose an additional precedence $t \prec t'$ on its own set of tasks T_i . The problem is that if we allow agents to make such decisions autonomously, there is a possibility that they never will be able to find a coordinated schedule for the total set of tasks. To see this, consider the following example.

Example There are two agents A_1 and A_2 , each having to complete two temporally constrained tasks (see Figure 2a). Now suppose that each of the agents decides autonomously to add additional precedence constraints (indicated by dashed arrows in Figure 2b) and each agent normalises its set of temporal constraints correspondingly. Since the precedence constraints are cyclic, it is not difficult to see that there cannot exist a feasible joint schedule for this set of tasks. ■

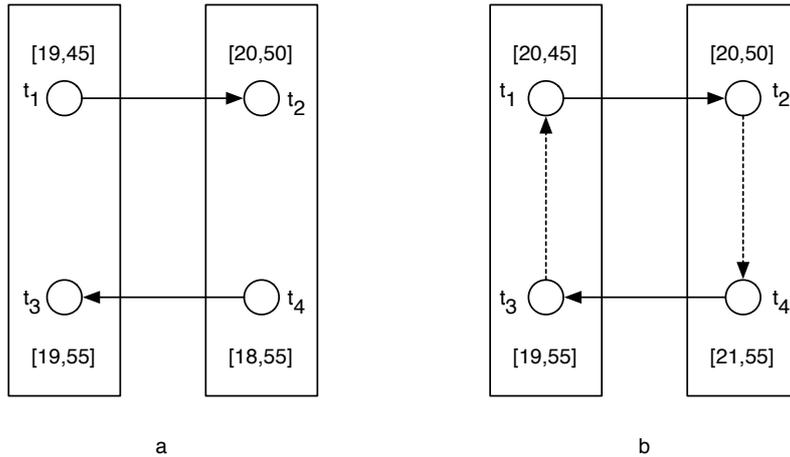


Figure 2: Autonomous planning of agents might cause failing of coordinated scheduling.

In general, we can state the autonomous planning+scheduling problem as follows: Given a coordination problem (T, \prec, I, A, g) with temporal intervals, how to ensure that a feasible joint schedule can be found for all the tasks in T if every agent A_i is free to compose its own ordering \prec_i on the tasks T_i assigned to him, while respecting the precedence relation \prec and the given temporal intervals?

This problem closely resembles the plan coordination problem we have investigated in [1, 6]. The only difference between the latter problem and our current problem is the incorporation of temporal constraints for tasks. We will first give a brief overview of this plan coordination problem and then we show how our current problem including temporal constraints can be solved by reducing it to the plan coordination problem.

The distributed plan coordination problem can be stated as follows: We have a set of tasks T together with a precedence relation \prec . There is also a set of agents $A = \{A_1, A_2, \dots, A_m\}$, each agent A_i having to achieve a partition T_i . We assume that each agent is an autonomous planner and is able to find a suitable plan (T_i, \prec_i) where each \prec_i is an ordering respecting the given precedence relation \prec . We say that there exists a feasible joint plan of the agents if the structure $(T, (\prec \cup \prec_1 \cup \dots \cup \prec_m)^+)$ is partially ordered (i.e., if the individual orderings \prec_i together with \prec do not induce a cycle). Here, if ρ is any relation, ρ^+ denotes its transitive closure.

In [1], we have investigated this problem and we have shown that it can be solved by adding a minimal set of precedence constraints, called a *coordination set*, $\Delta = \bigcup_{i=1}^m \Delta_i$ to the precedence relation \prec in such a way that it ensures that every locally determined partial order \prec_i respecting $\prec \cup \Delta_i$ can be used to constitute a global partial order $(\prec \cup \Delta \cup \prec_1 \cup \dots \cup \prec_m)^+$ that describes a joint plan to execute the tasks in T .

Example As an example, consider Figure 2 and neglect the temporal intervals for the tasks. If agent A_1 and A_2 are free to plan autonomously, they might come up with a jointly infeasible plan. This happens if agent A_1 adds $t_3 \prec_1 t_1$ and agent A_2 adds $t_2 \prec_2 t_4$, thereby creating the cycle $t_1 - t_2 - t_4 - t_3 - t_1$ (see Figure 2b). A solution is to add for example a set of constraints $\Delta = \{t_1 \prec t_3\}$. This constraint prevents inter-agent cycles to occur. ■

The idea is to solve our planning+scheduling problem by first reducing an instance with temporal constraints to a plan coordination instance without temporal constraints. Thereafter, we solve the plan coordination problem by identifying a set of suitable constraints and add these constraints to the given instance. Finally, the agents are allowed to plan autonomously and after planning we solve the resulting scheduling problem by our distributed scheduling algorithm. Summarising, the procedure is as follows:

- Reduce the coordination problem with temporal constraints to a coordination problem without temporal constraints.

- Determine a solution of the plan coordination problem, that is find a coordination set Δ .
- Let the agents plan autonomously, that is determine their orderings \prec_i respecting $\prec \cup \Delta$.
- Solve the resulting scheduling problem $(T, \prec_1 \cup \prec_2 \cup \dots \cup \prec_m, I, A, g)$ with temporal intervals by applying the basic coordination algorithm.

Here, it suffices to fully describe step 1: The reduction of the coordination problem with temporal constraints into a coordination problem without temporal constraints. The idea is the following: Given an instance (T, \prec, I, A, g) , we first collect all the begin and end points of the intervals $I(t)$. More precisely, for each task t , the time interval $I(t) = [lb(t), ub(t)]$ is coded into two values s and s' , where $s = lb(t) - 1$ and $s' = ub(t) + 1$. We collect the total set of all these values in the set $S = \{s_1, s_2, \dots, s_p\}$ and, without loss of generality, we may assume that $s_1 < s_2 < \dots < s_p$. Now, we construct the following coordination instance (T', \prec', A', g') , where

1. the set of tasks T' is extended with the set S : $T' = T \cup S$.
2. the precedence relation \prec' equals \prec extended with the total ordering imposed on S . Moreover, for each task t , associated with time points s and s' , two additional precedence constraints $s \prec' t$ and $t \prec' s'$ are constructed. The result then is $\prec' = \prec \cup \{(s_i, s_j) \mid 1 \leq i < j \leq p\} \cup \{(s, t), (t, s') \mid t \in T, s = lb(t) - 1, s' = ub(t) + 1\}$.
3. we add a special agent A_S taking care of the set of virtual tasks S : $A' = A \cup \{A_S\}$.
4. g' is g extended with $g(s) = A_S$ for every $s \in S$.

Example Given Figure 2, the result of the reduction to the plan coordination instance is depicted in Figure 3a. By adding the constraint $t_4 \prec t_2$ it is impossible to create an inter-agent cycle. After normalisation, the resulting schedule coordination instance is given in Figure 3b. ■

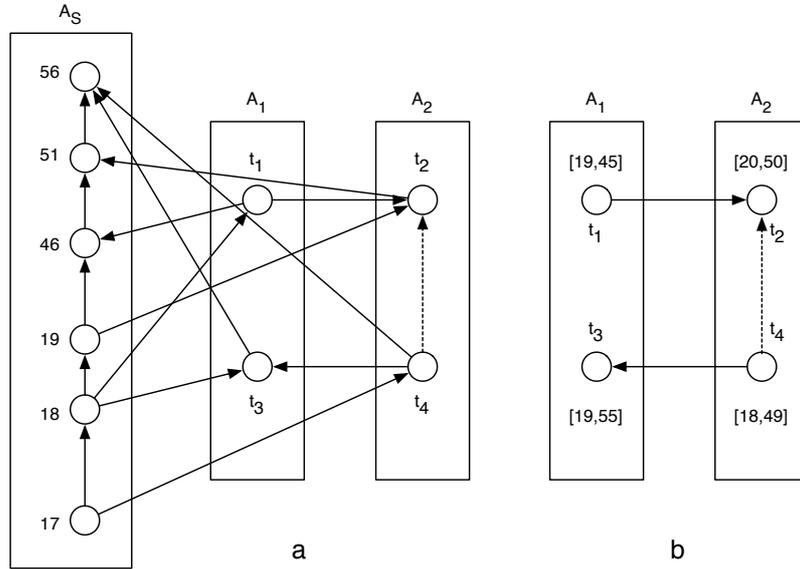


Figure 3: Reducing a temporal coordination problem to a plan coordination problem.

If we determine a coordination set Δ for this instance, we can simply add the resulting set of constraints to the original instance with temporal constraints. The agents are free to form their own plans with respect to these constraints and, if they succeed in finding locally feasible plans, we obtain a cycle free instance of the scheduling coordination problem. This instance then can be solved by applying Algorithm 1.

4 Concluding Remarks

In previous work, we addressed the distributed plan coordination problem. This problem amounts to finding a minimal set of additional constraints that allow a number of agents (planners) to autonomously plan their share of tasks, while guaranteeing a conflict-free global plan. However, this approach only considered the coordination stage and the planning stage: No attention was given to the scheduling phase (i.e., the actual assignment of execution times).

This paper extends our coordination formalism with a mechanism for temporal constraints. Information about the execution time windows of tasks can now be taken into account. We provide a basic coordination algorithm to fully decouple distributed plans. This means that each agent has complete freedom to schedule its tasks at times consistent with his local constraints; The resulting joint schedule will always be valid.

Since removing *all* inter-agent constraints (and thus overconstraining the agents' plans) is often not necessary in practise, we also describe a more sophisticated approach. In short, our method comes down to these steps:

1. Reduce the coordination problem with temporal constraints to a coordination problem without temporal constraints.
2. Determine a solution of the plan coordination problem (i.e., find a set of additional constraints that make the union of agent plans conflict free).
3. Let the agents plan autonomously, that is determine their local orderings of actions.
4. Solve the resulting scheduling problem with temporal intervals by applying the basic coordination algorithm.

4.1 Future Work

The framework that we presented here only handles *absolute* time windows for tasks. In the near future, we will also incorporate relative constraints, making it compliant with the STP theory. This means that durative tasks can be modelled. Also, the relative distance between tasks can be made variable, instead of the constant value of 1.

On top of this, we will progress to the area of tightly-coupled multi-agent plans. More specifically, we will provide methods to encode relations like simultaneity (i.e., when tasks t and t' have to occur at the same time) in our framework.

References

- [1] Pieter Buzing, Adriaan ter Mors, Jeroen Valk, and Cees Witteveen. Coordinating self-interested planning agents. *Autonomous Agents and Multi-Agent Systems*, 12(2):199–218, March 2006.
- [2] Rina Dechter. *Constraint Processing*. Morgan Kaufmann, San Francisco, CA, USA, 2003.
- [3] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence*, 49:61–95, 1991.
- [4] Luke Hunsberger. *Group Decision Making and Temporal Reasoning*. PhD thesis, Harvard University, Cambridge, MA, USA, June 2002.
- [5] Nidhi Kalra, Anthony Stentz, and David Ferguson. Hoplitest: A market framework for complex tight coordination in multi-agent teams. Technical Report CMU-RI-TR-04-41, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA, August 2004.
- [6] Jeroen Valk and Cees Witteveen. Multi-agent coordination in planning. In M. Ishizuka and A. Sattar, editors, *Proceedings of the 7th Pacific Rim International Conference on Artificial Intelligence (PRICAI'02)*, volume 2417 of *Lecture Notes in Computer Science (LNCS)*, pages 335–344, Tokyo, Japan, August 2002. Springer.

Discrete versus Probabilistic Sequence Classifiers for Domain-specific Entity Chunking

Sander Canisius^a Antal van den Bosch^a Walter Daelemans^b

^a *ILK / Language and Information Science, Tilburg University, The Netherlands*

^b *CNTS, University of Antwerp, Belgium*

Abstract

We present a comparative case study of discrete and probabilistic sequence classification methods applied to two real-world entity chunking tasks in the medical domain. It is shown that a discrete version of maximum-entropy models that does not coordinate its decisions is outperformed by both architecturally-augmented discrete versions, and probabilistic versions combined with an inference step to select the best output label sequence. In addition, we show that among the various sequence-aware methods evaluated in this study, be they discrete or probabilistic, no significant performance difference could be observed. This suggests that probabilistic sequence labelling methods are not fundamentally more suited for the type of sequence-oriented entity chunking tasks evaluated in this study than augmented discrete approaches. Future research should point out whether this result generalises to more types of sequence tasks in natural language processing.

1 Introduction

Many tasks in natural language processing involve the complex mapping of sequences to other sequences. One class of processing tasks is the identification of entities in text, or entity chunking. This task involves the identification of the beginning and the end of an entity chunk, which can span multiple words, as well as assigning a particular label to the identified chunk. In domain-specific entity chunking, the label is one out of a limited list of domain labels, the automatic identification of which in unseen text is relevant to further processes, for instance, information retrieval, information extraction, or question answering.

One typical machine-learning approach to entity chunking is to rephrase the sequence-to-sequence mapping task as a decomposition into a sequence of local classification steps. In each step, a fixed-length feature vector is mapped to an isolated symbol in the output sequence. The standard representational approach to decompose sequence processes into local-classification cases, is *windowing*. Within a window, fixed-length subsequences of adjacent input symbols, representing a certain contextual scope, are mapped to one output symbol, typically associated with one of the input symbols, for example the middle one. After all local classifications have been made, a simple concatenation of the predicted output symbols yields the complete output sequence. The fact that the classifier is only trained to associate subsequences of input symbols to single output symbols is a problematic restriction: it may easily cause the classifier to produce invalid or impossible output sequences, since it is incapable of taking into account any decisions it has made earlier, or even decisions it might have to make further on in the input sequence.

Developing techniques that manage to circumvent this restriction has been a popular topic in machine learning research in recent years. Many of the techniques proposed no longer compose output sequences simply by concatenating the isolated predictions for each input symbol, but try to optimise the likelihood of the entire output sequence, rather than predicting the sequence of individually most-likely output symbols. To this end, some way to estimate the quality of

an entire output sequence is required. Methods such as maximum-entropy markov models and conditional random fields express this sequence quality in terms of probabilities, which adds the extra requirement of an underlying classifier capable of predicting valid class probabilities for each possible output symbol.

A large class of classifiers, henceforth referred to as discrete classifiers, do not model such probability distributions, but only predict a single best class label for each given instance. With a windowing approach, only predicting a single class as opposed to giving conditional probability estimates for every possible class means there is basically just one opportunity for predicting each symbol of the output sequence. In the probabilistic case, it is possible to prefer an output label sequence where one of the symbols actually is the second-best option if that means the overall quality of the entire output sequence is improved. Most discrete classifiers, however, lack the means for making such global trade-offs. Nevertheless, discrete classifiers, which include many commonly used learning algorithms such as memory-based learning and decision trees are frequently used for sequence labelling tasks, and in this context, alternatives to the probabilistic methods have been developed, based on clever feature engineering, meta-learning, or post-processing.

In this paper, an empirical study is described in which a maximum-entropy classifier is used both as a discrete classifier and a probabilistic one to perform two domain-specific entity chunking tasks. First, the classifier is taken to be a discrete classifier. In this setup, three existing discrete sequence labelling methods are applied to the sample tasks. The methods try to improve performance on sequence labelling tasks by introducing new features, using classifier stacking, or predicting small subsequences of the output sequence instead of single symbols. Next, a number of methods to exploit the probabilistic character of the maximum-entropy classifier are tested. The methods differ in the way the probability of candidate output label sequences is estimated, the search algorithm employed for finding the optimal output label sequence, or in both. The methods tested are conditional markov models (CMM), maximum-entropy markov models (MEMM), and conditional random fields (CRF).

The structure of the paper is as follows. First, we introduce the two chunking sequence segmentation tasks studied in this paper, in Section 2. The two subsequent sections report on empirical results for the different methods proposed for correcting the near-sightedness problem of a naive baseline method based on simple windowing of the input sequence and no other information. We first report on experiments with discrete classification methods, viz. a feedback-loop approach, stacking, and predicting class trigrams in Section 3. Second, we present results obtained with probabilistic methods, viz. conditional markov models, maximum-entropy markov models, and conditional random fields, in Section 4. In Section 5, we discuss the implications of the experimental findings presented in the two foregoing sections. Finally, Section 6 sums up and discusses the main results of the comparison.

2 Data and Methodology

The two data sets that have been used for this study are examples of sentence-level entity chunking tasks: concept extraction from general medical encyclopedic texts (henceforth MED), and labelling of DNA, RNA, protein, cellular, and chemical terms in MEDLINE abstracts (GENIA). MED is a data set extracted from a semantic annotation of (parts of) two Dutch-language medical encyclopedias. On the chunk-level of this annotation, there are labels for various medical concepts, such as disease names, body parts, and treatments, forming a set of twelve concept types in total. Chunk sizes range from one to a few tokens. The target application for which this data set was developed is a question analyser for a question-answering system for factual medical question. As the input sentences for such a system will typically contain at least one domain-specific concept, all sentences without any concept have been removed from the data set. Using a 90%–10% split for producing training and test sets, there are 428,502 training examples and 47,430 test examples.

Bij [infantiel botulisme]_{disease} kunnen in extreme gevallen [ademhalingsproblemen]_{symptom} en [al-gehele lusteloosheid]_{symptom} optreden.

The GENIA corpus [5] is a collection of annotated abstracts taken from the National Library of Medicine’s MEDLINE database. Apart from part-of-speech tagging information, the corpus annotates a subset of the substances and the biological locations involved in reactions of proteins.

In contrast with the MED data set, GENIA does include sentences that do not contain any concept; in fact, this is the case for many of them. Using a 90%–10% split for producing training and test sets, there are 458,593 training examples and 50,916 test examples.

Most hybrids express both $[KBF1]_{protein}$ and $[NF-kappa\ B]_{protein}$ in their nuclei , but one hybrid expresses only $[KBF1]_{protein}$.

Apart from having a similar size, both data sets are alike in the sense that most words are outside chunks; for GENIA, many sentences may even contain no chunks at all. Thus, the class distributions of both tasks are highly skewed, and only a few words are actually relevant and should be assigned a non-negative class. In this respect, the tasks differ from, for example, syntactic sequence labelling tasks such as part-of-speech tagging or base-phrase chunking, where almost all tokens are assigned a relevant class. However, for all tasks mentioned, whenever chunks are present in a sentence, there is likely to be interaction between them, where the presence of one chunk of a certain type may be a strong indication of the presence of another chunk of the same or a different type in the same sentence.

2.1 Experimental Setup

The experiments in this study have all been performed using the maximum-entropy classification framework. This method is especially suited for the experiments, since maximum-entropy classifiers can both be used in discrete and in probabilistic mode. Any probabilistic classifier can easily be made to emulate a discrete one by taking the target label with the highest conditional probability as the predicted class. By doing this, the discrete sequence labelling methods and their probabilistic counterparts can be compared objectively without having to take into account differences originating, for example, from classifier biases. The current study uses the maximum-entropy classifier as implemented in the maxent toolkit (version 20040930) by Zhang Le¹.

In Section 4, several different probabilistic sequence classification methods are evaluated. Two of them –a maximum-entropy based conditional markov model, and a maximum-entropy markov model– have been implemented on top of the beforementioned maxent toolkit. For a third, a conditional random field model, the implementation as provided by MALLETT [2] has been used.

Instances for all experiments are generated for each token of a sentence, with features for seven-token windows of words and their (predicted) part-of-speech tags. The class labels assigned to the instances form an IOB encoding of the chunks in the sentence, as proposed by Ramshaw and Marcus [6]. In this encoding the class label for a token specifies whether the token is inside (I), outside (O), or at the beginning of a chunk (B). An additional type label appended to this symbol denotes the type of the chunk. The instances are used in exactly this form in all experiments for all methods; no feature selection or construction is performed to optimise the instances for a specific task or method. Keeping the feature vectors unchanged over all experiments and methods is arguably the most objective setup for comparing the results.

Generalisation performance is measured by the precision, recall, and F-score ($\beta = 1$) on correctly identified and labelled entity chunks in test data. Experimental results are presented in terms of a mean score, and an approximate 90%-confidence interval; both of those are estimated with bootstrap resampling [4]. To enhance readability, confidence intervals are assumed to be centred around the mean, where the width of the halves at both sides of the mean is taken to be the maximum of the true widths obtained in the resampling process.

3 Discrete Classification

In this section, a maximum-entropy classifier is treated as though it were a discrete classifier. This is achieved by always taking the target label with highest conditional probability. To start with, a *baseline* score has been established, for which a classifier is trained on examples representing fixed-width windows of input symbols only, mapping to IOB-style class labels, as described earlier. This baseline system is then supplemented with three methods for adding sequence-awareness to discrete classifiers: a feedback loop, stacking, and trigram classes. Each of these will be introduced briefly in the following subsections.

¹URL: http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html

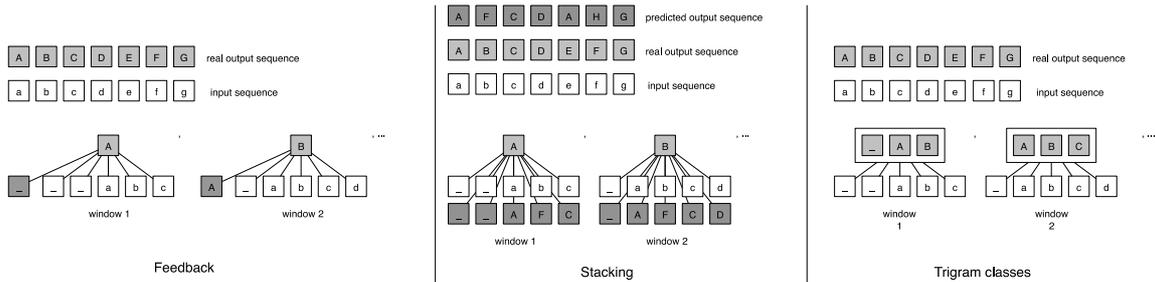


Figure 1: Three types of windowing processes aimed at optimising sequence predictions, generated from an input sequence mapping to an output sequence. Left: a feedback loop copying the previous prediction into the input window. Middle: windows created by the second-stage classifier in a stacking architecture, copying predictions of the first-stage classifier into the input window. Right: windows mapping to trigrams of classes.

Table 1: Mean performance scores and confidence intervals for the various classification methods on the GENIA task.

METHOD	PRECISION	RECALL	$F_{\beta=1}$
BASELINE	54.9 \pm 1.16	54.1 \pm 1.23	54.5 \pm 1.02
FEEDBACK	67.3 \pm 1.04	57.6 \pm 1.25	62.1 \pm 1.11
STACKING	57.8 \pm 1.21	55.3 \pm 1.07	56.5 \pm 1.11
TRIGRAM	61.8 \pm 1.15	56.0 \pm 1.27	58.8 \pm 1.12
CMM	67.7 \pm 0.96	57.9 \pm 1.07	62.4 \pm 1.01
MEMM	67.1 \pm 1.14	57.7 \pm 1.13	62.1 \pm 1.15
CRF	66.8 \pm 1.10	59.2 \pm 1.14	62.8 \pm 1.08

3.1 The Feedback-loop Method

One method for providing a classifier access to its previous decisions is a feedback-loop approach, which extends the windowing approach by feeding previous decisions of the classifier as features into the current input of the classifier. An early application of this method in the context of maximum-entropy classification is the work of Ratnaparkhi on part-of-speech tagging [7]. The left part of Figure 1 illustrates how the windowing approach is extended with information about previous classifier decisions as input features. The number of decisions fed back into the input can be varied. In the experiments described here, the feedback loop iteratively updates a memory of only the single most recent prediction, which was experimentally found to be an adequate setting.

3.2 Stacking

Stacking, a term popularised by Wolpert [9] in an artificial neural network context, refers to a class of meta-learning systems that learn to correct errors made by lower-level classifiers. We implement stacking by adding a windowed sequence of previous and subsequent output class labels to the original input features (here, we copy a window of three predictions to the input, centred around the middle position), and providing these enriched examples as training material to a second-stage classifier. The middle part of Figure 1 illustrates the procedure. Given the (possibly erroneous) output of a first classifier on an input sequence, a certain window of class symbols from that predicted sequence is copied to the input, to act as predictive features for the real class label.

3.3 Predicting Class Trigrams

Van den Bosch and Daelemans [8] recently proposed a new discrete method for sequence labelling based on predicting trigrams of class labels. In this method, each token in a sequence is labelled with a complex class label composed of the labels of the token itself, and those of the tokens directly

Table 2: Mean performance scores and confidence intervals for the various classification methods on the MED task.

METHOD	PRECISION	RECALL	$F_{\beta=1}$
BASELINE	62.3 \pm 1.12	60.8 \pm 1.06	61.5 \pm 0.98
FEEDBACK	68.5 \pm 1.16	60.0 \pm 1.13	63.9 \pm 0.89
STACKING	63.2 \pm 1.23	60.8 \pm 1.13	62.0 \pm 1.10
TRIGRAM	66.9 \pm 1.07	59.6 \pm 1.01	63.1 \pm 1.08
CMM	68.8 \pm 1.26	59.6 \pm 1.09	63.9 \pm 0.99
MEMM	68.8 \pm 1.09	59.3 \pm 1.26	63.7 \pm 1.09
CRF	66.8 \pm 1.14	60.2 \pm 1.14	63.4 \pm 0.99

to its left and to its right in the sequence. Doing this for each token of a sequence, every token’s label is effectively predicted three times: once by the preceding token’s trigram, once by its own trigram, and once by the following token’s trigram. Aiming to exploit this redundancy, a simple majority voting rule is applied to the overlapping and possibly conflicting predictions for a certain token. If all three overlapping trigrams predict a different class for a token, the confidence of the classifier for the predicted trigrams is used to break the tie.

The right part of Figure 1 illustrates the procedure by which windows are created with class trigrams. Each windowed instance maps to a class label that incorporates three atomic class labels, namely the focus class label that was the original unigram label, plus its immediate left and right neighbouring class labels.

3.4 Results

The top half of Tables 1 and 2 shows the performance scores for the discrete sequence labelling methods introduced in this section, and compares them with the performance of a naive baseline classifier that treats each token as an isolated classification case. The error reductions with respect to F-score attained by the best method, the feedback-loop method, is 6.2% for MED, and 16.7% for GENIA.

4 Probabilistic Classification

Rather than suggesting one target label that is expected to be the most likely class for a test instance, as do discrete classifiers, probabilistic classifiers compute the conditional probability for each possible target label. This property opens up interesting possibilities for considering alternative label sequences, different from the one obtained by concatenating all individually most likely token labels. As a consequence of the probabilistic interpretation that can be given to target labels, the probability of an entire sequence of labels can be determined using simple multiplication, provided certain independence requirements are met.

Sequence labelling methods based on probabilistic classifiers can generally be decomposed into two components: the first is concerned with estimating the conditional probability of a candidate output label sequence; the second dictates a procedure for efficiently finding the best label sequence out of all possible candidates. In this section, three different methods based on this general framework are applied to the sample tasks: conditional markov models, maximum-entropy markov models, and conditional random fields.

4.1 Conditional Markov Model

Conditional markov models (CMM), as used for example by Ratnaparkhi [7], can be seen as a probabilistic extension of the feedback-loop method described in Section 3.1. As with the method described earlier, a prespecified number of previous decisions of the classifier are fed back to the input as features for the current test instance. Unlike in the discrete case however, classification of

a token does not yield one partial labelling, namely, the partial labelling up to the current token followed by the current classification, but as many partial labellings as there are target labels, namely the partial labelling until the current token followed by any of the possible target labels.

As the use of a feedback loop makes the current classification depend on the results of previous classifications, each token in the sequence has to be classified in the context of each possible partial labelling up to that point. Clearly, this approach, if applied naively, gives rise to an exponential increase in possible partial labellings at each token. Therefore, CMMS employ a beam search to find the eventual best labelling. With beam search, at each point in time, only a prespecified number of partial labellings – those having highest probability – are considered for expansion, all the other candidates are discarded.

4.2 Maximum-entropy Markov Model

Another implementation of the idea of supplementing a probabilistic classifier with a search procedure is the maximum-entropy markov model (MEMM), proposed by McCallum et al. [3]. Derived from hidden markov models, MEMMs are modelled after a probabilistic state machine, in which, in the simplest case, a state corresponds to the output label of the previous token, and for each state, a separate conditional probability distribution determines the next state, that is, the output label for the current token, given the feature vector of this token. A slight modification of the Viterbi algorithm is used to determine the optimal path through the state machine given the input sequence.

4.3 Conditional Random Fields

Conditional random fields (CRF) [1] have been designed to resolve some of the shortcomings of MEMMs. The main difference lies in the number of probabilistic models used for estimating the conditional probability of a label sequence: MEMMs use a separate probabilistic model for each state, whereas CRFs have a single model for estimating the likelihood of an entire label sequence. The use of a single probabilistic model leads to a more realistic distribution of the probability mass among the alternative paths. As a result, CRFs tend to be less biased towards states with few successor states than CMMS and MEMMs.

4.4 Results

The performance scores for the probabilistic sequence labelling methods applied to the two benchmark tasks are listed in the bottom half of Tables 1 and 2. On MED, CMM attains the highest score, equal to the one obtained by the discrete feedback-loop method; the error reduction in F-score is 6.2%. The overall highest score on GENIA is obtained by CRF, with an error reduction of 18.2%. However, in none of the cases is the difference with the other high-ranked methods significant.

5 Discussion

The previous two sections reported on a series of experiments in which several machine learning approaches to domain-specific entity chunking have been evaluated. The approaches can be divided into an approach that ignores any sequential context apart from its window features, and those that have been extended specifically to deal with sequentially-structured data. Members of the latter group, in turn, can be categorised as either discrete or probabilistic methods. All of these approaches have been tested against two data sets; both of which are quite similar domain-specific entity chunking tasks. However, one striking difference is the presence of sentences without any concept in GENIA, and the absence of those in MED.

In order to put the results obtained in the previous two sections in perspective, this section will review the effect of the factors mentioned above.

5.1 Sequence-aware vs. sequence-unaware methods

As was already mentioned in the introduction of this paper, machine learning approaches not specifically equipped for classifying sequentially-structured data often have difficulty performing

tasks in which token labels are sequentially correlated; a situation which can be improved upon by more special-purpose sequence classification methods. The findings of the current study can only confirm this assumption. Comparing the performance of the baseline system with that of the best-performing sequence-oriented approach, the error in F-score is reduced by 18.2% on GENIA, while on MED a somewhat more modest but still significant reduction of 6.2% is observed.

A remarkable observation is that the sequence-oriented methods, be they discrete or probabilistic, all improve precision more strongly than recall. Whereas the two are reasonably in balance for the baseline system, there is a striking imbalance for the sequence methods. This situation is most apparent on the MED task, where the recall is not improved at all. From this, it can be concluded that the sequence methods do not necessarily predict more true positives –since recall would benefit from this as well– but rather manage to decrease the total number of positive predictions without negatively affecting the number of true positives. To a lesser extent, the same goes for the GENIA task.

5.2 Probabilistic vs. discrete approaches

The sequence-oriented methods surveyed in this paper are either based on a probabilistic base classifier, in case of which various different output label sequences are traded off on the sequence level; or on a discrete classifier, which only has a single opportunity for classifying each input token. In recent years, probabilistic approaches tend to be preferred for sequence-oriented tasks. It is claimed that discrete methods, being unable to support viterbi-like inference procedures, must resort to sub-optimal methods to classify sequential data [3].

However, the results presented in this paper do not support this claim. In fact, no significant differences in performance have been observed between the best performing probabilistic and discrete methods; neither in precision, nor in recall, and consequently not in F-score either. These findings suggest that, although discrete methods are unable to trade off entire sequence labellings on a global level, at least for the two tasks evaluated in the current study, the approaches used by the discrete methods provide suitable alternatives. Whether this observation holds more generally for other types of sequence processing tasks as well, will need to be pointed out by more extensive experimental studies.

5.3 Concept sparseness

One of the main differences between the two data sets used in this study lies in the distribution of concepts in sentences. In the MED data, every sentence contains at least one concept, whereas in GENIA, sentences without any concept occur quite frequently as well. This difference might provide a partial explanation for the different behaviour of the sequence methods with respect to recall on the two data sets. From the large difference between precision and recall, we already concluded that the sequence methods primarily improve precision by predicting fewer positives, rather than predicting more true positives. However, when predicting fewer positives, there is always the risk of missing some positive labels, that is, predicting false negatives, and thereby committing recall errors. A possible explanation for the better recall performance on GENIA then may be the fact that the chance of committing recall errors is simply lower on data sets that comprise a lower percentage of chunks to begin with.

6 Conclusion

Classifiers trained on entity chunking tasks that make isolated, near-sighted decisions on output symbols and that do not optimise the resulting output sequences afterwards or internally through a feedback loop, tend to produce weaker models for sequence processing tasks than classifiers that do. The two entity chunking tasks investigated in this paper are challenging tasks; not only because they demand the classifier to be able to segment and label variable-width chunks while obeying the syntax of the chunk analysis, but also because positive examples of labelled chunks are scattered sparsely in the data.

In this paper, special-purpose sequence labelling methods have been classified into two main categories. One of these categories expects the underlying classifier to be probabilistic, that is,

rather than predicting a single discrete class for each test instance, it should estimate the conditional probability of each possible target class. Given such a probabilistic output, a search algorithm is employed to find the most likely label sequence, as opposed to the sequence of individually most likely class predictions.

The other category uses classifiers that do not model conditional probability distributions, but instead predict a single most likely output class for each test instance. For this type of classifier there is a wide range of methods that aim to improve performance on sequence labelling tasks, including adding a feedback-loop to the classification procedure, classifier stacking, and predicting class trigrams.

In a series of experiments, several discrete, and probabilistic sequence labelling methods implemented on top of a maximum-entropy base classifier were applied to two benchmark tasks; both of them, high-level entity chunking tasks in the medical domain. Results from these experiments clearly show that the enhanced methods easily improve upon a baseline classifier that considers each token as a separate, isolated classification case. More surprising is the fact that there are hardly any differences in performance among the various sequence-oriented methods. In particular, when it comes to the type of sequence-oriented entity chunking tasks targeted in this study, we did not find any evidence supporting the claim that probabilistic methods have an advantage over discrete ones. This leads us to the assertion that, for this specific type of sequential tasks, no method is fundamentally more suited than another. In practise, this means that other factors than whether or not a classifier is capable of probabilistic predictions, may guide the choice for the specific type of learning method to use. In future work, we intend to investigate whether this result generalises to more types of sequence processing tasks in natural language processing.

Acknowledgements

This research was funded by NWO, the Netherlands Organization for Scientific Research, as part of the IMIX Programme.

References

- [1] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, Williamstown, MA, 2001.
- [2] A. McCallum. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [3] A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the 17th International Conference on Machine Learning*, Stanford, CA, 2000.
- [4] E. Noreen. *Computer-intensive methods for testing hypotheses: an introduction*. John Wiley and sons, 1989.
- [5] T. Ohta, Y. Tateisi, J.-D. Kim, H. Mima, and J. Tsujii. Genia corpus: an annotated research abstract corpus in molecular biology domain. In *Human Language Technology Conference (HLT 2002)*, pages 73–77, 2002.
- [6] L.A. Ramshaw and M.P. Marcus. Text chunking using transformation-based learning. In *Proceedings of the 3rd ACL/SIGDAT Workshop on Very Large Corpora, Cambridge, Massachusetts, USA*, pages 82–94, 1995.
- [7] A. Ratnaparkhi. A maximum entropy part-of-speech tagger. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing, May 17-18, 1996, University of Pennsylvania*, 1996.
- [8] A. Van den Bosch and W. Daelemans. Improving sequence segmentation learning by predicting trigrams. In I. Dagan and D. Gildea, editors, *Proceedings of the Ninth Conference on Computational Natural Language Learning*, 2005.
- [9] D. H. Wolpert. Stacked Generalization. *Neural Networks*, 5:241–259, 1992.

random games until the end. The outcome of these games is then statistically analyzed, e.g., by calculating the mean outcome. This value then constitutes the evaluation of s . Monte Carlo evaluation has been applied successfully in non-deterministic games, such as Bridge [10], Scrabble [8] and Poker [9]. In addition, it was introduced recently in deterministic games [7]. In the game of Go, which is resistant to easily developed evaluation functions [6], Monte-Carlo sampling techniques have yielded convincing results recently, posing an alternative to common knowledge-intensive evaluation functions.

Given the success of search algorithms that use Monte-Carlo evaluation in adversarial games (i.e., a multi-agent setting), this article examines the possibility of applying such a search algorithm, *Monte-Carlo Tree Search* (MCTS), to a single-agent task domain.

Production management problems. As a task domain for a comparison between our proposed algorithm MCTS and other algorithms, we use *production management problems* (PMPs) [12]. In summary, these problems can be defined as planning problems which require parameter optimization over time and can be addressed by the selection of actions with side effects. They contain the following four elements. First, there is a fixed set of *products* – the *size* of the problem is equivalent to the size of this set. Second, there is a fixed set of *production actions*, which are used to convert certain products into other products or to obtain one or more products. These actions may require time and money. Some actions may also produce money (selling actions). Third, there are *constraints* such as a limited amount of time or money available. Fourth, the goal for a problem solver is to produce as much as possible of some of the products, denoted as the *goal products*. This can be achieved by developing an optimal sequence (or plan) of actions, given the available products, actions, constraints and money. The complexity of these problems grows exponentially with the number of products and actions, as will be shown below.

In [12], PMPs have been addressed by applying two planning algorithms; one using a fixed heuristic (Fixed Planning Heuristics or FPH), and one using an evolutionarily optimized heuristic (Evolutionary Planning Heuristics or EPH). The authors conclude that EPH performs better than FPH, mainly because EPH is able to take into account the current state of the problem and because no limit was imposed on the time available to find a solution. It was observed that large PMPs (i.e., with 30 products) require a very long optimization time before a satisfactory outcome is reached, and FPH sometimes leads to better results in these problems, as EPH terminates with a clearly suboptimal solution.

Our contribution. In this paper, we introduce Monte-Carlo Tree Search (MCTS) as a potential method for production management problems. We show that (i) MCTS reaches at least the same score as EPH in problems of sizes $n \leq 24$ while using substantially less calculation time; (ii) MCTS outperforms EPH both in score and runtime in problems of size $24 < n \leq 40$ and (iii) the MCTS algorithm is able to address even larger problems in reasonable time.

The performance of MCTS on PMPs gives us information on the expected performance of this algorithm on other task domains related to search and games, for four reasons, viz. (1) PMPs resemble the problems typically addressed by search algorithms in games (i.e., complete information, discrete time steps), (2) PMPs are easy to formalize and implement, (3) PMPs are intractable problems, yet they are scalable to any desired size and (4) it is possible to develop difficult PMPs, even if these problems do not require adversarial planning or reasoning with uncertainty.

Structure of this article. In Section 2, we present a formalization of PMPs and briefly discuss the complexity of such problems. In Section 3, we discuss the algorithms used in previous research and this research. In Section 4, we look at the experimental setup and the results obtained with regard to performance and running times of the algorithms discussed. In Section 5, we discuss results, conclude and look at future work.

2 Formalizing production management problems

Formalization. In this section, we present a formalization of production management problems. As in [12], we focus on a simplified version for which we ignore non-determinism and earning additional money that can be utilized to cover the costs of actions. The resulting problem can be formalized as follows.

- P is the set of products $\{p_1, \dots, p_n\}$. The *size* of the problem, $|P|$, is denoted by n .
- The possibly infinite set of possible problem states is denoted by S . The problem state $s_t \in S$ at a certain moment in time (starting with s_0) is a tuple $(m_{s_t}, q_{s_t}(p))$. Here, m_{s_t} is the money available in this state, and the function $q_{s_t} : P \rightarrow \mathbb{N}$ defines the quantity available for each product in the state s_t .
- $G \subset P$ is the set of goal products. The reward per goal product, $r : P \rightarrow \mathbb{R}^+$, is specified by $p \notin G \rightarrow r(p) = 0$ and $p \in G \rightarrow r(p) > 0$. The total reward in a state s_t can be calculated using the function $R(s_t) = \sum_{p \in G} r(p) \cdot q_{s_t}(p)$. The goal of the problem is to reach a state s_t in which $R(s_t)$ is optimal.
- A denotes the set of actions that enable the transition from one state to another. Here, $c : A \rightarrow \mathbb{N}^+$ denotes the cost of action a . Due to this cost, we will have less money to spend in every subsequent state, and thus we ensure that any problem will have a finite number of possible actions. $t : A \rightarrow \mathbb{N}$ denotes the time action a takes to complete, assuming discrete timesteps. The function $in : A \rightarrow \mathcal{P}(P \times \mathbb{N}^+)$ denotes the amounts of products required to execute a ; $out : A \rightarrow \mathcal{P}(P \times \mathbb{N}^+)$ denotes the amounts of products produced by a if it is executed.

Additional constraints. For this research, we introduce five constraints in addition to this formalism, also introduced in [12]. PMPs based on the formalism and these additional constraints fall more or less within the same class of difficulty and possess some convenient properties such as a strict ordering of actions and guaranteed solvability. However, PMPs that adhere to these additional constraints are not essentially easier than PMPs that do not adhere to them.

1. Every action requires at most two products and produces one or two products. Every product is produced by exactly two actions.
2. Costs and durations of actions can be selected from a limited domain for all $a \in A$: $c(a) \in \{1, 2, 3, 4\}$ and $t(a) = 1$. We limit the domain for the coefficients x for required and produced products p by implying the condition $(p, x) \in in(a) \vee (p, x) \in out(a) \rightarrow x \in \{1, 2, 3, 4\}$.
3. Cycles in the product chain are prevented: An action cannot produce products with a lower index than the highest index of its required products, plus one.
4. We define $G = \{p_n\}$ and $r(p_n) = 1$.
5. The problem solver starts with an initial amount of money equal to $20 \cdot n$ and none of the products present. The number $20 \cdot n$ is intentionally rather small to keep the problem complexity manageable. Thus, $s_0 = (m_{s_0}, q_0)$ with $m_{s_0} = 20 \cdot n$ and $p \in P \rightarrow q_0(p) = 0$.

A small example PMP that adheres to the formalism and constraints presented here, is shown in Figure 1. For legibility, all coefficients have been omitted. Circles represent actions and squares represent products. For example, action A requires products p_2 and p_3 and produces the product p_4 . The goal product is p_4 , which can be produced by using various product chains, e.g., by using action C to produce products p_1 and p_2 , and then using action B to produce p_3 and p_4 . Some other product chains are also possible.

Complexity. Experimental results obtained using an algorithm that generates random instances of PMPs adhering to both the formalism and the additional constraints, show that $|A| \approx n + 1$. Solving such PMPs is a hard task. Using a mathematical formalization of PMPs, we were able to reduce the NP-Hard 3SAT problem to a PMP, showing that PMPs are also NP-Hard. This reduction is not included here due to space constraints.

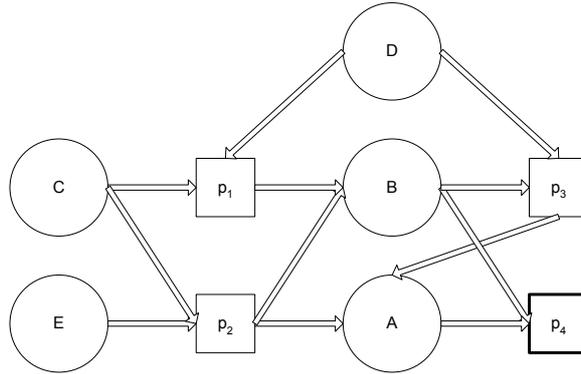


Figure 1: Example of a PMP that adheres to the formalism and constraints presented in this paper.

3 Methods

Fixed Planning Heuristics. In [12], two algorithms, both based on abductive planning, are proposed and tested, viz. Fixed Planning Heuristics (FPH) and Evolutionary Planning Heuristics (EPH). The FPH algorithm generates a plan every time an action must be selected. As a first step, the algorithm determines the actions that produce the goal product p_n (in Figure 1, $n = 4$ and actions A and B produce the goal product p_4). From these actions, it chooses the one that contributes the most to the goal at the lowest price. If this action is executable, it is returned as the action that will be executed. If it is not executable, the algorithm finds the best action that makes it as executable as possible. This criterium is determined using a heuristic that provides approximate information; it cannot be calculated exactly for all actions, due to the problem structure [12]. This process continues until an action is found that is executable, and this action is then chosen as the current action.

Note that the requirement for a heuristic in FPH has a lot in common with the requirement of an evaluation function in search algorithms. For a complete explanation of the heuristic designed for PMPs, we refer to [12].

Evolutionary Planning Heuristics. The Evolutionary Planning Heuristics (EPH) algorithm is a combination of FPH and evolutionary computation. Instead of a manually designed heuristic for the traversal of the plan-space search tree, the algorithm uses a heuristic that is generated by a neural network. The input of this network consists of the current problem state s_t . The output consists of a preference value for each action. The neural network is trained by weight optimization with a genetic algorithm, by means of offline learning. A single problem is repeatedly offered until the algorithm, starting from the state s_0 , is able to reach a state s in which the reward $R(s)$ is satisfactory.

This algorithm is interesting because it eliminates the need for a manually developed heuristic, or more broadly, the need for a manually developed evaluation function. As has been remarked above, the elimination of this need is a valuable contribution to the field of planning and search algorithms because many domains are resistant to manually developed evaluation functions. Since the research presented in [12] indicates that EPH, an algorithm without a manually developed evaluation function, is able to address PMPs rather successfully, we are interested in applying another such algorithm: Monte-Carlo Tree Search.

Monte-Carlo Tree Search. In summary, Monte-Carlo Tree Search is a tree search and expansion algorithm which uses Monte-Carlo sampling as its evaluation function and is both explorative and exploitative. Each node of the search tree contains a current problem

state, the possible actions, and some other information about this state, i.e., the mean value and the standard deviation of the simulations made starting from this node.

The algorithm works in two steps. First, the root node is expanded recursively to a certain depth (6 or 7 in our experiments). The current score of all newly created nodes is initially set to 0, because it is not known yet how to build the goal product. Second, a certain number of *simulated games* are played, according to Algorithm 1, which uses Algorithm 2 to traverse the inner nodes of the tree.¹ After every simulated game, the current score of all nodes on the path to the root of the tree is updated in Algorithm 1 according to the information obtained in the simulated game.

```

Data: Current game tree
Result: Update current_score
current_node ← root_node
while current_node is not a leaf node do
  | current_node ← action_selection(current_node)
end
while number_of_possible_actions ≥ 1 do
  | Perform_random_action();
end
value_of_this_simulation ← amount_of_goal_product_created
if value_of_this_simulation > score then
  | score ← value_of_this_simulation
end

```

Algorithm 1: Playing a simulated game.

```

Data: Current node
Result: Next node
Obj ← value_of_the_current_best_child_node
foreach child node m do
  | Vm ← mean_value_of_node
  | σm ← standard_deviation_of_Vm
  | Um(Obj) ← Erfc( $G \cdot \frac{O_{bj} - V_m}{\sqrt{2} \cdot \sigma_m}$ )
end
foreach child node m do
  |  $P_m \leftarrow \frac{U_m(O_{bj})}{\sum_{i \in M} U_i(O_{bj})}$ 
end
return node m with a probability Pm

```

Algorithm 2: Action-selection strategy.

In Algorithm 2, *Erfc* represents the complementary error function, i.e.,

$$Erfc(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-u^2} du$$

Moreover, *meanvalue_of_node* is the average of all the random games that have been played from this node, which is updated incrementally. *G* denotes the *greediness* of the algorithm. With a greediness of 1, the algorithm would stop considering an action only when it has been proved to be inferior to its siblings. With such a setting, the depth which can be achieved is very shallow. Otherwise than in adversarial searches, it is not mandatory

¹Algorithm 2 is based on the Central Limit Theorem and has common points with the algorithms developed for the Multi-Armed Bandit Problem such as Softmax or Exp3 [11]. Implementing and comparing such other algorithms with MCTS will be a topic of interest for future research.

Size	10	15	20	25	30	35	40	50
Problems tested	400	400	400	500	432	500	400	200
EPH								
Score	131.7	182.6	171.8	170.6	216.6	183.0	152.4	-
Score var	6.1	9.8	8.2	7.6	13.4	7.7	8.8	-
Time (s)	6.7	22.2	48.1	85.6	191.5	293.3	552.0	-
MCTS								
Score	112.0	158.6	159.5	178.9	241.4	226.7	243.4	278.4
Score var	6.2	9.9	8.2	7.6	13.5	8.0	9.9	24.4
Time (s)	5.4	11.5	20.4	40.2	92.9	170.3	241.2	462.0
Ran								
MCTS>EPH	0.0	0.0	0.0	0.999	1.0	1.0	1.0	-
tMCTS<tEPH	1.0	1.0	1.0	1.0	1.0	1.0	1.0	-
Count								
MCTS>EPH	20.0%	24.3%	36.5%	55.6%	61.9%	72.9%	88.0%	-

Table 1: Results for EPH and MCTS on problems with various sizes.

that the algorithm explores every possible answer of the opponent. Thus, the greediness parameter can be set to a larger value (around 100 in our experiments). The algorithm then performs so-called *forward pruning* and is able to explore to a depth greater than 200.

4 Experiments and results

In this section, we will discuss our experimental setup and results obtained. In our experiments, we offer a large number of PMPs, varying in size between $n = 10$ and $n = 50$. All problems adhere to the formalism and the additional constraints presented in Section 2. They are generated randomly, but use a seed so that we can compare the algorithms on the same problems.

Comparison of EPH with FPH. In [12], the authors compared FPH and EPH on problems of size $n \leq 30$. The average scores obtained by EPH for problems of sizes up to 30 were better than those obtained by FPH. A randomization test [13] was used to determine the probability that EPH was indeed better. For sizes up to $n = 21$, this probability was 1; for larger problems, it decreased slowly to 0.889 for size $n = 30$. Further results for FPH are omitted here. For a complete overview, we refer the interested reader to [12].

Comparison of MCTS with EPH. In our experiments, we compare EPH and MCTS on problems of size 10, 15, . . . , 40. Both the MCTS and EPH algorithms are allowed to run until no further improvement is made for a fixed number of iterations (100 generations of 50 individuals in EPH, 50,000 simulations in MCTS). Results are given in Table 1. First, we indicate the number of problems experimented on. Second, we show average scores including sample variances (i.e. σ/\sqrt{n}), and average running times per problem. A graphical representation of the average running times of MCTS and EPH is given in Figure 2(a). Third, we include the results of two randomization tests, indicating the probabilities that (i) MCTS reaches a better average score than EPH and (ii) MCTS is faster on average than EPH. All randomization tests were performed with 10,000 iterations. Fourth, we print the percentage of problems on which MCTS achieved a better score than EPH.

In the table, we see that, given the current termination conditions, MCTS achieves a lower score than EPH on smaller problems, but is faster. When we increase the problem size, MCTS eventually scores higher than EPH, while maintaining faster performance.²

MCTS on even larger problems. MCTS, unlike EPH, is able to find a solution for prob-

²Since MCTS is both running faster and scoring less for sizes up to $n = 20$, we assessed the effect of changing the termination conditions for MCTS: we increase the number of simulations allowed without further improvement to 75,000. We observed that MCTS becomes significantly slower than EPH, but still does not manage to reach the same score. Thus, EPH is clearly a better algorithm for small problems and MCTS wins on large problems. The exact problem size for which MCTS becomes both faster and better than EPH with high probability depends on the settings chosen (for MCTS: greediness, termination conditions; for EPH: population size, mutation probability, et cetera) and the actual problems addressed.

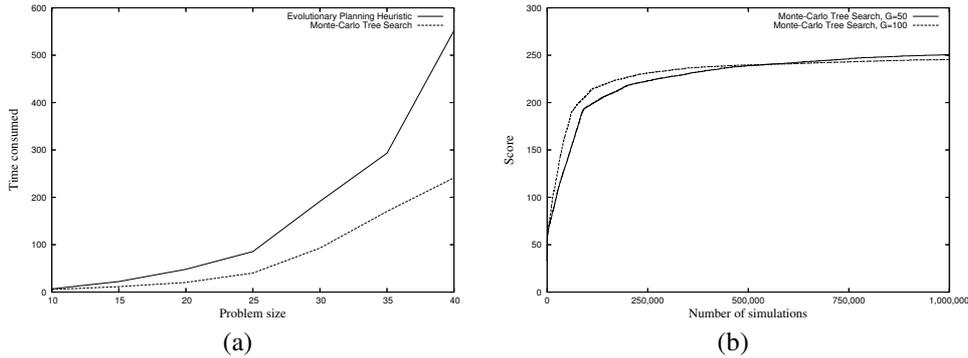


Figure 2: (a) Time consumed by MCTS and by EPH. (b) Speed of convergence of MCTS.

lems with $n = 50$ within reasonable time. Using the current termination conditions, EPH is too slow to solve large numbers of such problems. With less time-consuming conditions, the algorithm’s performance drops rapidly (as the authors observed while tuning them). In MCTS however, we can change the termination conditions in order to lower the running time without losing much of the score, as will be shown below. The results presented for $n = 50$ in Table 1 were obtained by running exactly 500,000 simulations for every problem.

Convergence of MCTS. We conduct an experiment on our test set of problems of size $n = 30$ to determine the convergence speed of MCTS when we (i) change the greediness parameter and (ii) change the termination conditions (i.e., we perform more or less simulations). We run a fixed number of simulations (1,000,000) on our test set and average the results found during and after these simulations. Two different greediness factors G have been tested: $G = 50$ and $G = 100$. Results are reported in Figure 2(b).

One can observe that changing the greediness by a factor of two does not influence the average result much, neither at the end of the simulation, nor during the simulation. Moreover, it can be seen that the algorithm would have performed nearly as well (230 on average instead of 241) with only half of the simulations (250,000 instead of 500,000); i.e., the algorithm would have been 50% faster and would have scored only 4% lower. Thus, changing the termination conditions does not drastically influence the score achieved, and we can make the algorithm substantially faster by allowing a slightly lower score.

5 Discussion and conclusion

In this paper, we presented a Monte-Carlo search framework for single-agent problems and games. It was compared to an adaptive algorithm presented in earlier research. The task domain for this comparison, the production management problem, has three major characteristics: (i) the problem is NP-Hard, (ii) the search space is scalable and (iii) it is difficult to build a good evaluation function. Hence, this problem is an adequate testbed for new search methods.

Earlier research [12] resulted in an evolutionarily optimized heuristic for a planning algorithm (EPH), and in this article, we present a search algorithm that employs a Monte-Carlo based evaluation function (MCTS). Experiments show that on small problems (e.g., with 10 products), EPH achieves a better score than MCTS. On larger problems (i.e., with 25 or more products), MCTS is faster than EPH and achieves much better scores. Furthermore, we see that MCTS can be modified in such a way that running times decrease significantly while scores decrease only slightly. Whether running times or scores are chosen as the main criterium, depends on the task domain.

In general, Monte-Carlo Tree Search can lead to good results in task domains which have the following characteristics: (i) the search space is large; (ii) there is no obvious evaluation function; and (iii) there are several different paths to achieve a good score. We show in this paper that Monte-Carlo Tree Search constitutes an interesting alternative to an evolutionary algorithm. Our next research issue will be to assess the performance of our algorithm in other complex task domains to which other computational intelligence methods have already been applied, including adversarial settings.

We believe that the research reported here can still be improved in several ways. Among them, promising to us are playing pseudo-random games instead of completely random games (which requires some domain knowledge) and the automatic adjustment of the greediness parameter in the algorithm.

Acknowledgments This work is financed by the Dutch Organisation for Scientific Research (NWO) in the framework of the project Go for Go, grant number 612.066.409. It is partially funded by the 'Breedtestrategie' programme of the Universiteit Maastricht. The authors wish to thank Ida Sprinkhuizen-Kuyper, Nico Roos and Jeroen Kuipers for their contributions to this research. And finally, thanks to Sophia Katrenko for providing wireless facilities when needed.

References

- [1] Judea Pearl. The solution for the branching factor of the alpha–beta pruning algorithm and its optimality. *Communications of the ACM*, 25:559–564, 1982.
- [2] Jonathan Schaeffer, Robert Lake, Paul Lu, and Martin Bryant. CHINOOK: The world man-machine checkers champion. *AI Magazine*, 17(1):21–29, 1996.
- [3] Dennis Breuker. *Memory versus Search*. PhD thesis, University of Maastricht, 1998.
- [4] Bruno Bouzy. History and territory heuristics for Monte-Carlo Go. In K. Chen et al, editor, *Joint Conference on Information Sciences*, page 4, 2005.
- [5] A. Nareyek. Choosing search heuristics by non-stationary reinforcement learning. In M. G. C. Resende and J. P. de Sousa, editors, *Metaheuristics: Computer Decision-Making*, pages 523–544. Kluwer Academic Publishers, 2003.
- [6] Martin Müller. Not like other games – why tree search in Go is different. In *Fifth Joint Conference on Information Sciences (JCIS)*, 2000.
- [7] Bruno Bouzy and Bernd Helmstetter. Monte Carlo Go Developments. In H. Jaap van den Herik, Hiroyuki Iida, and Ernst A. Heinz, editors, *Proceedings of the Advances in Computer Games Conference (ACG-10)*, pages 159–174. Kluwer Academic Publishers, 2003.
- [8] Brian Sheppard. World-championship-caliber scrabble. *Artificial Intelligence*, 134(1):241–275, 2002.
- [9] Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. The challenge of poker. *Artificial Intelligence*, 134(1):201–240, 2002.
- [10] Stephen Smith, Dana Nau, and Tom Throop. Computer Bridge - A Big Win for AI planning. *AI Magazine*, 19(2):93–105, 1998.
- [11] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.
- [12] Steven de Jong, Nico Roos, and Ida Sprinkhuizen-Kuyper. Evolutionary planning heuristics in production management. In *Proceedings of the BNAIC 2005, Brussels, Belgium*, 2005.
- [13] Paul R. Cohen. *Empirical Methods for Artificial Intelligence*. MIT Press, 2002.

Monte-Carlo Strategies for Computer Go

Guillaume Chaslot ^a Jahn-Takeshi Saito ^a Jos W.H.M. Uiterwijk ^a
Bruno Bouzy ^b H. Jaap van den Herik ^a

^a *MICC/IKAT, Universiteit Maastricht, The Netherlands*

^b *CRIP5, Université René Descartes, Paris, France*

Abstract

The game of Go is one of the games that still withstand classical Artificial Intelligence approaches. Hence, it is a good testbed for new AI methods. Amongst them, Monte-Carlo led to promising results. This method consists of building an evaluation function by averaging the outcome of several randomized games. The paper introduces a new strategy, which we call Objective Monte-Carlo, to improve this evaluation. Objective Monte-Carlo is composed of two parts. The first one is a move-selection strategy that adjusts the amount of exploration and exploitation automatically. We show experimentally that it outperforms the two classical strategies previously proposed for Monte-Carlo Go: Simulated Annealing and Progressive Pruning. The second part of our algorithm is a new backpropagation strategy. We show that it gives better results than Minimax in this context. Finally we discuss the extension of this method to other problems.

1 Introduction

Go is a 4000 years old game with simple rules. Nowadays, it is played by more than 40 million people, mostly in Asia, and there are hundreds of Go professionals. In Japan, China, and Korea, several TV channels and newspapers are dedicated to this game. Due to its simple rules and its fascinating structure, it constitutes an ongoing challenge for AI researchers. Up to 40 programs compete on various servers on the Internet, and during the Computer Games Olympiad, Go programs form the second-largest group after Chess. Despite the simplicity of the game, and the research effort, computer programs remain weaker than in other board games [14]. The reasons for this are: (i) the complexity of the evaluation function, and (ii) the size of the search space.

1.1 Search and Evaluation

The classical search paradigm for computational games is based on two tools: an evaluation function, and the Minimax back-propagation rule. The aim of these tools is to perform as much pruning as possible, e.g., as in $\alpha - \beta$ search [15] and Proof-Number Search [1]. Using these techniques, strong programs have been made for games like Checkers, Draughts, Othello, and Chess. In Go, all existing evaluation functions are highly biased. Hence, the classical concept of evaluating at the leafs of a tree does not work as well as in other games. Therefore, whereas research on the game of Chess has concentrated on the way of expanding the game tree, in the game of Go the quality of the evaluation function has a more important role. Monte-Carlo evaluation functions are shown to lead to rather precise evaluations and therefore are used in many programs.¹ They consist of playing random games from a position and inferring the value of this position from these games. In this paper we propose and test a move-selection and a backpropagation strategy which both improve the quality of Monte-Carlo evaluations.

¹INDIGO, the Go program by Bruno Bouzy, was ranked 3rd out of 9 in the Computer Olympiad 2005 in Taiwan. VIKING, OLEG, VEGOS, GOLOIS, CRAZYSTONE, GO81, and MANGO are other examples of Monte-Carlo Go programs.

1.2 Structure of the article

In Section 2, we present the structure of Monte-Carlo Go architectures, and algorithms previously used to improve them. In Section 3, we introduce Objective Monte-Carlo. In Section 4, we report on the experiments and their results. Section 5 contains the discussion. Finally, Section 6 concludes and proposes ideas for future research.

2 Related work

In this section, we first present the canonical structure of Monte-Carlo Go algorithms. Then we underline research ideas that have been developed to improve them. Finally we describe the main related algorithms that have previously been applied.

2.1 Monte-Carlo Architectures

In computer games, one possible Evaluation Function (EF) is the Monte-Carlo (MC) method. Given a board position B , its aim is to compute a value $V(B)$ for this position. Starting from the position B , MC plays a certain number of *simulated games*. A simulated game is a succession of moves (called *simulated moves*), played until the end of the game is reached.² The MC evaluation $V(B)$ is then deduced from the results of all the simulated games. In the simplest version, simulated moves are random moves, and $V(B)$ is the average of the outcomes of the simulated games. The outcome that we consider is the difference of territory between the two players. This approach leads to a rather simple program. This program only uses the rules of the game but is nevertheless stronger than human beginners.

2.2 Research on Monte-Carlo Evaluation

The aim of research on MC evaluation is to improve the level of this method by different techniques. We distinguish three kinds of enhancements:

1. Integration of *domain-dependent knowledge* (DDK) in simulated games. Examples of DDK in the game of Go are patterns, tactical goals, or local heuristics [13, 8, 10]. To improve the quality of the simulated games, random simulated games can be replaced by *pseudo-random games* in which moves are chosen partially at random and partially using DDK. A typical implementation gives *DDK-weights* to each move [2]. In the simulated games, the probability of playing a move is the weight of this move divided by the sum of the weights of all possible moves. In earlier research, DDK-weights have been set manually. Lately research has concentrated on learning these weights automatically [5]. Similar techniques are used in Poker [9], Scrabble [16], Bridge [17], etc. This method of enhancement will not be discussed any further in this article, which will focus on the two other methods.
2. One can use the results of previous simulations to choose the move played in subsequent games. The basic idea consists of selecting the move that has shown to lead to the best results more often. This idea was first developed for the game of Go by Brüggemann [7]. He used Simulated Annealing (SA) [11] to build a move ordering. Due to lack of computational power at the time, he made an approximation called “All-move-as-first Heuristic”, which attributed a value to each move independently of the order in which it was played. This idea was taken over by Bruno Bouzy, who applied Progressive Pruning (PP). He showed in [6] that the All-move-as-first Heuristic was not useful anymore with the increase of computational power. In [3], he used PP in a depth- N context with N stages of evaluation and $N - 1$ stages of selection. On the contrary, the algorithm that we propose does not make any distinction between a stage of selection and a stage of evaluation, but a smooth transition from evaluation to selection. Classical ways to use this kind of information are discussed further in this section. Our algorithm is described in detail in subsections 2.3.
3. We show in this article that it is also possible to improve the backpropagation function. This issue will be discussed in Section 3.

²It is possible to develop different approaches. For instance, SLUGGO does not play until the end, but only 16 simulated moves. Then it calls the EF of the open-source program GNUGO. This EF is quite slow. Furthermore, the simulated moves are played using GNUGO, which slows down the whole process. In fact, SLUGGO only acts as a meta program. For these reasons, SLUGGO needs a cluster of PCs to run.

2.3 Move-selection Strategies

The idea behind these strategies is to increase the ability of the Monte-Carlo evaluation to look ahead by exploring the more promising moves first. Before describing the strategies, a statistical background is required.

Each game i that is played has a result R_i . Let R be the random variable which takes the values R_i . The R_i values are bounded, so R has an average value μ and a standard deviation σ . In the case of 9×9 Go, $R_i \in [-81, 81]$ and σ is usually lower than 40.

The Central Limit theorem states that the standard deviation of the random variable $m_n = \frac{\sum_{i=1}^n R_i}{n}$ approaches $\frac{\sigma}{\sqrt{n}}$ when n approaches ∞ . Furthermore, the probability distribution of m_n approaches the normal distribution $N(\mu, \frac{\sigma}{\sqrt{n}})$. Thus, a confidence interval for m_n can be deduced. Let m_∞ be the value of the average that we would expect after an infinite number of stochastic games. There is 66% of confidence that m_∞ is within the interval $[m_n - \frac{\sigma}{\sqrt{n}}, m_n + \frac{\sigma}{\sqrt{n}}]$, 95% of confidence that m_∞ is within the interval $[m_n - 2 \cdot \frac{\sigma}{\sqrt{n}}, m_n + 2 \cdot \frac{\sigma}{\sqrt{n}}]$, etc. Progressive Pruning and Objective Monte-Carlo are based on the results provided by this theorem.

2.3.1 Progressive Pruning (PP)

Progressive Pruning (PP) is a strategy used in the Go program of Bruno Bouzy, INDIGO [4]. It first associates a confidence interval to each move. The lower bound of this interval is $V - \frac{\sigma \cdot A}{\sqrt{n}}$, and its upper bound is $V + \frac{\sigma \cdot A}{\sqrt{n}}$, where V is the current evaluation of this move, n is the number of games simulated from this move, σ is the standard deviation on the value of one simulated game, and A is a parameter. In the beginning, every possible move is played the same number of times. For each move m , when the upper bound of the confidence interval of move m is below the lower bound of the best sibling move, m is pruned. The size of the confidence interval is in proportion to $\frac{\sigma}{\sqrt{n}}$, so when n approaches ∞ , every move is pruned, except the best move. When A is small, the algorithm is fast, because it prunes quickly several moves. However, it might also prune the best move. When A is large, the probability that it prunes the best move decreases strongly, but the algorithm becomes slower.

2.3.2 Simulated Annealing (SA)

This strategy, derived from physics [11], has been used for the first time in the field of Computer Go by Brüggmann [7]. Its main conception is the following. In each step, a random action is carried out. This action changes a certain energy function E . Let ΔE be this change. If $\Delta E < 0$ the action is kept, otherwise it is kept only with the probability $e^{-\frac{\Delta E}{T}}$, where T is the current temperature. The strategy starts with a high temperature, which is then lowered. In the context of computer Go, each action is playing one move. If the energy function is the current evaluation of the move, then SA works as follows:

- A random move is made.
- If it is considered as the current best move, it is kept, and the simulated games continues.
- Otherwise there is a probability $e^{-\frac{\Delta E}{T}}$ to keep it, and a probability of $1 - e^{-\frac{\Delta E}{T}}$ to undo it. That means that the probability to keep the move is high if its current evaluation is near the evaluation of the best move.

The best annealing schedule has to be found experimentally.

3 Objective Monte-Carlo

Objective Monte-Carlo (OMC) is composed of two parts: a move-selection strategy, and a backpropagation strategy. In this section, we describe in detail these two new strategies. They will be tested independently in the next section.

3.1 Move-selection Strategy of OMC

The idea behind the move-selection stage of OMC is to exploit as much as possible the information provided by the Central Limit Theorem. We first give the pseudo code (see Algorithm 1), and then explain its main conception.

```

Data: Previous simulated move
Result: Next simulated move
 $O_{bj} \leftarrow \text{value\_of\_the\_current\_best\_child}$ 
foreach child move  $m \in M$  do
  |  $V_m \leftarrow \text{current\_value\_of\_move\_}m$ 
  |  $\sigma_m \leftarrow \text{standard\_deviation\_of\_}V_m$ 
  |  $U_m(O_{bj}) \leftarrow \text{Erfc}(\frac{O_{bj}-V_m}{\sqrt{2}\cdot\sigma_m})$ 
end
foreach child move  $m \in M$  do
  |  $P_m \leftarrow \frac{U_m(O_{bj})}{\sum_{i \in M} U_i(O_{bj})}$ .
end
Choose the next simulated move randomly according to the probability  $P_m$  for  $m \in M$ .

```

Algorithm 1: Move-selection Strategy of Objective Monte-Carlo.

In this algorithm *Erfc* stands for the complementary error function, i.e.,

$$\text{Erfc}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-u^2} du$$

For each possible score S , the Central Limit Theorem provides an approximation of the probability that the MC evaluation is S . Let $D_m(S)$ be the density function that gives the probability of a move m to have the average value S . $D_m(S)$ has the normal distribution $N(V_m, \frac{\sigma}{\sqrt{n}})$, where V_m is the current evaluation of the move m , and σ is the standard deviation of V_m .

Let O_{bj} be an objective score that the player aims to achieve. Using the distribution $D_m(S) = N(V_m, \frac{\sigma}{\sqrt{n}})$, we can compute the probability of the move m to be superior to O_{bj} . Let us call this probability $U_m(O_{bj})$. We have:

$$U_m(O_{bj}) = \frac{\sum_{x=O_{bj}}^{+\infty} D_m(x)}{\sum_{x=-\infty}^{+\infty} D_m(x)}$$

In Objective Monte-Carlo, $U_m(O_{bj})$ is considered as the *urgency* of a move, its importance. Thus the probability to chose the move is calculated according to these values, as shown in Algorithm 1. As can be seen, we approximate $U_m(O_{bj})$ by the *Erfc* function. This function would be too long to compute in real time, hence the *Erfc* values are stored in a table.

3.2 Backpropagation strategy of OMC

In this section we discuss how to build a backpropagation function better than Minimax for the MC context. This backpropagation strategy gives the value of move m according to the values and standard deviations of its children (see Algorithm 2). In a MC environment, when few simulations have been made, all the values calculated for the children are nearly random. Thus, the Minimax values measured for all children is the max of random numbers. Thus, it gives an overestimated value of the position. To avoid this error, the value returned by the backpropagation strategy should be close to the average value of the children in the beginning of the experiments. When the number of simulations made tend to infinity, on the contrary the value measured are perfectly accurate. So the context is exactly a Minimax context. Thus, when more simulations are made, the value returned should tend to the Minimax value.

In the beginning, all the urgencies U_c are nearly equal. So V_m can be approximated by:

$$V_m = \frac{\sum_{p \in M} V_p \cdot N_p}{\sum_{p \in M} N_p}$$

Therefore, in the begining the value of the move m is the average value of its children.

Moreover, when the number of moves tends to infinity, $U_p(O_{bj})$ tends to $\frac{1}{2}$ if p is the best move, and to 0 else.³

³This can be deduced from Algorithm 1.

```

Data: Values of the children of  $m$ 
Result: Value of  $m$ 
foreach child move  $c$  of  $m \in M$  do
  | Compute the value of  $U_c(Obj)$  according to Algorithm 1.
end
 $N_c \leftarrow$  Number_of_games_played_with_move_c
 $V_m = \frac{\sum_{c \in M} U_c \cdot V_c \cdot N_c}{\sum_{c \in M} U_c \cdot N_c}$ 
return  $V_m$ 

```

Algorithm 2: Backpropagation Strategy of Objective Monte-Carlo.

Thus this backpropagation tends to a Minimax backpropagation when the number of games played tends to infinity.

As a conclusion, this backpropagation strategy makes a soft transition from the average value to the Minimax value. This formula can only be applied in the case of Objective Monte-Carlo, where it is possible to compute the probabilities U_c .

4 Experiments

The aim of the experiments is to evaluate the strategies discussed in the previous section. In the first experiment we evaluate the impact of the move-selection strategy, and in the second experiment the backpropagation strategy. We have randomly selected one hundred positions from games of human players. The level of the selected players is centered on the level of computer programs. This level was chosen to use positions as close as possible to the positions which the program will actually encounter.

4.1 Experimental setup

For each position, we first calculate an accurate evaluation of this position at depth one. To that prospect, we develop a tree at depth one, and then from each leaf launch a classical Monte-Carlo evaluation. This classical program will be called MC-Minimax. The number of stochastic games played is 200,000 per leaf node⁴. The standard deviation of the score for a game of Go is below 40, so, according to the Central Limit theorem, the standard deviation on the MC-Minimax's value is less than 0.1 point. As it is an accurate evaluation of the position, in this paper we call this value the "real value" of the position.

For a given strategy s that played t simulated games, we define the average error of the strategy by:

$$E_s(t) = \frac{\sum_p |V_s^p(t) - V_r^p|}{100}$$

where p is one position, $V_s^p(t)$ is the value of the position p given by the strategy s and V_r^p is the real value of the position.

If we consider an evaluation at depth one, which evaluates every legal move from the root node, less than 20,000 simulated games can be played with tournament time settings.⁵ Therefore this value is used as the maximum number of simulated games in our experiments.

4.2 Results: Move-Selection Strategies

In this subsection, move-selection Strategies are evaluated at depth one. For all these strategies, the value returned is the Minimax value of the child nodes. The results are reported in Figure 1.

Several parameters had to be tuned. For SA it was important to select an adequate annealing schedule. If the annealing is too fast, the strategy does not converge to the real value of the position. If it is too slow then it converges to an accurate limit but very slowly. We tested 16 different annealing schedules and reported the best one (balancing speed and accuracy of

⁴The average branching factor of the 100 games of the testbed is around 50. The number of simulated moves played per game is around 120. Hence the total number of simulated moves played to build this testbed is $200,000 \times 50 \times 100 \times 120 = 120\text{billions}$. This required a couple of days on an Opteron 2.8GHz.

⁵This implies that, for a program able to look ahead, the number of games available is even lower.

convergence). For PP, a size for the interval of confidence has to be chosen. We used the one which is used in Bruno Bouzy’s program, INDIGO, because it has been carefully tuned. If this confidence interval is too small, it can happen that good moves are pruned, and the strategy does not converge to 0. If the confidence interval is too big, the convergence will be slower.

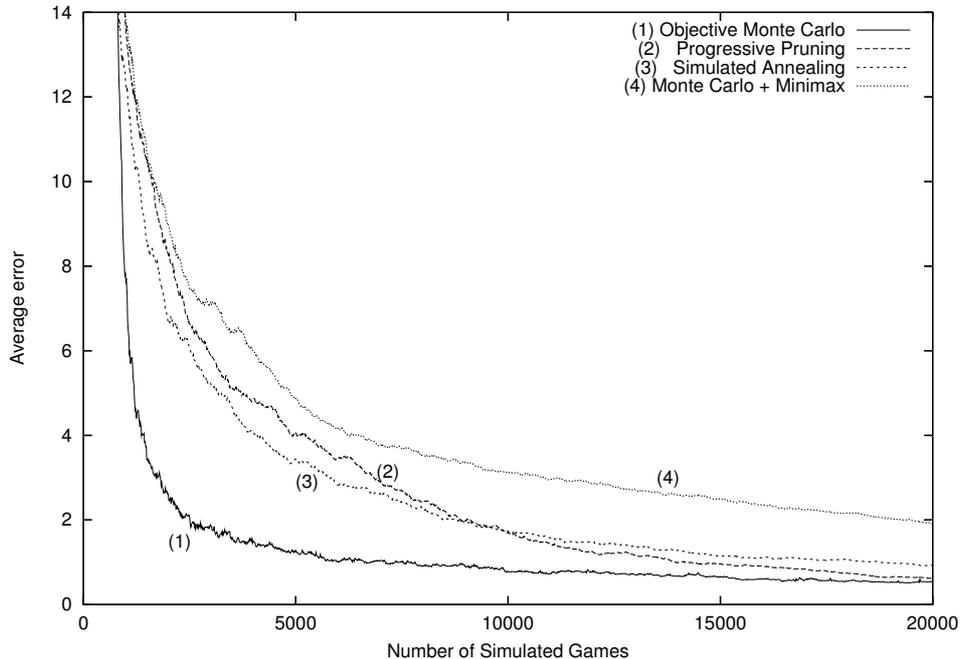


Figure 1: Convergence of Simulated Annealing, Progressive Pruning and Objective Monte-Carlo.

The OMC move-selection strategy converges faster than the other strategies. With the parameters used, PP converges slower but to an accurate limit. SA (resp. PP) could converge faster with a faster annealing schedule (respectively a smaller confidence interval). However then they would converge to a worse limit because in some situations the best move would be pruned. OMC, on the contrary, never prunes any move.

4.3 Results: Backpropagation Strategies

We compare OMC backpropagation with Minimax backpropagation. The results are shown in Figure 2. The first thing to note is that OMC backpropagation improves considerably on the results when few simulations have been made. Furthermore, OMC Backpropagation is more accurate than Minimax backpropagation whatever the number of simulated games performed.

5 Discussion

In this section, we first point out the characteristics of Monte-Carlo methods compared to classical methods. Then we focus on the specificity of our algorithm. Finally we discuss how to apply our strategy to other problems.

Monte-Carlo methods have the drawback to be slow. Indeed, the number of simulated games to be performed is important. Thus, Monte-Carlo programs are only able of performing shallow tree searches [3]. However, they are known to be particularly robust [6], so their lack of ability to look ahead is balanced by their ability to avoid global blunders. In this context, increasing the speed of convergence and building a new suitable tree search constitute two milestones.

To that prospect, OMC brings several improvements. Mostly it converges faster than usual strategies, and does not require any parameter to be tuned. In addition it presents two other advantages. First, this strategy can be applied in every node of the game tree. As it needs to store information about the values of the child nodes, the number of nodes in which it can be applied is proportional to the available memory. In our computer, we could apply it on 500,000 nodes, which corresponds to an average depth of three. This strategy is more efficient when

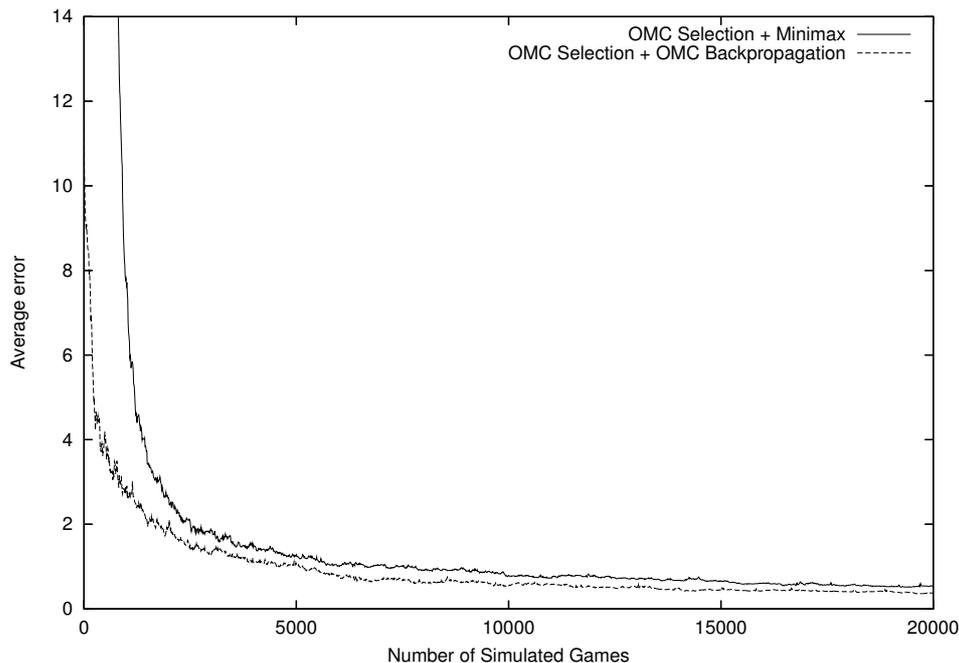


Figure 2: Convergence of OMC with Minimax backpropagation and with OMC backpropagation.

it is applied near the root node since the gain is related to the number of simulated games performed. We have shown the significant addition of our strategy at depth one. Our next research question will be to assess it at depth N for $N \geq 2$.

Another advantage of the OMC strategy is that it can adapt its style of play depending on who is winning. This can be done by adjusting O_{bj} slightly. If our program’s EF evaluates that the program is winning, then it is not important to evaluate how much it is ahead. The crucial point is to avoid mistakes. The opponent must make a very good move to come back. Thus, the Monte-Carlo strategy should explore the “dangerous” moves more often. This can be achieved by increasing the objective O_{bj} . Thus, changing O_{bj} relative to the best successor node can make the program play in a more secure way, or in a more competitive way. This introduces a bias in the evaluation. However, this kind of bias can be useful, as the program would control it. For instance, if the program is winning it is better to have an overestimated evaluation of the moves of the opponent than an unbiased evaluation, because it induces a more secure style of play. Another research issue will be to find an algorithm that adjusts this bias automatically.

The framework we developed in this article can be applied to different kinds of adversarial problems, since it only uses simulations and statistics from these simulations. However, when it is possible to build an accurate evaluation function, as in most of board games, classical methods still perform better than Monte-Carlo ones. We believe that Real Time Strategy (RTS) games are in some cases a good candidate for Monte-Carlo methods, since it can be hard to build an accurate evaluation function. An application of Monte-Carlo methods in RTS can be found in [12].

6 Conclusion and future research

In this paper we introduced the Objective Monte-Carlo move-selection strategy, which outperforms the classical strategies, without any need to tune any parameter like an annealing schedule or a confidence interval. Indeed, the only parameter of this strategy, O_{bj} , is adjusted automatically. Then we proposed a backpropagation strategy better than Minimax for the OMC context. This algorithm has been implemented in our Go program, MANGO, and its level will be assessed in future computer Go tournaments.

The use of this selection strategy and the backpropagation strategy leads to a tree search that does not require any evaluation function. Hence, this can be applied to any game where it is difficult to create an evaluation function without parameter tuning. Assessing the ability of

Objective Monte-Carlo to look ahead is our first topic for future research. Our next research topic is to apply this strategy to other games than Go. Another research issue is to adjust automatically the value of the objective to take into account the fact that the program is winning or losing.

7 Acknowledgments

The authors would like to thank Steven de Jong and the anonymous referees for their adequate advice. This work is financed by the Dutch Organization for Scientific Research (NWO) in the framework of the project Go for Go, grant number 612.066.409.

References

- [1] L. Victor Allis, Maarten van der Meulen, and H. Jaap van den Herik. Proof-number search. *Artificial Intelligence*, 66:91–124, 1994.
- [2] Bruno Bouzy. Associating domain-dependent knowledge and Monte Carlo approaches within a Go program. *Joint Conference on Information Sciences*, 2003.
- [3] Bruno Bouzy. Associating shallow and selective global tree search with Monte Carlo for 9x9 Go. In K. Chen, editor, *Proceedings of 4th Computer and Games Conference, Ramat-Gan*, 2004.
- [4] Bruno Bouzy. Move pruning techniques for Monte-Carlo Go. In *Proceedings of the 11th Advances in Computer Game conference*, page 15, 2005.
- [5] Bruno Bouzy and Guillaume Chaslot. Monte-Carlo Go Reinforcement Learning Experiments. In G. Kendall and S. Louis, editors, *IEEE 2006 Symposium on Computational Intelligence in Games, Reno, USA*, 2006.
- [6] Bruno Bouzy and Bernard Helmstetter. Monte Carlo Go Developments. In H. Jaap van den Herik, Hiroyuki Iida, and Ernst A. Heinz, editors, *Proceedings of the Advances in Computer Games Conference (ACG-10)*, pages 159–174. Kluwer Academic, 2003.
- [7] Bernd Brügmann. Monte Carlo Go. *White paper*, 1993.
- [8] Tristan Cazenave. Automatic acquisition of tactical Go rules. *3rd Game Programming Workshop in Japan, Hakone*, 1996.
- [9] Jonathan Schaeffer Duane Szafron Darse Billings, Aaron Davidson. The challenge of poker. *Artificial Intelligence*, 134(1):201–240, 2002.
- [10] Shun-Chin Hsu Jeng-Chi Yan and H.Jaap van den Herik. The sacrifice move. *ICGA Journal*, 28(4):223–234, 2005.
- [11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, pages 671–680, 1983.
- [12] Michael Buro Michael Chung and Jonathan Schaeffer. Monte Carlo Planning in RTS games. In Simon Lucas and Graham Kendall, editors, *Proceeding of the IEEE Symposium on Computational Intelligence and Games*, 2005.
- [13] Martin Müller. Pattern matching in Explorer: Extended abstract. In *Proceedings of the Game Playing System Workshop*, pages 1–3. ICOT, 1991.
- [14] Martin Müller. Not like other games – why tree search in Go is different. In *Fifth Joint Conference on Information Sciences (JCIS)*, 2000.
- [15] Judea Pearl. The solution for the branching factor of the alpha–beta pruning algorithm and its optimality. *Communications of the ACM*, 25:559–564, 1982.
- [16] Brian Sheppard. World-championship-caliber scrabble. *Artificial Intelligence*, 134(1):241–275, 2002.
- [17] Stephen Smith, Dana Nau, and Tom Throop. Computer Bridge - A Big Win for AI planning. *AI Magazine*, 19(2):93–105, 1998.

Towards a Framework for Goal Revision

Célia da Costa Pereira

Andrea Tettamanzi

Università degli Studi di Milano
Dipartimento di Tecnologie dell'Informazione
Via Bramante 65, I-26013 Crema (CR), Italy
pereira@dti.unimi.it, andrea.tettamanzi@unimi.it

Abstract

A rational agent revises its goals if something changes in its mental state. In this paper, we propose (i) a general framework based on classical propositional logic, to represent changes in the mental state of the agent after the acquisition of new information and/or after the arising of new desires; (ii) fundamental postulates that the function which generates the goal set must obey; and (iii) properties that this function must have to guarantee both the agent's maximal satisfaction and the consistency of the goal set.

1 Introduction

Although there has been much discussion on belief revision, goal revision has not received much attention. To the best of our knowledge, the few works on goal *change* found in the literature do not build on results on belief revision. That is the case of [4], in which the authors propose a formal representation for goals as rational desires and introduce and formalize dynamic goal hierarchies, but do not formalize explicitly beliefs and plans; or of [8], in which the authors propose an explicit representation of goals suited for conflict resolution based on a preference ordering of sets of goals. A more recent approach is [6], which models a multi-agent system in which an agent adopts a goal if requested to do so and the new goal is not conflicting with existing goals. This approach is based on goal persistence, i.e., an agent maintains its goals unless explicitly requested to drop them by the originating agent. The main lack of these approaches is that agents do not use their own knowledge for revising goals.

A static approach to the problem of how goals arise has been proposed within planning is over-subscription planning [7, 5]. In over-subscription planning, the problem of identifying the best subset of goals, given resource constraints, is addressed. Our work is very much in that line, except that it attempts to provide a model of rationality, which is a slightly different focus.

We propose an approach for dynamically constructing the goal set to be pursued by a rational agent, by considering changes in its mental state. The formalism is kept very simple because it is intended as a first step. In particular, we use an unsophisticated definition of planning rules. A more sophisticated — and realistic — formalism for expressing plans and planning rules could be used; however, planning *per se* is not the focus of this work and we decided to keep things simple for the sake of clarity.

2 Example

At the beginning of 2006, Nick has \$93,000. Nick wants to invest that amount and have a return of at least 40% at the end of the year with a probability of at least $\frac{2}{3}$ ($d_{40\%}$). Nick also wants to buy a house (d_H) at the beginning of 2007, and he knows he will need \$100,000 with at least 90% probability ($d_{\$100,000}$) for it. Nick believes that X, a stock currently trading at \$56, will reach \$80 (b_{80}), and *a fortiori* also \$60.24 (b_{60}), by the end of the year. Nick might face two alternative possibilities for investing in X: buy 1660 shares at \$56 (d_X), then

- A wait until X reaches \$80 to sell them (d_{80}), realizing a return of more than 40% with probability 68%, thus fulfilling $d_{40\%}$;
- B wait until X reaches \$60.24 to sell them (d_{60}), realizing a wealth of \$100,000 with probability 95%, thus fulfilling d_H but not $d_{40\%}$.

Failing to sell at the desired level, Nick will sell all of his X stock at market price at the end of the year. Under possibility A, this will yield enough money to buy a new house with a probability of 12%, which gives a total probability of 80% for the ability to buy a new house, not enough to fulfill d_H .

3 Preliminaries

We present the logical language, inspired by [3], which will be used throughout this paper, as well as the four bases representing the mental state of an agent.

Desires are necessary, not sufficient, conditions for action. When a desire is met by other conditions that make it possible for an agent to act, that desire becomes a *goal*. Therefore, given this technical definition of a desire, all goals are desires, but not all desires are goals.

We distinguish two sets of literals: the set \mathcal{D} of all possible desires of agents and the set \mathcal{K} of all possible knowledge items. For the sake of simplicity we make the assumption that desires and knowledge items are on completely different levels: a desire is not a knowledge and vice versa. An interpretation \mathcal{I} is an assignment of truth values to all propositions of \mathcal{D} and \mathcal{K} .

Definition 1 (Desire-generating Rule) *A desire-generating rule is an expression of the form $b_1 \wedge \dots \wedge b_n \wedge d_1 \wedge \dots \wedge d_m \rightsquigarrow d$, where $b_i \in \mathcal{K}$ and $d_j, d \in \mathcal{D}$, with $d \neq d_j$ for all j .*

The meaning of the rule is “if the agent believes b_1, \dots, b_n and desires d_1, \dots, d_m , then the agent must desire d as well”.

Definition 2 (Planning Rule) *A planning rule is an expression of the form $d_1 \wedge \dots \wedge d_n \rightarrow d$, where $d_i, d \in \mathcal{D}$, $d \neq d_i$ for all i .*

The rule states that if d_1, \dots, d_n are fulfilled then d is fulfilled as well ¹.

Given a desire-generating or a planning rule R , we shall denote $\text{lhs}(R)$ the set of literals that make up the conjunction on the left-hand side of R , and $\text{rhs}(R)$ the literal on the right-hand side of R . Furthermore, if S is a set of rules, we define $\text{rhs}(S) = \{\text{rhs}(R) : R \in S\}$.

Definition 3 (Agent’s bases) *An agent is equipped with four components:*

- *belief base:* $\mathcal{B} \subseteq \mathcal{K}^2$;
- *desire set:* $\mathcal{J} \subseteq \mathcal{D}$;
- *desire-generating rule base:* $\mathcal{R}_J = \{R : R = b_1 \wedge \dots \wedge b_n \wedge d_1 \wedge \dots \wedge d_m \rightsquigarrow d\}$ with $b_j \in \mathcal{K}$ and $d, d_k \in \mathcal{D}$;
- *planning rules base:* $\mathcal{P} = \{P : P = d_1 \wedge \dots \wedge d_n \rightarrow d\}$ with $d, d_j \in \mathcal{D}$;

The state of an agent is completely described by a 4-tuple $\mathcal{S} = \langle \mathcal{B}, \mathcal{R}_J, \mathcal{J}, \mathcal{P} \rangle$. The set \mathcal{B} represents the agent’s knowledge about the world, \mathcal{R}_J contains the rules which generate desires from beliefs and other (more basic) desires, \mathcal{J} contains all desires which may be deduced from the agent’s knowledge and the agents’s desire-generating rule base, and \mathcal{P} contains all available planning rules for achieving agent desires. For the sake of simplicity, we assume here that the planning rules are given and fixed.

In the example, Nick’s state at the beginning of the year, \mathcal{S}_0 , consists of $\mathcal{B} = \{b_{80}, b_{60}\}$, $\mathcal{R}_J = \{\rightsquigarrow d_H, \rightsquigarrow d_{40\%}, b_{80} \rightsquigarrow d_{80}, b_{60} \wedge \neg b_{80} \rightsquigarrow d_{60}\}$, $\mathcal{J} = \{d_H, d_{40\%}, d_{80}\}$, and $\mathcal{P} = \{d_{\$100,000} \rightarrow d_H, d_X \wedge d_{80} \rightarrow d_{40\%}, d_X \wedge d_{60} \rightarrow d_{\$100,000}, \neg d_{80} \rightarrow d_{60}, \neg d_{60} \rightarrow d_{80}\}$.

Definition 4 (Active Desire-generating Rule) *A desire-generating rule R is active in \mathcal{S} iff $\mathcal{S} \models \text{lhs}(R)$ ³.*

¹Note that the implications defined in desire-generating rules and planning rules are not material. So for example, from $\neg b$ and $a \rightarrow b$ or $a \rightsquigarrow b$, we can not deduce $\neg a$.

²Unlike \mathcal{J} , \mathcal{B} must be consistent.

³ $\mathcal{S} \models \text{lhs}(R)$ iff $\forall \mathcal{I}, \mathcal{I} \models \mathcal{S} \Rightarrow \mathcal{I} \models \text{lhs}(R)$.

The next three definitions introduce a notation to refer to the set of rules that become active, resp. inactive, in a given state \mathcal{S} , if a new belief or a new desire are introduced in the respective bases. The activation of a desire-generating rule brings about changes in the desire base of an agent. These changes, in turn, may cause the activation/deactivation of other rules.

Definition 5 (Rule activated by a Belief/Desire) *Let z be a new belief or desire in state \mathcal{S} . We define a sequence $\{A_i\}_{i=0,\dots}$ of subsets of $\mathcal{R}_{\mathcal{J}}$ as follows:*

$$\begin{aligned} A_0 &= \{R : z \in \text{lhs}(R) \wedge \forall x \in \text{lhs}(R), x \neq z \Rightarrow \mathcal{S} \models x\}, \\ A_i &= \{R : \text{lhs}(R) \cap \text{rhs}(A_{i-1}) \neq \emptyset \wedge \forall x \in \text{lhs}(R), x \notin \text{rhs}(A_{i-1}) \Rightarrow \mathcal{S} \models x\}. \end{aligned}$$

$\text{Act}_z^{\mathcal{S}} \equiv \bigcup_{i=0}^{\infty} A_i$ is the set of all desire-generating rules activated by belief or desire z in \mathcal{S} . A_0 is the set of the desire-generating rules directly activated by z , i.e. with z in their left hand side. A_i is the set of desire-generating rules indirectly activated by z by way of rules in A_{i-1} .

Definition 6 (Rule deactivated by a Belief/Desire) *Let z be a new belief or desire in the state \mathcal{S} . We define a sequence $\{D_i\}_{i=0,\dots}$ of subsets of $\mathcal{R}_{\mathcal{J}}$ as follows:*

$$\begin{aligned} D_0 &= \{R : \mathcal{S} \models \text{lhs}(R) \wedge \neg z \in \text{lhs}(R)\}, \\ D_i &= \{R : \mathcal{S} \models \text{lhs}(R) \wedge \text{lhs}(R) \cap \text{rhs}(D_{i-1}) \neq \emptyset\}. \end{aligned}$$

$\text{Deact}_z^{\mathcal{S}} \equiv \bigcup_{i=0}^{\infty} D_i$ is the set of all desire-generating rules deactivated by belief or desire z in \mathcal{S} . D_0 is the set of desire-generating rules directly deactivated by z , i.e. with $\neg z$ in their left hand side. D_i is the set of desire-generating rules indirectly deactivated by z by way of rules in D_{i-1} .

In all four cases, when speaking of rule activation (resp. deactivation), we will say a rule R is *downstream* of another rule R' if $R' \in A_0$ (resp. D_0) and $R \in A_i$ (resp. D_i), with $i > 0$.

Finally, we do not expect a rational agent to formulate desires out of whim, but based on some rational argument. To model that state of affairs, desire-generating rules play the role of rational arguments and we define a desire to be justified as follows.

Definition 7 (Justified Desire) *A desire d is justified in state \mathcal{S} iff d is in \mathcal{J} , i.e. there is an active desire-generating rule R in $\mathcal{R}_{\mathcal{J}}$ such that $\text{rhs}(R) = d$.*

4 Changes in the State of an Agent

The acquisition of a new belief in state \mathcal{S} , may cause a change in the belief base \mathcal{B} and this may also cause a change in the desire set \mathcal{J} with the retraction of existing desires and/or the addition of new desires. A desire d is retracted from the desire set \mathcal{J} if and only if d becomes not justified, i.e., all active desire-generating rule such that $\text{rhs}(R) = d$ become inactive. A desire d is added to a desire set \mathcal{J} if and only if the new information activates a desire-generating rule R with $\text{rhs}(R) = d$.

4.1 Changes caused by a New Belief

Let \mathcal{S} be the state of an agent, b a new piece of information, $*$ the well known AGM operator for belief revision [1], \oplus our operator for desire updating, and \mathcal{J} the base of agent's desires. Two considerations must be taken into account:

1. By definition of the revision operator $*$, $b \in \mathcal{B} * b$, thus all desire-generating rules $R \in \text{Act}_b^{\mathcal{S}}$ become active and all new desires $d = \text{rhs}(R)$ are added to the desire set \mathcal{J} .
2. If, before the arrival of b , $\neg b \in \mathcal{B}$ then all active desire-generating rules R , such that $\neg b \in \text{lhs}(R)$, become inactive and, if there is not an active desire-generating rule R' , such that $\text{rhs}(R') = \text{rhs}(R)$, then the desire $d = \text{rhs}(R)$ is retracted from the desire set \mathcal{J} .

We can summarize the above considerations into one desire-updating formula which tells how the desire set \mathcal{J} of a rational agent in state \mathcal{S} should change in response to the acquisition of a new belief b . Let $A_b^{\mathcal{S}}$ be the set of desires acquired because of the new belief b :

$$A_b^{\mathcal{S}} = \text{rhs}(\text{Act}_b^{\mathcal{S}}). \tag{1}$$

Let $L_b^{\mathcal{S}}$ be the set of desires lost because of the acquisition of the new belief b :

$$L_b^{\mathcal{S}} = \{d \mid d \in \text{rhs}(\text{Deact}_b^{\mathcal{S}}) \wedge \neg \exists R (\mathcal{S} \models \text{lhs}(R) \wedge R \notin \text{Deact}_b^{\mathcal{S}} \wedge \text{rhs}(R) = d)\}. \quad (2)$$

According to the above considerations, we have:

$$\mathcal{J} \oplus b = (\mathcal{J} \cup A_b^{\mathcal{S}}) \setminus L_b^{\mathcal{S}}. \quad (3)$$

It is easy to verify that $A_b^{\mathcal{S}} \cap L_b^{\mathcal{S}} = \emptyset$, for all state \mathcal{S} .

Proposition 1 *The order in which two different pieces of information b_i and b_j are acquired does not affect the content of the final set of desires:*

$$(\mathcal{J} \oplus b_i) \oplus b_j = (\mathcal{J} \oplus b_j) \oplus b_i. \quad (4)$$

The proof is based on the fact that (i) after acquiring b_i , all inactive desire-generating rules with $b_j \wedge \neg b_i$ in their left hand side, as well as their downstream rules will not be activated; (ii) $A_{b_k}^{\mathcal{S}} \cap L_{b_k}^{\mathcal{S}} = \emptyset$ for all k ; and (iii) rules deactivated with the acquisition of b_j have either $\neg b_j$ or $b_i \wedge \neg b_j$ in their left hand side, or are downstream of such rules.

Going back to the example, if Nick learns that earnings of X dropped 10% with respect to his estimates, he has to revise his belief that X will reach \$80. Therefore, his belief base and desire set become, respectively, $\mathcal{B} = \{-b_{80}, b_{60}\}$ and $\mathcal{J} = \{d_H, d_{40\%}, d_{60}\}$.

4.2 Changes caused by a New Desire

In this work, for the sake of simplicity, we consider that a new desire arises when a desire-generating rule with an empty left hand side is inserted into \mathcal{R}_J .

Let \otimes be the operator for updating a desire-generating rule base, \mathcal{S} be the state of the agent whose desire base is \mathcal{J} , and $A_d^{\mathcal{S}} = \{\text{rhs}(R) \mid R \in \text{Act}_d^{\mathcal{S}}\}$ be the set of desires acquired with the arising of d in the state \mathcal{S} . How does \mathcal{S} change with the arising of the new desire d ?

1. the desire-generating rule $\mapsto d$ is added to \mathcal{R}_J ,
2. d is added to \mathcal{J} ,
3. all desire-generating rules R in $\text{Act}_d^{\mathcal{S}}$ become activated, and all desires appearing in the right hand side of these rules are also added to \mathcal{J} .

Therefore,

$$\mathcal{J} \oplus d = \mathcal{J} \cup \{d\} \cup \{\text{rhs}(R), \forall R \in \text{Act}_d^{\mathcal{S}}\}. \quad (5)$$

$$\mathcal{R}_J \otimes d = \mathcal{R}_J \cup \{\mapsto d\}. \quad (6)$$

If, before learning about the earnings drop, Nick decides he also wants to buy a new car (d_C), then $\mathcal{R}_J = \{\mapsto d_H, \mapsto d_{40\%}, b_{80} \mapsto d_{80}, b_{60} \wedge \neg b_{80} \mapsto d_{60}, \mapsto d_C\}$, $\mathcal{J} = \{d_H, d_{40\%}, d_{80}, d_C\}$. \mathcal{B} does not change.

Proposition 2 *The order in which two different desires d_1 and d_2 arise does not affect the final state \mathcal{S} of an agent:*

$$\mathcal{J} \oplus d_1 \oplus d_2 = \mathcal{J} \oplus d_2 \oplus d_1. \quad (7)$$

$$\mathcal{R}_J \otimes d_1 \otimes d_2 = \mathcal{R}_J \otimes d_2 \otimes d_1. \quad (8)$$

The proof of these propositions is trivial.

Finally, we are interested in characterizing how the state \mathcal{S} of an agent evolves when acquiring new beliefs after the arising of new desires and vice-versa. In particular, we have to check that the order of acquisition of new beliefs and desires does not matter in determining the agent's final state:

Proposition 3 *Let \mathcal{J} be a set of desires, let b be a new piece of information and let d be a new desire for an agent. The new set of desires obtaining from the old set of desires \mathcal{J} by considering b and d does not depend of the order in which these two news components are known by the agent:*

$$\mathcal{J} \oplus b \oplus d = \mathcal{J} \oplus d \oplus b. \quad (9)$$

$$\mathcal{R}_{\mathcal{J}} \otimes b \otimes d = \mathcal{R}_{\mathcal{J}} \otimes d \otimes b. \quad (10)$$

The proof of Equation 9 in this proposition is based on an argument similar to the one used for proving Proposition 1; the proof of the other part is trivial.

5 Comparing Desires and Sets of Desires

An agent may have many desires. However, it is essential to be able to represent the fact that not all desires have the same importance or urgency for a rational agent. A natural choice for representing the importance of desires would be to adopt the same concept of utility.

A utility function for desires is a function $u : \mathcal{D} \rightarrow \mathbb{R}$ which associates a real value, utility, to all desires. An important assumption we have to make is that if $d \in \mathcal{J}$, i.e., d is desired by a rational agent, it must be $u(d) > 0$. In other words, a rational agent cannot desire to incur a cost and cannot waste time having desires which, if realized, would not bring any benefit.

In the example, we may assume the utilities Nick attaches to his desires are: $u(d_H) = 100$, $u(d_{40\%}) = 40$, $u(d_{80}) = 4$, $u(d_{60}) = 3$.

One problem with utilities is that, in general, we are not able to attach a precise numerical value to desires. However, in some contexts, “desires”, i.e., given sets of states of the world, might lend themselves naturally to an economic valuation. For instance, if the agent is an investor, its plans are the purchase or sale of financial instruments, and its desires are levels of wealth or return on investment, attaching a dollar value to desires would be the most natural way of representing their importance. An alternative approach would be to establish a (partial or total) ordering among desires. In either case, we can define preference between desires as follows.

Definition 8 (Preference between Desires) *A desire d is preferred to a desire d' , denoted $d \succeq d'$ iff the agent desires d at least as much as it desires d' .*

Of course, if utilities are defined, $d \succeq d'$ iff $u(d) \geq u(d')$. Therefore, in the example $d_H \succeq d_{40\%} \succeq d_{80} \succeq d_{60}$.

The \succeq relation, which is reflexive and transitive, can be extended from desires to sets of desires.

Definition 9 (Preference between Sets of Desires) *A set of desires D_1 is preferred to a set of desires D_2 , denoted $D_1 \succeq D_2$:*

- if utilities are defined, iff

$$\sum_{d \in D_1} u(d) \geq \sum_{d \in D_2} u(d);$$

- otherwise, iff one of the following three conditions is satisfied:

1. $D_2 \subseteq D_1$;
2. $\forall d \in D_2, \exists d' \in D_1, \text{ s.t. } d' \succeq d$;
3. neither 1 nor 2 are satisfied, and $\forall d \in D_1, \exists d' \in D_2, \text{ s.t. } d \succeq d'$.

For example, $\{d_H, d_{60}\} \succeq \{d_H\} \succeq \{d_{40\%}, d_{80}\} \succeq \{d_{40\%}, d_{60}\}$.

6 Revising Goal Sets

The main point about desires is that we expect a rational agent to try and manipulate its surrounding environment to fulfill them. In general, not all desires can be fulfilled at the same time, especially when they are conflicting [2]. Therefore, a rational agent will select a justified, coherent and feasible set of desires to realize.

Definition 10 (Coherence of a Desire) A desire d is coherent, w.r.t. \mathcal{S} , with a set of desires \mathcal{J} iff $\exists \mathcal{I} \models \mathcal{S}$, s.t. $\forall w \in \mathcal{J}, \mathcal{I} \models d \wedge w$.

Besides being coherent with other desires, a desire must be feasible. Nick might desire to get a 100% return with 99% probability from his investment: while that desire is coherent with Nick's other desires, unfortunately for him, it is not feasible. Given that some desires are not feasible, a rational agent should concentrate on feasible desires. To define what "feasible" means, we need a number of other definitions.

Definition 11 (Partial plan) A partial plan for achieving desire d in the state \mathcal{S} is a pair $\langle H, d \rangle$ such that: (i) $d \in \mathcal{J}$ and (ii) $H = \{d_1, \dots, d_n\}$ if there exists a planning rule $P = d_1 \wedge \dots \wedge d_n \rightarrow d$.

Note that a desire may have several partial plans.

The complete way to fulfill a given desire d is called in [2] a *complete plan*. A complete plan for a given desire d is an *AND* tree. Its nodes are partial plans and its edges represent the relationship between desires necessary for the justification of d . It is an *AND* tree because all these desires must be considered.

Definition 12 (Complete Plan) A complete plan for achieving a desire d in the state \mathcal{S} is a pair $\langle C, d \rangle$ with $d \in \mathcal{J}$ and C is a finite tree such that: (i) The root of the tree is a partial plan $\langle H, d \rangle$; (ii) A node $\langle \{d_1, \dots, d_n\}, d' \rangle$ has exactly n children $\langle H_1, d_1 \rangle, \dots, \langle H_n, d_n \rangle$, where $\langle H_i, d_i \rangle$ is a partial plan for d_i ; and (iii) The leaves of the tree are partial plans $\langle H_i, d_i \rangle$ with $H_i = \emptyset$. The function $\text{Des}(C)$ returns the set of all desires in C .

If Nick's mental state is \mathcal{S}_0 , then the unique complete plan for desire d_H has root $\langle \{d_{\$100,000}\}, d_H \rangle$ which has one child $\langle \{d_X, d_{60}\}, d_{\$100,000} \rangle$, which in turn has two children $\langle \emptyset, d_X \rangle$ and $\langle \{-d_{80}\}, d_{60} \rangle$. The latter has one child $\langle \emptyset, -d_{80} \rangle$.

The unique complete plan for desire $d_{40\%}$ has root $\langle \{d_X, d_{80}\}, d_{40\%} \rangle$ which in turn has two children $\langle \emptyset, d_X \rangle$ and $\langle \{-d_{60}\}, d_{80} \rangle$. The latter has one child $\langle \emptyset, -d_{60} \rangle$.

Definition 13 (Feasible Desire) A desire d is feasible if there exists a complete plan $\langle C, d \rangle$ to achieve it.

Definition 14 (Conflict between complete plans) Two complete plans $\langle C, d \rangle$ and $\langle C', d' \rangle$ are conflicting in a state \mathcal{S} , denoted as $\langle C, d \rangle \bowtie_{\mathcal{S}} \langle C', d' \rangle$, iff $\text{Des}(C) \cup \text{Des}(C') \cup \mathcal{B} \vdash \perp$.

Definition 15 (Feasible set of Desires) A set of feasible desires $\{d_1, \dots, d_n\}$ is said to be feasible in state \mathcal{S} iff $\forall \langle C_1, d_1 \rangle, \dots, \langle C_n, d_n \rangle, \text{Des}(C_1) \cup \dots \cup \text{Des}(C_n) \cup \mathcal{B} \vdash \top$.

It is clear that the set of Nick's desires in state \mathcal{S}_0 , $\{d_H, d_{40\%}, d_{80}\}$ is not feasible.

Definition 16 (Consistency of a desire with a set) A feasible desire d is consistent, w.r.t \mathcal{S} , with a set \mathcal{J} of desires, $\text{cons}(d|\mathcal{J})$, iff d is coherent with \mathcal{J} , justified and $\{d\} \cup \mathcal{J}$ is a feasible set.

The set of the goals a rational agent is pursuing must be consistent.

Definition 17 (Goal Set) A goal set in the state \mathcal{S} is a set of desires $\mathcal{G} \subseteq \mathcal{J}$ such that, $\forall d \in \mathcal{G}, \text{cons}(d|\mathcal{G})$.

6.1 Postulates for Goal Revision

In general, given a set of desires \mathcal{J} , there are many possible goal sets $\mathcal{G} \subseteq \mathcal{J}$. However, a rational agent in state $\mathcal{S} = \langle \mathcal{B}, \mathcal{J}, \mathcal{R}_J, \mathcal{P} \rangle$ will elect as the set of goals it is pursuing one precise goal set \mathcal{G}^* , which depends on \mathcal{S} .

Let us call G the function which maps a state \mathcal{S} into the goal set elected by a rational agent in state \mathcal{S} : $\mathcal{G}^* = G(\mathcal{S})$. This goal election function G must obey two fundamental postulates:

- **(G ⊙ 1)** $\forall \mathcal{S}, G(\mathcal{S})$ is a goal set;
- **(G ⊙ 2)** $\forall \mathcal{S}$, if $\mathcal{G} \subseteq \mathcal{J}$ is a goal set, then $G(\mathcal{S}) \succeq \mathcal{G}$, i.e., a rational agent always selects the most preferable goal set.

Let d be a new desire arising in state \mathcal{S} , let $\mathcal{S} = \langle \mathcal{B}, \mathcal{J}, \mathcal{R}_J, \mathcal{P} \rangle$ be the state of the agent before desire d , let $\mathcal{S}_d = \langle \mathcal{B}, \mathcal{J} \oplus d, \mathcal{R}_J \otimes d, \mathcal{P} \rangle$ be the state resulting from the arising of the new desire d in the state \mathcal{S} . The function G must respect the following properties:

- **(P1)** If $\text{cons}(d|G(\mathcal{S}))$ then $d \in G(\mathcal{S}_d)$;
- **(P2)** If $\neg\text{cons}(d|G(\mathcal{S}))$, and there is a goal g in $G(\mathcal{S})$ whose complete plans are all conflicting with all possible complete plans for d and $g \succeq d$ then $G(\mathcal{S}_d) = G(\mathcal{S})$. Thus, in case it is impossible to maintain both the new desire and the old desire, a rational agent maintains its old desire.
- **(P3)** If $\neg\text{cons}(d|G(\mathcal{S}))$ and for all goals g in $G(\mathcal{S})$ whose complete plans are all conflicting with all possible complete plans for d we have $d \succeq g$ then $d \in G(\mathcal{S}_d)$ and $g \notin G(\mathcal{S}_d)$.

Summarizing, a new desire d may become a goal in state \mathcal{S}_d in two cases:

- d is consistent with $G(\mathcal{S})$.
- d is not consistent with $G(\mathcal{S})$, but it is preferred to all goals whose complete plans are all conflicting with all its complete plans.

Let \odot be the goal revision operator. We have $G(\mathcal{S}) \odot d = G(\mathcal{S}_d)$.

6.2 Defining the Goal Set Election Function

We now propose three definitions of G , namely G_u , G_{\succeq} , and G_{\subseteq} which are applicable, respectively, to the case whereby utilities are defined, to the weaker case in which the total ordering \succeq of desires is available, and to the weakest case in which only a partial ordering or no ordering at all of desires is available.

Function G_u is computed by solving the following combinatorial optimization problem: given state \mathcal{S} ,

$$\begin{aligned} & \text{maximize} && u(\mathcal{G}) = \sum_{g \in \mathcal{G}} u(g), \\ & \text{subject to} && \mathcal{G} \subseteq \mathcal{J}, \\ & && \mathcal{G} \text{ is a goal set.} \end{aligned} \tag{11}$$

Going back to the example, $G_u(\mathcal{S}_0) = \{d_H, d_{80}\}$.

Function G_{\succeq} can be computed in $O(\|\mathcal{J}\|^2)$ time by means of the following algorithm. Given \mathcal{S} , consider any ordering $d_1, d_2, \dots, d_{\|\mathcal{J}\|}$ of the elements of \mathcal{J} .

1. $\mathcal{G} \leftarrow \emptyset$;
2. $i \leftarrow 0$;
3. if $i = \|\mathcal{J}\|$, end: $G(\mathcal{S}) = \mathcal{G}$;
4. $i \leftarrow i + 1$;
5. if $\text{cons}(d_i|\mathcal{G})$, then $\mathcal{G} \leftarrow \mathcal{G} \cup \{d_i\}$, go to 3;
6. let $D = \{g \in \mathcal{G} : \text{the plans for } g \text{ and the plans for } d_i \text{ are conflicting, or } \forall \mathcal{I}, \mathcal{I} \not\models g \wedge d_i\}$; if $\{d_i\} \succeq D$, then $\mathcal{G} \leftarrow \mathcal{G} \cup \{d_i\} \setminus D$, go to 2;
7. otherwise, go to 3.

In the worst case, i.e., when the desires are ordered in such a way that, each time a new desire is considered, it is inconsistent with \mathcal{G} and the condition in Step 6 is satisfied, the total number of iterations is $\|\mathcal{J}\|(\|\mathcal{J}\| - 1)/2$.

The above algorithm always terminates with the most preferable goal set, since, at each iteration t , \mathcal{G}_t is a goal set, $\mathcal{G}_{t+1} \succeq \mathcal{G}_t$, and the algorithm attempts adding all d_i 's to \mathcal{G} ; whenever a desire already in \mathcal{G} must be dropped, the algorithm attempts again adding all d_i 's to \mathcal{G} .

We observe that, in the case of the example, $G_{\succeq}(\mathcal{S}_0) = G_u(\mathcal{S}_0)$. This happens because we dispose of utilities and the preference relation derives from utilities when the latter are available.

Function G_{\subseteq} returns the maximal (for set inclusion) subset of consistent desires. The complexity of the corresponding problem appears to be NP-hard. We are still working on a proof of this claim.

In the example, $G_{\subseteq}(\mathcal{S}_0)$ may return two consistent sets of desires $\{d_H, d_{80}\}$ and $\{d_{40\%}, d_{80}\}$.

7 Conclusion

We believe the key points of this work are the following:

- (i) the proposition of a framework for goal revision;
- (ii) the idea of computing the goal set from the mental state of an agent, instead of a previous goal set;
- (iii) the provision of three alternative methods to find the most preferable goal set under different hypotheses about desire preference information.

Many simplifying hypotheses have been made, but we plan on relaxing them in future work. The cost of actions and, as a consequence, of plans, has not been considered, but its role is not negligible in real applications and should be taken into account, for example by discounting the utility of a desire with the cost of the optimal plan to achieve it; interactions and trade-offs between desires could be in order and should be properly accounted for. This direction of investigation calls for an integration of decision theoretic concepts and results in our work. The use of a classical logic framework should not be seen as a limitation, as the same, or very similar, results can be proved for different logical frameworks, e.g., argumentation.

References

- [1] Carlos E. Alchourrón, Peter Gärdenfors, and David Makinson. On the logic of theory change: Partial meet contraction and revision functions. *J. Symb. Log.*, 50(2):510–530, 1985.
- [2] L. Amgoud. A formal framework for handling conflicting desires. In *Proc. of the ECSQARU-2003 Conference*, pages 552–563, 2003.
- [3] L. Amgoud and I. Rahwan. An argumentation-based approach for practical reasoning. In *To Appear in Proc. of the AAMAS-2006 Conference*, 2006.
- [4] John Bell and Zhisheng Huang. Dynamic goal hierarchies. In *PRICAI '96: Proceedings from the Workshop on Intelligent Agent Systems, Theoretical and Practical Issues*, pages 88–103, London, UK, 1997. Springer-Verlag.
- [5] J. Benton, Minh Binh Do, and Subbarao Kambhampati. Over-subscription planning with numeric goals. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 1207–1213, Edinburgh, Scotland, UK, July 30–August 5 2005. Professional Book Center.
- [6] Steven Shapiro, Yves Lespérance, and Hector J. Levesque. Goal change. In Leslie Pack Kaelbling and Alessandro Saffiotti, editors, *IJCAI-05, Proceedings of the Nineteenth International Joint Conference on Artificial Intelligence*, pages 582–588, Edinburgh, Scotland, UK, July 30–August 5 2005. Professional Book Center.
- [7] David E. Smith. Choosing objectives in over-subscription planning. In Shlomo Zilberstein, Jana Koehler, and Sven Koenig, editors, *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*, pages 393–401, Whistler, British Columbia, Canada, June 3–7 2004. AAAI.
- [8] John Thangarajah, Lin Padgham, and James Harland. Representation and reasoning for goals in bdi agents. In *CRPITS '02: Proceedings of the twenty-fifth Australasian conference on Computer science*, pages 259–265, Darlinghurst, Australia, Australia, 2002. Australian Computer Society, Inc.

ACTIVE OBJECT DETECTION

G. de Croon E.O. Postma

*MICC-IKAT, Universiteit Maastricht, P.O. Box 616, 6200 MD, Maastricht, The Netherlands*¹

Abstract

Most existing object detection methods passively scan images to find the target object. Passive scanning is computationally expensive and inefficient: at each sampling point feature extraction is performed, while the probability of detecting an object is very low. In this article we explore the viability of active scanning for object detection. In active scanning, each feature extraction is utilised to constrain the further scanning process and to detect objects. We present an active object detection method and identify two requirements for the successful application of active scanning for object detection: (1) local samples contain information on the location of the object, and (2) subsequent samples should decrease the distance to a target object. We show that both requirements are met in a license plate detection task. Our active scanning method attains a test performance of 91.75% on the license plate task. We conclude that active scanning provides a fast and efficient alternative to passive scanning.

1 Introduction

Object detection is the automatic determination of the image locations of objects that are instances of a predefined class. Numerous methods for object detection exist (e.g., [2, 4, 5, 7]), most of which scan a part of the image at some stage of the object detection process. The scanning for objects is performed in a passive manner: local features are extracted at all points of a sampling grid defined by regularly-spaced locations on the image. For example, in the object detection method of Viola and Jones [7] the entire image is scanned at all points of a grid. More efficient variants, such as the one in [4], employ global features to determine a region of interest. Subsequently, passive scanning is performed within the region of interest. Despite such pre-selection schemes, passive scanning remains computationally expensive and inefficient: at each sampling point computationally costly feature extraction is performed, while the probability of detecting an object is very low. In this article we explore the viability of active scanning for object detection. In the active scanning for objects, all feature extractions are utilised both to constrain the further scanning process and to detect the object. We present an active object detection method that maps local image samples to shifting vectors indicating the next sampling position. The method takes successive samples towards the expected object location, while skipping regions unlikely to contain the object.

We identify two requirements for the successful application of active object detection. Firstly, local samples should contain some information on the location of the target object. Secondly, subsequent samples should decrease the distance to the object. We evaluate our active object detection method by determining empirically to what extent both requirements are met and by assessing its detection performance on a license plate detection task.

The remainder of the paper is organized as follows. We present the active object detection method in Section 2 and discuss the experimental setup in Section 3. Then we test to what extent the two requirements are met in Section 4. Also, we determine the performance of the active detection method on the license plate task. The results obtained lead us to introduce and test an extension of the method in Section 5. Then, we discuss the implications of our results in Section 6. Finally, we draw our conclusions in Section 7.

¹g.decroon@cs.unimaas.nl, postma@cs.unimaas.nl. This research is carried out within the ToKeN VindIT project (grant number 634.000.018) of the Netherlands Organisation for Scientific Research (NWO).

2 The Active Object Detection Method

The active object detection method consists of two parts: the first part extracts features from a local sampling window, called the fovea, in the image and the second part is a controller that computes the scanning shift vector from the extracted features. Figure 1 shows a schematic overview of the active object detection method and its two parts.

In the feature-extraction part of the method, a square window of size s^2 (with four inner quadrants) is extracted from the image and downsized by a factor r . The pixel gray values of each quadrant are concatenated to form a vector. We reduce the dimensionality of each vector using principal component analysis (PCA), yielding n components. The concatenation of the four vectors forms the input vector i (of length $4n$) to the second part of the method, the controller.

The controller is a feedforward multilayer neural network, as depicted in the right part of Figure 1. Boxes in the figure represent layers of neurons, lines indicate that the layers are completely connected. The controller has two output neurons (o_1 and o_2). The activities of these neurons specify the direction and magnitude of a shift of the sampling window in the image (dx, dy), with $dx = o_1m$ and $dy = o_2m$, where m is a scaling factor.

The active object detection method takes its initial sample from a random location and then proceeds by extracting the features from the window centered at that location, mapping these features on a vector towards a new location, moving towards the new location, and so forth. This cycle is repeated until a predefined maximum number of time steps has been reached.

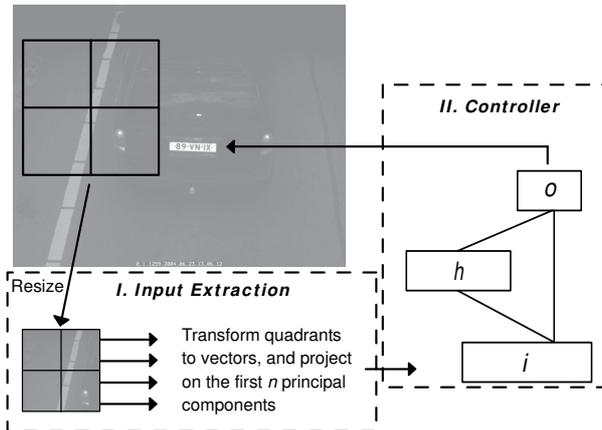


Figure 1: Overview of the object detection method.

3 Experimental Setup

3.1 The License Plate Detection Task

To evaluate the active object detection method, we apply it to a license plate detection task. For this task, we use a private data set of the company Prime Vision BV (<http://www.primevision.nl>). The data set consists of 2930 labeled gray-scale images containing photos of cars, motor bikes, and trucks on the highway. All photos have been taken from behind and above vehicles on the highway, under different lighting conditions. Each image contains a white alphanumeric registration code at the bottom of the image. Figure 2 shows some example images from the image set, in which we changed some of the license plate letters and numbers for privacy reasons (all images in this paper are reproduced with permission of Prime Vision). Each image is 1300 pixels wide and 1030 pixels high. We do not preprocess the images before applying the active object detection method to it. In the experiments reported in this article, we divide the image set into a training set of 2640 images and a test set of 290 images.

3.2 Experimental Settings

In our experiments the width and height of the fovea (i.e., the sampling window) is set to one third of the image width, $s = 433$ pixels. The window is downsized by a factor $r = 0.4$, using



Figure 2: Example images from the data set.

bicubic resampling. To compute the principal components we apply iterative simple PCA [6] on a collection of uniformly sampled quadrants from the training set images and retain the first $n = 10$ principal components. The controller is defined as a neural network with hyperbolic tangent transfer functions in both the hidden and output layers. The output values are multiplied by the scaling factor m that is set to half the width of the image, $m = 650$ pixels. If the shift results in a location outside of the image, the location is reset to the closest location for which this is not the case. The maximum number of time steps T is four, which implies that the method performs three scan shifts.

3.3 Training the Controller

We train the controller to map a local sample directly to the center of the target object in the following manner. We sample the training set images at uniformly distributed image locations, taking into account that the sample window has to stay within the boundaries of the image. For each sample location we extract the input vector i as described in Section 2.1, and store it for training. In addition, we store the relative vector of the sample location towards the target location (this vector represents the ideal scanning shift). The extracted inputs and calculated shift vectors serve as inputs and targets for the neural network training procedure, respectively. We train the feedforward multilayer neural network with one-step secant backpropagation [1]. We used a learning rate of 0.01 and trained the network for maximally 300 epochs with the help of a validation set, consisting of one tenth of the gathered inputs and targets. Training stopped when the validation set error increased (i.e., early stopping) or if the maximum number of epochs was met.

4 Experimental Evaluation

In this section, we verify if the two requirements for the viability of the active object detection method are met. We first determine whether local samples of the image contain information on the object location. Then, we verify whether subsequent samples approach the location of the object. Finally, we determine the performance of the trained active object detection method on the license plate detection task.

4.1 Do Local Samples Contain Information on Object Location?

To determine whether local samples of the image contain information on the object’s location, we investigate whether there are clusters of visual inputs that occur in a structural spatial relation to the center of the license plate. To this end, we first gathered 10 inputs per training image by sampling the image at uniformly distributed locations. On the basis of the resulting 26,400 inputs (our training set contains 2640 images), we determine k input clusters by using k -means clustering (see, e.g., [3]) with $k = 9$. To assess whether there is a structural spatial relation between the input clusters and the license plate location, we sample each test set image on all points of an evenly-spaced grid of 25 by 25 points, covering the whole image. For each input vector i we determine the nearest cluster center and store the x - and y -distance from the sample to the license plate, annotating it with the cluster number. In addition, we also store the absolute coordinates of the sample location, annotating it with the cluster number. Using these data, we are able to visualise the structural spatial relation of the input clusters relative to the target location and in absolute coordinates.

Figure 3 shows for each input cluster where it occurs most often in the image relative to the license plate location (high light intensity represents high occurrence). The center of each image represents the location of the license plate (indicated with a white cross and black circle).



Figure 3: Distribution of the nine input clusters relative to the target location. The center of the license plate is in the center of each image and is indicated by a white cross surrounded by a black circle. For each coordinate, the light intensity is proportional to the frequency of occurrence of each cluster. These results were obtained with a sampling window size of one third of the image width.

Figure 3 reveals that some clusters have a clear spatial relation to the center of the license plate, whereas for other clusters the relation is less clear. Input clusters that occur within a confined region of the image in the figure contain information on the location of the license plate. In contrast, input clusters that have a uniform distribution in Figure 3, do not contain information on the location of the license plate. Since in Figure 3 none of the input clusters is uniformly distributed, all of them contain some information on the license plate location. For example, Figure 3 shows that input clusters 1, 2, and 6 are located at different distances to the bottom of the license plate, where cluster 6 is almost located on the license plate. The figure also shows that cluster 4 usually occurs to the left of the license plate. Furthermore, clusters 7 and 8 mostly occur above the license plate. Only 1% of the input vectors was mapped to cluster 9.

Figure 4 shows the occurrence of clusters in absolute coordinates. The figure shows that cluster 6 (and hence the license plate) can occur almost anywhere in the image, with a higher probability of occurring in the middle. We can also see that input cluster 4 usually occurs in the left part of the image, which sometimes contains the left line that marks the border of the road.



Figure 4: Occurrence of the nine input clusters in the test images in absolute coordinates. High light intensity represents high occurrence.

Our analysis suggests that the local samples taken from our image set contain information on their spatial relation to the license plate location. Since the sampling window is rather large in our experiments (one third of the width of the entire image), one could argue that the samples we take are not very ‘local’. We assessed whether smaller windows can still contain information on the license plate location. Figure 5 shows the distributions of nine input clusters for a window size of one sixth of the image width. Clearly, there are still input clusters that are confined to small regions. However, the figure also shows that there is a relation between the window size and the distance at which local samples give information on the location of the license plate. This is reflected in the observation that confined clusters in Figure 5 are located closer to the license plate and are confined to smaller regions than those shown in Figure 3. The smaller window size seems to result in more information on the license plate location at a shorter distance and less information at a larger distance. These observations suggest an approach where the size of the sampling window depends on the distance to the object. We will adopt such an approach in our extended active detection method in Section 5.

4.2 Do Subsequent Samples Reduce the Distance to the Object?

Having established that local samples can contain information on the target location, we now turn to the second requirement for active object detection. In order to determine whether, in active



Figure 5: Distribution of the nine input clusters relative to the target location. The center of the license plate is in the center of each image and is indicated by a white cross surrounded by a black circle. For each coordinate, the light intensity is proportional to the frequency of occurrence of each cluster. These results were obtained with a sampling window size of one sixth of the image width.

scanning, subsequent samples reduce the distance to the object, we train the controller of our method on samples from the training set. Then, we apply the trained neural network to samples in the test set. For each sample we determine the distance to the center of the license plate before and after performing the shift as determined by the controller. Figure 6 shows a plot of the distance before and the average distance after performing a shift (dashed line). The standard deviation is represented by the gray, dotted lines. The solid line represents the case in which the distances before and after the shift were to be equal. The figure shows that the trained controller is unable to locate the license plate in one time step. If that were the case, the dashed line should correspond to the horizontal axis. However, it also shows that, for distances larger than 110 pixels, the average distance after the shift is smaller than the distance before the shift. Therefore, the distance to the center of the license plate becomes smaller if we take multiple samples in sequence. The fact that the dashed and solid line in Figure 6 intersect at 110 pixels indicates that on average, the method will not converge to the license plate location. This limitation will be discussed in Section 5. Figure 7 illustrates the shifts generated by the trained neural network for 10×10 initial points on a regular sampling grid superimposed on the central region of the image. The direction and length of each arrow emanating from these points represent the shift towards the next sampling location. The figure shows that most arrows point inwards towards the license plate, whereas a few arrows point towards the ‘wrong’ direction, e.g., the arrows at the bottom and the top left. These errors are due to the local nature of the sampling window. For instance, the wrongly-directed arrows at the bottom are due to the registration code at the bottom of the image.

These results indicate that in most cases, taking subsequent samples reduces the distance to the license plate location.

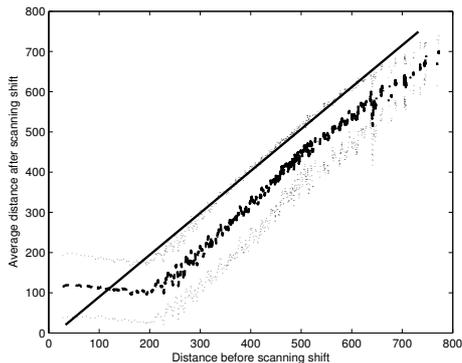


Figure 6: The dashed line is a plot of the distance before a shift and average distance after a shift in the image. The gray, dotted lines indicate the standard deviation. The solid line represents the case in which the distances before and after the shift were to be equal. For distances larger than 110 pixels, the average distance after the shift is smaller than the distance before the shift.

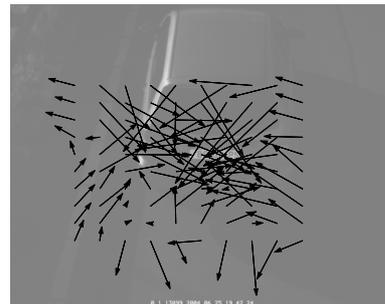


Figure 7: Arrows represent the magnitudes and directions of the shifts generated by the controller for 10×10 initial points on a regular sampling grid superimposed on the central region of the image. Most arrows point inwards towards the license plate.

4.3 Application of the Active Object Detection Method

Having established that the two requirements for active object detection are met, we now assess the performance of our method on the license plate task. Figure 8 shows four consecutive time steps of 10 different runs of the active object detection method. Per run, the center of the fovea is indicated with a white cross. The figure illustrates that the method approaches the license plate, but does not always reach the center of it. Testing 10 runs of the model on each image in the test set shows that the initial average distance to the center of the license plate is 323 pixels. After 3 scanning shifts, this distance is reduced to 179 pixels. This result agrees with the analysis performed in Section 4.1 and 4.2. Apparently, our method is capable to approach the target object, but it fails to reach it. In an attempt to understand what is causing this failure, we performed an analysis of the individual runs.

The analysis revealed that, generally, a few of the ten runs end up in a region far from the license plate. This is illustrated in Figure 8. At $t = 4$, two outliers are clearly visible at the top left. To compensate for these outliers, we compute the median of the ten locations. The median is represented by a circle in Figure 8. As can be seen, the median is rather close to the license plate. For testing our method on the test set, we use the median of 30 independent runs to indicate the location of the license plate. With 30 runs, at $t = 4$, the average distance of the median to the center of the license plate is 82 pixels, and only in 49% of the test images the median is located on the license plate. Since the detection performance is low, we propose an extension to the method to improve the performance in the next section.

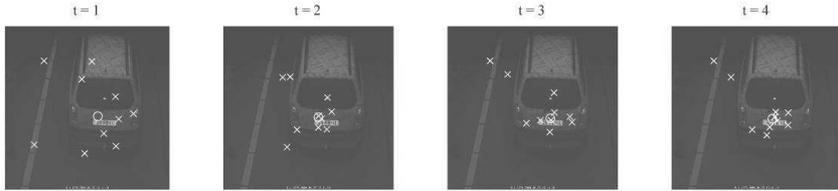


Figure 8: Four successive time steps of the active object detection method. At $t = 1$, the fovea is located at a random location. At $t = 4$, the method has performed three successive shifts in the image. The figure shows ten independent runs of the method, where the center of the fovea of each run is indicated with a white cross. The white circle represents the median coordinates of the fovea centers of all ten runs.

5 The Extended Active Object Detection Method

In subsection 4.3 we showed that on average our method fails in approaching the license plate sufficiently close to detect it. We discern at least two possible causes for this failure. Firstly, the neural network and the principal components are tuned on samples at various distances to the license plate, whereby the samples in the vicinity of the license plate are under-represented. As a result, the controller may perform worse near to the license plate. Secondly, the method may get stuck in a local optimum, even though the sample at that location does not resemble a license plate. These local optima affect the object detection performance. In this section, we first extend the active object detection method, so that it does not suffer anymore from the first cause of failure. Then, we further extend the method, so that it becomes less prone to the second cause of failure. We evaluate the impacts of the extensions, by determining the performance of the extended method on the license plate detection task. As in Section 4.3, we employ 30 independent runs and take the median of the locations reached at the final time step to indicate the license plate location.

The first extension entails employing a sequence of three neural networks, each of which is trained on samples at different distances from the license plate. The first and second neural network employ a sampling window size of one third of the image width, while the third network employs a sampling window size of one fourth of the image width. The first neural network is trained at uniformly distributed sample locations, as described in Section 3.3. Then, we determine the average distance to the license plate after applying this first network to the training images. We use this average distance to determine new principal components of samples extracted at or around this

distance and to train the second neural network. More specifically, we now sample (for both PCA and network training) at normally distributed locations centered at the license plate location. The standard deviation of the sampling is equal to the average distance to the target location of samples after the application of the first neural network. The third neural network is trained in the same manner, but tuned to the sample locations reached by the second neural network. Figure 9 shows the application of the extended active object detection method for six time steps, again showing a cross per run and a circle to indicate the median coordinate. The first extension results in a performance of 85% on the test set. Clearly, this extension yields a considerable improvement over the performance of 49% as achieved in the original method.

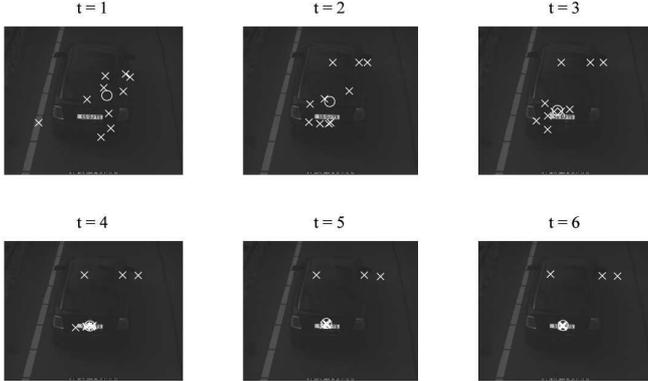


Figure 9: Example of the extended active object detection method. The first network takes one action, while the second and third network take two actions. The foveas of 10 example runs are indicated by white crosses, while the circle represents the median location of all 10 runs.

As stated before, our method can get stuck in local optima, even though the samples associated with these optima do not resemble the object at all. Therefore, we extend the method with a dedicated neural network that is specifically trained to estimate the distance to the target location. After training, the network maps the last sample of a run to an estimate of the distance d to the license plate. The feedforward multilayer neural network has 30 hidden neurons and one output neuron with hyperbolic tangent transfer function. The neural network is trained on the training set to output values $e \in [0, 1]$, according to the following function: $e = d/\theta$, if $d < \theta$, and $e = 1$ otherwise. We set θ , the threshold at which we cut off the distances, to one third of the width of the image. The effect of the second extension can be explained using Figure 9. The last image of Figure 9 shows that three out of ten runs end up far away from the license plate. The dark areas where they end up, clearly do not contain a license plate. If, at the end of multiple runs, we remove all samples for which $e > 0.10$, then these three runs are discarded. In this way we assure that we only accept sample locations very close to the center of the license plate, since $e = 0.10$ roughly corresponds to one fifth of the average license plate width. With the addition of the distance estimation network and by estimating the license plate location with the median of all runs with $e \leq 0.10$, the method attains a performance of 91.75% on the test set².

6 Discussion

The results in Section 3 and 4 show that active scanning can restrain the scanning process in a sensible manner. Moreover, our active scanning method has been shown to achieve a reasonable performance on a license plate detection task. There are two main issues that are of interest for the application of the active object detection method.

The first issue is the generality of the approach. In this paper, we have applied the method to a task of license plate detection. To determine the generality of the approach, it will be necessary to apply the active object detection method to other tasks. However, as mentioned in [4], most objects are situated in a more or less fixed context. This observation leads us to believe that our active object detection method can also be successfully applied to other types of problems.

²Note that the addition of the distance estimation network is necessary for tasks in which there can also be none or multiple objects in an image.

The second issue concerns the main advantage of the active object detection method with respect to passive object detection methods: the computational efficiency. As stated in the introduction, the passive scanning of traditional methods is computationally demanding, which may hamper application to large images and large data sets. Even though the active object method might need local samples with quite a large spatial extent and may also perform multiple runs to overcome local optima, the saving in computation time might be significant. In what follows, we provide a tentative indication of the computational costs of our active method as compared to (partially) passive methods. A traditional method that scans the entire image at all points of an $N_1 \times N_2$ grid performs $N_1 N_2$ feature extractions. In contrast, our active method performs $M(T - 1)$ features extractions, where M is the number of method runs, and T the number of time steps. If we take a separation of sampling points of 10 pixels for the passive scanning, then N_1 and N_2 are both approximately equal to 100. This leads to about 10,000 feature extractions, whereas our active scanning method with $M = 30$ and $T = 6$ performs 150 feature extractions. Active scanning requires roughly 66 times less computational effort under these conditions. Our tentative indication of the computational advantage of active scanning over its passive counterpart is only intended to give an idea of the computational demands of both methods. A more detailed comparison of both types of methods on a variety of tasks is needed to firmly establish the relative performance and computational characteristics of the active scanning approach.

7 Conclusions

Our analyses and experimental results show that active scanning is a viable approach to object detection. The active object detection method meets the two requirements for successful application to object detection. Firstly, local samples of the image contain information on their relation to the object's location. Secondly, consecutive local samples reduce the distance to the object's location. We introduced an extended active object detection method that employs three cascaded controllers for determining shifts and that estimates the distance from the last sample to the center of the license plate. The extended method generally shifts its sampling window to the center of the license plate: it detects 91.75% of the license plates in our test set. We conclude that active scanning is a viable and computationally efficient approach to object detection. Our future research aims at further establishing the advantages and disadvantages of active scanning compared to passive scanning in a variety of tasks. In addition, we envisage refining the implementation of the active object detection method by employing different feature extraction techniques and controllers.

References

- [1] R. Battiti. First and second order methods for learning: Between steepest descent and newton's method. *Neural Computation*, 4(2):141–166, 1992.
- [2] N.H. Bergboer, E.O. Postma, and H.J. van den Herik. A context-based model of attention. In *Proceedings of the 16th European Conference on Artificial Intelligence (ECAI), Valencia, Spain*, pages 927–931, 2004.
- [3] A.K. Jain, M.N. Murthy, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3), 1999.
- [4] K. Murphy, A. Torralba, D. Eaton, and W.T. Freeman. Object detection and localization using local and global features. In *Sicily workshop on object recognition*. Lecture Notes in Computer Science, 2005.
- [5] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38:15–33, 2000.
- [6] M. Partridge and R.A. Calvo. Fast dimensionality reduction and simple pca. *Intelligent Data Analysis*, 2:203–214, 1998.
- [7] P. Viola and M. J. Jones. Robust real-time object detection. *Cambridge Research Laboratory, Technical Report Series*, 2001.

Intelligent traffic simulation based on Multi-Agent system

Pavel Dergel¹ and Petr Fuks²

VSB technical university of Ostrava, Institute of geoinformatics, Czech republic,
pavel.dergel.st@vsb.cz, petr.fuks.hgf@vsb.cz

Abstract. This paper describes a process of design of an intelligent traffic simulator with a multi-agent system (MAS).

The simulation is based on real geographic data that are stored in a spatial database. The main aim of our work is to develop methods for building a real-world based MAS with cognitive agents. We verify the methods on the example of the traffic simulator, which simulates a movement of cars (mobile agents) through a road network which is represented by static agents. The mobile agents are able to use an artificial intelligence for making decisions and are communicating using a higher communication language based on W3C OWL.

Our future work will be focused on closer connection between geo-informatics and MAS which will allow us to develop more sophisticated simulations much closer to reality. . . .

1 Traffic simulation

In order to design simulation of traffic system we need to create virtual traffic infrastructure and simulate traffic participants like cars, buses, bikes and etc. According to this, our work is divided into the two parts:

- Environment modeling
- Creating of mobile agents

2 Environment modeling

A road network environment is for our purposes represented by agents. Roads, crossroads and other infrastructure elements are represented by agents. It allows us to completely build distributed environmental architecture as shown in figure 1.

The operational environment of the traffic simulation is composed of active components (agents) with mobile agents which have to interact while moving through the road network. The infrastructure agents provide mobile agents with necessary information about the current traffic situation, traffic signs, obstructions etc.

The environment is spatial. Some basic properties of space and how objects exist in space are part of the 'law of the universe'[4]. The environment has following properties.

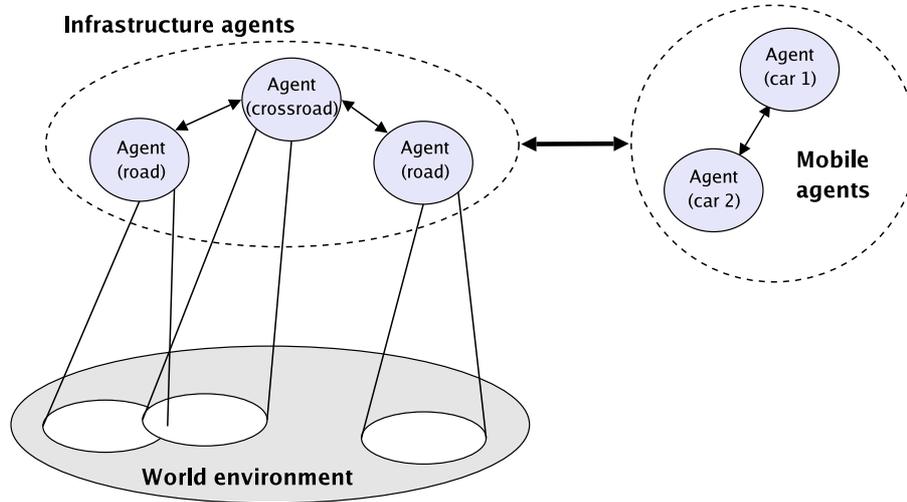


Fig. 1. Environment representation

- Each mobile agent is located at a determined location and can be located only at one point at the same time.
- The agent can move from one place to another. These moves must follow an established path and environment restrictions. The path is represented in discrete way.
- Non agent objects in the environment can be perceived by software or hardware sensors.

These points cover the basic environmental properties and allow mobile agents to make spatial decisions while moving.

2.1 Environment perceiving

There are two types of sensors, which agents can use for environment perceiving:

- **Hardware sensors** - the agent sense surrounding environment and transforms it into digital representation which is able to understand. Examples of such sensors are GPS receiver which calculates the current position, camera which provides agent with visual information etc.
- **Software sensors** - the agent is able to receive information about the environment from geographic database (figure 2).

For the simulation purposes we use only software sensors for environmental perceiving. Our future work will include an investigation of possibilities of the hardware sensors as well.

Access to the geographic database is provided by database agents. They are able to perform analytical operations upon geographical data (distance measuring, intersections, etc.) and transmit the results to other agents using the ACL communication language.

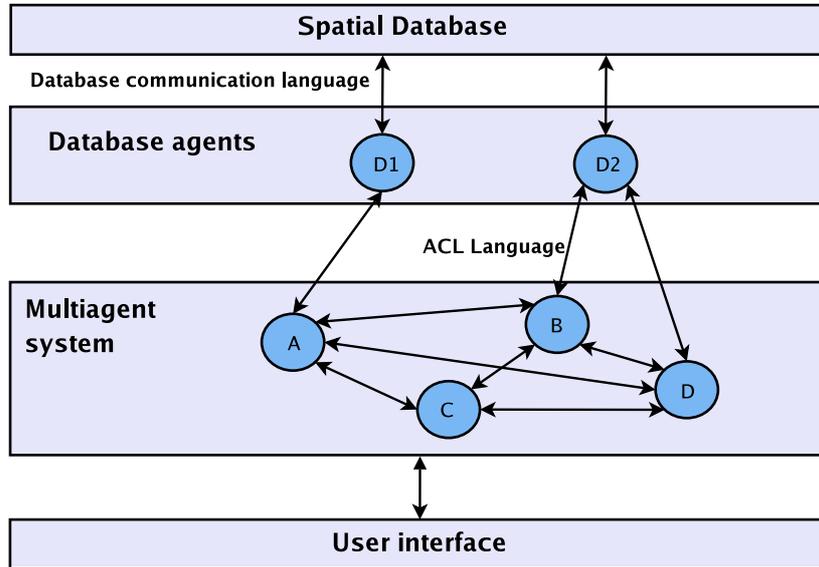


Fig. 2. Perceiving environment using spatial database

2.2 Time representation

The synchronisation of agents plays an essential role in traffic simulation and the agents are fundamentally autonomous and communicate asynchronously. The coordination of mobile agents in such an environment is very difficult. We are using a synchronisation mechanism and a discrete time representation. This approach allows us to describe the whole MAS as a state machine. Mobile agents are moving in turns which are controlled by the synchronisation agent. Discrete time representation simplifies agent decision process and coordination.

Synchronisation intervals are regulated by the synchronisation agent in a way that whole the simulation is getting closer to reality.

2.3 Ontology description

This phase defines the basic concepts used for communication between agents. We have developed the ontology using W3C OWL language. The ontology is represented in OWL language (extended RDF)[5].

There are many OWL editing programs (Protege, SWOOP, etc.). These programs provide convenient user interface for OWL creating. An example of a part of our infrastructure ontology created in Protege editor is shown on figure 3.

This ontology can be used as a belief base and represents the agent's knowledge. Agents are able to exchange beliefs between each other and are able to learn. It is also possible to use some inference mechanism and derive some new facts on the fly.

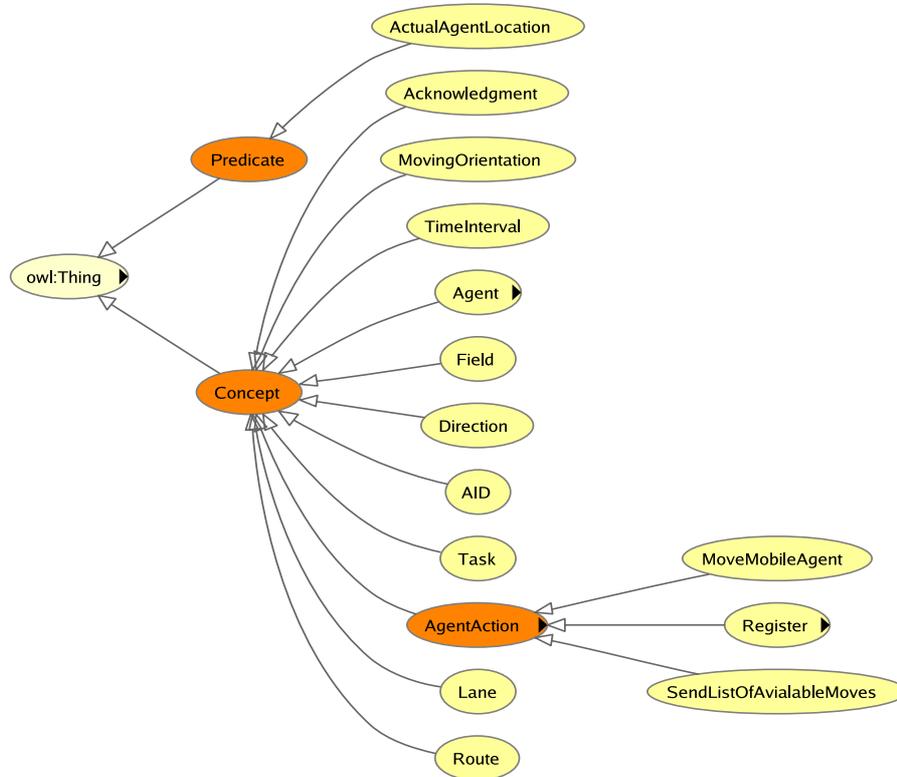


Fig. 3. Example of the infrastructure ontology.

3 Mobile Agent

We construct mobile agents that act in an artificial environment. The environment represents parts of the real world and can dynamically change over time. Agents are supposed to cope with the difficulties associated with these changes. The mobile agent is composed of three modules:

- sensor(s) module
- brain module
- driving module

According to this architecture, the simulation cycle of a mobile agent is divided into three logical phases: Sensing, Deliberation and Action. In the first phase of sensing, the agent gets data about surrounding agents and objects. The list of objects is transformed into facts (beliefs) which are stored in the agent's mind. Then the brain module will decide its corresponding actions according to the rules and current state in the deliberation phase, and finally in the third

phase, the agent will perform action. Actions of mobile agents are judged by infrastructure agents at the end of the simulation cycle (turn).

4 Implementation

The described traffic simulation has been implemented using the JADE (Java Agent DEvelopment Framework) environment (<http://jade.tilab.com>). It is a framework fully implemented in Java language. It simplifies the implementation of multi-agent systems through a middle-ware that complies with the FIPA specifications. The agent platform can be distributed across machines (which not even need to share the same OS).

We have decided to use JADE for several reasons:

- Implemented in Java language,
- FIPA compliant,
- Easily extensible and customisable,
- Contains a set of graphical tools that supports the debugging and deployment phases
- Released under GNU/LGPL license.

The deliberation phase of agent thinking will be implemented in Jena. Jena is a Java framework for building Semantic Web applications. It provides a programmatic environment for RDF, RDFS and OWL, SPARQL and includes a rule-based inference engine (<http://http://jena.sourceforge.net/>).

We have used the PostgreSQL database with PostGIS extension for storing geographic data. PostGIS adds support for geographic objects to the PostgreSQL object-relational database. In effect, PostGIS "spatially enables" the PostgreSQL server, allowing it to be used as a backend spatial database for geographic information systems (GIS). More information about PostgreSQL is available on <http://postgis.org>.

5 Practical applications

During the last decades, high growth in car use has resulted in a rapid increase of traffic congestions. Congestion wastes fuel and increases air pollution due to increased idling, acceleration, and braking. Since driving time is a non-productive activity, congestion reduces regional economic health by increasing drive times.

Without better control road transport will be less effective and more dangerous especially in big agglomeration. Today we already have some systems that inform drivers about situation on the road. But today's information systems only react to situations which already happened. When we improve these information system with some kind of predication we will be able to inform drivers and police in advance and avoid traffic collisions and accidents.

Intelligent traffic systems based on MAS can be successfully used in following areas:

- **Rescue systems** - advanced navigation through road network which will be able to avoid congestions and dynamically find optimal route which can minimize time needed to get to the place of an accident and help to save human lives.
- **Transportation of goods** - goods transportation and delivery in urban areas represents a very big problem for the majority of cities. The key to success for delivery companies is to transport goods at the right time, as fast as possible and for the lowest possible price. Wrong timing and coordination results in increased expenses, higher traffic intensity etc. Intelligent navigation systems can help to improve this situation.
- **Security systems** - Preventive safety applications help drivers to avoid or mitigate an accident through the use of in-vehicle systems which sense the nature and significance of the danger, while taking the driver's state into account. Preventive safety makes use of information, communications and positioning technologies to provide solutions for improving road safety. With such technology - which can operate either autonomously on-board the vehicle or co-operatively based on vehicle-to-vehicle or vehicle-to-infrastructure communication - the number of accidents and their severity can be reduced, leading to a decrease in the number of accidents.

6 Conclusion and future work

We have developed the multi-agent traffic simulation that allows us to solve simple tasks connected with road movement and navigation. The model represents a parallel world to the real road network which allows us to study the complexity of the traffic flow.

Computer simulation is a very useful tool for planning and predication of all kinds. If we collect enough data from road sensors we will be able to simulate parallel traffic system. This virtual world can show traffic bottlenecks and other traffic problems. Moreover we may combine traffic information with for example weather forecasting, or other information sources to simulate environment impact on traffic situation.

Our future work includes the following tasks:

- Developing new agent coordination and cooperation methods which will allow agents to negotiate during difficult traffic situations and find an appropriate solution.
- Improving navigation system for mobile agents by taking into account factors like actual (and expected) traffic situations, surface conditions, accidents, throughput etc.
- Developing 3D visualization and a control system which will be used as an user interface.
- Improving system performance to be able to run the simulation in real-time.

7 Acknowledgements

This research has been supported by the program "Information Society" of the Czech Academy of Sciences, project No. 1ET101940420 "Logic and Artificial Intelligence for multi-agent systems".

References

1. FIPA: The Foundation for Intelligent Physical Agents, <http://fipa.org>
2. Michael Wooldridge: *An Introduction to MultiAgent Systems*, (2002)
3. Franco Zambonelli, Nicholas R. Jennings, Michael Wooldridge: *Developing Multi-agent Systems: The Gaia Methodology*, (2003)
4. Andrew U. Frank, Steffen Bittner and Martin Raubal: *Spatial and Cognitive Simulation with Multi-agent Systems*
5. <http://www.w3.org/2004/OWL>
6. Lihua Wang, Stephen John Turner and Fang Wang: *Interest Management in Agent-based Distributed Simulations*, (2003).

Coordinated Communication, a Dynamical Systems Perspective

Bart De Vylder
Artificial Intelligence Laboratory, Free University of Brussels

September 8, 2006

Abstract

Over the past years, several computational models have been introduced to study the coordination of communication between distributed agents. Although these models have given valuable insights into the mechanisms required for letting agents develop a successful communication system, few theoretical results have been obtained which substantiate these findings. In this paper we introduce a theoretical framework which allows us to analyze and compare different existing models in a uniform way. Therefore we only look at the observable behavior of an agent and not at the internal mechanisms that cause that behavior. In particular, we define an agent's response function and argue that a stability analysis of its fixed points reveals crucial information about the convergence properties of the dynamical system of interacting agents.

1 Introduction

In the literature various models are described which investigate the origins and evolution of communication systems (e.g. [7, 16, 11, 6, 5]). These models were developed both to understand how natural communication systems might have arisen as well as to gain insights in the design of artificial agents capable of self-organizing a successful communication system. Most of these models are computational by nature and rather few theoretical results exist on this topic which explain or predict the observed phenomena (but see [13, 2, 9]). Such a mathematical foundation however could greatly benefit the research and provide it with new insights and concepts.

One of the most important reasons for the lack of a theory is the difficulty to derive the global dynamics of a population of interacting agents from the behavior of the individual agents, which can be an entity of arbitrary complexity itself. In this paper we try to get more insight in this relation between the individual and the resulting collective behavior. We take a system theoretical perspective on an agent and see it as a black box with inputs and outputs. This allows us to analyze agents irrespective of their internal mechanisms. As a result we will be able to put the different agent models as described in the literature in a common framework.

A dynamical systems perspective on the evolution of communication was also advocated in [3, 4]. In this work, the author studies the applicability of dynamical systems theory in the context of a particular agent architecture. In the paper at hand, we try to extend and generalize these results.

The remainder of the paper is structured as follows. In the next section we describe some existing models from the literature and introduce the framework of language games.

In section 3 the dynamical systems viewpoint is explained and the associated concepts defined. In section 4 we reconsider the different models described in section 2 from the literature and analyze them within this new framework.

2 Models for the Evolution of Communication

Different classes of models for the evolution of language can be distinguished. First, there exist models inspired by biological evolution (see e.g. [12, 10]). In these systems, the fitness of individuals plays a central role in the evolution of their language. Second, in the ‘Iterated Learning Model’ (see [8] for an overview) it is assumed that the most important factor shaping a language’s form is its learnability and not its communicative function. As the name already suggests, the key idea in this model is the repeated transmission of part of a language from adults to children, who become adults themselves after a learning phase. Third, there exist models that make use of the framework of ‘language games’ (see e.g. [19]). These models focus on the role of self-organization in the development of a communication system. Hereby agents negotiate a communication system by playing language games, hence there are no fixed teachers or learners.

In the remainder of this paper we will mainly use the framework of language games. In principle the agents playing the games can be models of animals who use different alarm calls for different sources of danger (see e.g. [1]), software agents which have to agree on labels for different objects, humans who use words and grammatical constructions to express an open ended variety of meanings, etc. In all cases some communication system is involved relating a set of meanings to a set of possible forms. We further assume that these forms are atomic (multiple word utterances are not considered) and will refer to them as words. Some models allow for agents to invent new words and to create new meanings when the environment requires a richer semantic representation (see e.g. [19, 20]). In this paper however we focus on models which in advance assume a fixed and finite set of meanings and words. A communication system then consists of a mapping from the meanings to the words, used for expressing meanings and a mapping from the words to the meanings, used for interpreting the words produced by other agents. In an optimal communication system, these two mappings are the same for all agents and bidirectional: the word produced for a particular meaning is subsequently interpreted again as having that meaning. Thus the task the agents face, is to arrive at the same, bidirectional mapping by interacting with each other and trying to align their individual mappings accordingly.

Within the framework of language games several variations exist. Common for all models is that at each time step two randomly chosen agents play a language game. One agent, the speaker, produces a word for a particular meaning, the *topic*. Another agent, the hearer, interprets this word. Specific models differ, among other things, in whether the hearer immediately is informed about the intended meaning or first has to guess it, whether only the hearer adapts itself after the game or also the speaker and if this is the case whether he takes the hearer’s guess into consideration. All these variations however fit in the framework we define in the next section.

3 Agents as Coupled Dynamical Systems

3.1 The Linguistic Behavior

Before explaining the main model we first introduce the concept of a *linguistic behavior*. The information transmitted between a speaker and a hearer only depends on the way the speaker produces words for the different meanings and on how the hearer interprets words as meanings. We name these mappings respectively the production and interpretation behavior of an agent. Moreover, if only the hearer adapts itself after a game, the only information transmitted is from the speaker to the hearer and this information consequently only depends on the speaker's production behavior. Given a particular type of language game, we will refer to the relevant behavior of an agent as its language, linguistic behavior or shortly behavior, interchangeably. So in the remainder of the paper, depending on the particular model, an agent's behavior can mean the combination of its production and interpretation behavior or only its production behavior.¹

Given meanings $M = \{m_1, m_2, \dots, m_{\#M}\}$ and words $W = \{w_1, w_2, \dots, w_{\#W}\}$, an agent's production and interpretation behavior can be represented as a *production matrix* and an *interpretation matrix*, respectively (see also [6, 15]), both of dimension $\#M \times \#W$. A production matrix is a row-stochastic matrix and gives for each meaning the probability that a particular word is used to express that meaning. An interpretation matrix is a column-stochastic matrix and gives for each word the probability that it is interpreted as a particular meaning. As an example consider the following production and interpretation matrices P and Q for $\#M = \#W = 3$:

$$P = \begin{array}{c|ccc} & w_1 & w_2 & w_3 \\ \hline m_1 & 0.2 & 0.4 & 0.4 \\ m_2 & 0.9 & 0.1 & 0 \\ m_3 & 0.3 & 0.6 & 0.1 \end{array} \quad Q = \begin{array}{c|ccc} & w_1 & w_2 & w_3 \\ \hline m_1 & 0.1 & 0.4 & 0.9 \\ m_2 & 0.7 & 0.5 & 0 \\ m_3 & 0.2 & 0.1 & 0.1 \end{array} \quad (1)$$

Typically an individual agent's production and interpretation behavior will be deterministic rather than stochastic: the matrices then contain exactly one 1 in each row or column, respectively. Still, the concept of a stochastic matrix instead of a function is useful as we will also associate production and interpretation matrices, i.e. a linguistic behavior, with a population instead of an individual agent, and these matrices will typically be stochastic instead of deterministic.

Given a production and interpretation matrix P and Q , we define the *communicative accuracy* as the probability that the word for a randomly chosen meaning is interpreted again as that meaning (see [6, 14, 15] for similar measures):

$$\text{ca}(P, Q) = \frac{1}{\#M} \sum_{i=1}^m \sum_{j=1}^w P_{i,j} Q_{i,j}. \quad (2)$$

We say that a communication system, defined by P and Q is optimal if $\text{ca}(P, Q) = 1$. This is only possible if $\#W \geq \#M$, if every meaning has a unique word and if every used word has the corresponding, unique interpretation. In the case $\#W = \#M$ we have $\text{ca}(P, Q) = 1$ if and only if P and Q are permutation matrices and equal.

The space of all valid linguistic behaviors, given a certain meaning and word set M and W , will be noted as B . Depending on the particular agent model we then have

$$B = \{P \mid P \text{ is a production matrix}\} \quad (3)$$

¹It is also possible to define models in which only the speaker updates after a game, in which case interpretation would be the only relevant behavior.

or

$$B = \left\{ \langle P, Q \rangle \mid \begin{array}{l} P \text{ is a production matrix} \\ Q \text{ is an interpretation matrix} \end{array} \right\} \quad (4)$$

Obviously, given this last definition we say that a behavior $b = \langle P, Q \rangle$ is optimal if $\text{ca}(P, Q) = 1$. In case (3) we do not have sufficient information in principle to conclude that the communication system is optimal. Still, we say a behavior $b = P$ is optimal if P associates with each meaning a unique word.

3.2 An Agent as a Black Box

Specific models of agents differ in how their linguistic behavior is determined and how it changes over time. In some models, the production and interpretation matrices are a function of one association matrix or lexical matrix, representing the strengths of the associations between meanings and words (e.g. in [18, 10, 16, 11]). The production matrix is constructed by taking for each row the maximal element, and the interpretation matrix is constructed by taking for each column the maximal element.²

In another model described in [15], an agent keeps track of an estimate of both the average production and interpretation matrix of other agents, and determines its own production and interpretation behavior on the estimated interpretation and production matrices, respectively.

In principle one can construct unlimitedly many different types of agent architectures, from neural networks to symbolic rewrite systems to cellular automata. Yet, whatever underlying structure an agent has and whatever state the agent is in, its interface to other agents is its linguistic behavior and not the underlying structure, which is not directly observable for other agents. Therefore we will explicitly distinguish between an agent's internal state and its behavior.

In general we will denote an agent's state space as S . The way in which an agent's state determines its behavior is written as a function $f : S \rightarrow B$. Due to interactions with other agents, an agent's internal state changes over time. Suppose an agent a_1 is at time t in state $s_1(t)$ and that he plays a game with agent a_2 in state $s_2(t)$ and behavior $b = f(s_2(t))$. The new state $s_1(t+1)$ of a_1 after the game depends on (i) its current state $s_1(t)$, (ii) whether he is speaker or hearer, (iii) the topic and (iv) the linguistic behavior b of a_2 . The factors (ii) and (iii) are random and mutual independent. We will formally summarize them in a random variable X , such that we can define the stochastic transition function δ of an agent as:

$$s(t+1) = \delta_X(s(t), b). \quad (5)$$

3.3 A Population

We have represented an agent as an abstract system with an internal state $s \in S$, a stochastic transition function $\delta_X : S \times B \rightarrow S$ and an output function $f : S \rightarrow B$. A population of agents consists of a collection of such (identical) systems which interact stochastically. That is to say, at every time step a speaker and hearer are chosen randomly, they play a game about a random topic and update their internal state. In the following, we refer to this system consisting of randomly interacting agents as the *original system*.

²This is of course conceptual, it does not mean that these matrices are explicitly constructed, just that an agent behaves exactly the same as if these matrices would have been constructed this way.

Now, given this assumption we can define the *population linguistic behavior* as the average of the agents' linguistic behaviors. This can encompass only the production matrix or both production and interpretation matrices. It is easily seen that, from a particular agent's point of view, playing a game with a randomly chosen agent is exactly the same as playing a game with the population linguistic behavior (i.e. speaking to/hearing the average interpretation/production matrix).

In general, the coupling of a large number of dynamical systems can lead to complicated dynamics. Therefore in this paper we will not try to derive quantitative but only qualitative properties of the dynamics of the linguistic behavior of agents. What we want to investigate is the relation between a particular agent model and the extent to which a population of such agents is capable to agree upon an optimal communication system. One desirable property is that once the agents have arrived at a particular optimal communication system, they don't change their language anymore. In all models we consider, this will be the case. Despite this property it is still possible that such an optimal state is not reached, for instance if the agents get stuck in a suboptimal communication system. The theory of dynamical systems is a natural framework to analyze these issues, in particular the stability analysis of fixed points. The definition of a fixed point of a stochastic system is somewhat more subtle than in the case of deterministic systems. Yet informally, we will look for population behaviors which on average lead the agents to produce exactly the same behavior. To properly define such fixed points we first define what it means for an agent to produce on average a certain behavior.

3.4 The Agent Response Function

In order to investigate how an agent would behave in the long run when subject to a fixed population behavior for a very long time we introduce the concept of an *agent response function*. This is a function $\psi : B \rightarrow B$, which maps a population behavior to the average agent behavior.

First of all we will investigate under what circumstances such a definition makes sense. Suppose at time $t = 0$ an agent is in state $s(0) \in S$. Then we assume that this agent repeatedly interacts with a fixed population behavior, say $b \in B$. As a result of these interactions the agent will stochastically traverse the state space S :

$$s(0) \xrightarrow{\delta_X(\cdot, b)} s(1) \xrightarrow{\delta_X(\cdot, b)} s(2) \xrightarrow{\delta_X(\cdot, b)} s(3) \dots \quad (6)$$

Thus the population behavior b defines a Markov chain on the state space S . We can define an agent response function if this Markov chain has a unique stationary distribution. Indeed, if a unique stationary distribution π_b exists, for which by definition it holds that³

$$\pi_b(s) = \int_S P(\delta_X(\sigma, b) = s) \pi_b(\sigma) d\sigma, \quad (7)$$

then the agent response function ψ is defined as follows:

$$\psi(b) = \int_S f(\sigma) \pi_b(\sigma) d\sigma. \quad (8)$$

$\psi(b)$ is the average behavior of an agent who traverses the state space S according to the stationary distribution π_b . We say that an agent (architecture) is *ergodic* if for

³In this paper we ignore technical difficulties that may arise with Markov chains on uncountably infinite state spaces.

every population behavior b the associated Markov chain over S has a unique stationary distribution π_b . One property of an ergodic agent is that its behavior in the long run does not depend on its initial state.

The definition of the agent response in terms of the stationary distribution does not mean that we have to explicitly calculate this stationary distribution to compute an agent's response to a certain population behavior. Indeed, precisely the ergodic property implies that the average of the agent's behavior over subsequent samples of the Markov chain is equal to the average with respect to the stationary distribution. In other words, the agent response can be empirically estimated: just let an agent interact with a particular population behavior and observe its average behavior (after a while in order to let the transient fade out). In section 4 we will investigate some models in this way.

From the viewpoint of agent design, we think ergodicity is a good property, because an ergodic agent which is suddenly confronted with a new communication system will eventually adapt to this new situation, irrespective of its 'age' or interaction history. On the other hand we are aware that most natural systems, humans in particular, have a strong non-ergodic component.

Suppose a population of identical, ergodic agents always succeeds in building an optimal communication system. What properties do we expect the response function of the agents to possess? First of all, we expect that an agent's response to an optimal linguistic behavior is that same behavior. Otherwise the population could never arrive at an optimal communication system. On the other hand, if an agent observes behavior which is not optimal, we don't expect the agent just to imitate that behavior, but instead to improve it. Typically this amounts to reducing homonymy and synonymy.

3.5 Fixed Points and their Stability

We have now come to the point where we can define the fixed points of a behavior for a population of ergodic agents. Not surprisingly, we will declare a behavior b^* a fixed point if the agent response to this behavior is again b^* :

$$b^* = \psi(b^*). \tag{9}$$

The reason that this definition makes sense is the following: suppose the population behavior has been equal to b^* for quite some time. Then each agent in the population will traverse its state space according to the Markov chain induced by b^* . As the agents are ergodic, this Markov chain has a unique stationary distribution π_{b^*} . Hence each agent's state is a sample of this distribution π_{b^*} . Consequently the larger the population, the better the collection of states of the agents will resemble π_{b^*} . But if the collection of states of the agents is close to π_{b^*} , then the expected new population behavior will again be close to b^* , as we have, with N the population size and $s_i \in S$ the state of the i^{th} agent:

$$b^* = \psi(b^*) \tag{10}$$

$$= \int_S f(\sigma) \pi_{b^*}(\sigma) d\sigma \tag{11}$$

$$\approx \frac{1}{N} \sum_{i=1}^N f(s_i), \tag{12}$$

Thus a fixed point of the agent response function is also a fixed point of the original system.

What kind of fixed points can we expect to appear? First of all, an optimal population behavior should be a fixed point, because if an agent is confronted with an optimal linguistic behavior, in the end he should adopt this behavior. On the other hand, there can exist suboptimal population behaviors that are fixed points as well, as they arise naturally as a result of symmetry constraints. That is to say, agents are assumed to have no initial preference over meanings or words and their associations. This implies that any permutation p of meanings and/or words of a behavior b must result in the same permutation of the agent response:

$$\psi(p(b)) = p(\psi(b)) \quad (13)$$

Therefore if b contains a symmetry in meanings or words, i.e. b is invariant under a permutation p : $b = p(b)$, then this symmetry is preserved under the agent response function as we have

$$\psi(b) = \psi(p(b)) = p(\psi(b)). \quad (14)$$

As an example, suppose agents only update as hearer, such that only the production behavior is relevant, and that there are two meanings and two words. Furthermore if we assume a symmetry in the two meanings then we have

$$\begin{pmatrix} a' & 1 - a' \\ a' & 1 - a' \end{pmatrix} = \psi\left(\begin{pmatrix} a & 1 - a \\ a & 1 - a \end{pmatrix}\right) \quad (15)$$

with $0 \leq a, a' \leq 1$. So the agent response function must preserve the symmetry present in its argument. In particular if $a = 0.5$ also a symmetry between the words arises and we necessarily have $a' = 0.5$ as well. So every subspace of B defined by a certain symmetry between meanings and/or words is invariant under ψ . While these subspaces not necessarily contain fixed points, in practice we will see they often do. Furthermore these fixed points are suboptimal behaviors, as an optimal behavior does not contain symmetries (because this would imply ambiguities in production or interpretation). Given the presence of these suboptimal fixed points, the crucial question becomes whether they are repellers or attractors in the dynamical system consisting of interacting agents. In the former case the population will be driven away from the suboptimal fixed point and have a chance to reach an optimal communication system. In the latter case an escape from the suboptimal fixed point can take very long (but not forever as there is always a strictly positive probability to escape in a stochastic system).

We now argue that we can determine the nature of these fixed point based on the agent response function ψ . Therefore we define a reduced, deterministic, model of the original system. In the original system, the population behavior is a function of the individual agents' behaviors, which in turn are a function of the agents' internal states. We now take the viewpoint of one agent and make abstraction of the states of the other agents by only considering the (global) population behavior. This agent is stochastically influenced by the population behavior but on the other hand also influences this behavior. As a member of the population the agent will slightly pull the population behavior toward its own. If at time step i the agent's state is s_i and the population behavior b_i , this results in the following system of coupled stochastic difference equations:

$$b_{i+1} = (1 - \beta)b_i + \beta f(s_i) \quad (16)$$

$$s_{i+1} = \delta_X(s_i, b_i), \quad (17)$$

with $0 < \beta < 1$ a constant which parametrizes the degree of influence an agent has on the population. One can think of β as being more or less inversely proportional to the

population size. Now we go a step further and assume that β is small. This means that b changes slowly as a function of i and hence that the agent has many interactions with a roughly constant population behavior. If we assume the agent is ergodic then its average response $f(s_i)$ will approximate $\psi(b)$. In this case we can approximate equations (16) and (17) by the following deterministic recurrence relation:

$$b_{i+1} = (1 - \beta)b_i + \beta\psi(b_i). \quad (18)$$

Moreover if we define $b(t) = b(i\Delta t) = b_i$ and let $\Delta t, \beta \rightarrow 0$ with $\frac{\beta}{\Delta t} = \alpha$, a constant, then we can transform (18) into the following ordinary differential equation:

$$\frac{db}{dt} = \alpha(\psi(b) - b). \quad (19)$$

Not surprisingly the fixed points of the system (19) are precisely those as defined in (9). But what is more, the derivation of (19) suggests that the stability of its fixed points also determines their stability (stable/attracting vs. unstable/repelling) in the original system.

If we apply linear stability analysis in a fixed point b^* of (19), we get the jacobian

$$J = \left. \frac{\partial(\psi(b) - b)}{\partial b} \right|_{b^*} = \left. \frac{\partial\psi(b)}{\partial b} \right|_{b^*} - I = J_\psi - I \quad (20)$$

with $J_\psi = \left. \frac{\partial\psi(b)}{\partial b} \right|_{b^*}$ and I the identity matrix. Let $\{\lambda_j\}$ be the eigenvalues of J and $\{\lambda'_j\}$ those of J_ψ . Then the criterion for stability is $\text{Re}(\lambda_j) < 0$ for all j or, equivalently, $\text{Re}(\lambda'_j) < 1$, as λ'_j is an eigenvalue of J_ψ iff $\lambda'_j - 1$ is an eigenvalue of J . Interestingly, $\text{Re}(\lambda'_j) < 1$ is implied by $|\lambda'_j| < 1$, which is the criterion for stability of the system (18) with $\beta = 1$:

$$b_{n+1} = \psi(b_n) \quad (21)$$

4 Examples and Applications

In this section we will analyze several types of agents, both to clarify the reasoning presented in the previous section and show its general applicability, as to analyze some existing agent architectures presented in the literature.

Examples 1 through 3 show agent models for which a stability analysis of the fixed points of the agent response function provides strong evidence that a population of such agents will always converge to an optimal communication system, which is in agreement with the findings in the corresponding publications. However for the agent model in example 4, a suboptimal but stable fixed point was found, which indicates a problem with respect to convergence to an optimal communication system. This was confirmed by a simulation of the original system.

Example 1

As a first, illustrative example we consider a population of agents which have to agree on a word to use for only one meaning.⁴ Only the hearer updates its state such that it suffices to only consider the production behavior of an agent/population. As there is only

⁴Systems with multiple meanings but no homonymy can also be studied in this way, as argued in [21].

one meaning, the production matrix consists of one row and contains the probabilities with which the different words are used:

$$P = (x_1 \ x_2 \ \dots \ x_{\#W}) \quad (22)$$

The agent architecture is simply a First-In-First-Out buffer which stores the last k observed words. The state space S is the set of all possible buffer contents, hence is finite and contains $k^{\#W}$ elements. An agent always produces the word which occurs most in its buffer. If there are multiple candidates he chooses randomly between them. In particular, we will study the case with only three words, $\#W = 3$, and a buffer length of $k = 3$ such that $\#S = 27$. If we write A, B and C for w_1 , w_2 and w_3 then these are some examples of the mapping from the agent's state to its behavior (with the buffer content as argument to f):

$$f(\text{CAA}) = (1 \ 0 \ 0) \quad (23)$$

$$f(\text{BAB}) = (0 \ 1 \ 0) \quad (24)$$

$$f(\text{BCA}) = \left(\frac{1}{3} \ \frac{1}{3} \ \frac{1}{3}\right). \quad (25)$$

With regard to the agent response function, we first note that the system is obviously ergodic. Moreover the calculation of the probabilities to be in each of the 27 states in the stationary distribution is straightforward. If the population production behavior is given by $P = (x_A \ x_B \ x_C)$, then the chance to be in a state s which has n_A A's, n_B B's and n_C C's is given by

$$\pi_P(s) = x_A^{n_A} x_B^{n_B} x_C^{n_C}. \quad (26)$$

This allows us to explicitly calculate the agent response function ψ as

$$\psi(P) = \sum_{s \in S} \pi_P(s) f(s), \quad (27)$$

which results in

$$\psi \left(\begin{pmatrix} x_A \\ x_B \\ x_C \end{pmatrix} \right)^T = \begin{pmatrix} x_A (x_A^2 + 3x_A(x_B + x_C) + 2x_Bx_C) \\ x_B (x_B^2 + 3x_B(x_A + x_C) + 2x_Ax_C) \\ x_C (x_C^2 + 3x_C(x_A + x_B) + 2x_Ax_B) \end{pmatrix}^T. \quad (28)$$

As $x_A + x_B + x_C = 1$ we can plot the vector field generated by this response function (through equation (19)) on a two-dimensional simplex as is shown in figure 1. The function ψ has 7 fixed points, of which three correspond to an optimal communication system: $(1 \ 0 \ 0)$, $(0 \ 1 \ 0)$ and $(0 \ 0 \ 1)$. The four others are the result of symmetries in P: $(\frac{1}{2} \ \frac{1}{2} \ 0)$, $(\frac{1}{2} \ 0 \ \frac{1}{2})$, $(0 \ \frac{1}{2} \ \frac{1}{2})$ and $(\frac{1}{3} \ \frac{1}{3} \ \frac{1}{3})$. As figure 1 suggests, only the first three are stable fixed points, the other four, sub-optimal fixed points are unstable and consequently pose no problem with respect to convergence to an optimal communication system. In [21], this convergence was proved in general and under rather weak conditions. In this example the conditions for convergence become $k \geq 3$.

Example 2

We consider again a system in which only the hearer updates after a game such that we only consider the production behavior. In this case however we have multiple meanings and homonymy can occur. The system is exactly like one of the models proposed in [7]. The agent's state is a lexical matrix which contains association strengths between

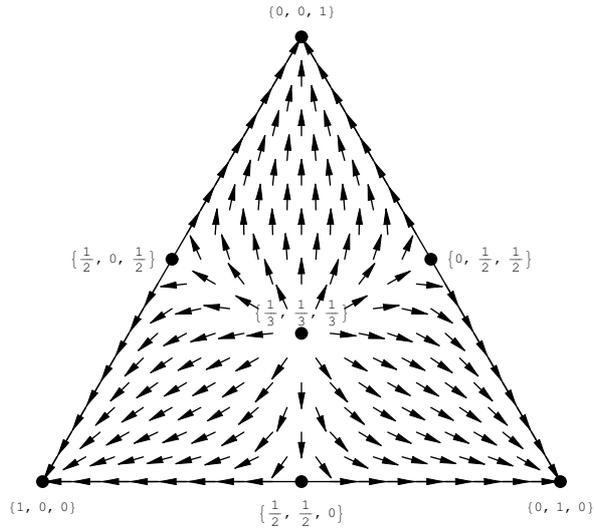


Figure 1: The vector field generated by ψ as defined in (28) on a two-dimensional simplex. Only the three corners are stable fixed points. The center point and the points in the middle of the edges are unstable.

meanings and words. These strengths lie between 0 and 1 and are updated as follows: when an agent, as a hearer, observes the meaning m_i expressed with word w_j , he updates his lexical matrix as follows:

$$\begin{pmatrix} & & & -\gamma & & & \\ & & & \vdots & & & \\ & & & -\gamma & & & \\ -\gamma & \cdots & -\gamma & \gamma & -\gamma & \cdots & -\gamma \\ & & & -\gamma & & & \\ & & & \vdots & & & \\ & & & -\gamma & & & \end{pmatrix}$$

with the only positive γ in the i^{th} row and j^{th} column. The values in this matrix are added to the lexical matrix (but all values are kept between 0 and 1), such that the strength of the used association is enforced and strengths of competing synonyms (on the same row) and homonyms (on the same column) are inhibited.

When expressing meaning m_i , the agent produces the word with the highest strength in row i . If there are multiple candidates one is selected at random. The interpretation of a word is analogous. While we do not have to consider this interpretation behavior in the fixed point analysis, we still need to verify that the convergence of the production behavior implies the convergence of the interpretation behavior. In this example this is the case because the lexical matrix of an agent which is subject to an optimal production behavior for some time will become a permutation matrix which generates the corresponding optimal interpretation behavior.

Again we are dealing with an ergodic agent such that the agent response function ψ is well-defined. Now consider the particular case of three meanings and three words, with $\gamma = 0.1$. As an example of an agent response we have (using the method (6)):

$$P = \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.5 & 0.25 & 0.25 \\ 0.25 & 0.5 & 0.25 \end{pmatrix} \quad \psi(P) \approx \begin{pmatrix} 0.37 & 0.23 & 0.4 \\ 0.67 & 0.13 & 0.2 \\ 0.11 & 0.76 & 0.13 \end{pmatrix} \quad (29)$$

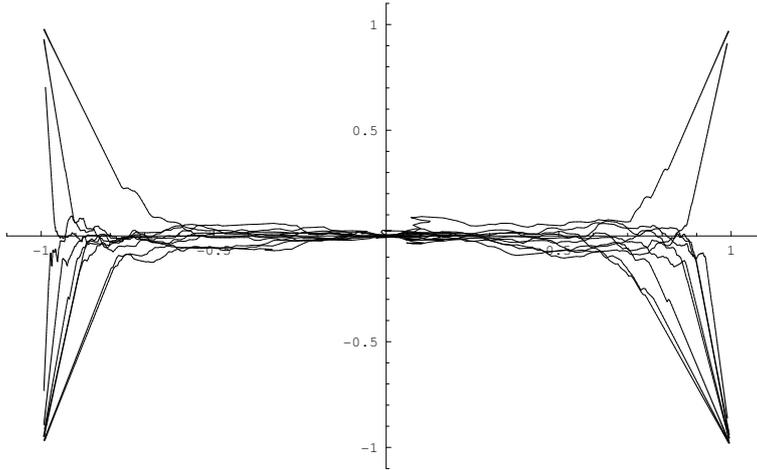


Figure 2: The escape from the unstable fixed point $P^{(3)}$ using the difference equations (16) and (17), with $\beta = 0.001$ and $b_0 = P^{(3)}$. The horizontal axis shows the difference between the initially equal values $(b_i)_{3,1}$ and $(b_i)_{3,2}$, the vertical axis shows the difference between the initially equal values $(b_i)_{1,3}$ and $(b_i)_{2,3}$. 20 trajectories are shown for i up to 5000.

in which we can observe an amplification of the values $P_{2,1}$ and $P_{3,2}$ and the beginning of the resolution of the conflict between the m_1 and m_2 through the increase of $P_{1,3}$.

The production matrices which associate each meaning with a different word are obviously stable fixed points. We will not exhaustively analyze all other possible fixed points, but will focus on one particular symmetry where the production matrix is of the form:

$$\begin{pmatrix} a & a & b \\ a & a & b \\ c & c & d \end{pmatrix} \quad (30)$$

with $2a + b = 1$ and $2c + d = 1$. There turn out to be three fixed points of this form:⁵

$$P^{(1)} = \begin{pmatrix} 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \\ 1/3 & 1/3 & 1/3 \end{pmatrix} \quad P^{(2)} = \begin{pmatrix} 0.5^- & 0.5^- & 0^+ \\ 0.5^- & 0.5^- & 0^+ \\ 0^+ & 0^+ & 1^- \end{pmatrix} \quad P^{(3)} = \begin{pmatrix} 0^+ & 0^+ & 1^- \\ 0^+ & 0^+ & 1^- \\ 0.5^- & 0.5^- & 0^+ \end{pmatrix} \quad (31)$$

$P^{(1)}$ is the fixed point which is necessarily present in all agents with no prior preference in meanings and words. All three fixed points turn out to be unstable. For example in figure 2 a projection of the evolution of the system is shown using equations (16) and (17), starting from the fixed point $P^{(3)}$. All trajectories escape from the suboptimal, unstable equilibrium point and converge toward an optimal equilibrium point.

Example 3

We now turn our attention to agents which update their state both as speaker and hearer. In other words, their state updates depend both on the production and interpretation behavior of the population. Several similar agent architectures of this type were introduced in the literature, e.g. in [17, 5, 11]. We consider the particular case of [11], whereby an agent makes use of a lexical matrix and produces and interprets as in

⁵The +/- signs in $P^{(2)}$ and $P^{(3)}$ indicate that the real fixed points are slightly shifted in that direction.

the previous example. The way in which this lexical matrix is updated however depends on the role of the agent in the game and also of the success or failure of the game. Assume for simplicity an equal number n of words and meanings: $n = \#M = \#W$. Consider a game where the speaker has to express meaning m_i and therefore uses word w_j , which the hearer interprets again as meaning $m_{i'}$. Only if $m_i = m_{i'}$ the game is successful. In this case the speaker increases the score of the used association (m_i, w_j) with γ and decreases the scores of competing words for the same meaning with $\frac{\gamma}{n-1}$. The hearer also increases the score of $(m_{i'}, w_j) = (m_i, w_j)$ with γ and decreases competing associations of the same word with other meanings with $\frac{\gamma}{n-1}$. If the game fails, the opposite changes are performed, i.e. the speaker decreases (m_i, w_j) and increases the strengths of synonyms, the hearer decreases $(m_{i'}, w_j)$ and increases the strengths of homonyms. The association strengths are always kept between 0 and 1. Note that in this model, when the game fails, the hearer does not take into account the intended meaning m_i to update its state.

In the calculation of the agent response function using (6), the agent participates in successive games as speaker and hearer with equal chance. We now consider again the case of three meanings and three words. As an example of an agent response we have, with $\gamma = 0.1$ and

$$b = \left\langle \left(\begin{array}{ccc} 0.6 & 0.2 & 0.2 \\ 0.25 & 0.5 & 0.25 \\ 0.3 & 0.3 & 0.4 \end{array} \right), \left(\begin{array}{ccc} 0.15 & 0.1 & 0.6 \\ 0.7 & 0.1 & 0.2 \\ 0.15 & 0.8 & 0.2 \end{array} \right) \right\rangle \quad (32)$$

that

$$\psi(b) \approx \left\langle \left(\begin{array}{ccc} 0.37 & 0.29 & 0.34 \\ 0.68 & 0.18 & 0.14 \\ 0.03 & 0.94 & 0.03 \end{array} \right), \left(\begin{array}{ccc} 0.44 & 0.11 & 0.71 \\ 0.53 & 0.05 & 0.18 \\ 0.03 & 0.84 & 0.11 \end{array} \right) \right\rangle \quad (33)$$

One can see that the different associations compete with each other in a non-trivial way.

Concerning the fixed points of the agent response function, we search again for behaviors with particular symmetries, namely behaviors of the form

$$\left\langle \left(\begin{array}{ccc} a & a & b \\ a & a & b \\ c & c & d \end{array} \right), \left(\begin{array}{ccc} e & e & f \\ e & e & f \\ g & g & h \end{array} \right) \right\rangle \quad (34)$$

with $2a + b = 1$, $2c + d = 1$, $2e + g = 1$ and $2f + h = 1$. Apart from the totally symmetrical behavior ($a = b = c = d = e = f = g = h = 1/3$), there are two other fixed points:

$$b^{(4)} = \left\langle \left(\begin{array}{ccc} 0.5^- & 0.5^- & 0^+ \\ 0.5^- & 0.5^- & 0^+ \\ 0^+ & 0^+ & 1^- \end{array} \right), \left(\begin{array}{ccc} 0.5^- & 0.5^- & 0^+ \\ 0.5^- & 0.5^- & 0^+ \\ 0^+ & 0^+ & 1^- \end{array} \right) \right\rangle \quad (35)$$

$$b^{(5)} = \left\langle \left(\begin{array}{ccc} 0^+ & 0^+ & 1^- \\ 0^+ & 0^+ & 1^- \\ 0.5^- & 0.5^- & 0^+ \end{array} \right), \left(\begin{array}{ccc} 0^+ & 0^+ & 0.5^- \\ 0^+ & 0^+ & 0.5^- \\ 1^- & 1^- & 0^+ \end{array} \right) \right\rangle \quad (36)$$

Again analysis using equations (16) and (17) showed that both fixed points are unstable.

Example 4

Finally we consider an agent architecture which has been described in [6] and later on in [15] and was dubbed ‘calculator’ and ‘obverter’ respectively. The main idea is that an agent calculates the best response to an estimate of the production and interpretation behavior of the population. More precisely, suppose the population has the behavior $\langle P, Q \rangle$ and the agent has an estimate of this $\langle \hat{P}, \hat{Q} \rangle$. The agent then constructs a production matrix P' based on \hat{Q} and an interpretation matrix Q' based on \hat{P} such that $\text{ca}(P', \hat{Q})$ and $\text{ca}(\hat{P}, Q')$ are maximal. For example we have

$$\hat{Q} = \begin{pmatrix} 0.2 & 0.4 & 0.5 \\ 0.3 & 0.2 & 0.2 \\ 0.5 & 0.4 & 0.3 \end{pmatrix} \quad P' = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad (37)$$

In [6] and [15] the matrices \hat{P} and \hat{Q} are constructed from a fixed number of samples of P and Q . In order to fit this into the language game framework, we let an agent remember the last k words used for each meaning and for each word its last k interpretations. These samples determine the estimates \hat{P} and \hat{Q} .

With regard to the agent response function, under the same symmetry restrictions as in (34), we observe approximately the same fixed points $b^{(4)}$ and $b^{(5)}$. But, unlike in the previous example, the equilibrium behavior $b^{(5)}$ turns out to be stable for larger values of k . To proof this we performed a linear stability analysis for different values of k . For each k , we first computed the agent response function ψ analytically. We then determined the exact fixed point, say b^* of ψ resembling $b^{(5)}$. Finally we calculated the jacobian J_ψ of ψ in b^* and examined its eigenvalues. As a behavior $b = \langle P, Q \rangle$ has only got 12 degrees of freedom, it suffices to calculate the jacobian of the function $\psi' : \mathbb{R}^{12} \rightarrow \mathbb{R}^{12}$ which indexes a behavior as follows.

$$\left\langle \left(\begin{array}{ccc} 1 & 2 & \cdot \\ 3 & 4 & \cdot \\ 5 & 6 & \cdot \end{array} \right), \left(\begin{array}{ccc} 7 & 9 & 11 \\ 8 & 10 & 12 \\ \cdot & \cdot & \cdot \end{array} \right) \right\rangle \quad (38)$$

For example, for $k = 7$, we have:

$$b^* = \left\langle \left(\begin{array}{ccc} 0.0033 & 0.0033 & 0.9934 \\ 0.0033 & 0.0033 & 0.9934 \\ 0.5 & 0.5 & 0 \end{array} \right), \left(\begin{array}{ccc} 0.0033 & 0.0033 & 0.5 \\ 0.0033 & 0.0033 & 0.5 \\ 0.9934 & 0.9934 & 0 \end{array} \right) \right\rangle, \quad (39)$$

and $J_\psi = \left(\begin{array}{c|c} 0 & J_\psi^* \\ \hline J_\psi^* & 0 \end{array} \right)$ with

$$J_\psi^* = \begin{pmatrix} 0.237 & 0. & -0.02 & 0. & -0.044 & 0. \\ -0.02 & 0. & 0.237 & 0. & -0.044 & 0. \\ 0. & 0.237 & 0. & -0.02 & 0. & -0.044 \\ 0. & -0.02 & 0. & 0.237 & 0. & -0.044 \\ -3.366 & -3.366 & 3.366 & 3.366 & 0. & 0. \\ 3.366 & 3.366 & -3.366 & -3.366 & 0. & 0. \end{pmatrix} \quad (40)$$

This J_ψ has the eigenvalues $\{\lambda_i\} = \{0.803, -0.803, 0.803, -0.803, -0.257, 0.257, -0.217, 0.217, 0, 0, 0, 0\}$ with $\text{Re}(\lambda_i) < 1$ but also $|\lambda_i| < 1$ such that both the systems (19) and (21) are stable.

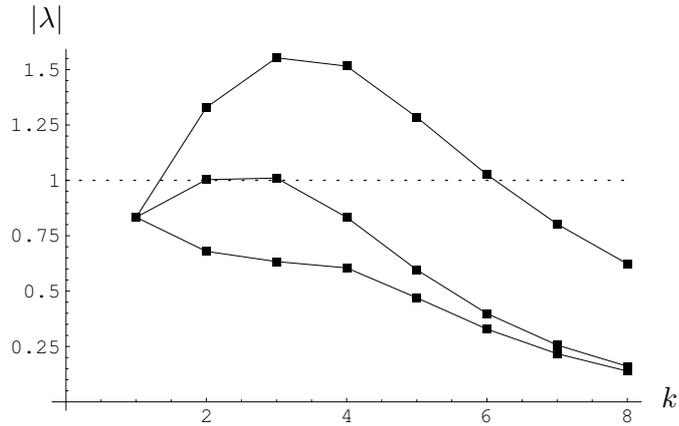


Figure 3: The three largest magnitudes of the eigenvalues of J_ψ as a function of the buffer length k .

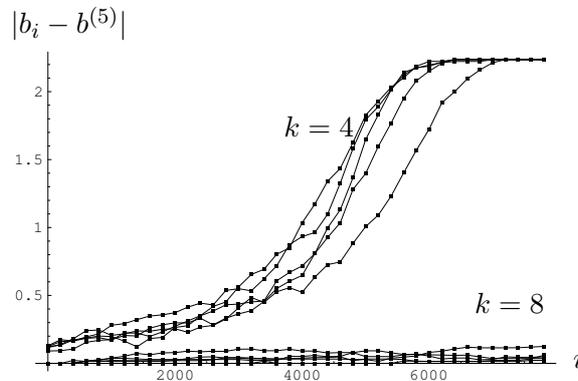


Figure 4: The evolution of the euclidean distance between the population behavior b_i and the fixed point $b^{(5)}$ for different buffer lengths k .

In figure 3 the three largest magnitudes of the eigenvalues of the jacobian are shown for increasing values of k . One can see that for $k \geq 7$ all eigenvalues lie within the unit circle such that both the systems (19) and (21) are stable in the fixed point $b^{(5)}$.

To verify that this stability also holds in the original system we ran a simulation with a population of 50 agents. Before the experiment we let each agent interact with $b^{(5)}$ for some time, such that the initial population behavior b_0 approximated $b^{(5)}$. Then the agents successively played language games and the evolution of the population behavior b_i was monitored. This was done for $k = 8$, a case for which we expect stability, but also for the instable case $k = 4$ as a reference experiment. In figure 4 the evolution of the distance⁶ between b_i and $b^{(5)}$ is shown for 5 independent runs in each case. The graphs suggest that there is indeed a fundamental difference between the two different cases. With $k = 4$, b_i moves away rapidly from $b^{(5)}$ and converges to an optimal behavior.⁷ If $k = 8$ this is not the case.

⁶The distance between two behaviors is the euclidean distance between the vectors composed of all elements of the production and interpretation matrix.

⁷For each optimal behavior $b^{\text{opt}} = \langle P^{\text{opt}}, Q^{\text{opt}} \rangle$ reached, we had $P_{3,3}^{\text{opt}} = Q_{3,3}^{\text{opt}} = 0$ which implies $|b^{\text{opt}} - b^{(5)}| = \sqrt{5} \approx 2.24$.

5 Conclusion

In this paper we presented a framework for analyzing the development of a coordinated communication system in a population of agents. This framework does not rely on any specific type of agent and can therefore be used to investigate and compare different agent models. The most important concept introduced, was the agent response function which captures the long term behavior of an agent when interacting with a population exhibiting a fixed language. From the original system of randomly interacting agents we derived a reduced, deterministic system in terms of this response function. We then argued that a stability analysis of the fixed points of this reduced model ties in with the ability of the original system to converge to an optimal communication system. More precisely we argued that, with respect to convergence, it is preferable that fixed points corresponding to suboptimal communication systems are unstable.

This framework was validated by applying it to several existing models from the literature. In most cases, our analysis indeed confirmed the results obtained in the publications. In one example however, a suboptimal but stable fixed point was found, which indicated a problem concerning the convergence to an optimal communication system. Further examination indeed showed that also in the original system, a population with an initial communication system near this point has severe difficulties to escape from the suboptimal state.

References

- [1] Dorothy L. Cheney and Robert M. Seyfarth. *How monkeys see the world: Inside the mind of another species*. University of Chicago Press, 1990.
- [2] Felipe Cucker, Steve Smale, and Ding-Xuan Zhou. Modeling language evolution. *Foundations of Computational Mathematics*, 4(3):315–343, July 2004.
- [3] E. D. de Jong. Attractors in the development of communication. In D. Floreano H. Roitblat J.-A. Meyer, A. Berthoz and S. Wilson, editors, *SAB00*, Honolulu, Hawaii, 2000. MIT Press.
- [4] Edwin D. De Jong. *Autonomous Formation of Concepts and Communication*. PhD thesis, Vrije Universiteit Brussel AI-lab, 2000.
- [5] Edwin D. de Jong and Luc Steels. A distributed learning algorithm for communication development. *Complex Systems*, 14(4), 2003.
- [6] James R. Hurford. Biological evolution of the Saussurean sign as a component of the language acquisition device. *Lingua*, 77(2):187–222, 1989.
- [7] F. Kaplan. Simple models of distributed co-ordination. *Connection Science*, 17(3-4):249–270, 2005.
- [8] S. Kirby and J. Hurford. The emergence of linguistic structure: An overview of the iterated learning model. In Angelo Cangelosi and Domenico Parisi, editors, *Simulating the Evolution of Language*, chapter 6, pages 121–148. Springer Verlag, London, 2002.
- [9] N. L. Komarova and M. A. Nowak. Language dynamics in finite populations. *Journal of Theoretical Biology*, 221(3):445–457, 2003.
- [10] Natalia L. Komarova and Martin A. Nowak. The evolutionary dynamics of the lexical matrix. *Bull. Math. Biol.*, 63(3):451–485, 2001.
- [11] Tom Lenaerts, Bart Jansen, Karl Tuyls, and Bart de Vylder. The evolutionary language game: An orthogonal approach. *Journal of Theoretical Biology*, 235:566–582, 2005.
- [12] M. A. Nowak, J. B. Plotkin, and D. Krakauer. The evolutionary language game. *Journal of Theoretical Biology*, 200(2):147–162, 1999.
- [13] Martin A. Nowak, Natalia L. Komarova, and Partha Niyogi. Evolution of universal grammar. *Science*, 291:114–118, 2001.
- [14] Martin A. Nowak and David C. Krakauer. The evolution of language. *Proc. Nat. Acad. Sci. USA*, 96:8028–8033, 1999.

- [15] Michael Oliphant and John Batali. Learning and the emergence of coordinated communication. *The newsletter of the Center of Research in Language*, 11(1), 1997.
- [16] K. Smith. The evolution of vocabulary. *Journal of Theoretical Biology*, 228(1):127–142, 2004.
- [17] L. Steels. A self-organizing spatial vocabulary. *Artificial Life Journal*, 2(3), 1996.
- [18] Luc Steels. The origins of ontologies and communication conventions in multi-agent systems. *Autonomous Agents and Multi-Agent Systems*, 1(2):169–194, 1998.
- [19] Luc Steels. Evolving grounded communication for robots. *Trends in Cognitive Sciences*, 7(7):308–312, 7 2003.
- [20] Luc Steels and Frédérique Kaplan. Bootstrapping grounded word semantics. In T. Briscoe, editor, *Linguistic evolution through language acquisition: formal and computational models*, pages 53–73. Cambridge University Press, Cambridge, 1999.
- [21] Bart De Vylder and Karl Tuyls. How to reach linguistic consensus: A proof of convergence for the naming game. *Journal of Theoretical Biology*, submitted, 2006.

All Roads Lead to Fault Diagnosis: Model-Based Reasoning with LYDIA

Alexander Feldman Jurryt Pietersma Arjan van Gemund

Delft University of Technology
Faculty of Electrical Engineering, Mathematics and Computer Science
Mekelweg 4, 2628 CD, Delft, The Netherlands
Tel.: +31 15 2781935, Fax: +31 15 2786632
e-mail: {a.b.feldman,j.pietersma,a.j.c.vangemund}@tudelft.nl

Abstract

Model-Based Reasoning (MBR) over qualitative models of complex, real-world systems has proven successful for automated fault diagnosis, control, and repair. Expressing a system under diagnosis in a formal model and inferring a diagnosis given observations are both challenging problems. In this paper we address these challenges. By building a fault model of a real-world artifact (the fuel-system of a light aircraft), we introduce the software package for MBR LYDIA and show its applicability in practice. We demonstrate how structure exploitation and compilation can be used to attack the main challenge to MBR - its high computational cost. Last, we compare our approach to other state-of-the art techniques for MBR and analyze its performance.

1 Introduction

Automated reasoning over qualitative models has proven successful for a wide range of diagnostic, control and repair tasks, involving wafer scanners, interplanetary space-probes, and large radio-telescopes [11]. Motivated by the success of systems like Livingstone [14], we have embarked on the design and implementation of a system of our own, named LYDIA¹. Comprising a *modeling language* and a set of translation and reasoning tools, LYDIA aims at improving the state-of-the-art in MBR by providing more *powerful* modeling primitives and *faster* diagnostic engines.

Recently there has been much interest in MBR and especially in model-based diagnosis (MBD) [7]. MBR involves two major challenges: (i) expressing the system under diagnosis in terms of a formal language, and (ii) performing the diagnostic inference given observations. In this paper we outline how the LYDIA approach addresses both challenges.

The biggest problem MBR is facing is its high computational cost. In particular, [17] shows that the time complexity of MBD related to the number of components comprising a system is in Σ_2P even in some very restricted cases. Researchers are looking for solution to this problem in two directions. One is exploiting structure which is present in any man-made system. Another, related approach is to compile the model to a form allowing faster, in the strict case polynomial-time, reasoning.

One of the most successful approaches to compilation is the work of Darwiche [3] in Decomposable Negation Normal Forms (DNNF). The latter work focuses on the algorithmic and computational aspects of propositional reasoning while our goal is to bridge the modeling, reasoning, and validation aspects of the whole concept. Similar to SAT, in diagnostic search, conflicts are an important source of speedup. The use of conflicts in MBR is one of the contributions in [18]. Exploiting hierarchy is another potential source of speedup. This can be used both in compilation-based approaches [2, 9] or directly [16].

¹Language for sYstem DIagnosis. The LYDIA package for model-based fault reasoning can be downloaded from <http://fdir.org/lydia/>.

One of the most comprehensive experiments involving model-based reasoning is the Deep Space 1 NASA mission [15], the latter aiming amongst others to test software components designated as strategic by NASA. The use of reasoning software for craft control, task planning, fault identification and recovery has resulted in reduced costs for mission execution as well as cheaper *design* of spacecraft systems which itself has contributed to the improved reliability throughout the whole mission.

The rest of this paper is organized as follows. In Section 2 we illustrate the main LYDIA language primitives by modeling the fuel system of a light aircraft. The workings of the diagnosis engine are discussed in the following Section 3. Experimental results and performance characteristics of our approach are shown in Section 4. Finally, conclusions and future work are discussed.

2 Modeling in Lydia

LYDIA is declarative modeling language specifically developed for Model-Based Diagnosis. The language core is propositional logic, enhanced with a number of syntactic extensions for ease of modeling. The accompanying toolset currently comprises a number of diagnostic engines and a simulator tool.

As a modeling example we use the fuel system of the Piper PA-28-161 light aircraft. Figure 1 shows its schematics. The composition of the model is determined by the level of detail we need to obtain in a system diagnosis. We assume that identifying one or more of the field-replacable components in Figure 1 is adequate for this purpose, therefore this schematic dictates the topology of the model.

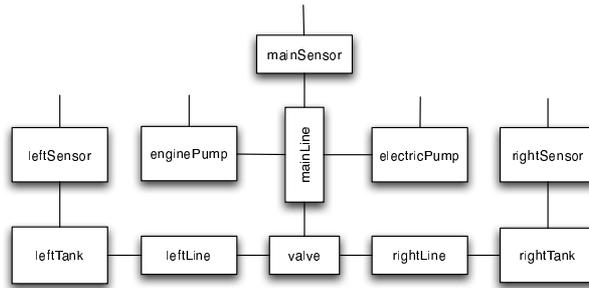


Figure 1: Schematics of the fuel system of Piper PA-28-161 light aircraft.

Next, we need to determine the proper system variables to be modeled. For this, the main functionality of the system is used as guidance. The main function of the fuel system is to provide an uninterrupted supply of fuel to the engine. This supply should have a specific quantity and should be controllable by the pilot. Hence, we choose fuel mass and its derivative fuel mass flow, as leading system variables. These parameters are influenced by all system components. This influence is partially controllable and measurable.

The next listing shows parts of the LYDIA source for our model. The keyword `system` indicates the definition of a component. Consider, for example, the system `fuelTank`. This component has three variables of type `mass`, defined with the keyword `type`. The health of this component is of type `hContain`. Health variables (which are to be solved by the Lydia diagnostic engine) in LYDIA are declared by the attribute `health`. An a priori probability is assigned as search heuristic for the diagnostic inference.

```

type mass = enum {zero, low, nom, high};
type hContain = enum { nom, leak, unknown };
type selectorPosition = enum { off, left, right };

system fuelTank(mass quantity, flow, lineFlow)
{
  hContain h;
  attribute health (h) = true;
  attribute probability (h) = cond (h) (hContain.nom -> 0.99; hContain.leak -> 0.009; ...);
}
  
```

```

(h = hContain.nom) => (((quantity = mass.zero) => (flow = mass.zero)) and (lineFlow = flow));
if (h = hContain.leak) {
  (quantity=mass.zero) => (flow=mass.zero);
  lineFlow = cond (flow) (mass.high -> mass.nom; mass.nom -> mass.low; mass.low -> mass.zero);
}
}

/* Definition of other components... */

system fuelSystem(selectorPosition selector, bool enginePumpOn, electricPumpOn, mass leftSensorMass,
  leftSensorFlow, rightSensorMass, rightSensorFlow, mainSensorFlowOut)
{
  mass leftMass, leftFlow, rightMass, rightFlow, leftLineFlow, rightLineFlow, mainFlow;
  mass leftLineFlowOut, rightLineFlowOut, mainFlowOut, mainMassOut, mainSensorMassOut;

  attribute observable(selector, enginePumpOn, electricPumpOn, leftSensorMass, leftSensorFlow) = true;
  attribute observable(rightSensorMass, rightSensorFlow, mainSensorFlowOut) = true;

  system fuelTank leftTank(leftMass, leftFlow, leftLineFlow);
  system fuelTank rightTank(rightMass, rightFlow, rightLineFlow);
  system sensor leftSensor(leftMass, leftFlow, leftSensorMass, leftSensorFlow);
  system sensor rightSensor(rightMass, rightFlow, rightSensorMass, rightSensorFlow);
  system line leftLine(leftLineFlow, leftLineFlowOut), rightLine(rightLineFlow, rightLineFlowOut);
  system selectorValve valve(selector, leftLineFlowOut, rightLineFlowOut, mainFlow);
  system pump enginePump(enginePumpOn, mainFlow), electricPump(electricPumpOn, mainFlow);
  system line mainLine(mainFlow, mainFlowOut);
  system sensor mainSensor(mainMassOut, mainFlowOut, mainSensorMassOut, mainSensorFlowOut);
}

```

For this particular component nominal (healthy) behavior is defined as having zero flow in case of an empty tank, i.e., quantity is zero, and the tank flow being equal to the flow in the tank line. This (weak) model would already be sufficient for diagnosis but for better diagnostic results we choose to extend it with an explicit failure mode. For a tank an obvious failure mode is a leak, in which case the line flow is less than the tank flow as some of the fuel also escapes through the leak. This is implemented with the health mode `hContain.leak`.

The behavior of the other components is defined in a similar fashion. All components are used in the top-level system model `fuelSystem`. This is where the components are instantiated and the model topology is created by sharing the variables between components. The main characteristic that makes diagnosis a non-trivial problem is limited observability. Not all physical system variables are observable. For this particular system the fuel mass and flow are only observable at sensor locations. Observable variables are identified with the attribute `observable`.

Besides the sensor values², the control variables are observable by definition. For this model there are three control variables. One controls the valve. It can be used to select which, if any, tank is used. This is implemented with the `selectorPosition` type. Two control variables are used to control both pumps, one of which is driven by the aircraft's engine the other by a separate electrical engine.

We have considered two failure scenarios. First the left tank is selected and the left sensor indicates a high flow, while other observations are nominal. As shown below, LYDIA infers that, based on the model and these observations, the most likely root causes are either a leak in the tank or a leak in the left line. Other (less likely) single faults are unknown faults in the tank, fuel line, or valve. These are followed by multiple failures that are even less likely (not shown). In a second scenario (also not shown), we keep the left tank selected and now the right sensor indicates a low flow, while other observations are nominal. LYDIA now places the root cause is on the right side of the aircraft, either in the tank or the line.

Below, we show the results from the first experiment in the transcript of the LYDIA sdNNF solver (for details on its workings cf. Section 3). We can see the assigned values to all observable variables and the six leading diagnoses. Note, that with weak-fault models the number of diagnoses is exponential to the number of non-failed components in the minimal-cardinality fault as each healthy component can be faulty, but still producing nominal output. Hence, in the scenario below

²In the real system, the main sensor is a pressure sensor and only measures flow. Hence the use of mass dummy variables.

the LYDIA solver asserts the faulty components (in our case a leaking left fuel tank) and does not commit on the state of the remaining variables.

```

@ start output <state>
observable enum leftSensorMass = nom
observable enum leftSensorFlow = high
observable enum rightSensorMass = nom
observable enum rightSensorFlow = zero
observable enum selector = left
observable enum enginePumpOn = true
observable enum electricPumpOn = false
observable enum mainSensorFlowOut = nom
@ stop output <state>
@ start output <fm>
(7.35542e-05) leftTank.h = leak, leftSensor.h = nom, leftLine.h = nom, valve.h = nom, ...
(7.35542e-05) leftTank.h = nom, leftSensor.h = nom, leftLine.h = leak, valve.h = nom, ...
(8.10724e-06) leftTank.h = nom, leftSensor.h = nom, leftLine.h = nom, valve.h = unknown, ...
(8.10724e-06) leftTank.h = nom, leftSensor.h = nom, leftLine.h = unknown, valve.h = nom, ...
(8.10724e-06) leftTank.h = nom, leftSensor.h = unknown, leftLine.h = nom, valve.h = nom, ...
(8.10724e-06) leftTank.h = unknown, leftSensor.h = nom, leftLine.h = nom, valve.h = nom, ...
...
@ stop output <fm>

```

10

From the above output, a LYDIA user can derive the health of the system. In reality, LYDIA is supplied with well-defined API for inclusion in higher-level frameworks, where diagnosis is a part of a *reactive loop* of fault monitoring, isolation, reconfiguration and recovery. In the next section we will look more closely on the computational aspects of LYDIA.

3 Computing Diagnosis with Lydia

In Section 2 we have built a LYDIA model. Our next goal is to provide an algorithm for efficient, sound and complete MBD, a core part of the LYDIA toolkit. We will discuss how a model description *SD* and an *observation OBS* (the latter is simply a run-time valuation over a set of variables designated as observable) allow us to compute *diagnosis* of our system. To facilitate the presentation we will formalize the definitions of system and diagnosis.

Definition 1 (System). A diagnostic problem *DP* is the ordered triple $DP = \langle SD, COMPS, OBS \rangle$, where *SD* is a set of propositional sentences describing the behavior of the system, *COMPS* is a set of components, contained in the system, and *OBS* is a term stating an observation over some set of “measurable” variables in *SD*.

In this approach for each component $c \in COMPS$ there is a corresponding propositional variable h_c representing its health state. We will call these variables h_c *health variables* and every instantiation of $\bigwedge_{c \in COMPS} h_c$ a *health state*.

Definition 2 (Diagnosis). A *diagnosis*³ for the system $DP = \langle SD, COMPS, OBS \rangle$ is a set $D \subseteq COMPS$ such that $SD \wedge OBS \wedge [\bigwedge_{c \in D} \neg h_c] \wedge [\bigwedge_{c \in (COMPS \setminus D)} h_c] \not\models \perp$.

A diagnosis *D* is a *minimal* if no other diagnosis D' , such that $D' \subset D$, exists. A *partial diagnosis* *P* is such a conjunction of health literals h_c or $\neg h_c$, $c \in COMPS$, that for every other conjunction ϕ which contains *P* it follows that $SD \wedge OBS \wedge \phi \not\models \perp$. Similarly, a *kernel diagnosis* is a partial diagnosis which is not contained in any other partial diagnosis [6].

From the above two definitions, it is visible that in order to perform diagnosis on a LYDIA model it is enough to translate it to a well-formed propositional formula (**Wff**) and to use an entailment mechanism for finding those diagnoses *D* which are consistent with $SD \wedge OBS$, that is *D* explain $SD \wedge OBS$. The LYDIA language is designed in such a way as to facilitate a conversion to a propositional **Wff** in polynomial time. This includes the normal language parsing, type-checking (LYDIA is a type-strict language), expanding arrays and array quantifiers and processing of variable attributes.

³Throughout this paper we consider consistency-based diagnosis as opposed to abductive diagnosis.

The LYDIA language supports variables both in the Boolean and finite integer (FI) domains. The LYDIA toolkit proposes two approaches for unifying this – encoding FI variables as Booleans and vice-versa. Encoding FI into Boolean is trivial and we will not discuss it for brevity. Working directly in the FI domain is a preferred option and below we introduce a multivalued representation very-close to the traditional Boolean one and suitable to conventional propositional algorithms (e.g., DPLL). The definitions explaining the use of multi-value logic in LYDIA follow.

Definition 3 (Multi-Valued Literal). A multi-valued variable $v_i \in V$ takes a value from a finite domain, which is an integer set $D_i = \{1, 2, \dots, m\}$. A positive multi-valued literal l_j^+ is a Boolean function $l_j^+ \equiv (v_i = d_k)$, where $v_i \in V, d_k \in D_i$.

Similarly, we introduce negative multi-valued literals. If not specified, a literal l_j can be either positive or negative.

Definition 4 (Multi-Valued Propositional Wff). A multi-valued propositional **Wff** is a formula over the multi-valued literals l_1, l_2, \dots, l_n , and the standard Boolean connectives $\neg, \Leftrightarrow, \Rightarrow, \wedge, \vee$.

Up until now, we have discussed the compilation of the original LYDIA model to a Boolean or multi-valued **Wff**. Actually, this is the conjunction of each system’s **Wff** as the actual hierarchy is preserved in this representation. This introduces the notion of *hierarchical system*, which will allow us to perform faster reasoning compared to algorithms working on “flat” representations⁴ [10].

Definition 5 (Hierarchical System). A hierarchical system is a rooted, edge-labeled, acyclic multidigraph $H = \langle V, \rho, E \rangle$, where every node $V_i, V_i \in V$, contains a knowledge base SD_i and a set of components $COMPS_i$. The multidigraph is such that $COMPS_1 \cap COMPS_2 \cap \dots \cap COMPS_n = \emptyset$. The root node is marked by ρ and the labels of the edges in E are maps $f : SD_i \rightarrow SD_j$ between the literals in the knowledge bases represented by the nodes V_i and V_j .

The introduction of hierarchical systems allows us to similarly define hierarchical CNF and hierarchical DNF (the latter is not DNF anymore but is a restricted form of Negation Normal Form). We call this hierarchical DNF semi-decomposable Negation Normal Form (sdNNF). A hierarchical system is simply a conjunction of Boolean or multi-valued **Wff**. It is possible to discard this information (i.e., to *flatten out* the hierarchy) and to continue transforming the **Wff** in its flat representation. A map showing the possible translations between a number of formats is shown in Figure 2.

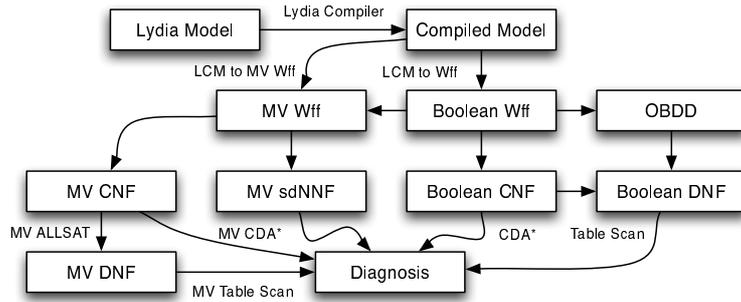


Figure 2: To compute diagnosis LYDIA translates the original model into a number of possible representation.

Instead of flattening a hierarchical system we will *selectively apply compilations on subsystems of SD*. The need to exploit hierarchy stems from the inherent high-computational price of MBR. By exploiting the hierarchical information and selectively compiling parts of the model it is possible to increase the diagnostic performance and to trade cheaper preprocessing time for faster run-time reasoning. Our hierarchical algorithm, being sound and complete, allows large models to be diagnosed, where compile-time investment directly translates to run-time speedup.

⁴For brevity, we refer to the classical diagnosis approach as “flat”, i.e., non-hierarchical.

Furthermore LYDIA *repartitions and coarsens* the original model in an attempt to minimize the subsystem connectivity, a process which leads to faster run-time fault diagnosis at the price of some pre-processing time. This is explained in the two algorithms below which show an advanced way for mixing compilation and hierarchical reasoning for fast diagnosis. The implementation of the two algorithms below is an important part of the LYDIA reasoning tool-kit.

The reason for the compilation map in Figure 2 is threefold. First, we need to reach logically equivalent representation which allows us to cross-validate the correctness of the LYDIA tools. The implementation of other state-of-the-art techniques (like CDA* [18]) allows us to verify the final diagnostic result and to compare the diagnostic performance under fair conditions. Finally, and most importantly, almost any translation causes combinatorial explosion with some models.

Models or submodels, depending on different characteristics (e.g., weak/strong modeling, etc.) can produce very different compilation results. More precisely, there are Boolean functions which have linear OBDD representation, but exponential irreducible CNF. Therefore it is beneficial to choose different representations for different models and parts of models. Full complexity analysis of all representations is impossible due to lack of space in this paper but is available in [4].

Algorithm 1 Compilation of a LYDIA Model to sdNNF.

```

1: function SDNNFFROMLYDIAMODEL( $L, K, M$ )
   inputs:  $L$ , a LYDIA model
              $K$ , number of partitions, integer
              $M$ , maximum DNF size, integer
   local variables:  $G$ , interaction graph
                      $P$ , graph partitioning a set of set of node indices
                      $N = \psi_1 \wedge \psi_2 \wedge \dots \wedge \psi_n$ , an NNF, a conjunction of Wff
                      $W$ , sdNNF, initially  $\emptyset$ 
2:  $N \leftarrow$  LYDIACOMPILE( $L$ ) ▷ Compilation, basic rewritings, etc.
3:  $G \leftarrow$  INTERACTIONGRAPH( $N$ ) ▷ A node for  $\psi_i$ , and an edge for  $\psi_j, \psi_k$  sharing a variable.
4:  $P \leftarrow$  PARTITIONGRAPH( $G, K$ ) ▷ Use, e.g., an approximate graph partitioning.
5: for all  $p \in P$  do
6:    $W \leftarrow W \wedge$  OBDDTODNF(NNFtoOBDD( $\bigwedge_{i \in p} \psi_i$ ))
7: end for
8: while  $\exists \gamma, \delta \in W : \text{COUNTSOLUTIONS}(\gamma, \delta) < M$  do
9:   SDNNFNODESMERGE( $W, \gamma, \delta$ ) ▷ Perform symbolic multiplication.
10: end while
11: return  $W$ 
12: end function

```

Algorithm 1 shows the compilation of a LYDIA model to sdNNF. First it converts the model L to a hierarchical multi-valued NNF. Let the NNF N be a conjunction of **Wff** $N = \psi_1 \wedge \psi_2 \wedge \dots \wedge \psi_n$. Then we build an interaction graph G in a fashion similar to [8] by having a node for each expression ψ_i in N and an edge in G if two expressions share a variable. We also weigh the edges of G with the number of variables two **Wff** ψ_i and ψ_j share.

Note that PARTITIONGRAPH is, itself, a computationally expensive process (it can be exponentially hard of the number of nodes in the graph G). LYDIA uses an approximation algorithm for graph partitioning [13] to solve the last problem. After the partitioning phase, we use an approximate model counter to merge nodes in the hierarchical description until any future merging would increase the number of terms in a node to a value exceeding a parameter M .

Next, we proceed with the run-time part, which is based on A*. We assume that components failures are independent and use the *a priori* probability of a fault term to guide a heuristic search for the most likely diagnosis. We assign the same small probability to all the components [5] as the reasoning technique is not probability-driven and it is possible to use other heuristics with similar results (e.g., the cardinality of a fault-mode).

Algorithm 2 computes diagnoses from the model produced by Algorithm 1 and OBS. The difference between the standard A* algorithm used in diagnosis, and this hierarchical version is that we try to find a conjunction of terms as opposed to conjunction of assumable variables consistent with OBS. The granularity of our approach is coarser and adjustable due to the parameterization of Algorithm 1 which allows trading space for time and in some cases reducing the overall computational complexity.

Algorithm 2 A* search in sdNNF.

```
1: procedure HIERARCHICALDIAGNOSIS( $H, OBS$ )
   inputs:  $H$ , root sdNNF node, each node is a disjunction of terms
            $OBS$ , an observation
   local variables:  $Q$ , priority queue
                      $s, c$ , terms
2:   PUSH( $Q, \text{INITIALSTATE}(H)$ ) ▷ Push the empty term on the queue.
3:   while ( $c \leftarrow \text{POP}(Q)$ )  $\neq \emptyset$  do ▷ Until all possible conjunction of terms are expanded.
4:     ENQUEUE SIBLINGS( $Q, c$ ) ▷ Push all paths from the root to terms in the same node as  $c$ .
5:     if DIAGNOSIS( $c \wedge OBS$ ) then ▷ True iff  $c$  contains a term from each node in  $H$ .
6:       OUTPUTDIAGNOSIS( $c$ )
7:     else
8:       if ( $s \leftarrow \text{NEXTBESTSTATE}(H, c)$ )  $\neq \perp$  then
9:         PUSH( $Q, s$ ) ▷ Choose the best state from the descendant of  $c$ .
10:      end if
11:    end if
12:  end while
13: end procedure
```

In this particular example Algorithm uses sdNNF but any hierarchical form with nodes consisting of tractable knowledge-bases will achieve similar results. In the main loop Algorithm 2 algorithm is chosen such a term c from the hierarchical node such that some heuristic estimate $f(c)$ is maximized. When a consistent conjunction of terms is found from all the nodes in the hierarchy, OUTPUTDIAGNOSIS is invoked to send the result to the user.

The auxiliary functions PUSH and POP perform the respective priority queue manipulation on Q (POP returns \emptyset if the queue is empty). The initial state in the search tree, returned by INITIALSTATE, is the empty term. The selection of the next candidate states to be added to the search queue is done by the functions NEXTBESTSTATE and ENQUEUE SIBLINGS. The former chooses the child state of the current state c and uses this term s from it, which again maximizes a utility function $f(s)$.

Algorithm 2 produces a diagnosis which completes the main goal of our exposition – to show a way for fast model-based fault diagnosis from modeling to the workings of a contemporary diagnostic search. Next we show some empirical result on the use of LYDIA.

4 Experimental Results

As the use of LYDIA in diagnosing models has already been demonstrated in Section 2, in this section we present experimental data on LYDIA’s inference performance.

We have derived a diagnosis benchmark from the ISCAS-85 set of combinatorial circuits [1]. Its basic characteristics are shown in Table 1. We have counted the number of variables $|SD|$ and the number of clauses in the CNF $|\Delta|$. The number of observable variables is denoted as $|OBS|$. The ISCAS-85 specification does not provide for a fault-modeling, hence we use our own standard logic component libraries allowing every gate-level component to fail. Throughout this paper we have used weak-fault models of the components.

In Table 1 we can see the compilation times for converting hierarchical CNF to hierarchical DNF and the time necessary for partial flattening. This time is denoted as T_c . The sum of the terms in each of the nodes of the hierarchical DNF is denoted as $|\phi_t|$. Note, that for the sdNNF we don’t have a partial flattening step, hence the compilation time is only the time for converting the hierarchical CNF to hierarchical DNF. The time for finding a leading single-fault diagnosis using Algorithm 2 is denoted as T_d . For a reference we have included the time for computing the same diagnosis using a flat A* solver and the results are in the T_f column.

By using Algorithm 1 for partial coarsening the sdNNF, we gain speedup by a factor varying from 2.9 to 12.3 (cf. Table 1) in comparison to the original uncoarsened sdNNF. For this improvement in speed we pay the price of increasing the representation size in comparison to the original sdNNF [12]. This increase in size is a factor of 5.9 – 49.7.

	Gates	$ SD $	$ \Delta $	$ OBS $	T_c [s]	$ \phi_t $	T_d [ms]	T_f [ms]
c432	146	328	486	43	1.48	10 579	10	200.32
c499	202	445	714	73	69.67	53 653	3.84	132.75
c1908	252	541	911	58	28.88	28 544	4.75	57.44
c880	383	826	1 112	86	2.28	29 305	4.51	373.23
c1355	514	1 069	1 546	73	68.57	43 677	16.36	793.2
c2670	983	2 153	2 856	226	3.43	36 817	111.47	12 676.65
c3540	1 297	2 685	3 861	72	7.27	82 808	187.32	11 038.84
c5315	2 202	4 796	6 983	295	6.13	73 576	2 796.49	53 387.5
c6288	2 416	4 864	7 216	64	4.07	74 014	123.38	67 506.23
c7552	3 024	6 390	9 085	325	93.11	213 382	746.95	133 234.74

Table 1: Compilation time, resulting sdNNF size and first single fault diagnosis search time for the ISCAS-85 benchmark suite models.

While model partitioning is a topic on its own, even these preliminary results suggest the existence of an optimal space/time trade-off which we intend to exploit in subsequent research. Experiments with hand-prepared hierarchies [11] show speedup growing faster than the model size and in the range of $10^2 - 10^5$.

5 Conclusion and Future Work

In this paper we presented the LYDIA language for model-based fault reasoning. We have illustrated the process of automatically computing a diagnosis by building a model of a fuel-system of a light aircraft. An example of diagnostic queries allowed the system to automatically compute the health state of the system. To alleviate the computational complexity of the diagnosis computation we have used algorithms implemented in the LYDIA toolkit. These algorithms allow compilation of the model to a form which facilitates faster automated reasoning.

We have shown a map of tools for model manipulation. Many ways for computing diagnosis are possible and this is due to the fact that depending on the model and the observation the solution landscape can expose very different search properties. The basic principles on which the LYDIA toolkit works are discussed in this paper and the detailed algorithms can be traced in the cited literature.

Automatic model partitioning, discovering of patterns in the model for faster reasoning, and more efficient compilation techniques are part of our plans future work. The current implementation of LYDIA allows reasoning over combinatorial models. Introducing time and state would increase the applicability of the technique while it will impose more challenges in comparing the algorithms with alternative techniques as the model-based community lacks established standards for representing dynamic systems. Studying the workings of model-based diagnosis techniques on a bigger set of benchmarks, including real-world problems would improve the reader in the benefits of automated reasoning.

Acknowledgments

We extend our gratitude to the anonymous reviewers for their insightful feedback.

References

- [1] Franc Brglez and Hideo Fujiwara. A neutral netlist of 10 combinational benchmark circuits and a target translator in fortran. In *Proc. ISCAS'85*, pages 695–698, 1985.
- [2] Adnan Darwiche. Model-based diagnosis using structured system descriptions. *JAIR*, 8:165–222, 1998.
- [3] Adnan Darwiche. New advances in compiling CNF into DNNF. In *Proc. ECAI'04*, pages 328–332, 2004.

- [4] Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002.
- [5] Johan de Kleer. Using crude probability estimates to guide diagnosis. *AI*, 45(3):381–291, 1990.
- [6] Johan de Kleer, Alan Mackworth, and Raymond Reiter. Characterizing diagnoses and systems. *Artificial Intelligence*, 56(2–3):197–222, 1992.
- [7] Johan de Kleer and Brian Williams. Diagnosing multiple faults. *JAI*, 32(1):97–130, 1987.
- [8] Rina Dechter. *Constraint Processing*. Morgan Kaufmann Publishers Inc., 2003.
- [9] Yousri El Fattah and Rina Dechter. Diagnosing tree-decomposable circuits. In *IJCAI'95*, pages 1742–1749, 1995.
- [10] Alexander Feldman and Arjan van Gemund. A two-step hierarchical algorithm for model-based diagnosis. In *Proc. AAAI'06*.
- [11] Alexander Feldman, Arjan van Gemund, and André Bos. A hybrid approach to hierarchical fault diagnosis. In *Proc. DX'05*, pages 101–106, 2005.
- [12] Mark Hansen, Hakan Yalcin, and John Hayes. Unveiling the ISCAS-85 benchmarks: A case study in reverse engineering. *IEEE Design & Test*, 16(3):72–80, 1999.
- [13] Burkhard Monien and Stefan Schamberger. Graph partitioning with the party library: Helpfulness in practice. *SBAC-PAD*, pages 1550–1533, 2004.
- [14] Nicola Muscettola, P. Pandurang Nayak, Barney Pell, and Brian C. Williams. Remote agent: To boldly go where no AI system has gone before. *AI*, 103(1-2):5–47, 1998.
- [15] Barney Pell, Douglas E. Bernard, Steve A. Chien, Erann Gat, Nicola Muscettola, P. Pandurang Nayak, Michael D. Wagner, and Brian C. Williams. An autonomous spacecraft agent prototype. In *Proc. AGENTS'97*, pages 253–261, 1997.
- [16] Gregory Provan. Hierarchical model-based diagnosis. In *Proc. DX'01*, 2001.
- [17] Farrokh Vatan. The complexity of the diagnosis problem. Technical Report NPO-30315, Jet Propulsion Laboratory, California Institute of Technology, 2002.
- [18] Brian Williams and Robert Ragno. Conflict-directed A^* and its role in model-based embedded systems. *Journal of Discrete Applied Mathematics*, 2004.

ON THE MODULARISATION OF INDEPENDENCE IN DYNAMIC BAYESIAN NETWORKS

Ildikó Flesch^a, Peter Lucas^a and Stefan Visscher^b

^aDepartment of Information and Knowledge Systems, ICIS,
Radboud University Nijmegen, The Netherlands

^bDepartment of Infectious Diseases
University Medical Center Utrecht, The Netherlands
Email: {ildiko,peter}@cs.kun.nl,S.Visscher@umcutrecht.nl

Abstract

Dynamic Bayesian networks are a special type of Bayesian networks, which explicitly deal with the dimension of time. They are distinguished into repetitive and non-repetitive networks. Repetitive networks have the same set of random (statistical) variables and independence relations at each time step, whereas in non-repetitive networks the set of random variables and the independence relations between these random variables may vary in time. Due to their structural symmetry, repetitive networks are easier to use and are, therefore, often taken as a standard. However, repetitiveness is a very strong assumption, which normally does not hold, since particular dependences and independences may only hold at certain time steps.

In this paper, we propose a new framework for the modularisation of non-repetitive dynamic Bayesian networks, which offers a practical approach to coping with the computational and structural difficulties associated with dynamic Bayesian networks. This framework is based on separating temporal and atemporal independence relations. We investigate properties of the modularisation and show the separation to be compositive.

1 Introduction

Probabilistic graphical models are increasingly adopted as tools for the modelling of domains involving uncertainty. For the development of practical applications, especially Bayesian networks have gained much popularity. When considering these application domains, it appears that so far limited attention has been given to the modelling of uncertain time-related phenomena, which occur in many of these domains. Bayesian networks in which some notion of time is explicitly dealt with are usually called *dynamic* Bayesian networks (DBNs) [4]. In some domains involving time, such as speech recognition, the use of DBNs has been extensively explored (e.g. [2]), and technical issues such as concerning reasoning (e.g. [6]) and learning (e.g. [3]) in DBNs have been investigated.

DBNs are distinguished into two main classes: repetitive and non-repetitive networks. Repetitive networks have the same set of random variables and independence relations at each time step, whereas in non-repetitive networks the set of random variables and the independence relations between these random variables may vary in time. The simpler structure of repetitive networks provides significant advantages in terms of computational complexity and ease of modelling. Therefore, they are often seen as the standard DBN model and they have been extensively explored (see [5] for an overview). However, repetitiveness is a very strong assumption that normally will not hold.

Recently, it was established that non-repetitive DBNs are practically useful [8]. Separating temporal and atemporal information in DBNs may be valuable, as it (i) helps experts gaining insight into the relations in the network, (ii) may help overcome computational limitations and (iii) provides an opportunity for learning procedures to obtain more accurate models. However, so far no research has been carried out to characterise temporal and atemporal independence relations.

In this paper, it is studied what happens with the represented Markov properties, and therefore also with the associated conditional independence assumptions, when we make an explicit

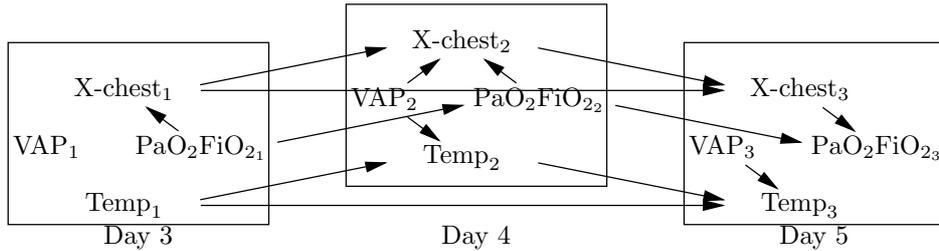


Figure 1: The non-repetitive dynamic Bayesian network for VAP.

distinction between temporal and atemporal structures. It is shown that this distinction allows decomposing the Markov properties into parts, such that the properties of the individual parts can be investigated separately. As we will see, these individual parts cannot be joined together to define the entire set of relations in the DBN without consideration of some significant properties. Therefore, the way temporal and atemporal parts of the Markov properties interact, is also studied. This yields an operator that joins atemporal and temporal relations together in a *correct* way. As considering the temporal and atemporal parts of a DBN, and their interaction, involves studying particular fragments of an independence relation, our research is related to work on multiply sectioned Bayesian networks [7] and multinet models [1]. Yet, the constraints imposed by the modelling of time in DBNs give rise to results which are nevertheless distinct.

This paper is organised as follows. In Section 2, we introduce a real-world medical example that illustrates the need of non-repetitive DBNs. In Section 3, some necessary concepts from graph theory as well as the basic principles of Bayesian networks are briefly reviewed. In Section 4, the definition of DBNs is introduced, which includes definitions for the ability to distinguish temporal and atemporal independence relations. Next, in Section 5, the join operator and its properties are introduced. Subsequently, in Section 6, the join operator will be applied to join atemporal and temporal relations to obtain the entire set of independence relations. Finally, in Section 7, we summarise the results that have been achieved.

2 Motivating example: the disease course of VAP

A real-world non-repetitive DBN of the disease course of a form of pneumonia is used as motivating example for the study of independence relations in non-repetitive DBNs. We briefly describe the clinical features of pneumonia and then discuss the construction of a DBN for this disease.

Pneumonia frequently develops in ICU patients, as these patients are critically ill and often do they need respiratory support by a mechanical ventilator. After admission to a hospital, all patients become colonised by bacteria. In particular, mechanically ventilated patients run the risk of subsequently developing pneumonia caused by these bacteria; this type of pneumonia is known as *ventilator-associated pneumonia*, or VAP for short. Typical signs and symptoms of VAP include: high *body temperature*, decreased lung function (measured by the $\text{PaO}_2/\text{FiO}_2$ ratio) and evidence of pneumonia on the chest *X-ray*. By carrying out a dependency analysis on a retrospective, temporal dataset, with data of ICU patients collected during a period of three years, we were able to study how independence information changed in the course of time. Taking the duration of mechanical ventilation as the parameter defining the time steps, taking into account knowledge of two infectious disease experts, we have focused on modelling the course of the development of VAP at 3, 4 and 5 days after admission. The resulting DBN is shown in Figure 1.

The dependence analysis for data on day 3 suggests that there is at that time no dependence between VAP and the signs and symptoms. However, a dependence between $\text{PaO}_2/\text{FiO}_2$ and chest X-ray was found, which seems logical: a pneumonia-affected lung, as demonstrated on the chest X-ray, will give rise to decreased lung function, measured by the $\text{PaO}_2/\text{FiO}_2$ ratio. These results are consistent with clinical evidence: after 3 days, VAP signs and symptoms become manifest. The dependence analysis for data on day 4 suggested, in addition, that as VAP develops (which is an infectious disease), body temperature is increased due to fever. Analysis of the data for day 5 shows again the relation between lung function ($\text{PaO}_2/\text{FiO}_2$) and chest X-ray. Apparently this is an important and strong relation. For day 5 the relation between VAP and X-ray of the chest

has disappeared. This is explained by noting that signs of the chest X-ray improve after start of treatment (a variable not modelled). The temporal arcs were subsequently learnt from the data taking the dependences just mentioned as a starting point. The result is a non-repetitive DBN.

3 Basic notions

We will be concerned in this paper with acyclic directed graphs (ADGs), denoted as a pair $G = (V, A)$, where V is a set of vertices and $A \subseteq V \times V$ is a set of *arcs*. A *directed path* is a sequence of vertices v_1, v_2, \dots, v_m , with $(v_k, v_{k+1}) \in A$ for each k , also denoted by $v_k \rightarrow v_{k+1}$, where v_1, v_2, \dots, v_{m-1} are required to be distinct. A *directed cycle* is a directed path with $v_1 = v_m$. A *trail* in a graph is a sequence of unique vertices v_1, v_2, \dots, v_m , where we have for each k that $v_k \rightarrow v_{k+1}$ or $v_{k+1} \rightarrow v_k$; each arc occurs only once. A *subtrail* of a trail v_1, v_2, \dots, v_m is a sequence v_i, v_{i+1}, \dots, v_j , $i < j$. A trail τ connecting vertices u and v is also written as $u \sim v$. The set of all trails of an ADG G is denoted by Θ . A graph $G'_{|\Theta'}$ $= (V', A')$ is called a *reduced subgraph* of graph $G = (V, A)$ with associated set of trails Θ' if $V' \subseteq V$ and A' consists of all arcs of the set of trails Θ' with $\Theta' \subseteq \Theta$.

Let X be a set of *discrete* random variables and let V act as its index set, i.e., X_v with $v \in V$ denotes a random variable and X_W with $W \subseteq V$ denotes a set of random variables. Furthermore, let P denote a joint probability distribution (JPD) of X . Let $U, W, Z \subseteq V$ be disjoint sets of indices, then X_U is said to be *conditionally independent* of X_W given X_Z , if

$$P(X_U | X_W, X_Z) = P(X_U | X_Z), \quad (1)$$

denoted by $U \perp\!\!\!\perp_P W | Z$. The entire set of independence relations of P is denoted by $\perp\!\!\!\perp_P$.

The independence relation $\perp\!\!\!\perp_P$ can also be represented, although not always perfectly (see below), by means of an ADG G , in which the entire set of independence relations is denoted by $\perp\!\!\!\perp_G$. In the graph, arcs represent dependences, and absence of arcs represents (conditional) independences. These independences can be read off by the d-separation criterion, defined as follows. A trail τ in an ADG G is said to be *blocked* by a set Z if one of the following conditions is satisfied: (i) $v \in Z$ and v appears on the trail τ , and either *no* or *only one* of the arcs of τ meeting at v is directed to v ; (ii) $v \notin Z$, $\delta(v) \cap Z = \emptyset$, where $\delta(v)$ are the descendants of v , and *both* arcs meeting at v on τ are directed to v (*convergent connection*). It is said that the sets U and W are *d-separated* by Z if any trail between a vertex in U and a vertex in W is blocked by the set Z ; formally: $U \perp\!\!\!\perp_G W | Z$. Otherwise, U and W are *d-connected* by Z , denoted by $U \not\perp\!\!\!\perp_G W | Z$.

Unfortunately, not every independence encoded in a JPD can be represented graphically by means of d-separation in an associated ADG. An ADG G is said to be a *directed I-map*, I-map for short, if each independence in G is also valid in P . If it is impossible to omit any of the arcs in an ADG G without losing the property that it is an I-map of P , G is said to be a *minimal* I-map of P .

Let $G = (V, A)$ be an ADG and let X_V be a set of random variables corresponding to the vertex set V and let P denote the joint probability distribution of X_V , then a *Bayesian network* is a pair $\mathcal{B} = (G, P)$, with G being an I-map of P .

4 Dynamic Bayesian networks

In this section, we begin by defining DBNs, which are extensions of ordinary Bayesian networks and allow modelling the uncertainty involved in processes regarding the dimension of time. Subsequently, two independence reading-off methods are defined for the ability to distinguish between atemporal and temporal independences. In this paper, *time* is denoted by T and is assumed to be a subset of the set of the natural numbers including zero; a *time point* t is then a member of T . From now on, let T stand for the time axis with associated total order $< \subseteq T \times T$.

4.1 Basic elements

Independence relationships between random variables with the *same* time point are represented by means of an acyclic directed graph, called a *timeslice*. Between timeslices, vertices corresponding

to random variables may be linked to each other by means of so-called temporal arcs. Thus, a DBN consists of two parts: (i) a time-independent atemporal part (the timeslices), and (ii) a time-dependent temporal part. First, we consider the atemporal part.

Definition 1 (timeslice and atemporal arcs) An ADG $G_t = (V_t, A_t^a)$, with the set of vertices V_t and the set of atemporal arcs $A_t^a \subseteq V_t \times V_t$, $t \in T$, is called a timeslice at time point t .

The set of all timeslices G of a DBN is taken as:

$$G = \{G_t \mid t \in T\} = \{(V_t, A_t^a) \mid t \in T\} = (V_T, A_T^a). \quad (2)$$

An arc $(u_t, v_{t'})$ with $t < t'$ is called a *temporal arc*. The set of temporal arcs of set G is denoted by A^t . Thus temporal arcs connect timeslices with strict direction from the past to the future giving rise to the following structure.

Definition 2 (temporal network) A temporal network N is defined as a pair $N = (V_T, A)$, where $G = (V_T, A_T^a)$ and $A = A_T^a \cup A^t$.

Clearly, a temporal network N is also an ADG. A DBN is defined as a pair $\mathcal{DBN} = (N, P)$, where P is the JPD on X_{V_T} . In the remaining part of this paper, when the symbol t is used as a superscript it indicates a temporal property; if it is used as a subscript it acts as a time index.

Both temporal and atemporal relations in the network can be represented by means of trails. An *atemporal trail* contains no temporal arcs and is denoted by τ^a . A *temporal trail* consists of at least one temporal arc and is denoted by τ^t . The sets of all atemporal and temporal trails are denoted by Θ^a and Θ^t , respectively. Considering the temporal relationships we only need to consider temporal trails resulting into a reduced temporal network.

Definition 3 (reduced temporal network) Let $N = (V_T, A)$ be a temporal network. Then, $N_{|\Theta^t} = (V_T, A_{\Theta^t})$ with the set of vertices V_T and the set of all arcs that are included in any temporal trail $A_{\Theta^t} \subseteq A$, is called a reduced temporal network.

Observe that the reduced temporal network is based on the set of temporal trails, which may consist of both atemporal and temporal trails. A further partitioning of the reduced temporal network is based on its set of arcs. This partitioning is obtained by decomposing a reduced temporal network into two parts, where one part consists of only atemporal and the another part of only temporal arcs. The *atemporal part of the reduced temporal network* is denoted by $N_{|\Theta^t, A_T^a} = (V_T, A_{\Theta^t, A_T^a})$, where V_T is the set of vertices and $A_{\Theta^t, A_T^a} \subseteq A_T^a$ consists of all atemporal arcs in the reduced temporal network. The *temporal part of the reduced temporal network* is denoted by $N_{|\Theta^t, A^t} = (V_T, A_{\Theta^t, A^t})$, where $A_{\Theta^t, A^t} \subseteq A^t$ consists of all temporal arcs in the reduced temporal network.

As an example of a structure of a DBN consider Figure 1, where timeslices are depicted by rectangles. It holds that $G_1 = (\{\text{VAP}_1, \text{X-chest}_1, \text{PaO}_2\text{FiO}_{2_1}, \text{Temp}_1\}, \{(\text{PaO}_2\text{FiO}_{2_1}, \text{X-chest}_1)\})$ and $(\text{X-chest}_1, \text{X-chest}_2) \in A^t$. An atemporal trail is $\tau^a = \text{PaO}_2\text{FiO}_{2_1} \rightarrow \text{X-chest}_1$ and a temporal one is $\tau^t = \text{X-chest}_1 \rightarrow \text{X-chest}_2 \leftarrow \text{PaO}_2\text{FiO}_{2_2}$.

4.2 Atemporal and temporal d-separation

As a DBN includes temporal and atemporal elements, the question is how to distinguish between these relations. In this section, we define the necessary independence reading-off methods.

We start by considering time-independent relations by ignoring the role played by temporal arcs. To obtain these relations, we apply the *atemporal d-separation* criterion, as follows. Let $U, W, Z \subseteq V_T$ be distinct vertex sets, then, if each atemporal trail connecting any vertex in U with any vertex in W is blocked by the set Z , then U and W are said to be *atemporally d-separated*, denoted by $\perp\!\!\!\perp_G$, given Z written $U \perp\!\!\!\perp_G W \mid Z$; otherwise, they are said to be *atemporally d-connected*, formally $U \not\perp\!\!\!\perp_G W \mid Z$. Atemporal d-separation among vertices belonging to only one timeslice G_t is denoted by $\perp\!\!\!\perp_{G_t}$; similarly, atemporal d-connection is in that case denoted by $\not\perp\!\!\!\perp_{G_t}$.

Temporal d-separation is applied to obtain the time-dependent conditional independences of the temporal network as follows. If each temporal trail between any vertex in U and any vertex in W is blocked by the set Z , then, U and W are said to be *temporally d-separated* given Z , written $U \perp\!\!\!\perp_{N_{|\Theta^t}} W \mid Z$; otherwise, they are called *temporally d-connected*, denoted by $U \not\perp\!\!\!\perp_{N_{|\Theta^t}} W \mid Z$. The entire set of independence relations in the reduced temporal network is denoted by $\perp\!\!\!\perp_{N_{|\Theta^t}}$.

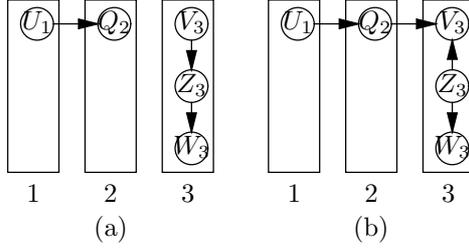


Figure 2: Temporal networks (a) and (b).

Finally, we also apply d-separation defined in Section 3 to the entire DBN to obtain all (conditional) independences. The resulting relation $\perp\!\!\!\perp_N$ denotes the set of independences in network N , and is also interpreted as minimal I-map of P .

5 The join operator

In the previous section, DBNs and their (a)temporal relations have been defined. In this section, we define the join operator applied to correctly join atemporal and temporal independence relations of DBNs. In Section 5.1, two significant properties are introduced, which have to be included in the definition of the join operator. In Section 5.2, the join operator and its properties are considered.

5.1 Dependence preservation and independence concatenation

In this section, we introduce the dependence preservation and independence concatenation properties, which are necessary to join two independence relations in the *right* way. Dependence preservation will allow us to join an independence and a dependence relation in a correct way, whereas independence concatenation joins two independence relations taking into account that these relations after the join process may give rise to a dependence, as we will see.

We begin with the discussion of dependence preservation. The reason that dependence preservation is required for joining independence statements can be explained in terms of the concepts of consistency and dominance as follows.

Let the independence relations $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$ be defined on the same vertex set V . Then, if there exist $U \perp\!\!\!\perp W \mid Z$ and $U \not\perp\!\!\!\perp W \mid Z$ for arbitrary, mutually disjoint sets of vertices $U, W, Z \subseteq V$, then these independence statements and therefore independence relations $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$ are said to be *inconsistent*. Otherwise, the statements are *consistent*. The purpose is to join independence relations together; however, when two independence statements and, therefore, relations are inconsistent, a choice has to be made between the independence and dependence resulting in an inconsistency. In other words, one statement has to *dominate* the other one. If the relations $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$ are inconsistent due to the statements $U \perp\!\!\!\perp W \mid Z$ and $U \not\perp\!\!\!\perp W \mid Z$, then $U \not\perp\!\!\!\perp W \mid Z$ is said to *dominate* $U \perp\!\!\!\perp W \mid Z$. Since dominance has to be taken into account during joining independences, the following property is defined.

Definition 4 (dependence preservation) *Let $\perp\!\!\!\perp$, $\perp\!\!\!\perp'$ and $\perp\!\!\!\perp''$ be independence relations all defined on V . Then, if $U \not\perp\!\!\!\perp W \mid Z$ or $U \not\perp\!\!\!\perp' W \mid Z$ and it holds that $U \perp\!\!\!\perp'' W \mid Z$ for all $U, W, Z \subseteq V$, then it is said that $\perp\!\!\!\perp''$ satisfies the dependence preservation property with regard to $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$.*

Next, we define the independence concatenation property. This takes into account that when independence relations are combined, an independence may change into a dependence. This is demonstrated in Figure 2. In both temporal networks (a) and (b) we have that $U_1 \perp\!\!\!\perp_{N_{\{\Theta^t, A_T^t\}}} W_3 \mid V_3$ and $U_1 \perp\!\!\!\perp_{N_{\{\Theta^t, A^t\}}} W_3 \mid V_3$. However, in temporal network (a) it holds that $U_1 \perp\!\!\!\perp_{N_{\{\Theta^t\}}} W_3 \mid V_3$ and in temporal network (b) we have $U_1 \not\perp\!\!\!\perp_{N_{\{\Theta^t\}}} W_3 \mid V_3$. Observe that if joining two independences results in a dependence, then this dependence is always represented by a temporal trail, which *has* to consist of at least one atemporal and at least one temporal subtrail. Therefore, the (in)dependence relations can be examined separately using blockage in terms of atemporal and temporal subtrails, yielding the following proposition, as basis for the definition of independence concatenation.

Proposition 1 Let $U, W, Z \subseteq V_T$ be disjoint sets of vertices and let temporal trail τ^t connect vertices u and w , $u \in U, w \in W$, in the reduced temporal network $N_{|\Theta^t}$. If (i) one of the (a)temporal subtrails of τ^t is blocked by Z or (ii) each (a)temporal subtrail of τ^t is not blocked by Z but there is at least one convergent connection on two consecutive (a)temporal subtrails with vertices not included in Z , then the temporal trail τ^t is blocked by Z . Otherwise, τ^t is not blocked by Z .

Note that Proposition 1 provides the basis for considering the case, when joining two independences results in a dependence. To ensure that these dependences are included in the new joined independence relation the independence concatenation property is defined.

Definition 5 (independence concatenation) Let $U, W, Z \subseteq V_T$ be disjoint vertex sets in a temporal network N , and let G, G' and G'' be reduced subgraphs of N with corresponding sets of trails Θ, Θ' and Θ'' . If each trail $u \sim w$, $u \in U, w \in W$ in subgraphs G and G' is blocked by Z and if for each $q \in V_T \setminus (U \cup W)$ one of the trails $u \sim q \in \Theta$ and $q \sim w \in \Theta'$ is blocked by Z or these two trails do not constitute a convergent connection at $q \in Z$, then if for $U \perp\!\!\!\perp_G W \mid Z$ and $U \perp\!\!\!\perp_{G'} W \mid Z$, it holds that $U \perp\!\!\!\perp_{G''} W \mid Z$, then it is said that $\perp\!\!\!\perp_{G''}$ satisfies the independence concatenation property with regard to $\perp\!\!\!\perp_G$ and $\perp\!\!\!\perp_{G'}$.

5.2 The join operator

In this section, the join operator is defined and significant properties of this operator are given (other properties are omitted because of space limitations). We start by defining the join operator.

Definition 6 (join operator) Let $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$ be two independence relations defined on the same vertex set V . The join operator is denoted by \circ . The join of these two relations, denoted by $\perp\!\!\!\perp \circ \perp\!\!\!\perp' = \perp\!\!\!\perp''$, is then again an independence relation, $\perp\!\!\!\perp''$, defined on V , that satisfies the dependence preservation and the independence concatenation properties.

The dependence preservation property can be understood in terms of the union of graphs.

Proposition 2 Let $G = (V, A), G' = (V, A')$ and $G'' = (V, A'')$ be three ADGs, where $A \cup A' \subseteq A''$. Then, it holds that $\perp\!\!\!\perp_{G''} \subseteq \perp\!\!\!\perp_G \circ \perp\!\!\!\perp_{G'}$.

Clearly, dependence preservation results in the property that I-mappedness is preserved. In Proposition 2 we have used the subset relation \subseteq , as the resulting ADG need not precisely consist of the union of the set of arcs of graphs G and G' . If the graph G'' equals their union and does not include extra dependences, the join operator is said to be *minimally* dependence preserving. Finally, we have the following property:

Proposition 3 Let $\perp\!\!\!\perp$ and $\perp\!\!\!\perp'$ be two independence relations defined on the same vertex set V . Then, for the independence relation $\perp\!\!\!\perp'' = \perp\!\!\!\perp \circ \perp\!\!\!\perp'$ it holds that $\perp\!\!\!\perp'' \subseteq (\perp\!\!\!\perp \cap \perp\!\!\!\perp')$.

6 Temporal and atemporal independence: their interaction

In Section 4, we have defined DBNs, with a graphical representation consisting of atemporal and temporal independence relations. In Section 5, the join operator has been defined such that it satisfies dependence preservation and independence concatenation. In this section, based on the two previous sections, we investigate how to employ the join operator for joining the temporal and atemporal independence relations underlying temporal networks and, therefore, DBNs, to support the modelling of *non-repetitive* DBNs. In Section 6.1, we show that the join operator can be used for joining the atemporal and temporal parts of the reduced temporal network. In Section 6.2, the same is shown for the atemporal and temporal relations of the entire temporal network.

6.1 Joining atemporal and reduced temporal networks

In this section, we start by the consideration of the relations $\perp\!\!\!\perp_{G_t}$ and $\perp\!\!\!\perp_G$. Here, the following proposition establishes the connection between the d-separation relations of the individual timeslices G_t and of the set of timeslices G .

Proposition 4 Let $\mathcal{DBN} = (N, P)$, with temporal network $N = (V_T, A)$, $G = (V_T, A_T^a)$, $A = A_T^a \cup A^t$ and JPD P , then: (i) $\perp\!\!\!\perp_G = \bigcap_{t \in T} \perp\!\!\!\perp_{G_t}$, and (ii) $\perp\!\!\!\perp_G = \bigcup_{t \in T} \perp\!\!\!\perp_{G_t}$.

Proof: Case (i). The conditional independence relation $\perp\!\!\!\perp_{G_t}$ consists of the complete set of conditional independences regarding its own vertex set V_t , which is, except when G_t is an empty graph, not completely included in any other $\perp\!\!\!\perp_{G_{t'}}$, $t \neq t'$. Therefore, the complete set of conditional independences of G is obtained by the intersection. Furthermore, the atemporal relations *always* describe independences between vertex sets of different timeslices. These sets of independences are the same for each relation $\perp\!\!\!\perp_{G_t}$, and these independence statements are also included in $\perp\!\!\!\perp_G$ by means of intersection.

Case (ii). $\perp\!\!\!\perp_G = \overline{\bigcap_{t \in T} \perp\!\!\!\perp_{G_t}} = \bigcup_{t \in T} \perp\!\!\!\perp_{G_t}$. \square

Observe that according to Proposition 4, the join operator \circ is interpreted as the intersection of the independence relations, which does not hold in general.

The connection between the join operator for reduced temporal networks is established by the following proposition.

Proposition 5 Let the reduced temporal networks $N_{|\Theta^t, A_T^a}$ and $N_{|\Theta^t, A^t}$ be the atemporal and temporal parts of the reduced temporal network with independence relations $\perp\!\!\!\perp_{N_{|\Theta^t, A_T^a}}$ and $\perp\!\!\!\perp_{N_{|\Theta^t, A^t}}$, respectively. Then, there exists a join operator \circ , such that the independence relation of the reduced temporal network $N_{|\Theta^t}$ is equal to $\perp\!\!\!\perp_{N_{|\Theta^t}} = \perp\!\!\!\perp_{N_{|\Theta^t, A_T^a}} \circ \perp\!\!\!\perp_{N_{|\Theta^t, A^t}}$.

Proof: Since the join operator satisfies independence concatenation and dependence preservation, joining two independence relations, the join operator has inserted all the independences, which are obtained applying the d-separation criterion. Furthermore, these two properties ensure us that there is also no incorrect relation included in $\perp\!\!\!\perp_{N_{|\Theta^t}}$. \square

We have joined two independence relations; however, we still have to show the correctness of the join operator in terms of the union of graphs. *Soundness* of the join operator means that all independence statements obtained by joining two independence relations can be read off from the union of the underlying graphs, whereas *completeness* means that none of the independence statements of the union of the graphs has been omitted in the resulting independence relation.

Theorem 1 If it holds that $\perp\!\!\!\perp_{N_{|\Theta^t}} = \perp\!\!\!\perp_{N_{|\Theta^t, A_T^a}} \circ \perp\!\!\!\perp_{N_{|\Theta^t, A^t}}$, then the join operator \circ is sound and complete.

Proof: Soundness: By the temporal d-separation criterion, since the two independence relations between two vertex sets will only be joined into an independence relation, if each trail connecting these two vertex sets is blocked by the conditioning vertex set, independence in the resulting relation is implied. Completeness: We prove the completeness of the join operator by proving that if $N_{|\Theta^t, A_T^a}$ and $N_{|\Theta^t, A^t}$ are I-maps then $N_{|\Theta^t}$ is also an I-map. The preservation of I-mappedness of the reduced temporal network follows from Proposition 2. Furthermore, the join operator satisfies also the minimally dependence preserving property, since $N_{|\Theta^t}$ does not contain any extra arcs. \square

6.2 Joining it all together

In this subsection, the temporal and atemporal independence relations are joined together, yielding the relation $\perp\!\!\!\perp_N$. Recall that the atemporal relations are defined by the atemporal properties of the graph. The relation $\perp\!\!\!\perp_G$ is obtained by the application of the concept of atemporal d-separation. Temporal relations are relations which are recovered by the temporal d-separation criteria and are denoted by $\perp\!\!\!\perp_{N_{|\Theta^t}}$. The following proposition shows that these relations can be linked to each other by means of the join operator.

Proposition 6 Let $\perp\!\!\!\perp_G$ and $\perp\!\!\!\perp_{N_{|\Theta^t}}$ be atemporal and temporal independences of the temporal networks. Then, there exists a join operator \circ , such that the independence relation of the temporal network is equal to $\perp\!\!\!\perp_N = \perp\!\!\!\perp_G \circ \perp\!\!\!\perp_{N_{|\Theta^t}}$.

Proof: Let $U, W, Z \subseteq V_T$ be disjoint vertex sets in the temporal network N . As each trail $u \sim w$, $u \in U, w \in W$ is blocked by Z then $U \perp\!\!\!\perp_G W \mid Z \circ U \perp\!\!\!\perp_{N_{|\Theta^t}} W \mid Z = U \perp\!\!\!\perp_N W \mid Z$ holds. \square

Figure 3 summarises the way to join independence relations defined in propositions 4, 5 and 6.

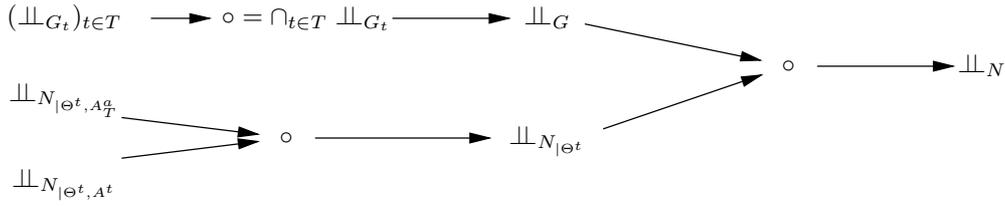


Figure 3: Joining temporal and atemporal independence relations.

Theorem 2 *If it holds that $\perp_N = \perp_G \circ \perp_{N_{|\Theta^t}}$, then the join operator \circ is sound and complete.*

Proof: Soundness follows from the temporal d-separation criterion. If each trail connecting the two joined vertex sets is blocked by the conditioning vertex set, independence in the resulting relation is implied. The proof of completeness is also similar to the proof of Theorem 1. If the relations \perp_G and $\perp_{N_{|\Theta^t}}$ are I-maps then the relation \perp_N is also an I-map according to Proposition 2. \square

Finally, the various independence relations can be compared to each other.

Proposition 7 *The independence sets \perp_G , $\perp_{N_{|\Theta^t}}$, $\perp_{N_{|\Theta^t, A_T^t}}$, $\perp_{N_{|\Theta^t, A^t}}$ and \perp_N also satisfy the following properties:*

- $\perp_{N_{|\Theta^t}} \subseteq \perp_{N_{|\Theta^t, A_T^t}}$, $\perp_{N_{|\Theta^t}} \subseteq \perp_{N_{|\Theta^t, A^t}}$;
- $\perp_N \subseteq \perp_G$, $\perp_N \subseteq \perp_{N_{|\Theta^t}}$.

7 Conclusions

The aim of the research described in this paper was to study how the modelling non-repetitive DBNs can be simplified by distinguishing between time-independent and time-dependent independence relations. As this gave rise to various separate, but linked, independence relationships, the usual property that independence and dependence complement each other no longer holds. We introduced a join operator with special semantics to overcome this problem. Using the join operator allows one to build DBNs in a modular fashion, hence the title of the paper. As far as we know, this paper offers the first systematic method for building non-repetitive DBNs. Much work still needs to be done to bridge the gap between the theoretical work in this paper and practice.

References

- [1] E. Castillo, J.M. Gutiérrez, and A.S Hadi. Expert Systems and Probabilistic Network Models. Springer-Verlag, New York, 1997.
- [2] M. Deviren and K. Daoudi. Continuous speech recognition using dynamic Bayesian networks: a fast decoding algorithm. In: Proc PGM'02, Cuenca, Spain, 2002, pp. 54–60.
- [3] N. Friedman, K. Murphy, and S. Russell. Learning the structure of dynamic probabilistic networks. In: Proc 14th UAI, 1998, pp. 139–147.
- [4] F.V. Jensen. Bayesian Networks and Decision Graphs. Springer, New York, 2001.
- [5] K.P. Murphy. Dynamic Bayesian Networks: Representation, Inference and Learning. PhD Thesis, UC Berkeley, 2002
- [6] U. Kjaerulff. A computational scheme for reasoning in dynamic probabilistic networks. In: Proc UAI'92, 1992, pp. 121–129.
- [7] Y. Xiang, and V. Lesser. Justifying multiply sectioned Bayesian networks. In: Proc 6th Int Conf on Multiagent Systems, Boston, 2000, pp. 349–356.
- [8] A. Tucker and X. Liu. Learning Dynamic Bayesian Networks from Multivariate Time Series with Changing Dependencies. The 5th International Symposium on IDA, 2003.

Intelligent Tutoring Agent for Settlers of Catan

Jeroen Geuze^{a,†}

Egon L. van den Broek^{a,b,†}

^a *Nijmegen Institute for Cognition and Information, Radboud University Nijmegen
P.O.Box 9104 6500 HE Nijmegen, The Netherlands
geuze.j@gmail.com*

^b *Center for Telematics and Information Technology, University of Twente
P.O.Box 217, 7500 AE Enschede, The Netherlands
e.l.vandenbroek@utwente.nl*

Abstract

An Intelligent Tutoring Agent (ITA) for the board game Settlers of Catan (SoC) is introduced. It uses CLIPS knowledge bases, connected by JCLIPS to a JAVA implementation of SoC. It is founded on a new theoretical framework that describes the development of negotiation skills in children. Using this framework, the ITA helps children in developing negotiation skills through play, which makes it unique in its kind.

1 Introduction

The concept of learning by playing has proved its use for virtually all species, including man kind. Playing and games are even considered as essential for human development [3]. Consequently, various disciplines have included games as a field of research [5], among which AI. However, especially for AI, mostly board games are topic of research; e.g., Chess, Checkers, and Go [5]. This paper issues another type of board game: Settlers of Catan (SoC) [10]; Figure 1 provides a brief description of the game. We choose SoC since we were interested in the development of negotiation processes with children and for playing SoC, negotiation is one of the most important skills. Another advantage of SoC is its popularity.

This paper introduces an Intelligent Tutoring Agent (ITA) that helps children understand the negotiation processes through playing SoC. For the latter purpose, we first developed a theoretical framework, founded on 1) Piaget's theory of child development and 2) adult negotiation strategies, as will be described in Section 2. Subsequently, the expert system of the agent founded on the theoretical framework is documented in Section 3. The development of the ITA itself is described in Section 4. The paper ends with a Discussion (Section 5) in which we reflect upon the work presented.

2 Theoretical Framework

The amount of research done on either children and board games or on children and negotiation is very limited [5]. Therefore, we choose to develop a new theoretical framework founded on two theories: 1) the child development theory of Piaget [7] and 2) Thompson and Hastie's [12] theory on negotiation between adults. The combination of the latter theories, enabled us to determine the initial level of understanding of the negotiation process in children. We will now first briefly introduce both theories and derive some general cognitive notions. Subsequently, the general cognitive notions are related to the stages Piaget distincts. Knowledge gathered through the latter three steps is merged to a new theoretical framework.

[†]At the time this research was conducted, both authors were affiliated with the Department of Artificial Intelligence, Division of Mathematics and Computer Science, Vrije Universiteit Amsterdam.

The players in the game represent the eponymous Settlers, establishing a colony on the previously uninhabited Island of Catan. The island itself is laid out randomly at the beginning of each game from terrain hexes depicting mountains, plains, woods, etc. The settlements founded by the players provide their owners with natural resources dependent on the adjacent terrain, and dependent on die rolls each turn. These natural resources are used for activities such as building roads, upgrading settlements to cities, and purchasing development cards which represent a variety of advances toward civilization. The winner is the first player to accumulate ten victory points via these activities.

Players are allowed to trade among each other the resources they have produced, and to trade “off the island” for a hefty price. It is difficult for any one player to produce all the resources necessary for progress, so astute trading is the strategic heart of the game. Player interaction is further complicated by the presence of a robber, which is used to steal from other players and hinder their production of resources. There is no combat. Apart from the robber and refusal to trade there is no way to harm other players.

Source: http://www.absoluteastronomy.com/encyclopedia/S/Se/Settlers_of_Catan.htm

Figure 1: Brief description of the game Settlers of Catan (SoC).

Piaget distinct four stages of cognitive development [7]:

1. Sensory-motor stage (0-2 years): Characterized by the development of sensor and motor reactions.
2. Pre-operational stage (2-7 years): Characterized by egocentrism, rigidity of thought, semi-logical reasoning, and limited social cognition.
3. Concrete operational stage (7-11 years): The development of the operation: an internalized mental action that is part of an organized structure. Consequently, the child starts to think in terms of operations.
4. The formal operational period (11-15 years): The ability to generate hypotheses about the logical relations between the outcomes of concrete operations develops.

However, the theory of Piaget does not discuss negotiation. Hence, for the latter purpose another theory, that of Thompson and Hastie was taken into account.

According to Thompson and Hastie [12], six judgments can be distinguished in negotiating:

1. Judgments of the Other Party; e.g., the other party’s strength or firmness, competitiveness, and fairness.
2. Judgments of the Self that involve the negotiator’s tastes, values, and preferences.
3. Interpersonal Judgments of Utilities: the negotiator’s perception of his own utilities, of the utilities of the other party, and the amount of resources that will be negotiated about.
4. Judgments of Offers and Counter-offers.
5. Judgments of Outcomes: to determine the value of a solution and, consequently, the satisfaction.
6. Judgments of Negotiation Process concern the fairness and satisfaction of the whole negotiation process. They include judgments about the rules, fair play, norms, etc.

The general cognitive aspects were determined taking the following aspects into consideration: 1) They should be based on the judgments put forward by Thompson and Hastie [12], 2) they should be applicable on the domain of Settlers of Catan and be generalizable to a broader domain, and 3) they should be able to match with the stage theory of child development to tailor these aspects to children. The general cognitive aspects and the judgments they are based upon are summarized in Table 1.

The fourth and last step in setting up the theoretical framework was to denote for each stage defined by Piaget [7], whether or not each of the general cognitive aspects is already present in children in that stage of development. Let us illustrate the latter by sketching the decision process for the *Feeling for utility to other party* aspect. In the pre-operational stage, this aspect is not yet developed since children have a limited

Table 1: Cognitive aspects and corresponding judgments from which they were inferred from.

Cognitive Aspect	Judgment(s)
Feeling for numbers	4
Feeling for fairness	4
Feeling for utility to self	2, 3, 4, 5
Feeling for utility to other party	1, 3, 4, 5
Ability to keep one's goal in mind	
Ability to hypothesize	4, 5

social cognition and are egocentric. Children in this stage have problems taking another's' viewpoint; e.g., when a child is looking at a picture in a book and draws his mother's attention to this picture, it does not realize that the mother is not able to see the picture when she is sitting opposite the child. From this, one can deduce that they will not be able to determine the utility of a negotiation to another party. In the concrete operational stage, the egocentrism fades away and children's limited social cognition is improved, enabling them to make judgments about utility to another by the end of the concrete operational stage. Once a certain level of development has been reached there will be no regression; therefore, once a cognitive aspect has been developed, it will not be lost. This explains why the Feeling for utility to other is also developed in the formal operational stage.

Table 2 summarizes the results of mapping the cognitive aspects to Piaget's stage theory. This table denotes that in the concrete operational stage, most of the cognitive aspects are developed. From these results, the target group age could be determined. According to Piaget [7], the developments are only complete at the end of each stage. He also showed, using the containment experiment [7], that children are not able to learn a certain task when they have not yet reached the required level of development. Keeping this in mind, the target group age was determined to be around 10 years of age. The mapping of the aspects to Piaget's theory was confirmed by a semi-experiment, in which three 10 year old children played a game of Settlers of Catan. This is also the lower age limit set by the makers of Settlers of Catan.

The findings presented in Table 2 are also confirmed by Gobet [5] who states: "Parker argues that children understand the concept of rules ... at about 6 years of age, even if they sometimes forget specific rules or change them during play. They also understand the concepts of winning and losing. But they only apply rules consistently ... at 11 or 12 years of age." (p.139) and "Sutton-Smith ... reports an association between level of games played by children and their presumed level of cognitive development based on their age." (p. 140).

Table 2: Cognitive Aspects developed in each stage of Piaget's theory, where X and ✓ denote respectively that a cognitive aspect is or is not developed.

Aspects / Stage	Pre-operational	Concrete operational	Formal operational
Feeling for numbers	X	✓	✓
Feeling for fairness	X	✓	✓
Feeling for utility to self	✓	✓	✓
Feeling for utility to other	X	✓	✓
Ability to keep goal in mind	✓	✓	✓
Ability to hypothesize	X	X	✓

3 Expert System

Founded on the theoretical framework discussed in the previous section, the expert system of the ITA was developed. This enabled the ITA to trace the level of development of children. The latter is achieved by determining what cognitive aspects are not fully developed and give advice accordingly.

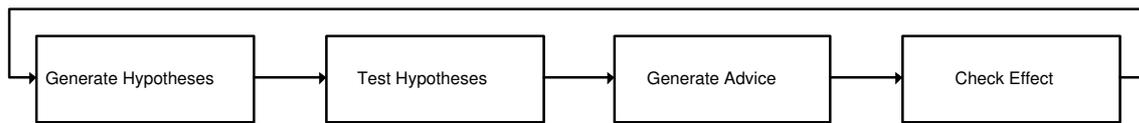


Figure 2: Global architecture of the expert system, adopted from [11]. The Generate Hypotheses module puts forth a hypothesis about whether or not a cognitive aspect is developed, this is then checked by the Test Hypotheses module. The Generate advice module supplies an advice message when the previous module has found an aspect undeveloped. The Check Effect module determines whether the same mistake is made again.

3.1 Architecture

For the architecture of the expert system, a number of expert system models were considered. Schreiber et al. [9] describe a number of knowledge model templates in their book on knowledge engineering. When considering these knowledge templates, three can be seen as candidates for the architecture of the expert system of the ITA: the assessment, diagnosis, and monitoring templates. The assessment template only takes one point in time into account. The monitoring template only registers a discrepancy and does not look at the cause of it. The diagnosis template, on the other hand, does incorporate both these characteristics and is, therefore, the most suitable of the three. The architecture of the expert system is based on this knowledge template.

The global architecture of the expert system is presented in Figure 2. Based on the negotiation history of the user, hypotheses are generated consisting of the cognitive abilities that have been defined (e.g., the hypothesis that the *Feeling for fairness* aspect has not yet been fully developed). Subsequently, the hypotheses will be tested; this will be done by checking whether or not all the facts necessary for this hypothesis are satisfied. Hence, the agent will check if there is more evidence that this hypothesis is true; if not, it will wait until there is or until the game ends. Once a hypothesis is validated, advice is generated based on this hypothesis. This advice is generated based on the number of times a certain error has been made, which is determined by the check effect component. The advice will become more explicit as more errors are made.

3.2 From framework to expert system

The rules in the modules of the expert system were designed from global to specific, with the cognitive aspects as starting point. The agent assumes that all the cognitive aspects have been developed and tries to disprove them. For the latter purpose, first the system determines whether or not a cognitive aspect needs to be elicited. Next, the system checks whether or not that aspect has been developed by asking a set of questions. Figure 3 provides some sample questions for the *Feeling for fairness* aspect.

Feeling for fairness:

- How often does the subject try to close an unfair (asking more than giving) deal?
 - 0 - 5 *
 - 5 - 10
 - > 10
- How often does the subject accept an unfair deal?
 - 0 - 5 *
 - 5 - 10
 - > 10

* Expected interval if cognitive ability has been fully developed.

Figure 3: Questions for cognitive aspect Feeling for fairness.

For each of the six cognitive aspects, questions were defined and, subsequently, converted into symbolic rules. An example of a symbolic rule used in the *Generate Hypothesis* and *Test Hypotheses* modules is provided in Figure 4. For a full list of questions and the symbolic rules deduced from these questions, see [4].

IF	Offer unfair deal
OR	Accept unfair deal
THEN	Hypothesis(NOT feeling for fairness)
<hr/>	
IF	Hypothesis(NOT feeling for fairness)
AND	Offer unfair deal
AND	Accept unfair deal
THEN NOT	Feeling for fairness

Figure 4: Example rule from the *Generate Hypotheses* (top) and *Test Hypotheses* (bottom) modules; see also Figure 2.

Each module of the expert system architecture contains its own knowledge base. These knowledge bases have been implemented in CLIPS [8]. The CLIPS knowledge bases and their inference engine are connected to the ITA, which is developed in JAVA, using JCLIPS [6], allowing JAVA to control the CLIPS inference engine. The ITA, which is connected to a digital version of SoC (JSettlers), determines whether or not the basic facts used in the knowledge bases are true. An example of a rule utilized by the ITA for determining two specific facts (*offer unfair deal* and *accept unfair deal*, used in Figure 4) is shown in Figure 5. The combination of Figures 3, 4, and 5 illustrates the phases in which the cognitive aspects from the theoretical framework are implemented into an expert system.

Given: Negotiation(fromPlayer, toPlayer, giveSet, getSet),	
IF	fromPlayer == monitoredPlayer
AND	giveSet < getSet
THEN	Offer unfair deal
IF	toPlayer == monitoredPlayer
AND	giveSet < getSet
AND	offer accepted
THEN	Accept unfair deal

Figure 5: Example rules for specific facts: offer and accept an unfair deal. The given fact containing information about what is offered (*giveSet*) and what is asked (*getSet*) and which players are involved is obtained from the digital version of Settlers of Catan. Offering an unfair deal holds when the monitored player is offering and when he asks more resources than he is willing to offer. Accepting an unfair deal holds when the accepting player is the monitored player, when he is accepting less resources than he gives in return, and when the offer is actually accepted.

4 Intelligent Tutoring Agent

The ITA is a package that is integrated into the original JSettlers client (the *SOCPlayerClient* class, where SOC stands for Settlers of Catan) Consequently, the agent will be automatically started by the JSettlers client when a game is started. Moreover, the ITA is able to retrieve information about the game through the JSettlers client.

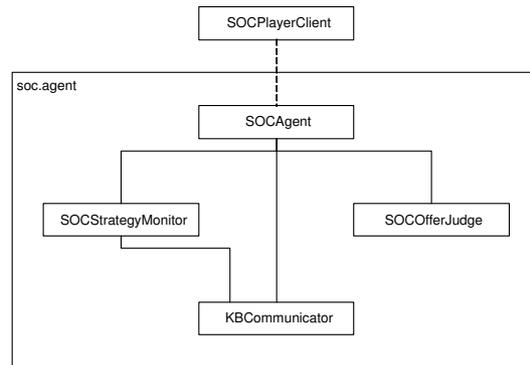


Figure 6: Architecture of the Intelligent Tutoring Agent: The agent is started by the SOCPlayerClient, and uses the Strategy Monitor and Offer Judge to handle some of the information processing. Both the SOCAgent class as the StrategyMonitor class utilize a knowledge base. The KBCommunicator class handles the communication with the knowledge bases.

4.1 Architecture

The agent uses a number of components for information processing. The three most important information processing components are: 1) the Strategy Monitor, 2) the Offer Judge, and 3) the KBCommunicator (Knowledge Base Communicator). Each of these components is used by the agent to handle part of the incoming game information. The Strategy Monitor handles information about strategies the player is or is not using, it utilizes a knowledge base to deduct crucial information. The Offer Judge handles all information about negotiations during the game. The KBCommunicator is responsible for the communication between the ITA and the expert system. Both the main class of the agent (SOCAgent) and the Strategy Monitor utilize knowledge bases and, thus, use the KBCommunicator. See Figure 6 for a schematic overview of the architecture of the ITA.

4.2 Interface

The interface is an added panel to the original JSettlers interface. This panel contains an area where the user can communicate his strategy to the agent. It contains a button that will open a new window that shows statistical information on the negotiation history. Moreover, tips to the user can be displayed, dependent on which strategy the player is using. Changes in the cognitive aspects detected by the agent and the corresponding text messages are provided to the user inside a dialog window. These windows will pop-up at the end of a turn. The agent will collect messages until the user ends his turn and they will be displayed then. A screenshot of the JSettlers interface with the added ITA component is provided in Figure 7.

4.3 Strategies

As mentioned in the previous section, the interface offers the user the possibility to communicate the strategy he is using to the agent. The agent is also able to determine what strategy the player is using by analyzing the behavior of the player during the game. A separate knowledge base has been developed to capture the knowledge needed with each strategy. When the agent has determined that the player is using a certain strategy, the user is informed and the strategy is marked in the user interface. Then, the agent can offer tips about negotiation possibilities. For instance, if a player is following the longest road strategy¹, he needs wood and stone to be able to buy roads. If the player has enough stone, but is lacking wood, the agent will suggest to the user he negotiates with fellow players to obtain the needed resource, wood in this case. An example of such a suggestion can be seen in the *Tips* section in Figure 7.

¹The longest road strategy is focused on obtaining the longest trade route award. This award can be obtained by owning the longest set of uninterrupted roads, this award is worth two victory points.

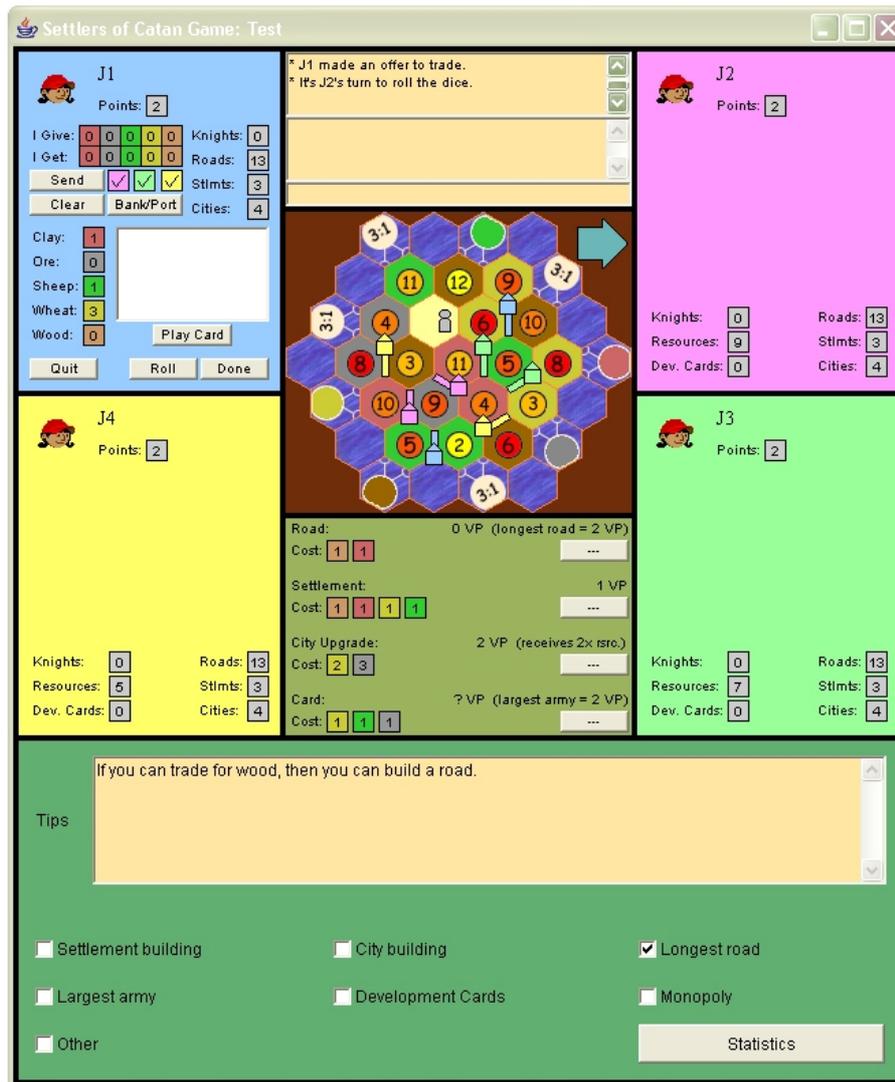


Figure 7: Screenshot of JSettlers with the ITA. At the bottom of the interface, the ITA interface can be seen. By pressing the statistics button, a window pops-up, containing statistical information on the negotiations. An aspect button in that window enables the pop up of a window that shows what cognitive aspects are still under investigation by the ITA.

5 Discussion

A new theoretical framework is presented based on the theories of Piaget [7] and of Thompson and Hastie [12], which defines six cognitive aspects needed for successful negotiation. Next, the newly developed Intelligent Teaching Agent (ITA) is presented, which utilizes an expert system for Settlers of Catan (SoC), based on the identified six cognitive aspects. This enables the ITA to compare the behavior of children to the knowledge of an expert negotiator. Subsequently, based on the latter comparison, advice can be provided to the child that plays SoC. Hence, through playing children can learn, supervised by the ITA, how to negotiate.

Currently, follow-up research is planned that aims to determine the learning effect of the current implementation of the ITA. To achieve the latter, experiments will be conducted in which subjects play SoC with and without the ITA. Consequently, both situations (i.e., with and without ITA) can be compared.

Follow-up research will aim at the improvement of the learning effect. Such an improvement can be realized in multiple ways. A more complex student model can be used; e.g., a qualitative model [2] or a neural network. Such a model would allow the ITA to make a more precise estimate of player's deficiencies, enabling it to offer even more tailored advice to the player. Alternatively, the method of providing the

messages to the player can be changed. Currently, this is done in a straight forward manner. When this is done in a more dynamic way (e.g., by using an animated agent), the learning effect could be improved [1].

In this paper, an Intelligent Tutoring Agent (ITA) is introduced for the complex, interactive board game Settlers of Catan. The agent utilizes a newly developed theoretical framework, which enables it to judge the level of understanding of negotiation in children and help them in developing it. A unique ITA is introduced, which helps children in developing negotiation skills through playing.

Acknowledgments

We thank Catholijn Jonker for her help during the start of the project and Maarten Menken for his help on JCLIPS. Robert Thomas is honored for his work on the JSettlers project. Further, we thank the students and children who participated voluntarily in our experiments. Nanja Smets and Giovanni Piantoni for reviewing the manuscript.

References

- [1] A.L. Baylor and Y. Kim. Pedagogical agent design: The impact of agent realism, gender, ethnicity, and instructional role. In *Intelligent Tutoring Systems, 7th International Conference*, pages 592–603, 2004.
- [2] B. Bredeweg and R. Winkels. Qualitative models in interactive environments: An introduction. *Interactive Learning Environments*, 5(1):1–18, 1998.
- [3] J.S. Bruner, A. Jolly, and K. Sylva, editors. *Play: its role in development and evolution*. Harmondsworth, 1976.
- [4] J. Geuze. An intelligent teaching agent for settlers of catan. Master’s thesis at the Department of Artificial Intelligence, Vrije Universiteit Amsterdam, 2005.
- [5] F. Gobet, A.J. de Voogt, and J. Retschitzki. *Moves in mind : The psychology of board games*. Psychology Press, 2004.
- [6] M.R. Menken. JCLIPS – CLIPS for Java. <http://www.cs.vu.nl/~mrmenken/jclips/> [Last accessed on September 6, 2006].
- [7] P.H. Miller. *Theories of Developmental Psychology*, chapter 1. Worth Publishers, 4th edition, 2002.
- [8] Gary Riley. CLIPS: A tool for building expert systems. <http://www.ghg.net/clips/CLIPS.html> [Last accessed on September 6, 2006].
- [9] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N. Shadbolt, W. van de Velde, and B. Wielinga. *Knowledge Engineering and Management, The CommonKADS Methodology*, chapter 6. MIT Press, 2000.
- [10] T. Smith. *The Settlers of Catan*. Mayfair Games, 1996. http://unicornsrest.org/reference/catan/mf_rules.htm [Last accessed on September 6, 2005].
- [11] Mark Stefik. *Introduction to knowledge systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1995.
- [12] L. Thompson and R. Hastie. Judgement tasks and biases in negotiation. *Research on Negotiation in Organizations*, 2:31–54, 1990.

MINING FOR STABLE PATTERNS: REGULAR INTERVALS BETWEEN OCCURRENCES

Edgar de Graaf Walter A. Kusters

Leiden Institute of Advanced Computer Science
Leiden University, The Netherlands
{edegraaf,kusters}@liacs.nl

Abstract

We propose a new measure of support (the number of occurrences of a pattern in a dataset), where we count the number of times a pattern occurs (nearly) in the middle between two other occurrences. The more a pattern occurs in the middle of two occurrences (transaction wise) the more stable or regular the interval between occurrences is. We will define this measure and show how it can be implemented in ECLAT in order to mine for frequent patterns. Our goal is to find all patterns that occur with some fixed interval, e.g., a certain phone-call occurs almost every Friday.

1 Introduction

Mining frequent patterns is an important area of data mining where we discover substructures that occur often in (semi-)structured data. In this work we will further investigate one of the simplest structures: itemsets. Much research has been done in the area of frequent itemset mining. We will propose an algorithm that discovers patterns that occur at regular moments, or rather in regular intervals. This will enable us to mine for events that occur, e.g., every Friday. The technique is extendable to more complicated structures like sequences.

Stable patterns are patterns that occur frequently and with certain intervals. The *interval* is the number of transactions *without* the pattern in between two transactions (events) *with* the pattern. Instead of discovering stable patterns one could consider to add an extra item to the events, for example the day of the week. However by discovering stable patterns we hope to find more unexpected intervals, e.g., every three hours a pattern occurs. This type of pattern is interesting because it shows events occurring with a regular interval, e.g., every morning around 9:10 coffee and a cookie are bought.

We will define this type of support and show its usefulness. To this end, this paper makes the following contributions, with emphasis on the first and second:

- We will **define stable patterns and show that they possess the APRIORI property**. This means that if pattern p' is contained in pattern p , then the stability value of p is at most equal to that of p' . The property guarantees efficient mining implementations.
- Furthermore we will **propose an algorithm** for the discovery of stable patterns and discuss its efficiency.
- Finally we will show that **this enables us to find new and interesting patterns** via explorative experiments on real and synthetic datasets.

Our working example is the mining of an access log from the Computer Science department of Leiden University. This access log will first be converted to sets of properties we are interested in, e.g., pages visited every hour. From here on we call this dataset the *website* dataset.

This research is related to work done on the (re)definition of support, using time with patterns and the incorporation of distance measured by the number of transactions between pattern occurrences. The notion of support was first introduced by Agrawal et al. in [1] in 1993. Since then many new and faster algorithms where proposed. We make use of ECLAT, developed by Zaki et al. in [10]. Steinbach et al. in [8] generalized the notion of support providing a framework for different

definitions of support in the future. Our work is also related to work described in [7] where association rules are mined that only occur within a certain time interval. Furthermore there is some minor relation with mining datastreams as described in [2, 6, 9], in the sense that they use time to say something about the importance of a pattern.

Finally this work is related to some of our earlier work. Results from [5] indicated that the biological problem could profit from incorporating consecutiveness into frequent itemset mining, which was elaborated in [3]. In the case of stable patterns we also make use of the transactions and the distance between them. Secondly in [4] it was mentioned that support is just another measure of saying how good a pattern fits with the data. There we defined different variations of this measure, and stability can be seen as one such variation.

The formal definitions concerning stable patterns and an algorithm are given in Section 2. In Section 3 we present experimental results, and we conclude in Section 4.

2 Stable Patterns

In this section we will define stable patterns. In particular, patterns that occur at regular intervals (e.g., at equidistant time stamps) will be called stable. In order to judge this property, we will determine how often events occur “in the middle” between two other events. We also describe an algorithm to find these patterns.

The intuitive idea of computing the variation in average distance between occurrences will not easily guarantee the APRIORI property, which is what we really need in the algorithm.

2.1 Definition

In this paper a dataset consists of a series of transactions that take zero time. Each transaction is an itemset, i.e., a subset of $\{1, 2, 3, \dots, max\}$ for some fixed integer max . The transactions can have time stamps; if so, we assume that the transactions take place at different moments. We choose some notion of *distance* between transactions; examples include: (1) the distance is the time between the two transactions and (2) the distance is the number of transactions (in the *original* dataset) strictly in between the two transactions. We will define $Trans(p)$ as the series of transactions that contain pattern (i.e., itemset) p ; the *support* of a pattern p is the number of elements in this ordered series.

We now define *w-stable patterns* as itemsets that occur frequent (support $\geq minsup$) in the dataset and that have *stability value* $\geq minstable$, where the values $minsup$ and $minstable$ are user defined thresholds. In order to define the stability value, we first introduce *w-good triples*: a *w-good triple* (L, M, R) consists of three transactions L, M and R , occurring in this order, such that $|distance(L, M) - distance(M, R)| \leq 2 \cdot w$; here w is a pregiven small constant ≥ 0 , e.g., $w = 0$. The stability value of a pattern p is the number of *w-good triples* in $Trans(p)$, plus the number of transactions in $Trans(p)$ that occur as left endpoint in a *w-good triple* (the L mentioned above), plus the number of transactions in $Trans(p)$ that occur as right endpoint in a *w-good triple* (the R mentioned above).

Note that the stability value of a pattern p' with $p' \subseteq p$ is at least equal to that of p : the so-called APRIORI or anti-monotone property. Also note that the stability value remains the same if we consider the dataset in reverse order.

We now show that equidistant events are “very” stable (in case $w = 0$):

Theorem Suppose that $Trans(p)$ has n elements, so p has support n . If $Trans(p)$ satisfies:

1. $n - 2$ elements occur as the left endpoint of a 0-good triple,
2. $n - 2$ elements occur as the right endpoint of a 0-good triple, and
3. the number of 0-good triples equals $\lfloor n/2 \rfloor (\lceil n/2 \rceil - 1)$
i.e., for even n : $n/2 (n/2 - 1)$; for odd n : $((n - 1)/2)^2$

then the transactions in $Trans(p)$ are *equidistant*. The values in 1, 2 and 3 are maximal, as is their sum.

Proof We proceed from the right (formally by induction). The end of the sequence $Trans(p) = (T_1, T_2, \dots, T_n)$ looks like:

...	$\mathcal{L}\&\mathcal{R}$	$\mathcal{L}\&\mathcal{R}$	$\mathcal{L}\&\mathcal{R}$	$\mathcal{L}\&\mathcal{R}$	$\mathcal{L}\&\mathcal{R}$	\mathcal{R}	\mathcal{R}
...	T_{n-6}	T_{n-5}	T_{n-4}	T_{n-3}	T_{n-2}	T_{n-1}	T_n
...	6	5	4	3	2	1	0

Here \mathcal{L}/\mathcal{R} denotes: this T_i is a left/right endpoint in a 0-good triple; the numbers beneath the T_i 's indicate the number of times T_i is the middle of a 0-good triple.

First observe T_{n-2}, T_{n-1} and T_n , where T_{n-2} is a left endpoint of a 0-good triple; this implies that $distance(T_{n-2}, T_{n-1}) = distance(T_{n-1}, T_n) = a$ for some a .

Now suppose we have the following situation: $T_i = L$ (with $i \geq \lfloor n/2 \rfloor$) is the left endpoint of a 0-good triple (L, M, R) , for some $M = T_j$ with $j > i$; furthermore $a = distance(T_\ell, T_{\ell+1})$ for all $\ell > i$. Now T_j occurs $n - j$ times as middle of 0-good triples, whose right endpoints are the consecutive T_{j+1}, \dots, T_n . We can conclude that $distance(T_i, T_{i+1}) = a$. So we have $distance(T_\ell, T_{\ell+1}) = a$ for $\ell = \lfloor n/2 \rfloor, \lfloor n/2 \rfloor + 1, \dots, n$.

Similarly, using the right endpoints, one can show that $distance(T_\ell, T_{\ell+1}) = b$ for some b ($\ell = 1, 2, \dots, \lfloor n/2 \rfloor$). Using $\ell = \lfloor n/2 \rfloor$ we see that $a = b$. □

Example 1 Suppose we have the following 9 itemsets in our dataset:

- transaction 1: $\{A, B, C\}$
- transaction 2: $\{D, C\}$
- transaction 3: $\{A, B, E\}$
- transaction 4: $\{E, F\}$
- transaction 5: $\{A, B, F\}$
- transaction 6: $\{E, F\}$
- transaction 7: $\{A, B, F\}$
- transaction 8: $\{E, F\}$
- transaction 9: $\{A, B, C\}$

As distance we take the number of intermediate transactions. The stability value (with $w = 0$) of $\{A, B\}$ is $4 + 3 + 3 = 10$, the maximal value possible: there are 4 0-good triples; we have 3 transactions that are left (right) endpoint of a 0-good triple (see picture below, left; dots indicate the transactions that include $\{A, B\}$). If we insert two transactions $\{E, F\}$ between transaction 1 and 2, and also two between 8 and 9, we still have 4 0-good triples, but now we only have 2 transactions that are left (right) endpoint of a good 0-triple (see picture below, right), leading to stability value $4 + 2 + 2 = 8 < 10$. This example shows that condition 3 from the Theorem is in itself not sufficient yet in order to guarantee equidistance.



2.2 Algorithm

We now consider algorithms that find all stable patterns, given a dataset. Thanks to the APRIORI property many efficient algorithms do exist. However, the really fast ones rely upon the concept of FP-TREE or something similar, which does not keep track of stability. This makes these algorithms hard to adapt for discovering stable patterns.

ECLAT [10] is a fast algorithm that does not make use of FP-TREES; it grows patterns recursively while remembering which transactions contained the pattern, making it very suitable for our purpose. In a recursive step only these transactions are considered when counting the occurrence of a pattern. All counting is done by using a matrix and patterns are extended with new items using the order in the matrix. This can easily be adapted to incorporate stability.

Now suppose that $Trans(p) = T^{parent}$, with $n = |T^{parent}|$, is the ordered series of transactions (augmented with their index numbers from the *original* dataset) that contain itemset p . The algorithm below (Algorithm 1) will calculate the stability value when adding a new *item* to p . The algorithm will also calculate the support and the new series of transactions T^{child} that will be considered in the next step of a frequent pattern mining algorithm: the ECLAT algorithm is extended to STABLECLAT. The *child* is the *parent* itemset p extended with the new *item*. Note that *Left* and *Right* are *sets*. In line (9) we add the *index numbers* of the transactions. The function $contains(trans, item)$ checks if the transaction $trans$ contains the item $item$, the function $has(T^{parent}, index)$ verifies that transaction $index$ is in T^{parent} ; T_{index} is the transaction as retrieved from the original dataset. The

mindepth threshold defines from which depth the stability should be calculated — otherwise, for small itemsets with large supports the computation would become cumbersome. The *depth* is the recursive depth that is equal to the size of the child pattern that we are considering.

Algorithm 1 Stability Value

```

1: support := 0, stable := 0, Left :=  $\emptyset$ , Right :=  $\emptyset$ ,  $T^{child}$  := empty series,  $i := 1$ 
2: while  $i \leq n$  do
3:   if contains( $T_i^{parent}$ , item) then
4:      $T^{child} := T^{child}$  with  $T_i^{parent}$  appended, support := support + 1
5:     if depth  $\geq$  mindepth then
6:        $j := i + 2$ 
7:       while  $j \leq n$  do
8:         if contains( $T_j^{parent}$ , item) then
9:            $middle := (T_i^{parent} + T_j^{parent}) \bmod 2$ ,  $index := (T_i^{parent} + T_j^{parent}) / 2$ 
10:          if  $middle = 0$  and has( $T^{parent}$ , index) and contains( $T_{index}$ , item) then
11:            stable := stable + 1
12:             $Left := Left \cup \{T_i^{parent}\}$ ,  $Right := Right \cup \{T_j^{parent}\}$ 
13:          end if
14:        end if
15:         $j := j + 1$ 
16:      end while
17:    end if
18:  end if
19:   $i := i + 1$ 
20: end while
21: stable := stable +  $|Left|$  +  $|Right|$ 

```

This algorithm will only increase *stable* if the pattern is *exactly* in the center of two transactions containing the pattern (so $w = 0$); this can be easily generalized. It is possible that the pattern doesn't occur in the center transaction but in a transaction that is very near. A threshold $w > 0$ can be specified, as in the previous section. Suppose T_i is the outer left transaction and T_j is the outer right transaction, then we consider every T_ℓ , where $i < (i + j)/2 - w \leq \ell \leq (i + j)/2 + w < j$. Now our algorithm needs to check if the pattern occurs in one of these transactions.

Example 2 Suppose we have the same 9 transactions as in the previous example. If $w = 1$ then the stability value of $\{A, B\}$ will be $10 + 2 = 12$: transactions 1 and 7 (and 3 and 9) are now also endpoints of a 1-good triple.

The maximal stability value depends on the size of the database. This makes setting the minimal stability threshold *minstable* somewhat difficult. To make it easier one only needs to give a number *dist*, where $dist > 0$. With *dist* we calculate the *stable* value if the distance between all transactions containing the pattern is precisely *dist*. In this calculation we disregard the count of the left and right endpoints. This *dist* can now be used to propose a reasonable value for *minstable*, where D is the original dataset:

$$minstable = \binom{\lceil |D|/dist \rceil}{2}$$

Most frequent itemsets or patterns have a high stability value because it is more likely that a center transaction contains the pattern. However these patterns will not necessarily make a stable pattern more apparent. Furthermore it might also be contained in many transactions that don't form a stable interval. In order to solve this problem we can divide the stability value by the square of *support* and let $newstable = stable/support^2$. However we will lose the anti-monotone property, so this will only be useful as a post-processing step. We choose to divide by $support^2$ because *stable* can maximally become

$$\binom{support}{2} + 2 \cdot (support - 2) < support^2$$

In such a way we remove the influence of a high support on stability.

3 Results and Performance

The experiments were done for three main reasons, and are of exploratory nature. First of all by using the synthetic dataset we *show that patterns with a stable interval will be found*. Secondly with the website dataset we *show that the algorithm also finds good stable patterns for real problems*. And finally with a synthetic dataset and with the website dataset we *want to examine the efficiency of the algorithm compared to normal ECLAT*. Of course the normal ECLAT algorithm only finds frequent itemsets and no stable patterns. However the goal is to show the influence of the search for stable patterns on speed. Our implementation of ECLAT that discovers stable patterns is called STABLECLAT. All experiments were done on a Pentium 4 2.8 GHz with 512MB RAM.

The synthetic datasets can be seen as a supermarket that sells newspapers and credits for cell phones. The combination of the two is sold every day in the morning at least x times. The first dataset contains 1,000 transactions and 110 items. Of these 110 items 10 occur every 4 transactions. Also each item has a support of 200. From here on we call this dataset *news&credit small*. The second dataset contains 5,000 transactions and 110 items. Of these 110 items 10 occur every 10 transactions. Also each item has a support of 1,000. From here on we call this dataset *news&credit large*. The STABLECLAT algorithm was also tested on a real dataset. This dataset is based on an access log of the website of the Computer Science department of Leiden University, as said before. It contains all 1,991 items of the webpages that were visited, grouped in one hour blocks, so each of the 744 transactions contains the pages visited during one hour. This dataset will be called *website*.

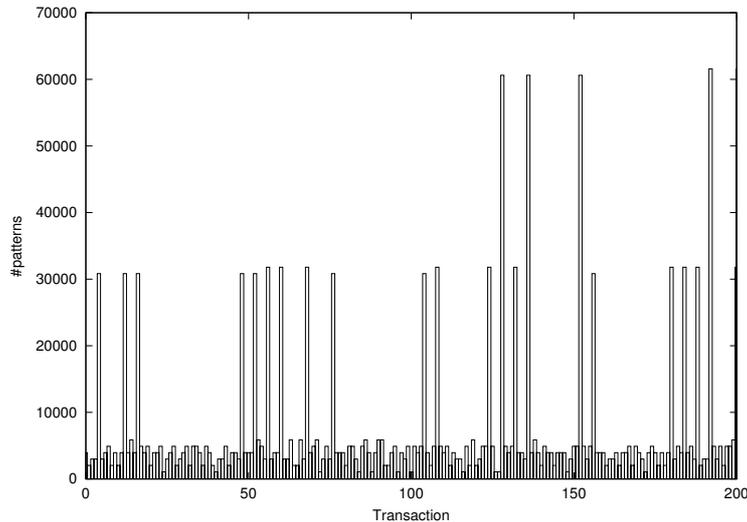


Figure 1: Occurrence graph of news&credit small using traditional support ($minsup = 25$)

Figure 1 and Figure 2 show an *occurrence graph*. For each transaction the number of frequent patterns contained by it are counted and plotted. The use of a minimal stability threshold will give less patterns because it filters out non-stable patterns. The items in the news&credit small dataset all occur 200 times, so their support is 200. Usually transactions are made up out of 20 randomly selected items. However every 4 transactions we randomly select 10 items and the remaining 10 items are the items that will be in the stable pattern. In this way there are several stable patterns, but also many unstable ones. Figure 1 shows that with only support we will not discover these stable patterns easily. When we use a minimal stability threshold (with the *dist* parameter from Section 2) we are able to see these patterns as we show with Figure 2. In all experiments we let $mindepth = 2$.

Figure 3 shows the occurrence of one pattern in the first 100 transactions from the website dataset. Transaction 0 is on Saturday 1 May 2004 5:00 and transaction 100 is on Wednesday 5 May 2004 9:00. First we searched the patterns with minimal stability value $\binom{744}{2}$. Then we selected one pattern where the fraction $stable/support^2$ is maximal. The figure shows a regularity in the occurrence of the selected pattern, which consists of four personal homepages and a main page.

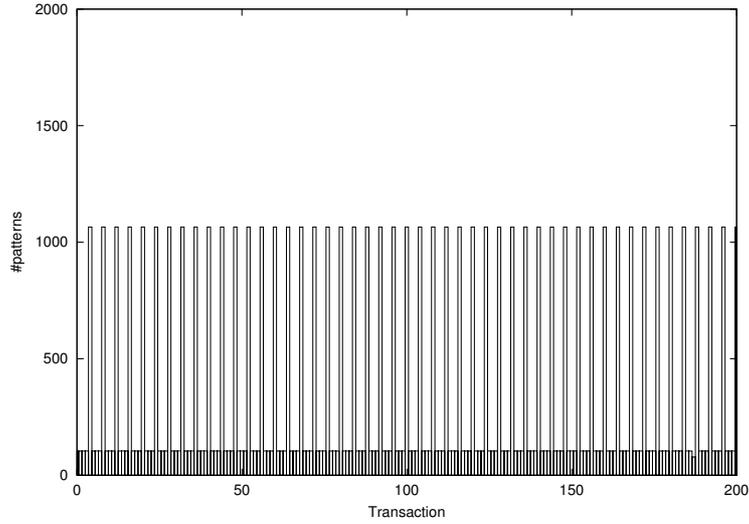


Figure 2: Occurrence graph of news&credit small using traditional support *and* stability ($minsup = 25$, $dist = 20$)

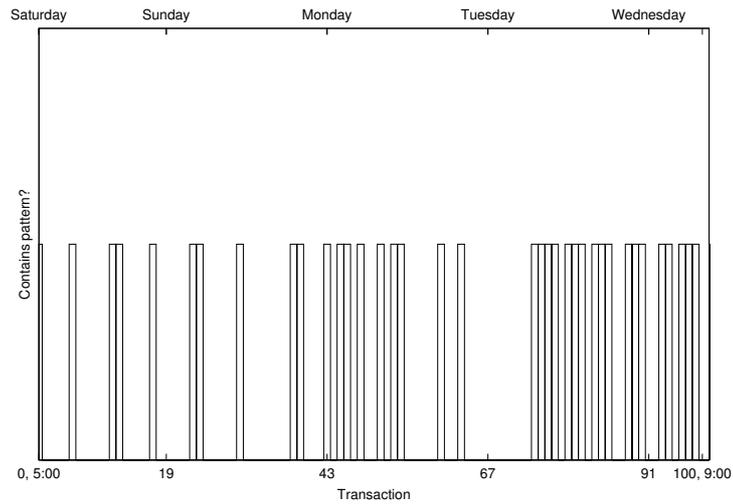


Figure 3: The occurrence of one stable pattern for website ($minsup = 100$, $dist = 7$)

	news&credit small	news&credit large	website
<i>support only</i>	1	36	179
<i>with stability</i>	2	140	251

Table 1: Time in seconds needed to mine each of the three datasets ($minsup = 100$, $dist = 4$)

Table 1 gives an indication of the influence of stability calculation on the speed of the algorithm. The news&credit large dataset shows a clear slowdown because of the occurrence of more frequent patterns. This means that more combinations have to be checked for a pattern occurring in the center transaction of a series of transactions, which requires an amount of time that is quadratic in the number of elements in this series.

4 Conclusions and Future Work

When we use stability in our search for patterns, we are able to find patterns that occur with some regular interval. The measure we proposed in this paper still enables us to prune using anti-monotonicity.

Using the distance between transactions like it is done in this paper is an interesting area of research. In the future we want to examine new measures that would enable us to visualize other types of behavior. Also we want to see if we can speed up the search for stable patterns, e.g., by using heuristics or by improving the stability measure. Furthermore, we would like to compare our approach with post-processing methods.

Acknowledgments This research is carried out within the Netherlands Organization for Scientific Research (NWO) MISTA Project (grant no. 612.066.304). We would like to thank Carlos Soares.

References

- [1] Agrawal, R., Imielinski, T., Srikant, R.: *Mining Association Rules between Sets of Items in Large Databases*. In Proc. ACM SIGMOD Conference on Management of Data (1993), pp. 207–216.
- [2] Chen, Y., Dong G., Han J., Wah, B., Wang J.: *Multidimensional Regression Analysis of Time-series Data Streams*. In Proc. 28th Int. Conference on Very Large Data Bases (VLDB 2002), pp. 323–334.
- [3] Graaf, E.H. de, Graaf, J.M. de, Kusters, W.A.: *Using Consecutive Support for Genomic Profiling*. ECML/PKDD-2006 Workshop on Data and Text Mining for Integrative Biology, Berlin, Germany, 2006.
- [4] Graaf, E.H. de, Kusters, W.A.: *Efficient Feature Detection for Sequence Classification in a Receptor Database*. In Proc. Seventeenth Belgium-Netherlands Conference on Artificial Intelligence (BNAIC 2005), pp. 81–88.
- [5] Graaf, J.M. de, Menezes, R.X. de, Boer, J.M., Kusters, W.A.: *Frequent Itemsets for Genomic Profiling*. In Proc. 1st International Symposium on Computational Life Sciences (CompLife 2005), LNCS 3695, Springer, pp. 104–116.
- [6] Giannella, C., Han, J., Pei J., Yan, X., Yu, P.: *Mining Frequent Patterns in Data Streams at Multiple Time Granularities*. In Proceedings of the NSF Workshop on Next Generation Data Mining (NGDM 2002), pp.191–210.
- [7] Li, Y., Ning, P., Wang, X.S., Jajodia, S.: *Discovering Calendar-based Temporal Association Rules*. In Proc. of the 8th Int Symposium on Temporal Representation and Reasoning (TIME 2001), pp. 111–118.
- [8] Steinbach, M., Tan, P., Xiong, H., Kumar, V.: *Generalizing the Notion of Support*. In Proc. 10th Int. Conf. on Knowledge Discovery and Data Mining (KDD 2004), pp. 689–694.
- [9] Teng, W., Chen, M., Yu, P.S.: *A Regression-based Temporal Pattern Mining Scheme for Data Streams*. In Proc. 29th Int. Conference on Very Large Data Bases (VLDB 2003), pp. 93–104.
- [10] Zaki, M., Parthasarathy, S., Ogihara, M., Li, W.: *New Algorithms for Fast Discovery of Association Rules*. In Proc. 3rd Int. Conf. on Knowledge Discovery and Data Mining (KDD 1997), pp. 283–296.

Question Answering for Dutch: Simple does it

Arjen Hoekstra Djoerd Hiemstra Paul van der Vet Theo Huibers

Faculty of Electrical Engineering, Mathematics and Computer
Science, University of Twente, P.O. Box 217 7500AE Enschede,
ahoekstr@cs.utwente.nl, hiemstra@cs.utwente.nl,
p.e.vandervet@utwente.nl, t.w.c.huibers@utwente.nl

Abstract

When people pose questions in natural language to search for information on the web, the role of question answering (QA) systems becomes important. In this paper the QA-system simpleQA, capable of answering Dutch questions on which the answer is a person or a location, is described. The system's algorithm does not use a lot of complex NLP-techniques, but instead uses the magnitude of and redundancy on the World Wide Web to its advantage. The system has been evaluated on the DISEQuA corpus and performed quite well: MRR near 0.5. For further improvements it can easily be extended by adding more rewrite rules and applying more sophisticated filtering and tiling.

1. Introduction

Most information need on the Internet is satisfied by sending queries to web search engines. These queries normally aren't posed in natural language, but by providing some keywords. However, there is a part of the web search engine users that does use natural language as a query. Search engine log files show that 1% of ten million queries are posed as questions in natural language, for instance "Where does Sean Hannity live?" [6]. If this habit becomes more and more common the need for systems that can interpret questions posed in natural language becomes more important. Systems that can fulfil this task are called Question Answering (QA) systems.

Nowadays, the QA-systems to which much of the research is devoted, are easily becoming quite complex, as much Natural Language Processing (NLP) is involved. One can think of part-of-speech tagging, parsing, use of grammars and lexicons and much more. But how much can be achieved if not a lot of fancy NLP techniques are used, but a relatively simple algorithm instead? Dumais, Banko and Brill investigated this by developing an algorithm that does not use much NLP, but uses the gigantic size of the Web to their advantage [2]. Their system was able to answer English factoid questions formulated in natural language.

The main goal of this research is to examine whether their method is a feasible one, only this research will focus on Dutch questions instead of English. This study is limited to questions on which the answer type is a person or a location.

In the following section the method used to design and implement the simpleQA-system is described. Afterwards the results of an evaluation performed on the system are presented and discussed. Finally conclusions will be drawn and recommendations for future work will be made.

2. The simpleQA system

The simpleQA system was designed making use of the algorithm as described in [2]. This algorithm is composed of the following steps:

1. Question analysis
2. Query rewriting/weighting
3. Web retrieval
4. N-gram mining and filtering
5. N-gram tiling
6. Pick top N-gram

The system consists of five core components: Analyzer, Retriever, Extractor, Tiler and Selector. These components and their relation to the algorithm steps will be described in the next paragraphs. Some parts of the architecture of a QA-system from van Langen [4], personQA, were reused for the development of simpleQA.

2.1. Analyzer

The purpose of the analyzer is to convert a question in multiple queries which can be sent to a search engine. This is accomplished by analyzing what kind of question is posed and subsequently rewriting the question in multiple queries and weighting each of these queries.

Question analysis

Two types of questions can be posed: questions on which the answer is a person or questions on which the answer is a location. The system does not have to distinguish between these kinds of questions, because the answer type is the same (a named entity). But it has to recognize what kind of pattern the question contains. Patterns are recognized by making use of regular expressions, using only limited knowledge of natural language. Only question words and auxiliary verbs are recognized. Based on this information, positions of VP's and NP's in the sentence are determined. No grammar or part-of-speech tagging is used. The possible Dutch person question patterns that can be distinguished can be found in Table 1. One might think of other kinds of Dutch person questions (like "Hoe

luid(t)de de ([Adj/Prefix]) naam van [NP]?)” which are not covered by the rewrite rules of the system. The reason for this is that those are considered as old fashioned, they are not used much anymore in everyday conversations and hence it will not be likely they will be used when posing questions. However the system can handle these questions by making a simple “AND” query of the non-stop words. It just does not add any other queries.

The possible location question patterns that can be recognized can also be found in Table 1. The pattern “Waar vond [NP] plaats?” is an example of a more aimed pattern. It also fits in “Waar [VP]?” but by explicitly recognizing it, better queries can be made. In the future the same can be done for queries like “Waar bevindt [NP]?” or “Waar ligt [NP]?” to boost the system’s performance. A second thing that could catch the eye is that some location patterns much resemble person patterns. “Welk(e) [NP] is [NP]?” and “([Prep]) welk(e) [NP] [VP]?” are even exactly the same and thus merged in the question analysis part. Also “Wat is [NP]?” is merged because it is almost the same as “Wie is [NP]?”. As the question words are stripped of the question and both are three letter words the same query rewriting can be applied.

Table 1. Person and location question patterns

Person question patterns	Location question patterns
Wie is [NP]?	Wat is [NP]?
([Prep]) wie [VP]?	Welk(e) [NP] is [NP]?
Welk(e) [NP] is [NP]?	([Prep]) welk(e) [NP] [VP]?
([Prep]) welk(e) [NP] [VP]?	Waar vond [NP] plaats?
Hoe heet(te) [NP]?	Waar [VP]?
Wat is/was de ([Adj]/[Prefix]) naam van [NP]?	

Query rewriting and weighting

For each question one or more queries are constructed. Two other variables are added to the query: the direction in which the answer is expected and the weight of the query. Direction can be “LEFT”, “RIGHT” or “ANYWHERE”.

Direction is only added when it is absolutely certain the answer must be in that direction. For example when the question “Wie is de minister-president van Nederland?” is posed, the query [“de minister-president van Nederland is?”] is constructed. One would expect the answer to be to the right of this query. However, there can well be web pages which contain something like “Jan-Peter Balkende, de minister-president van Nederland is vanochtend aangekomen in...”. When in this case the direction “RIGHT” is added the correct answer will not be found in this snippet. Hence instead of “RIGHT”, “ANYWHERE” is applied to the query.

The weight component is an integer value between 1 and 5 and is determined by the likelihood of finding a correct answer with the query. For a question like “Wie was president van de Verenigde Staten tijdens de Tweede Wereldoorlog”

it is more likely to find a correct answer with [“was de president van de Verenigde Staten tijdens de Tweede Wereldoorlog”] than with [president AND Verenigde AND Staten AND tijdens AND Tweede AND Wereldoorlog]. Weights will be used later on to score possible answers. All query details units (query itself, direction and weight) constructed for a question are passed on to the retriever.

2.2. Retriever

The Retriever implements the algorithm’s third step: **web retrieval**. The search engine used for the QA system is Google. With help of the Google API [3] a maximum of 1000 queries a day can be sent to Google, which is able to return the data found by the queries in different formats. The query received from the Analyzer is sent to Google. The web pages found are restricted to pages in Dutch and a maximum of forty pages is returned. This is less than the number of pages that yielded the best results for [2], which were 200 pages. But because the collection of Dutch web pages is significantly smaller than the English one, the number of pages should be less. Otherwise too many nonsense answers will be retrieved. For each web page a snippet, which contains all of the query words, is extracted. Only unique URL’s are investigated, as each query is weighted, it does not matter whether a different query returns the same web page. High weighted queries are the first ones to be sent, so it is guaranteed the highest weight is applied to a web page. All unique URL’s are sent, together with their weights and answer directions from the analyzing part, to the Extractor.

2.3. Extractor

The Extractor’s goal is to **mine N-grams** from the snippets and to **filter these N-grams**. An N-gram is just a sequence of N words. It returns a set with possible answers mapped on their weights. Each snippet passed through by the retriever is examined by the Extractor. The snippet is searched for possible answers in the direction provided by the query details. Possible answers are then filtered. Named entity recognition is performed by matching regular expressions. When a duplicate answer is found, its score is added to the score of the existing answer. Finally one other filter is applied to each answer: when an answer is contained in the question it is considered as an unwanted answer and removed from the map. The candidate answers are now further examined by the Tiler.

2.4. Tiler

The Tiler performs the **N-gram tiling** part of the algorithm: answers with matching components are merged. To accomplish this, the Tiler runs through

the set of possible answers. When two different answers contain the same components they can be tiled. For instance, when possible answers are ‘Balkenende’ and ‘Jan Peter Balkenende’ these will be combined into one answer ‘Jan Peter Balkenende’. The actual tiling is only performed when the score of the answer which contains the other answer is higher than the score of the other answer itself. Otherwise there would be a high chance on bad tiling. The map of tiled answers is now passed on to the Selector.

2.5. Selector

The selector creates a ranked list of ten answers where the **best ranked can be picked as the actual answer** on the question. Answer formulation only consists of the answer itself. Parts of the question could be included (by distinguishing patterns again) in the answer formulation, but this is quite trivial and does not make answers on this kind of factoid questions more valuable.

3. Evaluation

In this section the evaluation of the simpleQA-system will be discussed. The first paragraph describes the used evaluation method, the second paragraph presents the results and in the final paragraph a brief discussion of the results will be given.

3.1. Method

To evaluate the developed system two question sets (a person set and a location set) were presented to the system. The questions originated from the DISEQuA corpus [5], which was developed for the Cross Language Evaluation Forum (CLEF). Because the answer type of each question was marked in this corpus, person and location questions could easily be extracted from it. Another advantage of using this corpus is that van Langen’s personQA-system [4] and Bouma’s system Joost [1] also used this one. Hence the systems can be compared well by using the same test corpus. Just like the evaluation of personQA, person answers are also considered correct when only the last name is correct. For instance, for the question “Wie is de Italiaanse minister van buitenlandse zaken?”, “Franco Frattini” and “Frattini” are both considered to be correct answers.

3.2. Results

The test results contained questions for which no answers were available in DISEQuA. These questions were removed leaving 87 person questions (from a total number of 90 in DISEQuA) and 72 location questions (from a total number of 85 in DISEQuA). The person questions are compared to the results of

personQA [4] and Bouma’s system Joost [1]. The Joost system was evaluated more restrictive (only exact answers correct and only the top five answers were considered).

From Table 2 one can conclude the scores of van Langen’s personQA system are improved by implementing our algorithm. The simpleQA system achieves an MRR of 0.4844 opposed to 0.2909 of personQA. When applying a paired sample t-test the difference between the MRR’s proves to be significant at the .01 level. Also the percentage of answers found and the percentages first answers correct are higher than those of personQA. Compared to Joost, the MRR, the percentage of first and found answers are all slightly lower. Because further results of Joost were not available the significance of the MRR difference could not be determined.

The location question results are slightly lower than those of the person questions, but they are in the same range. For the location results no systems were found which performed a similar evaluation.

Table 2. Results on the CLEF person and location question set

	Person questions			Location questions
	simpleQA	personQA	Joost	simpleQA
N	87	87	87	72
MRR	0.4844	0.2909	0.5804	0.4350
Answers found	63.2 %	46.0 %	65.5 %	56.9 %
First answer correct	40.2 %	20.7 %	52.6 %	36.1 %

The Microsoft researchers [2], who used the same kind of algorithm as simpleQA, tested on an English TREC-9 question set and thus cannot be compared in a good way. It does, just like Joost, only consider the top five ranked answers and their test set contained more kinds of factoid questions than just location and person. It obtained an average MRR of 0.507 and found answers on 61% of the questions.

3.3. Discussion

In general the simpleQA system seems to perform quite well. It improved the score of personQA significantly and got a lot closer to scores of Joost. Of course it can still be improved. When analyzing the results some issues attract attention.

The first one is the number of “welk(e)” (which) – question that are answered incorrectly. This is 76 % of the total of person “welk(e)”-questions and 44% of the location ones. An explanation for this is that there are not many rewrite rules for this kind of questions, because one would need to apply more NLP to determine the parts of speech. For example, one would like to rewrite a question like “Welke Franse ex-minister werd tot een gevangenisstraf veroordeeld wegens corruptie?” into [“werd tot een gevangenisstraf veroordeeld

wegens corruptie” AND “Franse ex-minister”]. But to accomplish this one would have to know where the NP starts and ends. This would make things much more complicated. One could for instance determine where the verb is in the sentence to determine the end of the NP, but for this part-of-speech tagger would be needed. So, because there were less rewrite rules and hence not so many queries for “welk(e)”-questions, the system performs a less focused search, which results in worse answers.

Another matter that catches the eye is the way of responding on questions that require a very specific location as an answer. A lot of these questions were answered incorrectly. One logical explanation of this issue is that because the answer has to be very specific it is hard to find exactly that answer. An answer like “Pretoria” (on the question “Wat is de hoofdstad van Zuid-Afrika?”) is easier to find than “25 kilometer ten westen van Bonn” (answer of the question “Waar ligt Euskirchen?”), because it is less specific. A second explanation for this matter is the use of the named entity filter causing to miss some answers. Because it is quite restrictive to avoid nonsense answers (for example by discarding answers starting with stop words), it could also remove this kind of answers. In the future the filter could be made somewhat more intelligent to adapt itself for this kind of questions.

Questions which the system can handle very well most of the time are questions like “Wie is de president van....?” or “Wat is de hoofdstad van...?”. For this kind of questions a lot of web pages are available and hence there is more information available to base an answer on.

4. Conclusions and future work

The main goal of this research was to determine whether the approach to use data redundancy on the Web for question answering, as described by Dumais, Banko and Brill [2], is also suitable for Dutch questions. To investigate this, their algorithm was implemented in the core of a former made QA-system by van Langen [4]. Subsequently the developed system, simpleQA, was evaluated on a CLEF corpus. Evaluation proved simpleQA significantly outperforms van Langen’s system. Bouma’s system, Joost [1], performs better, but this system uses a full-strength Dutch dependency parser. Whereas Bouma claims that advances in QA should come from “strengthening the Dutch NLP infrastructure”, we believe that there is a lot of room for improvement using simple retrieval techniques that take advantage of the redundancy on the World Wide Web.

The scores between the Microsoft system and simpleQA cannot be compared one on one, because the systems capabilities and the evaluation methods are not the same. However, despite these differences, the scores of simpleQA and the Microsoft system seem quite similar. The Microsoft system method was more

restrictive, but used more NLP than the simpleQA system does. So roughly can be said the simpleQA system performs rather well, even though it has a smaller set of web pages available (the Dutch collection of web pages is a lot smaller than the English one).

Recommendations for future work to improve the performance of simpleQA are to extend the number of rewrite rules and to apply more sophisticated filtering and tiling.

More rewrite rules can be implemented by making minimal use of some NLP techniques, for instance a lexicon (e.g. Dutch WordNet) or a part-of-speech tagger. A lexicon would be convenient for making passive form queries or to identify other relations between words. A part-of-speech tagger could help to determine where parts-of-speech begin and end, which would for instance be of use when rewriting “welk(e)”-questions.

The actual filtering and tiling parts of the system certainly help it to perform better, but they can be improved. At the moment filtering is implemented in the same way for each question, but it could be more focused on the kind of question. Also with regard to the tiling part can be made some progress. At the moment only answers that exactly contain smaller answers are merged, but also answers that would differ in one or two characters could be merged (e.g. Tokyo and Tokio). This would contribute to the performance of the system.

References

- [1] G. Bouma, Question answering for Dutch using dependency relations, September 2003, Groningen, the Netherlands, http://odur.let.rug.nl/~gosse/Imix/project_description.pdf.
- [2] S. Dumais, M. Banko, E. Brill e.a., Web Question Answering: Is more always better?, in *Proceedings of the 25th Annual International ACM Sigir Conference on Research and Development in Information Retrieval (SIGIR 2002)*, pp. 291-298, Microsoft Research, 2002.
- [3] Google Web APIs, <http://www.google.com/apis/>.
- [4] M.C.G. van Langen, Building a system for answering Dutch Person Questions, in *Proceedings of the 2nd Twente Student Conference on Information Technology*, pp 134-141, January 2005.
- [5] B. Magnini, S. Romagnoli, A. Vallin e.a., Creating the DISEQuA Corpus: a Test Set for Multilingual Question Answering, In Working Notes for the CLEF 2003 Workshop, Trondheim, Norway, August 2003.
- [6] M. de Rijke, Question Answering: Now and Next, Slides Invited Talk, University of Twente, October 2005.

Analysing Clickstream Data: From Anomaly Detection to Visitor Profiling

Peter I. Hofgesang Wojtek Kowalczyk

Vrije Universiteit Amsterdam, De Boelelaan 1081A, 1081HV Amsterdam

Abstract

The paper presents results of analysis of clickstream data used in the ECML/PKDD Discovery Challenge 2005. We focused on two aspects: detection of anomalies and profiling visitors of internet shops. Several unusual patterns were discovered with the help of simple tools such as frequency tables, histograms, etcetera. Some insight into click-behaviour of web visitors was obtained with the help of finite mixtures of multinomial distributions and a graphical representation of navigation trees.

1 Introduction

During the 16th European Conference on Machine Learning (ECML) and the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), both held in October, 2005 in Porto (Portugal), a traditional Discovery Challenge took place, [1]. One of the tasks consisted of analyzing log data from web servers of several internet shops. The main objective of the analysis was to demonstrate “original uses or combinations of data mining methods on the data sets” or to gain “new insights in the application domain thanks to a data mining approach.”

In this paper we present some results of our analysis that concern two issues: detecting anomalies in the data and profiling of visitors.

We used standard statistical and data mining techniques for exploratory data analysis and anomalies detection. A model of finite mixtures of multinomial distributions [2] was applied to the session data to extract user profiles for further analysis. Furthermore we used a tree-like visualisation of profile sequences to reveal hidden sequential information about the user sessions.

The rest of this paper is organised as follows. In Section 2 we review the essential results of the exploratory data analysis and describe the anomalies found in the data and the necessary steps performed for the further analysis. In Section 3 we present the process of identification of user profiles and a tree-like visualisation of profile sequences. Finally, Section 4 contains our conclusions.

2 Exploratory Data Analysis

The original data set contains about 3.5 million records that reflect internet traffic registered during a period of 25 days by websites of 7 internet shops. Records can be grouped with help of the provided session identifier field (SID) into 522410 unique sessions that originate from 79526 different IP-addresses. The distribution of sessions and visited pages over seven shops is shown in Table 1:

As we can see the most frequently visited shop was the one with home appliances, perhaps due to the broadest selection of offered product categories, whereas the shop with computers was least popular. It is also interesting to notice that the shop selling only MP3 players (thus a product from a single product category) had a relatively large number of visits.

To get an insight into the geographical location of visitors we translated IP-addresses into country codes and looked at their distribution. As expected, it turned out that most IP-addresses are from the Czech Republic (50238) and neighbouring countries: Slovakia (1991), Poland (492), Germany (343), The Netherlands (240), Italy (231), and Hungary (210). In general, about 10% of the traffic is generated from abroad.

ShopId	Assortment	#sessions	#pages	#pages/session
10	Cameras	80740	509688	6.3
11	Audio	57086	400045	7.0
12	TV's, Video	78845	645724	8.2
14	Home Appliances	156285	1290870	8.3
15	MP3 players	67987	308367	4.5
16	Mobile Phones, PDA's	53305	298030	5.6
17	Computers, Software	28179	164447	5.8

Table 1: Distribution of sessions and pages per shop.

2.1 Anomalies in the data

While analysing some basic properties of sessions (length, duration, internal consistency, etcetera) we encountered a number of problems. Some of these problems were minor and could be easily fixed or ignored, others were so serious that we decided to redefine the concept of a session.

2.1.1 Multiple IP-addresses per session

It is logical to expect that every session involves exactly one IP-address (“IP”). However, we spotted 3690 sessions with more than one IP-address. Most of these sessions involve 2, 3 or 4 different IPs (3051, 362, 113, respectively), but there are sessions with more than 20 IPs (for example, the session with SID = `9f412dc3a7bd7681e29125c66da3318f` refers to 22 different IP-addresses). More interestingly, some sessions use IP-addresses from several countries (e.g. SID = `4f840dcfbd33f940cee24e2d7d352ec7`).

2.1.2 Multiple shops per session

Although most sessions refer to 1 shop only (in 522398 cases) there are sessions that refer to 2, 3, or 5 shops (there are 9, 2 and 1 such sessions, respectively).

2.1.3 Very long sessions

In the data there are 476 sessions that lasted more than 24 hours; e.g., the session with SID = `35c97651004351765628dffe50209a18` lasted more than 18 days! (a closer look at this session revealed that it consists of two “normal” 2-3 minutes long sessions that were separated by a break of 18 days).

2.1.4 Very intensive sessions

Some sessions were very intensive: they contain many pages that are visited in a relatively short time. There are 2865 sessions with more than 100 pages, 19 sessions contain more than 1.000 pages, and there are 2 sessions with more than 10.000 visited pages. The longest session (with SID = `35c97651004351765628dffe50209a18`) contains 15454 pages that were visited in less than 7 hours. Most likely such sessions were generated by robots or spiders.

2.1.5 Frequent IP-addresses with short sessions

Some IP-addresses occur in numerous (short) sessions. For example, there are 29320 sessions, all taking place in less than 20 hours, that were originated from a single IP-address: `147.229.205.80`. Moreover, 5 IP-addresses were used in more than 10.000 sessions each, and another 372 IPs occurred in more than 100 sessions. Most probably, all these sessions were not generated by a human, but by a program that was fetching large collections of pages.

Length	Count (old)	Count (new)
1	318523	65258
2	24762	31821
3	17353	18828
4	15351	16332
5	15361	15509
6	13455	13448
7	10958	10883
8	9045	9095
9	7939	8070
10	7028	7091
...

Table 2: Distribution of the length of sessions before (old) and after recoding (new).

2.1.6 Sequences of short sessions

Almost 61% of the sessions are of length 1. However, when we order these sessions by the corresponding IP-address and time we will notice that in some cases they form larger sessions (visits from the same IP-address to the same shop, within a reasonably short time interval, for the same type of pages).

2.1.7 Parallel sessions

In some cases it seems that several sessions could be combined into a longer one. We have the impression that when a user opens a URL in a new window a new session ID is generated. This might lead to some errors in the modelling of the visitor’s behaviour. For example, we noticed that users often select a product they want to buy and then open a new window to complete the transaction. Unfortunately, in that case the transaction is recorded with a new session identifier and the context is lost.

It is likely that some of the above anomalies were caused by errors made during data preparation. Others might signal some security problems. Further investigation of these issues could be quite useful, provided more “background information” were available.

2.2 Data cleaning and pre-processing

In order to avoid problems listed in the previous paragraph we decided to redefine the concept of a session in the following way: a session is a chronologically ordered sequence of visits (records) that originate from the same IP-address, are related to the same shop, and do not contain gaps longer than 30 minutes (see [3]). Under this definition the number of sessions of length 1 was reduced from about 300.000 to 65.000, whereas the frequency of longer sessions remained almost the same (see Table 2).

For further analysis we decided to delete all sessions of length 1 as they would obfuscate the modelling process. Additionally, we removed about 12.000 sessions that were longer than 50 pages, because such sessions were most likely generated by robots, spiders, or special programs.

For the purpose of modelling sequences of visited pages we mapped the different page categories into integers. Table 3 shows the mapping of the 13 most frequent categories. The category 13 (“the rest”) refers to unknown URLs—URLs that were not specified in the description of the challenge.

3 Identification and analysis of user profiles

One of our main goals is to provide an insight into users’ behaviour on the websites. Therefore we clustered the users using a finite mixture model proposed by Cadez et al. (2001) [2]. We used the resulting clusters to establish different profiles for users and to label all visits by one of the profiles.

pageId	URL	description
1	/	main page
2	/ct/	product category
3	/ls/	product sheet
4	/dt/	detail of product
5	/znacka/	brands
6	/akce/	actual offers
7	/df/	comparison
8	/findf/	fulltext search
9	/findp/	parameters based search
10	/setp/	setting displayed parameters
11	/poradna/	on-line advice
12	/kosik/	shopping cart
13	the rest	the unknown URLs

Table 3: Mapping of content categories

Furthermore we analysed the sequences of profiles for each user over the whole period to get a better insight into how users behave throughout their visits. We present a transition matrix and a tree-like visualisation of the profile sequences. The tree visualisation shows the most typical profile paths over the whole dataset.

3.1 Profiles of users

In their paper Cadez et al. (2001) [2] proposed a generative mixture model for predicting user profiles and behaviours based on historical transaction data. A mixture model is a way of representing a more complex probability distribution in terms of simpler models. It uses a Bayesian framework for parameter estimation and the mixture model addresses the heterogeneity of page visits. Cadez et al. (2001) presented both a global and an individual model. In this paper we apply the global mixture model because of the relatively small number of sessions per users and because of the uncertainty of the IP addresses as unique user identifiers.

After the analysis of different setups for modelling we chose the four component mixture model as the most suitable for the given data. Each cluster is characterised by a vector of frequencies with which members of such a cluster visit specific pages. These frequencies can be visualised in a bar chart (see Figure 1)—a kind of a “group profile”. Additionally, the parameter alpha can be interpreted as the size of the cluster.

To interpret the clusters it is necessary to look at the page id – URL mapping table (Table 3). In Figure 1 we can identify four different user profiles. Three of these profiles cover some kind of product information acquisition while the fourth one describes a behaviour of (potential) buyers. Profiles, in order of appearance, are:

- Profile 1. General overview of the products. Users within this category are interested in a particular product category but they don’t have any specific product or brand preferences or they would like to take an overview of all the products in the category.
- Profile 2. Focused search. These users know exactly which products they are looking for. They search for specific products and brands.
- Profile 3. Potential buyers. This profile describes potential customers that already placed some products in their shopping cart. The profile also includes customers that actually finalise their transactions, i.e., buy the selected products.
- Profile 4. Parameter based search. Users within this category look for products with specific expectations in mind. They use parameter based search and compare different products.

It is interesting to look at (dynamic) relations between the identified profiles. In particular, we are interested in the following questions:

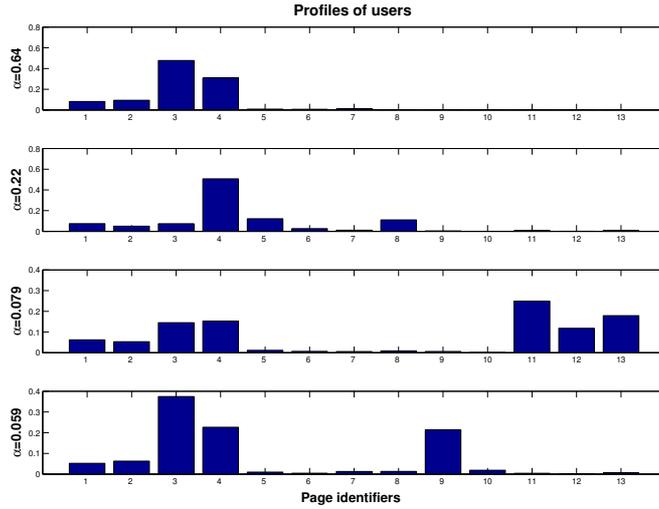


Figure 1: User profiles of the visitors of the web shop

	P1	P2	P3	P4
P1	0.7208	0.1592	0.0621	0.0579
P2	0.5908	0.2828	0.0710	0.0553
P3	0.5022	0.1616	0.2873	0.0489
P4	0.6000	0.1702	0.0685	0.1613

Table 4: Transition probabilities of profiles

- How do these profiles relate to each other over sessions from the same user?
- Did buyers buy before already?
- Do buyers come back after a purchase? If so, how do they behave afterwards?

To answer these questions we assigned the appropriate profile identifier to each session of the users and analysed the resulting profile sequences.

3.2 The transitions of profiles

Given the sequences of profiles for each user we can create a transition matrix of the profiles. We count the frequencies of all possible transitions given all the consecutive profile pairs of the sequences. By normalising the resulting frequency matrix we get the transition probabilities (Table 4). Each x_{ij} element of the table represents the probability of the transition from profile i to profile j ($i, j = 1 \dots 4$).

The most significant transitions are transitions to Profile 1. The high proportions of self-transitions show that many users do not change profile during their visits.

3.3 A tree-like visualisation of profile sequences

To get an insight into how users switch between the identified profiles we replaced their sessions by the corresponding profile types. In other words we created, for every user, a sequence of profile types. Then we analyzed these sequences with help of navigation trees that were introduced in [4].

Figure 2 shows the frequent profile sequences of users. The tree model consists of nodes with their profile labels in specific colours. There is a special virtual node—the root of all sequences. It contains additional information (name of the tree, support rate). Nodes are connected with lines (edges) in different thickness marking the frequencies of the given paths. Edges are labelled with the

percentage of (sub)sequences crossing the given node. The absolute number of sequences finished at the given node is given in parentheses. The size of the tree is controlled by the so-called support threshold. The tree contains only (sub)sessions that are more frequent than this given threshold.

Figure 2 shows that most of the users (68.1%) started their visits exhibiting the patterns of Profile 1 (general overview). Furthermore most of these users did not change profile during their visits. The second most relevant profile is Profile 2 (focused search), 20.6% of the users started their visits according to the patterns of this profile.

Figure 3 presents the frequent profile paths of users who had at least one “Potential buyers” profile during their visits. We see again that the general overview and focused search profiles are dominant in these sequences as well. Surprisingly, most of these users (52.9%) started their visits according to the potential buyer profile. This assumes that they bought products (or used the shopping cart) immediately during their first visit. Moreover, most of the users turned into a potential buyer after several visits of overviews of the products (Profile 1).

4 Conclusions

In the paper we applied several data mining techniques to the Challenge data. We used conventional tools (frequency tables, counts, distributions, etcetera) to develop an overall picture of the data and to spot several anomalies. We subsequently used mixture models to group visitors into 4 groups. Finally, we generated navigation trees to visualise typical navigation patterns of clients that approached the most interesting phase of any session: finalisation of the transaction. There are three main conclusions that can be drawn from our results.

First, using a simple mapping of IP-addresses to countries, we discovered that about 10% of the traffic was generated from countries around the Czech Republic. Therefore, it is worthwhile to consider internationalisation of the analysed web shops.

Second, we discovered several anomalies in the data. Some of them could be easily explained (e.g., by the fact that some sessions were generated by robots), others might be symptoms of more serious problems. For example, some anomalies may be caused by errors in scripts that are embedded in pages (a page may for example refer to itself), malicious attacks from outside, fraud attempts etcetera. Perhaps it would be meaningful to run systematic analysis of log files along the lines presented in Section 2.

Finally, we have identified 4 groups of visitors. The most significant group includes users that search for a general information on products from a specific product category. The second group of users is focused on details of concrete products. The third group includes users that use the parameter-based search engine to find products they are looking for. Finally, potential buyers that actually put products into their shopping carts and in some cases purchase them form the fourth group.

Knowledge of these groups may help in redesigning shops’ websites or just in understanding the different paths followed by visitors.

References

- [1] ECML/PKDD Discovery Challenge 2005. October 2005, Porto, Portugal.
<http://lisp.vse.cz/challenge/ecmlpkdd2005/chall2005.htm>.
- [2] Igor V. Cadez, Padhraic Smyth, and Heikki Mannila. Probabilistic modeling of transaction data with applications to profiling, visualization, and prediction. In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 37–46, New York, NY, USA, 2001. ACM Press.
- [3] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.
- [4] Peter I. Hofgesang. Web usage mining - structuring semantically enriched clickstream data. In *MSc. Thesis, Vrije Universiteit*, Amsterdam, The Netherlands, 2004.

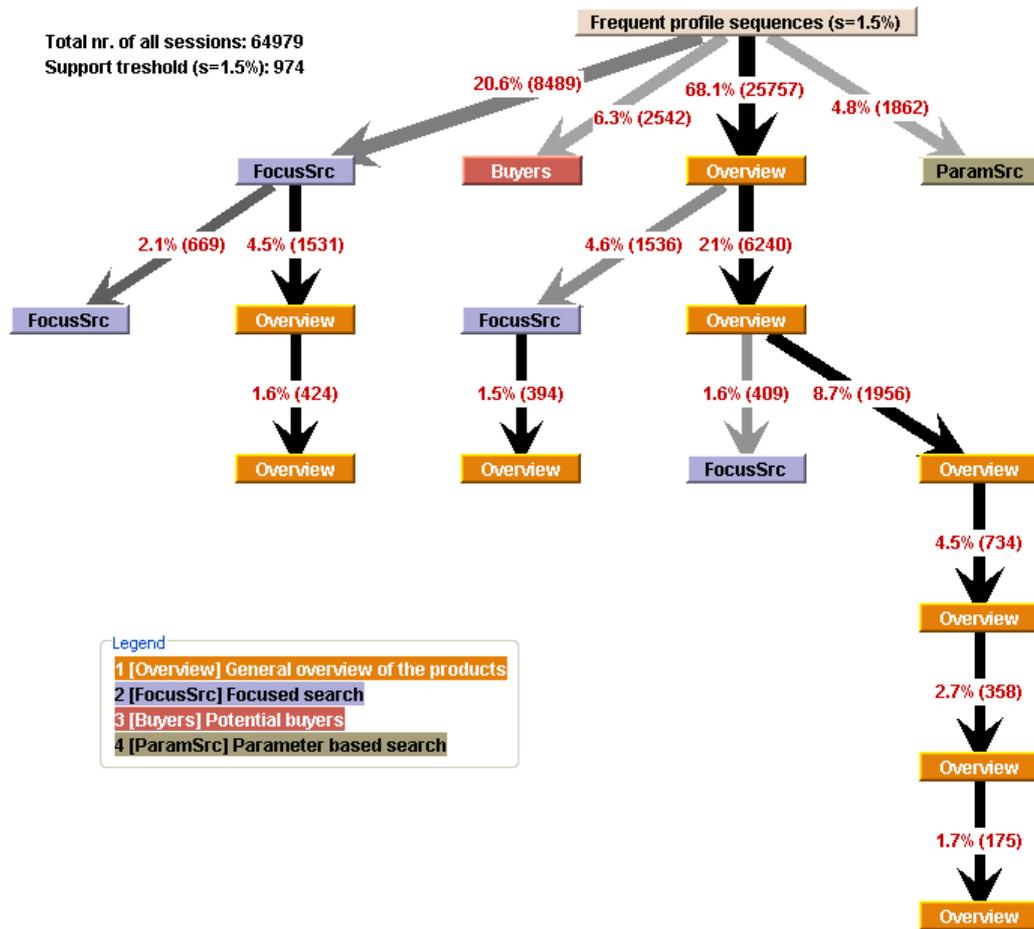


Figure 2: Tree-like visualisation of frequent profile sequences

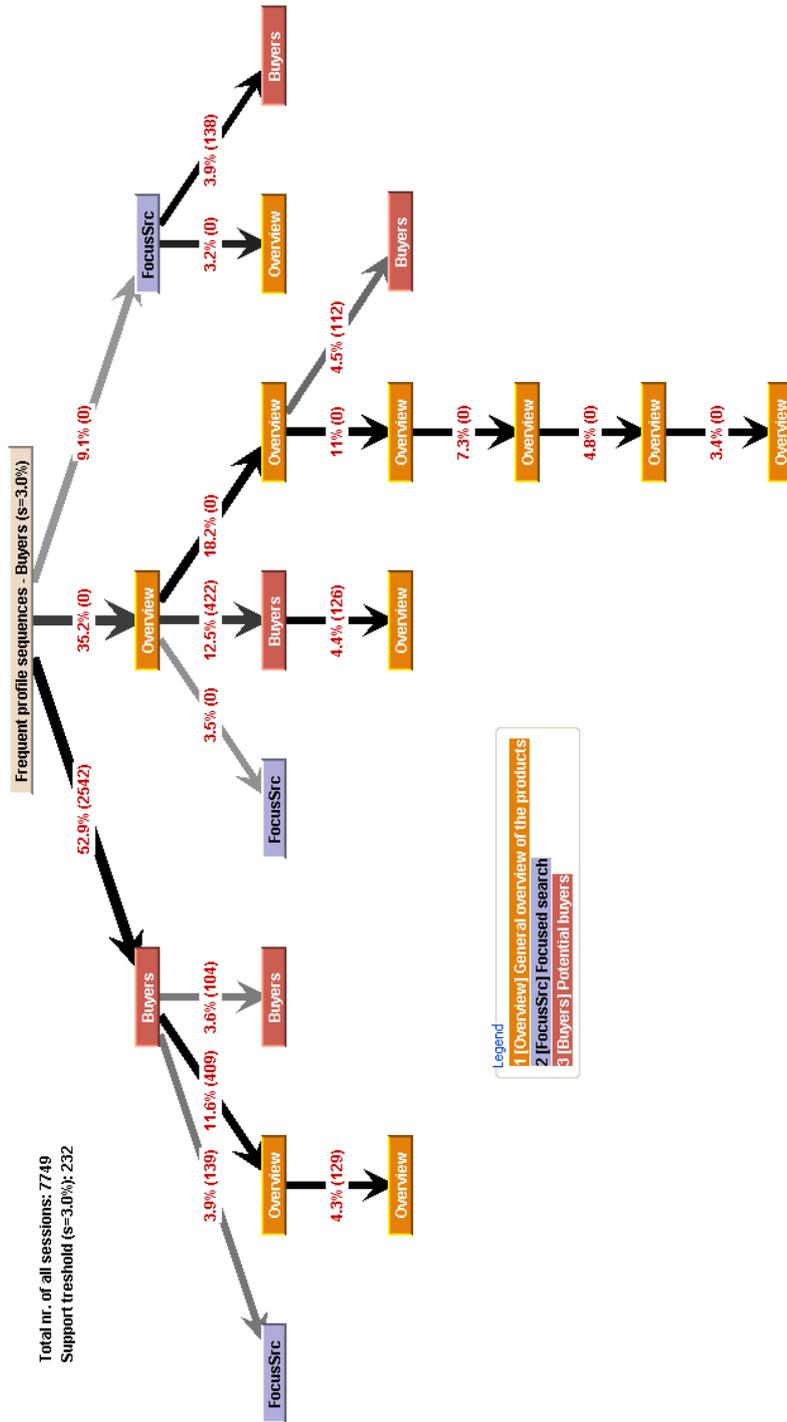


Figure 3: Tree-like visualisation of frequent profile sequences – Buyers

Evaluation of Temporal Aspects of Web Page Visits

Peter I. Hofgesang
hpi@few.vu.nl

*Vrije Universiteit Amsterdam
Department of Computer Science, De Boelelaan 1081A, 1081HV Amsterdam*

Abstract

What is the intention of an online visitor? This is one of the most valuable pieces of information for an online service provider. In the case of a real-world shop, customers have the ability to explicitly express what they are looking for. However, on the web, the intention is hidden and can only be partially revealed from implicit indicators in the traces users leave behind while they browse through a website.

The vast majority of researchers in web usage mining (WUM) exploit only two types of information: the order of visited pages and their popularity - i.e. the number of times they were visited. However, several studies in information retrieval and human-computer interaction have suggested a third factor, the time spent on web pages (TSP), as an important measure of intention and relevance. The key contributions of this paper are: (1) an extensive survey of possible factors that influence the TSP measure, and (2) a similarity measure that applies to TSP and can be used to cluster users based on their assumed intentions. Our experiments are based on log files generated by several commercial websites.

1 Introduction

In the real world, the customer has the ability to express himself naturally; he can use his (native) language for assistance and he can describe his purpose. In contrast to the real-world situation, when a user – a potential client – enters the website of an online service provider he is usually left to explore on his own. His explicit goal and intention are hidden (from the website owner) and they only manifest partially in the form of implicit interest indicators trapped in so-called server- or client-side (web access) log files. Therefore, a conventional interaction that would mediate the supply and demand between two parties is not possible. The interest indicators in the log files include the objects – most often web pages – visited by the user, and the order and time stamp of these visits. However, the vast majority of WUM researchers usually apply only one or two types of information: the list of web pages visited by the users and the order in which these pages are visited. These are considered to be the most important types of information for characterising user behaviour. In addition, their importance also results from being able to be recorded accurately and automatically on the server-side web access log.

The web access log data contain yet another measure – the time spent on pages (TSP) – that is a well-recognised relevance and interest indicator in other fields such as information retrieval (IR), human-computer interaction (HCI) and E-Learning (see section 2 for related research). It could easily be assumed that TSP would also be a clear and natural indicator of importance of a page in WUM: the more time users spend on a web page, the more important the page is assumed to be for them. However, only very few articles in WUM present models applying the TSP measure. Why is TSP not interesting to researchers? Why do researchers believe that the frequency measure is a much more relevant indicator of user interest?

In our paper we present the related literature concerning the TSP over different fields (section 2). Our contributions are as follow: we give an extensive overview of the factors influencing the TSP measure (section 3 and 4). Specifically, in section 5 we present the statistical properties of TSP using several real-world data sets. Section 6 introduces a similarity measure that applies the TSP and can be used to cluster users based on their assumed intentions. Lastly, we summarise our conclusions in section 7.

2 Related Research

Much work has been done on the implicit measures of user preference in the field of IR, HCI and E-Learning. Kelly and Teevan[10] give an overview of related work on implicit feedback techniques in IR.

One of the earliest evaluations of the time aspects was presented by Morita and Shinoda[12]. Their experiments showed a positive correlation between the user interest and the reading time of articles. Konstan et al.[11] applied collaborative filtering on Usenet news data to facilitate the selection of relevant information. Their study showed that users' explicit ratings were strongly correlated with the implicit TSP measure. Ding et al.[5] presented a usage-based ranking algorithm for web IR systems that applies the TSP against standard selection-frequency-based ranking. The study of Kellar et al.[9] focused on the relation between web search tasks and the time spent on reading results. Their results support the correlation and show that it is even stronger as the complexity of a given task increases.

Despite the proven relevance of the time spent on (web) pages to user interest, only a relatively small number of research apply this measure in WUM. Some of this research uses TSP to classify web pages into link and content pages. The idea behind this is that link pages are used only as traversal paths in order to reach the content pages that are of the interest to the user. In their paper, Cooley et al.[4] distinguished auxiliary and content pages based on the reference length of the pages – the amount of time a user spends viewing a page – for finding association rules. Xie and Phoha[18] used the same definition to get rid of auxiliary pages and to run their clustering algorithm only on the content pages.

In addition, some researchers applied the TSP measure in order to cluster web users. Heer and Chi[8] used multi-modal clustering and reported that weighting navigation paths by view time improves accuracy. Shahabi et al.[14] defined a similarity angle through the feature space based on the total time spent on all common sub-sequences of each session pairs. Banerjee and Ghosh[1] also proposed a similarity measure similar to [14], but with the main difference of introducing an importance factor to weight the difference of time pairs within the common sub-sequence. Xiao et al.[17] defined four similarity based measures, among them a viewing-time- and a matrix-based method to cluster web users based on their common interest. Gunduz and Ozsu[7] combined the order of visited pages and visit time into a similarity metric for page prediction.

3 Influential factors: Frequency vs. TSP

Users' click information forms the basis of WUM. This information implicitly comprehends the order and the occurrence of visited pages. However, the data do also include another measure: the page view time. In theory, both measures are good indicators of user interest: the more frequently a page was visited and/or the more time was spent on it, the more interesting the page is supposed to be to the user. However, on the one hand, the frequency measure is a widely accepted indicator of user interest and, on the other hand, TSP seems to be excluded from WUM research. In this section, we give an overview of the possible influential factors and compare their effects on the frequency and TSP measures.

3.1 Website Hierarchy

The hierarchy of a website has a strong influence on the frequency measure [19]. Pages at the top of the hierarchy get traversed more often as intermediate nodes to reach the desired pages. For example, the root of a website (the home page) often gets the most hits. The position of a web page within the hierarchy has no direct effect on TSP.

3.2 Data preprocessing: filtering out robot transactions and session identification

Robots are automated programs that systematically fetch information from websites. Due to malicious robot transactions, which do not identify themselves, the complete removal of such sessions can be rather difficult if not impossible. A single robot session can result in hundreds or thousands

of "artificial" clicks. As a result of these systematic visits, the effect in the case of TSP is a more or less equal page view time (usually very short) throughout such sessions. Setting up thresholds to eliminate robot transactions is not straightforward.

Session identification is another non-trivial step in web data preparation. Because there are several possible session identification methods, the page frequencies within sessions may differ accordingly. The most common method of session identification is called the time frame identification [4]. This method sets a threshold for maximal page view time to form sessions. Therefore, TSP has an upper limit in this threshold (in practice the threshold is around 30 minutes). Other methods, e.g. session identification by cookie information, may lead to extreme values of several days or weeks of page view time.

3.3 Distraction

The effect of distraction is quite obvious and one of the most important issues for TSP. Chatting, answering a telephone call, having a coffee break, etc. all result in a longer TSP although the user is not actively looking at the page. Distraction has no clear influence on the frequency measure; however, interruption of a user may result in revisits and, in the case of larger websites, may lead to the user getting lost in the structure.

3.4 Page type

In addition, the type (information page, contact form, etc.), quality (density, layout, complexity, etc.) and other parameters (length, etc.) of a web page may also influence the TSP measure. Nowadays, websites generate thousands of pages dynamically. Identification of granularity strongly depends on the application and on human labelling criteria.

Furthermore, two other issues may also influence TSP. The speed of reading differs between individuals resulting in unequal TSP measurement. Network traffic (bandwidth) and server load may also alter page view times considerably.

4 TSP Data Preparation

In the previous section, we gave an overview of the most important factors that influence the frequency and TSP measures. Here, we review the essential aspects to consider when preparing web data and, in particular, TSP information. Let us first present a notation for observed session data. We augment session information with the time stamp and TSP.

Notation 1 Let us denote our observed data set as $D = \{s_1, \dots, s_i, \dots, s_N\}$, where s_i is the i^{th} session. Each s_i consists of an ordered sequence of one or more triplets of page identifiers (of the visited page), time stamps of visits and page view times: $s_i = (\langle p_{i1}, T_{i1}, t_{i1} \rangle, \dots, \langle p_{ij}, T_{ij}, t_{ij} \rangle, \dots, \langle p_{ni}, T_{ni}, t_{ni} \rangle)$, where each $p_{ij} \in P = \{p_1, \dots, p_m, \dots, p_M\}$ (the set of all page identifiers), T_{ij} is the actual time stamp of the transaction, and t_{ij} is the time spent on page p_{ij} .

A natural way to calculate TSP (t_{ij}) for a given page (p_{ij}) is to subtract the time stamp of the page from the time stamp of the following page:

$$t_{ij} = T_{ij+1} - T_{ij} \quad (j < ni, ni > 1).$$

An ideal page view time is the pure time spent on actively reading or interacting (scrolling, filling in a form, etc.) with the given page. The above calculation simplifies this ideal measure at several points, as described in the previous section. To calculate the ideal TSP, we would have to consider the time spent on network traffic, server-side page generation time and distraction, which is the most difficult factor and impossible even to approximate based on pure server-side data. We can summarise the (server-side) calculation of the ideal TSP using the following formula: $t_{ideal} = T_{ij+1} - T_{ij} - T_{networkTraffic} - T_{serverPageGeneration} - T_{distraction}$

The first issues to mention are network traffic and server load or server page generation time ($T_{networkTraffic}$ and $T_{serverPageGeneration}$). Both server- and client-side solutions are possible to measure the overhead. On the server side, the time required to generate the requested page can be measured exactly and an estimation can be calculated for the network relay based on request-response times. Client-side measures (e.g., [14]) based on specialised browsers or scripts can measure the page view time directly. A client side solution was proposed by Shahabi et al. [14]. However in their survey Srivastava et al. [15] claim that client and server side measure of TSP is equal or that client side measures are even worse because of the overhead.

As we described in previous sections, coffee breaks, parallel browsing activity, etc. all distract the users' attention. It is impossible to identify such activities on the server side. Client-side measures (such as mouse movement, page scrolling, lost-focus attribute of the current browser, eye-tracking, etc.) can be used to approximate distraction. In practice, however, researchers tend to depend on heuristics gained from observing the real data. It is common to set a threshold for maximal (reasonable) page view time and replace the extreme values by some standard view time calculated from the observed data. There is no golden rule for this threshold in WUM. It depends on the domain (e.g. a news portal has long articles, while a retail shop has images and short descriptions) as well as the users' reading capacity. Standard statistical outlier detection algorithms do not work because they would identify most of the important page view times, where users spent relatively more time, as outliers.

In the literature, several different criteria and thresholds are used to eliminate TSP outliers. Claypool et al. [3] defined the minimal (1 second) and the maximal (20 minutes) page view time and removed the outliers (general web browsing). Rafter and Smyth [13], in a job recruitment web service environment, defined a normal reading time using the median of median reading time values per individual job access for both users and jobs. The extreme values were then identified as the values larger than twice the normal reading time and were replaced by this value. In their experiment these definitions resulted in 48 seconds of normal view time and about 1.5 minutes for outlier threshold. Farzan and Brusilovsky [6] use TSP to weight page visitation frequency. They drew the minimum TSP threshold at 5 seconds and the maximum at 10 minutes and, in the case of an outlier, they left the frequency count intact.

Note that web access log data do not contain enough information to calculate TSP for the last visited pages (t_{ni}). This is due to the stateless status of the HTTP protocol. However, TSP of the last page can be rather informative. Did the user find the information he wanted or did he just leave the browser idle? Again, in the case of special applications, client-side measurement may provide a solution. In the case of internal websites (requiring login), we can also obtain this information, and extending the web pages by special scripts (which requires some modification of the pages) would also provide a solution for publicly available sites.

Nowadays, most websites provide dynamically generated pages, thus the concept of a web page is no longer well defined. In the case of a retail shop, there could be tens of thousands of articles and other parameters that can define a single web page. We can set the granularity on the article level (e.g. a specific pair of shoes) or on the level of article categories (e.g. shoes), etc. The selection of "interest" entities indirectly influences TSP values.

Furthermore, TSP values can be normalised by some page parameters (e.g. length, density, readability). In the paper by White et al. [16], for instance, reading time was normalised by document length. In contrast to this, Morita and Shinoda [12] concluded that message length or reading difficulty level are not correlated with reading time. Additionally, the heterogeneity of most websites (text, images of different resolutions, etc.) – unlike the homogenous collections of news digests, "uniform" documents, etc. – aggravate the normalisation process.

5 Properties of the TSP

Here we give an overview of the most important characteristics of TSP using real-world data sets. The data include two retail shop data sets and clickstream data of a bank's online service. During data preprocessing we identified and removed robot transactions and needless or noisy traffic. We identified unique users based on their cookie information or, when absent, their IP address, and

collected their sessions using the time frame identification method [4] with a maximal gap of 30 minutes. Note that further preprocessing of TSP, as described in Section 4, is performed for the evaluation of clustering in the following section.

Figure 1(a) shows the TSP distributions of the three data sets. Surprisingly, the Bank and Retail 1 data have similar characteristics. In general, all three distributions have the common characteristics of a significant peak of around 5-10 seconds, and a very long and tapering tail.

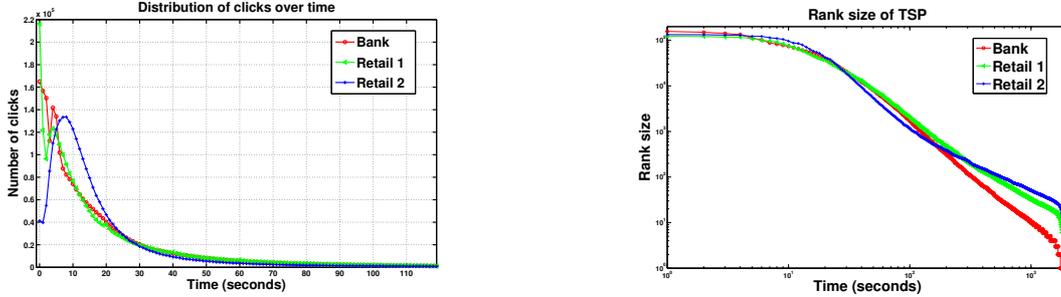


Figure 1: (a) The TSP distributions of the three datasets (first 2 minutes, rounded TSP values) (b) Rank size of the TSP distributions over the three datasets (complete 30 minutes time range) - logarithmic scales on both axes

Yan et al. [20] observed that the distribution of time spent on pages is roughly Zipfian. We ranked our data sets in descending order by frequency values and plotted them on a graph (Figure 1(b)) using logarithmic scales on both axes. Zipf curves follow a straight line when plotted on a log-log scale. Our graphical analysis shows that none of the three distributions follow the shape of the classic Zipf curve. Both the largest and smaller sizes appear to differ from the simple power laws.

6 Cluster users based on their assumed intentions

In this section we present the clustering of sessions based on a similarity metric together with our experimental results. The goal was to group users based on assumptions of their intentions. It is a rather ambitious goal given that web access log data not only lack explicit descriptive measures, but also are an incomplete record of available implicit indicators.

As stated previously, TSP is considered by prior research in IR and HCI to be a good indicator of user interest. In our approach, we combined TSP and the visitation frequency of pages, since this latter measure is the most accepted indicator of user interest within the WUM community.

Let us first define the notion of the composite session that forms the base of our clustering.

Definition 1 We combine several aggregates of interest indicators to measure the relevance of a page for a user within a session. The **composite session** of the i^{th} session is

$$CS_i = (cs_{i1}, \dots, cs_{im}, \dots, cs_M), \quad \text{where} \quad cs_{im} = w_i \prod_{k=1}^K \text{measure}_k(i, m),$$

the components are normalised and combined interest values for all page types. $\text{measure}_k(i, m)$ is the k^{th} measure calculated for the m^{th} page type within the i^{th} session and w_i is a normalisation factor for the i^{th} session.

In our experiments we use two measures ($K = 2$),

- $\text{measure}_1(i, m) = \sum_{j=1}^{ni} \begin{cases} 1 & \text{if } p_{ij} = p_{im} \\ 0 & \text{otherwise} \end{cases}$ – the frequency component
- $\text{measure}_2(i, m) = \sum_{j=1}^{ni} \begin{cases} t_{ij} & \text{if } j < ni \text{ and } \\ & p_{ij} = p_{im} \\ 0 & \text{otherwise} \end{cases}$ – the time component

with $w_i = 1/\max_m(\text{measure}_1(i, m))\max_m(\text{measure}_2(i, m))$ normalisation factor.

The idea behind this measure is that it biases toward pages that occurred frequently and that more time was spent on. The assumption is that the intention of a user is better reflected by popular pages where users also spend more time.

When measuring the similarity of two sessions, we defined an ideal baseline session based on the two composite sessions in a way that the components sum up to exactly 1.

Definition 2 The *baseline session* is defined as the pairwise mean values of the two composite sessions (s_α, s_β) normalised by their sum:

$$BS_{\alpha\beta} = (bs_{\alpha\beta 1}, \dots, bs_{\alpha\beta m}, \dots, bs_{\alpha\beta M}), \quad \text{where} \quad bs_{\alpha\beta m} = \frac{cs_{\alpha m} + cs_{\beta m}}{\sum_{m=1}^M cs_{\alpha m} + \sum_{m=1}^M cs_{\beta m}}$$

In addition, we introduced a penalty function that degrades the components of the ideal baseline session.

Definition 3 Our *similarity measure* is defined by the sum of pairwise differences between baseline session components and the minimum of the baseline components and the standard deviation of the pairwise composite session components:

$$SM(\alpha, \beta) = \sum_{m=1}^M \widehat{bs}_{\alpha\beta m}, \quad \text{where} \quad \widehat{bs}_{\alpha\beta m} = bs_{\alpha\beta m} - \min(bs_{\alpha\beta m}, \text{std}(cs_{\alpha m}, cs_{\beta m})),$$

where $\text{std}(cs_{\alpha m}, cs_{\beta m})$ is the standard deviation of values $cs_{\alpha m}, cs_{\beta m}$.

Note that this measure is defined to sum up to 1 in cases where the original sessions were identical, and 0 if they were completely different. In other cases, a value between 0 and 1 reflects the similarity of the two sessions.

In addition, note that, in practice, most sessions include only a few visits to distinct pages; therefore, the composite sessions are very sparse and calculations can be simplified accordingly.

6.1 Clustering experiment

In our experiments we analysed web usage data of a retail company’s website (Retail 2). We performed the data preprocessing steps described in section 5. In addition, a server-side approximation was available for network latency and server page generation time, which we used to normalise the time spent on page values. After extensive exploration of the data we chose a fixed, 5-minute threshold to detect page view outliers. We replaced these values by the mean page view time (without the outliers) within the sessions.

Our retail shop data set contained 1 month of traffic, which we transformed into 2.7 million sessions. In addition, we removed sessions that contained fewer than 3 clicks, assuming that most of them were accidental visits. From the remaining set, we randomly selected 5,000 sessions. We calculated the pairwise similarity matrix (only the upper triangle) based on our similarity measure, transformed the values into 2D space using multidimensional scaling and, finally, we used k-means clustering with different k values (1-10).

We analysed the k-clusters and evaluated them based on analysis of the sessions belonging to each cluster. We found that sessions within their clusters are similar and that clusters reflect our assumptions of initial intentions. Figure 2(a) shows 5 clusters based on average histograms of the composite sessions.

We compared our results to the results of a frequency-based clustering model (our implementation is based on [2]) (Figure 2)(b). In contrast to the frequency-based model, our results show two "focused" groups with interests only in single categories (women’s fashion and sales of women’s fashion). In the results using the other model, these categories were also dominant within their clusters but they were also mixed together with other categories – which probably had short page view times. Our model clustered men’s fashion together with other categories, such as household appliances and televisions, while the frequency-based model formed a focused, separate cluster for this category.

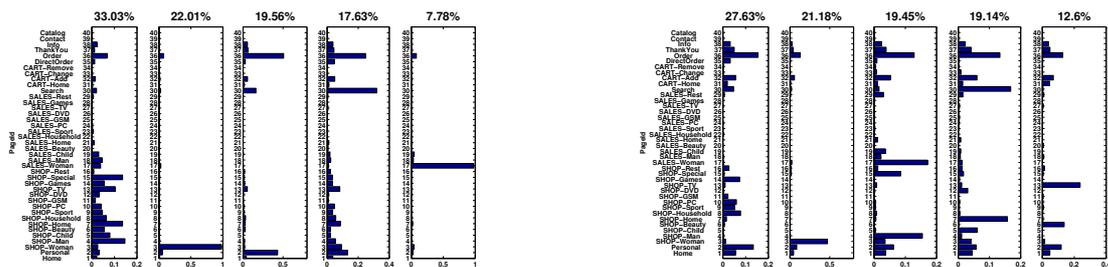


Figure 2: (a) 5 clusters from composite sessions based on the similarity measure (b) 5 frequency clusters based on [2] for comparison

7 Conclusion

In this paper we investigated possible reasons for the relative disregard of a presumably relevant interest indicator in WUM, the time spent on pages. We gave an extensive overview of the literature concerning TSP in different fields. We outlined the most likely influential factors of the TSP measure in comparison with the more widely applied frequency measure. We described the problems and gave a methodology for TSP preprocessing. We gave an overview of the statistical properties of page view time for several real-world data sets. Lastly, we defined a similarity measure based on the combination of frequency and TSP measures and evaluated it by clustering retail web shop data.

Both frequency and TSP measures are influenced by several factors. While frequency seems to be a more solid, "plug-and-play" measure, the strong influence of distraction and hardware performance makes TSP more vulnerable. However, we believe that after careful preprocessing of web data, the TSP measure is of great value in identifying user intention. Special attention should be paid to robot filtering and session and page identification. The effects of network and server overhead should be normalised. TSP values that possibly comprise distraction factors should be identified and replaced by normalised values. Thresholds should be chosen with special attention to the application domain. For many applications, the combination of TSP and frequency measures can be the optimal choice.

References

- [1] A. Banerjee and J. Ghosh. Clickstream clustering using weighted longest common subsequences. In *Proceedings of the Web Mining Workshop at the 1st SIAM Conference on Data Mining*, 2001.
- [2] I. V. Cadez, P. Smyth, E. Ip, and H. Mannila. Predictive profiles for transaction data using finite mixture models. In *Technical Report UCI-ICS*, pages 01–67, 2001.
- [3] Mark Claypool, Phong Le, Makoto Wased, and David Brown. Implicit interest indicators. In *IUI '01: Proceedings of the 6th international conference on Intelligent user interfaces*, pages 33–40, New York, NY, USA, 2001. ACM Press.
- [4] Robert Cooley, Bamshad Mobasher, and Jaideep Srivastava. Data preparation for mining world wide web browsing patterns. *Knowledge and Information Systems*, 1(1):5–32, 1999.
- [5] Chen Ding, Chi-Hung Chi, and Tiejian Luo. An improved usage-based ranking. In *WAIM '02: Proceedings of the Third International Conference on Advances in Web-Age Information Management*, pages 346–353, London, UK, 2002. Springer-Verlag.
- [6] Rosta Farzan and Peter Brusilovsky. Social navigation support in e-learning: What are real footprints? In *Proceedings of IJCAI'05 Workshop on Intelligent Techniques for Web Personalization*, pages 49–56, 2005.

- [7] S. Gunduz and M. Ozsü. A web page prediction model based on click-stream tree representation of user behavior. In *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 535–540, 2003.
- [8] Jeffrey Heer and Ed H. Chi. Separating the swarm: categorization methods for user sessions on the web. In *CHI '02: Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 243–250, New York, NY, USA, 2002. ACM Press.
- [9] Melanie Kellar, Carolyn Watters, Jack Duffy, and Michael Shepherd. Effect of task on time spent reading as an implicit measure of interest. *ASIST 2004*, pages 168–175, 2004.
- [10] Diane Kelly and Jaime Teevan. Implicit feedback for inferring user preference: a bibliography. *SIGIR Forum*, 37(2):18–28, 2003.
- [11] Joseph A. Konstan, Bradley N. Miller, David Maltz, Jonathan L. Herlocker, Lee R. Gordon, and John Riedl. GroupLens: Applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87, 1997.
- [12] Masahiro Morita and Yoichi Shinoda. Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference (SIGIR '94)*, pages 272–281, New York, NY, USA, 1994.
- [13] Rachael Rafter and Barry Smyth. Passive profiling from server logs in an online recruitment environment. In *IJCAI Workshop on Intelligent Techniques for Web Personalization (ITWP 2001)*, pages 35–41, 2001.
- [14] C. Shahabi, A. M. Zarkesh, J. Adibi, and V. Shah. Knowledge discovery from users web-page navigation. In *Proceedings of the 7th International Workshop on Research Issues in Data Engineering (RIDE '97)*, page 20, Birmingham, England, 1997.
- [15] Jaideep Srivastava, Robert Cooley, Mukund Deshpande, and Pang-Ning Tan. Web usage mining: discovery and applications of usage patterns from web data. *SIGKDD Explor. Newsl.*, 1(2):12–23, 2000.
- [16] Ryen W. White, Ian Ruthven, and Joemon M. Jose. Finding relevant documents using top ranking sentences: an evaluation of two alternative schemes. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 57–64, New York, NY, USA, 2002. ACM Press.
- [17] Jitian Xiao, Yanchun Zhang, Xiaohua Jia, and Tianzhu Li. Measuring similarity of interests for clustering web-users. In *ADC '01: Proceedings of the 12th Australasian conference on Database technologies*, pages 107–114, Washington, DC, USA, 2001.
- [18] Yunjuan Xie and Vir V. Phoha. Web user clustering from access log using belief function. In *K-CAP '01: Proceedings of the 1st international conference on Knowledge capture*, pages 202–208, New York, NY, USA, 2001. ACM Press.
- [19] Dongshan Xing and Junyi Shen. Efficient data mining for web navigation patterns. *Information & Software Technology*, 46(1):55–63, 2004.
- [20] Tak Woon Yan, Matthew Jacobsen, Hector Garcia-Molina, and Umeshwar Dayal. From user access patterns to dynamic hypertext linking. In *Proceedings of the fifth international World Wide Web conference on Computer networks and ISDN systems*, pages 1007–1014, Amsterdam, The Netherlands, 1996.

CHECKING GUIDELINE CONFORMANCE OF MEDICAL PROTOCOLS USING MODULAR MODEL CHECKING

Arjen Hommersom Perry Groot Peter Lucas

*Institute for Computing and Information Sciences, Radboud University Nijmegen
P.O.Box 9010 6500GL Nijmegen, The Netherlands
{arjenh, perry, peterl}@cs.ru.nl*

Abstract

Medical guidelines and protocols are documents aimed at improving the quality of medical care by offering support in medical decision making in the form of management recommendations based on scientific evidence. Whereas medical guidelines are intended for nation-wide use, and thus omit medical management details that may differ among hospitals, medical protocols are aimed at local use within hospitals and, therefore, include detailed information. Although protocols are often constructed on the basis of medical guidelines, the question is to which extent a protocol conforms to the guideline. Formal methods are applied to shed light on this issue. A Dutch medical guideline regarding the treatment of breast cancer, and a Dutch protocol based on it, are taken as an example.

1 Introduction

Medical management is increasingly based on recommendations from the medical scientific community, summarised in medical guidelines. A medical guideline is an extensive document, developed by a working group of professionals involved in the management of the disorder covered by the guideline. It has been shown that medical guidelines can improve healthcare outcomes [15] and may even reduce the cost of care up to 25% [5]. In practice, however, local hospitals often work with local adaptations of medical guidelines called protocols. The need for a protocol in conjunction with a guideline is twofold: firstly, a guideline is an extensive document (e.g., the Dutch breast-cancer guideline is 121 pages in A4 format), and, therefore, it is not easy to locate relevant information; secondly, details about duration, dose, or actual procedure have been omitted from the guideline, and, thus, are added in a protocol to complement the information that is in the guideline. Hence, basically, a medical protocol is a summary of the most important sections that are in the guideline, mostly recommendations, supplemented with hospital-specific details concerning the treatment. This implies that many sections in a protocol may be very similar to related sections in a guideline. However, there may also be differences, partly due to differences in opinion between the guideline designers and protocol designers, and partly due to the difference in purpose of a guideline and protocol.

Although much research has already focused on designing computer-oriented languages and developing tools for their employment, not many researchers seem to note that guidelines and protocols are different levels of abstraction of medical management. This view raises a number of issues currently not addressed in literature. Firstly, guidelines are typically under-constrained thereby omitting many details about treatment order. This contrasts for example with [13] which considers a guideline to be some form of program, but in which no execution paths are excluded that are illogical for medical management in practice. Secondly, guidelines are usually considered to be solitary objects and their verification still takes a lot of effort. Clearly, reuse of verification efforts would be useful for the verification of protocols which are adapted from a guideline. No reference has been made to verifying adaptations of guidelines.

Sofar, protocol conformance to guidelines has only been looked at from an informal angle [9]. In this paper we address the problem of protocol conformance to guidelines using formal methods. This is done by interpreting guidelines as defining (logical) constraints on the medical management of patients performed in practice, whereas protocols are interpreted as more or less executable models. This approach was inspired by a statement by Wiersma and Burgers that “recommendations in guidelines should not only be based on

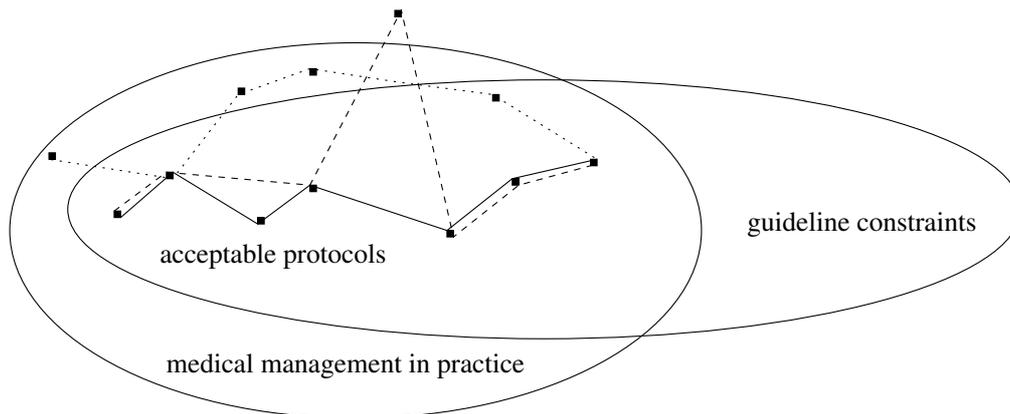


Figure 1: Sketch of medical management paths occurring in a protocol.

evidence extracted from scientific literature, but take into account the context of daily medical practice as well” [14]. In principle this approach would allow one to discover flaws or suboptimal management actions in the medical management in practice, assuming that a given protocol and guideline are correct, or to find incorrect or suboptimal medical management decisions in a protocol or guideline, assuming that the medical management in practice is correct and optimal. In this paper, we assume that the guideline augmented with information from general medical practice is correct and use model checking techniques to find flaws in the protocol.

2 Approach

In this section, we first introduce a method for checking the conformance of protocols to guidelines. Subsequently, preliminaries to formal methods are presented.

2.1 Verification of Protocol Conformance to Guidelines

The premise in this paper is that clinical practice guidelines and protocols provide *necessary*, but not *sufficient* conditions for making medical decisions. In other words, the guideline requires interpretation of medical doctors to apply the advises in practice. Thus, treatments which are acceptable given a strict interpretation of the protocol, might not be acceptable in practice. For example, the Dutch breast cancer guideline does not exclude the possibility of informing the patient *after* treatment about the different treatment possibilities. Clearly, such incompleteness of the guideline or protocol will not cause problems in practice as the advises are not mechanically followed by the physician. Hence, differences related to such a (lack of) advise can be considered irrelevant in practice. This is abstractly represented in Figure 1, which shows several paths of a protocol compared with all paths allowed by guideline and medical management in practice. In this case, violations occurring in the dotted line are most interesting, as the violation in the dashed line will not occur in practice.

Thus, the question is whether or not a protocol, restricted to medical management in practice, conforms to the constraints that the guideline imposes. The problem of this approach is that this medical management in practice cannot easily be assessed, because it is based on general and abstract medical knowledge, which is difficult to articulate and express directly, which makes it difficult to elicitate and formalise. Therefore, the approach we have taken in this paper is that we have formulated a concrete description of treatment paths that are taken in practice, extracted from a medical textbook. This does not formalise the *general* medical practice, but provides a good estimation of what is usually done in practice for a specific domain. Then, the approach consists of the following steps. Firstly, concrete treatment paths are formulated, which are known to be part of medical management in practice. Secondly, this information is weakened up to a point it is consistent with the protocol, i.e., we investigate which paths of medical practice are consistent with the protocol. Finally, the protocol restricted by medical practice is compared with the constraints imposed by the guideline.

In other words, the guideline restricted to medical practice, i.e., the intersection of Figure 1, is defined

as the gold standard. Finding a deviation from this gold standard in a protocol might indicate a problem in the protocol, although it is possible that it only indicates an error in the medical textbook or guideline. Nonetheless, we believe that such information is more valuable to guideline and protocol developers than deviations which are not even part of common sense medical practice.

2.2 Formal Preliminaries

It has been shown in [8] that the step-wise, possibly iterative, execution of a guideline can be described precisely by means of temporal logic. The logic that we use here for specifying properties of medical guidelines is a combination of Computation Tree Logic (CTL) [3, 4] and Linear Temporal Logic (LTL) [10]. CTL uses atomic propositions and Boolean connectives (e.g., \neg , \vee , \wedge) to build up more complicated expressions for describing properties of states. Furthermore, CTL formulas can be composed of *path quantifiers* and *temporal operators* for describing properties of *computation trees*, i.e., all paths that are possible from a certain state. The path quantifiers are **A** and **E** to specify that all of the paths or some of the paths starting at a specific state have some property. The temporal operators describe properties of a path through the tree. The four temporal operators used are **X**, **G**, **F**, and **U**. With $\mathbf{X}\varphi$ being true if φ holds in the next state, $\mathbf{G}\varphi$ if φ holds in the current state and all future states, $\mathbf{F}\varphi$ if φ holds in some state in the future (or is true in the current state), $\varphi\mathbf{U}\psi$ if φ holds until ψ holds, i.e., there is a state on the path where ψ holds and in every preceding state φ holds. LTL provides operators for describing events along a *single computation path*. Each formula is of the form $\mathbf{A}f$, with f being a path formula, which is either an atomic proposition or inductively defined as $\neg f$, $f \vee g$, $f \wedge g$, $\mathbf{X}f$, $\mathbf{F}f$, $\mathbf{G}f$, or $f\mathbf{R}g$ with f, g path formulas.

In model checking literature, the approach of verifying a restricted part of the system (e.g., the protocol consistent with medical practice) is called *modular verification* (cf. [7]). In the assumed-guarantee paradigm, the specification of a module consists of a specification of guaranteed behaviour assuming that the system behaves in a certain way, i.e., the assumed behaviour. In this paper, the assumed behaviour is written down in LTL and the guaranteed behaviour in CTL. The assume-guarantee assertions are written down as $[\varphi]M\langle\psi\rangle$, meaning that the CTL formula ψ holds in the computation tree that consists of all computations of the program, described by M , that satisfy the LTL formula φ . The tool used for performing the modular verification of medical guidelines is Cadence SMV.¹

3 Medical Management in Breast Cancer

First, we give an informal description on the medical management as stated in the CBO guideline, the IKO protocol, and the textbook of Roses [11] that deals with locoregional treatment of operable breast cancer, i.e., T1-2 N0-1 M0 breast cancer according to the TNM classification system [6]. Thereafter, we give formalisations according to the approach described in the previous section.

3.1 Informal Description of Medical Management

According to the CBO guideline there are only two options for local treatment of operable invasive breast cancer: breast-conserving therapy (BCT) or modified radical mastectomy (MRM). BCT implies ample local excision of the tumour, an axillary staging procedure, and radiotherapy of the breast. MRM involves a total resection of the breast (mastectomy) and dissection of the axillary nodes (AND). The aim of BCT is to achieve a survival rate comparable to that following MRM with an optimal cosmetic result in terms of the treated breast. BCT is usually the preferred treatment unless the patient has a clear preference for MRM or there are contra indications for BCT, i.e., there is either (1) multicentricity (two or more tumour foci in different quadrants), (2) diffuse malignant microcalcifications, or (3) previous radiotherapy of the breast. Whereas these three contra indications are obtained *before* surgery, one other contra indication for BCT is obtained *during* surgery, i.e., (4) the margins of the local excision remain tumour-positive after repeated local excision attempts. In this case, local excision attempts are unsuccessful in removing the primary tumour and treatment therefore switches to MRM.

Treatment of the axillary nodes is also part of the treatment of breast cancer as the pathologic assessment of axillary lymph nodes remains the most important prognostic variable for the invasive breast cancer patient. An optimal assessment would be achievable by means of a complete axillary node dissection. However,

¹<http://www.cis.ksu.edu/santos/smv-doc/>

AND may lead to morbidity, e.g., pain, limited shoulder movement. An alternative for axillary staging is the sentinel node procedure (SNP), which only dissects the sentinel nodes, i.e., those nodes that drain the area of the breast where the primary tumour is located and thus are most likely to contain metastasis. The SNP is currently the standard procedure for axillary staging in breast cancer provided that the contra indications do not hold, where contra indications of SNP are defined as (1) suspected or proven malignancy in the axillary nodes, (2) tumour > T2, (3) multiple tumour foci, or (4) potentially disrupted lymph drainage due to recent axillary surgery or a large biopsy cavity following tumour excision. When the SNP is not possible, complete axillary node dissection should be carried out. Furthermore, treatment of the axilla is indicated (i.e., dissection, radiotherapy) for all forms of lymph node metastasis.²

Whereas the CBO guideline and IKO protocol lack many details about treatment order, [11] provides a more detailed description. In addition, according to [11], the sentinel node procedure (SNP) is started before segmental excision (i.e., used in BCT) or mastectomy. The sentinel nodes (SNs) are then immediately sent to the pathology lab, where they are examined during surgery. If the SNs are found to be positive, axillary dissection can be completed during the primary breast surgery in one setting. Furthermore, [11] differs with the CBO guideline and IKO protocol in the case of recurrent tumour positive resection margins in the BCT treatment. Whereas CBO and IKO recommend to switch the treatment to MRM, which includes axillary dissection, [11] only recommends a mastectomy with axillary dissection dependent on sentinel node histopathology.

3.2 Formalisation of Medical Management

Here, we introduce the constraint-based representation of the guideline, an executable model of the protocol, and the model of the medical management performed in practice.

3.2.1 Constraint-Based Representation of the CBO Guideline

The language we use for atomic propositions consists of medical actions *Actions*, medical plans *Plans*, and data structures *Data*:

$$\begin{aligned} \text{Actions} &: \{\text{tumour-excision, mastectomy, AND, SNP}\} \\ \text{Plans} &: \{\text{TREATMENT, BCT, MRM, AXILLA-STAGING}\} \\ \text{Data} &: \{\text{CI-BCT, CI-SNP, TF, SN, ITC}\} \end{aligned}$$

with $\text{CI-BCT, CI-SNP} \in \{\top, \perp\}$ denoting the contra indications for BCT and SNP respectively, $\text{SN} \in \{\text{unknown, neg, pos}\}$ denotes whether there is a metastasis found in the lymph nodes after performing the SN procedure, $\text{TF} \in \{\text{unknown, } \top, \perp\}$ denotes whether the re-section margins are tumour free, and $\text{ITC} \in \{\text{unknown, } \top, \perp\}$ denotes whether the tumour cells are isolated when the sentinel node is found positive. In formulas, we write the variable name if it is meant that the variable is equal to \top , and the negated variable name is used to denote the respective variable is equal to \perp . The final representation in temporal logic of the medical management in the CBO guideline is shown in Figure 2.

Some constraints given by the guideline are not easily expressible in temporal logic as other modalities than treatment order are involved. For example, the preference for BCT over MRM and the preference for the SNP over axillary dissection for staging the axilla. Furthermore, certain assumptions regarding the patient data are implicit in the guideline. For example, the status of the resection margins (tumour free (TF) or not (\neg TF)) becomes known after tumour excision and the existence of metastasis ($\text{SN}=\text{pos}$ or $\text{SN}=\text{neg}$) becomes known after the SNP. Here we have chosen not to consider these more implicit constraints.

3.2.2 Asbru Representation of the IKO Protocol

To transform the IKO protocol into a more or less executable model, which can be verified with respect to the constraints set by the CBO guideline, we have chosen to use the guideline representation language Asbru [12] as intermediate representation. We use Asbru, because its semantics has been defined precisely in previous research [1] and can be translated automatically into SMV for model checking purposes [2].

The Asbru model constructed (Figure 3) consists of nine plans ordered in a hierarchy. Arrows indicate sequentially executed sub-plans, dashed lines unordered executed sub-plans. The latter is in particular used when the protocol lacks any information about treatment order, e.g., MRM with sub-plans AND and

²The CBO guideline differs at this point with the IKO protocol as it makes an exception for isolated tumour cells.

Constraints related to control structure
(1) $\mathbf{AG}(\text{TREATMENT} \rightarrow \mathbf{AF}(\text{BCT} \vee \text{MRM}))$
(2) $\mathbf{AG}(\text{CI-BCT} \rightarrow \neg \text{BCT})$
(3) $\mathbf{AG}(\text{BCT} \rightarrow \mathbf{AF}(\text{AXILLA-STAGING} \vee \text{MRM}) \wedge \mathbf{AF} \text{ tumour-excision})$
(4) $\mathbf{AG}(\text{MRM} \rightarrow \mathbf{AF} \text{ AND} \wedge \mathbf{AF} \text{ mastectomy})$
(5) $\mathbf{AG}(\text{AXILLA-STAGING} \rightarrow \mathbf{AF}(\text{AND} \vee \text{SNP}))$
(6) $\mathbf{AG}(\text{CI-SNP} \rightarrow \neg \text{SNP})$
(7) $\mathbf{AG}(\text{tumour-excision} \rightarrow ((\neg \text{TF} \rightarrow \mathbf{AF} \text{ MRM}) \wedge (\text{TF} \rightarrow \mathbf{AG} \neg \text{MRM})))$
(8) $\mathbf{AG}(\text{SNP} \rightarrow (\text{SN} = \text{pos} \wedge \neg \text{ITC} \rightarrow \mathbf{AF} \text{ AND}))$
(9) $[[\mathbf{G} \neg \text{MRM}]]M \langle \mathbf{AG}(\text{SNP} \rightarrow \mathbf{AG}(\text{ITC} \rightarrow \mathbf{AG} \neg \text{AND})) \rangle$
(10) $\mathbf{AG}(\text{TREATMENT} \rightarrow (\text{CI-SNP} \rightarrow \mathbf{AF} \text{ AND}))$
Constraints related to data
(11) $(\text{CI-BCT} \rightarrow \mathbf{AG} \text{ CI-BCT}) \wedge (\neg \text{CI-BCT} \rightarrow \mathbf{AG} \neg \text{CI-BCT})$
(12) $(\text{CI-SNP} \rightarrow \mathbf{AG} \text{ CI-SNP}) \wedge (\neg \text{CI-SNP} \rightarrow \mathbf{AG} \neg \text{CI-SNP})$

Figure 2: Constraint-based representation of the CBO guideline. BCT = breast conserving treatment, MRM = modified radical mastectomy, CI-BCT = contra indications for BCT, SN = result of sentinel node procedure, CI-SNP = contra indications for SNP, TF = tumour free resection margins. Constraint (9) is restricted to a part of the model, denoted by the assumption $\mathbf{G} \neg \text{MRM}$. All other constraints ψ are short for $[\top]M \langle \psi \rangle$.

mastectomy. The top level plan `treatment` first executes BCT unless there are contra indications (filter condition). If BCT successfully completes, so will `treatment` (wait-for condition), else MRM will be executed. The execution of BCT surgery and MRM by BCT is analogous. BCT surgery may execute its sub-plans in any order. To allow for a particular order we use a manual activation which we assume to occur eventually. The SNP and tumour excision may abort (abort condition) in case of positive SNs or not tumour free re-section margins respectively, which is then propagated up the hierarchy, resulting in BCT executing MRM. (cf. [1] for details about the Asbru semantics.)

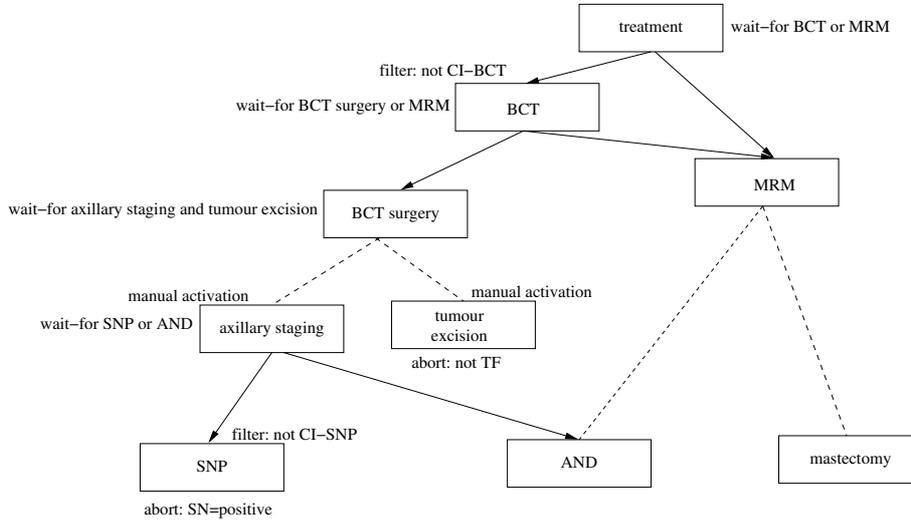


Figure 3: Asbru interpretation of the IKO protocol. Arrows represent sequential plans, dashed lines represent unordered sub-plans

In the SMV model, which is automatically constructed from the Asbru model, most variables dealing with patient data are initialised as *unknown* and receive an indeterministic value in the second step to make sure there is only one root of the model. Furthermore, we assume that they do not change during the treatment. The only variables that are initialised at a later stage are the status of the sentinel node, which becomes known during the SNP and whether or not the re-section margins are tumour free, which becomes known after excising the tumour. Furthermore, fairness constraints have been added to ensure that the manual activation of both the axillary staging and the tumour excision eventually occurs. In other words,

the patient will not wait indefinitely for the treatments to start.

3.2.3 Decision Tree of Medical Management in Practice

Information from [11] can be represented in a decision tree as shown in Figure 4, which deals with the ordering of medical actions treating the primary tumour (BCT and MRM) and the axilla (SNP and/or AND).³ Nodes represent medical actions or plans, arcs represent constraints. A path from the root node to a leaf node represents a treatment path, which defines the order of medical actions when the constraints on the path are satisfied. Leaf nodes labelled with BCT or MRM mean the wrap-up of the medical strategy followed.

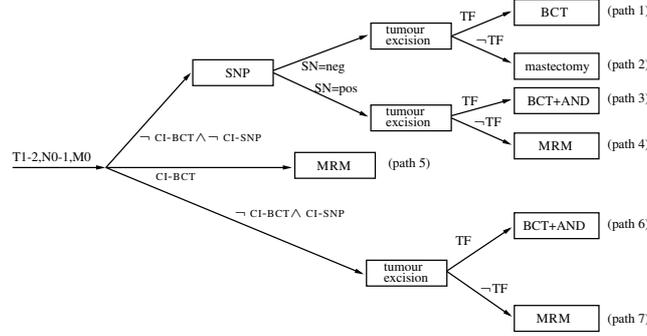


Figure 4: Background knowledge: possible treatment paths for surgery of operable invasive breast cancer. CI-BCT = contra indications BCT, CI-SNP = contra indications SNP, TF = tumour free resection margins, AND = axillary node dissection.

The formal interpretation of this decision tree is two-fold. First, we can give a CTL characterisation of these paths in terms of an existential path operator, denoted by Δ , which allows us to verify whether each of these paths exists in the protocol. Each action that occurs in the decision tree has been interpreted as a plan activation in the execution of the Asbru model, although the technical details of this have been omitted here for clarity. The first two paths are described as follows:

- $$(1) \quad \mathbf{EX}(\neg \text{CI-BCT} \wedge \neg \text{CI-SNP} \wedge \mathbf{EF}(\text{SNP} \wedge \text{SN} = \text{neg} \wedge \mathbf{EF}(\text{tumour-excision} \wedge \text{TF} \wedge \mathbf{AG}(\neg \text{mastectomy} \wedge \neg \text{AND})))) \quad (2) \quad \mathbf{EX}(\neg \text{CI-BCT} \wedge \neg \text{CI-SNP} \wedge \mathbf{EF}(\text{SNP} \wedge \text{SN} = \text{neg} \wedge \mathbf{EF}(\text{tumour-excision} \wedge \neg \text{TF} \wedge \mathbf{EF}(\text{mastectomy} \wedge \mathbf{AG}(\neg \text{AND}))))))$$

On the other hand, we may view the decision tree as a number of constraints on medical management in practice. To this end, we extract a number of assumptions from the decision tree about the normal implementation of the advises given in the protocol. It is quite obvious to guarantee that such statements are sound with respect to the decision tree. In this case study, we consider the following LTL assumptions Γ .

- (1) $(\neg \text{CI-BCT} \wedge \neg \text{CI-SNP}) \leftrightarrow \mathbf{F} \text{ SNP}$
- (2) $(\mathbf{F} \text{ SNP}) \rightarrow ((\neg \text{tumour-excision} \mathbf{U} \text{ SNP}) \wedge \mathbf{F} \text{ tumour-excision})$
- (3) $((\mathbf{F} \text{ SN} = \text{neg}) \wedge (\mathbf{F} \text{ TF})) \rightarrow (\neg(\mathbf{F} \text{ AND}) \wedge \neg(\mathbf{F} \text{ MRM}))$
- (4) $((\mathbf{F} \text{ SN} = \text{neg}) \wedge (\mathbf{F} \neg \text{TF})) \rightarrow ((\mathbf{F} \text{ mastectomy}) \wedge \neg(\mathbf{F} \text{ AND}))$
- (5) $((\mathbf{F} \text{ SN} = \text{pos}) \wedge (\mathbf{F} \text{ TF})) \rightarrow ((\mathbf{F} \text{ AND}) \wedge \neg(\mathbf{F} \text{ MRM}))$
- (6) $((\mathbf{F} \text{ SN} = \text{pos}) \wedge (\mathbf{F} \neg \text{TF})) \rightarrow \mathbf{F} \text{ MRM}$
- (7) $(\text{CI-BCT} \rightarrow (\neg(\mathbf{F} \text{ tumour-excision}) \wedge \mathbf{F} \text{ MRM}))$
- (8) $(\neg \text{CI-BCT} \wedge \text{CI-SNP}) \rightarrow \mathbf{F} \text{ tumour-excision}$
- (9) $(\neg \text{CI-BCT} \wedge \text{CI-SNP} \wedge (\mathbf{F} \text{ TF})) \rightarrow ((\mathbf{F} \text{ AND}) \wedge \neg(\mathbf{F} \text{ MRM}))$
- (10) $(\neg \text{CI-BCT} \wedge \text{CI-SNP} \wedge (\mathbf{F} \neg \text{TF})) \rightarrow \mathbf{F} \text{ MRM}$

Assumption (1) and (2) deals with the use of sentinel node procedure and the order between this and the excision of the tumour. Assumptions (3) to (6) are concerned with paths (1) to (4). Assumption (7) deals with path (5). Finally assumptions (8) and (9) deals with path (6) and (7).

³We abstract from radiotherapy and isolated tumour cells.

4 Model Checking Results

As explained in Subsection 2.1, the medical management stated by the IKO protocol is less precise than the medical management performed in practice. Typically, one would expect the medical management in the protocol to be under-constrained when compared to the medical management in practice. With the SMV model checker we were able to verify that all paths described by Δ , except (2), can occur in the IKO protocol. Path (2) does not hold in the IKO protocol because it recommends a MRM whereas [11] recommends a mastectomy, i.e., axillary dissection is included in the medical management according to the protocol, but not according to [11]. The guideline does not provide specific evidence related to this advice, which suggests that both possibilities are acceptable. Nonetheless, such differences should be discussed with medical experts to find out whether the protocol or textbook is incomplete or incorrect.

Because treatment path (2) from the medical textbook is not part of the protocol, it follows that sentence (4) of Γ can not be coherent with the model (i.e., from (4) it follows the antecedent of (4) is false), so in this form it is not usable. We could therefore either adapt the assumption so that it corresponds to the guideline or omit it. Here, we have omitted it. Let Γ' be Γ without (4), then we verify each guideline constraint $[\varphi]M\langle\psi\rangle$ by model checking $[\varphi, \Gamma']M\langle\psi\rangle$ using SMV on the Asbru model of the IKO protocol. This shows that constraint (9) does not hold in the Asbru model of the IKO protocol, indicating a difference between protocol and guideline with respect to medical management in practice. The reason for this difference can be tracked back to preliminary evidence stated in the guideline, which has a low certainty degree, i.e., a low *level of evidence*. Although, in this case the difference between protocol and guideline is clear and could also have more easily been found through an informal analysis, this is largely because the protocol and guideline have a very similar structure and their recommendations are almost identical. However, the approach taken is independent of the underlying structure of the protocol and guideline. Therefore, this case study shows that formal techniques can be used to compare guideline and protocol independent of their underlying document structure.

The resources to check that paths from the decision tree exists were minimal. The number of nodes in the binary decision diagram (BDD) was only 100.000, and terminated within 3 seconds on a modern pc. The number of BDD nodes allocated for checking that the protocol conforms with the guideline was 250.000, and time spent on the verification of all the constraints took 175 seconds. While it is to be expected that more resources will be needed for larger case-studies, we have not used advanced techniques such as bounded model checking, variable ordering, or proof decomposition to reduce the computational complexity.

5 Discussion

The aim of this work was to use formal methods for obtaining insight into the differences and similarities between guidelines and protocols, based on the assumption that protocols should be looked upon as local modifications of guidelines. In the work presented here the view was taken that medical guidelines and protocols usually only specify necessary but not sufficient constraints on medical management.

In our study we have setup a modular model checking approach for checking the conformance of a protocol to the guideline from which it has been adapted. In this approach, medical guidelines and protocols are considered to be constraints on medical management. Medical guidelines are representable as temporal logic formulas whereas protocols are interpretable as more or less executable models. Furthermore, medical management as used in practice was used as additional background knowledge for restricting the guideline and protocol thereby rejecting treatment paths which are illogical for medical management in practice. In this paper we have applied the modular model checking approach to the CBO breast cancer guideline, the IKO protocol, and used [11] for additional background knowledge. We have shown that with this approach interesting differences between guidelines and protocols with respect to background knowledge can be obtained. These differences can then be communicated to health care professionals for further clarification. In our case, one of the medical oncologists responsible for the IKO protocol confirmed that some differences found could be traced back to low levels of evidence, suggesting that such differences may be ignored.

The research presented in this paper is a novel approach in locating differences between protocols and guidelines giving a promising starting point for further investigating the relations between guidelines, protocols, and medical management in practice. Some questions still remain for further research. Firstly, a limitation of this research is that it was only based on one reference protocol on breast cancer treatment, selected at the start of this research. A second protocol for this type of breast cancer by the NKI (the Netherland Cancer Institute) is available, however, this protocol is very different from the first IKO protocol and

the guideline, resulting in other challenges than those discussed in this paper. Secondly, a question that emerged during the course of our research was whether the level of evidence as indicated by the oncologist can be incorporated in our approach as differences based on low levels of evidence can be ignored. Thirdly, formally obtaining differences between protocol and guideline may be useful for protocol designers. In this research, we have only been able to find end point protocols; as a consequence, the transformation process could only be described as consisting of a single step, which in reality may be a more iterative process.

References

- [1] M. Balsler, C. Duelli, and W. Reif. Formal semantics of Asbru - an overview. In *Proceedings of the International Conference on Integrated Design and Process Technology*, Passadena, June 2002. Society for Design and Process Science.
- [2] S. Bäumlner, M. Balsler, A. Dunets, W. Reif, and J. Schmitt. Verification of medical guidelines by model checking – a case study. In A. Valmari, editor, *Proceedings of 13th International SPIN Workshop on Model Checking of Software*, volume 3925 of LNCS, pages 219–233. Springer-Verlag, 2006.
- [3] M. Ben-Ari, Z. Manna, and A. Pnueli. The temporal logic of branching time. *Acta Inf.*, 20, 1983.
- [4] E.M. Clarke and E.A. Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. In *Logic of Programs: Workshop*, number 131 in LNCS. Springer, May 1981.
- [5] P. Clayton and G. Hripsak. Decision support in healthcare. *International Journal of Biomedical Computing*, 39:59–66, 1995.
- [6] F.L. Green, D.L. Page, I.D. Fleming, A. Fritz, M.C. Balch, D.G. Haller, and M. Morrow. *AJCC Cancer Staging Manual*. Springer-Verlag, New York, 2002.
- [7] O. Kupferman and M.Y. Vardi. Modular model checking. *Lecture Notes in Computer Science*, 1536:381–401, 1998.
- [8] M. Marcos, M. Balsler, A. Teije, and F. Harmelen. From informal knowledge to formal logic: A realistic case study in medical protocols. In *Proceedings of EKAW*, pages 49–64. Springer, 2002.
- [9] M. Marcos, B. Martínez-Salvador, A.J. Hommersom, P. Groot, P.J.F. Lucas, A. Jovell, and S. Blancafort. Case-study in transformations for protocol development from guidelines. Technical Report D5.1, Procure II, 2006.
- [10] A. Pnuelli. A temporal logic of concurrent programs. *Theoretical Computer Science*, 13:45–60, 1981.
- [11] D.F. Roses. *Breast Cancer*. Elsevier, Philadelphia, PA, 2nd edition, 2005.
- [12] Y. Shahar, S. Miksch, and P. Johnson. The Asgaard project: A task-specific framework for the application and critiquing of time-oriented clinical guidelines. *AIME*, 14:29–51, 1998.
- [13] A. ten Teije, M. Marcos, M. Balsler, J. van Croonenborg, C. Duelli, F. van Harmelen, P.J.F. Lucas, S. Miksch, W. Reif, K. Rosenbrand, and A. Seyfang. Improving medical protocols by formal methods. *Artificial Intelligence in Medicine*, 36(3):193–209, 2006.
- [14] J.J.E. van Everdingen, J.S. Burgers, W.J.J. Assendelft, J.A. Swinkels, T.A. van Barneveld, and J.L.M. van de Klundert, editors. *Evidence-based Guideline Development*, chapter Formulating Recommendations, page 171. Bohn, Houten, 2004.
- [15] S. Woolf, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw. Potential benefits, limitations, and harms of clinical guidelines. *British Medical Journal*, 318:527–530, 1999.

CONTEXT SPECIFICATION AND UPDATE MECHANISMS FOR DIALOGUE MANAGEMENT

Simon Keizer Roser Morante

University of Tilburg, P.O.Box 90153, 5000 LE Tilburg

Abstract

We present a dialogue management approach based on a rich context model and a system of dialogue acts that takes the multidimensional nature of communication into account. The focus in this paper is on the specification of the context model and the update algorithm, which makes explicit how utterances produced by either the user or the system change the context model, and more in particular, how every dialogue act contributes to changing the information state. The context model and the update mechanisms have been implemented in a dialogue manager that operates within an interactive question answering system.

1 Introduction

In this paper we present a dialogue management approach based on updating a rich context model, that contains several types of information. Part of this information is a representation of the essential processes of information transfer and grounding in terms of beliefs and goals. The context model and the update mechanisms have been implemented in a dialogue manager that operates within an interactive question answering system. Our work builds on prior research on dialogue modelling in the BDI (Beliefs, Desires, Intentions) paradigm [2, 5], and has strong similarities with the *Information State* approach to dialogue management [8].

During a dialogue, the context model is updated under the influence of both user and system utterances, depending particularly on the *dialogue acts* performed in those utterances. The dialogue acts used are organised in a multidimensional taxonomy, in which each dimension represents a different aspect of the communication process. We propose a context update model that takes the multidimensional nature of communication into account, with the aim of designing a dialogue system that can handle flexible and natural dialogues. The model involves both the context specification and an algorithm for updating the context.

The input to the context update algorithm is an abstract representation of the utterance produced in the dialogue, either by the user or by the system. When the system processes a user utterance, the various language analysis components produce that abstract representation, generally in terms of dialogue acts, except if the system had problems in identifying the dialogue acts performed in the utterance. The starting point of updating the context model with a dialogue act produced by the user consists of creating beliefs about the dialogue act *preconditions*, which connect the dialogue act to the context model. For example, the user asking a question leads to the system's beliefs that the user wants to know something and that the user believes that the system knows that something, that something being determined by the semantic content of the dialogue act. When the system produces a system utterance, a multi-agent dialogue act generator produces the abstract representation, in terms of dialogue acts. The basis for the system to generate a dialogue act depends on the information in the context model, which has to satisfy the dialogue act preconditions.

The paper is organised as follows: in Section 2, Dynamic Interpretation Theory (DIT) is introduced, which is the theoretical framework for our research; in Section 3, the PARADIME dialogue manager is described, involving dialogue act generation and specification of the context model; in Section 4, the context update model is presented; Section 5 shows an example of the interaction

between the context model and the dialogue act generator. Finally, Section 6 closes the paper with some conclusions and suggestions for future research.

2 Context modelling and dialogue acts in DIT

In Dynamic Interpretation Theory (DIT) [3], a dialogue is modelled as a sequence of utterances expressing sets of *dialogue acts* (DAs). These are semantic units, operating on the information states of the participants. Formally, a dialogue act in DIT consists of a *semantic content* (SC) and a *communicative function* (CF), the latter specifying how the information state of the addressee is to be updated with the former upon understanding the corresponding utterance. Dialogue acts are organised in a taxonomy¹ consisting of several *dimensions* [4], reflecting the idea that during a dialogue several aspects of the communication need to be attended to simultaneously by the participants, causing utterances to be *multifunctional*: Task-Oriented acts, Auto-Feedback (acts dealing with the speaker's processing of the addressee's previous utterances), Allo-Feedback (acts dealing with the addressee's processing of the speaker's previous utterances), several dimensions of Interaction Management (IM), such as turn-taking, time-management, and dialogue structuring, and Social Obligations Management (SOM). The multidimensional organisation of the taxonomy supports this multifunctionality in that it allows several dialogue acts to be performed in each utterance, at most one from each dimension. The dimension a dialogue act belongs to is determined either by a *dimension-specific* CF (DS-CF), or by a *general-purpose* CF (GP-CF) combined with a SC in a particular dimension (e.g., a YNQ about auto-feedback, as in "did you say tomorrow?").

A participant's information state in DIT is represented in his *context model*, and contains all information considered relevant for his interpretation and generation of dialogue acts. A context model is structured into several components, the most relevant of which being (i) the *Linguistic Context*: information about the utterances produced in the dialogue so far (a kind of extended 'dialogue history') and about any planned dialogue acts (a 'dialogue future'), (ii) the *Semantic Context*: current information about the task/domain, including assumptions about the dialogue partner's information, (iii) the *Cognitive Context*: the current processing states of both participants (on the levels of perception, interpretation, evaluation, and task execution), as viewed by the speaker, and (iv) the *Social Context*: communicative pressures (to perform social acts).

DIT distinguishes between four levels of understanding that the participants may reach in processing each other's utterances. In a dialogue system, the level of *Perception* is associated with successful speech recognition. *Interpretation* corresponds to being able to identify the intentions of the speaker in producing the utterance, i.e., to recognise the dialogue act(s) performed with the utterance. *Evaluation* level understanding indicates that the beliefs that result from the (pre-conditions of the) dialogue act identified at the interpretation level are consistent with the own information state. Reaching the *Execution* level means being able to do something with the result of the evaluation. For example, in the case of a question, it consists of finding the information asked for; in the case of an answer, it means taking over the content of the answer.

3 The Paradime dialogue manager

The current implementation of the PARADIME dialogue manager is integrated in an interactive question answering (QA) system, as developed in the IMIX multiproject², of which PARADIME (Parallel Agent-based Dialogue Management Engine) is a subproject. The task domain at hand concerns encyclopedic information in the medical domain. The system consists of several input analysis modules (ASR, syntactic analysis in terms of dependency trees, and shallow semantic tagging), three different QA modules that take self-contained domain questions and return answers retrieved from several electronic documents with text data in the medical domain, and a presentation module that takes the output from the dialogue manager, possibly combining any QA-answers to be presented, to produce a multimodal system utterance. The dialogue management module provides support for more interactive, coherent dialogues, in which problems can be solved both

¹See web page <http://ls0143.uvt.nl/dit/>.

²IMIX (Interactive Multimodal Information eXtraction) is funded by the Dutch national foundation for scientific research; see <http://www.nwo.nl/imix>.

about communication and question answering processes [1]. This module includes submodules for multimodal reference resolution and dialogue act recognition.

In the remainder of this section, we will describe the two main components of the dialogue manager: the dialogue act generator and the context model.

3.1 Dialogue act generation

Having several dimensions of dialogue acts that each attend to a different aspect of communication, the generation of (system) dialogue acts should also happen along those dimensions. As a dialogue act in a dimension can be selected independently of the other dimensions, the generation process is divided over a set of *Dialogue Act Agents* (DAAs) operating in parallel on the information state of the system, each agent dedicated to generating dialogue acts from one particular dimension.

All of the DAAs continuously monitor the context model and, if appropriate, try to generate candidate dialogue acts from their associated dimension. This process of monitoring and act generation is modelled through a triggering mechanism: if the information state satisfies the agent's *triggering conditions*, i.e., if there is a motivation for generating a dialogue act from a particular dimension, the corresponding agent gets triggered and tries to generate such a dialogue act. For example, the Auto-Feedback Agent gets triggered if a processing problem is recorded in the own processing state of the Cognitive Context. The agent then tries to generate a negative auto-feedback act in order to solve the processing problem (e.g., “Could you repeat that please?” or “Did you say ‘five’?”).

The dialogue acts generated by the DAA are kept in the linguistic context as candidates. Although the dimensions of dialogue acts are considered to be orthogonal, the order of performance and combination of the candidates has to be determined yet, taking into account the relative importance of the dimensions at the given point in the dialogue. Therefore, an additional *Evaluation Agent* monitors the list of candidates and decides which of them can be combined into a complete system utterance for generation, and when.

For the first version of the dialogue manager [6] a limited system functionality was defined, and following from that a simplified version of the dialogue act taxonomy. This simplification includes the merging of SOM and the various dimensions in the IM layer into one dimension IM-SOM, following the observation that utterances with a SOM function very often also have a function in the IM layer.

In Figure 1, a schematic overview of the multi-agent dialogue manager is given. It shows the context model with four components, a set of DAAs, and an Evaluation Agent. The Control module deals with writing the results from language analysis (ultimately in terms of dialogue acts, via DAR) to the Linguistic Context, and sending dialogue act representations of system utterances to language generation (NLG). For the version of the dialogue manager with the richer context model and update mechanism that we present here, an additional Context Manager is introduced, that reads information about new utterances in the Linguistic Context and carries out the necessary updates.

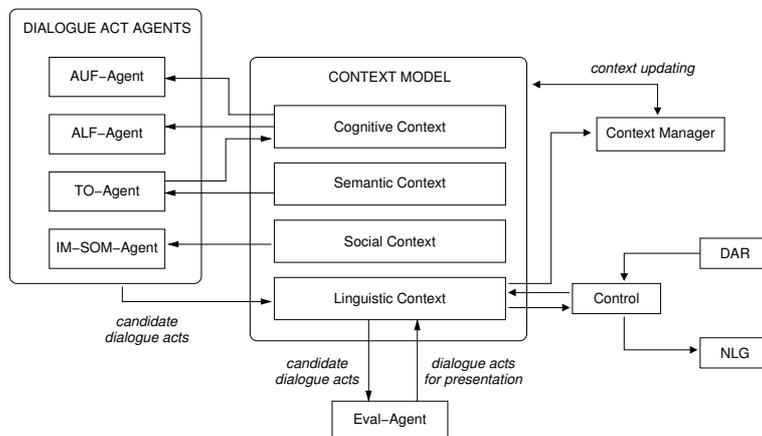


Figure 1: Architecture of the PARADIME dialogue manager.

3.2 Specification of context model

The context model we propose follows the structure according to DIT, as reflected in the feature structure representation of Figure 2.

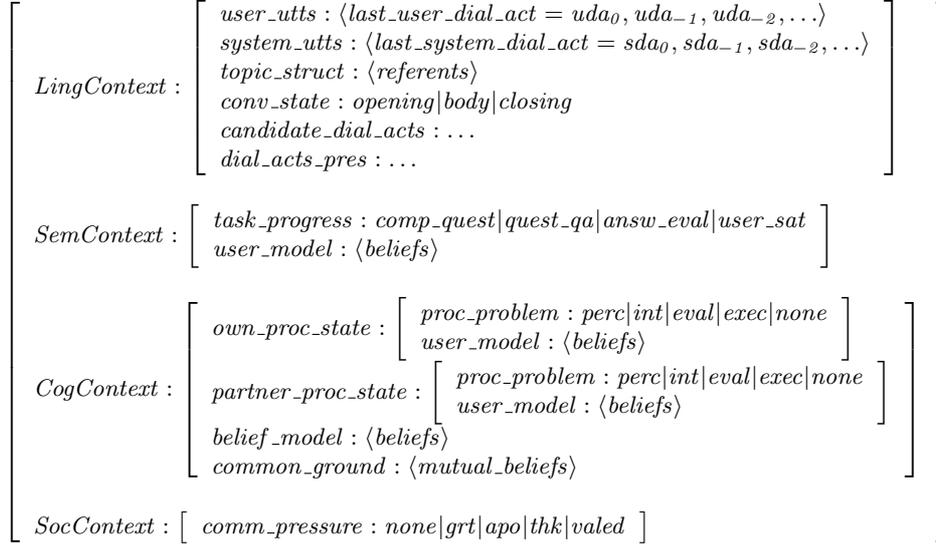


Figure 2: Feature structure representation of the PARADIME context model.

The Linguistic Context contains features for storing dialogue acts performed in the dialogue so far: *user_utts* and *system_utts*. In addition, *topic_struct* and *conv_state* contain information about the topical and conversational structure. The feature *candidate_dial_acts* stores the dialogue acts generated by the DAAs, and *dial_acts_pres* stores the current combination of dialogue acts available for direct presentation as a multifunctional system utterance.

The Semantic Context contains a state variable for keeping track of the question answering process. The feature *task_progress* distinguishes between the states of composing a self-contained question to send to the QA modules (*comp_quest*), waiting for the QA results in case a QA-question has been sent (*quest_qa*), evaluating the QA results (*answ_eval*), and discussing the results with the user (*user_sat*). Besides this task-specific feature, there is a feature *user_model* containing information about the user's beliefs and goals concerning the task-domain. For question answering, this information can be interpreted as a specification of the user's information needs (as built up through the questions asked by the user), and of the user's current knowledge about the domain (as built up through the answers and other information given by the system).

The Cognitive Context contains two features representing the processing states of the system (*own_proc_state*) and the user (*partner_proc_state*) as viewed by the system. Both contain two features: one indicating whether or not a processing problem was encountered, and if so, on which level of processing this happened, and one containing information about the user's beliefs and goals related to the processing state. The Cognitive Context also has a feature *common_ground*, containing beliefs the system believes to be mutually believed. Finally, a feature *belief_model* is used, containing all (non-mutual) beliefs in the context model. This feature is used to have a reference to all beliefs in one place, making part of the update mechanism more convenient. These beliefs have cross-links to other parts of the context model, wherever appropriate (e.g., to the user model of the Semantic Context).

The Social Context is specified in terms of communicative pressures; currently, we only use one feature indicating whether or not a pressure exists for performing a social obligations management act, and if so, for which one (e.g., *comm_pressure*: *grt* indicates a pressure for the system to respond to a greeting).

4 Context update model

The model for updating the context makes explicit how every dialogue act contributes to changing the information state, it defines the types of effects that an utterance provokes in dialogue participants, and it establishes the operations that cause the change of state in the context.

The aspect of the context model related to information transfer and grounding is represented in terms of beliefs [7]. Basically, the types of beliefs we distinguish are represented by means of several operators (weak belief, strong belief, mutual belief, knows value of, wants), that allow to represent the meaning of dialogue acts. As a dialogue evolves, new beliefs are created and existing beliefs may change or be cancelled. Those changes are modelled by means of the operations of *creation*, *strengthening*, *cancellation* and *adoption*. Dialogue acts have different types of effects on dialogue participants: effects of understanding and adopting information in the Addressee, and effects of expectations of understanding and adoption, and strengthening in Speaker and Addressee.

The beliefs can be integrated into the feature structure representation of the context model. Figure 3 illustrates how a belief with a certain operator and containing nested beliefs can be represented.

$$\left[\begin{array}{l} \text{operator : } \textit{belief} \\ \text{agent : } S \\ \\ \text{argument : } \left[\begin{array}{l} \text{operator : } \textit{wants} \\ \text{agent : } U \\ \\ \text{argument : } \left[\begin{array}{l} \text{operator : } \textit{knows} \\ \text{agent : } U \\ \text{argument : } \textit{null} \\ \text{scope : } \forall x.\textit{conc}(x) \\ \text{proposition : } \textit{def}(x) \end{array} \right] \end{array} \right] \end{array} \right]$$

Figure 3: Feature structure representation of $\textit{belief}(S, \textit{wants}(U, \forall x.\textit{conc}(x) \rightarrow \textit{knows}(U, \textit{def}(x))))$.

4.1 The update algorithm

The update algorithm takes as input an abstract representation of a system or user utterance, as recorded in the Linguistic Context. If the system has produced an utterance, this representation consists of a set of dialogue acts. If the user has produced an utterance, the representation depends on the level of understanding reached by the system. If the level of understanding was at least interpretation (INT), it consists of a set of dialogue acts; if the level was at most perception (PER), it consists of merely meta-information about NLP of the utterance, reflecting that no (sufficiently reliable) dialogue act representation could be found. Whether the level of understanding is evaluation (EVAL) and/or execution (EXE) depends on inferences on the context model during updating and during the process of dialogue act generation.

4.1.1 Processing a user utterance

When a user utterance is processed, the context model changes, depending on meta-information resulting from the process of interpretation and the dialogue act information in case of success. Task/domain dialogue acts (creating beliefs about the user's beliefs and goals about the domain) and negative feedback acts (cancelling such beliefs) will provoke changes in the Semantic Context. Problems encountered by the system in processing the user utterance, and negative allo-feedback acts will provoke changes in the Own Processing State of the Cognitive Context. Negative auto-feedback acts, or reasoning about the information in the context model after any kind of dialogue act, will provoke changes in the Partner Processing State of the Cognitive Context. IM-SOM acts will change the information about the conversational structure in the Linguistic Context, and create communicative pressures in the Social Context.

Algorithms 1 and 2 formalize the update mechanism, including the above mentioned context changes.

Algorithm 1 Update algorithm for processing user utterances.

```
if understanding level is none then
  record PERCEPTION-level processing problem in cogn_context:own_proc_state;

else if understanding level is PERCEPTION then
  record INTERPRETATION-level processing problem in cogn_context:own_proc_state;

else
  for all dialogue acts da do
    if dialogue act has dimension-specific CF then
      if dialogue act is (overall) POS-AUTO-FEEDBACK then
        create understanding beliefs UB from understanding and adoption effects created from previous (user)
        utterance;
        record beliefs UB in cog_context:belief_model;
        if understanding level is INTERPRETATION then
          mark beliefs UB as pending;
        end if
        create mutual beliefs MB about expectations of being understood;
        if understanding level is EXECUTION then
          check if beliefs can be adopted, and if so, record the adopted beliefs in cog_context:belief_model;
          create mutual beliefs about expectations (by user) of this adoption of beliefs;
        end if
        else if dialogue act is NEG-AUTO-FB-PERC or NEG-AUTO-FB-INT then
          cancel all (mutual) beliefs created from previous (system) utterance;
        else if dialogue act is NEG-AUTO-FB-EVAL then
          mark all beliefs from understanding created from previous (system) utterance as pending;
          cancel all other beliefs created from previous (system) utterance;
        else if dialogue act is NEG-AUTO-FB-EXE then
          cancel all beliefs about expected adoption created from previous (system) utterance;

        else if dialogue act is NEG-ALLO-FB-INT then
          cancel all beliefs from understanding created in the previous user utterance;

        else if dialogue act is a im-som act then
          create appropriate communicative pressure in soc_context;
        end if

      else if dialogue act has general-purpose CF then
        apply Algorithm 2;
      end if

      check if any (mutual) beliefs can be strengthened;

    end for

  end if
```

4.1.2 Processing a system utterance

When the system has produced an utterance, the dialogue acts representing that utterance also cause a context update. Negative auto-feedback acts deal with processing problems as recorded in the Own Processing State of the Cognitive Context, but actually solving those problems depends on the user's response. IM-SOM acts release the communicative pressure that triggered those acts, and also change the conversational state. The generation of dialogue acts with general-purpose CFs provokes the creation of mutual beliefs about expectations (of the system) of understanding and adoption (by the user), based on the system's existing beliefs corresponding to the preconditions of the dialogue act (i.e., the beliefs that caused this dialogue act to be generated). Space limitations prevent us from giving a detailed algorithm for this update.

Algorithm 2 Update algorithm for processing a user dialogue act with general-purpose CF.

```
create beliefs PB about the user believing the preconditions corresponding to the dialogue act;
record beliefs PB in cog_context:belief_model;
if content type is task-domain then
  record a link to beliefs PB in sem_context:user_model;
end if
if content type is auto-feedback then
  record a link to beliefs PB in cog_context:partner_proc_state:user_model;
end if
if content type is allo-feedback then
  record a link to beliefs PB in cog_context:partner_proc_state:user_model;
end if
if understanding level is INTERPRETATION then
  mark beliefs PB as pending;
end if
if understanding level is EXECUTION then
  if cog_context:belief_model contains beliefs that can be adopted then
    create beliefs AB from adoption;
    create mutual beliefs about (the user's) expectations of adoption;
  end if
end if
```

5 Context management and DA generation: an example

In this section, we will illustrate how the rich context model and update procedure is used for dialogue act generation. Consider the following dialogue fragment, in which the user asks a question about the medical domain, the system answers the question, and then the user asks a follow-up question:

U0: where is the trapezium located?

S1: the trapezium is in the lower neck.

U2: and how should I train it in order to prevent RSI?

Analysing U0 will result in only one dialogue act; it has a general-purpose CF with SC in the task-domain dimension, represented as $WHQ(loc, trap)$, i.e., the speaker asks for the location (loc) of the trapezium muscle ($trap$). According to the update algorithm, this leads to the creation of a number of beliefs, including the following ones:

$$belief(S, wants(U, \forall x. trap(x) \rightarrow knows(U, loc(x)))) \quad (1)$$

$$belief(S, belief(U, \forall x. trap(x) \rightarrow knows(S, loc(x)))) \quad (2)$$

Because the dialogue act belongs to the task-domain dimension, these beliefs are recorded in the user model of the Semantic Context. The first belief indicates that the system believes that the user *wants* to know something about the task-domain. This new information in the Semantic Context involving a user's goal will trigger the TO-Agent to generate a task-domain dialogue act in order to satisfy this goal. In the example, the system has been able to find the information it believes the user wants, and produces a WHAnswer with a SC representing that information. In the update procedure for this system utterance, a belief about the user understanding the system's answer and a belief about the user adopting the information in the answer are created:

$$bel_mut_bel(S, wk_bel(S, belief(U, belief(S, \forall x. trap(x) \rightarrow loc(x) = conc_def)))) \quad (3)$$

$$bel_mut_bel(S, wk_bel(S, belief(U, \forall x. trap(x) \rightarrow loc(x) = conc_def))) \quad (4)$$

The second belief about adoption includes the system's weak belief that the user has adopted the information given by the system. This adoption would mean that the user's goal is satisfied, but, as the system has only a weak belief about this, the system cannot assume this satisfaction yet.

In U2, the user gives (implicit) positive feedback, and therefore, the system creates the following belief in updating his context with U2:

$$\text{belief}(S, \text{belief}(U, \forall x. \text{trap}(x) \rightarrow \text{loc}(x) = \text{conc_def})) \quad (5)$$

This means that the system now strongly believes that the user has adopted the information given, and consequently, that the info-need is considered to be satisfied.

This example showed how the information in the context model is exploited in generating dialogue acts in the case of a non-problematic exchange in the task-domain dimension. However, the context model will also allow for dialogue act generation in the other dimensions, in cases of more complex dialogues in which for example communication problems occur.

6 Conclusion and future research

We have presented an approach to dialogue management involving a rich context model and a mechanism for updating it with dialogue acts from a multidimensional taxonomy. The dialogue manager generates combinations of dialogue acts through several dialogue act agents operating on the context model. In an example we have shown how such a dialogue act agent may be triggered by new information in the context and generate a dialogue act as part of a system utterance to be produced.

The current version of the dialogue manager uses the information in the context model in a rather shallow way, despite the rich (belief) representations that are available. In a next version, the dialogue manager should exploit the beliefs in the context in a more intelligent way. For example, instead of just signalling a new user information need and thereupon sending the question to the QA modules, it can construct improved queries for the QA modules by considering (and reasoning over) the entire user model of the Semantic Context. The same goes for dealing with (problems in) the communication process, both regarding the own and partner processing states.

References

- [1] R. op den Akker, H. Bunt, S. Keizer, and B. van Schooten. From question answering to spoken dialogue: Towards an information search assistant for interactive multimodal information extraction. In *Proceedings of the 9th European Conference on Speech Communication and Technology, Interspeech 2005*, pages 2793–2796, 2005.
- [2] J. F. Allen and C. R. Perrault. Analyzing intention in dialogues. *Artificial Intelligence*, 15(3):143–178, 1980.
- [3] H. Bunt. Dialogue pragmatics and context specification. In H. Bunt and W. Black, editors, *Abduction, Belief and Context in Dialogue*, Studies in Computational Pragmatics, pages 81–150. John Benjamins, 2000.
- [4] H. Bunt. Dimensions in dialogue act annotation. In *Proceedings Fifth International Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, 2006.
- [5] P. R. Cohen and C. R. Perrault. Elements of a plan-based theory of speech acts. *Cognitive Science*, 3:177–212, 1979.
- [6] S. Keizer and H. Bunt. Multidimensional dialogue management. In *Proceedings of the SIGdial Workshop on Discourse and Dialogue*, pages 37–45, Sydney, Australia, 2006.
- [7] R. Morante and H. Bunt. Micro-analysis of the belief transfer in information dialogues. In C. Gardent and B. Gaiffe, editors, *DIALOR'05 Proceedings of the Ninth Workshop on the Semantics and Pragmatics of Dialogue*, pages 135–138, 2005.
- [8] D. R. Traum and S. Larsson. The information state approach to dialogue management. In Jan van Kuppevelt and Ronnie Smith, editors, *Current and New Directions in Discourse and Dialogue*, pages 325–354. Kluwer, Dordrecht, 2003.

HUMANS AND AGENTS TEAMING FOR AMBIENT COGNITION

Masja Kempen Manuela Viezzer Niek Wijngaards

Delft Co-operation in Intelligent Systems Laboratory – Thales Research and
Technology Nederland, P.O. box 90, NL-2600 AB, Delft.

Abstract

In actor-agent teams human and artificial entities interact and cooperate in order to enhance and augment their individual and joint cognitive ergonomic and problem solving capabilities. Also actor-agent communities can benefit from ‘ambient cognition’, a novel further reaching concept than ambient intelligence that hardly takes into account the resource limitations and capabilities changing over time of both humans and agents in collaborative settings. The Dutch Companion project aims at the realization of an agent that takes advantage of the ambient cognition concerning actor-agent system dynamics such that natural emotion-sensitive interaction with an actor over a longer period of time can be sustained. We elaborate on our vision of pursuing ambient cognition within actor-agent systems and present the plans and expected results of the Dutch Companion project.

1. Introduction

Ambient intelligence [13] requires enabling technologies that allow humans to meaningfully interact with their environment, and to obtain support in their activities. This implies a number of interesting and complex cross-disciplinary research and development challenges.

In artificial intelligence, cognitive science and cognitive engineering a main challenge is building sustainable actor-agent systems that can individually and/or collectively anticipate and attend to their own or environmental dynamics. The anticipation and attention measures foreseen and taken by such systems will decisively determine whether these systems can achieve their ambient cognitive goals and accommodate their own and environmental changes, in particular those involving their collaboration [11].

Another primary challenge concerns the optimization of multimodal interactions between humans and agent systems. In this respect current human speech and gesture recognition systems are still lacking an important feature, namely perceiving, understanding and enacting human emotions or other cognitive-ergonomic aspects of humans. Humans expect from other humans and therewith of agents some emotional sensitivity and skills in their interactions.

Summarizing, the question “Can the interaction between human and ambient intelligent systems include emotions?” is an exponent of these arguments for embodying emotional cognition into ambient intelligent networks

There to, the research project ‘Dutch Companion’ started early 2006 to focus on building an autonomous ambient cognitive agent system embodied in the iCat. This system should recognize emotions, take decisions on emotions, and convey emotions via facial expressions, such that it ultimately supports or even augments a person’s cognitive abilities and interacts with this person in an emotional sensible way.

The structure of this position paper is as follows. Section 2 introduces the term Ambient Cognition by describing the challenges for realizing actor-agent communities and teams that go beyond those of ambient intelligence. Section 3 focuses on one particular added value of Ambient Cognition provided by emotional agents. Section 4 describes how the Dutch Companion research project will embody emotion in the iCat. The concluding Section 5 describes expected results for the near and long term future.

2. Beyond Ambient Intelligence

What is currently missing in Ambient Intelligence is ‘emotional and social intelligence’ going beyond user and preference awareness, in other words some indispensable features of *cognition*. We do not imply agents that emulate human actors, but instead augment or complement actors’ cognitive capabilities. We envision ambient intelligence evolving towards *ambient cognition* – an environment which augments human cognition and supports humans in achieving their goals. Not only is responding to and providing with the human with support an important ingredient, but more crucial are ‘services’ such as support delivered at the right time, depending on the situation, context, and preferences, knowledge level, and skills of humans. Intelligent artificial entities may put themselves forward as agents, with which humans can team up and form longer-term relationships. These cognitive agents are capable of profiling their human companions, not only regarding personal preferences, but also regarding moods, emotions, goals, intentions, etc. An important task for the cognitive agents is to reduce the (cognitive) workload of and stress among humans, and facilitate the humans’ continued effectiveness in performing tasks and achieving (joint) objectives.

We regard the human inside its ambient intelligent environment as partaking in an actor-agent team (AAT). In this construct, actor refers to humans, and agents to artificial systems, ranging from sensors to highly intelligent, cognitive systems. Here teams are associated with systems that collaborate, whereas actor-agent communities (AAC) are distributed actor-agent teams of which the performance hinges on communication [14].

Teams are often considered to solely consist of humans [10]. In our opinion, intelligent systems such as agents and robots can also be team-members, equivalent in status to humans. This is in contrast to a large amount of system-level teams and agent research, which concentrates on agent-based support for individual human team members [12]. In our view, agents may also take the initiative and give orders to human (and other agent) team members.

The rationale for forming human only teams is that they perform better than individuals on their own, assuming that the right people are together in a team. Teams are built up out of individuals each with their own strengths, capacities and personalities, which should lead to efficient collaboration.

But what happens when artificial agents become team members? We would expect the team performance to increase. However, actors interacting with agents bring to bear new forms of collaboration, interaction, and communication schemes introducing new hurdles to be taken by the team members. This unmistakably affects team performance as a whole in a rather unpredictable way.

Hoffman et al. [5] emphasize that AACs require analysis as a whole, rather than from an agent-systems point of view alone. They identify actors, agents and context as analysis-triple. For actors, their cognitive capacities, perceptual capacities and goals are listed as distinguishing characteristics, for agents their computational and interface capabilities, and the context is characterized by requirements, constraints and potentials. Herein agents should amplify human intelligence, which implicates compensation for human shortcomings and boosting competencies. This strengthens us in hypothesizing that actor-agent teams perform better than actor-only teams or agent-only teams.

3. Emotional Agents

Most of today’s information processing systems interact with their users via built-in, pre-defined rule-based interfaces. They have little or no knowledge of the environment or their user(s). In fact these systems assume a static user environment, which can seldom be guaranteed. In the best of cases the interaction models contain some built-in knowledge of the user’s capabilities, but lack any knowledge of the current state of the user’s tacit or background knowledge, the user’s situational awareness, skills and actual “state-of-mind”.

This contrasts heavily with the standard human-human interaction, which contain more elaborate patterns of use and situational awareness, thus grounding the belief that the conception and design of complex information systems that are based on cognitive capabilities will be the key differentiator between future products.

Cognitive abilities ranging from autonomous learning to (naturalistic) decision making or perspective taking, will allow information processing systems to recognize their user and understand what they want to accomplish, assess the user's state-of-mind, occupation, knowledge and capabilities and, using this information, focus their attention on and anticipation of the perceived user's needs for information presentation, processing and other support. Moreover, the implementation of reasoning mechanisms and representations that are similar to what humans use, for example because they take into account gestures as communicative means, or they formalize and make use of emotions, will make human-machine interaction more natural and possibly more effective. This improves acceptance and enhances the results of human-machine collaborative tasks. An argued viewpoint on human-system interaction we subscribe to is that of [9]: "In the future, conversational interfaces play the most important role."

Humans are remarkably good at detecting emotions in others and we use this capability (implicitly and extensively) in all our communications. This leads to the assumption that the correct detection of the emotional state of communicative partners is essential for a naturally developing communication and natural interaction. There is an important role put aside for emotions in future intelligent systems, acknowledged by research in areas such as affective computing and social robotics [8,3,4].

The ability to deal with emotions should be addressed both at the system level, by providing an agent with the ability to evaluate the emotional atmosphere of its surroundings, and at the level of personal reasoning, by taking into account how the emotional state affects the deliberation process of the agent itself, and by giving the agent the ability to display its emotional state. The agent's awareness of the emotional atmosphere contextualizes its interaction with the user.

A further research challenge concerns the coupling of perceived emotional behaviors with mechanisms for diverse AAT and AAC concept formation and learning. The challenge is in line with e.g. the proposal of [3] to develop personalized robot companions, where the robot is both individualized, i.e. no two robots will be the same, and personalized, that is its individuality reflects the needs and requirements of the environment where it is operating. Personalization is deemed necessary because of human nature: people have individual needs, likes and dislikes, etc. that a companion would have to adapt to. What is argued is that, rather than relying on an inbuilt fixed repertoire of social behaviors, a robot should be able to learn and adapt to the social 'etiquette' of the people it is living with.

4. Dutch Companion

The Dutch Companion project focuses on augmenting agents such that they are sensitive to emotions of humans and are capable of portraying emotions as well: a step towards ambient cognitive interaction.

4.1. Approach

The main focus of Dutch Companion is on improving the effectiveness of actor-agent interaction through making an agent aware of human emotions and expected social interactions. An agent can then respond not only with appropriate content but also with appropriate emotional state. The agent will have the embodiment of Philips' iCat (see Figure 1; [6]). The iCat is a plug & play desktop user-interface robot, capable of mechanically rendering facial expressions. The iCat will be situated inside the kitchen and learns to take preferences and emotional state into account when suggesting meals or activities to their inhabitants.

The project addresses the following: (i) The ability to focus attention on the user's needs for information and assistance as a function of the user's situation, goals and current (cognitive) capabilities and emotional states; (ii) The ability to adapt in real time to the behavior and responses beyond the mere use of some built-in static user model.

The first point will be addressed by looking at the use of sound recognition and analysis [1] to enhance its situational awareness. The classification of particular sounds from the environment will be used by the iCat to alert the user, depending on the current user context. Via deliberation it determines whether or not the user should be alerted.

The second point will be addressed by detection of emotion in the user's voice, in combination with the emotional output from other modalities, such as facial recognition and gestures. The perceived emotional state can then be used as a controlling input for the iCat's deliberation.

The personal and social based deliberation designs and implementations will be based on agent-oriented programming, in particular 3APL [2]. A central element of 3APL is the way the agent deliberates how to perform plans given certain goals and beliefs, in particular which goals and plans to select and pursue, and which goals and plans to revise under the circumstances given. This so-called 'deliberation cycle' has to be extended to also include the effect of emotional states, as well the influence of social aspects such as norms and obligations arising from the social context [7].

In addition we will couple the emotion recognition with the communication and dialogue iCat module, thus changing the nature of the dialogue and the content and modality of the communication depending on the emotional state of the user.



Figure 1. Philips' iCat

Note that interacting with agents requires integration of different fields in AI, cognitive science, computer science, etc. For example, speech recognition is still difficult in uncontrolled environments. Equally challenging still is computer vision. If we foresee a future of actor agent communities, all these issues on the sensor/input side, as well as cognitive and actuator side need to be cross-disciplinary resolved. Starting by building a small system and deploy it in an operational setting, will uncover new unforeseen research issues.

Furthermore, note that such ambient cognitive systems will always serve specific purposes and not yet provide generic solutions to 'universal' problems. In different situations or context, people have different tasks, or make more or less use of certain abilities, so

artificial systems should know what to support in which context. For example, a human driving a car needs different support than human inside its home.

4.2. Scenarios

In this section a number of illustrative scenarios are described to highlight the role and use of emotions via the 'iCat' as the personal assistant of the lead actor: Anna. The iCat has some built-in and/or acquired knowledge about Anna (the user) regarding health and dietary history, her food preferences, dietary requirements/restrictions and the like.

Collaborative Planning

Anna has her day off and just cleaned the kitchen. Anna looks happy because she loves her kitchen to be spic and span. It is 4 o'clock and the iCat, who is linked to Anna's calendar, alerts Anna that she has to pick up her child Lisa from day-care in an hour. In addition, the iCat asks if she wants a suggestion for a recipe for dinner this evening. Anna would like that. iCat takes into account what Anna has cooked in the past week and also the current stock in the fridge. iCat looks up a recipe in a database on the Internet to find a recipe with cauliflower, because that vegetable is still in the fridge since two days. He suggests cauliflower with a béchamel-cheese sauce, potatoes and fried chicken. iCat looks happy, smiles and tells Anna that all the ingredients are present and that there is no need to rush to the store. He asks Anna if she would like her favorite music to be put on and directs Anna to go and relax on the couch for half an hour before she should be off to the day care centre, and can start cooking when Anna gets home with Lisa. He knows how much Anna enjoys and needs these short moments for relaxation. iCat controls the stereo in the living room so that Anna already hears her favorite music station and has activated the smell diffuser around the couch, so that Anna lies in an invigorating aroma.

This scene shows how a cognitive emotional agent can support a human in scheduling and planning daily activities. Note that this support can be reactive as well as proactive. It shows how the agent makes use of personal preferences and controls devices around the house. The scene also shows that iCat engages in meaningful conversation, about both alerting a day-care deadline and what to make for dinner. Thus iCat strives to avoid frustration by constantly interfering with Anna's activities. In addition, iCat aims to support Anna, including facilitating relaxation now, so that Anna has more energy available for collecting Lisa from daycare and cooking dinner later.

Monitoring and Responding

Time to start cooking. Lisa likes to help her mum cook. Anna puts her on a stool and lets her play with some water and a spoon in the sink. Anna has cooked before, so iCat knows not to interfere and unnecessarily advise her to start with peeling the potatoes, because they need longest preparation time. What iCat does observe is that Anna left a rather sharp knife on the counter within reach of Lisa. iCat looks shocked and informs Anna about the knife. Anna looks with fear and surprise and takes the knife away. iCat then smiles and tells both Anna and Lisa, winking, that it is all right and how lucky Lisa is to have him.

This scene highlights 2 issues. First of all, it is imperative that the agent knows the skill levels of its user regarding tasks and their context. Once iCat starts to proactively give advice to its user that the user might not want or already know it, Anna becomes annoyed by iCat, and will turn it off. Second, it is important that iCat has some knowledge of estimating social situations. Having a child next to you, whilst cooking, places an additional demand on one's attention. Focusing on two different things at the same time is a difficult cognitive task for humans let alone for agents. If iCat knows this and recognizes Anna's cognitive weakness and knows about Lisa's unpredictability, it can proactively distribute its own attention on detecting potential dangerous situations that may either cause harm to Anna or Lisa. This is an example of augmenting human social cognition.

Coaching a human

Time for the béchamel-sauce. Anna is anxious to get this done properly, as she finds this difficult to do. And yes, when adding the milk and stirring, the sauce starts clotting instantly. Anna looks disappointed and mutters something under her breath. iCat notices this, switches to the role of a coach, and puts Anna at ease, telling her to keep on stirring, because clotting is part of the process anyway. He tells her to turn down the heat and keep stirring, smiling encouragingly. It works and Anna looks relieved and smiles at the iCat. He responds back with a “well done”. However, Anna is so engaged in the sauce making that she has neglected the chicken that is now starting to burn. The sensitive smell sensors detect burning meat and this triggers the iCat to warn Anna with a shocked expression: “Turn down the heat for the chicken, before it burns!”

The final scene shows the role of emotion detection and production. iCat’s awareness of how humans use emotions allows him to take that into account in the deliberation process and take appropriate actions. And it allows him to convey appropriate facial expressions, gestures and dialogue. Using appropriate emotions when responding, can achieve certain effects at the human recipient, including making messages more acceptable. At least this is what we hope and need to test.

5. Expected Results

We described the importance and use of emotions in actor-agent teams and communities. The Dutch Companion project is expected to deliver concrete results based on experimentation and research regarding embodied emotional agents and natural actor-agent interaction. Our focus is explicitly neither on ICT architecture nor infrastructure; we assume them to be available in adequate quantity and quality for our purposes. Let us summarize the expected near- and long-term results; the near-term refer to architectural and experimental aspects, and the long-term to future work.

5.1. System Architecture

From an overall AAT or AAC system’s point of view, we foresee Dutch Companion to deliver implemented new functional system architectural insights by combining (i) sound recognition, (ii) emotion recognition, (iii) affect controlled deliberation, (iv) concept formation, (v) affect controlled communication and dialogue, and (vi) cognitive system architectures.

5.2. Validating Experiments

We develop and deploy a demonstrator framework for autonomous cognitive systems (ACSs) that show the added value of AAT and AAC system architecture. In particular, for these purposes we assess the iCat demonstrator with its users. This assessment will be mainly conducted in the Home Lab of Philips Research in Eindhoven. The iCat will be placed in the kitchen of the Home Lab and connected to the home-network. We test how users will interact with the iCat, focusing on the effectiveness, enjoyment and acceptance of using such a domestic companion to assist and accompany a user doing (kitchen) tasks.

In the long-term, it is of interest for our stakeholders to find effective forms of actor-agent interaction. For example, should each human have its own personal agent, should one agent serve more than humans, or should one human have more than one agent? Teams and communities consist of more than one or two.

5.3. Towards True Teams and Communities

Bonding within either teams or communities to some extent assumes shared awareness, norms and goals: ‘getting into, acquainted, and involved with each other’. This implies the following mutual reciprocal processes to be in place:

- Forming a sound theory for AAT and AAC systems
- Taking into account human cognition, social norms and culture as well as technological advancements.
- Implementing and testing related empirical (cognitive) AAT and AAC models.

Regarding emotions, what predictable but diverse individual traits and activity patterns of agents should be modeled and realized in relation to one or more actors within specific situational contexts to achieve certain goals? For example, should an agent be a playmate or a teacher?

5.4. Towards Ambient Cognition

The current abundance and omnipresence of ICT architectures and infrastructures enable ubiquitous, pervasive, sentient, and ambient intelligent computing, communication, cooperation and competition of both artificial and societal organizations and structures. However, they appear to us as merely *nice to have*s in a rather unstructured and unorganized ICT infrastructure. In general they still lack cognitive engineering capabilities, namely those for anticipation and selection of attention.

Instead of perpetually handcrafting standalone ICT infrastructures and integrating them, smart human-system network interaction paradigms are needed such that ambient intelligent or actor-agent systems can continuously select and embody (after learning) suitable anticipatory and selection of attention schemes. Only then can those novel integrated ICT architectures and infrastructures be brought to life. In short, new paradigms like those proposed in [11] for co-existence and co-evolution of humans, machines and their extensions are needed in order to simultaneously sustain all.

Bringing ICT to life requires moving from nice to have features provided by ambient intelligence to *need to have* features, services, and better quality of life provided by ambient cognition. This shift that we envision will be enabled by building autonomous cognitive systems such as one headed for by the Dutch Companion project. This is fully in line with ongoing research activities in the area of ambient intelligence, where the emphasis is on the development of greater user-friendliness, usability, more efficient services support, and user-empowerment.

Acknowledgements

This work is supported by SenterNovem, Dutch Companion project grant nr: IS053013 and the Dutch Ministry of Economic Affairs, ICIS research program, grant nr: BSIK03024. The Dutch Companion and the ICIS project are both hosted by the DECIS Lab (<http://www.decis.nl>). We explicitly acknowledge Dr. A. H. Salden for his invaluable comments regarding the past, present and future of ambient intelligence.

References

- [1] T. C. Andringa. *Continuity Preserving Signal Processing*. Ph.D. Thesis, University of Groningen, The Netherlands, 2002.
- [2] M. Dastani, M.B. van Riemsdijk, F. Dignum and J.-J. Ch. Meyer. A Programming Language for Cognitive Agents: Goal-Directed 3APL, In M. Dastani, J. Dix, A. El Fallah-Seghrouchni, editors, *Programming Multi-Agent Systems (Proc. ProMAS 2003)*, LNAI 3067, Springer, Berlin, pages 111-130, 2004.
- [3] K. Dautenhahn. Robots We Like to Live With?! - A Developmental Perspective on a Personalized, Life-Long Robot Companion. *Proceedings IEEE Ro-man 2004, 13th IEEE International Workshop on Robot and Human Interactive Communication September 20-22, 2004* Kurashiki, Okayama Japan, IEEE Press, pages 17-22, 2004.

- [4] T. Fong, I. Nourbakhsh and K. Dautenhahn. A Survey of Socially Interactive Robots. *Robotics and Autonomous Systems*, 42, pages 143-166, 2003.
- [5] R. Hoffman, P. Hayes, K. M. Ford and P. Hancock. The Triples Rule. *IEEE Intelligent Systems*, 17(3), pages 62-65, 2002.
- [6] iCat Research Platform, Experimentation Platform for Human-Robot Interaction Research, available from <http://www.hitech-projects.com/icat/platform.php>
- [7] J.-J. Ch. Meyer. Reasoning about Emotional Agents. In R. López de Mántaras & L. Saitta, editors, *Proc. 16th European Conf. on Artif. Intell. (ECAI 2004)*, IOS Press, pages 129-133, 2004.
- [8] R. Picard. *Affective Computing*. MIT Press, 1997.
- [9] H. Raffler. *Intelligente Autonome Systeme: Trends, Anwendungen und Interaktion*, presentation from Siemens AG, Corporate Technology, http://www.akinfor.de/image/intelligente_autonome_systeme.pdf, 2003.
- [10] E. Salas, T. L. Dickinson, S. A. Converse and S. I. Tannenbaum. Toward an Understanding of Team Performance and Training. In R. W. Swezey and E. Salas, editors, *Teams: Their Training and Performance*, Norwood, NJ: Ablex, pages 3-29, 1992.
- [11] A. H. Salden & M. Kempen. Sustainable Cybernetic Systems - Backbones of Ambient Intelligent Environments. In P. Remagnino, G.L. Foresti and T. Ellis, editors, *Ambient Intelligence: a Novel Paradigm*, Springer, pages 213-238, 2005.
- [12] Sycara, K. and Lewis, M. Integrating Intelligent Agents into Human Teams. In E. Salas and S. Fiore, editors, *Team Cognition: Understanding the Factors that Drive Process and Performance*, Washington, DC: American Psychological Association, 2004.
- [13] W. Weber, J. Rabaey and E. Aarts. *Ambient Intelligence*. Springer-Verlag, 2005.
- [14] N. Wijngaards, M. Kempen, A. Smits and K. Nieuwenhuis. Towards Sustained Team Effectiveness. In G. Lindemann et al., editors, *Selected revised papers from the workshops on Norms and Institutions for Regulated Multi-Agent Systems (ANIREM) and Organizations and Organization Oriented Programming (OOP) at AAMAS'05*, LNCS, vol. 3913, pages 33-45, 2006.

On Market-based Allocation of Flow Resources and Transport Network Characteristics

Koen Kok ^{ab}

Hans Akkermans ^a

^a *Vrije Universiteit Amsterdam, Faculty of Science, Business Informatics Section.*

^b *Energy research Center of the Netherlands (ECN), Intelligent Energy Grids,
P.O. Box 1, Petten. Email: j.kok@ecn.nl*

Abstract

Market-based allocation of flow resources is a specific field of study in artificial intelligence research. The main focus of the research in this field has been on optimally allocating the flow commodity among competitive agents and on efficient algorithms for performing this task. A topic that seems underexposed is the interaction between the market algorithms and the underlying transport network. Specific network characteristics, like maximum connection capacities and transport losses, may constraint or influence the market results.

In this paper we present a general theory of locational pricing, which is based on a widely accepted theory of electricity pricing taken from the field of power systems economics. We reformulate and generalize this theory to make it applicable to agent-based allocation of flow resources in general. Further we describe a possible implementation in an agent-based market algorithm and briefly discuss possible directions for future explorations on this topic.

1 Introduction

Electronic markets form a class of mechanisms that provide distributed decision making among self-interested agents. Electronic markets are known to allocate scarce resources to individual competitive agents in a Pareto optimal way. As such, electronic markets are studied in computer science fields such as multi agent systems and electronic commerce. A specific research subfield is that of allocation of flow resources. Flow resource allocation problems are characterized by the existence of a transportation network, i.e. a set of locations with links identifying feasible transportation operations. Further, flow commodities are continuous, i.e. infinitely divisible. Relevant industrial applications can be found in electrical power management, where large numbers of electricity producing and/or consuming devices are coordinated in a distributed fashion in order to optimize total energy balance over time [3, 6, 11].

In the computer science literature the main emphasis has been on optimally allocating the flow commodity among competitive agents and on efficient algorithms for performing this task (see for instance [10,12]). The market-based resource allocation algorithms resulting from this research field do not take characteristics of the underlying transport network into account. For instance, capacity constraints in the transport network may have impact on the feasibility of a particular allocation outcome. By not considering these constraints in the used market algorithm, it is implicitly assumed that the network has infinite, or at least sufficient, capacity. Other network characteristics relevant to certain application cases are transport losses and flow-dependent transport costs. As far as we are aware of, the influence of network characteristics on allocation outcomes is not yet described from the viewpoint of market-based flow commodity allocation in multi agent systems. This seems to be an open problem in artificial intelligence research.

However, network-market interaction is an actual problem in the electricity transmission network of several world regions. As a result, in the combined research domains of economics and power systems engineering, there is a fair amount of knowledge on this topic. A common way of incorporating characteristics of the electricity transmission network into the outcomes of electricity wholesale markets is introducing locational aspects to pricing of electricity. In both Europe

and North America locational pricing is being introduced for cross-border trade regulation and transmission system congestion management [4, 5].

The concept of locational pricing, or spot pricing, for networked services was first introduced by Vickrey [8]. He presented it under the name of “reponsive pricing” in a study related to public utility services in general. In his first text the emphasis was on “curtailment premiums” rather than on marginal costs varying in time and location. In later documents (e.g. [9]) on electricity pricing he developed the idea further, for instance by incorporating short-term marginal costs. Another early publication on the concept is by Agnew, who developed a model for varying congestion tolls in highway and communications networks, incorporating capacity constraints and time delays. A detailed historic bibliography on the topic of locational pricing can be found in [7].

During the 1980s, Bohn, Schweppe, *et al.* developed a comprehensive theory for spot pricing of electricity [1, 2]. The book [7] that resulted from this research became a standard work in power systems economics. The formulation of their theory is highly oriented towards bulk transmission and wholesale trade of electricity, both in terminology as in modeling details. In this document we take the core of this theory and generalize it to allocation of flow commodities in general. Further we reformulate it in computer science terminology. This generalized theory of locational pricing is presented in section 2. To make this theory applicable in agent-based resource allocation, one needs a network flow model and a market optimization algorithm. In section 3 we describe a general network flow model that can be used for this purpose. A possible market algorithm for locational pricing is described in section 4. This section also includes some examples in subsection 4.3. Section 5 ends the article with conclusions and some directions for future research.

2 A General Theory of Locational Pricing

The core of the electricity spot pricing theory of Schweppe *et al.* [7] is a decomposition of locational prices into specific price components. In this section we generalize this decomposition to one suitable for pricing of flow commodities in general.

2.1 Price Decomposition

A locational price is determined by the supply/demand conditions that exist at a certain time t . Further, this price is influenced by both the characteristics of the transport network as well as the spacial distribution of supply and demand in the network. More precisely, locational prices in a transport network at a certain time depend on:

- **Demand:** The total demand in the network and the corresponding customer benefit levels.
- **Supply:** The availability and costs of production (including purchases from outside the network, if any).
- **Network:** The availability and costs of the transport network, plus, depending on the physical characteristics of the flow commodity, network losses and/or changes in the amount of commodity stored in the network itself.
- **Spacial demand/supply distribution:** The specific locations of production and consumption in the network.

The locational price at network node k is denoted as $\rho_k(t)$ and can be decomposed into five price components:

$$\begin{aligned}
 \rho_k(t) = & \quad \lambda(t) && \text{[Production Marginal Costs]} \\
 & + \gamma_{QS}(t) && \text{[Production Quality of Supply]} \\
 & + \eta_{QS,k}(t) && \text{[Network Quality of Supply]} \\
 & + \eta_{L,k}(t) && \text{[Network Marginal Losses]} \\
 & + \eta_{M,k}(t) && \text{[Network Marginal Maintenance]}
 \end{aligned} \tag{1}$$

Although not all of these components are equally relevant in the scope of this text, we give a short description of each component. In the paragraphs that follow we will describe the most relevant components in greater detail.

- $\lambda(t)$: The (short-run) marginal costs for producing the flow commodity and supplying it to the network with respect to the demand at time t . Also referred to as “system lambda”. Depending on the particular commodity this may include (a combination of) costs for raw materials, fuel, maintenance, etc.¹ If in a market-based setting network characteristics are not being considered, this component corresponds with the general equilibrium price.
- $\gamma_{QS}(t)$: The Production Quality of Service component arises when total production limits are approached. The component serves as a “curtailment premium”, used at critical situations, when the total supply $s(t)$ exceeds the critical production level $s_{CRIT}(t)$. This level is based on available production capacity and operating reserve requirements. Most of the times, $\gamma_{QS}(t)$ is very small or zero with a large, rapid increase when the critical production capacity is being approached. In the latter case, $\gamma_{QS}(t)$ is increased to the point where the total demand is lowered to equal the critical production capacity.
- $\eta_{QS,k}(t)$: The Network Quality of Supply component $\eta_{QS,k}(t)$ is a price mechanism to allocate the use of scarce network capacity. Analogously with $\gamma_{QS}(t)$, the Network Quality of Supply component becomes large in magnitude when the capacity of the network is being approached. This is one of the price components that brings locationality into the pricing scheme: the magnitude of $\eta_{QS,k}(t)$ is dependent of the location in the network and can be different for each network node k . Each $\eta_{QS,k}(t)$ is chosen in such a way that all network flow magnitudes are less than or equal the maximum capacity of the individual network connections. In paragraph 2.2 we will look into this component in more detail.
- $\eta_{L,k}(t)$: The transportation of certain flow commodities, like for instance electrical energy, is subject to transportation losses. These losses can be seen as network operating costs. The magnitude of network losses is dependent on network characteristics and flow magnitudes. Since different network locations will face different losses, different network nodes k may have different magnitudes for $\eta_{L,k}$. In paragraph 2.3 we will look into this component in more detail.
- $\eta_{M,k}(t)$: This price component reflects the marginal network maintenance costs. Depending on the physical characteristics of a specific flow commodity f , transporting f over a network connection may inflict maintenance costs. The $\eta_{M,k}(t)$ gives the incremental network maintenance costs resulting from an increase in demand at network node k . In paragraph 2.4 we will look into this component in further detail.

Since this document focusses on the influence of the underlying transport network on the outcome of the resource allocation problem, the production quality of service term $\gamma_{QS}(t)$ is out of scope here. Hence, the locational price as used in the remainder of this text has the following form:

$$\rho_k(t) = \lambda(t) + \eta_{QS,k}(t) + \eta_{L,k}(t) + \eta_{M,k}(t) \quad (2)$$

In the following three subsections we assess the η -components, followed by a subsection on calculation of $\rho_k(t)$.

2.2 Network Quality of Supply

Assume in a network one line, line i with flow $z_i(t)$, is overloading. Then a possible form of $\eta_{QS,k}(t)$ is:

$$\eta_{QS,k}(t) = \theta_{QS,i}(t) \frac{\partial z_i(t)}{\partial d_k(t)} \quad (3)$$

where $d_k(t)$ is the demand located at node k , and $\theta_{QS,i}(t)$ has units of marginal costs. Then, the price component at network node k resulting from this overloaded line i is equal to some term $\theta_{QS,i}(t)$ multiplied by the incremental flow over i as caused at node k . A market clearing mechanism can be used to find an appropriate value for $\theta_{QS,i}(t)$. Then, during price forming, $\theta_{QS,i}(t)$ is adjusted until customers and producers respond to change usage and production so that the line overload does not occur.

¹In the original text of Schweppe *et al.* [7] the system lambda is further decomposed in two components, common to electricity generation: γ_F , generation marginal fuel and γ_M , generation marginal maintenance.

Note that $\eta_{QS,k}(t)$ can both be positive or negative, dependent on the sign of the partial derivative term. Those locations k where an increase in demand $d_k(t)$ leads to a decrease in flow $z_i(t)$ have a negative $\eta_{QS,k}$. Since both producers and consumers at k see the same locational price $\rho_k(t)$, and since increasing demand at k will have an equal effect as decreasing supply at k , both producers as consumers have equal incentive to respond to prevent the line overload.

Further, note that actors having a higher ‘network distance’ from an overloaded line will have a lower influence on the flow over that line. Hence, their $\partial z_i(t)/\partial d_k(t)$ is lower and, thus, their incentive to respond is lower.

Equation (3) gives $\eta_{QS,k}(t)$ for the situation where only line i overloads. The full equation becomes:

$$\eta_{QS,k}(t) = \sum_{i=1}^{N_L} \theta_{QS,i}(t) \frac{\partial z_i(t)}{\partial d_k(t)} \quad (4)$$

where N_L is the number of lines in the network.

2.3 Network Losses

The marginal cost for network losses at location k is given by:

$$\eta_{L,k}(t) = \lambda(t) \frac{\partial L(t)}{\partial d_k(t)} \quad (5)$$

where $L(t)$ equals the total network losses at time t . Thus, the network losses price component at node k is equal to the marginal production price multiplied by the incremental system losses as caused at k . When the losses of an individual line depend on the actual line flow, then $L(t) = \sum_i L_i[z_i(t)]$, and (5) can be expanded into:

$$\eta_{L,k}(t) = \lambda(t) \sum_{i=1}^{N_L} \frac{\partial L_i[z_i(t)]}{\partial z_i(t)} \frac{\partial z_i(t)}{\partial d_k(t)} \quad (6)$$

2.4 Network Maintenance

The marginal network maintenance component can be defined as:

$$\eta_{M,k}(t) = \frac{\partial N_M(t)}{\partial d_k(t)} \quad (7)$$

where $N_M(t)$ are the total network maintenance costs caused at time t . When $N_{M,i}[z_i(t)]$ is (a model of) the maintenance cost of a given line i , to a certain extend depending on the line flow $z_i(t)$, (7) can be expanded to:

$$\eta_{M,k}(t) = \sum_{i=1}^{N_L} \frac{\partial N_{M,i}[z_i(t)]}{\partial z_i(t)} \frac{\partial z_i(t)}{\partial d_k(t)} \quad (8)$$

2.5 The Locational Price

Substituting the above results in (2) yields (omitting the time dependency from here on):

$$\rho_k = \lambda + \lambda \frac{\partial L}{\partial d_k} + \sum_{i=1}^{N_L} \theta_{QS,i} \frac{\partial z_i}{\partial d_k} + \frac{\partial N_M}{\partial d_k} \quad (9)$$

In a market-based setting the nodal demands d_k are ordinary, or *Walrasian*, demand functions. Thus, the nodal demand will be a function of the local price ρ_k . Each demand function $d_k(\rho_k)$ may be an aggregation of the demand functions of a number of agents located at k . Note that a positive value of d_k indicates a subtraction of the commodity at node k , where a negative value indicates an injection.

Calculating $\rho_k(t)$ for all nodes $k = 1 \dots N_n$ resolves into a search for a set of values for λ and $\theta_{QS,i}, \forall i$ that yield prices ρ_k and demands $d_k(\rho_k)$ such that the following constraints are met:

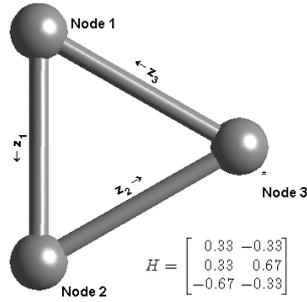


Figure 1: A triangular network with corresponding transfer matrix H . The positive line directions are shown by the arrows in the line labels. Node 3 is chosen to be the swing node and the resistances of the individual lines are chosen to be equal.

$$\sum_{k=1}^{N_n} d_k(\rho_k) = L + \Delta S \quad (10)$$

$$|z_i| \leq z_{i,max}, \forall i \quad (11)$$

The first constraint (10) is the commodity balance constraint. The total demand and supply in the network must equal the total network losses L plus the change in the amount of commodity stored in the network itself, denoted by ΔS . Note that the physics of the commodity in question determine the existence of L and ΔS and how they are modeled. The inequality (11) gives line capacity constraints for each individual line in the network.

3 Network Flow Analysis

Naturally, network flow calculations will play a central role in market algorithms for locational pricing. Regardless of the underlying physics, for most types of commodities a (first-order approximate) flow model can be based on a network transfer matrix. A network transfer matrix holds the relation between the injections y_k (local supply minus local demand) at all network nodes k and the line flows z_i at all lines i :

$$z = Hy \quad (12)$$

The individual matrix elements $H_{i,k}$ represent the influence of the injection at node k on the flow along line i . Thus:

$$H_{i,k} = \frac{\partial z_i}{\partial y_k} \quad (13)$$

The mathematics behind the construction of network transfer matrices does not allow for all nodes k to be part of the matrix, since this would result in a singular matrix. In order to solve this problem, one node is left out of both H and y . This node is generally referred to as the “swing node”, as the injection at that node follows from the flow calculation as a ‘balancing item’, rather than being given as an input as is the case with the other y -values. The swing node is generally indicated with a star (*). It’s injection y^* follows from the commodity balance constraint:

$$y^* = - \left[\sum_k y_k + L + \Delta S \right] \quad (14)$$

where L equals the total network losses and ΔS denotes the change in the amount of commodity stored in the network itself. The magnitude of L and ΔS , if they exist at all, depend on the underlying physics of the commodity in question.

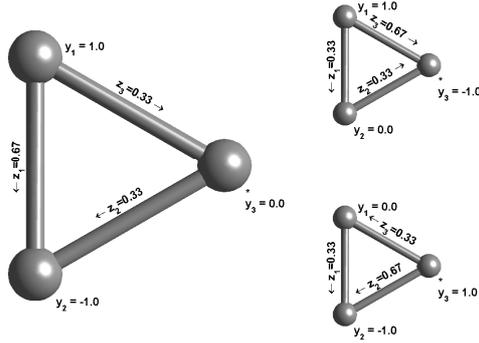


Figure 2: Network flows as a result of injections $y_1 = 1$ and $y_2 = -1$ (right). Superpositional decomposition of these flows into $y = (1, 0)^T$ (top left), and $y = (0, -1)^T$ (bottom left). In these figures, the line labels give the absolute flow magnitudes, the arrows indicate the actual flow directions

Figure 1 shows a triangular network, composed of three nodes and three lines, with the corresponding network transfer matrix. Note that each column k of the matrix describes the flow path of the commodity injected at node k to the swing node. This is visualized in figure 2. The left side of the figure shows the resulting flows for injections $y = (1, -1)^T$, in a network model without losses and inherent storage:

$$\begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix} = \begin{pmatrix} 0.33 & -0.33 \\ 0.33 & 0.67 \\ -0.67 & -0.33 \end{pmatrix} \begin{pmatrix} 1 \\ -1 \end{pmatrix} = \begin{pmatrix} 0.67 \\ -0.33 \\ -0.33 \end{pmatrix}$$

The resulting flows can be regarded as a superposition of two effects, as shown in the right side of the figure. Firstly, injecting commodity at node 1 and the resulting flow to the swing node. And secondly, a subtraction at node 2, resulting in a flow from the swing node.

Note that flow analysis is ‘injection-oriented’, where the locational pricing theory is ‘demand-oriented’. It may be clear to the reader that the nodal injection y_k has a definition opposite to the nodal demand d_k :

$$y_k = -d_k(\rho_k) \quad (15)$$

4 A Market Algorithm

In this section we outline one possible market algorithm for applying the locational pricing theory of section 2 using the load flow model of section 3.

4.1 Outline

In an agent-based market solution, there exists a market-facilitating software agent located on one central place. This market agent holds knowledge on the network topology and characteristics in the form of the network transfer matrix. Further, it holds models of transport losses, inherent storage and network maintenance costs. Each network node has an associated agent that represents all demand and supply at the node in question. For each market run, each nodal agents aggregates the demand functions of the demand and supply sources directly attached to the node into a nodal demand function. The central market agent collects all these nodal functions which are input to an algorithm based on classical optimization theory.

Using the network transfer matrix the injection at the swing node can be computed in two distinct ways. Firstly, the swing-nodal injection follows from the flow analysis. Substituting (15) into (14), and assuming the swing node has the highest node number, yields:

$$y_1^* = \left[\sum_{k=1}^{N_n-1} d_k(\rho_k) - L - \Delta S \right] \quad (16)$$

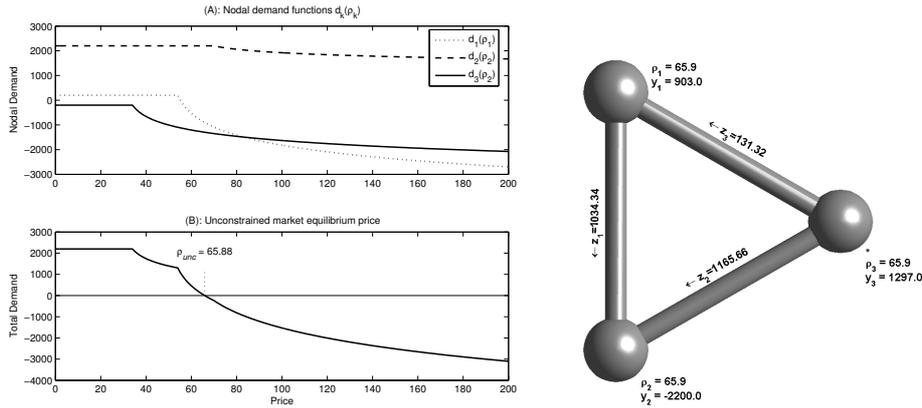


Figure 3: *Left*: (A): Nodal demand functions $d_k(\rho_k)$ for $k = 1 \dots 3$. (B): Aggregated demand function and unconstrained market equilibrium price. *Right*: Resulting network flows.

Secondly, the swing node injection follows from the local demand function and the local price:

$$y_2^* = -d_k(\rho_k), \text{ with } k = N_n \quad (17)$$

Any set of prices $\rho_k, k = 1 \dots N_n$ that results in equal values for y_1^* and y_2^* complies with the commodity balance constraint (10). Accordingly, the optimization algorithm searches the space defined by $(\lambda, \theta_{QS,1}, \theta_{QS,2} \dots \theta_{QS,N_i})$ for a root of $y_1^* - y_2^*$, under the line capacity constraints (11).

4.2 Price Component Interpretation

Using the type of flow calculations as described in section 3, the different price components gain a special interpretation either due to the way of calculation or simplifications in the flow model.

Being the balancing item in the flow calculation, the injection at the swing node has no modeled influence on any flow in the network. As a result of this simplification in the flow model, the η -terms are zero for the swing node. So, λ turns out to be the price at the swing node.

As described earlier, the result of the load flow calculation of section 3 is a superposition of all nodal injections being transported to the swing node. The H -matrix describes the flow paths from every individual node to the swing node. The losses component $\eta_{L,k}$ are the losses of transporting the injection at k to the swing node. A similar behavior can be seen for the network maintenance component $\eta_{M,k}$, when a flow-dependent cost model is used as in (8).

4.3 Example

In this example we apply the described algorithm on the triangular network of figure 1. We assume the nodal demand functions to be as depicted in graph (A) in figure 3. Further we assume absence of network losses and inherent storage (i.e.: $L = 0$ and $\Delta S = 0$), and zero network maintenance costs (i.e.: $\eta_{M,k} = 0, \forall k$). When all three lines in the network have sufficient capacity, then all nodes will have the same unconstrained equilibrium price ρ_{UNC} , as shown in graph (B). The righthand side of the figure shows the resulting network flows for this case.

Figure 4 shows how differences in the locational price ρ_k emerge when one or more lines are constrained. The left figure shows the market outcome and resulting flows for the case when $z_{MAX,2}$, the maximum flow for line 2, is set to the value of 1100. As one would expect, the line constraint has the biggest effect on the prices of the two nodes directly attached to the constrained line. The price in the consuming node 2 goes up to lower demand there, while the price in the producing node 3 goes down in order to lower the injection at that location. The right figure shows how the situation changes when an extra constraint $z_{MAX,1} = 800$ is added. In that case, the total network feed into node 2 is limited to 1900, which dictates the local price in that node. The prices in the other two nodes are chosen such that the respective nodal injections result in the right proportions of the two line flows z_1 and z_2 .

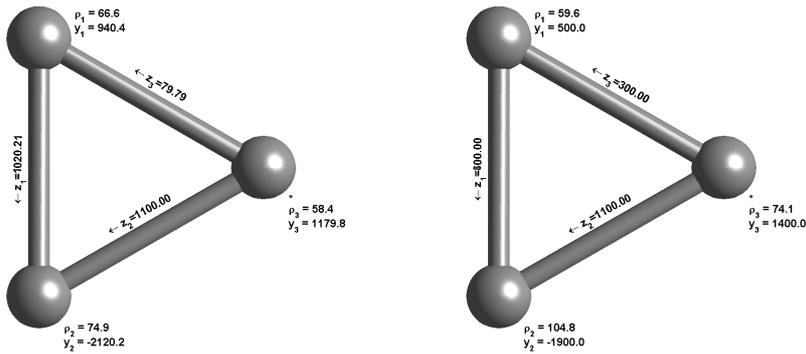


Figure 4: Locational differences in commodity price ρ_k caused by line capacity constraints. *Left*: Line 2 is bound to a maximum line flow of $z_{MAX,2} = 1100$. *Right*: Both lines 1 and 2 are bound to $z_{MAX,1} = 800$ and $z_{MAX,2} = 1100$, respectively.

5 Conclusion and Future Research

The general theory of locational pricing as presented in this paper provides a framework for incorporating network characteristics in market-based solutions for allocation of flow commodities. These characteristics either put constraints on the market outcome (e.g. maximum network connection capacities) or influence the outcome (e.g. transport losses and other network–usage related costs).

Topics for future research include:

- **Algorithms** that can efficiently be applied to large-scale networks or to specific network topologies.
- **Case studies** on the application of the framework for specific commodity types.
- Other **flow modeling types** and their influence on the **market algorithm design**.

Acknowledgment

The European Commission partially supported this research in the context of the FENIX project (*Flexible Electricity Network to Integrate the eXpected ‘energy evolution’*), Nr SES6 - 518272.

References

- [1] Roger E. Bohn. *Spot Pricing of Public Utility Services*. PhD thesis, MIT, 1982. also available as MIT Energy Lab Technical Report MIT-EL 82-031.
- [2] Roger E. Bohn, Michael C. Caramanis, and Fred C. Schweppe. Optimal pricing in electrical networks over space and time. *Rand Journal of Economics*, 15(3):360–376, Autumn 1984. available at <http://ideas.repec.org/a/rje/randje/v15y1984iautump360-376.html>.
- [3] Per Carlsson. *Algorithms for Electronic Power Markets*. PhD thesis, Uppsala University, Sweden, 2004.
- [4] ETSO and EuroPEX. Flow-based market coupling; a joint ETSO–EuroPEX proposal for cross-border congestion management and integration of electricity markets in Europe. Interim report, September 2004.
- [5] W. Hogan. Getting the prices right in Pennsylvania–New Jersey–Maryland (PJM). *Electricity Journal*, 11:61–67, August/September 1998.
- [6] Koen Kok, Cor Warmer, and René Kamphuis. PowerMatcher: multiagent control in the electricity infrastructure. In *AAMAS ’05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, volume industry track, pages 75–82, New York, NY, USA, 2005. ACM Press.
- [7] Fred C. Schweppe, Michael C. Caramanis, Richard D. Tabors, and Roger E. Bohn. *Spot Pricing of Electricity*. Kluwer Academic Publishers, 1988.
- [8] William Vickrey. Responsive pricing of public utility services. *Bell Journal of Economics and Management Science*, 2(1):337–346, Spring 1971.
- [9] William Vickrey. Responsive pricing and homeostatic control. mimeo, 1979.
- [10] Michael P. Wellman. A market-oriented programming environment and its application to distributed multicommodity flow problems. *Journal of Artificial Intelligence Research*, 1:1–23, 1993.
- [11] Fredrik Ygge and Hans Akkermans. Power load management as a computational market. In Victor Lesser, editor, *Proceedings of the First International Conference on Multi-Agent Systems*. MIT Press, 1996.
- [12] Fredrik Ygge and Hans Akkermans. Resource-oriented multicommodity market algorithms. *Autonomous Agents and Multi-Agent Systems*, 3(1):53–71, 2000.

Bio-Inspired Evacuation Routing Support

Anthonie van Lieburg

Martijn Neef

TNO Defence, Safety and Security, The Hague, The Netherlands

Abstract

This paper describes a decentralized and adaptive evacuation routing system. This system consists of a network of communicating sensors and actuators that dynamically guide evacuees towards the nearest exit. The approach makes use of self-organizing gradient following mechanisms that are inspired by biological pattern formation processes.

1 Introduction

This paper describes the use of biologically inspired algorithms in the application of an adaptive evacuation routing system. We study these systems, and their associated phenomena like self-organization, because of their specific characteristics like fault-tolerance, self-configuration, and scalability, which are desirable for systems that are designed to function in dynamic and unpredictable environments. Because many biological systems face similar challenges, a bio-inspired approach may help us find novel approaches to adaptivity, coordination and autonomy. In this study, we discuss an adaptive evacuation routing system that guides evacuees to safe havens using a simple biologically inspired, self-organizing mechanism.

2 Bio-Inspired Algorithms and Self-Organization

For ages, nature has been a valuable source for technological improvements [4]. With the advent of powerful computing devices, many biological models have found their way into algorithm design, providing novel approaches to computational problems. A classic example thereof is the development of the genetic algorithm [11], but many other examples exist, such artificial immune systems for computer security and data mining [13], ant colony optimization algorithms for path planning and telecommunication routing [7, 8, 9, 21], the use of dominance hierarchies in wasp colonies for task allocation [5, 8] and multi-robot coordination [15, 18].

A particular interesting phenomenon often associated with biological pattern formation is self-organization. Self-organization is an often used term in computer science nowadays. Self-organization is about emergent global system behaviour that stems from local interactions between the different entities that form the whole system, without explicit representation of these global patterns on the level of the individual components, and without any centralized model of the environment. Ant colony behaviour is an illustrative example of self-organization [3]. There is no central controller telling the ants where to go, or what to do. Moreover, none of the ants has a notion of the state of other ants, and their interaction is indirect, by means of the environment. The ant colony has an innate capability to organize itself, apparently without being capable of maintaining any internal state, or any other cognitive capabilities.

In a broader context, self-organization is a fundamental principle in many natural systems, and is the prime subject of complexity theory research, which studies the emergence of system properties and system structure in complex systems that are made of several critically connected parts. These systems can spontaneously evolve from one state to another, under changing circumstances. Self-organization is equivalent to a system traversing from one state to another. If we perturb a complex self-organizing system (for instance by changing its structure, or its surroundings), then the system as a whole will tend to find new stable states, and move away from unstable states along certain trajectories. Bak [3], Lewin [16], Kauffman [14] among others provide general overviews to the domain of self-organization and

complexity theory. Parunak and Bruecker provide some clear insight of engineering self-organizing systems [19, 20]. We use these types of phenomena and theoretic notions in the design of our system.

3 Adaptive Evacuation Routing

Efficient evacuation is a critical aspect of large-scale incidents. The main goal of evacuation is to guide people away from an affected region into safe surroundings, and to minimize the time and risk it takes to get there. Despite the fact that most buildings have predefined evacuation plans and procedures, the evacuation itself is very much a dynamic process. Determining of the optimal evacuation route is a complicated matter due to the amount of dynamic parameters involved. The best route for anyone involved greatly depends on the specifics of the incident, the actual state of the building, the condition and location of the evacuees and so forth. For instance, standard evacuation routes may be blocked by smoke or fires, and corridors may be too small to carry the number of evacuees.

In this research, we propose a fully distributed and adaptive approach to evacuation routing support, thus reflecting the dynamic nature of the process. We aim to create an adaptive evacuation routing system that guides evacuees to safety using dynamic signs and alarms. To achieve this behaviour, we envision a network of low-cost smart devices that control sensors (e.g. smoke sensors) and actuators (e.g. direction signs or sound devices). Figure 1 gives an illustration of such an evacuation routing system.

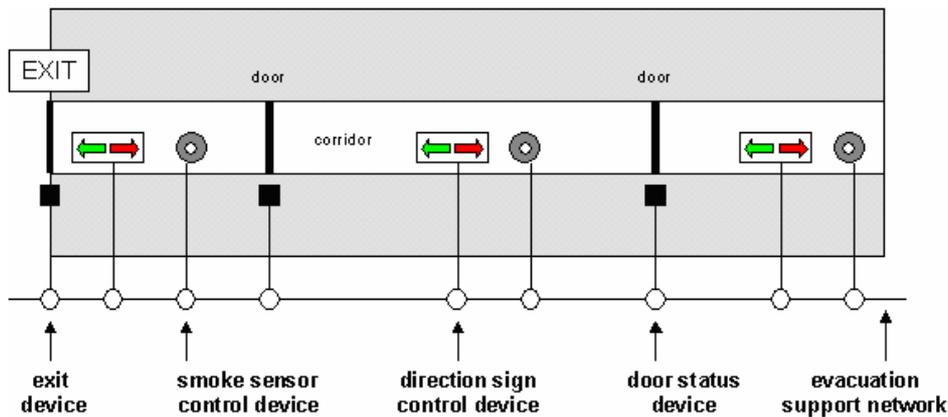


Figure 1: Illustration of the evacuation support concept. Various sensors and actuators are controlled by networked devices (the connected ovals at the bottom) to jointly provide evacuation support.

In this example, each section contains three types of devices: smoke detectors, direction sign and door status monitors. These devices are controlled by 'smart' network entities that effectively represent the devices. Note that the network layout mimics the actual corridor layout and thus gives us implicit knowledge about the layout of the environment, which we utilize in our routing optimization procedure. These devices are capable of local communication and together decide on the safest path to nearest emergency exits. The dynamic exit indicators inform evacuees about safe directions, while smoke sensors determine dangerous areas and provide the trigger for starting evacuation procedures.

3.1 The Adaptive Routing Algorithm

There are several challenges in developing such an algorithm. First, it should be able to find a safe evacuation path from any location in the network, and be able to consider obstructions. Secondly, the algorithm must fit the limited computational and communicative capabilities of the devices used to carry the algorithm. Finally, the algorithm must be distributed and robust. In a disastrous environment, there are no guarantees that data networks survive, so the algorithm must take full advantage of being distributed, and must have a high degree of robustness against structural damage.

We have applied a bio-inspired gradient approach as can be found in pattern formation mechanisms of many biological systems [10]. The use of chemical gradients is a widely used mechanism to differentiate within an initially uniform medium (like for example a growing embryo). The basic idea is that by creating gradients, the differences in (relative) concentration at different locations provide a clear distinctive marker. Using these markers, a cell is able to determine its required differentiation at that location, similar to Akam's model of striping formation in *Drosophila* flies [1].

This mechanism can be used to determine the relative position of devices. By starting the gradients at the exits, all other connected devices can determine the relative position towards the nearest exit by inspecting the local level of the gradient, assuming the diffusion of the signal gradient through the network has been possible. In this way, the network becomes a kind of potential field [12], in which at any location the nearest exit can be found by following the gradient. This simple approach is in essence enough to let direction signs point towards the closest exits, even if sections have become inaccessible due to hazardous conditions. The textbox below displays the algorithm per node in pseudo-code.

```

Function UpdateGradients()
  Select case ThisDevice.Type
  Case ThisDevice.Type = "EXIT"
    GradientValue = SomeLargeConstant
    For all devices connected to ThisDevice
      NewValue = GradientValue - DistanceBetweenDevices
      Message = NewValue
      Send Message to connected device
    End for
  Case ThisDevice.Type = "DETECTOR" or "SIGN"
    If ReceivedGradientValue > ThisDeviceGradientValue
      GradientValue = ReceivedGradientValue
      GradientIdent = ReceivedGradientIdent
      If ThisDevice.Type = "SIGN"
        UpdateSignDirection(GradientIdent)
      End if
    Else
      If ReceivedGradientIdent = GradientIdent
        GradientValue = ReceivedGradientValue
        If ThisDevice.Type = "SIGN"
          UpdateSignDirection(GradientIdent)
        End if
      End if
    End if
    If GradientValue > 0 then
      If ThisDeviceState = "ALARM"
        NewValue = GradientValue / LargeConstantNumber
      Else
        NewValue = GradientValue - DistanceBetweenDevices
      End if
      For all devices connected to ThisDevice
        Send NewValue to connected devices
      End for
    End if
  End select
End function

Function ProcesIncomingMessages()
  Message = OldestMessageInFIFOBuffer
  If MessageDestinationIdent = ThisDeviceIdent
    ReceivedGradientValue = MessageGradientValue
    ReceivedGradientIdent = MessageSenderIdent
  End if
  RemoveMessageFromFIFOBuffer
End function

Function MainLoop
  Forever loop
    Updategradients
    ProcesIncomingMessages
  End loop
End Function

```

Textbox 1: Pseudo-code for computing the potential field at a single network node. Note that code for message handling and the management of connections are not shown.

For instance, if smoke sensors detect smoke, their networked counterparts may block received signals, and effectively prevent hazardous sections to be included in the evacuation routing. Add other types of

sensors (such as camera devices or movement detection) and constraints that are even more intricate can be included in the evacuation routing process.

The use of gradients provides a simple and robust method to create and maintain positional information within a dynamic network. The algorithm is capable of adaptive and dynamic routing, is robust against structural disruption and can deal with new scenarios. An additional capability that can be added to the basic algorithm is that of capacity-based routing – routing that is based on the number of evacuees. We base this extension on an ant colony optimization algorithm [2, 6] in which motion detectors spawn ‘ant-like messages’ into the network. These messages follow the basic plan towards the nearest exits, leaving a virtual ‘pheromone’ trail along the path they travelled. Devices use the level of this trail to update their routing costs towards all neighbouring devices. After the update, the network re-computes the basic evacuation plan while incorporating the update routing costs. After a number of iterations, the network will settle in a quasi-stable state, and thus provide a dynamic evacuation route that is suitable for the density at that moment. Besides using motion detectors to update routing costs, other constraints could be included as well. Since the connection between nodes is the primary control of the gradient spreading process, other aspects such as the orientation of nodes could be used to steer the diffusion process towards better evacuation routing plans.

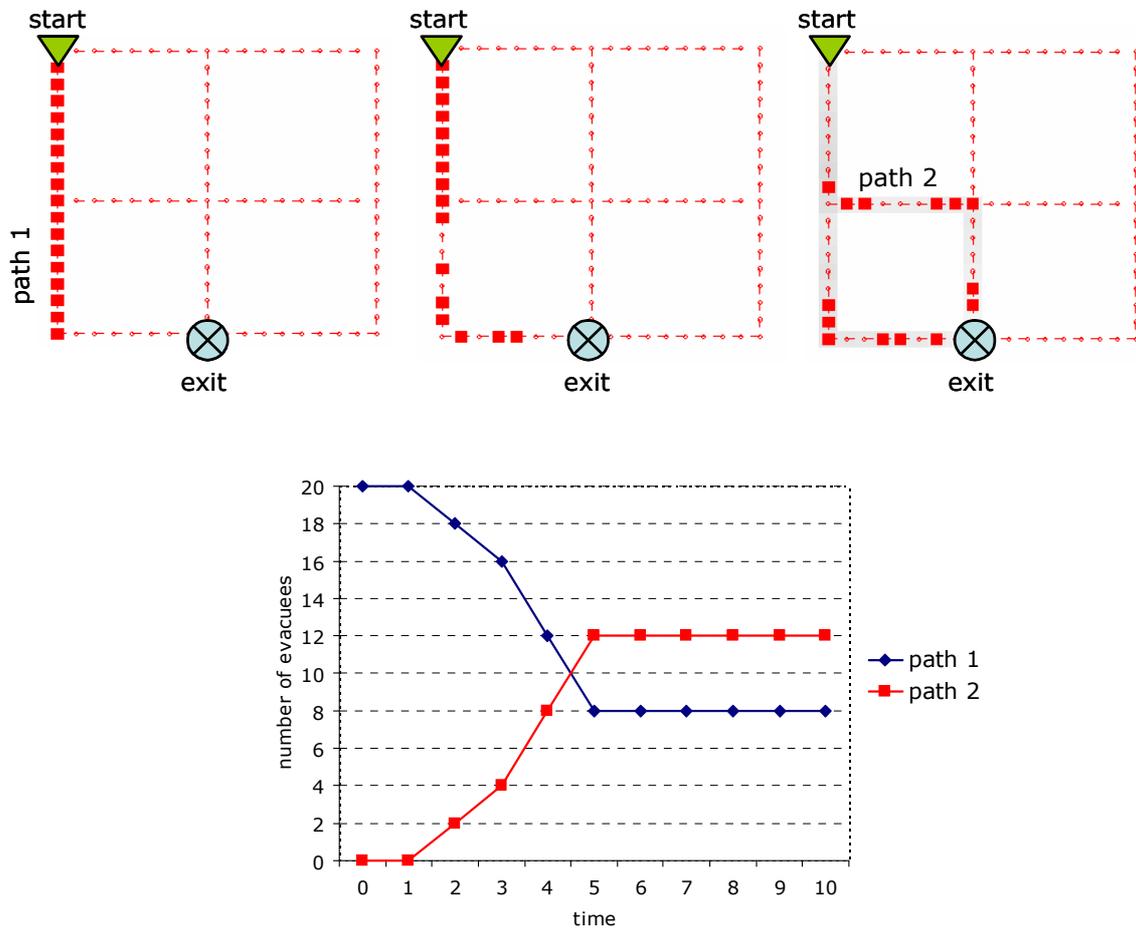


Figure 2: Graphs illustrating route adaptation due to capacity effects. Top figures show start positions (left) of evacuees (dotted lines), initial route (middle, Path 1 in graph) and adapted route plan (right, Path 2 in graph). Note the switch in path usage due to capacity effects. Since the path usage is determined at a single intersection, the switch is rather fast.

3.2 Implementation Considerations

The evacuation algorithm was tested in a simulation environment, and is currently being implemented in a hardware prototype. This prototype will include a network of small communicating processing units, interfaced to exit indicators and smoke detectors. These battery-powered devices include a microprocessor with (ad-hoc) radio communication and interface capabilities. Experiments in the software environment have revealed various implementation considerations, which will have to be dealt with in practical prototypes.



Figure 3: Screenshot of the software validation environment. The figure shows the evacuation network in a building with four exits (squares). The arrows point towards the closest exit.

One of the biggest challenges is acquiring positional information. The algorithm assumes that each device knows their relative position to their direct neighbours. In the test environment, line-of sight has been used as an approximation. Since the hardware devices only communicate by radio, retrieving positional information is difficult and would require additional sensors. As an intermediate solution, a human operator could provide the positioning of and linkage between devices.

Asynchronous execution or intermittent communication might cause disturbances. Each device will have its own processor clock and although differences might be small, synchronous activity cannot be guaranteed. Effects of asynchronous update schemes have been studied using the software environment, and the evacuation algorithm deals well with asynchrony. Practical tests will need to confirm this, and provide cues on how to resolve lost communication because of asynchronous or flawed communication.

Although devices are uniform, depending on their interface they may have different roles in the system. To simplify the hardware implementation it is assumed that an operator will assign these roles to installed devices, but there might be ways to automate these chores, or facilitate self-organization.

4 Discussion

This paper described the use of biologically inspired algorithms to create a self-organizing, adaptive distributed application. Although the followed approach is fairly simple, it provides us with hands-on experience in actually implementing a real-life application drawn from these principles. We realize that routing can be solved in many other established ways, but because to the inherent dynamic nature of the evacuation application case and the adaptive potential of biological systems, we chose to experiment with bio-inspired approaches. We have found that investigating biological analogies for system behaviour offers a wealth of interrelated mechanisms suitable for many aspects of system development. Although the proposed gradient approach is quite simple and perhaps not even exclusively bio-inspired, it can be used as a basis for more complex mechanisms like reaction-diffusion systems and associated morphological pattern formation processes. A growing number of researchers are following the same line of thought for various challenges, such as distributed tracking in a sensor networks [17] and robotic path

planning [12]. Altogether, using bio-inspired distributed methods seems viable and worthwhile investigating further. This paper stems from own ongoing research in this direction, and through such a step-by-step learning process we hope to gain more insight and know how in the application and construction of artificial complex adaptive systems in a variety of domains.

5 Acknowledgements

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. The ICIS project is hosted by the DECIS Lab, the open research partnership of Thales Nederland, the Delft University of Technology, the University of Amsterdam and the Netherlands Organization for Applied Scientific Research TNO.

References

- [1] Akam, M. (1989), *Drosophila* development: Making stripes inelegantly. *Nature* 341, pp. 282-283.
- [2] Appleby, S. and Steward, S. (1994), Mobile software agents for control in telecommunication networks. *British Telecom Technology Journal*, 12, pp. 104-113.
- [3] Bak, P. and Chen, K. (1991), *Self-Organized Criticality*. *Scientific American*, January 1991, pp. 46-53.
- [4] Benyus, J.M. (1998), *Biomimicry: Innovation Inspired by Nature*. Perennial, New York, USA.
- [5] Bonabeau E., Sobkowski A., Theraulaz G., Deneubourg J.L (1997), Adaptive Task Allocation Inspired by a Model of Division of Labour in Social Insects. *Bio-Computation and Emergent Computing*. World Scientific, pp. 36-45.
- [6] Bonabeau B., Henaux, F., Guerin, S., Snyers, D., Kuntz, and Theraulaz, G. (1998), Routing in Telecommunications Networks With 'Smart' Ant-like Agents. In Albayrak, S. and Garijo, F.J. (Eds.), *Intelligent Agents for Telecommunication Applications*, Springer Verlag, pp. 60-71.
- [7] DiCaro G. and Dorigo M. (1998), AntNet: Distributed Stigmergetic Control for Communications Networks. *Journal of Artificial Intelligence Research*, vol. 9, pp. 317-365.
- [8] Dorigo M. and Stützle T. (2004), *Ant Colony Optimization*. MIT Press, Cambridge, MA, USA.
- [9] Dorigo M., Maniezzo, V. and Coloni, A. (1996), Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics*, 26(1), pp. 29-41.
- [10] Driever, W. and Nüsslein-Volhard, C. (1988), A gradient of bicoid protein in *Drosophila* embryos. *Cell*, vol. 54, pp. 83-93.
- [11] Holland J. (1975), *Adaptation in natural and artificial systems*. University of Michigan Press, Ann Arbor, MI, USA.
- [12] Hwang, Y.K., and Ahuja, N. (1992), A Potential Field Approach to Path Planning. *IEEE Transactions on Robotics and Automation*, 8(1), pp. 23-32.
- [13] Farmer, J.D., Packard N.H. and Perelson A.S. (1986), The immune system, adaptation and machine learning. *Physica D*, vol. 22, pp. 187—204.
- [14] Kauffman, S. (1993), *The Origins of Order*. Oxford University Press, Oxford, UK.
- [15] Kube, C.R., and Zhang, H. (1993), Collective Robotics: From Social Insects to Robots. *Adaptive Behaviour*, 2(2), pp. 189-219.

- [16] Lewin, R. (1992), *Complexity: Life at the Edge of Chaos*. Macmillan, New York, USA.
- [17] Li, Q., Peterson, R., DeRosa, M. and Rus, D. (2003), Reactive Behavior in Self-reconfiguring Sensor Networks. *ACM Mobile Computing and Communications Review*, 7(1), pp. 56-58.
- [18] Pagello, E., Montsello, F. Dangelo, A., Garello F., and Ferrari, C. (1999), Cooperative Behaviors in Multi-robot Systems through Implicit Communication. *Robotics and Autonomous Systems*, 29, pp. 65-77.
- [19] Parunak, H.V.D., and Brueckner, S. (2001), Entropy and Self-Organization in Multi-Agent Systems. In *Proceedings of The Fifth International Conference on Autonomous Agents (Agents 2001)*, Montreal, Canada, ACM, pp. 124-130.
- [20] Parunak, H.V.D., and Brueckner, S. (2004), Engineering Swarming Systems. In F. Bergenti, M.-P. Gleizes, and F. Zambonelli (Eds.), *Methodologies and Software Engineering for Agent Systems*. Kluwer, pp. 341-376.
- [21] Schoonderwoerd, R., Holland, O., Bruten, J., and Rothkrantz, L. (1997), Ant-based Load Balancing in Telecommunications Networks. *Adaptive Behavior*, 5(2), pp. 169-207.

Applying Data Mining for Early Warning in Food Supply Networks

Yuan Li^a Mark R. Kramer^a Adrie J.M. Beulens^a Jack G.A.J. van der Vorst^b

^a Information Technology Group
^b Operations Research and Logistics Group
Wageningen University, The Netherlands

Abstract

In food supply networks, quality of end products is a critical issue. The quality of food products depends in a complex way on many factors. In order to effectively control food quality, our research aims at implementing early warning and proactive control systems in food supply networks. To exploit the large amounts of operational data collected throughout such a network, we employ data mining in various settings. This paper investigates the requirements on data mining posed by early warning in food supply networks, and maps those requirements to available data mining methods.

1. Introduction

Food quality problems in food supply networks form a critical issue for both consumers and food companies. However, in recent years, food quality crises occurred frequently all over the world. A recent case is dioxin contamination in pork in Belgium, the Netherlands, and Germany. In order to effectively control food quality, we need early warning systems to predict potential problems and give suggestions for proactive control.

The primary source of information on food supply networks is expert knowledge. However, expert knowledge is not always sufficient to deal with new quality problems in a direct way. This is partly due to the complexity of food supply networks. Further, food products and food processing procedures show inherent uncertainty and variability. Recent developments in information systems of food supply networks provide us with possibilities to discover valuable information about quality problems from recorded data. We deal with these problems with the help of a powerful quantitative method – data mining.

Data mining has been successfully applied in many areas, such as biology, finance, and marketing. However, the uptake of this technique in food supply networks has not matched the amount of applications in business. Historically food supply networks were less automated than other businesses. However, in recent years, food industry began to build information systems to collect data about various stages of food supply networks. These information systems provide us with opportunities to employ data mining techniques to discover interesting relations for food quality problems. Because of inherent properties of food products, it is not straightforward to adopt applications from other areas to the field of food supply networks.

In our research, we are aiming at employing data mining techniques to construct early warning systems in food supply networks. Such an early warning system will adaptively identify new problems in food quality, aid domain experts in discovering possible causes for these problems, and monitor those causal factors to predict potential food quality problems. We anticipate taking even a step further towards proactive control to provide measures to prevent or correct food quality problems.

2. Food Supply Networks

According to Santoso et al. [17], a supply chain is a network of suppliers, manufacturing plants, warehouses, and distribution channels organized to acquire raw materials, convert these raw

materials to finished products, and distribute these products to customers. Van der Vorst et al. [21] extend this to a food supply network, referring to an interconnected system with a large variety of complex relationships such as alliances, horizontal and vertical cooperation, forward and backward integration in supply chains.

Figure 1 depicts a supply chain network with accompanying monitoring and control systems. Processes can be seen at different levels, depending on the kind of problems considered. They may be as large as complete farms, factories or warehouses, or as small as one individual activity. The monitoring system gathers performance data of processes and their inputs and outputs. The control system can influence settings of the processes involved. Together, the monitoring system and control system manage the whole series of processes.

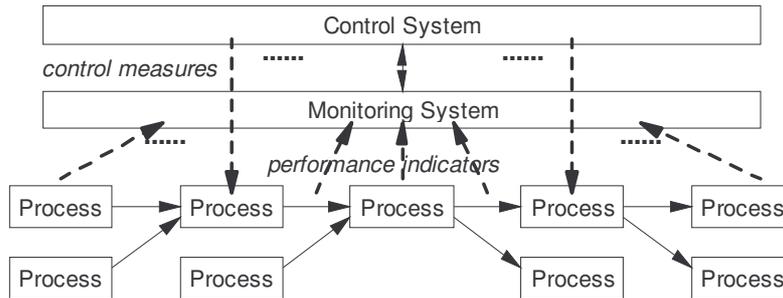


Fig. 1. An example supply network

Food supply networks have specific characteristics compared to other supply networks [21]. A food supply network includes multiple stages, global sourcing, variety of sources, leading to a complex network structure. Moreover, food is an inhomogeneous material, with a wide uncertainty and variability in quality and quantity of supply, as well as yield of production processes.

Current research on performance measurement systems in food supply networks provides ways to quantify quality attributes of food products [20]. Each factor influencing a quality attribute has a certain effect. For instance, temperature is an important factor for many quality attributes of food products. In order to model the effect of those factors, we need quantitative models, especially when variability and uncertainty characterize the food quality attributes.

Next to uncertainty and variability, theoretical understanding of food quality problems is scarce. As a result, one has to rely on other sources of knowledge. Current advances in information systems in food supply networks provide us with another way to deal with this kind of problems: to induce knowledge from data.

2.1. Information Systems in Food Supply Networks

In food supply networks, the use of information systems has increased over the last decades, on the level of the network as well as on the level of participating companies. One reason is that the need for transparency in food supply networks has increased [7]. Transparency implies that extensive information associated with food products should be recorded along the supply network, e.g. regarding production data, product identification, and product and process properties. Several modern logistical and management trends such as E-commerce, scanning, total quality management, and HACCP also generate bulk data.

Due to these advances of information systems in food supply networks, large amounts of data about food production and processing are recorded every day. Apart from their original purpose, the information implicitly present in these data is valuable as a basis for implementing early warning and proactive control. First, abnormal change in these data will give indications for potential problems. Second, many unknown causal relations may exist between recorded data about inputs, controls and production means, and operational performance of processes inside food supply networks. Knowledge about these relations provides a possibility to prevent

problems by monitoring and proactively influencing the corresponding determinant factors. So it is worthwhile to employ data mining methods to adaptively identify new problems and discover causal relations from recorded data.

3. Framework for Early Warning Systems

Early warning systems are well known in natural sciences. These systems, based on historical monitoring, local observation, or computer modeling, predict natural disasters, such as floods [6] or earthquakes [23], and help to prevent or reduce their impact. In food technology, Costello et al. [1] presented a prototype sensor system for the early detection of microbially linked spoilage in stored wheat grain.

The early warning systems we intend to build should not only predict potential food quality problems, but also help to identify relations between determinant factors and quality attributes of food products. Ultimately, the knowledge about these relations and the decision varieties associated with these factors will enable proactive control to prevent those problems.

To achieve our objectives of early warning, we designed a framework for early warning systems in food supply networks (see Figure 2). The main distinguishing attribute of our approach is the aim for adaptivity. Other early warning systems are designed only for predicting specific predefined problems. Due to uncertainty in food supply networks, we do not have predefined knowledge about all kinds of problems that could occur.

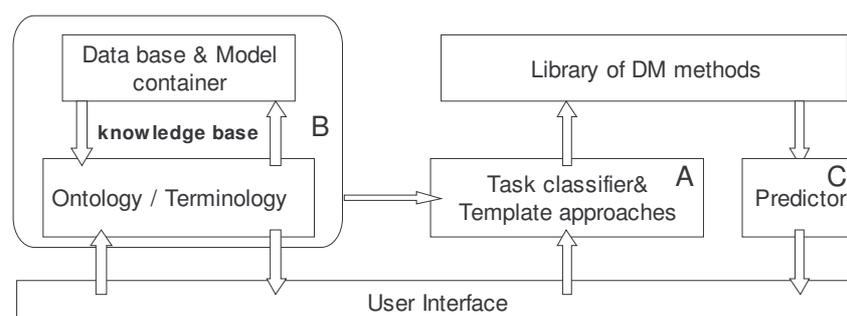


Fig. 2. Framework for early warning system in food supply networks (letters A, B, C for reference in section 4)

In our framework, users with new problems follow a template approach, which will guide them to find causes of the problem, select appropriate data mining methods, and instantiate a new predictor. The knowledge base provides valuable references for all these steps. This knowledge base will be continuously extended with new cases and domain knowledge from experts. So it will be helpful not only for us to construct early warning systems, but also for other stakeholders to deal with similar problems.

4. Requirements on Data Mining

4.1. Requirements Imposed by Early Warning in Food Supply Networks

An effective early warning system helps to identify potential problems, and to analyze their characteristics and causes. An early warning system according to the framework of Fig.2 operates in an iterative way. After problem detection, it is necessary to explore determinant factors for the problem detected. Then we try to describe the structure and form of relations between factors and performance measures in an appropriate way. After this, we can update the model in our knowledge base with obtained new knowledge. Of course, domain experts may also make their contribution to the model. With the updated model, we could again monitor the determinant factors in food production and predict potential problems.

In order to realize early warning in food supply networks, we defined the following essential functional requirements. These requirements were informally obtained by generalizing existing early warning systems discussed above and by considering the characteristics of food supply networks.

1. *Prediction*: The primary purpose of early warning is to forecast potential problems as early as possible. So early warning systems should be able to describe future states of performance measures based on monitoring of historical and current determinant factors. (C in fig. 2)
2. *Problem detection*: In operational use, early warning systems should be able to detect new problems in food quality when they occur, and describe the characteristics of those problems. (A in fig. 2)
3. *Finding determinant factors*: After identifying a problem (either automatically or by expert input), it is necessary to investigate what are determinant factors for this problem, and how those determinant factors influence the performance measures involved. (A+B in fig. 2)

The first three requirements above relate to processes, whereas the next three relate to representation.

4. *Complex structure representation*: Due to the complexity of food supply networks, long chains of causal relations between factors may exist. Moreover these chains may branch and join at certain points. As a result, our system has to be able to represent those complex causal relations. (B in fig. 2)
5. *Different representation forms*: Many different kinds of relations between factors may exist. Relations between food quality problems and determinant factors may appear in different forms. Sometimes we find quantitative mathematical formulas directly relating one or more determinant factors to a performance measure. In other instances the relation takes the form of a conditional probability distribution. For some cases we do not have enough detailed knowledge and we can only give a model as a black box. So early warning systems require the possibility to show different representation forms. (B in fig. 2)
6. *New knowledge incorporation*: Our early warning systems should be capable of incorporating new knowledge when it is discovered, either from newly detected problems or by expert input (e.g. due to the continuous development in food supply networks). (B+C in fig. 2)

It is not necessary that one technique satisfies all requirements at once, because different techniques could be used for different steps in our framework.

4.2. Contribution of Data Mining to Fulfill the Requirements

In this section we look at how to employ data mining to fulfill the requirements discussed above. Our focus is mainly on the functional aspects, which means we look at what are the functions of data mining, and how these functions can be used to fulfill the requirements of early warning systems. Definitions of data mining functions are gathered from Fayyad [2] and Freitas [5]. Below, we provide tables that present suitable data mining methods for each function, to facilitate selecting appropriate methods for specific tasks.

Prediction. Prediction of food quality problem based on historical and current data of determinant factors is one of the principle requirements imposed by early warning. And fortunately one of the merits of data mining techniques is the prediction power. There are many success stories on application of data mining methods for prediction of food quality in food supply networks [16, 19].

Problem Detection. Signals that trigger early warning usually appear as anomalies in monitored performance measures first: deviations from the established norm and expected behavior. Deviation detection from various types of data is a prominent function of data mining [5]. Literature shows the application of several data mining techniques to identify problems in food production. One example is the use of neural networks in an X-ray system to identify contaminants in packaged food products [13].

Finding Determinant Factors. In order to identify determinant factors for food quality problems, ultimately the knowledge of domain experts is indispensable. However, data mining

may help to quickly find candidate factors. Applicable data mining functions include causation modeling, factor selection, and to some extent classification and regression.

Causation models describe the causal relations between determinant factors and performance measures. Currently there are two methods available: Bayesian causal discovery [14] and constraint based causal discovery [18]. However, both methods are only at an experimental stage. The number of successful applications of these two techniques is limited. See Freedman [4] for causes of this problem.

Besides causation modeling, various factor selection methods for data mining purposes could help to quickly find a set of potentially relevant factors to focus on, and determine the relative importance of each factor. Note however, that factor selection is not regarded as a data mining function, but rather as a preprocessing step. There are many methods for selecting relevant factors [12]. In general, factor selection methods can either use a filter approach or a wrapper approach. A filter approach selects relevant factors independently of the data mining techniques used for prediction. But a wrapper approach can only be used in combination with a specific data mining technique [10].

Some data mining methods for classification and regression can also help to find relevant factors. For example, neural networks can be used for classification. With sensitivity analysis on neural networks [24], we can find which factors are influencing the outcomes of the neural network.

No matter what techniques are selected for causal relation discovery, the result should be checked by domain experts before being applied in practice. Only with the interaction between domain experts and data mining we can find valuable relations.

Complex Structure Representation. In order to represent interactions that span multiple stages of a food supply network, we need methods that are able to describe relations between factors from various stages. For this purpose, we may apply dependency modeling and causation modeling.

Dependency modeling describes significant dependencies among variables [3]. It is different from causation modeling in that causal relations are intuitively stronger than dependencies. The latter only indicates a correlation between determinant factors and performance measures, but there is no causality semantics in this relationship. Causation models not only show correlations, but also indicate that those determinant factors actually cause the observed effects.

One of the candidate methods for dependency modeling is Bayesian networks. This method builds a graphical network to describe the complex structure. Variables are represented as nodes, and dependencies between variables are represented as links. Variability in variables is described by conditional probability distributions, specifying the probability for each variable given the values of the ones linked to it.

There are many ways to combine the requirements imposed, functions of data mining, data mining methods, and some other aspects. Listing all possible combinations would yield a multi-dimensional table of all aspects. Here, we present the most important combinations of two dimensions in Table 1 and Table 2. The requirements of describing different kinds of relations and novel relation incorporation will be dealt with in following subsections.

In Table 1, we summarize the use of data mining for different functional requirements as reported in literature. In Table 2, we compare some commonly used data mining methods against the data mining functions mentioned above. We will use such tables for technique selection. For each non-empty cell we did find applications in literature. We use ‘valid’ to represent the fact that corresponding applications have been reported in literature (e.g. [13] shows the application of classification methods for deviation detection). Due to space limitations, we can not give all references here. Cells stating ‘helpful’ indicate that the technique may yield candidate relations but does not derive determinant factors by itself. See also the discussion above.

Table 1. Functions of DM vs. requirements from early warning system

Function of data mining	Requirements imposed by early warning system			
	1.Predict	2.Detect problem	3.Find determinant factors	4.Describe complex structure
Deviation detection		Valid		
Factor selection *			Helpful	
Classification	Valid		Helpful	
Regression	Valid		Helpful	
Dependence model	Valid		Helpful	Valid
Causal model	Valid		Valid	Valid

* Factor selection methods are usually regarded as a pre-processing step for data mining rather than a separate data mining function.

Table 2. Applicability of DM methods for specific functions

DM Function	DM methods				
	Decision tree	Neural network	Bayesian network	Association rule	Nearest neighbors
Deviation detection	Valid	Valid	Valid	Valid	Valid
Classification	Valid	Valid	Valid		Valid
Regression	Valid	Valid	Valid		Valid
Dependence model			Valid	Valid	
Causal model			Valid		

Different Representation Forms. Table 3 shows relevant representation forms and corresponding data mining methods.

Selection of appropriate representation forms depends on the purpose of representation, the data characteristics, and the knowledge on the relation. For example, if the purpose is to represent causal relations to users of an early warning system, then we should choose more understandable representation forms, such as decision tree or association rule.

New Knowledge Incorporation. As discussed above, there are two kinds of novel relations that need to be incorporated: relations discovered from new problems, and relations obtained from domain experts. The requirement to incorporate such new relations can be fulfilled easily by some of the data mining methods, such as association rules and Bayesian networks. For example, in a case study to predict DOA (Death On Arrival) in a chicken supply network, we obtained knowledge that DOA will increase with transportation density. It is easy to add this knowledge as an association rule to a rule set, or as a node and an arc to a Bayesian network. Table 3 also provides an overview of capabilities of different methods for incorporating new knowledge into models constructed with these methods.

Table 3. Representation forms of DM methods and extensibility of corresponding models

Data mining method	Representation form [3]	Novel knowledge incorporation
Decision trees	Decision trees	Easy
Association rules	Rules	Easy
Neural networks	Linear or Nonlinear model	Difficult
Nearest neighbors	Example-base methods	Difficult
Bayesian networks	Probabilistic graphical dependency model	Easy

4.3. Discussion on Technique Selection

There are many aspects for technique selection. Function, representation form, and capability for novel knowledge incorporation are important aspects. The data format is another aspect. Various kinds of performance measures are available for food quality; some are quantitative, such as

body weight; some are qualitative, such as objective evaluation of meat color. The data format (nominal, numerical) has to be taken into consideration as well when selecting a technique.

However, the quality of a model also depends on how well the model class is able to represent patterns in the data set. Some research on automatic technique selection has already been conducted. Three main lines of interest have been found: heuristic expert rules [9], meta-learning [8], and landmarking [15]. Verdenius [22] has used data class boundary characteristics for selecting techniques. He distinguishes orthogonal and non-orthogonal (linear and nonlinear) class boundaries. Decision trees are especially suitable for orthogonal instead of non-orthogonal class boundaries, while neural networks are also good at non-orthogonal class boundaries.

5. Conclusion

This paper explains why and how data mining can be helpful in building up early warning systems in food supply networks. The construction and functioning of early warning systems will inevitably require the involvement of domain experts. However, with the help of data mining, we can complement their knowledge with new, interesting relations.

We investigated the functional requirements for data mining in food supply networks, and presented an overview of applicable data mining methods for those requirements. This overview forms a starting point for technique selection for specific applications of data mining. Ongoing research will refine the framework and requirements introduced in this paper.

As part of our research, we use a number of case studies to investigate the applicability of data mining methods. The first case study has been reported elsewhere [11]; the next one is in progress. In subsequent steps in our research, we will use the information gained by these case studies to build a knowledge base for early warning in food supply networks. Data mining technique selection will be one of the components of this knowledge base.

References

- [1] B. Costello, R. J. Ewen, H. Gunson, N. M. Ratcliffe, P. S. Sivanand and P. T. N. Spencer-Phillips, A prototype sensor system for the early detection of microbially linked spoilage in stored wheat grain, *Measurement Science & Technology*, 14 (2003), pp. 397-409.
- [2] U. M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, From Data Mining to Knowledge Discovery: An Overview, in U. M. Fayyad, G. Piatetsky-Shapiro, et al., eds., *Advances in Knowledge Discovery and Data Mining*, AAAI-Press, Menlo Park (Ca), 1996, pp. 1-37.
- [3] U. M. Fayyad, G. Piatetsky-Shapiro and P. Smyth, The KDD process for extracting useful knowledge from volumes of data, *Communications Of The ACM*, 39 (1996), pp. 27-34.
- [4] D. A. Freedman, Graphical models for causation, and the identification problem, *Evaluation Review*, 28 (2004), pp. 267-293.
- [5] A. A. Freitas, *Generic, Set-Oriented Primitives to Support Data-Parallel Knowledge Discovery in Relational Database Systems*, University of Essex, UK, 1997.
- [6] J. G. Grijzen, X. C. Snoeker and C. J. M. Vermeulen, An information system for flood early warning, 3rd international conference on floods and flood management, Delft Hydraulics Publication, Florence, Italy, 1992, pp. 27.
- [7] G. J. Hofstede, A. J. M. Beulens and L. Spaans-Dijkstra, Transparency: Perceptions, Practices and Promises, in T. Camps, P. Diederens, et al., eds., *The emerging World of Chains and Networks: Bridging Theory and Practice*, Reed Business Information, 2004.
- [8] A. Kalousis and M. Hilario, Model Selection via Meta-learning: a Comparative Study, *International Journal of Artificial Intelligence Tools*, 10 (2001).

- [9] Y. Kodratoff and V. Moustakis, *Managing Machine Learning Application Development and Organizational Implementation*, Tutorial notes, ECAI (1994).
- [10] R. Kohavi and G. H. John, *The Wrapper Approach*, in H. Liu and H. Motoda, eds., *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, 1998, pp. 33-50.
- [11] Y. Li, M. R. Kramer, A. J. M. Beulens and J. G. A. J. v. d. Vorst, *Using data mining to improve operations management in food supply networks*, in J. Bijman, S. W. F. Omta, et al., eds., *International agri-food chains and networks: management and organization*, Wageningen Academic Publishers, 2006, pp. 163-177.
- [12] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery and Data Mining*, Kluwer Academic Publishers, 1998.
- [13] D. Patel, E. R. Davies and I. Hannah, *Towards a breakthrough in the detection of contaminants in food products*, *Sensor Review*, 15 (1995), pp. 27-28.
- [14] J. Pearl, *Causality: Models, Reasoning, and Inference*, Cambridge University Press, 2000.
- [15] B. Pfahringer, H. Bensusan and C. Giraud-Carrier, *Meta-learning by landmarking various learning algorithms*, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000)*, Morgan Kaufmann, San Francisco, California, 2000, pp. 743--750.
- [16] J. Rousu, L. Flander, M. Suutarinen, K. Autio, P. Kontkanen and A. Rantanen, *Novel computational tools in bakery process data analysis: a comparative study*, *Journal Of Food Engineering*, 57 (2003), pp. 45-56.
- [17] T. Santoso, S. Ahmed, M. Goetschalckx and A. Shapiro, *A stochastic programming approach for supply chain network design under uncertainty*, *European Journal of Operational Research*, 167 (2005), pp. 96-115.
- [18] C. Silverstein, S. Brin, R. Motwani and J. Ullman, *Scalable techniques for mining causal structures*, *Data Mining And Knowledge Discovery*, 4 (2000), pp. 163-192.
- [19] B. Vallejo-Cordoba, G. E. Arteaga and S. Nakai, *Predicting milk shelf-life based on artificial neural networks and head-space gas chromatographic data*, *Journal of Food Science*, 60 (1995), pp. 885-888.
- [20] J. G. A. J. Van der Vorst, *Performance measurement in agrifood supply chain networks: an overview*, in C. Ondersteijn, ed., *Quantifying Supply Chain Performance*, Kluwer, Dordrecht, 2005.
- [21] J. G. A. J. Van der Vorst, A. J. M. Beulens and P. van Beek, *Innovations in logistics and ICT in food supply chain networks*, in W. M. F. Jongen and M. T. G. Meulenberg, eds., *Innovation in Agri-food Systems - Product Quality and Consumer Acceptance*, Wageningen Academic Publishers, The Netherlands, 2005, pp. 245-292.
- [22] F. Verdenius, *Methodological Aspects of Designing Induction-based Applications*, Universiteit van Amsterdam, 2005, pp. 160.
- [23] Y. M. Wu, J. K. Chung, T. C. Shin, N. C. Hsiao, Y. B. Tsai, W. H. K. Lee and T. L. Teng, *Development of an integrated earthquake early warning system in Taiwan - Case for the Hualien area earthquakes*, *Terrestrial Atmospheric And Oceanic Sciences*, 10 (1999), pp. 719-736.
- [24] J. T. Yao, *Sensitivity analysis for data mining*, *Proceedings of the 22nd International Conference of the North American Fuzzy Information Processing Society NAFIPS'2003*, Chicago, Illinois, 2003, pp. 420-425.

Multi-Attribute Vickrey Auctions when Utility Functions are Unknown

Tamás Máhr ^{ab}

Mathijs M. de Weerd ^a

^a *Technical University Delft, Postbus 5031, 2600 GA Delft*

^b *Almende BV, Westerstraat 50, 3016 DJ Rotterdam*

Abstract

Multi-attribute auctions allow negotiations over multiple attributes besides price. For example in task allocation, service providers can define their service by means of multiple attributes, such as quality of service, deadlines, or delay penalties. Auction mechanisms assume that the players have evaluation functions over the space of attributes that assign a single value to any combination of attributes. This value (or cost) is directly comparable to price. We argue that in some situations, some of the attributes are difficult to convert to cost, e.g., in transportation it is often important which driver is going to deliver a given truckload. Such personal preferences of a customer are difficult to quantify.

To allow negotiations over such non-monetary attributes we relax the requirement of universally comparable utility functions, and give an incentive-compatible auction mechanism that uses only preference orders of the parties and not globally comparable function values. The suggested mechanism assumes that the bidders and the auctioneer have individual total orders over the space of possible contracts, but no utility functions. Each bidder places its bids using its own order, and the winner is chosen by the auctioneer's order. The actual attribute values are chosen based on the second-best bid. It is shown that this Vickrey intuition yields an incentive-compatible mechanism.

1 Introduction

Multi-attribute auctions extend traditional price-only negotiations to negotiations over price and other attributes. One example is procurement, when multiple suppliers compete over different quality attributes (e.g. color, or warranty) and price. In such an environment, decisions are based on scoring functions that express the values of different attribute combinations in money (cost functions), which can be simply compared to price to express utility. In some cases, however, it is not possible to express all attributes in terms of costs, even when it is possible to compare two choices. In a task allocation setting, the auctioneer may value a contract with an old customer more than one with an unknown one, but there is no natural way to express this in money. Our goal is to find an auction mechanism, where the bidders are best off by bidding the equivalent of their private value (incentive compatibility), and that can be used in case cost functions are unknown, or difficult to determine.

One of the most well-known incentive compatible mechanisms is the Vickrey auction [11]. Vickrey auctions are second-price sealed-bid auctions, where the optimal strategy is to bid one's true valuation. The original mechanism has been generalized in different ways. The Generalized Vickrey Auction [5, 10] is a mechanism to solve a rather complex combinatorial auctioning problem where multiple divisible goods are to be allocated to multiple bidders. Another generalization is when multiple attributes of a certain indivisible good are considered. For example Che [1] has introduced multi-dimensional auctions, where bids consist of values of multiple attributes, including price. In his approach every party has a scoring function that assigns a value to a combination of attributes expressing his own valuation (costs, preferences, etc.). Bidders place bids based on their own scoring function, and the auction mechanism selects the bidder with the highest score according to the auctioneer's scoring function. Che showed that his proposed mechanism, where the contracted values of the attributes (including price) are determined by the second-best combination of attribute-values is incentive compatible.

Following Che's footsteps, Parkes and Kalagnanam [6, 7] have proposed iterative mechanisms that are incentive compatible, efficient, and allow iterative revelation of private values.

While Che focused on buyer-optimality, Parkes et al. concentrated on market-efficient mechanisms. David et al. [2] studied the multi-attribute auction as well. They considered a first-score sealed-bid and four different English auction mechanisms, and analyzed the applicability of these mechanisms in different environments, providing optimal strategies for bidders and the auctioneer. Especially, they have suggested optimal scoring rules for the auctioneer, and have shown under which conditions it is optimal to share the truth with the bidders.

In some cases, however, it may be difficult to compute scores for different attribute values. The evaluation of scoring functions might include lengthy computations (e.g. predictions), or the quantification of certain attributes might not make much sense (e.g. color). One example in task allocation could be, when the negotiation includes the tool to be used to execute the task. It can make a difference for the auctioneer if tool A or tool B is used (as part of the quality of service), but he may not be able to assign a value to the different configurations. A practical example from transportation is that certain customers prefer certain truck drivers over others. They might be friends, or a long term trust relationship could already have developed between them, or simply because the driver speaks the language of the customer. In such cases it is very difficult for the parties to assign values to different combinations of attribute values, even though they clearly have a preference order over them.

Either when the auction mechanism optimizes buyer's utility, or market efficiency, it is commonly assumed that the sellers' evaluations are comparable, since they are always expressed as costs (for a good review on auctions in artificial intelligence research see [9]). We relax this assumption and require the existence only of a total order over the attribute-value combinations. This is different from the commonly used assumption in that it does not require the existence of a universally comparable value (cost). We show for a multi-attribute case that auctions can be conducted even if bidders and the auctioneer cannot assign values to different attribute combinations (Section 2). We describe a 'multi-attribute no-value' version of the Vickrey mechanism (Section 3), and prove that it is incentive compatible (Section 4). The main properties of the proposed mechanism are then discussed in Section 5, and a conclusion wraps our contribution up (Section 6).

2 Auctioning Without Values

Let us consider the auctioning of tasks. The result of such an auction is a contract on the execution of the auctioned task. A contract consists of n attributes (such as price, quality, etc.) of continuous or discrete values.

$$C(a_1, \dots, a_n)$$

Suppose, that both the m bidders and the auctioneer have a total ordering over the different combinations of attribute configurations: (A^n, \preceq_a) for the auctioneer and (A^n, \preceq_{b_j}) for bidder j , $\forall j = 1..m$. Note that in a traditional multi-dimensional auction setting everybody would assign a value to every combination of attribute values. This would implicitly define the ordering, but would also imply that it is possible to convert every combination of attributes to a utility for all bidders.

Having only a total order over the attribute values means (see Figure 1) that on presentation of two combination of attribute values every party can tell his preferred one, although he cannot assign a value to it. Certain combinations can be considered equal, belonging to an equivalence class.

Equivalence classes represent such combinations of attribute values that are equally good for the given player. If one combination of attributes is chosen then the players can still offer other combinations in the same equivalence group for the opposing players to choose from.

Definition 1. The *equivalence class* of (a_1, \dots, a_n) for agent j consists of all combinations of attribute values that are equivalent with (a_1, \dots, a_n) for agent j .

$$[(a_1, \dots, a_n)]_{\preceq_j} = \{(a'_1, \dots, a'_n) | (a'_1, \dots, a'_n) =_j (a_1, \dots, a_n)\}$$

Besides having an order over combinations of attribute values (A^n, \preceq) , players have separate orderings over most individual attribute values too $((A_1, \preceq_{A_1})$, where A_1 is the set of values attribute 1 can take on). They may prefer higher or lower prices, better or worse qualities, etc. It is reasonable to assume that a combination of preferred attributes is preferred over a combination of less preferred attributes (monotonicity).

$$\begin{aligned} a_1 \preceq_{A_1} a'_1 \wedge \dots \wedge a_n \preceq_{A_n} a'_n &\Rightarrow \\ (a_1, \dots, a_n) &\preceq (a'_1, \dots, a'_n) \end{aligned} \quad (1)$$

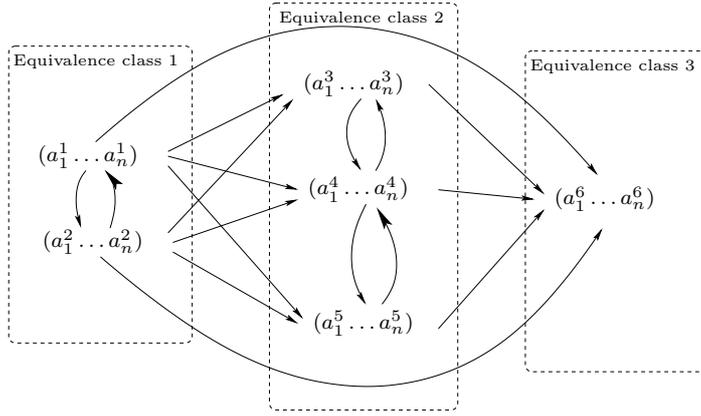


Figure 1: Total order over the combinations of n attributes

Another reasonable assumption to make is that the auctioneer's and the bidders' preference over the individual attributes are opposing. If the auctioneer tries to achieve a contract with low price and high quality, then the bidders will want high price and low quality. This implies that the bidders ordering over the individual attributes are the same (but the ordering of combinations is usually different for each bidder).

Bids consist of combinations of attributes ($bid_j = (a_1, \dots, a_n)_j$). Bidders have to determine every attribute before they can submit their bid. The chosen combination, or alternatively any other in the same equivalence group is their bid. The auctioneer collects all bids, and selects the most preferred one by the auctioneer. The bidder that submitted the most preferred bid is the winner.

For every bidder and every attribute there is a value, which is called the private value (a_{ij}^p is the private value of bidder j for attribute i). This value represents a threshold, below which the bidder does not want to win the auction. In case of price, the private value is the cost, in quality it is a basic quality level the bidder support anyway, etc. The combination of private values define a special equivalence class that we call the *private class*. The bidder is better off not winning the auction with a contract with attribute combination that is less preferred than his private class.

How to determine the values of the attributes for bidding depends on the auction mechanism. In an ascending auction bidders start from highly preferred values for every attribute and during consecutive bids, they move towards less preferred values following some bidding-strategy. They stop when the combination of attribute values belongs to the private class. In our proposed multi-attribute auction mechanism (see Section 3), the contract attributes are determined by the equivalence class of the second-best bid. This resembles the Vickrey mechanism, and indeed we will show (in Section 4) that the bidders best option is to place a bid from their private class.

3 The Mechanism

In the original Vickrey mechanism the bidders' optimal strategy is to bid their true valuation. When tasks are on auction and bids represent prices bidders ask to perform the tasks, the lowest bid is the winner, and the second-lowest bid determines the price. In such a situation agents' bid should be their cost. If their bid is higher, they reduce their chance of winning the auction but the price (determined by the second-lowest bid) remains the same. If their bid is lower than their cost then they have a higher chance of winning the auction, but the excess chance is exactly the chance that the price will not cover their cost.

Our mechanism is based on the recognition that the outcome of the auction does not depend on the values the auctioneer/bidders assign to attribute-value combinations. It only depends on how they order these combinations. Thus, it is not required to have an evaluation function over the attributes, but everybody has to have a total order over them.

The following steps are included in our mechanism, independent of the type of auction that is used.

1. Bidders $j \in \{1 \dots m\}$ select a combination of attribute values that they intend to bid (bid_j). Since it is equivalent to any other bid from the same equivalence class, we represent

their bid by the whole class.

$$[bid_j]_{\preceq_{b_j}} \quad (2)$$

2. A set of best bids (one for each agent j) is selected by choosing the most preferred bid by the auctioneer (using \preceq_a) in every equivalence class that was bidden.

$$\{\widehat{bid}_j | \widehat{bid}_j = \max_{\preceq_a} [bid_j]_{\preceq_{b_j}}, j = 1..m\} \quad (3)$$

3. The best of the bests is selected as winner.

$$bid_{win} = \max_{\preceq_a} (\widehat{bid}_1, \dots, \widehat{bid}_m) \quad (4)$$

4. The contracted attributes are determined by the second-best bid (bid_{2nd}). It is equivalent for the auctioneer to contract the exact attribute-values in bid_{2nd} , or to contract any other value combination that belong to the same equivalence class.

$$bid_{2nd} \in [bid_{2nd}]_{\preceq_a} \quad (5)$$

5. The best combination of attribute values regarding the winner's ordering in $[bid_{2nd}]_{\preceq_a}$ is selected as contract attributes.

$$C(a_1, \dots, a_n) = \max_{\preceq_{win}} [bid_{2nd}]_{\preceq_a} \quad (6)$$

The mechanism is backed by the same underlying thoughts as the single-issue Vickrey auction. The benefit of the winner bidder is defined by his competitors, which results in small benefit in a highly competitive market, and higher benefit, if the competition is not so tough.

The mechanism can be executed in four different ways regarding the private information of the parties. If the auctioneer's ordering is public, then the selection in step 2 can be executed by the bidders and they can submit their best bid regarding the auctioneer's ordering. If it is not public, then the bidders have to submit their whole equivalence class (Equation 2), and the auctioneer has to determine the best ones. Similarly, step 5 can be executed by the auctioneer, if the winner bidder shares his ordering with him. Otherwise, the auctioneer has to offer the whole equivalence class (eq_a) to the winner, who has to select the best combination of attribute-values himself.

The best performance in terms of communication is achieved, if no ordering is private. In this case, the amount of data to be transmitted is minimal, since the bidders can reduce their bids to one combination of attributes, and the auctioneer himself can determine the winner and the contract-values too. On the other hand, if all ordering information is private, whole sets of bids have to be communicated by the bidders, and the auctioneer also has to send his equivalence class to the winner to determine the contract-values. The minimal private information that has to be shared between the bidders and the auctioneer is the equivalence classes of the bids. The alternatives, where only one side has private information represent a trade-off between communication costs and information sharing.

4 Optimal Strategy

In a single-issue Vickrey auction bidders are best off bidding their 'true value'. In case tasks are on auction and bids represent prices bidders ask for performing the given task, the 'true value' to bid is the cost of performing the task. Bidders are better off by not winning the auction for a price that does not cover their costs.

In the proposed multi-attribute auctioning mechanism all combinations of attribute-values are ordered by the auctioneer (\preceq_a) and by the bidders ($\preceq_{b_j}, j = 1..m$) separately. Every bidder has a special value for every attribute, which is his private value regarding the given attribute (a_i^p is the private value of attribute i). In case the auction would only be about attribute i , values that are worse than a_i^p would not be acceptable for the bidder. In the multi-attribute case the equivalence class of bidder j that contains $a_i^p, i = 1..n$ is called the *private class* and any solution that is less preferred by \preceq_{b_j} is not acceptable for the bidder.

Definition 2. The *private class* of bidder j is the equivalence class that contains the attribute combination, where every attribute takes its private value.

$$(a_1^p, \dots, a_n^p)_j \in [bid_j^p]_{\preceq_{b_j}} \quad (7)$$

The private class contains minimal solutions for the bidder. He is better off not winning contracts with attribute-combinations that are less preferred than the private class. Note that single attribute values can be worse than the bidder's private value for that attribute, but the other attributes have to compensate for it.

Theorem 1. *In a multi-attribute second-price sealed-bid auction mechanism where the bidders' and the auctioneer's total ordering over the attribute values are diametrically opposed in the attribute space and are monotone in the individual attributes, the bidders are best off placing the bids in their private class.*

Proof. Bidders have two options other than bidding from their private class.

- Suppose a bidder places a bid that is preferred by him over bids in his private class:
 - Suppose he wins the auction. The actual attribute-values of the contract are determined by the second-best bid. That is the same as when he would have bidden his private class. Hence it is not beneficial to place a more preferred bid than those in the private class.
 - Suppose he loses the auction. There are two cases:
 - * If the winning bid is better for the auctioneer than any bid from the bidder's private class, he would have lost the auction anyway.
 - * If the winning bid is worse for the auctioneer than the best bid from his private class, then he could have won the auction by bidding truthfully.
- Suppose he places a bid that is less preferred by him than the ones in his private class:
 - Suppose he wins the auction. There are two cases:
 - * If the second-best bid is less preferred by the auctioneer than the best bid in his private class, then the contracted values will be the same as when he would have bidden his true value.
 - * If the second-best bid is more preferred by the auctioneer than the best bid in his private class, then he would have lost the auction by bidding truthfully. Though he wins the auction, the contracted combination of attribute-values are less preferred by him than the ones in his private class, which means that he would be better off by not winning the auction at all (see Equation 7).
 - Suppose he loses the auction. He would have lost the auction even if he had bid his private class.

In all possible outcomes, the bidder is not better off than by bidding from his private class. This proves that every bidder's best strategy to bid from their private class. \square

5 Mechanism properties

Mechanism design is the problem of how to design the (social) rules of an interaction such that a some objective function is maximized even if the players are intended to handle only in their own interest [8]. Jackson [3] describes the most important characteristics of such social choice functions. The desirable equilibrium properties of a multi-attribute auction include: efficiency, buyer-optimality, individual-rationality, and budget-balance.

The mechanism described in this paper offers a dominant strategy for the bidders, which is to place a bid from their private class. Let us suppose that the auctioneer also has a private equivalence class, and that he does not make a contract with less preferred (by \preceq_a) attribute values than this class. Then the mechanism has the following properties in equilibrium.

Efficiency: An efficient algorithm allocates a good to the one who values it most. Analogously, an efficient algorithm allocates a task to the one who executes it for the cheapest price. The ultimate goal is to maximize the difference between the buyer's and the seller's valuation. In our case the traditional definition of efficiency does not apply, due to the lack of universally comparable valuation functions. One cannot tell which bidder prefers the buyer most. Our multi-attribute mechanism allocates the task to the one whose offer has the best combination of attributes (for the auctioneer). Thus it implements a locally efficient allocation from the buyer's point of view.

Buyer-optimality: Buyer optimality in task allocation means that the auctioneer has to pay the least possible amount for the winner to execute his task. Just as the single issue Vickrey auction, our mechanism is not buyer-optimal. The winner bidder would execute the task even if a less preferred contract (from the winner's point of view) was made. Generally it is not possible for a mechanism to be efficient and buyer-optimal at the same time [4].

Individual-rationality: A mechanism is individually rational for the players if they are not better off not participating in the mechanism at all. In a traditional single or multi-attribute auction, individual rationality means that neither the auctioneer nor the bidders have negative utility. In our mechanism the auctioneer and the winner have to prefer the contract over their private classes for the mechanism to be individually rational. In equilibrium, the bidders place no worse bids than their private class, and the contracted values will be no less preferred by the winner than his bid. On the other hand, the auctioneer may not sell the task with a contract worse than his private class. Thus individual rationality holds.

Budget-balance: In a single-issue (money) case the payment made by the buyer should equal the payment received by the seller for this property to hold. This can be generalized to multi-attribute case, where such a 'balance' is required in every attribute. In our case the attribute values are fixed in the contract: the seller provides the exact contracted values to the buyer.

Though the original definition of efficiency is not applicable in case of non-monetary attributes, the mechanism implements efficient outcome in a local sense, from the auctioneer's point of view. No best bidder could be selected for the auctioneer than the actual winner, but from the market's point of view there could be a better choice. Thus local efficiency may not express the same objective measure as the original one. If a locally efficient allocation was an efficient allocation in case we could express the value of the attributes in money is an open question.

However, local efficiency is enough for the mechanism to be ex-ante individually rational in equilibrium. If every bidder places a bid from his private class, then the mechanism ensures that the outcome will be as good for the winner as his bid. Any bidder can safely participate in the auction, as long as he bids from his private class, he cannot be worse off than not participating at all.

6 Conclusions and Future Research

Auctions are a popular form of distributed rational decision making. Several mechanisms have been developed for allocating a single task to a single server, multiple goods to multiple buyers, or a single good with multiple attributes to a single buyer. At the design of such mechanisms, the usual assumption to make is that the bidders and the auctioneer have a utility function over the space of the single or multiple attributes. This utility function implicitly expresses the players' preferences by assigning a value to the different outcomes.

In the current study we have relaxed this assumption, and have shown how multi-attribute auctions can be conducted when no utility value can be assigned to combinations of attribute values. We argue that this is important because it can capture situations with numerically not expressible attributes, which were impossible to capture before. We have suggested an incentive-compatible mechanism to allocate a single task with multiple attributes to a single server based only on preference orders. The mechanism implements a locally efficient outcome in an equilibrium, and it is individually rational for the players.

To extend the theory from total orders to partial orders is an interesting direction for further research. Besides, more investigation is required to determine the communicational needs of the mechanism. Currently it is not clear, what is the best way to represent and communicate equivalence classes based on the parties' orderings. Another future assignment is to implement the mechanism in artificial agents to enable them to negotiate over numerically not (easily) expressible attributes. One possible application is in multi-agent logistics, where truck agents and container agents negotiate over price, decommitment-penalty, deadlines, etc. Such bilateral multi-attribute negotiations render logistics to be a promising application domain.

7 Acknowledgments

This work was conducted in the DEAL project (Distributed Engine for Advanced Logistics), which is funded by the Dutch Ministry of Economic Affairs. Special thanks are due to Valentin Robu.

References

- [1] Y. K. Che. Design competition through multidimensional auctions. *RAND Journal of Economics*, 24:668–680, 1993.
- [2] E. David, R. Azoulay-Schwartz, and S. Kraus. Protocols and strategies for automated multi-attribute auctions. In *Proceedings of the First Joint Conference on Autonomous Agents and Multiagent Systems*, pages 77–85, 2002.
- [3] M. O. Jackson. Mechanism theory. In Ulrich Derigs, editor, *Encyclopedia of Life Support Systems*. EOLSS Publishers, Oxford UK, 2003.
- [4] Anton Likhodedov and Tuomas Sandholm. Auction mechanism for optimally trading off revenue and efficiency. In *EC '03: Proceedings of the 4th ACM conference on Electronic commerce*, pages 212–213, New York, NY, USA, 2003. ACM Press.
- [5] Jeffrey K. MacKie-Mason and Hal R. Varian. Generalized Vickrey auctions. Technical report, Un. of Michigan, July 1994.
- [6] D. Parkes and J. Kalagnanam. Iterative multiattribute Vickrey auctions. Technical report, Harvard University, 2002.
- [7] David C. Parkes and Jayant Kalagnanam. Models for iterative multiattribute vickrey auctions. *Management Science*, 51:435–451, 2005. Special Issue on Electronic Markets.
- [8] T. Sandholm. Automated mechanism design: A new application area for search algorithms. In *Proceedings of the International Conference on Principles and Practice of Constraint Programming (CP-03)*, Kinsale, Ireland, 2003.
- [9] Tuomas W. Sandholm. Distributed rational decision making. In Gerhard Weiss, editor, *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, pages 201–258. The MIT Press, Cambridge, MA, USA, 1999.
- [10] Hal R. Varian. Economic mechanism design for computerized agents. In *Proc. of Usenix Workshop on Electronic Commerce*, July 1995.
- [11] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *Journal of Finance*, 16:8–37, 1961.

AGENT-BASED SCHEDULING FOR AIRCRAFT DEICING ¹

Xiaoyu Mao ^{ab} Adriaan ter Mors ^{ac} Nico Roos ^b Cees Witteveen ^c

^a *Almende B.V.*,

Westerstraat 50, 3016 DJ Rotterdam, The Netherlands

^b *IKAT, University of Maastricht,*

P.O. Box 616, 6200 MD Maastricht, The Netherlands

^c *EWI, Delft University of Technology,*

P.O. Box 5031, 2600 GA Delft, The Netherlands

{xiaoyu, adriaan}@almende.com

Abstract

The planning and scheduling of the *deicing* and *anti-icing* activities is an important and challenging part of airport departure planning. Deicing planning has to be done in a highly dynamic environment involving several autonomous and self-interested parties. Traditional centralized scheduling approaches do not acknowledge the autonomy of parties involved. We therefore propose an agent-based scheduling approach for solving the aircraft deicing problem where stakeholders are modelled as autonomous agents. Based on our view that multi-agent scheduling is scheduling combined with agent coordination, we propose a simple *first come, first served* heuristic combined with the concept of decommitment penalties. We demonstrate the feasibility of the approach through a series of experiments.

1 Introduction

The planning and scheduling of the *deicing* and *anti-icing* activities at airports is an important and challenging part of airport departure planning. Aircraft deicing and anti-icing is required in winter time when frost, snow, and ice can form on the wings and fuselage of an aircraft. Such a layer of frost or ice on aircraft surfaces influences the aircraft's aerodynamic properties which may cause a loss of lift that could result in a crash. Deicing refers to the removal of frost, snow, or ice from aircraft surfaces, while anti-icing is the application of a layer of viscous fluid onto aircraft surfaces that should prevent snow or ice from accumulating. Since the deicing and anti-icing operations are always performed together, in the remainder of this paper we will not distinguish them and will use the term *deicing* to refer to both deicing and anti-icing.

Planning and scheduling the deicing of aircraft has to be done in a highly dynamic environment involving several autonomous and self-interested parties. The dynamic nature of the aircraft deicing problem stems from the fact that in many temperate climate zones as found in Western Europe, the process of deicing is not part of the original flight plan, and thus it has scheduled as part of *operational* (i.e., short-term) planning. Moreover, during wintry conditions involving snow and ice, airport capacities will be greatly reduced — again, in temperate climate zones, this is not taken into account in the flight schedules — putting a great strain on the re-planning capabilities of all parties involved. The parties involved are self-interested and often have conflicting interests. For instance, airlines and pilots will be concerned with the effects of deicing on their flight schedules, air traffic control will be responsible for safe flight movements, the airport itself will strive for a maximum utilization of its facilities (runways, gates, etc.), and the ground servicing companies performing the deicing will want to operate as efficiently as possible.

In this paper, we propose an agent-based scheduling architecture for solving the aircraft deicing problem where

- multiple stakeholders are modelled as autonomous agents having their own interests and value systems,

¹This research is supported by Dutch technology exchange program Casimir, funded by NWO (Netherlands Organization for Scientific Research). Project No.CSI4006: Hybrid Human-Agent Networks.

- a fall-back to local agent planning is allowed to deal with the dynamic nature of the problem, i.e., in case the global, coordinated plan is rendered infeasible by incidents.

Furthermore, we investigate how we can combine traditional scheduling heuristics with agent coordination. For the local agent strategies we simply choose First Come, First Served (FCFS), as it is quite representative of scheduling heuristics, and to coordinate the agents we will introduce the notion of *decommitment penalties*: in case an agent has reserved a specific slot for deicing at the *deicing station*, but due to a delay at the gate it cannot make this slot, it will have to pay a decommitment penalty. In a FCFS environment, the first agent to request a resource will be assigned the use of it. This will encourage agents to reserve resources as quickly as possible. This is, however, not what we want in an environment where long-term plans are unstable, and will be subject to re-planning. The decommitment penalties should counter this effect by creating an incentive only to reserve a resource when an agent is sufficiently sure that it does not need to decommit later on. Hence, the introduction of decommitment penalties requires agents to reason about uncertainty, and to reserve slots only when they have a certain degree of confidence they can honour their agreements. Decommitment penalties are therefore a way to tackle the problem of sharing the use of scarce resources among self-interested agents.

The remainder of this paper is organized as follows. In Section 2, we describe the background of the airport deicing scheduling problem, and we link it to the problem of multi-agent scheduling. In Section 3 we will give a formal model of the deicing scheduling problem and we will introduce a simple solution scheme. The agent coordination mechanism — the decommitment penalty — will be discussed in Section 4; in Section 5 we will discuss some experimental results obtained with the use of decommitment penalties. Section 6 concludes with a look to the future.

2 Background Description

As soon as an aircraft has safely landed at the airport, a sequence of tasks must be planned and scheduled before the aircraft can take off again. By reducing the time an aircraft spends on the ground between flights (the *turnaround* time), an airline can handle more flights a day, thereby increasing its revenues. Aircraft deicing is just one of the *ground servicing tasks* that an aircraft must undergo, but it plays a significant role in the turnaround time, due to its close relation with departure planning: after an aircraft has been deiced (and in particular, *anti-iced*) it needs to take off quickly before its ‘anti-icing’ layer wears off, and new ice will re-form on the aircraft surfaces. The maximum amount of time that may elapse between deicing and take-off (called the *holdover time*) will typically be around 15 minutes (though it depends on the severity of the weather conditions), which is quite short when you realize that in this time the aircraft still has to taxi from the deicing station to the runway.

An aircraft may have several alternatives to receive deicing treatment: most commonly, it will taxi to one of the deicing stations, located (hopefully) at strategic positions around the airport; second, it can also be deiced at the gate, in which case a *deicing vehicle* will drive to the gate at which the aircraft is docked. Since the total deicing capacity at an airport is usually limited, careful planning and scheduling of these resources is of crucial importance to efficient departure planning.

Like many real world problems, the problem of deicing resource management exhibits characteristics of both planning² and scheduling³. It is a scheduling problem in the sense that aircraft tasks have to be located in time on resources, and it is a planning problem in the sense that an aircraft has a number of choices with regard to which deicing resource to make use of — and this choice of deicing resource has implications for other airport planning problems like arrival planning, departure planning, and taxiway planning. Nevertheless, the management of deicing resources can best be characterized as a scheduling problem as it involves only a small, fixed number of choices, and because the focus is more on time and resource constraints, rather than ordering of actions (cf. [6]).

Based on a classification of scheduling problems by Yang et al.[10], we classify the airport deicing scheduling problem as a Multi-mode Resource-Constrained Project Scheduling Problem(MRCPSP), with the following characteristics:

²The *planning* problem is usually defined as finding a sequence of actions that will transform the initial state of the world into a state where the goals are attained.

³The *scheduling* problem is to assign limited resources to tasks over time to optimize one or more objectives.

- *non-regular objective function*: rather than the traditional objective of minimizing the total makespan of a project, each aircraft aims to depart close to its Target Time of Departure (TTD);
- *generalized precedence constraints*: between deicing and take-off not only exists a precedence constraint, but also an additional temporal constraint — the holdover time;
- *stochastic processing time*: the duration of deicing depends on how much ice has accumulated on the aircraft, which often only becomes clear as the aircraft arrives at the deicing station;
- *transportation delays*: an aircraft must taxi between deicing and take-off (the jobs), but this taxi time depends on its choice of runway and deicing station;
- *non-preemptive operations*: once the deicing process has commenced, it is not interrupted to make way for another aircraft

Scheduling literature surveys [2, 3, 4] reveal that all algorithms for determining an optimal solution of the MRCPSPP as well as most heuristic approaches are based on centralized problem solving. A centralized approach is impractical, however, when jobs and resources are distributed over different stakeholders. We therefore propose that in this domain, scheduling should not be done *for* multiple agents (or at least, not exclusively), but *by* different agents. Moreover, as different stakeholders can have different and often conflicting interests, there will be a limit to how much they are willing to accommodate the scheduling of other agents, and how much planning information they are willing to exchange.⁴

Of course, if agents were to schedule completely independently of each other, the union of their plans would show many conflicts. In the airport deicing domain, these conflicts will concern the simultaneous use of scarce resources. We therefore define the problem of multi-agent scheduling as follows (cf. [1]):

Definition 1 (Multi-Agent Scheduling) *Given a set of agents each with a set of jobs to schedule, and a set of resources to schedule them on, each agent should find an individual schedule for its jobs in such a way that none of the resource capacity constraints are violated.*

Obviously, satisfying all resource constraints will not happen by magic; the agents will need some coordination mechanism that will safeguard these constraints. Therefore, we can summarize the multi-agent scheduling problem as follows:

$$\boxed{\text{Multi-Agent Scheduling} = \text{Scheduling} + \text{Coordination}}$$

3 Formal Modelling

In this section we will present a formal model of the aircraft deicing scheduling problem. The model presented below is a simplified version of the problem described in the previous section, as we leave out other airport planning problems as runway planning and gate allocation, and as a result certain ‘inter-planning-problem constraints’ such as the holdover time are not taken into account. The following model has also been used as the basis of the experiments, which are described in Section 5.

Definition 2 (Aircraft Deicing Scheduling Problem) *The aircraft deicing scheduling problem is a tuple $\langle A, D, c, \tau, d, l \rangle$ where*

- A is a set of n aircraft agents,
- D is a set of m deicing station resources,
- $c : D \rightarrow \mathbb{N}$ is a capacity function specifying the number of aircraft that can simultaneously be serviced at the deicing station (i.e., the number of deicing bays),
- $\tau : A \rightarrow \mathbb{N}$ is a function indicating the Target Start Deicing Time (TSDT) for each aircraft,
- $p : A \rightarrow \mathbb{N}$ is function that specifies the deicing process duration for a certain aircraft,
- $l : \mathbb{N} \times A \rightarrow \mathbb{N}$ is a function that assigns a cost to the delay of an aircraft,

⁴In [9], this problem is studied from a multi-agent *planning* perspective.

The target start deicing time $\tau(a_i)$ defined in Definition 2 is the earliest possible start deicing time for aircraft agent a_i , which is in fact the time when all other ground services for this agent are assumed to be finished. A solution to an instance $\langle A, D, c, \tau, d, l \rangle$ of the deicing problem is a multi-agent schedule given by the vector $S = \langle (d_1, I_1), \dots, (d_n, I_n) \rangle$ where (d_i, I_i) is a tuple in which d_i is the deicing station assigned to agent a_i during interval I_i such that

$$I_i = [s_i, s_i + p(a_i)] \quad \wedge \quad s_i \geq \tau(a_i) \quad (1)$$

A feasible schedule satisfies the following resource constraints: at every point in time t , the deicing resource utilization for each resource agent does not exceed the resource capacities. We have:

$$\forall t \forall d \in D |\{a_j \in A \mid (d, I_j) \in S \wedge t \in I_j\}| \leq c(d) \quad (2)$$

Given a Target Start Deicing Time for each aircraft agent a_i , the objective is to find a schedule S where the delay cost of all aircraft is minimized:

$$\min \sum_{a_i \in A} l(s_i - \tau(a_i), a_i) \quad (3)$$

where $s_i - \tau(a_i)$ is the delay of aircraft a_i .

4 Decommitment in FCFS

A standard heuristic approach to scheduling processes uses a *First Come, First Served* (FCFS) queue. It is starvation-free, very simple, and in some cases reasonably efficient. Therefore, it seems a reasonable choice for Air Traffic Control (ATC) to process aircraft requests for services as deicing.

If we apply the FCFS policy in a multi-agent environment, agents are encouraged to reserve a slot as soon as possible. In a highly dynamic environment, in which incidents frequently occur that invalidate existing plans, this is not what we want; in fact, it is almost the opposite of what we want. Instead, we would encourage agents to schedule slots only in a future of sufficiently clarity. Therefore, we investigate whether we can design a coordination mechanism that can provide the correct incentives by making use of *decommitment penalties*.

In previous research, decommitment has been primarily used to enable agents to explore new opportunities from the domain or from other agents [7, 5, 8]; an example might be a package-delivery agent that decommits the contract for one package so that it is able to accept a more profitable package to deliver [7]. We propose that the concept of decommitment penalties can also be used to coordinate agents.

When an aircraft agent reserves a particular time slot at a resource such as a deicing station, it will commit to turn up at that deicing station at the specified time. If the aircraft fails to show up (possibly because of some incident that occurred on the airport, such as taxiway congestion, or delayed refuelling), it has to pay a decommitment penalty to the deicing station. Hence, with the introduction of decommitment penalties, agents have an incentive to reserve as late as possible; after all, if it reserves a slot five minutes from now, it will be fairly certain it can make this slot. On the other hand, the old incentive for scheduling as early as possible remains: agents will still want to schedule slots before other agents get them.

4.1 The Aircraft Problem

We assume that an aircraft has to obtain exactly one slot for deicing. It can obtain this slot by reserving a free slot at a deicing resource. However, with a certain probability incidents occur that make it impossible for the aircraft to be present at the deicing station at the agreed time. When such an event occurs, it must decommit and pay a decommitment penalty, which we assume to be an airport-wide constant δ . If the aircraft decommits, then it has to try again to reserve a slot at a deicing station. We assume that the agent can see when the first available slot is at all deicing stations (we will refer to this time as the Earliest Available Slot Time (t_{EAST})). To simplify the aircraft agent strategy, we assume that a deicing station never has to decommit.

As an aircraft agent can see the earliest available slot, it has to solve the following decision problem:

Do I reserve the currently available first slot, or do I reserve a slot at a later time?

To answer this problem, the agent a has to be able to evaluate his two different options. To judge whether the decision to reserve now has any merit, the agent needs to value the current slot based on how close it is to the $\tau(a)$, but it also needs to estimate the probability it will have to decommit from the slot. We assume that this probability of having to decommit is dependent on how far in the future the reserved slot is. Judging the option of reserving a slot at a later time is more difficult, as it needs to predict the availability of deicing slots in the future. This availability depends on at least the following factors:

1. incidents may occur to other aircraft, freeing up deicing resources,
2. the passage of time; if a slot is available 11 minutes from now, then, if no-one else takes it, there will be a slot 10 minutes in the future one minute from now,
3. other agents reserve slots.

Trying to incorporate all these factors into a realistic model is a formidable task, especially as the slot-reserving behaviour of agents may be subject to their perception (and prediction) of other agents' behaviour. Therefore, we will make the following simplifying assumptions to make the task of foretelling the future a more tractable one:

- If an agent has to decommit from a slot, then it will have to find a new slot. Apart from the time lost in decommitment, we assume that deciding whether or not to take the current slot is independent of time. Hence, we assume no peak hours in which finding a slot is more difficult.
- The delay an agent suffers when it has to decommit will mainly depend on the time it decommits; here we assume a constant value for this delay.
- When an aircraft opts to postpone its decision to reserve a slot until the next round, and it turns out that another agent has reserved the previously earliest slot, then the new t_{EAST} is simply the old t_{EAST} plus some constant value.

Armed with these simplifications, we can develop a strategy for an aircraft agent.

4.2 Aircraft Agent Strategy

The heuristic we will investigate in this section can be described as follows:

Reserve the earliest available slot if the expected cost of reserving this slot is less than the expected cost of reserving a slot the next round; otherwise, postpone the reservation decision until the next round.

We will now introduce a number of functions to be able to define the expected cost of reserving the earliest available slot. We assume that the probability of an incident occurring is a linear function of the time passed since the reservation time t . Hence, the probability that an incident will occur until the reservation is used equals:

$$p_d(t, t_{\text{EAST}}) = \min(1, \beta \cdot (t_{\text{EAST}} - t)) \quad (4)$$

where $0 \leq \beta \leq 1$, t is the time the reservation is made, and t_{EAST} is the start time of the reserved slot. An aircraft can reserve a slot from the Target Start Deicing Time onwards, i.e., $\tau(a) \leq t_{\text{EAST}}$. If the aircraft obtains a slot that is later than this target time, it incurs additional cost ('an aircraft on the ground costs the airline money'). We assume that the delay cost per minute α is the same for all aircraft, resulting in the following cost function:

$$l(t) = \alpha \cdot t \quad (5)$$

where t is the time in minutes that the aircraft is delayed; if an aircraft has reserved a slot at t_{EAST} , then its delay will be $t_{\text{EAST}} - \tau$. Ultimately, an aircraft's delay cost will be related to its take-off time, but as deicing is quickly followed, in general, by take-off, a delay in deicing will translate into a delay in take-off time.

We can now define the cost of being delayed as a result of having to decommit. First of all, an agent has to pay the decommitment penalty δ ; second, if t_d stands for the time decommitment occurs, then the aircraft has wasted $(t_d - t)$ minutes (where t is the time at which the slot it reserved). Again, we assume that this

quantity $(t_d - t)$ will in fact delay deicing — and therefore take-off — by $(t_d - t)$ minutes. For simplification purposes, we will assume (i) that $(t_d - t)$ is a fixed value, and so we define the cost of decommitment as:

$$dcp = \delta + l(t_d - t) \quad (6)$$

The cost specified in Equation 6 is the *immediate* cost that the aircraft incurs — this is the cost the aircraft will have to pay anyway — without taking into account that there is a chance of having to decommit again in the future. Using the above definitions, an agent can calculate the expected cost of reserving a slot at time t with earliest available slot time t_{EAST} :

$$E_{\text{res}}(t, t_{\text{EAST}}) = p_d(t, t_{\text{EAST}}) \cdot dcp + (1 - p_d(t, t_{\text{EAST}})) \cdot l(t_{\text{EAST}} - \tau) \quad (7)$$

Note that a more realistic model for the cost of reserving a slot would be forward recursive: in case an aircraft has to decommit, it will have to try to get a slot again in subsequent rounds, again with the possibility of having to decommit, adding to its cost. Equation 7 effectively cuts off this forward recursion after one step, by taking into account only the immediate cost for decommitment.

The expected cost of waiting until the next round is given by the following function:

$$E_{\text{wait}}(t) = p_T(t, t_{\text{EAST}}) \cdot E_{\text{res}}(t^+, t_{\text{EAST}}^+) + (1 - p_T(t, t_{\text{EAST}})) \cdot E_{\text{res}}(t^+, t_{\text{EAST}}) \quad (8)$$

in which $p_T(t, t_{\text{EAST}})$ stands for the probability of another agent having reserved between time t and t^+ the slot starting at t_{EAST} . This probability function is based on the number of aircraft in the system, and the scarcity of the deicing resources. We assume aircraft take-off times are independent of each other and are uniformly distributed over time, and so we model the probability $p_T(t, t_{\text{EAST}})$ with a Poisson distribution ($f(k; \lambda) = \frac{e^{-\lambda} \lambda^k}{k!}$) where:

$$p_T(t, t_{\text{EAST}}) = 1 - f(0, \frac{t^+ - t}{|D| \cdot T}) = 1 - e^{-\frac{|A| \cdot (t^+ - t)}{|D| \cdot T}} \quad (9)$$

where T is the time in minutes over which these aircraft are distributed (e.g., we could have a simulation run of $T = 360$ minutes in which $|A| = 100$ aircraft have to be deiced using $|D| = 4$ deicing stations).

Note that we have assumed above in Equation 8 that the value of t_{EAST}^+ is simply the value t_{EAST} plus some constant. Equation 8 basically expresses that by not reserving a slot this round, there is a chance that another agent reserves the previously earliest available slot, and you consequently have to schedule a later slot t_{EAST}^+ (which will result in more delay); on the other hand, if no agent has reserved the slot starting from t_{EAST} , then this possibility is still open to you at time t^+ . By this time, the probability of decommitment will have lowered (i.e., $p_d(t^+, t_{\text{EAST}}) < p_d(t, t_{\text{EAST}})$), and thus reserving this slot at time t^+ will have a lower expected cost.

The agent strategy we propose in this section is simple: in case $E_{\text{res}} < E_{\text{wait}}$, the agent will reserve at time t the slot starting at t_{EAST} , otherwise it will wait until the next round. In the next section, we will investigate whether reasoning about decommitment in this way results in improved performance.

5 Experimental Results

In this section, we will evaluate the efficiency of determining the time to reserve a slot using the decommitment formulas from the previous sections, assuming the deicing stations apply a FCFS queue. We compare this with a FCFS queue in which the aircraft agents request slots arbitrarily.

We conducted these experiments using only a single deicing station having 3 bays; deicing processing durations and aircraft delay cost functions are assumed to be the same for all aircraft (we chose a deicing time of 20 minutes), and the Target Start Deicing Times were randomly distributed over a six-hour interval. For these parameters, the maximum number of aircraft that can be serviced without delay equals $n = \frac{3 \times 6 \times 60}{20} = 54$, assuming a maximally convenient distribution of aircraft Target Start Deicing Times (τ). This means

that with a random distribution of τ , we can expect some delays regardless of the scheduling strategy in case we have more than 54 aircraft. Some further parameter values are $\alpha = 1$, $\beta = \frac{1}{60}$, and $dcp = 50$. We have tested both strategies for different numbers of aircraft, and for each number of aircraft, we have performed three runs. The averaged results are displayed in Table 1.

Number of aircraft	FCFS		FCFS + DC	
	Total delay	Standard deviation	Total delay	Standard deviation
10	0	0	0	0
15	0	0	0	0
20	42	6.08	9	1.19
25	114	13.5	19	2.35
30	99	9.32	56	4.94
35	312	18.8	98	6.18
40	546	30.7	208	9
45	1044	41.4	391	12.9
50	2343	82.33	579	14.4
55	2973	83.5	1611	24.23
60	4220	113.5	1770	24.3
65	6090	124.33	3007	31.6
70	7392	140	4567	40
75	9453	158.33	6395	43.17
80	11667	169.67	7790	49.83

Table 1: Total delay and standard deviation in FCFS and FCFS+DC

In Figure 1.a, we see that using decommitment-penalty reasoning, we end up with a lower delay regardless of the number of aircraft. The reason for this is that the decommitment penalties encourage the agents to schedule only a relatively short period of time (no more than an hour) in advance of their Target Start Deicing Time. As a result, the Earliest Start Time (EST) algorithm emerges as the combination of aircraft agent strategies. From our experiments, this heuristic proves to be more efficient than arbitrary-order scheduling.

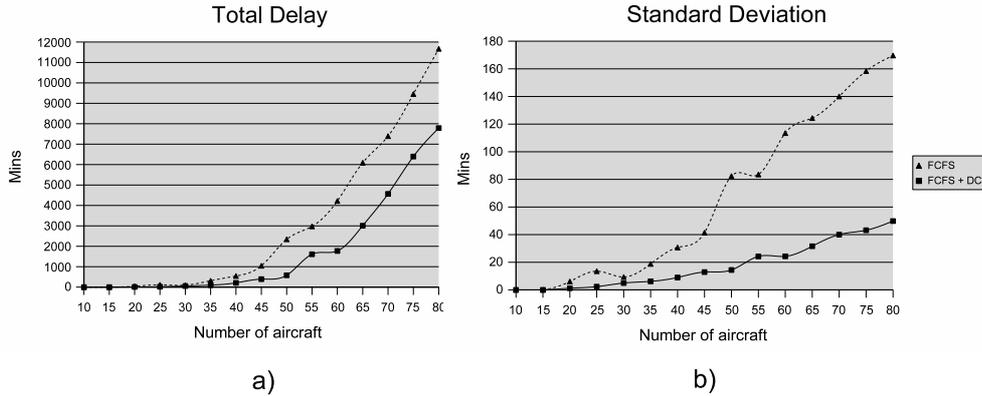


Figure 1: Summarized results of total delay and standard deviation in FCFS and FCFS+DC

Figure 1.b shows the standard deviation for delays of individual aircraft. The higher standard deviation for the non-decommitment version is caused by the fact that some aircraft have very little delay, while others have great deal of it. Using decommitment reasoning, delay is distributed more evenly over the aircraft. In a sense, decommitment reasoning thus also improves the fairness of agent interactions.

6 Conclusions & Future Work

In this paper we have discussed an agent-based model for the scheduling of airport deicing services. By introducing the idea of decommitment penalties for aircraft that reserve slots but fail to turn up, we enabled an agent-based scheduling of the problem that improves on a naive (and greedy) random-order First Come, First Served queue.

Given the early stage of our research, options for future work are too numerous to list exhaustively. We would like to investigate other scheduling strategies in conjunction with decommitment penalties. Also,

our results currently rely on many simplifying assumptions, and it would be interesting to see whether the conclusions of this paper hold up if we relax some of these assumptions.

Another extension of high priority is to look at the relation with other airport planning and scheduling problems. In itself, the deicing problem as formulated in the formal model of Section 3 is not that exceptional. What makes the problem interesting to look into is its relation to other planning problems, possibly involving other planning agents. The challenge for airport deicers lies in inserting the deicing scheduling problem into other well-known scheduling problems — into existing plans, even — such as arrival and departure planning, which, under everyday conditions, do not take into account an extra trip to the deicing station.

Acknowledgments

This research is funded by NWO (Netherlands Organization for Scientific Research), Grant No. CSI4006, which is gratefully acknowledged. We are indebted to Peet van Tooren, Jeroen Valk, Tamás Máhr, Gerald de Jong, and Jan Peter Larsen (Almende B.V.) as well as Jaap van den Herik (University of Maastricht) for their help in this research.

References

- [1] Mathijs de Weerd, Adriaan ter Mors, and Cees Witteveen. Multi-agent planning: An introduction to planning and coordination. In *Handouts of the European Agent Summer School*, pp. 1-32, 2005.
- [2] Willy Herroelen, Bert De Reyck, and Erik Demeulemeester. Resource-constrained project scheduling: a survey of recent developments. *Comput. Oper. Res.*, 25(4):279-302, 1998.
- [3] Rainer Kolisch and Sönke Hartmann. Heuristic algorithms for solving the resource-constrained project scheduling problem: Classification and computational analysis. In J. Weglarz, editor, *Project scheduling: Recent models, algorithms and applications*, pages 147 - 178, Kluwer, Amsterdam, the Netherlands, 1999. Kluwer academic.
- [4] Rainer Kolisch and Sönke Hartmann. Experimental investigation of heuristics for resource-constrained project scheduling: An update. *European Journal of Operational Research*, 23, May 2005.
- [5] Tuomas Sandholm and Victor Lesser. Leveled commitment contracts and strategic breach. *Games and Economic Behaviour*, 25:212-270, 2001.
- [6] David E. Smith, Jeremy Frank, and Ari K. Jónsson. Bridging the gap between planning and scheduling. *Knowl. Eng. Rev.*, 15(1):47-83, 2000.
- [7] P. J. 't Hoen and J. A. La Poutré. A decommitment strategy in a competitive multi-agent transportation setting. In *AAMAS '03: Proceedings of the second international joint conference on Autonomous agents and multi-agent systems*, pages 1010-1011, New York, NY, USA, 2003. ACM Press.
- [8] Vanessa Teague and Liz Sonenberg. Investigating commitment flexibility in multi-agent contracts. *Game Theory and Decision Theory in Agent-Based Systems*, pages 267-292, 2002.
- [9] Jeroen Valk. *Coordination among Autonomous Planners*. PhD thesis, Delft University of Technology, December 2005.
- [10] Bibo Yang, Joseph Geunes, and William J. O'Brien. Resource-constrained project scheduling: Past work and new directions. Technical report, Department of Industrial and Systems Engineering, University of Florida, June 2001.

ARGUMENTATION TO COMPOSE SERVICES¹

Maxime Morge ^a

Jean-Christophe Routier ^b

Yann Secq ^b

^a *Dipartimento di Informatica - University of Pisa, via F. Buonarroti, 2 I-56127 PISA*

^b *LIFL - University of Lille 1, Bat M3 F-59655 VILLENEUVE D'ASCQ*

Abstract

We propose in this paper a framework for inter-agents dialogue on actions, which formalize a deliberative process. This framework bounds a dialectical system in which argumentative agents arbitrate and play to reach a practical agreement. For this purpose, we propose an argumentation-based reasoning to manage the conflicts between plans having different strengths for different agents. Moreover, we propose a model of agents which justify the plans to which they commit and take into account the plans of their interlocutors. In the scope of our dialectical system, an agent is responsible of the final decision outcome which is taken according to the authority of the players, the uttered plans and her own rules and priorities. We illustrate this paper with a services composition.

1 Introduction

The collaboration between autonomous and social agents to solve complex tasks is an open problem with many application areas such as cooperative robotics or services composition. The conflicts in the interests and perspectives of agents is the main characteristic of such systems. In this paper, we aim at formalizing a deliberative process with a formal framework for inter-agents interaction. For this purpose, argumentation is a promising approach for reasoning with inconsistent information and conflicting objectives. In this paper, we extend [8] such as arguments are plans and we focus on a dialogical mechanism between software agents to reason, exchange and compose them.

Paper overview. In section 1, we provide the syntax and the semantics of the planning language. Section 3 presents the argumentation framework which manages the interaction between conflicting plans. In accordance with this background, we describe in section 4 our model of agents. In section 5, we define the formal area in which the agents deliberate. Section 6 presents the protocol used to reach a practical agreement. This paper closes with discussion about related works in section 7.

2 Planning language

In this section, we present the syntax and the semantics of the formal language that we use to express automated planning problem instance : the Action Description Language (ADL). This is an extension of the STRIPS language (*Stanford Research Institute Problem Solver*),

The classical planning environments are fully observable, deterministic (the result of actions are foreseeable), discrete (the environment states are finite) and static (changes happens only when the agents acts). Planners decompose the world into logical conditions and represent a state. For example, Rich \wedge Famous represents the state of an *happy* agent. Literals in first-order state descriptions must be ground and function-free. Contrary to $\text{in}(a_1, \text{Ottawa})$, literals such as $\text{in}(a_x, y)$ or $\text{in}(\text{Daughter}(a_x), \text{Lille})$ are not allowed.

A planning problem instance is composed of an initial state, the specification of the goal which the planner is trying to reach, and a set of possible actions. Each action is defined in terms of preconditions and postconditions.

Definition 1 An *instance* is defined by an ordered pair $\mathcal{I} = \langle \mathcal{L}, C \rangle$ where:

¹This work is supported by the CPER TAC of the region Nord-Pas de calais and the european fund FEDER and the first author is supported by the Sixth Framework IST programme of the EC, under the 035200 ARGUGRID project.

- \mathcal{L} is the first-order logic language. We call **condition** a conjunction of literals. The empty conjunction (denoted \emptyset) is true;
- $C = A \cup \{i, g\}$ is a **competence**, i.e. a set of rule where:
 - A is a set of action. Each action a is a 4-tuple of conditions $a : \langle \gamma, \delta \rangle \leftarrow \langle \alpha, \beta \rangle$, where α must be true for the action to be executable, β must be false, γ is made true by the action and δ ones is made false. The empty action is denoted $\epsilon : \langle \emptyset, \emptyset \rangle \leftarrow \langle \emptyset, \emptyset \rangle$;
 - $i : \langle \gamma_i, \delta_i \rangle \leftarrow$ is the initial state, given as the pair of conditions, which are respectively true and false ;
 - $g : \langle \gamma_g, \delta_g \rangle \leftarrow$ is the specification of the goal state given as a pair, which specify that the condition α_g (resp. β_g) is true (resp. false), in order for a state to be considered as a goal.

On one hand, a state *a fortiori* the initial state i is a totally specified state represented as a conjunction of positive ground literals (γ_i) and a conjunction of negative ground literals (δ_i). On the other hand, the specification of a goal g is a partially specified state represented as a conjunction of positive ground literals (α_g) and a conjunction of negative ground literals (β_g). A state ($s : \langle \gamma_s, \delta_s \rangle \leftarrow$) satisfies a goal ($g : \langle \alpha_g, \beta_g \rangle \leftarrow$) if s contains all the atoms in g and possibly others ($\alpha_g \subseteq \gamma_s$ and $\beta_g \subseteq \delta_s$). An action a is specified in terms of positive preconditions (α) and negative preconditions (β) that must hold before it can be executed, the positive effects (γ) and the negative effects (δ) that ensure when it is executed. For example, an action for flying a plane from one location to another is: $\text{fly}(\text{from}, \text{to}) : \langle \text{in}(\text{buyer}, \text{to}), \text{in}(\text{buyer}, \text{from}) \rangle \leftarrow \langle \text{in}(\text{buyer}, \text{from}), \emptyset \rangle$. This rule is also called action schema, meaning that it represents a number of different actions that can be derived by instantiating the variables *from*, and *to* to different constants. The action name $\text{fly}(\text{from}, \text{to})$ serves to identify the action, Any variables in the preconditions or in the postconditions must also appear in the action's parameter list. One of the most important restrictions is that literals be function-free. With this restriction, we can be sure that any action schema can be propositionalized, so turned into a finite collection of purely propositional action representations with no variables. Since the preconditions expressing the facts that a flight cannot be made from an airport to itself, cannot be expressed succinctly in STRIPS, we prefer the Action Description Language to the STRIPS language. To improve readability, we have divided the preconditions and the effects into lists for positive and negative literals. In ADL, the fly action could be written as : $\text{fly}(\text{from}, \text{to}) : \langle \text{in}(\text{buyer}, \text{to}), \text{in}(\text{buyer}, \text{from}) \rangle \leftarrow \langle \text{in}(\text{buyer}, \text{from}), \text{from} = \text{to} \rangle$.

Having defined the syntax for the representation of planning problems, we can now define the semantics. For this purpose, we specify a direct transition from a state ($s : \langle \alpha_s, \beta_s \rangle \leftarrow$) into successor-state axioms whose semantics come from first order logic. An action ($a : \langle \gamma_a, \delta_a \rangle \leftarrow \langle \alpha_a, \beta_a \rangle$) is applicable in any state that satisfies the preconditions. More formally:

$$s, a \vdash \begin{cases} \langle (\alpha_s \cup \gamma_a) - \delta_a, (\beta_s \cup \delta_a) - \gamma_a \rangle & \text{if } \alpha_a \subseteq \alpha_s \text{ and } \beta_a \subseteq \beta_s \\ \langle \alpha_s, \beta_s \rangle & \text{else} \end{cases}$$

Starting in a state s , the result of executing an **applicable action** a is a state s' such that any positive literals in the effect are added to s' and any negative literals are removed from s' . For example, suppose the current state is described by: $s : \langle \text{in}(\text{buyer}, \text{Paris}), \text{in}(\text{buyer}, \text{Lille}) \rangle$. This state satisfies the precondition: $\langle \text{in}(\text{buyer}, \text{from}), \text{from} = \text{to} \rangle$ with substitution $\{\text{Paris}/\text{from}, \text{Montreal}/\text{to}\}$. Thus, after the concrete action $\text{fly}(\text{Paris}, \text{Montreal})$ the current state becomes $s' : \langle \text{in}(\text{buyer}, \text{Montreal}), \text{in}(\text{buyer}, \text{Paris}) \wedge \text{in}(\text{buyer}, \text{Lille}) \rangle \leftarrow$. Note that if a positive effect is already in s it is not added twice, and if a negative effect is not in a , then that part of the effect is ignored. This definition embodies the assumption that every literal not mentioned in the effects remains unchanged. In this way, we avoid the frame problem. Since the states could be represented by a pair of conditions, two (totally or partially specified) states $s_1 : \langle \alpha_1, \beta_1 \rangle \leftarrow$ and $s_2 : \langle \alpha_2, \beta_2 \rangle \leftarrow$ are **incompatible** (denoted $s_1 \perp s_2$) iff: $(\alpha_1 \cap \beta_2) \cup (\alpha_2 \cap \beta_1) \neq \emptyset$.

Finally we can define the solution for a planning problem. The transition relation can be extended such that: $c, \epsilon \vdash c$ and $c, (a_1, a_2, \dots, a_n) \vdash (c, a_1 \vdash), (a_2, \dots, a_n)$. A plan for an instance is a sequence of actions such that the state that results from executing the actions from the initial state satisfies the goal conditions. A **plan** $P = \langle (a_1, a_2, \dots, a_n), i_P, g_P \rangle$ is a triple where:

- i_P is a totally specified state;
- g_P is a partially specified state;
- (a_1, a_2, \dots, a_n) is a sequence of actions ($\forall i a_i \in A$) such as $i_P, (a_1, a_2, \dots, a_n) \vdash g_P$.

P satisfies S iff $\gamma_i \subseteq \gamma_{i_P}, \delta_i \subseteq \delta_{i_P}, \alpha_g \subseteq \alpha_{g_P}, \beta_g \subseteq \beta_{g_P}$. In other words, a solution for a planning problem is just a plan that, when executed in the initial state, results in a state that satisfies the goal. Below we will use this planning language within our argumentation framework.

3 Argumentation-based plan

We present in this section an argumentation framework built around the previous language, which manage the interaction between plans. In order to deliberate, we consider a set of audiences ($a_1, \dots, a_n \in \mathcal{U}_A$) which adhere to different plans with a variable intensity. These audiences consider the same instance (denoted $\mathcal{I} = \langle \mathcal{L}, \mathcal{C} \rangle$) and the same transition rule (denoted \vdash). At first, we consider that the audiences share an value-based competence, *i.e.* a set of rules promoting values:

Definition 2 The *value-based competence* $AC = \langle \mathcal{C}, V, \text{promote} \rangle$ is defined by a triple where:

- \mathcal{C} is a competence, *i.e.* a finite set of rules;
- V is a non-empty finite set of values $\{v_1, \dots, v_t\}$;
- $\text{promote} : \mathcal{C} \rightarrow V$ maps from the rules to the values.

We say that the rule c relates to the value v if c promotes v . For every $c \in \mathcal{C}$, $\text{promote}(c) \in V$.

To distinguish different audiences, values, both concrete and abstract, constitute starting points [2]. Values are arranged in hierarchies. For instance, an audience will value both justice and utility but an argument may require the determination of a strict preference between the two. Since audiences are individuated by their hierarchies of values, the values have different priorities for different audiences. The **value-based competence of the audience a_i** is a 4-tuple $AC_i = \langle \mathcal{C}, V, \text{promote}, \ll_i \rangle$ where $AC = \langle \mathcal{C}, V, \text{promote} \rangle$ is a value-based competence as previously defined and \ll_i is the priority relation of the audience a_i , *i.e.* a strict complete ordering relation on V .

A priority relation is a transitive, irreflexive, asymmetric, and complete relation on V . It stratifies the competence into finite non-overlapping sets. The priority level of a non-empty competence $C \subseteq \mathcal{C}$ (written $\text{level}_i(C)$) is the least important value promoted by one element in C . On one hand, a priority relation captures the value hierarchy of a particular audience. On the other hand, the competence gathers the rules shared by the audiences. Plans are built upon this competence. Since the plans can be conflictual, *i.e.* leads to incompatible goals, plans can be considered as arguments.

Definition 3 Let C be a competence. An **argumentative plan** is a triple $A = \langle s_P, i_P, g_P \rangle$ where i_P (*resp.* g_P) is an initial state (*resp.* the specification of a goal) and $s_P = (a_1, \dots, a_n)$ a sequence of actions in C such as: $i_P, (a_1, \dots, a_n) \vdash g_P$. (i_P, s_P) is the premise of A , denoted $\text{premise}(A)$. g_P is the conclusion of A , denoted $\text{conclusion}(A)$. The argumentative plan A' is a **sub-plan** of A iff its sequence of actions is a subsequence of s_P . A plan is **trivial** iff the sequence of actions is empty.

In other words, argumentative plans (plans for short) are relations of consequence between a premise and a conclusion. Since the competence C can be conflictual, the set of plans (denoted $\mathcal{P}(C)$) will conflict. The relation of attack between plans is built upon the incompatibility between their conclusions.

Definition 4 Let C be a competence and $A, B \in \mathcal{P}(C)$ two plans. A **attacks** B iff: $\exists A_1 = \langle s_1, i_1, g_1 \rangle, B_2 = \langle s_2, i_2, g_2 \rangle \in \mathcal{P}(C)$ respectively sub-plan of A and B such as $g_1 \perp g_2$.

Because each audience is associated with a particular priority relation, the audiences individually evaluate the strength of plans. According to the audience a_i , the **strength** of A (written $\text{strength}_i(A)$) is the least important value promoted by one rule in the premise. In other words, the strength of plans depends on the priority relation. Since the audiences individually evaluate the strength of plans, an audience can ignore the attack of a plan over another plan.

Definition 5 Let $AC_i = \langle \mathcal{C}, V, \text{promote}, \ll_i \rangle$ be the value-based competence of the audience a_i and $A = \langle s, i, g \rangle, B = \langle s', i', g' \rangle \in \mathcal{P}(C)$ two plans. A **defeats B for the audience a_i** iff $\exists A_1 = \langle s_1, i_1, g_1 \rangle, B_2 = \langle s_2, i_2, g_2 \rangle \in \mathcal{P}(C)$ respectively sub-plan of A and B such as: i) $g_1 \perp g_2$; ii) $\neg(\text{strength}_i(A_1) \ll_i \text{strength}_i(B_2))$. Similarly, we say that a set S of plans **defeats B** if B is defeated by a plan in S .

Contrary to the relation of attack, the relation of defeat is asymmetric and subjective. Considering the individuated viewpoint of each audience, we focus on the subjective acceptance.

Definition 6 Let $AC_i = \langle C, V, promote, \ll_i \rangle$ be the value-based competence of the audience a_i . Let $A \in \mathcal{P}(C)$ be a plan and $S \subseteq \mathcal{P}(C)$ a set of plans. A is **subjectively acceptable by the audience a_i with respect to S** iff $\forall B \in \mathcal{P}(C) \text{ defeats}_i(B, A) \Rightarrow \text{defeats}_i(S, B)$.

The set of subjectively acceptable plans consists of a consistent position, also called preferred extension, which is a maximal set of acceptable plans [6]. In other words, this set defends itself from all attacks, and cannot be extended without introducing a conflict. Since the priority relation is an ordering relation, the set of acceptable plans is unique and non-empty [4]. The following example illustrate this argumentation-based reasoning framework for plans.

Example 1 Let us consider a_1 , a service provider which wants to sell transport tickets. The value-based argumentative competence of the audience a_1 is represented in the table 1. This audience is associated with

\ll_1	V	C
↑	v_1	$g(\text{price}) : (\text{in}(\text{buyer}, \text{Ottawa}) \wedge \text{paid}(\text{price}), \emptyset) \leftarrow$ $i : (\text{in}(\text{buyer}, \text{Lille}) \wedge \text{budget}(> 2000)) \leftarrow$
	v_2	$\text{fly}(\text{from}, \text{to}, \text{price}) : (\text{in}(\text{buyer}, \text{to}) \wedge \text{paid}(\text{price}), \text{in}(\text{buyer}, \text{from})) \leftarrow \langle \text{greater}(\text{budget}, \text{price}) \wedge \text{in}(\text{buyer}, \text{from}), \text{from} = \text{to} \rangle$ $\text{bus}(\text{from}, \text{to}, \text{price}) : (\text{in}(\text{buyer}, \text{to}) \wedge \text{paid}(\text{price}), \text{in}(\text{buyer}, \text{from})) \leftarrow \langle \text{greater}(\text{budget}, \text{price}) \wedge \text{in}(\text{buyer}, \text{from}), \text{from} = \text{to} \rangle$ $\text{train}(\text{from}, \text{to}, \text{price}) : (\text{in}(\text{buyer}, \text{to}) \wedge \text{paid}(\text{price}), \text{in}(\text{buyer}, \text{from})) \leftarrow \langle \text{greater}(\text{budget}, \text{price}) \wedge \text{in}(\text{buyer}, \text{from}), \text{from} = \text{to} \rangle$
	v_3	$\text{fly}(\text{Paris}, \text{Montreal}, 800)$
	v_4	$\text{fly}(\text{Bruxelles}, \text{Montreal}, 650)$
	v_5	$\text{bus}(\text{Montreal}, \text{Ottawa}, 50)$
	v_6	$\text{bus}(\text{Montreal}, \text{Ottawa}, 100)$
	v_7	$\text{train}(\text{Lille}, \text{Bruxelles}, 100)$
	v_8	$\text{train}(\text{buyer}, \text{Lille}, \text{Paris}, 150)$

Table 1: Argumentative competence of the service provider a_1 .

a competence, i.e. a set of rules ($g(\text{price}), i, \dots$) and a set of values (v_1, \dots, v_8). The rules corresponding to the goal specification ($g(\text{price})$) and the initial situation (i) promote the value v_1 . The rules of common sense promote the value v_2 : “take a flight” ($\text{fly}(\text{from}, \text{to}, \text{price})$), “take a train” ($\text{train}(\text{from}, \text{to}, \text{price})$) and “take a bus” ($\text{bus}(\text{from}, \text{to}, \text{price})$). The other particular rules such as “take TGV” ($\text{train}(\text{Lille}, \text{Paris}, 150)$) promote the values v_3, \dots, v_8 . According to an audience, a value above another one in the figure has priority over it. a_1 prefers Air France to Air Canada and the cheapest connexion. The four following plans conflict:

$A = (i, (\text{train}(\text{Lille}, \text{Paris}, 150), \text{fly}(\text{Paris}, \text{Montreal}, 800), \text{bus}(\text{Montreal}, \text{Ottawa}, 50)))$

$B = (i, (\text{train}(\text{Lille}, \text{Paris}, 150), \text{fly}(\text{Paris}, \text{Montreal}, 800), \text{bus}(\text{Montreal}, \text{Ottawa}, 100)))$

$C = (i, (\text{train}(\text{Lille}, \text{Bruxelles}, 100), \text{fly}(\text{Bruxelles}, \text{Montreal}, 650), \text{bus}(\text{Montreal}, \text{Ottawa}, 50)))$

$D = (i, (\text{train}(\text{Lille}, \text{Bruxelles}, 100), \text{fly}(\text{Bruxelles}, \text{Montreal}, 650), \text{bus}(\text{Montreal}, \text{Ottawa}, 100)))$

The strength of A and B is v_7 and the strength of C and D is v_8 . Therefore, A and B defeats C and D but C and D does not defeat A and B . The set $\{A, B\}$ is subjectively acceptable for AC_1 with respect to $\mathcal{P}(C)$.

In the next section, we shift from the zero-agent notion of acceptability to the one-agent notion of conviction in order to take into account not only plans shared by different audiences but also plans exchanged by agents.

4 Model of agents

In multi-agent setting it is natural to assume that the agents do not all have exactly the same beliefs and capacities. Since the competences of agents can be common, complementary or contradictory, agents exchange their plans and argue. For this purpose, our agents individually value the perceived commitments with respect to the estimated reputation of the agents from whom the plan is obtained. The agent a_i , which has a personal competence C_i , a set of personal values V_i and a priority relation \ll_i , record in the commitment store CS_j^i which contains the rules taken before or at time t by the agent a_j . Moreover, the agent a_i individually values the reputation of their interlocutors with her reputation relation \prec_i , i.e. a strict complete ordering relation on \cup_A . The rules in the commitment store CS_j^i relate to the reputation value v_j^i .

The personal competences of agents are not necessarily disjoint. We call **common competence** the set of rules explicitly shared by the agents: $C_{\Omega_A} \subseteq \cap_{a_i \in \cup_A} C_i$. Similarly, we call **common values** the values

explicitly shared by the agents: $V_{\Omega_A} \subseteq \bigcap_{a_i \in \mathcal{U}_A} V_i$. The common rules relate to the common values. For every $c \in \mathcal{C}_{\Omega_A}$, $\text{promote}(c) = v \in V_{\Omega_A}$. The personal rules can be complementary or contradictory. We call **joint competence** the set of rules distributed in the system: $\mathcal{C}_{\mathcal{U}_A} = \bigcup_{a_i \in \mathcal{U}_A} \mathcal{C}_i$. The agent own rules relate to the agent own values. For every $c \in \mathcal{C}_i - \mathcal{C}_{\Omega_A}$, $\text{promote}_i(c) = v \in V_i - V_{\Omega_A}$.

Reputation is a social concept that links an agent to her interlocutors. It is also a leveled relation [9]. The individuated reputation relations, which are transitive, irreflexive, asymmetric, and complete relations on \mathcal{U}_A , preserve these properties. $a_j \prec_i a_k$ denotes that an agent a_i trusts an agent a_k more than another agent a_j . In order to take into account the rules notified in the commitment stores, each agent is associated with the following **extended competence**: $AC_i^* = \langle \mathcal{C}_i^*, V_i^*, \text{promote}_i^*, \ll_i^* \rangle$, which is a value-based competence where:

- $\mathcal{C}_i^* = \mathcal{C}_i \cup [\bigcup_{j \neq i} \text{CS}_j^i]$ is the extended personal competence of the agent composed of the personal competence and the set of perceived commitments;
- $V_i^* = V_i \cup [\bigcup_{j \neq i} \{v_j^i\}]$ is the extended set of personal values of the agent composed of the set of personal values and the reputation values associated with her interlocutors;
- $\text{promote}_i^* : \mathcal{C}_i^* \rightarrow V_i^*$ is the extension of the function promote_i which maps from the rules in the extended personal competences to the extended set of personal values. On the one hand, the personal rules relate to the personal values. On other hand, the rules in the commitment store CS_j^i relate to the reputation value v_j^i ;
- \ll_i^* is the extended priority relation of the agent, *i.e.* an ordering relation on V_i^* .

Since the deliberation is a collaborative social process, the agents share common rules (common goal, common initial situation, common sense, ...) of prime importance. That is the reason why the common values have priority over the other values. Since the agents argue, they estimate themselves more authoritative than their interlocutors. That is the reason why the personal values have priority over the reputation values. In other words, the extended priority relation of the agent is constrained as follows: $\forall a_j \in \mathcal{U}_A \forall v_\omega \in V_{\Omega_A} \forall v \in V_i - V_{\Omega_A} (v_j^i \ll_i^* v \ll_i^* v_\omega)$. We can easily demonstrate that the extended priority relation is a strict complete ordering relation. The **agent a_i is convinced by the rule c** iff c is the conclusion of an acceptable argument by the audience a_i with respect to $\mathcal{P}(\mathcal{C}_i^*)$.

The agents utter messages to exchange their rules. The syntax of messages is in conformance with the common **communication language**, $\mathcal{CL}_{\mathcal{U}}$. A message $M_k = \langle S_k, H_k, A_k \rangle \in \mathcal{CL}_{\mathcal{U}}$ has an identifier M_k . It is uttered by a speaker ($S_k = \text{speaker}(M_k)$) and addressed to a hearer ($H_k = \text{hearer}(M_k)$) $A_k = \text{act}(M_k)$ is the speech act of the message. It is composed of a locution and a content. The locution is one of the following: question, assert, unknow, concede, challenge, withdraw. The content, also called **hypothesis**, is a rule or a set of rules. As in [8] the speech acts have an argumentative and public semantics. We have presented here a model of agents who exchange hypothesis and argue. In the next section, we bound a formal area to shift from the one-agent notion of conviction to the n-agent notion of provability.

5 Dialectical system

When a set of social and autonomous agents deliberate, they reply each other in order to reach the goal of the interaction. Since we want to warrant that a practical agreement will be reached, we need to bound a formal area, called dialectical system, in which agents play and arbitrate. Moreover, we add a third agent who arbitrates in accordance with the estimated authority of the players, the uttered plans and her own rules and priorities.

During exchanges, the speech acts are not isolated but they respond each other. The syntax of moves is in conformance with the common **moves language**. A move $\text{move}_k = \langle M_k, R_k, P_k \rangle \in \mathcal{ML}_{\mathcal{U}}$ has an identifier move_k . It contains a message M_k as defined before. The moves are messages with some attributes to control the sequence. $R_k = \text{reply}(\text{move}_k)$ is the identifier of the move to which move_k responds. A move (move_k) is either an initial move ($\text{reply}(\text{move}_k) = \text{nil}$) or a replying move ($\text{reply}(\text{move}_k) \neq \text{nil}$). $P_k = \text{protocol}(\text{move}_k)$ is the name of the protocol which is used. A dialectical system is composed of three agents. In this formal area, two agents play moves in front of a third agent to check that the initial hypothesis, *i.e.* the topic, can be reached.

Definition 7 Let $AC_{\Omega_A} = \langle C_{\Omega_A}, V_{\Omega_A}, promote_{\Omega_A} \rangle$ be a common value-based competence and g_0 a rule (the specification of a goal). The **dialectical system** on the topic g_0 is a 7-tuple $DS_{\Omega_M}(c_0, AC_{\Omega_A}) = \langle N, judge, H, T, protocol, Z, (u_p)_{p \in N} \rangle$ where:

- $N = \{init, part\} \subset \mathcal{U}_A$ is a set of two agents called players: the initiator and the partner;
- $judge \in \mathcal{U}_A$ is a third agent with a personal competence which contains the common competence ($C_{judge} \supseteq C_{\Omega_A}$);
- $\Omega_M \subseteq \mathcal{ML}_{\mathcal{U}}$ is a set of well-formed moves;
- H is the set of histories, i.e. the sequences of well-formed moves;
- $T : H \rightarrow N$ is the turn-taking function determining the speaker of a move;
- $protocol : H \rightarrow \Omega_M$ is the function determining the moves which are allowed or not to expand an history;
- Z is the set of deliberation, i.e. the terminal histories which consist of maximally long histories.

In order to be well-formed, the initial move is a question about the topic from the judge to the initiator and the partner and a replying move from a player references an earlier move uttered by one of the other players. Obviously, all moves should contain the same value for the protocol parameter. The judge computes the final practical agreement. At the history h , the judge is associated to the extended competence $AC_{judge}^*(h) = \langle C_{judge}^*(h), V_{judge}^*, promote_{judge}^*, \ll_{judge}^* \rangle$ where:

- the extended personal competence contains the common competence and the commitments of players: $C_{judge}^*(h) \supseteq C_{\Omega_A} \cup CS_{init}^{judge}(h) \cup CS_{part}^{judge}(h)$;
- the extended set of values is composed of the common values and the reputation values of the two players: $V_{judge}^* = V_{\Omega_A} \cup \{v_{init}^{judge}, v_{part}^{judge}\}$.

The set of convincing plans for the judge depends on the history, the reputation of players and her own rules and priorities. The reputation relation of the judge corresponds to the global social order. s_1 is **provable at the history** h (written $provable^h(s_1)$) if $s_1, \epsilon \vdash g_0$ and the judge is convinced by s_1 at the history h . The deliberation computes the n-agent notion of provability. In this way, the arbitrage of the judge depends on the plans exchanged and the estimated authority of the players and her own rules and priorities.

We have bound here the area in which the deliberations take place to define the n-agent notion of provability. In order to deliberate, we formalize in the next section a protocol.

6 Protocol

When a set of social and autonomous agents deliberate, they collaborate to confront their convictions. In this section we illustrate our dialectical system with a protocol where agents reach a practical agreement by verbal means [1]. In this paper, we formalize this protocol. The protocol consists of the sequence rules represented in the table 2. Each rule specifies the authorized replying moves. For example, the rule of “Assertion/Refutation” (written $sr_{A/R}$) specifies the authorized moves replying to the previous assertion ($assert(H)$). The speech acts resist or surrender to the previous one. Contrary to the resisting acts, the surrendering acts close the deliberation. A concession ($concede(H)$) surrenders to the previous assertion. A challenge ($challenge(h)$) and a refutation ($assert(h_2)$) resist to the previous assertion. As previously said, the speech acts $question(h)$, $challenge(h)$, $unknow(h)$, and $withdraw(h)$ are used to manage the sequence of moves. On one side, a question initiates the deliberation. On the other side, a plea of ignorance and a withdrawal close the deliberation. A challenge is a request for a plan.

In order to confront her conviction with the partner, an agent initiates a deliberation. If the partner has no plan for the topic, she pleads ignorance and closes the deliberation. If the players have the same convictions, the judge is convinced and the deliberation closes. Otherwise, the goal of the deliberation is to reach a practical agreement by verbal means. The following example illustrates such a protocol.

Sequences rules	Speech acts	Resisting replies	Surrendering replies
$sr_{Q/A}$	$question(g_0)$	$assert(s_1), s_1, \epsilon \vdash g_0$	$unknow(g_0)$
$sr_{A/R}$	$assert(H)$	$challenge(h), h \in H$ $assert(h_2), \exists h_1 \in H h_1 \perp h_2$	$concede(H)$
$sr_{C/A}$	$challenge(h)$	$assert(H), H, \epsilon \vdash h$	$withdraw(h)$
sr_T	$unknow(H)$	\emptyset	\emptyset
	$concede(H)$	\emptyset	\emptyset
	$withdraw(H)$	\emptyset	\emptyset

Table 2: Set of speech acts and the potential answers.

$\mathcal{C}_1^* - \mathcal{C}_{\Omega_A}$		\mathcal{C}_{Ω_A}	$\mathcal{C}_2^* - \mathcal{C}_{\Omega_A}$	
\mathcal{C}_1	CS_2^1	$c_{11}, c_{12}, c_{21}, c_{22}, c_{23}$	CS_1^2	\mathcal{C}_2
premise(B)	\emptyset	0	\emptyset	premise(C)
\leftarrow buyer question($g(\text{price})$) \rightarrow				
premise(B)	\emptyset	1	\emptyset	premise(C)
\rightarrow a ₁ assert($g(1000)$) \rightarrow				
premise(B)	\emptyset	2	$g(1050)$	premise(C)
\leftarrow a ₂ assert($g(800)$) \leftarrow				
premise(B)	total(a _x , 800)	3	$g(1000)$	premise(C)
\leftarrow buyer challenge($g(800)$) \leftarrow				
premise(B)	total(a _x , 800)	4	$g(1000)$	premise(C)
\leftarrow a ₂ assert(premise(C)) \leftarrow				
premise(B)	premise(C)	5	total(a _x , 1000)	premise(C)
\leftarrow buyer challenge($g(1050)$) \leftarrow				
premise(B)	premise(C)	6	\emptyset	premise(C)
\rightarrow a ₁ assert(premise(B)) \rightarrow				
premise(B)	premise(C)	7	premise(B)	premise(C)
\leftarrow buyer concede(premise(A)) \rightarrow				
premise(B)	premise(C)	8	premise(B)	premise(C)

Table 3: Deliberation to reach an agreement

Example 2 Let us consider a deliberation between two services providers (a_1 and a_2) in front a buyer who judges. The value-based competence of the agents a_1 (resp. a_2) is composed of the common competence and the rules in the premise(B)) (resp. the rules in the premise(C)). The value-based competence of the buyer is composed of the common competence and the rule $g(< 1050)$ The commitments stores result from the sequence of moves (cf table 3). The arbitrage of the buyer depends on the advanced plans, the estimated authority of the players and her personal rule total(buyer, < 1050). At the end of the dialogue, the buyer composes the services and is convinced by a plan for transportation from Lille to Ottawa which costs less than 1050 euros (the argument A).

We have formalized here a protocol to reach a practical agreement. Since this paper extends [8], we can warrant as in [8] that the dialogue are finite and leads to an agreement.

7 Related works

Classically, argumentation has been mainly concerned with theoretical reasoning to check beliefs veracity [6, 3, 4]. A coherent framework has been proposed in [8] to reconcile, combine and extend these technics. In this paper, our contribution, like other recent works [12, 13, 14], is concerned with practical reasoning with our own instantiation of the *abstract* argumentation framework of Dung [6] (cf section 3).

On one hand, Amgoud [12] has presented an argumentation framework for generating argumentative plans from a given set of beliefs, goals and planning rules. This work was later was extended in [13] with argumentation frameworks that generate the goals themselves from beliefs. In [11], the generation of goals

are more general. On the other hand, Hulstijn and van der Torre [14] propose argumentative plans which contains only goals in the conclusions. By contrast with Amgoud's and Hulstijn's frameworks which focus a generic mechanism allowing an agent to compose her beliefs, goals and plans for generating consistent plans or consistent goals, we focus in this paper on a dialogical mechanism between software agents to jointly elaborate common plans, reason, exchange and compose them (cf sections 4, 5, and 6).

8 Conclusions

We have proposed in this paper a framework for inter-agents dialogue on actions, which formalize a deliberative process. This framework bounds a dialectical system in which argumentative agents arbitrate and play to reach a practical agreement. For this purpose, we have proposed an argumentation-based reasoning to manage the conflicts between plans having different strengths for different agents. Moreover, we have proposed a model of agents which justify the plans to which they commit and take into account the plans of their interlocutors. In the scope of our dialectical system, a third agent is responsible of the final decision outcome which is taken according to the authority of the players, the uttered plans and her own rules and priorities. We have illustrated this paper with a services composition.

References

- [1] D. Walton and E. Krabbe. *Commitment in Dialogue*. SUNY Press, 1995.
- [2] Perelman C. and Olbrechts-Tyteca L. *Traité de l'Argumentation - La Nouvelle Rhétorique*. Presses Universitaires de France, 1958.
- [3] Leila Amgoud and Claudette Cayrol. A reasoning model based on the production of acceptable arguments. *Annals of Maths and AI*, 34(1-3):197–215, 2002.
- [4] T.J.M Bench-Capon. Value based argumentation frameworks. In *Proc. of NMR*, pages 444–453, 2002.
- [5] Trevor Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.*, 3(13):429–448, 2003.
- [6] Phan Minh Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357, 1995.
- [7] Antonis C. Kakas and Pavlos Moraitis. Argumentative agent deliberation, roles and context. In Jurgen Dix, João Alexandre Leite, and Ken Satoh, editors, *Electronic Notes in Theoretical Computer Science*, volume 70. Elsevier, 2002.
- [8] Maxime Morge. Collective decision making process to compose divergent interests and perspectives. In *Proc. of BNAI*, 2005.
- [9] C. Castelfranchi and R. Falcone. Principles of trust in mas: Cognitive anatomy, social importance, and quantification. In *Proc. of ICMAS'98*, pages 72–79, 1998.
- [10] L. Amgoud and S. Kaci. On the generation of bipolar goals in argumentation-based negotiation. In *ArgMAS: State of the art survey*, LNAI 3366, 192-207, 2005.
- [11] I. Rahwan and L. Amgoud. An argumentation-based approach for practical reasoning. In *Proc. of AAMAS*, 2006.
- [12] L Amgoud. A formal framework for handling conflicting desires. In *Proc. of ESSQARU*, 2003, LNAI 2711, p. 552-563.
- [13] L Amgoud and C Cayrol On the use of ATMS for handling conflicting desires. In *Proc. of KR*, 2004.
- [14] J. Hulstijn and L. van der Torre. Combining goal generation and planning in an argumentation framework. In *Proc. of Workshop on Argument, Dialogue and Decision, at NMR*, Whistler, Canada, June 2004.

A Taxonomy of Human – Agent Team Collaborations

Martijn Neef

TNO Defence, Safety and Security

Abstract

Future command teams will be heavily supported by artificial actors. This paper introduces a taxonomy of collaboration types in human – agent teams. Using two classifying dimensions, coordination type and collaboration type, eight different classes of human – agent collaborations transpire. These classes might aid designers in pinpointing the socio-technical design issues associated with these kinds of hybrid organizations.

1 Introduction

In many domains, human operators are experiencing increasing difficulties in accomplishing their objectives because of rising system complexity and higher performance demands. Modern command teams, such as those present in civil control centers and on military platforms, need to deal with more information, more stakeholders and more complex systems than they used to, and perform under severe time constraints. To counter these developments and to uphold performance in future scenarios, new levels and forms of crew support will need to be developed.

With the long-awaited advent of concrete results in the intelligent agent domain and the overall advances in computational science, many previously considered far-fetched visions on symbiotic human-machine cooperation have become closer to reality. The insertion of intelligent agents as active participants in an organization will significantly alter team dynamics in general, and will transform standard command teams into adaptive human – agent teams. In order to benefit from the enhanced capabilities that these new types of teams offer, we need collaboration schemes between human and artificial actors that harmonize with their respective competences and demands.

This paper explores various types of collaborations in human – agent teams, and introduces a concept taxonomy of human – agent team collaboration types.

2 Human – Agent Teams

Most research on multi-agent systems has focused on agents themselves instead of agents as parts of a human organization. As we are going towards more advanced types of agent-supported support, we need new paradigms for interactions between human and artificial actors. Humans and agents will need to cooperate in a dynamic and harmonious manner, and to realize this we will need new perspectives on how to design support systems. Instead of just focussing on the technological challenges of agent-based support (e.g. agent design), or the human aspects of working with artificial actors (e.g. human factors research), we also need to focus our attention to the specific aspects of the interplay between them, and how to forge effective collaborations [10, 11].

What we are actually facing is a new type of team, the *human – agent team* [1]. A team is a group of two or more people, that exists for a limited amount of time and whose members work have specific roles and work together towards a common objective in a dynamic, interactive and adaptive fashion [7, 10]. Furthermore, in a team, there is some organizational structure among the team members, and task completion requires the dynamic interchange of information, the coordination of task activities, and constant adjustment to task demands. Ideally, a human – agent team should fulfil these characteristics as

well. However, when dealing with artificial team members, several other critical aspects play a role. For instance, the artificial agents must be directable by others, transparent in their behaviour and must be able to make their state and intentions obvious to their team-mates [2]. From an agent design perspective, as well as from a team-organization perspective these are important aspects to consider.

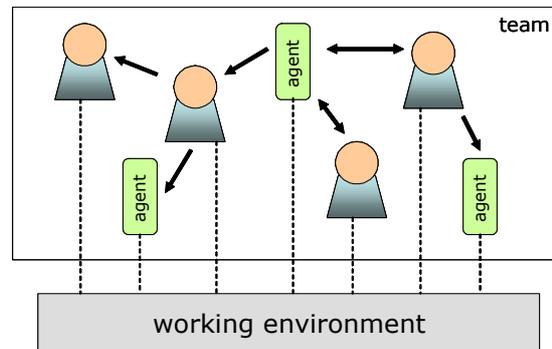


Figure 1: An example of a human – agent team. Both human and artificial team members are in control of other team members, and jointly they perform tasks in their working environment. The arrows denote control direction.

One way of looking at human – agent teams is by examining their organizational structure. One of the most obvious distinctions one can make is that of relationship cardinality: the cardinality of the relationship between the human and the artificial actors in a team. This is an important dimension for classification, because cardinality leads to demands with respect to sociality capabilities. A ‘single human – single agent’ collaboration requires far less social capabilities on the agent part than a ‘multiple human – multiple agents’ collaboration. In the former setup, the agent would only have to have knowledge of a single human user, whereas in the latter situation, agent would need to have knowledge of more than one user.

Given an organization consisting of human and synthetic actors, there are many ways to divide the labour between humans and machine. Designers must carefully examine the functional requirements and assess the implications on technical feasibility of a certain task allocation strategy. Certain tasks may too hard to automate well. Usually, engineers take on a very divisional stance when it comes to dividing tasks between humans and machines, and designing the interaction between them. Well-known exemplary allocation models are those of Sheridan [9] and Parasuraman [6]. However, division of labour is not just about ‘allocation’ – it is also about team member interaction and mutual support. For example, in a true symbiotic system, machines would support humans in their decision-making and cooperate on hard tasks and, vice-versa, humans would come to the aid of systems when they encounter tasks that require knowledge beyond their reach [3]. Therefore, an important questions is *who* is in charge of allocating tasks, and *when* should task allocation occur?

Closely related to task allocation is the issue of autonomy. Most research endeavours in computer science are about giving automation more autonomous capabilities. However, in operational settings, there is a distinct requirement to harness and control autonomy of systems. Systems may, or may not be allowed to make mission-critical decisions or perform potentially life-threatening activities without proper authorizations. This subject is of utmost interest to systems designers, because the greatest part of human – machine interaction is about controlling functionality [1]. If the system has a certain degree of autonomy, then interaction should be about controlling autonomy. Agent autonomy is a complex issue to tackle, and it is at the heart of proper human – agent team design. If autonomy is not properly controlled, any attempt for implementing agents in an operational setting will most likely fail. In that sense, it should be regarded as a primary driving force for human – agent systems.

3 Taxonomy

Teams are complex objects to describe or model, and human – agent teams even more because of their additional dependence on technology. This section introduces the two dimensions that give us a taxonomy of human – agent team collaborations. This taxonomy might help us pinpoint socio-technical design issues associated with the kind of hybrid organizations we are after. The dimensions we use are the *coordination type* and *collaboration type*. We believe that these two dimensions are enough to describe the principal types of human – agent teams. Yanco and Drury have used a similar approach to build a taxonomy of all forms of human – robot interactions [13], although they limited themselves largely to a structural taxonomy.

Coordination entails all activities that are required to manage these dependencies. For our purposes, we use the three elemental types of coordination mechanisms from Mintzberg's work on coordination in organizational structures [4]: *standardized coordination* (standard procedures have been established for actors to follow), coordination by *direct supervision* (one actor has some degree of control over others and can control information, resources, and behaviour) and *joint coordination* (actors communicate and collaborate to coordinate the activities).

An often-used distinction in the field of team research is that of taskwork versus teamwork [7]. Taskwork refers to activities that team members carry out to achieve a set goal, such as performing their tasks and handing over task outcomes to other team members. In this research we use the phrase *taskwork-oriented collaboration* to refer to all interactions in a human – agent team that concern and control task execution. It does not include strategic team decision-making mechanisms and deliberate coordination procedures. Teamwork has more to do with the activities that surround and control the task executions in a team, such as task allocation, coordination, resolving issues and inter-team communication. We view teamwork primarily from the *command and control* point of view. *Teamwork-oriented collaboration* is about who has command skills in human – agent teams, and tasks that are associated with being in command.

These dimensions lead to a taxonomy that consists of eight specific types of collaboration types human – agent teams (Figure 2).

3.1 Teamwork oriented collaborations

Under *static division of labour*, there is a clear and unmistakable partition of activities between humans and agents, which does not change over time or under certain circumstances. Both agents and humans have a specific tasking order. The agents always act as they are expected to. Consequently, there are no conflicts with respect to autonomy and task performance is, under anticipated circumstances, reliable.

Adjustable automation refers to the capability of the human operator to alter the level of automation to let it better fit his needs. For example, one may choose to change the system's behaviour from just providing an alternative solution to letting it actually implement that alternative. Many concrete applications that display this type of collaboration can be found in the aerospace domain, even though most do not rely on agent technology [1].

In a *mixed initiative setting*, the initiative to take on a task is determined through negotiation between participating actors – team-members, whether they are software agents or humans, divide tasks at the time they need to be performed. There is no predisposed division of labour and the coordination is adaptive and ad-hoc. This would allow for an extensive degree of adaptivity, but would most likely not be appropriate in fast-paced information processing environments. Coordination through negotiation takes time and such a setup is known to be time consuming [5].

Adaptive automation is the situation where the agent has the capability to alter its own level of automation in response to the performance and state of the human operator. It also includes the ability for the human user to change the behaviour of the agent. Under adaptive automation, the division of labour between human operator and the agent is flexible rather than fixed. The objective is that human and agent form a dyadic team, jointly working towards a common objective adaptively helping each other out. Theoretically, this would be a desirable collaboration, but in practice, it is hard to make it work. Human users quickly feel like they lose control over their artificial counterparts, and it is difficult to define adaptive behaviours for agents that agree with the way humans work [12].

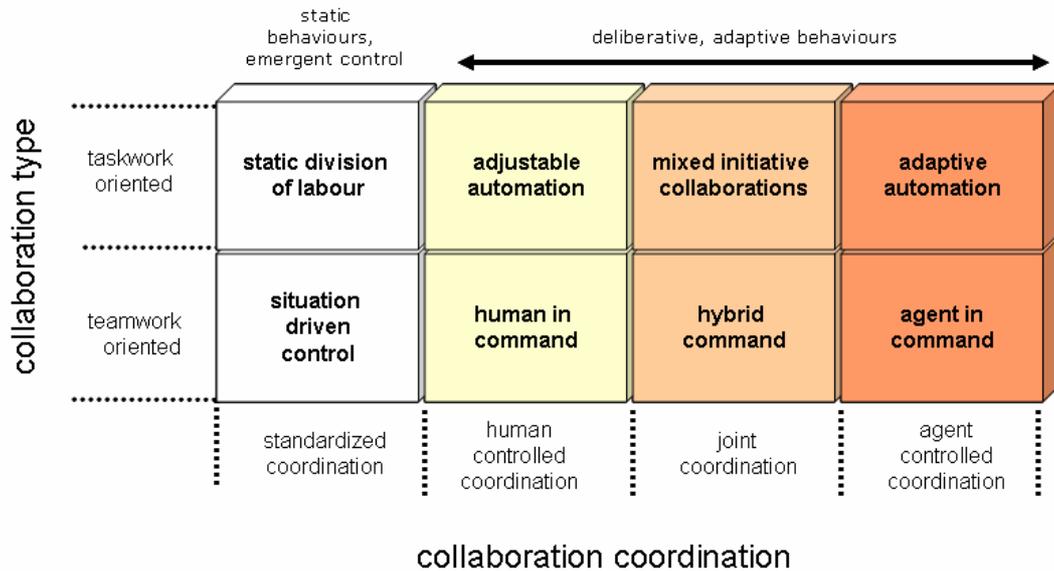


Figure 2: A taxonomy of collaboration types in human – agent teams.

3.2 Taskwork oriented collaborations

Under *situation driven control*, there are not a fixed division of tasks in the team between humans and agents. This type is in contrast to the ‘static division of labour’ type we discussed earlier. A crucial aspect in this setup is that there is no designed facility for adaptive delegation of tasks: every entity does what he or it is supposed to do, based on the state of the environment and based on rules that prescribed the reactive behaviour. In this type of settings, there is *emergent control* – coordinated actions are a result of multiple actors doing their job. There is about standardization on procedures, not on tasks, as was the case in the static division of labour type of collaboration. One could think of reactive service-oriented systems as an example of this kind of collaboration. Agents only come into play when they are triggered to do so.

Human in command task delegation is the team-oriented variant of ‘adjustable automation’. In this type, the human operators are in primary command and are able to delegate tasks to other (artificial) actors in order to alleviate their workload. The process of delegating tasks should be such that the agent team members coordinate delegate tasks amongst themselves without involvement of the human actors. In order to prevent loss of control, the operator that delegated the task will need to be informed of the progress in some way. A good example is the ‘work-centered agents’ approach of Scott [8], where the operator specifically creates and instructs an agent to perform a certain task

In an *agent in command* arrangement, an artificial agent manages the task allocation process. The agent has an awareness of the state and competences of every team members and allocates tasks accordingly. This service implies that the agent has team awareness and task knowledge. Depending on the precise implement, the act of command may or not be a deliberate act on the part of the agent. This type of arrangement agrees with recent, agent-oriented work in the ‘computer-supported collaborative work’ domain.

Hybrid command refers to team settings in which command is in the hands of both a human and an artificial actor. In this setting, we view the task of commanding a team as a deliberate act. The commanders have a certain awareness of the situation, are driven by a desire to achieve an objective, and are able to communicate this to others. Of course, in the hands of a human commander, commanding is always a deliberate act. This is not so the case for artificial agents. In hybrid command, we see human and artificial commanders teaming up and jointly coordinate their subordinates. Therefore, great emphasis needs to be placed on harmonizing communication means between the commanding actors – they need to be able to share their views on the world and the state of their respective teams.

4 Discussion

This paper introduced a taxonomy of human – agent collaborations in team settings. Based on classifications on two dimensions, *collaboration coordination* and *collaboration type*, eight different classes of human – agent collaborations were identified. The taxonomy was intended to stimulate researchers in human – agent interaction to investigate the various new types of interactions between humans and autonomous systems that emerge from technological advances.

The taxonomy may help to gain insight in the socio-technical challenges in development agent-based support for critical command environments. However, this issue goes well beyond the technological challenges of creating problem-solving agents. The very dynamics of such human – agent teams form a research topic on their own. No single type of human – agent team collaboration will be suitable for any situation. For instance, standardized coordination forms (such as the *static division of labour* form) are not well suited for unfamiliar situations, but fare very well under time-pressed and routine environments. Each form has specific aspects that make it more or less suited for a certain circumstance. In a human – agent team, the collaboration form does not need to be persistent. Adaptivity is generally considered to be a key aspect of good teamwork, and a human – agent teams should be able to switch collaboration types with changing conditions. Teams should select that form of collaboration that best suits the situation. Likewise, multiple collaborations can co-exist in a single team at the same time, and most likely will. Effective human – agent teams will be adaptive in that sense, and all actors, both human and artificial, must have the capacity to cope with dynamic collaboration modes.

This research, the taxonomy, is not finished in any way and is very much open to discussion. The specifics of the types are debatable, and there is a lot of related research that seems hard to match to the taxonomy. However, for every type we identified, matching examples can be found in research and practice. We believe that the approach taken in this paper is a good starting point to build further research in human – agent team design. The paper should be read as a discussion article and should lead to more detailed, multidisciplinary exploration of the subject and, lead to experimental settings where the taxonomy can be really tested on its worth.

5 Acknowledgements

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024. The ICIS project is hosted by the DECIS Lab, the open research partnership of Thales Nederland, the Delft University of Technology, the University of Amsterdam and the Netherlands Organization for Applied Scientific Research TNO.

References

- [1] Bradshaw, J. M., Sierhuis, M., Acquisti, A., Feltovich, P., Hoffman, R., Jeffers, R., Prescott, D., Suri, N., Uszok, A., and Van Hoof, R. (2003), Adjustable autonomy and human - agent teamwork in practice: An interim report on space applications. In H. Hexmoor, R. Falcone, and C. Castelfranchi (Eds.), *Agent Autonomy*, Kluwer, Dordrecht, The Netherlands, pp. 243-280.
- [2] Klein, G., Woods, D.D., Bradshaw, J.M., Hoffman, R.R. and Feltovich, P.J. (2004), Ten Challenges for Making Automation a "Team Player" in Joint Human-Agent Activity. *IEEE Intelligent Systems*, 19(6), pp. 91-95.
- [3] Lewis, M., Sycara, K., and Payne, T. (2003), Agent Roles in Human Teams. *Notes of the AAMAS 2003 Workshop on Humans and Multiagent Systems*, Melbourne, Australia.
- [4] Mintzberg, H. (1979), *The Structuring of Organisations*. Prentice-Hall, Englewood Cliffs, USA.
- [5] Norman, D.A. (1990), The 'problem' of automation: Inappropriate feedback and interaction, not 'over-automation. *Philosophical Transactions of the Royal Society of London*, B 327, pp. 585-593.

- [6] Parasuraman, R., T. B. Sheridan, et al. (2000), A Model for Types and Levels of Human Interaction with Automation. *IEEE Transactions on Systems, Man, and Cybernetics*, 30(3), pp. 286-297.
- [7] Salas, E., Dickinson, T. L., Converse, S. A., and Tannenbaum, S. I. (1992), Toward an understanding of team performance and training. In R. W. Swezey & E. Salas (Eds.), *Teams: Their training and performance*. Ablex, Norwood, USA, pp. 3-29.
- [8] Scott, R., Roth, E.M., Deutsch, S.E., Malchiodi, E., Kazmierczak, T.E., Eggleston, R.G., Kuper, S.R. and Whitaker, R.D. (2005), Work-Centered Support Systems: A Human-Centered Approach to Intelligent System Design. *IEEE Intelligent Systems*, 20(2), pp. 73- 81.
- [9] Sheridan, T. B., and Verplank, W. (1978), *Human and Computer Control of Undersea Teleoperators*. Technical Report, Man-Machine Systems Laboratory, Department of Mechanical Engineering, MIT, Cambridge, USA.
- [10] Sycara, K. and Lewis, M. (2004), Integrating intelligent agents into human teams. In E. Salas and S. Fiore (Eds.), *Team Cognition: Understanding the Factors that Drive Process and Performance*, American Psychological Association, Washington DC, USA.
- [11] Woods, D. D. (1986), Cognitive technologies: The Design of Joint Human-Machine Cognitive Systems. *AI Magazine*, 6(4), pp. 86-92.
- [12] Hollnagel, E. and Woods, D. D. (2005), *Joint Cognitive systems: Foundations of Cognitive Systems Engineering*. Taylor & Francis, Boca Raton, USA.
- [13] Yanco, H. A. and Drury, J. L. (2004), Classifying Human-Robot Interaction: An Updated Taxonomy. *Proceedings of the 2004 IEEE International Conference on Systems, Man and Cybernetics*, pp. 2841 – 2846.

Hierarchical Reinforcement Learning with Deictic Representation in a Computer Game

Marc Ponsen Pieter Spronck Karl Tuyls

MICC-IKAT, Universiteit Maastricht, PO Box 616, 6200 MD
Maastricht, The Netherlands

Abstract

Computer games are challenging test beds for machine learning research. Without applying abstraction and generalization techniques, many traditional machine learning techniques, such as reinforcement learning, will fail to learn efficiently. In this paper we examine extensions of reinforcement learning that scale to the complexity of computer games. In particular we look at hierarchical reinforcement learning applied to a learning task in a real time-strategy computer game. Moreover, we employ a deictic state representation that reduces the complexity compared to a propositional representation and allows the adaptive agent to learn a generalized policy, i.e., it is capable of transferring knowledge to unseen task instances. We found that hierarchical reinforcement learning significantly outperforms flat reinforcement learning for our task.

1 Introduction

In reinforcement learning (RL) problems, an agent interacts with its environment and iteratively learns a policy. Policies are usually represented in a tabular format, where each cell includes a state or state-action value representing, respectively, the desirability of being in a state or the desirability of choosing an action in a state. In previous research, this approach has proven to be feasible in 'toy' domains with limited state and action spaces. In contrast, in more complex domains the number of states grows exponentially, resulting in an intractable learning problem. Modern computer games are typical examples of such complex domains. They present realistic models of real-world environments and offer a unique set of artificial intelligence challenges, such as dealing with huge state and action spaces and real-time decision making in stochastic and partially observable worlds.

Reducing the policy space through abstraction or applying generalization techniques is essential to enable efficient RL in computer games. In this paper we investigate temporal abstraction in RL, that is, we implement a modified version of Hierarchical Semi-Markov Q-learning (HSMQ) and empirically compare its effectiveness with flat Q-learning in a sub-domain of a real-time strategy game. Furthermore, we employ a deictic world state representation [7]. This allows the adaptive agent to generalize and transfer knowledge to unseen task instances.

The remainder of the paper is organized as follows. In Section 2 we will discuss related work. Section 3 describes reinforcement learning. Section 4 introduces

hierarchical reinforcement learning and the algorithm used in the experiments. In Section 5 we will introduce the task we propose to solve. Section 6 describes a flat and hierarchical representation of this problem, while Section 7 presents experimental settings and compares results. We conclude in Section 8.

2 Related Work

For an extensive overview of RL related work we refer to [13]. Several hierarchical RL techniques have been developed e.g., MAXQ [3], Options [13], HAMs [10], HASSLE [1] and HQ-learning [14]. A summary of work on hierarchical RL is given by [2]. In this section we specifically focus on RL research in computer games. Previous RL research in computer games either assumed appropriate abstractions and generalizations or addressed only very limited computer game scenarios.

For example, Spronck et al. [12] and Ponsen et al. [11] implemented a RL inspired technique called *dynamic scripting* in several computer games. They report good learning performances on a challenging task: learning to win computer games. However, the technique requires a considerably reduced state and action space (often temporally extended actions) to work efficiently. In contrast, in this research we allow agents to plan with actions that take exactly one time step.

Marthi et al. [9] applied hierarchical RL to scale to complex environments. They learned navigational policies for agents in a limited real-time strategy computer game domain. Their action space consisted of partial programs, essentially high-level pre-programmed behaviors with a number of *choice points* that were learned using Q-learning. Our work differs from theirs in that we use a different hierarchical RL technique.

Driessens [5] combined RL with regression algorithms to generalize in the policy space. He evaluated his relational RL approach in two computer games. Guestrin et al. [8] also learned generalized policies in a limited real-time strategy game domain by solving a relational Markov decision process. Similarly, we also allow the agent to learn a generalized policy by employing a deictic state representation.

3 Reinforcement Learning

Most RL research is based on the framework of Markov decision processes (MDPs). MDPs are sequential decision making problems for fully observable worlds with a Markovian transition model. MDPs can be defined by a tuple $(s_0, t, S, A, \delta, r)$. Starting in an initial state s_0 at each discrete time-step $t = 0, 1, 2, \dots$ an adaptive agent observes an environment state s_t contained in a finite set of states $S = \{s_1, s_2, \dots, s_n\}$, and executes an action a from a finite set $A = \{a_1, a_2, \dots, a_m\}$ of admissible actions. The agent receives an immediate reward $r : S \times A \rightarrow \mathbb{R}$, and moves to a new state s' depending on a probabilistic transition function $\delta := P(s'|s, a)$. The learning task in MDPs is to find a policy $\pi : S \rightarrow A$ for selecting actions with maximal expected (discounted) reward.

Temporal difference learning methods, and in particular Q-learning, are popular to solve MDPs because these require no model (i.e., transition and reward

functions are unknown) and can be applied online. These characteristics make Q-learning a suitable learning algorithm for computer games. Namely, obtaining correct and complete models, even for limited computer game scenarios, can be difficult. Furthermore, learning in computer games should preferably take place online. The *one-step* Q-learning update rule is denoted as:

$$Q(a, s) \rightarrow (1 - \alpha)Q(a, s) + \alpha \left[r + \gamma \max_{a'} Q(a', s') \right] \quad (1)$$

where α is the step-size parameter, and γ the discount-rate.

4 Hierarchical Reinforcement Learning

Hierarchical RL (HRL) is an intuitive and promising approach to scale up RL to more complex problems. In HRL, a complex task is decomposed into a set of simpler subtasks that can be solved independently. Each subtask in the hierarchy is modeled as a single MDP and allows appropriate state, action and reward abstractions to augment learning compared to a flat representation of the problem. Additionally, learning in a hierarchical setting can facilitate generalization, e.g., knowledge learned by a subtask can be transferred to other subtasks.

HRL relies on the theory of Semi-Markov decision processes (SMDPs). SMDPs differ from MDPs in that actions in SMDPs can last multiple time steps. Therefore, in SMDPs actions can either be primitive actions (taking exactly 1 time-step) or temporally extended actions. While the idea of applying HRL in complex domains such as computer games is appealing, few studies in this respect actually exist [2].

We adopted a HRL method similar to Hierarchical Semi-Markov Q-learning (HSMQ) described in [4]. HSMQ learns policies simultaneously for all non-primitive subtasks in the hierarchy, i.e., $Q(p, s, a)$ values are learned that denote the expected total reward of performing task p starting in state s , executing action a and then following the optimal policy thereafter. Subtasks in HSMQ include termination predicates. These partition the state space S into a set of active states and terminal states. Subtasks can only be invoked in states in which they are active, and subtasks terminate when the state transitions from an active to a terminal state. We added to the HSMQ algorithm described in [4] a pseudo-reward function [3] for each subtask. The pseudo-rewards tell how desirable each of the terminal states are for this subtask. Algorithm 1 outlines our HSMQ inspired algorithm. Q-values for primitive subtasks are updated with the one-step Q-learning update rule, while the Q-values for non-primitive subtasks are updated based on the reward r collected during execution of the subtask and a pseudo reward \hat{R} .

5 Reactive Navigation Task

Real-Time Strategy (RTS) games require players to control a civilization and use military force to defeat all opposing civilizations that are situated in a virtual battlefield in real time. In this study we focus on a single learning task in RTS games: we learn a policy for a worker unit in the Battle of Survival (BoS) game.

Algorithm 1: Modified version of the HSMQ algorithm. The update rule for non-primitive subtasks (line 13) differs from the original implementation.

```

1 Function HSMQ(state  $s$ , subtask  $p$ ) returns float;
2 Let  $Totalreward = 0$ ;
3 while ( $p$  is not terminated) do
4   Choose action  $a = \Pi(s)$ ;
5   if  $a$  is primitive then
6     Execute  $a$ , observe one-step reward  $r$  and result state  $s'$ ;
7   else if  $a$  is non-primitive subtask then
8      $r := \text{HSMQ}(s, a)$ , which invokes subtask  $a$  and returns the total reward
      received while  $a$  executed
9      $Totalreward = Totalreward + r$ ;
10  if  $a$  is primitive then
11     $Q(p, a, s) \rightarrow (1 - \alpha)Q(p, a, s) + \alpha \left[ r + \gamma \max_{a'} Q(p, a', s') \right]$ ;
12  else if  $a$  is non-primitive subtask then
13     $Q(p, a, s) \rightarrow (1 - \alpha)Q(p, a, s) + \alpha \left[ r + \hat{R} \right]$ ;
14 end
15 return  $Totalreward$ ;

```

BoS is a RTS game created with the open-source engine Stratagus. A worker unit should be capable of effective navigation and avoiding enemies. We captured these tasks in a simplified BoS scenario. This scenario takes place in a fully observable world that is 32 by 32 grid cells large and includes two units: a worker unit (the adaptive agent) and an enemy soldier. The adaptive agent has to move to a certain goal location. Once the agent reaches its goal, a new random goal is set. The enemy soldier randomly patrols the map and will shoot at the worker if it is in firing range. The scenario continues for a fixed time period or until the worker is destroyed by the enemy soldier.

Relevant properties for our task are the locations of the worker, soldier and goal. All three objects can be positioned in any of the 1024 different locations. A propositional format of the state space describes each state as a feature vector with attributes for each possible property of the environment, which amounts to 2^{30} different states. As such, a tabular representation of the value functions is too large to be feasible. Additionally, such encoding prevents any opportunity for generalization for the learning algorithm. A deictic state representation identifies objects relative to the agent. This reduces state space complexity and facilitates generalization. We will discuss the state features used for this task in Section 6.

The proposed task is complex for several reasons. First, the state space without any abstractions is enormous. Second, the game state is also modified by an enemy unit, whose random patrol behaviour complicates learning. Furthermore, each new task instance is generated randomly (i.e., random goal and enemy patrol behavior), so the worker has to learn a policy that generalizes over unseen task instances.

6 Solving the Reactive Navigation Task

We compare two different ways to solve the reactive navigation task, namely using flat RL and HRL. For a **flat representation** of our task, the deictic state representation can be defined as the Cartesian-product of the following four features: `Distance(enemy,s)`, `Distance(goal,s)`, `DirectionTo(enemy,s)` and `DirectionTo(goal,s)`. The function `Distance` returns a number between 1 and 8 or a string indicating that the object is more than 8 steps away in state s , while `DirectionTo` returns the relative direction to a given object in state s . Using 8 possible values for the `DirectionTo` function, namely the eight directions available on a compass rose, and 9 possible values for the `Distance` function, the total state space is drastically reduced from 2^{30} to a mere 5184 states. The size of the action space is 8, namely containing actions for moving in each of the eight compass directions. The scalar reward signal r in the flat representation should reflect the relative success of achieving the two concurrent sub-goals (i.e., moving towards the goal while avoiding the enemy). The environment returns a +10 reward whenever the agent is located on a goal location. In contrast, a negative reward of -10 is returned when the agent is being fired at by the enemy unit. Each primitive action always receives the usual reward of -1. An immediate concern is that both sub-goals are often in competition. Certainly we can consider situations where different actions are optimal for the two sub-goals, although the agent can only take one action at a time. An apparent solution to handle these two concurrent sub-goals is applying a hierarchical representation, which we discuss next.

In the **hierarchical representation**, illustrated in Figure 1, the original task is decomposed into two simpler subtasks that solve a single sub-goal independently. The *to goal* subtask is responsible for navigation to goal locations. Its state space includes the `Distance(goal,s)` and `DirectionTo(goal,s)` features. The *from enemy* subtask is responsible for evading the enemy unit. Its state space includes the `Distance(enemy,s)` and `DirectionTo(enemy,s)` features. The action spaces for both subtasks include the primitive actions for moving in all compass directions. The two subtasks are hierarchically combined in a higher-level *navigate* task. The state space of this task is represented by the `InRange(goal,s)` and `InRange(enemy,s)` features, and its action space consists of the two subtasks that can be invoked as if they were primitive actions. `InRange` is a function that returns *true* if the distance to an object is 8 or less in state s , and *false* otherwise.

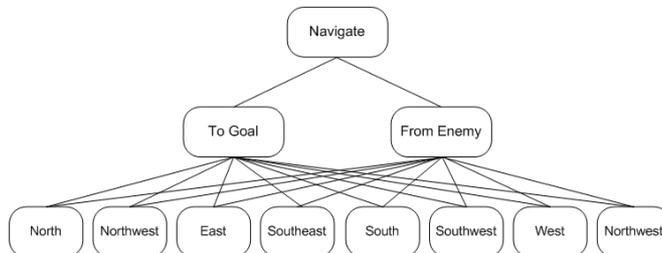


Figure 1: Hierarchical decomposition of the reactive navigation task.

These new features can in fact be defined in terms of already existing features, so we are not introducing any additional domain knowledge compared to the flat representation. The *to goal* and *from enemy* subtasks terminate at each state change on the root level, e.g., when the enemy (or goal) transitions from in range to out of range and vice versa. The pseudo-rewards for both subtasks were set to +100 whenever the subtask succeeded in achieving its task while it was active and 0 otherwise. The *navigate* task never terminates while the primitive subtasks always terminate after execution. The state spaces for the two subtasks are of size 72, and for *navigate* of size 4. Therefore, the state space complexity in the hierarchical representation is approximately 35 times less than with the flat representation presented previously. Additionally, in the hierarchical setting we are able to split the reward signal, one for each subtask, so they do not interfere. The *to goal* subtask rewards solely moving to the goal (i.e., only process the +10 reward when reaching a goal location). Similarly, the *from enemy* subtask only rewards evading the enemy. Based on these two reward signals and the pseudo-rewards, the root *navigate* task is responsible for choosing the most appropriate subtask. For example, let's assume the worker at a certain time decided to move to the goal and it took the agent 7 steps to reach it. The reward collected while the *to goal* subtask was active is -7 (reward of -1 for all primitive actions) and $+10$ (for reaching the goal location) resulting in a $+3$ total reward. Additionally, a pseudo-reward of $+100$ is received because *to goal* successfully terminated, resulting in a total reward of $+103$ that is propagated to the *navigate* subtask, that is used to update its Q-values. The Q-values for the *to goal* subtask are updated based on the immediate reward and estimated value of the successor state (see equation 1).

7 Experimental Results

We evaluated the performance of flat RL and HRL in the reactive navigation task. The step-size and discount-rate parameters were set to respectively 0.2 and 0.7. These values were determined during initial experiments. We emphasized longer exploration for the *to goal* and *from enemy* subtasks compared to the *navigate* task, since more Q-values require learning for these subtasks. Therefore, we used Boltzmann action selection with a relatively high (but decaying) temperature for the *to goal* and *from enemy* subtasks and ϵ -greedy action selection at the top level, with ϵ set to 0.1 [13].

A training session (when Q-values are adapted) lasted for 30 episodes. An episode terminated when the adaptive agent was destroyed or until a fixed time limit was reached. During training, random instances of the task were generated, i.e., random initial starting locations for the units, random goals and random enemy patrol behaviour. After a training session, we empirically validated the current policy on a test set consisting of 5 fixed task instances that were used throughout the entire experiment. These included fixed starting locations for all objects, fixed goals and fixed enemy patrol behaviour. We measured the performance of the policy by counting the number of goals achieved by the adaptive agent (i.e., the number of times the agent was successful at reaching the goal loca-

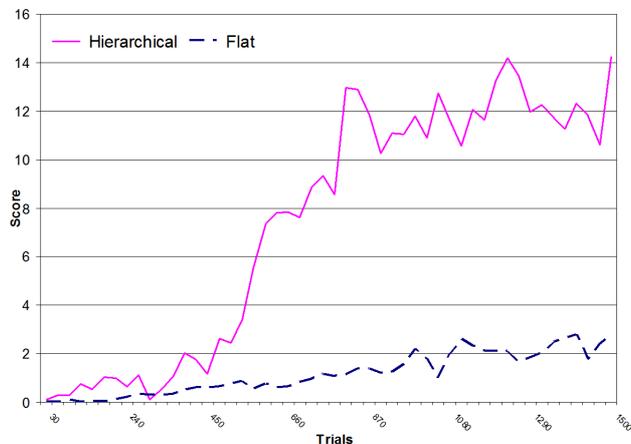


Figure 2: The average performance of Q-learning over 5 experiments in the reactive navigation task for both flat and HRL. The x-axis denotes the number of training trials and the y-axis denotes the average number of goals achieved by the agent for the tasks in the test set.

tion before it was destroyed or time ran out) by evaluating the greedy policy. We ended the experiment after 1500 training episodes. The experiment was repeated 5 times and the averaged results are shown in Figure 2.

From this figure we can conclude that HRL clearly outperforms flat RL: we see faster convergence to a suboptimal policy for HRL, while flat RL is still in the early stages of learning. HRL allowed more state abstractions, thus resulting in fewer Q-values that required learning. Furthermore, HRL is more suitable in dealing with concurrent and competing subtasks due to the split (and therefore more informative) reward signal. We expect that even after considerable learning with flat RL, HRL will still achieve a higher overall performance.

8 Conclusion

We investigated the application of RL to a challenging and complex learning task, namely navigation of a worker unit in an RTS game. We discovered that a deictic representation of the state space allows an adequate reduction of the number of states to make the learning task feasible. Additionally, a generalized policy was learned that scales to new task instances. We further found that by imposing a hierarchy on the problem, its complexity was further reduced and produced significantly better results in all aspects than a flat RL algorithm.

Deictic representations are considered a step in between a propositional representation and a fully relational representation [6, 5]. It is claimed that as a learning task increases in complexity, performance with deictic representations will decrease because of their inherent partial observability [7]. We expect that relational representations might be better suited to deal with the high complexity of commercial games than deictic representations. For future work, we therefore intend to work towards a fully relational reinforcement learning algorithm.

Acknowledgments

The first author is sponsored by DARPA and managed by NRL under grant N00173-06-1-G005. The views and conclusions contained here are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA, NRL, or the US Government. The second author is funded by a grant from the Netherlands Organization for Scientific Research (NWO grant No 612.066.406). The third author is sponsored by the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

References

- [1] B. Bakker and J. Schmidhuber. Hierarchical reinforcement learning based on subgoal discovery and subpolicy specialization. In *Proceedings of the Conference on Intelligent Autonomous Systems (IAS-08)*, pages 438–445, 2004.
- [2] A.G. Barto and S. Mahadevan. Recent advances in hierarchical reinforcement learning. *Discrete Event Dynamic Systems: Theory and Application*, 13(4):341–379, 2003.
- [3] T.G. Dietterich. Hierarchical reinforcement learning with the MAXQ value function decomposition. *Journal of Artificial Intelligence Research*, 13:227–303, 2000.
- [4] T.G. Dietterich. An overview of MAXQ hierarchical reinforcement learning. In *Proceedings of the Symposium on Abstraction, Reformulation and Approximation (SARA-00)*, *Lecture Notes in Computer Science*, volume 1864, pages 26–44, 2000.
- [5] K. Driessens. *Relational Reinforcement Learning*. PhD thesis, Katholieke Universiteit Leuven, 2004.
- [6] S. Dzeroski, L. De Raedt, and K. Driessens. Relational reinforcement learning. *Machine Learning*, 43:7–52, 2001.
- [7] S. Finney, N.H. Gardiol, L.P. Kaelbling, and T. Oates. Learning with deictic representation, 2002.
- [8] C. Guestrin, D. Koller, C. Gearhart, and N. Kanodia. Generalizing plans to new environments in relational MDPs. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI-03)*, 2003.
- [9] B. Marthi, S. Russell, and D. Latham. Writing Stratagus-playing agents in concurrent ALisp. 2005.
- [10] R. Parr and S. Russell. Reinforcement learning with hierarchies of machines. In *Proceedings of Conference on Advances in Neural Information Processing Systems (NIPS-97)*, pages 1043–1049, 1997.
- [11] M. Ponsen and P. Spronck. Improving adaptive game AI with evolutionary learning. In *Proceedings of Computer Games: Artificial Intelligence, Design and Education (CGAIDE-04)*, pages 389–396, 2004.
- [12] P. Spronck, M. Ponsen, E. Postma, and I.G. Sprinkhuizen-Kuyper. Adaptive game AI with dynamic scripting. *Machine Learning: Special Issue on Computer Games*, 2006.
- [13] R. Sutton and A. Barto. *Reinforcement Learning: an introduction*. MIT Press, Cambridge, MA, USA, 1998.
- [14] M. Wiering and J. Schmidhuber. HQ-learning. *Adaptive Behavior*, 6(2):219–246, 1997.

What Are Treebank Grammars?

D. Prescher^a R. Scha^a K. Sima'an^a A. Zollmann^b

^a *ILLC, University of Amsterdam* ^b *LTI, Carnegie Mellon University*

Abstract

State-of-the-art syntactic disambiguators for natural language employ "Treebank Grammars": probabilistic grammars directly projected from annotated corpora (treebanks). Treebank Grammars mark a paradigm shift from the manually constructed, a priori fixed linguistic grammars. In this paper we show that for describing these systems in the framework of Statistical Estimation Theory one must assume an unbounded number of parameters. The unboundedness assumption of Treebank Grammars expresses persistent uncertainty over the formal grammar of natural language. We argue that embracing the unboundedness assumption also brings the justification of smoothing techniques within the scope of Estimation Theory.

1 Introduction

A serious problem in Natural Language Processing is *ambiguity*. Formal grammars of natural languages are usually designed to describe the complete set of syntactic possibilities that a language offers. They tend to assign a large set of possible syntactic structures to virtually every utterance, although only one of these would be perceived by a human language user. To resolve this ambiguity, a parser must be able to rank the different analyzes of an utterance, and select one of them as the most plausible one. To this end, natural language parsers use *probabilistic grammars* [14, 1, 4, 6, 3], which assign a probability to every syntactic analysis they generate. Disambiguation decisions are then based on the probabilities of the different alternative structures of an input utterance.

This paper discusses parsing/disambiguation systems which are based on probabilistic grammars. In particular, it investigates how the rules and/or the probabilities of such grammars may be learned from a *treebank*, i.e., from a corpus of utterances which have been annotated with "correct" syntactic structures. Since the early nineties, several systems have been developed which assume a *grammar formalism* (i.e., the *format* for the grammar rules), and then learn the actual rules as well as their probabilities from the corpus. Such systems may be said to learn a *treebank grammar*. Since their estimator learn the whole grammar (rather than just the rule probabilities of a given grammar), they deal with an infinite rule set: the set of all possible rules allowed by the grammar formalism. We will call this infinite rule set an "infinite grammar", although this is dangerous terminology: in the parlance of theoretical linguistics, finiteness is considered an essential property of grammars. Clearly, this is a non-trivial extension. Though the "infinite grammars" which are learned by implementable estimators are necessarily representable by finite means, there does not need to be a finite bound on the size of this representation.

We will show that treebank grammars based on sufficiently rich grammar formalisms can approximate arbitrary parse-tree probability distributions arbitrarily closely. But this power has a cost: if we allow the training corpus to grow indefinitely, the grammar that is learned may also grow arbitrarily large.

We conclude the paper by discussing the well-known "sparse data problem" in the context of natural language parsing. As we know from Zipf's law, new words, new categorizations of existing words, and new syntactic combinations should keep appearing as the corpus grows. This implies that probability estimation by means of frequency counts will *never* be an adequate predictor for future utterances. To deal with this problem, existing systems employ techniques for *smoothing* their probability estimates: they reserve probability mass for the unseen events and redistribute this mass on the basis of various heuristics [8, 13, 11]. To justify smoothing techniques in the framework of Estimation Theory, we need

precisely the generalization that we introduced before: smoothing is concerned with estimating the values for an *infinite* dimensional parameter vector.

Before we discuss these matters in more detail, the next section introduces some basic concepts about formal and probabilistic grammars, and it reviews the current practice in treebank-based probabilistic parsing.

2 Current Practice in Corpus-Based NLP

The first subsection below describes, in very general terms, the kind of grammar formalisms being used in probabilistic parsing/disambiguation systems. The second subsection then gives an informal impression about the way in which the rules and/or probabilities of such grammars are learned from treebanks. More precise investigations of that issue constitute the topic of the rest of the paper.

2.1 Probabilistic Grammars

Probabilistic grammars are based on *rewrite grammars*. In particular, many are based on *Context-Free Grammars* (CFGs). A CFG is a tuple $\langle \mathcal{N}, \mathcal{W}, \mathcal{R}, S \rangle$, where \mathcal{N} and \mathcal{W} are disjoint finite sets of *non-terminal* and *terminal* symbols, respectively; $S \in \mathcal{N}$ is a distinguished start symbol, and \mathcal{R} is a finite set of *productions*, of the form $A \rightarrow \beta$, where $A \in \mathcal{N}$ and β is a sequence of elements from $\mathcal{N} \cup \mathcal{W}$. A CFG defines a rewrite system, where S is rewritten, in consecutive steps, using the productions; at each step the result of rewriting is a sequence of symbols from $\mathcal{N} \cup \mathcal{W}$. Such a rewrite process, called a *derivation*, terminates when the resulting sequence consists of only terminal symbols; such a sequence is called a *sentence* of the language generated by the grammar.

The graphical representation of a CFG-derivation is at the same time the parse-tree of the sentence. In other grammar formalisms, such as *Tree-Substitution Grammars* (TSGs) and *Tree-Adjoining Grammars* (TAGs) [10], the one-to-one correspondence between sentence structure and derivation structure is given up. Grammars of this sort resemble CFGs, with one important difference: the rewrite process explicitly generates parse-trees rather than symbol sequences. The production rules now consist of a non-terminal symbol on the left hand side of the arrow, and an *elementary-tree* on the right hand side; the rule specifies that the rewrite process may select a node which is labeled with this non-terminal, and expand it into a copy of this tree.

A *Probabilistic CFG* (PCFG) extends a CFG with a probability function $\pi : \mathcal{R} \rightarrow [0, 1]$, which assigns to each production a probability value. The probability of derivation r_1, \dots, r_n , involving productions $\forall i : r_i \in \mathcal{R}$, is calculated under an independence assumption between the individual productions, i.e. $p(r_1, \dots, r_n) = \prod_{i=1}^n \pi(r_i)$. The probability of a sentence u ($p(u)$) is given by the sum of the probabilities of all derivations that generate u .

Like PCFGs, *Stochastic TSGs* (STSGs) and *Stochastic TAGs* (STAGs) extend their conventional counterparts with a probability function over the productions (elementary-trees). Note, however, that these formalisms allow multiple derivations of the same parse-tree. Hence, the probability of a parse-tree t , $p(t)$, is now the sum of the probabilities of the derivations that generate t .

2.2 Treebank grammars and Language Models

Two issues play a role in how probabilistic grammars are actually built (1) the way the symbolic grammar is obtained, (2) how the probabilities of the grammar productions are estimated from an actual corpus. The earliest probabilistic parsing systems concentrated on estimating the probabilities of the productions of *broad-coverage, linguistically motivated, manually constructed grammars* e.g. [7]. In contrast, state-of-the-art approaches e.g. [15, 2, 4, 6] have abandoned the broad-coverage linguistic grammars for the sake of so-called *treebank grammars* (a term coined by Charniak [4]). The characteristic property of a treebank grammar is that its rules are projected directly from a corpus consisting of a sample of utterances that are manually annotated with the correct syntactic structures (a so-called *treebank*). See the example in Figure 1.

When seen from a broader perspective than parsing, treebank grammars were inspired by the statistical *corpus-based approaches* that underly *Statistical Language Models* (SLMs) that are so prevalent in speech-recognition [9]. In its purest form, a statistical language model stands for a probability distribution over a set of sequences of words (utterances). This statistical notion can be seen as an extension of the formal notion of a language, i.e. a set of utterances. In speech-recognition, a language model employs a probabilistic regular grammar (Finite State Automaton) where the probability of every word is conditioned on the preceding words in the utterance, i.e. a so called Markov grammar. Markov

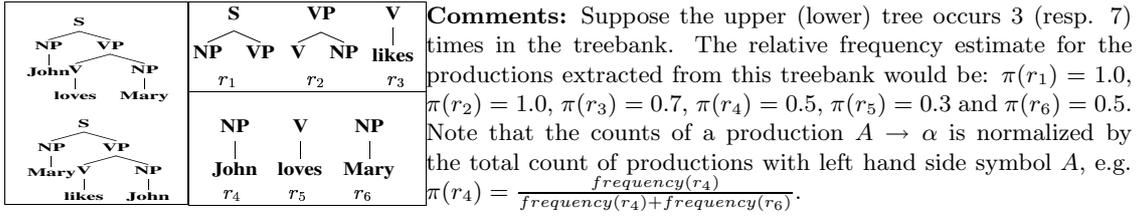


Figure 1: (left) A treebank, (right) The treebank CFG (see comments).

grammars can be seen as the first treebank grammars; an n^{th} -order Markov grammar is obtained by collecting from a large corpus all the $(n+1)$ -grams together with their frequency counts.

This paper studies Treebank and Corpus-based Grammars from the point of view of statistical Estimation Theory. We shall see that a satisfactory account of the treebank grammar approach requires some changes in this theory.

3 Standard Estimation Theory

Our basic setting is a random experiment with an underlying sample space and a probability measure p . A *random experiment* is an experiment whose outcome cannot be predicted with certainty. A *random variable* X is a measurement in the context of the random experiment; it is random in the sense that its value depends on the outcome of the experiment. Each time the experiment is run, an outcome of the experiment occurs, and X takes on a value $x \in S$. The range S of X is called the *sample space* of X . Assuming that S is countable, X is a discrete variable characterized by its *probability distribution* $p_X : S \rightarrow [0, 1]$, where $p_X(x) := p(X = x)$ for all $x \in S$. Finally, we assume that X 's distribution is parameterized, i.e., dependent on some *parameters*, i.e., non-random quantities in the random experiment that, once chosen, remain constant. In most cases, (one or more of) the parameters are unknown, and must be estimated from the outcome vector X .

Random Samples, Statistics, and Estimators Let X_1, \dots, X_n be independent random variables with the same distribution as the variable X from the above setting. Then $\langle X_1, \dots, X_n \rangle$ is called a *random sample* of X . The outcome of the variables in a particular trial, $\langle x_1, \dots, x_n \rangle$, is called an *observation sequence*. A *statistic* is a random variable derived from the random sample $\langle X_1, \dots, X_n \rangle$. Examples for statistics are the *sample mean* $\bar{X}_n = \frac{1}{n} \sum_{i=1}^n X_i$ and the *sample variance* $s_n^2 = \frac{1}{n-1} \sum_{i=1}^n (X_i - \bar{X}_n)^2$.

Guessing the distribution of X from an observation sequence involves assumptions on what kind of distributions are admissible. More formally, estimation is based on a *model* \mathcal{M} —the set of admissible distributions. The ‘true’ distribution to be estimated from the observation sequence is assumed to be an instance of \mathcal{M} . In standard estimation theory, the model \mathcal{M} is often characterized by a set $\Theta \subseteq \mathbb{R}^k$ of *parameter vectors*, such that there is a one-to-one correspondence between Θ and \mathcal{M} . An example is the model of all Gaussian distributions, where each distribution is characterized by the parameters $\langle \mu, \sigma \rangle \in \mathbb{R} \times \mathbb{R}_+$ (μ being the distribution’s expected value and σ its standard deviation). The process of estimation can now be described in terms of parameters. Let p_θ be the distribution corresponding to parameter vector $\theta \in \Theta$ and θ^* the parameter vector of the “true” distribution $p_X = p_{\theta^*}$.

An *estimator* est_n of θ^* is a statistic from the random sample $\langle X_1, \dots, X_n \rangle$ whose range is Θ . Thus, the estimator is a random variable with a distribution, an expected value, and so on. The observed value $\text{est}_n(\langle x_1, \dots, x_n \rangle)$ of the outcome of a particular random experiment is called the *estimate* of θ^* from the observations $\langle x_1, \dots, x_n \rangle$. For example, the prominent *maximum-likelihood method* aims at maximizing the probability of the observation sequence: $\hat{\theta} = \arg \max_{\theta} \prod_{i=1}^n p_\theta(x_i)$. If such an estimate $\hat{\theta}$ exists for each possible observation sequence, the corresponding statistic $\hat{\theta}(\langle X_1, \dots, X_n \rangle)$ is called a *maximum-likelihood estimator* of θ^* .

Properties of Estimators in Standard-Estimation Theory When investigating the properties of estimators, all distributions $p \in \mathcal{M}$ that could underly X are considered. This means that X is distributed according to a distribution p_θ depending on a parameter vector $\theta \in \Theta$. In other words, we implicitly assume in the following that $p_X = p_\theta$.

Bias and Consistency. The (random) *error* is the difference between the estimator and the parameter vector: $\text{est}_n - \theta$. The expected value of the error is known as the *bias*: $\text{bias}_\theta(\text{est}_n) = \text{E}(\text{est}_n - \theta)$. If

$\text{bias}_\theta(\text{est}_n) = 0$ (i.e., $E(\text{est}_n) = \theta$) for all $\theta \in \Theta$, then est_n is said to be *unbiased*. We also expect our estimators to improve, in some sense, as n increases. As the quality of an estimator can be measured by a *loss function* (e.g. by $\text{loss}_\theta(\text{est}_n) = \|\text{est}_n - \theta\|^2$), a sequence of estimators is called *consistent* if for each $\theta \in \Theta$, the expected loss approaches zero as n goes to infinity: $\lim_{n \rightarrow \infty} E(\text{loss}_\theta(\text{est}_n)) = 0$.

Minimal Sufficiency. An estimator is a statistic with the parameter space Θ as its range. A statistic $U = h(X_1, \dots, X_n)$ is called *sufficient for a parameter* θ if U contains all of the information about θ that is available in the entire random sample¹. Obviously, the entire random sample is trivially sufficient. There is, however, usually a statistic U that is sufficient *and has a smaller dimension*, so that we can achieve real data reduction. Naturally, we would like to find the statistic U that has the smallest dimension possible.

In summary, standard estimation theory assumes that: (1) the parameter space is finite-dimensional, (2) the true distribution is in the model, and (3) estimation is/should be based on minimal sufficient statistics. In the following sections, we investigate whether these assumptions hold for treebank grammars.

3.1 The Parameters of Probabilistic Parsing

PARSES		STSG1			STSG2		
t_1	t_2	0.25	0.25	1.0	0.5	0.5	0.5
	S	S					
S	$\hat{A}a$	$\hat{A}a$	S	A	S	S	A
$\hat{A}a$			$\hat{A}a$			$\hat{A}a$	
	a	a	a	a	a	a	a

Comments: Both STSGs generate the same parse distribution $\{\langle t_1, 0.5 \rangle, \langle t_2, 0.5 \rangle\}$. (Note that the left STSG generates t_2 in two different derivations each contributing 0.25 to its probability: (1) $(S \rightarrow A \ a \rightarrow a \ a)$ and (2) combine $(S \rightarrow A \ a)$ with $(A \rightarrow a \ a)$).

Figure 2: Two STSGs generating same parse distribution. See comments.

In order to describe the learning of probabilistic parsers from an Estimation Theory perspective, we first need to pin down the parameters that are being estimated. In general, probability estimation from a corpus is used for two tasks: (1) estimating the production probabilities of an a priori fixed probabilistic grammar, or (2) estimating a probability distribution over parses. In the first case, the actual parameters are the *production probabilities*, while in the latter these are the *parse probabilities*. Remarkably, for estimating a certain probability distribution over parses one could employ different (kinds of) probabilistic grammars as shown in Figure 2. Hence, the two tasks, estimating a probabilistic grammar or estimating a parse distribution, are not necessarily equivalent.

What is then the appropriate kind of parameters for ambiguity resolution? Note that the role of the probabilities is to rank the different parses in order to facilitate the selection of the most probable one. Clearly, if we had access to estimates of parse probabilities (a parse distribution), we would have *sufficient* means for disambiguation. Hence, the parameters that are subject to estimation should be *parse probabilities*.

Note that for estimating production probabilities one must pin down a target probabilistic grammar prior to estimation. Pinning down the grammar prior to estimation implies a very strong assumption: the chosen grammar reflects the exact nature of natural language syntax. Clearly, this assumption does not always hold for existing formal grammars. If parse probabilities are the subject of estimation, why then does existing work in NLP highlight the estimation of production probabilities?

The reason is quite simple: the direct estimation of parse probabilities implies an *infinite* dimensional vector of parameters, i.e. the parses. This does not go well with Estimation Theory, which assumes a finite dimensional parameter vector $(\langle \theta_1, \dots, \theta_n \rangle)$. Work on probabilistic parsing tackles this infinity by assuming that a known probabilistic grammar, with a finite set of productions, generates the parse distribution. Let be given a set of parses \mathcal{T} over finite sets of nonterminal \mathcal{N} and terminal \mathcal{W} symbols. If we assume now that the set of parses \mathcal{T} is generated by a grammar $G = \langle \mathcal{N}, \mathcal{W}, \mathcal{R}, S \rangle$, e.g. a CFG, then each parse t is generated by a G -derivation. For a given treebank T_1, \dots, T_x , the Likelihood under a parse probability function p is given by $L(T_1, \dots, T_x, p) = \prod_{i=1}^x p(T_i)$. If for all $1 \leq i \leq x$, T_i is derived by some derivation r_1, \dots, r_n , then $p(T_i) = \prod_{j=1}^n \pi(r_j)$. Therefore, the commonly used estimator, the Maximum-Likelihood of a treebank, simply reduces to the Maximum-Likelihood over a “corpus of

¹Sufficiency is related to the concept of data reduction: If we can find a sufficient statistic U taking values in a m -dimensional space, then we reduced the original data vector $\langle X_1, \dots, X_n \rangle$ (whose dimension n is usually large) to the statistic U (whose dimension m is usually much smaller) with no loss of information.

independent grammar productions”. This means that under the assumption that a given probabilistic grammar generates the distribution over \mathcal{T} , estimating the production probabilities of this grammar will also yield parse estimates. The question is, of course, whether the assumption of knowing the grammar productions is a reasonable one for natural language parsing? As we see in the sequel, this assumption has been found to be too strong for dealing with various problems, and a different approach has emerged, *Treebank Grammars*.

There are a few important realizations that arise in this section: (1) the actual goal is to estimate a (possibly) *infinite* parse distribution (of which the treebank is a finite sample), (2) by assuming that a given grammar generates the parse distribution, a structure is imposed on the infinite dimensional parse-space such that the dimension of the parameter vector is reduced to a finite dimension (the number of grammar productions).

4 Treebank Grammars and Estimation Theory

The common feature of treebank-based approaches to stochastic parsing is that they do not assume an *a priori* grammar, but learn a *treebank grammar* from a corpus. In what follows, we confront treebank-based approaches to stochastic parsing with estimation theory. We briefly review so-called *Data-Oriented Parsing* (DOP), one of the earliest and most radical approaches to treebank grammars: it suggests using an annotated corpus directly as a stochastic grammar [15]. After reviewing DOP, we point out the incompatibilities between DOP and Estimation Theory as a starting point for our further discussion concerning other treebank grammar approaches.

Data-Oriented Parsing Like other treebank models, DOP acquires from a treebank TB a finite set \mathcal{F} of productions, called *fragments*² or *subtrees*, together with their probability estimates. An important feature of DOP is that the set \mathcal{F} consists of *all* fragments of the treebank trees. The set of fragments \mathcal{F} is employed as an *STSG* with the same start symbol, nonterminal and terminal sets as the treebank trees (see section 2 for STSGs). Like other STSGs, a DOP grammar is based on fragment probabilities that allow the calculation of derivation and parse probabilities:

Fragment probability: To each $t \in \mathcal{F}$, a real number $\pi(t) \geq 0$ is assigned, such that for non-terminal label A , π induces a probability distribution on the set of fragments t whose root label $\text{root}(t)$ is A . I.e.: $\sum_{t : \text{root}(t)=A} \pi(t) = 1$.

Derivation and Tree probabilities: The probability of a derivation d of a parse-tree x is the product of the fragment probabilities in d , and the probability of a parse-tree x with a set of derivations $\mathcal{D}(x)$ is the sum of the probabilities of all derivations in $\mathcal{D}(x)$:

$$(1) \quad p(d) = \prod_{t \in \mathcal{F}} \pi(t)^{f_t(d)} \qquad (2) \quad p(x) = \sum_{d \in \mathcal{D}(x)} p(d)$$

where $f_t(d)$ is the number of times the fragment t occurs in the derivation d .

Estimation Theory and DOP Recall that the focus of estimation for parsing is a probability distribution over parses. The statistics employs a *corpus*, that has been generated in accordance with some unknown probability distribution, in order to infer that distribution.

Given a probability model comprising several distributions which might have generated the corpus, an *estimation method* selects one *instance* of the probability model as its best guess about the original distribution. From the perspective of DOP, a probability model simply bundles specific probability distributions on the set \mathcal{X} of derivable parse trees, where each model instance p is induced by a function π on a given set \mathcal{F} of tree fragments such that the equations (1) and (2) are satisfied. In other words, a *DOP model* is defined on the basis of a given set \mathcal{F} of tree fragments such that

$$\mathcal{M}_{\text{DOP}}(\mathcal{F}) = \left\{ p \in \mathcal{M}(\mathcal{X}) \mid \exists \pi \text{ such that } \forall x \in \mathcal{X} : p(x) = \sum_{d \in \mathcal{D}(x)} \prod_{t \in \mathcal{F}} \pi(t)^{f_t(d)} \right\}$$

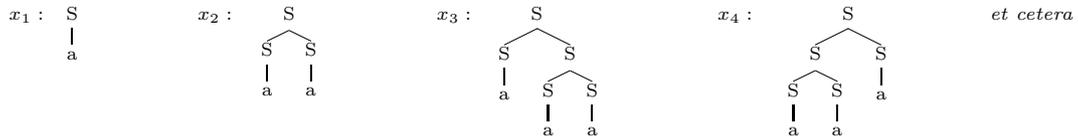
Moreover, given such a DOP model, an estimator est_n for this model can be described as a statistic from a random sample $\langle X_1, \dots, X_n \rangle$, having the DOP model as its range. In other words, an observed

²A connected subgraph of a treebank tree t is called a *fragment* iff it consists of one or more context-free productions.

value $\text{est}_n(\langle x_1, \dots, x_n \rangle)$ of this estimator is an instance of $\mathcal{M}_{\text{DOP}}(\mathcal{F})$, thereby estimating this model instance by exploiting the corpus $\langle x_1, \dots, x_n \rangle$. This result can be expressed by:

$$\langle x_1, \dots, x_n \rangle \mapsto \text{est}_n(\langle x_1, \dots, x_n \rangle) \in \mathcal{M}_{\text{DOP}}(\mathcal{F})$$

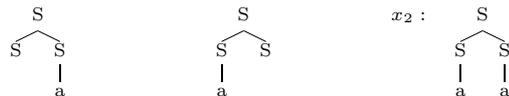
To study the asymptotic behavior of DOP, we will now investigate a sequence $\text{est}_1, \text{est}_2, \text{est}_3, \dots$ of DOP estimators. Starting with an example, the following figure displays the parse trees that are derivable by the context-free grammar with the rules $S \rightarrow SS$ and $S \rightarrow a$.



From the perspective of DOP, however, each context-free grammar *is* a DOP grammar. Each CFG rule can be interpreted as a local tree, and the set \mathcal{F} of tree fragments of the DOP grammar simply consists of these local trees. In our example, \mathcal{F} consists of the two fragments:



From the perspective of estimation theory, we are now assuming that a model instance of this DOP grammar is used for sampling. Then the specific corpus $\langle x_1, \dots, x_n \rangle$ is clearly one of the possible observations of the random sample $\langle X_1, \dots, X_n \rangle$. The crucial point is now that the set of fragments that are read off from $\langle x_1, \dots, x_n \rangle$ includes fragments that are *not* in \mathcal{F} — at least if we assume that the corpus size n is big enough. Here, we check e.g. for $n \geq 2$ that



are not fragments of the DOP grammar that generated the corpus. There is nothing special in the given example that stops us from generalizing this finding:

By assuming that the treebank is sampled from a DOP grammar that is interesting enough, i.e., that can be used to derive an infinite number of trees, then one aspect of the asymptotic behavior of DOP estimation is that *the symbolic DOP grammar grows as the treebank grows*. As an effect, in the limit of the treebank size, DOP risks learning an arbitrarily large grammar.

Although it is obvious that this phenomenon does not occur for simpler grammar formalisms (like PCFGs and probabilistic finite-state automata), we conjecture that it also occurs for other *higher-order* treebank grammars (like unification-based grammar formalisms). We think also that there is a connection to the task of estimating *a priori* grammars in the framework of *unsupervised learning*.

Summarizing the results of this section, we started out by learning tree distributions, and we find out that treebank grammars are indeed doing so. A surprise is, however, that DOP learns an arbitrarily large grammar (in the limit). We conjecture, however, that this is a more general problem for other kinds of treebank grammars. As we will point out in the next section, this is an extreme case of other common phenomena in Natural Language Processing.

4.1 Related Corpus-based Methods

As we have seen, theoretically speaking, DOP aims at estimating the probabilities of an infinite dimensional parameter vector. This implies that DOP estimation is incompatible with Estimation Theory. In this section we will show that this is not unique for DOP; other commonly used corpus-based methods based on Markov Models and other treebank grammars employ smoothing techniques that have the same property: they all assume an infinite grammar.

A common problem in speech and language processing is the problem of “unknown words”, i.e. words that have not occurred in the training corpus. As the corpus grows, novel words will occur all the time. Open category words, such as proper nouns and compound nouns, are a common problem, but even novel verbs are made up on the fly all the time (e.g. “googling someone” for searching on the web). The situation is more severe with languages with rich morphology (e.g. Arabic or Turkish). The

problem of unknown words has been linked to Zipf’s law [16]: there is always a considerable tail of very low frequency phenomena to be expected to occur in the future. This problem has major consequences for the estimation of production probabilities, because one cannot determine the finite set of allowed words (terminal symbols) a priori to estimation.

For language models based on Markov processes over word sequences (e.g. Hidden Markov Models for speech-recognition) [9], it is not possible to fix a finite set of parameters (bigrams or trigrams). Therefore, Markov Grammars are usually obtained directly from a large corpus (n-grams together with their estimates). Theoretically speaking, we must assume an infinite dimensional parameter vector if we are to explain the asymptotic estimation properties of the corpus-based Markov models.

We may stretch the unknown word problem one step further to unknown category words, i.e. words for which some part-of-speech tag categories are not in the corpus. Similarly, unknown productions could occur: in the well-known Penn Wall Street Journal treebank [12] many productions occur only once, hinting at the fact that other novel productions are likely to occur in new utterances.

The problem of the “unknown events” constitutes the smallest common deviser for all language (and other) modeling activities. Various smoothing techniques have been developed for dealing with this problem, see [5]. These smoothing techniques are used as follows: **1.** Estimate the parameters of a (finite) grammar, including a special symbol UNKNOWN, a category of unknown events, **2.** Use a mapping from a word to itself if it is known, or else to the UNKNOWN category, and **3.** Reserve and distribute probability mass to the map into UNKNOWN. Theoretically speaking, the second step (the mapping) can only be described by an infinite set of rules that maps a novel word to its UNKNOWN.

5 Discussion and conclusion

As we have seen in the two preceding sections, it seems necessary and reasonable to lift certain restrictions on the grammar assumed to generate a natural language. Recall that for an ordinary CFG (cf. Section 2), the nonterminals \mathcal{N} , terminals \mathcal{W} , as well as the productions \mathcal{R} are required to be finite sets. Which finiteness-restrictions should we abolish to fit the treebank models?

In Section 5, we observed that the number of productions (subtrees) of DOP treebank grammars may increase as training treebank size increases. In the limit, DOP then seems to estimate a infinite grammar; however, so far we were not able to express this behavior in a formal way. Allowing for \mathcal{R} to be a *countable* set enables the generation of new tree distributions. Thus, the model $\mathcal{M}_{\text{DOP}}(\mathcal{R})$ (where the productions \mathcal{R} are a (countable) set of DOP fragments) can now be chosen much richer. Consider the special case that \mathcal{R} is chosen as the set of all fragments from a given countable set T of full parse trees. Then, given a DOP estimation method $\langle \text{est}_i \rangle_{i \in \mathbb{N}_+}$, the following desirable property holds:

For any infinite sequence $\langle t_i \rangle_{i \in \mathbb{N}_+}$ of observations $t_i \in T$, if the corresponding sequence $\langle \text{est}_i(\langle t_1, \dots, t_i \rangle) \rangle_{i \in \mathbb{N}}$ of DOP estimates converges to a probability distribution, then that distribution is an instance of the model $\mathcal{M}_{\text{DOP}}(\mathcal{R})$.³

Furthermore, the increased generality of $\mathcal{M}_{\text{DOP}}(\mathcal{R})$ now allows estimators to learn tree distributions that could not be learned before, simply because they were not allowed to be generated due to the finiteness condition on the productions of the underlying grammar.

Another problem we encountered in Sections 5 and 6 was the handling of unknown words. Firstly, with growing training data, previously unseen words are bound to occur and keep occurring. No finite *a-priori* set of terminals can account for this. Secondly, smoothing techniques used in state-of-the-art statistical language modeling implicitly make use of an infinite number of unknown-word productions. Defining both \mathcal{W} and \mathcal{R} as countable sets prepares the grounds for a proper formal treatment of these phenomena.

In certain grammar formalisms, also a countable set of *nonterminals* could be desirable. If \mathcal{W} , \mathcal{N} and \mathcal{R} are allowed to be countable, the resulting set of utterances will also be countable, since the set of possible derivations from the grammar—finite sequences of rules, i.e., of tuples over $\mathcal{W} \cup \mathcal{N}$ —is countable.

In light of the discoveries made in the preceding sections it is intriguing to review the development of probabilistic parsing. Early work on the estimation of parse distributions from treebanks started out by introducing grammar formalisms in order to avoid working with an infinite number of parameters. State-of-the-art treebank grammars adopted this strategy but, from a theoretical perspective, end up

³This is true because $\mathcal{M}_{\text{DOP}}(\mathcal{R})$ is then identical to the set of all probability distributions over T .

again with an infinite number of parameters (although all actual models only deal with a finite number of parameters).

The main explanation for this seemingly circular development is that the kind of grammars that are suitable for capturing natural language syntax are far more complex than any existing formal grammars. The fact that language is a continuously and rapidly evolving phenomenon raises a question as to whether any a priori fixed, finite grammar will be able to approximate language syntax (including phenomena as unknown words). From this perspective, it seems that the treebank grammar approach, together with smoothing techniques, can be seen as an *efficient* solution for this problem.

References

- [1] E. Black, F. Jelinek, J. Lafferty, D.M. Magerman, R. Mercer, and S. Roukos. Towards History-based Grammars: Using Richer Models for Probabilistic Parsing. In *Proceedings of the 31st Annual Meeting of the ACL (ACL'93)*, Columbus, Ohio, 1993.
- [2] R. Bod. *Enriching Linguistics with Statistics: Performance models of Natural Language*. PhD dissertation. ILLC dissertation series 1995-14, University of Amsterdam, 1995.
- [3] R. Bod, R. Scha, and K. Sima'an, editors. *Data Oriented Parsing*. CSLI Publications, Stanford University, Stanford, California, USA, 2003. not refereed.
- [4] E. Charniak. Tree-bank Grammars. In *Proceedings AAAI'96*, Portland, Oregon, 1996.
- [5] S. Chen and J. Goodman. *An empirical study of smoothing techniques for language modeling*. Technical Report TR-10-98, Harvard University, August 1998.
- [6] M. Collins. Three generative, lexicalized models for statistical parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL) and the 8th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 16–23, Madrid, Spain, 1997.
- [7] T. Fujisaki, F. Jelinek, J. Cocke, E. Black, and T. Nishino. A probabilistic method for sentence disambiguation. In *Proceedings of the 1st International Workshop on Parsing Technologies*, 1989.
- [8] I.J. Good. The population frequencies of species and the estimation of population parameters. *Biometrika*, 40:237–264, 1953.
- [9] F. Jelinek. *Statistical Methods for Speech Recognition*. Cambridge: MIT Press, 1997.
- [10] A.K. Joshi. Tree Adjoining Grammars: How much context sensitivity is required to provide a reasonable structural description. In I. Karttunen D. Dowty and A. Zwicky, editors, *Natural Language Parsing*, pages 206–250, Cambridge, U.K., 1985. Cambridge University Press.
- [11] S.M. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech and Signal Processing (ASSP)*, 35(3):400–401, 1987.
- [12] M. Marcus, B. Santorini, and M. Marcinkiewicz. Building a large annotated corpus of English: The Penn treebank. *Computational Linguistics*, 19:313–330, 1993.
- [13] H. Ney, S. Martin, and F. Wessel. Statistical language modeling using leaving-one-out. In S. Young and G. Bloothoof, editors, *Corpus-based Methods in Language and Speech Processing*, pages 174–207. Kluwer Academic, Dordrecht, 1997.
- [14] Arto Salomaa. Probabilistic and Weighted Grammars. *Inf.Control*, 15:529–544, 1969.
- [15] R. Scha. Taaltheorie en taaltechnologie; competence en performance. In *Computertoepassingen in de Neerlandistiek, LVVN-jaarboek*. English translation as: Language Theory and Language Technology; Competence and Performance <http://iaaa.nl/rs/LeerdamE.html>, 1990.
- [16] G.K. Zipf. *Human behavior and the Principle of Least Effort: an Introduction To Human Ecology*. Addison-Wesley, 1949.

Improving Ant Colony Optimization to solve the Travelling Salesman Problem

Amilkar Puris^a Rafael Bello^a Ann Nowe^b Yailen Martínez^a

^a Department of Computer Science, Universidad Central de Las Villas, Cuba.

^b Comp Lab, Department of Computer Science, Vrije Universiteit Brussel, Belgium.

Abstract

In this paper we propose an improvement of Ant Colony Optimization to solve the Travelling Salesman Problem. In this case, we use the Ant Colony System algorithm. The basic idea is to split the heuristic search performed by ants into two stages. That is, a multilevel approach is used. Also, the effect of using local search was analyzed. Experimental results obtained show this new approach significantly improves the Ant Colony System in terms of the computation time needed.

1. Introduction

Artificial ants are simple agents that incrementally build a solution by adding components to a partial solution under construction. In Ant Colony Optimization (ACO) a colony of artificial ants cooperates to look for good solutions to discrete problems. ACO is a metaheuristic used to guide other heuristics in order to obtain better solutions than those that are generated by local optimization methods. This computational model was introduced in 1991 by M. Dorigo and co-workers [2]. Also, you could see [3, 5].

Ant System (AS) is the first ACO algorithm; it was introduced using the Travelling Salesman Problem (TSP) [4]. In the TSP, we have a set of N fully connected cities $\{c_1, \dots, c_n\}$ by arcs (i,j) ; each arc is assigned a weight d_{ij} which represents the distance between cities i and j , the goal is to find the shortest possible tour visiting each city once before returning to initial city. When ACO is used to solve this problem, pheromone trails (τ_{ij}) are associated with arcs to denote the desirability of visiting city j directly from city i . Also, the function $\eta_{ij} = 1/d_{ij}$ indicates the heuristic desirability of going from i to j . Initially, ants are randomly assigned to cities. In the successive steps an ant k applies a random proportional rule to decide which city to visit next according to expression (1):

$$p_{ij}^k = \frac{(\tau_{ij})^\alpha * (\eta_{ij})^\beta}{\sum_{l \in N_i^k} (\tau_{il})^\alpha * (\eta_{il})^\beta} \quad \text{if } j \in N_i^k \quad (\text{neighborhood of ant } k) \quad (1)$$

Where α and β are two parameters to point out the relative importance of the pheromone trail and the heuristic information, respectively. After all ants have built their tours the values τ_{ij} are updated in two stages. First, τ_{ij} values are decreased by evaporation, $\tau_{ij}=(1-\rho)*\tau_{ij}$, using the parameter ρ , where $0<\rho<1$. This is done to avoid unlimited accumulation of pheromone. Secondly, all ants increase the value of τ_{ij} on the arcs they have crossed in their tours, $\tau_{ij}=\tau_{ij}+Inc_{ij}$, where Inc_{ij} is the amount of pheromone deposited by all ants which included the arc (i,j) in their tour. Usually, the amount of pheromone deposited by ant k is equal to $1/C_k$, where C_k is the length of the tour of ant k.

Ant Colony System (ACS) is a successor algorithm of Ant Systems. ACS uses the following pseudorandom proportional rule to select the next city j from city i.

$$j = \begin{cases} \arg \max_{l \in N_i^k} \{ (\tau_{il})^\alpha * (\eta_{il})^\beta \} & \text{if } q \leq q_0 \\ \text{random selection according to (1)} & \text{otherwise} \end{cases} \quad (2)$$

where q is a random variable uniformly distributed in [0,1], q_0 is a parameter taken in the interval [0,1], which controls the amount of exploration, and $\alpha=1$ in the random selection (expression 1). In ACS, ants have a local pheromone trail update ($\tau_{ij}=(1-\xi)*\tau_{ij} + \xi*\tau_{ij}(0)$) applied after crossing an arc(i,j), where $\tau_{ij}(0)$ represents the initial value for the pheromone, and a global pheromone trail update ($\tau_{ij}=(1-\rho)*\tau_{ij} + \alpha*Inc_{ij}$) executed only by the best-so-far ant.

In this paper, we propose a new approach to ACO in which the search process developed by ants is split into two stages. This is called Two-Stage Ant Colony System algorithm (TS-ACS); the performance of it is studied in the case of the Travelling Salesman Problem.

2. Related Works

The problem of finding low cost tours in reasonable time rather than solving the problem to optimality in the case of the TSP using a multilevel approach was addressed in [8, 9]. The multilevel idea was first proposed by Bernard and Simon [1] as a method in speeding-up the recursive spectral bisection algorithm partitioning unstructured problems. The approach presented in [8] progressively coarsens the TSP, initializes a tour and then employs either the Lin-Kernighan or the Chained Lin-Kernighan algorithms to refine the solution on each

coarsened problem; the resulting multilevel algorithm is shown to considerably enhance the quality of tours.

On the other hand, authors in [6] present a multilevel approach to ACO, for solving the mesh partitioning problem in the finite element methods. The multilevel ACO algorithm performs very well and it is shown to be superior to several classical mesh partitioning methods. Their studies show that ACO was successful to solve the graph-partitioning problem in the case of graphs of smaller size, so they enhanced the basic ACO with a multilevel technique. That is, a set of the largest independent subgraph is created from the original graph, these are optimized using ACO, and then the optimized partition is expanded.

In the model proposed in this paper we use a multilevel approach in ACO to solve the TSP. However, this model differs from the previous approach [6] in some aspects: (i) the search process developed by the ants is divided in two stages instead of on a grouping of cities; (ii) the search process developed by ants is partitioned into two more simple search processes instead of partitioning the problem into more simple subproblems; (iii) the partial solution obtained in the first level is used as initial state for the search of the ants in the second level.

3. Two-Stage ACS Algorithm

The TS-ACS proposed in this investigation is based on the following idea: to divide the search process performed by the ants in two stages, so that, in the first stage preliminary results are reached (partial solutions) that serve as an initial state for the search performed by the ants in the second stage. In the case of TSP, this means that tours containing a subset of cities are generated in the first stage, in the second stage, these routes will serve as an initial state for the ants.

The determination of the initial state in which the search process starts has been an interesting problem in heuristic search. The aim is to be able to move the initial state towards the goal state. Of course, it is necessary to consider an adequate balance between the computational cost of obtaining that initial state and the total cost.

More formally, the purpose is the following. Let E_i be an initial state randomly generated, or obtained by any other method without a significant computational cost, E_i^* is an initial state generated by some method M that moves it towards the goal state, $C_M(E_i^*)$ denotes the cost of obtaining the state E_i^* from E_i using the method M , and $CC_{HSA}(x)$ is the computational cost to find a solution from state x using a Heuristic Search Algorithm (HSA). Then, the objective is that $C_M(E_i^*) + CC_{HSA}(E_i^*) < CC_{HSA}(E_i)$.

In the TS-ACS algorithm proposed here, the procedure to generate E_i^* and the HSA are both the ACS algorithm, so the goal is $C_{ACS}(E_i^*) + CC_{ACS}(E_i^*) < CC_{ACS}(E_i)$. As ACS is used in both stages, the difference between the 2 stages is obtained by giving different values to some parameters in each stage.

A ratio (r) is introduced in order to establish the relative setting of the values of the parameters in both stages; the ratio indicates which proportion of the complete search is given in each stage. For instance, if $r=0.3$ the first stage covers 30% of the search process and the second the remaining 70%.

The setting of the ratio r has a high influence on the overall performance of the algorithm. A high value of r , say almost 1, causes the state E_i^* to be closer to the goal state, by doing so the value of $C_{ACS}(E_i^*)$ may increase and the value of $CC_{ACS}(E_i^*)$ will decrease. But, in addition to this balance between the costs of $C_{ACS}(E_i^*)$ and $CC_{ACS}(E_i^*)$, we have the problematic about how much the space search is explored; while more greater is the rate r , the search in the second stage decreases for several reasons among them there are less ants working, and the amount of cycles decreases.

Therefore, a key point is to study what value of rate r is the best in order to obtain the best balance between the searches in both stages. This value must allow (i) To minimize the value of $C_{ACS}(E_i^*) + CC_{ACS}(E_i^*)$, and (ii) To allow an exploration of the search space that guarantees to find good solutions.

4. Two-Stage ACS in the Travelling Salesman Problem

When applying the ACS algorithm to the TSP, in every cycle the ants begin the search starting from a randomly selected city, and choose the next city to visit using rule (2). On the contrary, the TS-ACS constructs partial tours in the first stage; these serve as initial state for the ants in the second stage of the search process. That is, ants use the partial tours built in the first stage as the starting point in the second stage.

In TSP the parameters whose values are set depending on the ratio r are: the quantity of ants (m), the number of cycles (nc), and the amount of cities (cc) that must be included in the tour in each stage. The setting of the parameters is done in the following way. Consider a parameter setting for the ACS algorithm as follows: $m=10$, $nc=1000$, and $cc=30$; and a setting of the ratio as $r=0.3$. Then the values of these parameters for the TS-ACS are $m_1=10*0.3=3$, $nc_1=1000 * 0.3 = 300$ and $cc_1=30 * 0.3 = 9$ for the first stage; and $m_2=7$, $nc_2=700$ and $cc_2=30$. This means that 3 ants execute the ACS algorithm during 300 cycles starting from random cities and constructing tours of 9 cities. In the second stage, 7 ants will execute ACS algorithm during 700 cycles forming tours of 30 cities. When the first stage is finished, we select a subset of solutions (denoted by EI) containing a quantity (cs) of the best solutions (tours with shortest distance) found in the first stage.

A local search strategy is introduced to improve the results of the first stage which explores small regions of the solution space. This means, the partial solutions in the set EI resulting from stage 1 are improved using the well known 2-opt procedure [10]. The TSP 2-exchange neighbouring of a candidate solution

s consist of the set of all solutions s^* that can be obtained from s by exchanging two pairs of arcs in any possible way [5].

This improved EI set provides the initial states for the second stage. This means that in every cycle of the second stage each ant chooses, in a random way, an element of EI as an initial tour and positions itself in a randomly selected city belonging to this tour. After that, the ant will add other nc_2 cities to this tour.

The TS-ACS-TSP algorithm is given below: Given the parameters (beta, rho, epsilon, cc, factor r, number of solutions in EI (cs))

P0: Define the quantity of ants (m) either using input data or by using some method depending on the number of cities.

P1: Stage 1.

P1.1: Calculate the parameters for the first stage:

$$m_1 = r \cdot m; \quad nc_1 = r \cdot nc; \quad c_1 = r \cdot cc$$

P1.2: Apply the ACS algorithm, which in the first stage develops nc_1 cycles.

P1.3: Set of tours \leftarrow Tours generated by ACS algorithm in the first stage.

P2: Stage 2.

P2.1: Calculate the parameters for the second stage:

$$m_2 = m - m_1; \quad nc_2 = nc - nc_1; \quad cc_2 = cc$$

EI \leftarrow Selecting the best cs solutions from Set of tours.

P2.2: Apply a local search to improve partial solutions in EI set.

P2.3: Apply the ACS algorithm, which in the second stage develops nc_2 cycles, using the elements of EI like initial states for the ants in the second stage.

5. Experimental Results

A comparative study of ACS and TS-ACS algorithms in the TSP was done by using publicly available (or benchmark) data base for this problem [7].

For these experiments we use the following parameter setting: $\rho=0.1$, $\xi=0.1$, a quantity of ants $m=10$, and a number of cycles $nc=1000$. Firstly, we studied the values $\{1, 3, 5\}$ and $\{0.3, 0.6, 0.9\}$ for β and q_0 respectively; all combinations of these values were analyzed. Table 1 shows the different values of β and q_0 with which the best solution was obtained for each database. These values β and q_0 were selected to develop the next experiments. The column "Best solution" indicates the best solution reported back for that problem in the corresponding database, and the column "BS with ACS" contains the best solution found by our implementation of the ACS algorithm.

Table 1: Best solutions obtained by using ACS algorithm.

Data base	Best solution (BS)	Best solution with ACS	Time (ms)	β	q_0
bays29.tsp	2020	2048	1062	5	0.6
berlin52.tsp	7542	7650	3801	5	0.6
rd100.tsp	7910	8670	18140	5	0.6
ch150.tsp	6942	6867	54001	5	0.6
kroA200.tsp	29368	32712	112982	5	0.6
tsp225.tsp	3919	4285	148351	3	0.9
a280.tsp	2579	2904	271179	3	0.9
lin318.tsp	42029	47328	398572	3	0.9
pcb442.tsp	50778	58265	973212	3	0.9
rat783.tsp	8806	12255	5081400	3	0.9

Secondly, we developed experiments to determine an adequate value for the ratio. The values $r = \{0.2, 0.25, 0.3, 0.4, 0.5\}$ were studied. Table 2 and 3 show the results averaged over 6 runs. The quality of the solution and the time cost were measured. Moreover, we made the comparison without using local search and using the 2-opt procedure to improve the solutions.

The comparison with respect to the quality of the solution is presented in Table 2. The column “BS with ACS plus 2-opt” contains the best solution found by the ACS algorithm plus a local search developed with the 2-opt procedure; you can see the local search improves the quality of solution reported back in Table 1 (column “Best solution with ACS”). The column “BS with TS-ACS” contains the best solution found by TS-ACS-TSP algorithm (without step P2.2) and the corresponding ratio with which it was found; the column “BS with TS-ACS plus 2-opt after first stage” contains the best solution found by TS-ACS-TSP algorithm plus a local search developed with 2-opt procedure applied to the solutions obtained after stage 1 (that is, including step P2.2). In this case, the local search also improves the TS-ACS-TSP algorithm. Finally, the column “BS with TS-ACS plus 2-opt after both stages” contains the best solution found by TS-ACS-TSP algorithm plus a local search developed with 2-opt procedure applied to the solutions obtained after each stage; after the first stage, the local search improves the EI set (step P2.2), and after step P2.3 improves the resulting solutions of the second stage. These are the closest to the best solutions and its computational cost is lower than the ACS algorithm. The time cost to obtain these solutions is presented in Table 3 (measured in seconds). The TS-ACS-TSP algorithm significantly saves time.

Table 2. A comparison between ACS and TS-ACS in TSP (quality solution)

Data base	Best solution	BS with ACS plus 2-opt	BS with TS-ACS	BS with TS-ACS plus 2-opt after first stage	BS with TS-ACS plus 2-opt after both stages
bays29.tsp	2020	2041	2058 (0.25)	2045 ($r=0.2$)	2026 (0.2)
berlin52.tsp	7542	7564	8034 (0.25)	7863 (0.25)	7542 (0.25)
rd100.tsp	7910	8345	8668 (0.2)	8658 (0.3)	8159 (0.2)
ch150.tsp	6528	6673	6932 (0.3)	6896 (0.3)	6670 (0.2)
kroA200.tsp	29368	31598	32994 (0.2)	32889 (0.2)	31970 (0.25)
tsp225.tsp	3919	4115	4357 (0.25)	4204 (0.25)	4102 (0.2)
a280.tsp	2579	2827	3002 (0.2)	2944 (0.25)	2878 (0.25)
lin318.tsp	42029	46006	47370 (0.25)	47003 (0.3)	46086 (0.2)
pcb442.tsp	50778	56861	58420 (0.2)	58369 (0.3)	56790 (0.2)
rat783.tsp	8806	11917	12181($r=0.3$)	12202 (0.25)	11931(0.3)

Table 3. A comparison between ACS and TS-ACS in TSP (time cost)

Data base	BS with ACS plus 2-opt	BS with TS-ACS	BS with TS-ACS plus 2-opt after first stage	BS with TS-ACS plus 2-opt after both stages
bays29.tsp	1231	468	606	651
berlin52.tsp	3937	1672	1719	1734
rd100.tsp	18721	10609	12492	14202
ch150.tsp	57803	20122	21984	31874
kroA200.tsp	123852	64588	66781	74873
tsp225.tsp	155151	74255	76120	78677
a280.tsp	295464	126823	135987	146898
lin318.tsp	430637	154963	172492	190220
pcb442.tsp	1040741	396250	425612	478066
rat783.tsp	5315734	2262624	2490555	2983343

6. Conclusions

We have presented an improvement of the Ant Colony Optimization based on a multilevel approach. It consists of splitting the search process developed by ants into two stages. The values of some parameters (number of ants, quantity of cycles, number of cities included in each stage, etc.) are assigned a different value in each stage according to a ratio which indicates what proportion of the complete search corresponds to each stage.

The experimental results using the Travelling Salesman Problem can be summarized in the following way: (i) In most of the cases TS-ACS algorithm reaches better solutions than ACS. In case ACS yields the better result, the difference to the results obtained by TS-ACS is not relevant; (ii) The time in which TS-ACS algorithm gets those results is about between 40% and 50% of

the time needed by the ACS algorithm, that is, the cost in time is reduced by half; (iii) A value of rate r between 0.2 and 0.35 produces the best results; greater values for the ratio r decrease the time cost but also the quality (because the search space is not sufficiently covered); (iv) The introduction of a local search developed with the 2-opt procedure improve the performance of this approach.

Acknowledgment

The authors would like to thanks VLIR (Vlaamse InterUniversitaire Raad) for supporting this work under the IUC Program VLIR-UCLV.

References

- [1] S.T. Bernard, and H.D. Simon. A fast multilevel implementation of recursive spectral bisection for partitioning unstructured problems. *Concurrency: Practice and Experience* 6 (2), pp. 101-117. 1994.
- [2] M. Dorigo. Ottimizzazione, apprendimento automatico, ed algoritmi basati su metafora naturale. PhD thesis, Dipartimento di Elettronica, Politecnico di Milano, Milan, Italy (1992).
- [3] M. Dorigo et al. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics-Part B*, vol. 26, no. 1, pp. 1-13. 1996.
- [4] M. Dorigo, and L.M. Gambardella.. Ant colonies for the traveling salesman problem. *BioSystems* no. 43, pp. 73-81, 1997.
- [5] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press. 2004.
- [6] P. Korosec et al. Solving the mesh-partitioning problem with an ant-colony algorithm. *Parallel Computing* 30, pp. 785-801. 2004.
- [7] The TSPLIB Symmetric Traveling Salesman Problem Instances, <http://elib.zib.de/pub/Packages/mp-testdata/tsp/tsplib/index.html>
- [8] C. Walshaw. A multilevel approach to the traveling salesman problem. *Operations Research* 50(5), 862-877, 2002.
- [9] C. Walshaw et al. The multilevel paradigm: a generic Meta-heuristic for Combinatorial Optimization Problems? Technical Report 00/IM/63, Univ. Greenwich, London SE10 9LS, UK. 2000.

Pattern Knowledge for Proof-Number Search in Computer Go

Jahn-Takeshi Saito Guillaume Chaslot Jos W.H.M. Uiterwijk
H. Jaap van den Herik

*MICC/IKAT Universiteit Maastricht
P.O. Box 616, 6200 MD Maastricht, The Netherlands*

Abstract

Over the years, proof-number search has successfully been applied to many game domains. This article proposes two new pattern-based heuristics for move ordering in proof-number search. One heuristic applies patterns directly, the other heuristic uses patterns to control Monte-Carlo sampling. The test domain is the Life-and-Death problem in the game of Go. Experimentally, we found some advantages in memory consumption indicating that the pattern heuristics are promising. Future research will prove the patterns' quality for proof-number-search applications.

1 Introduction

Proof-number search (*PNS*, [2]) is a tree-search algorithm for finding binary goals, i.e., goals which can either be reached (a proof) or not reached (a disproof). *PNS* performs search on AND-OR trees, in which AND and OR levels alternate. This characteristic makes *PNS* an attractive choice for the combinatorial games domain. *PNS* algorithms have enjoyed both prolific improvements and popularity in practical combinatorial-game applications since they were introduced about a decade ago.

The improvements suggested to extend standard *PNS* can be divided into two categories. The first category contains procedural and implementational refinements. These often yield speed gains on a sub-algorithmic level (e.g., [12], [10]). The second category contains heuristic extensions of *PNS* (e.g., [13], [14], [7]). Such extensions may help to order branches to reach the proof or disproof with fewer node expansions. For instance, Monte-Carlo (*MC*) sampling ([1]) can successfully be applied as a move-ordering heuristic by an algorithm termed Monte-Carlo Proof-Number Search (*MC-PNS*, [11]).

This article contributes to the line of research on move-ordering heuristics for *PNS*. It investigates the question: what methods are suitable to extend *PNS* and *MC-PNS*? To this end, domain-specific knowledge is included in *PNS* and *MC-PNS*. The domain chosen as test bed is Go ([4], [9]). In computer Go, pattern knowledge has been accepted as a successful means for representing and applying domain knowledge (e.g., [9]). So, we propose to extend *PNS* and *MC-PNS* by pattern knowledge in order to assess the appropriateness of pattern extensions of *PNS* variations. The patterns proposed in this work are automatically generated in a statistical evaluation of self-play ([5]). They offer the flexibility to be used in a Monte-Carlo sampling context as given by *MC-PNS*.

The course of this article is as follows: Section 2 outlines the *PNS* and *MC-PNS* algorithms as far as required for understanding the pattern integration. Section 3 introduces the pattern-generation process and formulates two pattern heuristics for *PNS* and the *MC-PNS* framework. Section 4 describes the experimental setup for testing the pattern extension. Section 5 presents the experimental results and discusses them. Section 6 concludes by summarizing the findings and giving an outlook to future work.

2 Proof-Number Search

This section describes the basic *PNS* algorithm (section 2.1) and explains how move-ordering heuristics can be added to *PNS* (section 2.2). An example of such a heuristic, *MC* search, is given in section 2.3.

2.1 Standard proof-number search

PNS is a best-first search algorithm for proving or disproving binary goals in AND-OR trees. For OR-level nodes, proving one successor node is sufficient to prove the goal. For proving AND-level nodes, all successor nodes must be proved. For disproving OR- or AND-level nodes the corresponding reverse holds. *PNS*'s best-first heuristic is defined as follows: expand first the node that promises to require the least amount of further node expansion to prove or disprove the goal.

A *proof number* (pn) and a *disproof number* (dn) are introduced for each node M . They keep record of which node is most promising in the whole search tree. Following the heuristic above, pn is the number of nodes which are at least required to prove the goal at M . Analogously, dn is the number of nodes which are at least required to disprove the goal starting at M . We denote the corresponding pn and dn for some node X by $pn(X)$ and $dn(X)$ if required by the context.

The *PNS* algorithm has two stages which are repeated during the search. In the first stage, *PNS* descends from the root node to a leaf node with lowest pn or dn . The numbers pn and dn do not increase along the path. The smallest number is therefore a static characteristic of the path's leaf node. In OR nodes, the branch with smallest pn is chosen for descending. In AND nodes the branch with smallest dn is selected. When the leaf node (L) with smallest value is found, L is expanded. The values of the newly expanded children are set as follows: $pn = 0$ and $dn = \infty$ in case the goal is proved, $pn = \infty$ and $dn = 0$ in case the goal is disproved, $pn = 1$ and $dn = 1$ otherwise. According to the number of its children $pn(L)$ and $dn(L)$ are set. The second stage tracks back from the leaf node to the root node. Starting at L , pn and dn are adjusted for all predecessors following the path back up to the root node.

The repetition of this cycle terminates when $pn(\text{root}) = 0$ or $pn(\text{root}) = \infty$. In the first case, the goal is proved as no further node requires to be expanded. In the second case, a disproof is reached because an infinite number of expansions would be required to prove the goal. In conclusion, *PNS* produces the correct solution but requires maintenance of a large search tree.

2.2 Heuristic move ordering for *PNS*

The most common approach for adding a heuristic evaluation function to *PNS* is applying the heuristic in each newly expanded leaf node. So far, for each newly expanded non-terminal node X , *PNS* sets $pn(X) = dn(X) = 1$. These default assignments can be replaced by a heuristic evaluation. We suggest the following setting: $pn(X) = h_{pn}(X)$ and $dn(X) = h_{dn}(X)$. Here, h_{pn} denotes a heuristic estimate for reaching a proof and h_{dn} denotes a heuristic estimate for reaching a disproof.

h_{pn} and h_{dn} are propagated through the proof tree by *PNS*'s rule for back-propagation. The only constraint for maintaining the algorithm's correctness is that the values of $h_{pn}, h_{dn} > 0$. For reasons of consistency and efficiency, it seems desirable to set h_{pn} anti-proportional to h_{dn} (e.g., $h_{pn}(X) = 100$ and $h_{dn}(X) = 3$). Since pn s and dn s are propagated independently of each other, this is not required. In practice, h_{pn} and h_{dn} can be derived from the same evaluation.

2.3 Monte-Carlo Proof-Number Search

An example of a heuristic function for move ordering is *MC* search. The heuristic is based on *MC* sampling ([1]). In combinatorial games, *MC* sampling follows a simple principle. Random play is considered in order to gain an evaluation for a game state S under investigation. The board configuration of a board game constitutes such a game state. Random play consists of a random sequence of legal moves. Each sequence is evaluated, e.g., by scoring the sequence's end position. The value of S is a statistical aggregate of all random sequence values. The mean of the scores of all random sequences is an example for such an aggregate.

Applying *MC* sampling as a heuristic in *PNS* is called *MC-PNS*. *MC-PNS* launches N random games with a fixed number of moves from the leaf node X to be evaluated. The statistical evaluation performed is a ratio. It takes into account the number of sequences yielding a proof (N_+) for setting the value of dn . It takes into account the number of sequences yielding a disproof (N_-) for setting the value of pn . It may seem paradoxical that counting fewer sequences proving the goal corresponds to the value of a greater proof number. The underlying rationale is that the proof number represents the minimal number of nodes which require expanding. Thus, a smaller proof number represents better chances of reaching the goal soon.

More formally the heuristic is $h_{pn}(X) = \frac{N_-+1}{N}$ and $h_{dn} = \frac{(N_++1)}{N}$.

Previous work ([11]) yielded that *MC-PNS* doubles the speed of the search process for Life-and-Death problems (cf. section 4.1) and reduces the number of visited nodes to a quarter compared to standard *PNS*.

3 Computer-Go patterns for *PNS*

This section highlights the relevance of patterns in computer Go (section 3.1) and introduces the patterns (section 3.2) employed in the experiment described in section 4. Section 3.3 describes pattern-based heuristics for *PNS* and *MC-PNS*.

3.1 Patterns in computer Go

Patterns are a standard means for representing knowledge in computer Go (e.g., [8], [3], [4]). A pattern in computer Go is a configuration of intersections. High-level representations contain features additional to the colouring of the intersections. Two examples are: (1) tactic information on connectivity and (2) Life-and-Death status. The sizes of a pattern vary considerably depending on the application's purpose. Large-scaled patterns are employed for openings, while tactical analysis is often guided by small patterns. Patterns can be hand-tuned or auto-generated ([3], [6], [15], [5]).

3.2 3×3 patterns

The patterns applied in the experiment described in section 4 are auto-generated 3×3 patterns. They are small patterns, but they offer the advantage of low pattern-matching costs. This characteristic is a precondition to any application in the Monte-Carlo framework as given by *MC-PNS*. The patterns should be matchable each move in each simulated game. The patterns are generated by statistical ranking in self play as described in [5]. Each of the 6561 patterns describes the desirability of the move in the centre expressed by an integer value. The patterns do not account for information on the edge of the Go board. They were generated for the whole game and are not tailored to Life-and-Death problems (cf. section 4.1).

3.3 Two pattern-based heuristics

The patterns described in section 3.2 are the ingredients for two heuristic variations of *PNS*. We refer to them as (1) PNS_p and (2) $MC-PNS_p$.

First, PNS_p applies the patterns to a node X by assigning the suggested pattern value of the last played intersection to $pn(X)$ and $dn(X)$. The desirability depends on the neighbours surrounding the intersection last played. The intersection's pattern value v ranges between 0 and 340. The values of pn and dn are set as $pn(X) = \frac{1}{v+1}$ and $dn(X) = \frac{v}{340}$. Local knowledge on the desirability of the move last played is thus taken into account.

Second, $MC-PNS_p$ is to be considered as a variation of *MC-PNS* (cf. section 2.3). The patterns are employed to alter the probability distribution of the moves selected for the random sequence. Pure *MC-PNS* plays randomly distributed legal moves in each random sequence implying a uniform distribution of the probability for selecting a move. In $MC-PNS_p$ this uniform distribution is altered by the pattern values. Whenever a move is played in a random sequence, the pattern values of the intersections next to that move are updated. These neighbours' new pattern values are set according to a matching pattern. The probability to play at a neighbour is proportional to its

pattern value. Thus, moves evaluated well by the patterns are more likely to be played than moves less promisingly assessed.

The intuition underlying both heuristics is that main lines of play should be considered first. These lines are given preference by the pattern evaluation introduced. Comparing the two heuristics, we may predict that $MC-PNS_p$ requires considerably more time for pattern matching because patterns are matched for each move in each random sequence. PNS_p matches the patterns only once at each new expansion.

4 Experimental Setup

The experimental setup is described below. The test set is in section 4.1, the machinery used in section 4.2, and the experimental procedure followed is in section 4.3.

4.1 Test set

In the past, PNS variations were employed for solving instances of the Life-and-Death problem which is a frequently occurring sub-problem in Go games. The problem consists of finding a sequence of moves by which either one of the opponent's groups is circumvent and thus captured, or by which an own group is prevented from being circumvent by the opponent. For instance, Black moves first and has to determine the group's status as either alive, dead, ko or seki. For the current investigation, however, the problem is reduced to a binary classification of the target group to be either alive or dead. The test set consists of 30 Life-and-Death problems. We remark that search for solving the Life-and-Death problem has been addressed before (e.g., [16]). The general Life-and-Death problem consists of a locally bounded game position with a target group of stones.

In order to fit the algorithm to the Go domain, the goal to prove, i.e., the status of a group of stones as either alive or dead, is checked by a simple status-detection function. This function is called by each simulated move to determine whether to play further moves or stop expanding. Each random sequence halts after either the goal has been reached or a depth of ten has been arrived at. Five random sequences are launched to evaluate a position. These parameter settings are in accordance with earlier findings ([11]).

4.2 Machinery

The tested algorithms were implemented in a C++ framework. All experiments were conducted on a Linux workstation with AMD Opteron architecture and 2.8 GHz clock rate. 32 GB of working memory were available. The pattern matcher was implemented following the descriptions of section 3.1. A simple function was added to detect the Life-and-Death status of a position. Tree search and MC evaluation are carried out in a brute-force manner. In the experiment, Zobrist hashing ([17]) was implemented to store proof and disproof numbers estimated by the MC evaluation. The game positions are not stored in the nodes in order to save memory. Thus, a complete sequence is played in each expansion stage (cf. section 2.1). The proof tree is fully stored in memory. The speed of the MC engine is about 5,000 full game sequences per second without and about 4,500 full game sequences per second with pattern matcher.

4.3 Experimental procedure

The four variations of the PNS algorithm outlined in section 2.1 were tested: (1) PNS (without heuristics), (2) PNS_p (PNS with pattern heuristic), (3) $MC-PNS$, and (4) $MC-PNS_p$ ($MC-PNS$ with pattern heuristic). Each variation was tested on the test set of 30 Life-and-Death problems described in section 4.1. The time spent and the number of nodes expanded for solving each test position were measured for each variation. The number of nodes is an indication of efficiency. Simultaneously, the memory consumption is measured by the same number of nodes. In order to account for the randomness of the MC evaluations and potential inaccuracy for measuring the small time spans well below a 10th of a second, each configuration was applied to each test case 20 times.

Algorithm	Time rank	Space rank
<i>MC-PNS</i>	1.63	1.63
<i>MC-PNS_p</i>	2.6	1.6
<i>PNS</i>	2.7	3.6
<i>PNS_p</i>	3.0	3.12

Table 1: Time and space ranking of the four compared *PNS* variations averaged over 30 test problems.

5 Results and Discussion

In this section, results are outlined (section 5.1) and discussed (section 5.2).

5.1 Results

The experiment required about six hours to complete. All variations solved all problems correctly. The variations that solved the test cases in decreasing order of average speed were as follows: first *MC-PNS*, second *MC-PNS_p*, subsequently *PNS*, while *PNS_p* required most time to solve the problems.

We note that on each test case a ranking can be established for a variation. Each variation then ranks either first, second, third, or fourth in terms of speed performance. Similarly there is a rank for space consumption. The average ranking is introduced to compare the performance of the variations on all tests. (The average rankings for finding a solution are depicted in table 1.)

The average ranking by time consumption yields roughly the same order as the ranking by space consumption, with one significant exception: *PNS* solves the problems faster than *PNS_p* on average (in 17 out of 30 cases), but *PNS* consumes more space than *PNS_p* (in two thirds of the cases). However, it should be remarked that there are two large outliers to this finding.

The results can be structured by noting the explicit occurrence of two features, the *MC* feature and the pattern feature. The *MC* feature is present in *MC-PNS* and *MC-PNS_p*; the pattern feature is present in *PNS_p* and *MC-PNS_p*.

The speed of *MC-PNS* variations was about twice as high as for *PNS* on average. Patterns slowed down the speed for finding a solution in both, *PNS* and *MC-PNS*, whenever they were applied. This tendency was particularly noticeable in *MC-PNS_p*. The variations relying on patterns expanded on average about 6% fewer nodes for each solution but the variation between the feature groups was larger than within each feature group as can be seen in figure 1. Analogously, the variation of time consumption is smaller between the feature groups than within. The internal variation of time consumption is larger within the *MC* feature group than within the pattern feature group. The reverse holds for space consumption.

5.2 Discussion

The findings of the previous section can be summarized in three observations. (1) The variation within the feature groups is smaller than between the feature groups with the *MC* heuristic yielding much better results than the pattern heuristic. (2) The patterns slow down the speed of the solver and particularly hinder it in case of *MC-PNS_p*. (3) There is a positive impact of the pattern heuristic, evident in the better space ranking of *PNS_p* compared to that of *PNS*.

These three observations can be explained as follows. (1) The patterns' positive effect is outweighed by the cost of matching they impose. (2) While the advantage can be seen in reduced space consumption, its pruning effect is too weak to result in time savings. (3) The outliers can be explained by the nature of pattern knowledge. That must be regarded as static and impedes those search branches multiply. This must require exceptions from the patterns generally recommendable. The effect is particularly strong because the patterns are not specialized on Life-and-Death problems (section 3.2). So, the combination of patterns and *MC* evaluations in the manner proposed does not seem promising. However, overall the results, though not practically applicable in the current state, can still be seen as encouraging for future work since several opportunities for improvement exist (see section 6).

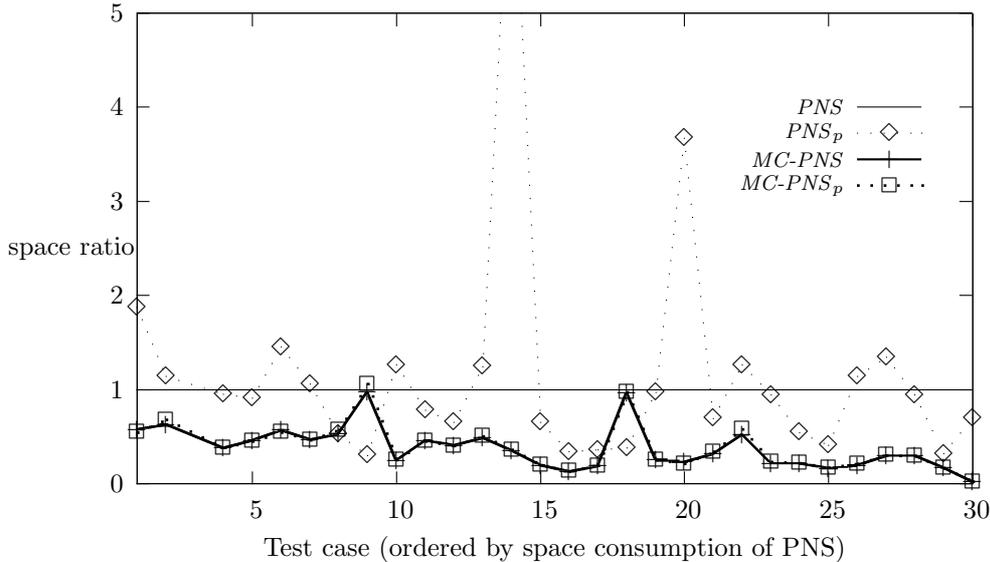


Figure 1: Space consumption of the variations expressed as ratio compared to the *PNS* benchmark on all 30 test cases.

6 Conclusion and future work

This work described an experiment assessing the suitability of small patterns for *PNS* in a subdomain of computer Go. *PNS* variations applying patterns with and without *MC* evaluation were introduced. Difficulties in pattern heuristics for *PNS* were revealed. A positive effect could be measured in spite of unspecifically trained and easily extendable patterns. Therefore, we may conclude that the application of patterns should be explored further in future research.

Meanwhile, much space is left for improvements. (1) Future investigations should consider specially trained patterns with border information. The patterns could be trained either on Life-and-Death problems or on Go endgames which resemble characteristics of Life-and-Death problems and are easily available. (2) The combination of *MC* sampling and patterns may take different forms than the ones proposed. (3) It remains promising to test whether pattern knowledge can be useful in the upper levels of the search tree in which unspecific patterns can be expected to be successful. (4) Search at deeper levels can be guided by *MC* sampling.

7 Acknowledgements

We thank Bruno Bouzy for his willingness to share the patterns developed in co-operation with him. This work is financed by the Dutch Organization for Scientific Research NWO as part of the project GO FOR GO, grant number 612.066.409.

References

- [1] Bruce Abramson. Expected-outcome: a general model of static evaluation. In *IEEE Transaction on PAMI*, volume 12, pages 182–193, 1990.
- [2] L. Victor Allis, Maarten van der Meulen, and H. Jaap van den Herik. Proof-number search. *Artificial Intelligence*, 66:91–124, 1994.
- [3] Bruno Bouzy. Go patterns generated by retrograde analysis. In Jos W.H.M. Uiterwijk, editor, *The CMG Sixth Computer Olympiad. Computer-Games Workshop*, 2001.
- [4] Bruno Bouzy and Tristan Cazenave. Computer Go: An AI oriented survey. *Artificial Intelligence*, 132(1):39–103, 2001.

- [5] Bruno Bouzy and Guillaume Chaslot. Monte-Carlo Go reinforcement learning experiments. In *Computational Intelligence in Games*, 2006.
- [6] Thore Graepel, Mike Goutrie, Marco Krüger, and Ralf Herbrich. Learning on graphs in the game of Go. In *International Conference on Artificial Neural Networks (ICANN-01)*, 2001. Vienna, Austria.
- [7] Tomoyuki Kaneko, Tetsuro Tanaka, Kazunori Yamaguchi, and Satoru Kawai. Df-pn with fixed-depth search at frontier nodes. In Hitoshi Matsubara, editor, *10th Game Programming Workshop in Japan*, pages 1–8, 2005. Hakone, Japan. In Japanese, abstract in English.
- [8] Takuya Kojima and Atsushi Yoshikawa. Knowledge acquisition from game records. In Johannes Fürnkranz and Miroslav Kubat, editors, *Workshop Notes: Machine Learning in Game Playing*. 1999.
- [9] Martin Müller. Computer Go. *Artificial Intelligence*, 134:145–179, 2002.
- [10] Ayumu Nagai. *Df-pn Algorithm for Searching AND/OR Trees and Its Applications*. PhD thesis, University of Tokio, 2002.
- [11] Jahn-Takeshi Saito, Guillaume Chaslot, Jos W.M.H. Uiterwijk, and H. Jaap van den Herik. Monte-Carlo Proof-Number Search for Computer Go. In H. Jaap van den Herik, Paolo Ciancarini, and Jeroen Donkers, editors, *Computers and Games 2006*. International Computer Games Association, 2006.
- [12] Masahiro Seo, Hiroyuki Iida, and Jos W.H.M. Uiterwijk. The PN*-search algorithm: application to tsume-shogi. *Artificial Intelligence*, 129(1-2):253–277, 2001.
- [13] Seiichi Tanaka, Iida Hiroyuki, and Yoshiyuki Kotani. An approach to tsume-shogi: Applying proof-number search with estimation function of mating. In Hitoshi Matsubara, editor, *Game Programming Workshop in Japan '95*, pages 138–147, Kanagawa, Japan, 1995. In Japanese, abstract in English.
- [14] Seiichi Tanaka and Yoshiyuki Kotani. Checkmate search with checkmate estimation function. In Hitoshi Matsubara, editor, *Game Programming Workshop in Japan '96*, pages 141–149, Kanagawa, Japan, 1996. In Japanese, abstract in English.
- [15] Eric C. D. van der Werf. *AI techniques for the game of Go*. PhD thesis, Universiteit Maastricht, 2004.
- [16] Thomas Wolf. Forward pruning and other heuristic search techniques in tsume Go. *Information Sciences*, 122:1:59–76, 2000.
- [17] Albert L. Zobrist. A new hashing method with application for game playing. *ICCA Journal*, 13(2):69–73, 1990.

ENABLING MEDICAL EXPERT CRITIQUING USING A BDI APPROACH

Robert-Jan Sips^a

Loes Braun^a

Nico Roos^a

^a*Department of Computer Science, Universiteit Maastricht
P.O.Box 616, NL-6200 MD Maastricht*

Abstract

Expert critiquing systems were introduced to assist physicians in decision making, without forcing them to comply to a gold standard of care. Critiquing systems do this by providing critique on a physician's decisions, rather than telling him/her exactly what to do. In order to perform this task, a critiquing system must have knowledge of the diagnosis and the treatment processes, and must be able to link the actions performed by a physician to this knowledge.

The development of *formal languages for describing medical guidelines* (protocols) and the nationwide introduction of *electronic patient records* (EPR) in the Netherlands, facilitates the development of a new generation of medical critiquing systems. Essential to the success of the new generation critiquing systems is the ability to match the actions prescribed in a medical guideline to the physician's actions reported in the EPR. Some authors have claimed that such a matching process is infeasible. This paper will show, however, that a BDI (*beliefs, desires and intentions*) approach enables a highly successful matching process thereby enabling expert critiquing based on an EPR.

1 Introduction

Expert critiquing systems Expert critiquing systems were introduced to assist physicians in decision making, without forcing them to comply to a gold standard of care [12, 19]. Expert critiquing systems do this by providing critique on a physician's decisions, rather than telling him exactly what to do [12]. The first expert critiquing systems developed by Miller [12] relied heavily on user interaction, which resulted in rejection of his systems [8]. To avoid this problem, Van der Lei [19] proposed a system which gathers all its information from computer-stored medical records, critiquing a physician without relying on user interaction. The current developments, such as the introduction of an electronic patient record (EPR) in the Netherlands support this proposal.

A major problem with the critiquing systems proposed by Miller and Van der Lei is that these systems cannot cope with deviations from the underlying model. Moreover, the systems were not able to deal with the question why a physician was performing an action, which is essential to provide a grounded critique. To address both problems, Shahar *et al.* [16] and Advani *et al.* [1, 2] suggest to perform critiquing by assessing the compliance of a physician's intentions with the intentions behind a medical guideline. They also stress that plan recognition is an indispensable prerequisite for the performance of critiquing.

Marcos *et al.* [10] claim that matching actions performed by a physician to those prescribed by the guideline is very difficult. Moreover, they claim that the direct use of intentions for critiquing, either for matching of actions or for studying their appropriateness, is not possible. According to Marcos *et al.*, this is caused by difficulties inherent to modeling and acquisition of these intentions. The intentions of the guideline are not stated explicitly, which makes them very hard to model, and the intentions reported by experts almost always differ. It is not only a problem of vocabulary, but also a matter of differences in the degree of detail, abstraction level, etc.

Medical guidelines Medical guidelines were introduced to standardize medical practice. The use of medical guidelines has been shown to reduce practice variations [9], improve practice quality [17] and improve the cost efficiency of medical care [11]. To improve medical guidelines [11] and to make medical guidelines computer-understandable, formal languages [14] for describing medical guidelines have been developed.

The most-developed examples [18] of these formal languages are ASBRU, EON, GLIF, GUIDE, PRODIGY and PROFORMA¹.

Since the guidelines standardize medical practice, their introduction has faced great resistance from the medical field [4]. Physicians refer to standardized care as cookbook medicine [21], hampering the intellectual process of treatment and diagnosis. In contrast, expert critiquing systems only give feedback on the physicians decisions rather than prescribing actions. However, expert critiquing systems could greatly benefit from medical guidelines because the new evidence-based guidelines reflect the latest of medical knowledge in the field [21].

Beliefs Desires and Intentions (BDI) To provide a physician with proper critique based on medical guidelines, we must be able to match the physician's actions reported in the EPR to those prescribed by the medical guideline. However, directly matching executed actions with prescribed actions is hard. As was pointed out by Woolf *et al.* [21], medical practice variations are mostly caused by factors as the treating physician, the hospital or the geographical location. For instance, when a physician prescribes a short working bloodglucose-lowering substance, e.g., Velosulin, the intention of lowering the blood-glucose level could just as well be served by prescribing Humalog, or Novorapid. In other words, one physician may prefer Velosulin, because it is available in the local drug store, or because he has good experiences with it, while the other, who is, for example, located 20 miles south, may prefer Novorapid, for the same reasons. Hence, we should not focus on the prescribed and reported actions but on the *intentions* behind these actions. This implies that the language for describing medical guidelines should support the description of intentions in a computer interpretable way. ASBRU, EON, GUIDE and PROFORMA are the only languages which specify intentions formally, from which ASBRU has the most extensive intention-modeling capacity [13].

Choosing an intention-based approach, we can step by step refine the general *intention* of curing a patient via the general *intentions* of diagnosis and treatment to specific tests and medical interventions. Medical guidelines are used to make these refinement steps. Since the medical guidelines may contain choices, different options (often denoted as the *desires* [20]) may be generated. A physician following the medical guidelines should have chosen one of the *options*, thereby creating a new refined *intention*. Intentions are refined chronological (in a lazy way) based on observations and the results of the tests performed (the *beliefs*). Hence, the application of intention-based medical protocols is in fact an application of a *belief*, *desires* and *intention* approach [5, 15, 20].

In the next section we will show how the BDI approach can be used to identify the diagnosis and treatment guidelines that match the current physician's treatment process reported in the EPR as well as the current location in the guideline. The key element of the process will be the intentions behind executable actions.

2 Intention-based diagnosis and treatment identification

Intentions Medical actions vary from medicine prescriptions to blood tests and from examining x-rays to surgical interventions. All of these actions can be used in a different context within different treatments: measuring one's blood type (referred to in literature as an ABO test [3]) could be used in a guideline containing a blood transfusion to determine the applicability of the blood for transfusion. However, in a guideline for Jaundice in newborns, the guideline can prescribe the same action to determine the likeliness of a blood-group antagonism in the newborn, i.e., the likeliness that the newborn has antiserum to its own blood [6]. Because the intention of an action can change based on the context in which the action is performed, it is hard to model the possible intentions behind an action.

When we reexamine the example above, we see that in all cases the ABO test is used to determine the patients blood group. This is a context independent intention of this action. Based on these observations, we define two types of intentions:

- High-level intentions specifying an intention of an action in a specific context, e.g., determine possibility of blood group antagonism.
- Low-level intentions specifying a context independent intention of an action, e.g., determine blood group.

¹Originally denoted as *Proforma* [18].

The high-level intentions relate to the intentions of medical guidelines and low-level intentions relate to a physician's actions and to actions in medical guidelines. In the following paragraphs we will explain how we can exploit the relation between high-level and low-level intentions and medical guidelines.

Applying BDI to medical guidelines Based on the physician's actions reported in the EPR, the critiquing system must determine to what extent the guideline has been followed by the physician in order to provide adequate critique. Hence starting from a general high-level intention such as "diagnose and treat hyperbilirubinemia", the critiquing system must search through the possible execution of the guideline. We interpreted this as a refinement of intentions in a BDI approach of practical reasoning. We will illustrate the BDI approach using parts of the guideline for diagnosis and treatment of hyperbilirubinemia, which was developed in the Asgaard project [11]. The critiquing system starts with searching for a plan to refine the high-level intention "diagnosis and treatment hyperbilirubinemia".

```

plan
Diagnosics-and-treatment-hyperbilirubinemia
  intentions
    avoid intermediate state: (bilirubin = transfusion)
  conditions complete-condition:
    (or (jaundice-clinically-significant = no)
        explanation ...
        (pathologic-reason = yes))
  plan-body type = sequentially
    wait-for none
    ask term-child
    ask age-child
    Diagnostics-hyperbilirubinemia
    Treatment-hyperbilirubinemia

```

On inspection of the plan, the critiquing system will subsequently check the EPR whether the condition "jaundice-clinically-significant = no" is among the preliminary reported *beliefs*. If this is not the case, it searches for a plan that realizes the high-level intention "is-known-parameter (jaundice-clinically-significant)". The following plan realizes this intention.

```

plan Jaundice-determination
  intentions achieve overall-state:
    is-known-parameter(jaundice-clinically-significant) in NOW
  plan-body type = sequentially
  ...

```

After having refined the high-level intention behind the conditions, the critiquing system continues with the plan body. When processing the plan body, it also treats the sub-plans as high-level intentions. Hence, the high-level intention "Treatment-hyperbilirubinemia" is realized by the plan:

```

plan Treatment-hyperbilirubinemia
  intentions
    avoid intermediate-state: (bilirubin = transfusion)
    achieve overall-state: (bilirubin = observation)
  plan-body type = parallel
    wait-for one
    wait-for (or Regular-treatments Exchange-transfusion)
    Regular-treatments on-abort Exchange-transfusion
    Exchange-transfusion
    cyclical-plan
    do type = sequentially
      wait-for all
      ask TSB-value
      ask age-child
      retry-delay: min = 12h, max = 24h

```

Note that the plans that make up the medical guideline may contain choices in the plan conditions and in the plan body. These choices represent the *options* one of which a physician may *desire* to use. If the EPR does not indicate the choice made by the physician, the critiquing system must consider each option separately, resulting in a set of possible execution sequences of actions. The critiquing system must subsequently identify the *best matching* execution sequence w.r.t. the actions reported in the EPR.

Actions and low-level intentions To explain the trace of actions performed by a physician and reported in the electronic patient record, we have to match these actions with the actions prescribed by one of the execution sequences of a guideline. Directly matching executed actions with prescribed actions is hard. As was pointed out in the Introduction, medical practice variations are mostly caused by factors as the treating physician, the hospital or the geographical location [21]. Because of these seemingly arbitrary factors influencing a physician's acting, there is no point in trying to match guideline-prescribed and observed actions. Therefore, we replace both the guideline-prescribed actions and the observed actions by low-level intentions.

In the Netherlands, some types of actions are thoroughly documented and restricted by the health care insurance companies. For example, the types and brand names of medicines which can be prescribed in the Netherlands are documented in the Farmacotherapeutisch kompas (pharmacotherapeutic compass) [7]. The pharmacotherapeutic compass groups medicines with similar effects into pharmaceutical groups. For instance, short working bloodglucose-lowering substances such as: Velosulin, Humalog and Novorapid are in the same pharmaceutical group, but differ in brand names. In order to reason about the low-level intentions of a medicine prescription, we replace each medicine prescription by the corresponding pharmaceutical group. So the system can reason about, e.g., the prescription of a short working blood-glucose lowering substance, instead of the prescription of Velosulin. We believe that this translates more naturally to a physician's thinking. In a similar way, we have used the Merck-manual [3] for replacing each treatment- and diagnosis-action by a low-level intention. An example is the action "Measure temperature", which can be used for various (high-level) intentions, such as determining whether a patient has fever, or assessing whether a patient suffers from Hypothermia, etc. However, the low-level intention linked to this action is in all cases "is-known-parameter(temperature)". Using the medical literature mentioned above, it is possible to obtain a complete set of possible actions a physician might perform, linked with the possible low-level intentions that may be pursued by these actions.

Matching To identify the execution sequence of the guideline a physician is following, we define a measure of the distance between an action sequences prescribed by an execution of the guidelines and the action sequences reported in the EPR.

The **distance** is defined as (1) the number of actions in de EPR that are matched by the execution sequence of the guidelines minus (2) the number of unmatched actions in the execution sequence that occur before the last matched action in the execution sequence.

A match between an action prescribed by a guideline and a physician's action is defined as follows.

A **match** is an action performed by the physician with a low-level intention that is contained in either (1) the low-level intentions of one of the guideline actions, (2) one of the guideline's action effects or plan effects, (3) a guideline plan precondition.

In the following algorithm for matching the actions performed by a physician to those prescribed by a guideline, we assume that we know the most general high-level intention of the physician; e.g., "treat hyperbilirubinemia".

1. Let $e = \langle l_1, \dots, l_n \rangle$ be an execution sequence of low-level intentions realizing a high-level intention h . We can then define $E(h)$ to be **all** legal execution sequences for a high-level intention h . h is the most general high-level intention of the guideline.
2. Let $A = \langle a_1, \dots, a_m \rangle$ denote the physician's actions. Then $t(A) = \langle t_1, \dots, t_m \rangle$ is the corresponding low-level trace for the physician's actions.

We can now define the minimal execution sequence e that explains the maximal number of actions in A . Note that no two l s in $t(A)$ can be matched to the same l in e .

3. Let $M \subseteq \{1, \dots, m\} \times \{1, \dots, n\}$ be a mapping from the indexes of a trace $t = \langle t_1, \dots, t_m \rangle$ to the indexes of the execution sequence $e = \langle l_1, \dots, l_n \rangle$ satisfying the following four conditions:
 - $(i, j) \in M$ iff $t_i = l_j$;
 - for every $(i, j) \in M$ and $(x, y) \in M$: $i = x$ iff $j = y$;
 - if for no j : $(i, j) \in M$, then $t_i \notin \{l_y \mid \exists x : (x, y) \in M\}$;

- there is no mapping M' satisfying the first three items above such that:
 $|\{i \mid (i, j) \in M'\}| = |\{i \mid (i, j) \in M\}|$ and
 $\max(\{j \mid (i, j) \in M'\}) < \max(\{j \mid (i, j) \in M\})$.

The first item guarantees that there are only mappings between identical intentions. The second item guarantees that there are only one-to-one mappings. The third item guarantees that all possible intentions of a trace are mapped to an intention in an execution. Finally the last item minimizes the non-matched gaps in the execution.

4. Let $k = \max\{j \mid (i, j) \in M\}$. Then the measure of the match between a trace t and an execution e is defined as:

$$match(t, e) = |M| - |\{j \mid j < k, \nexists (i, j) \in M\}|$$

5. The best match for the physician's actions A is the execution $e \in E(h)$ for which $match(t(A), e)$ is maximal.

3 Experiments

To give a *proof of principle* of our approach described in the previous section, we implemented a prototype and evaluated the prototype using an ASBRU-modelled protocol, constructed in the Asgaard project [11]. This protocol models the diagnosis and treatment of *hyperbilirubinemia* or Jaundice in healthy newborns. The protocol is based on the Jaundice protocol of the American Association for Pediatrics (AAP), which is intended for the management of Jaundice in healthy term (defined as 37 completed weeks of gestation) newborns [11]. Jaundice is a common disease in newborn babies and is caused by an elevated blood-bilirubin level [11]. For an overview diagnosis and treatment, we refer to the Merck manual [3], page 2156.

Data of treatments were obtained from Marcos *et al.*, who provided us with the test data used in their critiquing experiment [11]. Marcos *et al.* obtained their data from a pediatrician addressing patient cases. Since the pediatrician consulted by Marcos *et al.* received the protocol prior to addressing the patient cases, we asked Twan Mulder MD PhD, who works as a pediatrician / neonatologist at Maastricht University Hospital to address patient cases without prior knowledge of the protocol. In this way, we gathered solutions to 12 cases. The pediatrician consulted by Marcos *et al.* provided solutions to 7 cases and T.Mulder provided solutions to case 5 cases.

We divided the practice samples obtained from Mulder and by Marcos *et al.* into test executions. A test execution is a prefix of the physician's solution, always starting at the first action. These test executions simulate the step by step updating of the EPR by a physician.

In the first experiment we measured the performance of our algorithm on all test executions based on the normal execution order. With this test we want to determine the following.

- The overall performance of our algorithm (which percentage of a sequence is correctly classified).
- Whether there is a significant difference in performance on the solutions of a physician who is familiar with the guideline used and one who is not.

Figure 1 shows the average results of matching these sequences with the protocol for all 12 cases.

In the second experiment we measured the performance of our algorithm on the solutions in a different order. This was done to obtain a measure of the performance on a real patient record, where the physician enters the information on consultation basis and where it is not guaranteed that the actions are entered in the order in which they are performed. To do so, we used two execution orders, reason by encounter (RBE) and backwards execution.

- In reason by encounter, we assumed that a physician performs 3 action per consult and that these 3 actions need not be entered into the electronic patient record in the order in which they were performed.
- Backwards execution was used to determine the performance of our algorithm on actions in the completely opposite order, i.e., backwards execution.

Based on these experiments, we statistically tested the differences between the chronological execution order, reasoning by encounter and backward execution to determine the dependence of our algorithm on the correct sequence of actions.

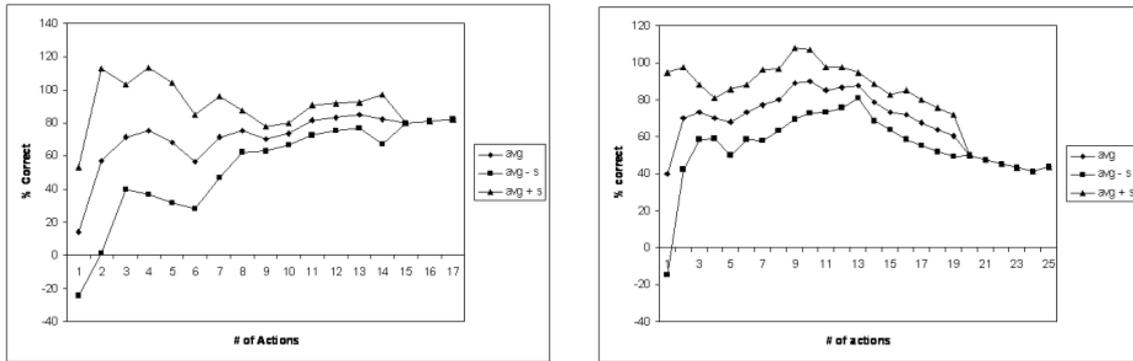


Figure 1: Average results of matching Marcos' and Mulder's actions respectively to the protocol. *avg* denotes sample mean, and *avg-s* and *avg+s* denote sample variance.

4 Discussion

On inspection of the experimental results when using the chronological execution order, we see that, on average, 69.4% of the actions in a test sequence are matched correctly. When we have a closer look at the misplaced² action types, we see that 19 out of the 27 misplaced action types cannot be matched due to incompleteness of the guideline. The guideline is incomplete with respect to the specification of treatment for several causes of hyperbilirubinemia. The following plan gives an illustration.

```

plan Check-for-rapid-TSB-increase
  intentions
    achieve overall-state: (and is-known-variable(possibility-of-G6PD)
      is-known-variable(possibility-of-hemolytic-disease))
  conditions
    filter-precondition: (and (TSB - decrease = no) in NOW
      (TSB - change > 0.5) in NOW)
  plan-body type = sequentially
    wait-for all
      possibility-of-hemolytic-disease <- yes
    if (age = day2) then
      possibility-of-G6PD <- yes
      Exit-possibility-of-G6PD
    else
      possibility-of-G6PD <- no
      Exit-possibility-of-hemolytic-disease

```

When we discard the actions which cannot be matched, we observe an average performance of 80.1%.

We can also see that the average performance on short test sequences is worse than on long sequences. This is caused by an error in the protocol constructed in the Asgaard project. The first action performed by the pediatrician is often "Measure TSB value". The algorithm can only match this action to a similar action in sequences where the pediatrician has already prescribed treatment. Looking at the Merck manual [3] and the Diagnostic Compass [6], we find that the TSB value (Total Serum Bilirubinemia) is the criterium for the presence of hyperbilirubinemia and should therefore be the first action of the protocol. In contrast to other actions performed by the pediatricians, which intentionally fell outside of the scope of the protocol (the protocol was designed to only model a certain subset of the underlying causes of jaundice), the action "measure TSB value" clearly falls within the scope of the protocol. However, it is modeled in the protocol in the wrong place. Thus the lack of an initial measurement of TSB indicates a gap in the protocol. When we discard the action "measure TSB value", which cannot be matched correctly due to a gap in the protocol and the actions which have no match in the protocol, we observe that, on average, our algorithm matches 95.8% of the actions correctly to with one of the treatment sequences defined by the guideline.

Based on the results of the first experiment, we investigated whether there is a significant difference in the performance of our algorithm on solutions of a pediatrician familiar with the protocol and on solutions of a pediatrician unfamiliar with the protocol. The results indicate a better performance on the pediatrician

²Incorrectly matched are those actions that either could not be matched to the guideline or were linked to the wrong entity

consulted by [11]. To determine whether this difference is significant, we conducted a statistical test. The result of the test showed that the difference is not significant. Similarly, we investigated whether the results of the second experiment were significantly worse than those of the first experiment. The statistical test showed the differences with reasoning by encounter and backward execution are not significant.

5 Conclusion

In this paper we have shown that the application of the ideas of practical reasoning, which are formalized in the BDI architecture, enables us to solve a problem that hampered the development of expert critiquing systems. The experiments show that the proposed approach is very promising, especially after correcting the results for the incompleteness of the guideline used in experiments and for an error found in the guideline during the experiments. After correcting for the incompleteness and the error, on average, 95.8% of the actions performed by a physician could be matched with one of the treatment sequences defined by the guideline. This result shows that the physicians follow the guidelines rather well. Moreover, it provides a solid base for a critiquing system. When a physician is skipping expected actions or performing unexpected actions, there is a reasonable chance that the feedback given by the critiquing system is relevant. The chance of relevance is increased by the observation that we can critique based on a physician's *intentions* rather than the fact that he or she performs an action that was not modeled by the guideline (or does not perform an action that was modeled by the guideline).

The measure that we used to determine the distance between the sequence of actions prescribed by the guidelines and the performed actions reported in the EPR does not explicitly take deviations with respect to the order prescribed by guideline into account. This has been a deliberate choice made to increase the robustness of the matching process with respect to variations in the order of performed actions. As a side effect, the current matching process is sensitive to errors in the protocol. Future research will therefore focus on developing better measures that can cope with errors in an underlying guideline without increasing the sensitivity to variations in the order of performed actions. Moreover, future research will also address more complex situations, such as the most general intention being unknown and the treatment of more than one disorder at the same time. Finally, research will have to be directed towards creating the actual critiquing system. This task will not be trivial, because such a system will need to determine the right degree of relevance for a certain critique (neither should a physician be overflowed with very detailed critique, nor should the system not critique a physician skipping a critical action).

References

- [1] A. Advani, K. Lo, and Y. Shahar. Intention-based critiquing of guideline-oriented medical care. In *Proceedings American Medical Informatics Annual Symposium 1998*, pages 483–487, 1998.
- [2] A. Advani, Y. Shahar, and M.A. Musen. Medical quality assessment by scoring adherence to guideline intentions. *Journal of the American Medical Informatics Society*, 9:S9–S97, 2002.
- [3] M.H. Beers and R. Berkow. *The Merck Manual of Diagnosis and Therapy, Seventeenth Edition*. Merck Research Laboratories, Whitehouse Station, 1999.
- [4] M. Berg. *Rationalizing medical work*. PhD thesis, Maastricht University, 1995.
- [5] M E Bratman, D J Israel, and M E Pollack. Plans and *Computational Intelligence*, 4:349–355, 1988.
- [6] Commissie Aanvullende Diagnostiek. *Diagnostisch kompas*. College voor Zorgverzekeringen, Amstelveen, 2003.
- [7] Commissie Farmaceutische Hulp. *Farmacotherapeutisch kompas*. College voor Zorgverzekeringen, Amstelveen, 2002.
- [8] Abigail S. Gertner. Plan recognition and evaluation for on-line critiquing. *User Modeling and User-Adapted Interaction*, 7(2):107–140, 1997.
- [9] R.P. Giugliano, C.A. Camargo, D.M. Loyd-Jones, J.D. Zagrodsky, J.D. Alexis, K.A. Eagle, V. Fuster, and C.J. O'Donnell. Elderly patients receive less aggressive medical and invasive management of unstable angina. *Archives of Internal Medicine*, (158):1113–1120, 1998.

- [10] M. Marcos, G. Berger, F. van Harmelen, A. ten Teije, H. Roomans, and S. Miksch. Using critiquing for improving medical protocols: Harder than it seems. In *Proceedings of the Eighth European Conference on Artificial Intelligence in Medicine*, pages 431–441, 2001.
- [11] Mar Marcos, Michael Balsler, Annette ten Teije, and Frank van Harmelen. From informal knowledge to formal logic: a realistic case study in medical protocols. In *Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management*. Springer-Verlag, 2002.
- [12] P.L. Miller. *Expert critiquing systems: Practice based medical consultation by computer*. Springer-Verlag, New York, 1986.
- [13] M. Peleg and J. Tu. Comparing models of decision and action for guideline-based decision support: a case-study approach: Stanford university, 2002. Report No.: SMI-2002-0922.
- [14] M. Peleg, S. Tu, J. Bury, P. Ciccarese, J. Fox, R.A. Greenes, R. Hall, P.D. Johnson, N. Jones, A. Kumar, S. Miksch, S. Quaglino, A. Seyfang, E.H. Shortliffe, and M. Stefanelli. Comparing computer-interpretable guideline models: a case-study approach. *Journal of the American Medical Informatics Association*, (10):52–68, 2004.
- [15] A. Rao and M. Georgeff. Bdi agents: From theory to practice. In *Proceedings of the 1st International Conference on MultiAgent Systems*, pages 312–319, 1995.
- [16] Y. Shahar, S. Miksch, and P. Johnson. A task-specific ontology for the application and critiquing of time oriented clinical guidelines. In *Proceedings of the Sixth Conference on Artificial Intelligence in Medicine in Europe*, pages 51–61, 1997.
- [17] S.B. Soumerai, T.J. McLaughlin, J.H. Gurwitz, E. Guadagnoli, P.J. Hauptman, C. Borbas, N. Morris, B. McLaughlin, X. Gao, D.J. Willison, R. Asingera, and F. Gobel. Effect of local medical opinion leaders on quality of care for acute myocardial infarction: a randomized controlled trial. *Journal of the American Medical Association*, (279):1392–1394, 1998.
- [18] D.R. Sutton and J. Fox. The syntax and semantics of the *proforma* guideline modelling language. *Journal of the American Medical Informatics Association*, (10):433–443, 2003.
- [19] J. van der Lei. *Critiquing based on computer stored medical records*. PhD thesis, Erasmus University Rotterdam, 1991.
- [20] M. Wooldridge. *An Introduction to MultiAgent Systems*. John Wiley & Sons, Chichester, England, 2002.
- [21] S.H. Woolf, R. Grol, A. Hutchinson, M. Eccles, and J. Grimshaw. Clinical guidelines: Potential benefits, limitations, and harms of clinical guidelines. *Bio Medical Journal*, (318):527–530, 1999.

SEQUENCE MINING ON WEB ACCESS LOGS: A CASE STUDY

Carlos Soares ^a Edgar de Graaf ^b Joost N. Kok ^b
Walter A. Kusters ^b

^a *LIACC/Faculty of Economics, University of Porto*
Rua de Ceuta 118, 6 andar, 4050-190 Porto, Portugal
csoares@liacc.up.pt

^b *LIACS, University of Leiden, Leiden, The Netherlands*
{edegraaf, joost, kusters}@liacs.nl

Abstract

We present a case study in which sequence mining algorithms were applied to web access log data. The data are from a portal that is targeted for business users. In this portal, like in many others, content is described using a set of descriptors, such as keywords, category and type. We investigate whether representing content by the type rather than its identifier enables existing sequence mining methods to obtain interesting patterns. Rather than a more traditional approach based on measures such as support and confidence, we analyze results from an application perspective. This enables us to identify opportunities for improving and extending these methods.

1 Introduction

Understanding the behavior of users in websites is essential for successful e-business (e.g., pricing [3] and improving loyalty [10]). The most common approach to this task is based on the analysis of web access logs, which store clickstream information, i.e., information about the requests performed by the users of a website. Given the information in web access logs, which can be regarded as sequences of clicks, a natural approach to analyse the navigation behavior of users is the use of sequence mining methods [8].

The case study in this work is PortalExecutivo.com (PE), a Portuguese web portal targeted to business executives. The goal of PE is to become an essential information tool for its customers by facilitating the access to a large body of relevant content. One interesting feature of PE is that although only registered users can access content items, guest users (i.e., non-registered users) are able to navigate within the structure of the portal and view summaries of those items. Like for any portal, understanding the behaviour of its users is essential for PE to provide a better service to them. Improvements can be made regarding services (e.g., if email alerts are not used, maybe they should not be provided), structure (e.g., relocation of a sub-tree of the content hierarchy which is not commonly accessed), meta-data (i.e., meta-data describing content, such as keywords and categories) and type and quantity of content (e.g., increasing the number of items from a particular source which is very popular). Two questions that are relevant to analyze the behavior of users are: Which are the paths most frequently followed by the users? Are there differences in the navigation patterns of guest and registered users?

The use of data mining techniques to address this kind of questions is usually referred to as web usage mining (e.g., [9]). The use of sequence mining techniques for web usage mining is increasingly popular (e.g., [5]). One method of analysis would be to do process mining as discussed in [13, 14]. The browsing behaviour will result in sequences of clicks and these sequences can be interpreted as Petri nets describing this behaviour. However, probably there are multiple types of users with different Petri nets and it is difficult to separate these users in different workflows.

In this paper we present a case study of the application of sequence mining algorithms to address those questions. However, most papers focus on research issues (e.g., proposing and comparing methods) and evaluate results in terms of technical measures (e.g., confidence and support) which relate only indirectly to application goals. By taking on this application perspective we are able to identify advantages and disadvantages of this approach, focusing our analysis of results on a business point-of-view rather than a more technical perspective. We also investigate the effect of using an alternative representation of items in the sequences. The representation is based on the *meta-data* (i.e., descriptors such as type, category, keywords) which are often used to describe content in portals. Although the paper focuses on a case study, we believe that conclusions are applicable to a wide range of websites.

A summary of the necessary background is given in Section 2. More detail concerning the application is given in Section 3. In Section 4 we present results, which are discussed in Section 5. Some conclusions are given in Section 6.

2 Background

In this section, we provide background concerning the Web and Sequence mining, which puts our work in a suitable context.

2.1 Web Usage Mining

The amount of information available on the web makes the use of semi-automatic tools essential to obtain knowledge that is useful to users. One approach to this problem is *web mining*, which consists of the application of data mining methods for that purpose. A good overview of web mining can be found in [9].

One of the sub-categories into which web mining is commonly divided is *web usage mining*. It is defined as the “application of data mining techniques to discover usage patterns from Web data” [12]. Most of the time this is achieved by analyzing access log data that represents clicks that users make and is generated by the web server. A number of problems arise in the use of this type of data, which include the accurate identification of the user and navigation that does not generate requests to the server (e.g., pages stored in local cache). A number of different web usage mining applications have been reported, such as personalization, user profiling and site modification.

2.2 Sequence Mining

The goal of sequence mining is to identify interesting sequential patterns in a database of sequences [1]. Data is stored as sequences in many applications, such as in biology [4], shopping baskets [1] and web access logs [11]. Here, we will limit our description to basic concepts which are required to understand the rest of the paper. More information can be found in many papers dedicated to this area (e.g., [8, 2, 4]).

We assume we have a database D of sequences of items, where each item is an atomic representation of an object of interest in the application (e.g., a request for a content item in a web portal). We call a given sequence $d = (d_1, d_2, \dots, d_m)$ a *super-sequence* of a sequence $s = (s_1, s_2, \dots, s_k)$ if $k \leq m$ and for each s_i ($1 \leq i \leq k$) there is a d_{j_i} ($1 \leq j_i \leq m$) with $s_i = d_{j_i}$ and $j_{i-1} < j_i$ ($i > 1$). We denote this with $s \prec d$. The sequence s is called a *sub-sequence* of d . Commonly a sequence $d \in D$ is said to support a *pattern* s if the pattern is a sub-sequence in the sequence d :

$$\text{supp}(s, d) = \begin{cases} 1 & \text{if } s \prec d; \\ 0 & \text{otherwise,} \end{cases}$$

We then can define the *sequence_support* of pattern s in database D as

$$\text{sequence_support}(s, D) = \sum_{d \in D} \text{supp}(s, d)$$

We call s a *frequent subsequence* if its sequence support is larger than or equal to a user-defined threshold *minsupp*. This defines *sequential patterns* on sequences of items. Another definition of

sequential patterns was given by Agrawal et al. in [1], in which they define sequential patterns on sequences of item sets.

The most common task addressed in sequence mining research is the identification of frequent patterns. PREFIXSPAN as described in [8] searches for those patterns with *support* larger than or equal to a given support threshold *minsupp*. The algorithm starts with all frequent sub-sequences of size one. For each sub-sequence a projected database is created. The *projected database* is a database of pointers to the first item occurring after the current pattern, also called the *prefix*. A sequence is only in the projected database if it contains the prefix. The set of frequent sub-sequences is extended with all frequent sub-sequences of size two by only looking in the projected database. Again for each frequent sub-sequence of size two a corresponding projected database is created. This process continues recursively until no extension is frequent anymore.

More recently, other tasks have been addressed, such as classification of sequences. A few methods have been proposed to search for maximally discriminating patterns, such as PREFIXTWEAC [4] and correlated association rule mining [15]. The main parameter of PREFIXTWEAC, used in this work, is the number of k maximal discriminating patterns (i.e., the k most different patterns). This k can be used to prune, even though the difference measure doesn't have the anti-monotone property. This technique is described in more detail in [7].

2.3 Sequence Mining to Mine Web Access Log Data

Sequence mining algorithms are commonly used to analyze access log data [5, 11]. However, experimental evaluation in most papers is done from a technical perspective (e.g., computational efficiency). For understandable reasons, there are very few examples where the results of a sequence mining algorithm are evaluated from the point-of-view of the application. A notable exception is the work of Spiliopoulou and Pohle [11]. They apply a sequence mining algorithm to the logs of a website containing a database of German schools. The structure of the site is changed based on the patterns found and the impact of those changes is quantified using suitable metrics (contact and conversion efficiency).

The questions addressed by these authors are similar to the ones that are investigated in this work. Besides identifying frequent patterns in navigation sequences, the navigation patterns of customers and non-customers patterns are also compared. A heuristic in two phases is proposed: first frequent patterns are identified in customer sessions. Then, these patterns are applied to non-customer sessions to find interesting differences.

3 Problem Definition

PortalExecutivo.com (PE) is a Portuguese web portal targeted to business executives. The business model of PE is subscription-based, which means that only paying users have full access to content through web login. However, users can freely browse the site's structure and glimpse the kind of content which is available. Although some of the content is provided by PE, the majority originates from a large number of partners, which include several publications (The Economist, Wired, etc.) and companies (Accenture, Portugal Telecom, McKinsey, etc.). Value is added not only by concentrating content in a single access point but also by structuring and interrelating content items in a way that makes it easier for the user to find relevant information. This is achieved by providing several mechanisms (e.g., a search engine) that are based on quite a rich set of meta-data fields that are associated with each item, including keywords, categories, source and authors.

The goal of PE is to facilitate the access of its members to relevant content. To pursue this objective, it is essential to obtain as much information about the profile and the behavior of its users as possible. A number of different questions arise, including the two which we address in this work:

- Which are the paths most frequently followed by the users?
- Are there differences in the navigation patterns of guest and registered users?

Answers to the first question can be used, for instance, to identify problems in the structure that cause inefficient navigation. Additionally, those patterns can be used to identify content which a

users	3,237
sessions	71,547
accesses	1,784,642
content items	18,959
content types	651

Table 1: Basic statistics for the access log data.

Field	1st click	2nd click	3rd click	...
Title	home page	Analysis of National Budget	The Xmas trap	...
Type	navigation	article	news	...
Source	—	PE	clipping service	...

Table 2: Alternative representations for a given session using different fields from the meta-data describing content items.

user is expected to be interested in and, thus, provide recommendations. As for the second question, differences between guest and registered users could be used, for instance, to adapt the site to the behavior of the former, with the goal of convincing them to register.

Commonly, web usage mining is carried out with data from web access logs generated by the web server. However, the pre-processing of access logs required to obtain data which is ready for mining is not a trivial task [12]. Problems for which there is really no definite answer include the identification of sessions (i.e., sequences of clicks that constitute one session of the user on the site) and the identification of the user within and across sessions. However, in this case study we are able to avoid these issues by the use of access data which is stored by the content management system of PE in a table of a relational database. There are 6 fields in this table: content id(entifier), session id, user id, time stamp, title and type of content item. Our analysis has focused on two of these, namely the id, which uniquely identifies content items, and the type, which aggregates items into groups, such as articles, news items, homepage, etc. This table can be regarded as a clean version of the access logs. We have used data from the period between May 2002 and November 2005, for which a number of statistics is given in Table 1.

4 Mining Different Representations of Sequences

In a typical application of sequence mining to web access logs, each sequence is a session represented as a set of content item ids. Given the large number of different content items (18,959), not many sequences are expected to achieve the minimum support. In spite of this, it may be expected that in such a portal, some content items are very popular (e.g., an interview with Jack Welch or an analysis of the national budget) and, thus, patterns relative to these items could exist.

As mentioned earlier, content items in PE are described not only by the id (or title) but also by a set of descriptors (i.e., meta-data) which include author, category, type and keywords. These descriptors can provide an alternative representation of content items, as shown in Table 2. These alternative representations can be exploited for sequence mining. Different patterns will be obtained that provide different perspectives on the navigation patterns of the users. For instance, with the original representation it is possible to obtain patterns such as: “users that access the analysis of the national budget for 2004, followed by an interview to the Minister of Finance”. By representing content items using their type or source, patterns such as “users that read news, followed by articles” and “users that read content from the clipping service, followed by one from Wired”, respectively, can be obtained. Alternatively, given that meta-data descriptors usually provide a higher-level representation of the content, the domain is usually smaller (e.g., 651 different content types), which means that more patterns are expected to have higher support and, thus, more knowledge may be obtained. On the other hand, this approach is more useful for characterization of the navigation patterns than for recommendation of content. For instance, given the first pattern above, if a user accesses the analysis of the national budget for 2004, the portal may recommend the interview with the Minister of Finance to this user. Using the second pattern for this purpose is more difficult because, if a user reads a news item, then it is necessary to choose which, from the

many articles available, to recommend.

4.1 Results of Frequent Itemset Mining

To address the first question, we have used our implementation of the PREFIXSPAN algorithm for frequent sequence mining [8]. All patterns with a minimum support of 100 were analyzed.

The results confirm our expectations. We obtained approximately 5,000 patterns and very few were interesting. Most of them were related to navigation pages (i.e., non-leaf nodes in the site structure). Our hypothesis that some content items could be sufficiently popular to appear in sequences was confirmed in very few cases only (e.g., an analysis of the national budget for 2004, following the homepage of the portal).

Next, we applied PREFIXSPAN to the sequences in which the items were represented as the type of content (2nd line in Table 2). As expected, many more interesting patterns were obtained. For instance, we observed that in 1,261 sessions, the user browses through several articles and then prints one of them. Additionally, in 859 sessions, the user accesses the stock exchange quotes, followed by news. Although further investigation is required to investigate these patterns, they can trigger changes in the way PE provides its service to these users. The first pattern may indicate that these users search for content to read while returning home. If this is the case, then these are clearly users that could benefit from a personal recommendation mechanism in the portal triggered at the most suitable time (i.e., when the user usually returns home). The second pattern suggests that it could be a good idea to have a page where both quotes and recent news items (particularly, financial news) could be provided to these users.

4.2 Results of Sequence Difference Mining

Concerning the second question, with the goal of understanding the difference between browsing behaviour of registered and unregistered users (i.e., guests), we have addressed it with the PREFIX-TWEAC algorithm [4]. We have set the time window parameter to 10 and we have analyzed the 100 *maximal discriminating patterns* obtained with a minimum support of 100. These patterns *do* occur for one group of users, but *do not* occur for another group.

As in the earlier set of experiments, we started by representing items by their ids. Again, the results were not outspoken. First of all they are not very discriminating: the maximally discriminating pattern matches 171 sessions of registered users and 298 guest sessions. Additionally, the patterns obtained are not very interesting from an application point-of-view.

Using the alternative representation, where the algorithm is applied to sequences of types of content items, more interesting results were obtained. For instance, as could be expected, registered users have longer threads of articles, news items and other functionalities than guests. A more interesting observation, from a business perspective, is that guest users access the areas of the partners less often than registered users. Given that this is one of the main features of the portal, we would like guest users to be more exposed to it. Therefore, the partners should be made more visible to guest users, for instance, by highlighting the most recent contributions or even by making selected content available for free.

5 Discussion

We start by discussing the representation of the sequences. Next we identify a few problems which could lead to variants of sequence mining algorithms.

5.1 Representation of Items

In general, our results show that for descriptive applications, in which the goal is to understand the behavior of the users, more interesting knowledge can be obtained by representing the items in sequences using more abstract representations, when these are available. In our case, we have used meta-data containing alternative descriptions of the items. However, this approach may not be suitable if the goal is to provide recommendations. That is, given a frequent sequence (A,B,C)

and a user who browsed (A,B), suggest C. In this case, the most common approach, in which items are represented by their ids, may be better.

Additionally, in the current definition of sequential patterns it is possible to have an unlimited amount of items, in the sequence in the database, between two items from the pattern. So if the pattern is (A,B) then the sequence in the database containing the pattern could be (A,C,C,C,C,C,B). This means that the patterns that are obtained hide some information. This may be an advantage, if the loss consists of irrelevant details. For instance, one of the patterns observed was that many users access the stock exchange quotes, followed by news. This is an interesting pattern, independent of how much browsing the user does in between these two types of content. On the other hand, the missing information may be important for the interpretation of the pattern. For instance, in the other pattern mentioned above, concerning users that browse through several articles and then print one item, it may actually mean that the item that is printed is not an article. This may be addressed by looking for patterns consisting of adjacent items or by imposing a constraint on the size of the gap or the size of the pattern [2, 6]. Alternatively, this issue could be addressed by extending the representation of the rules with some statistics describing the distribution of the items in the gaps. For instance, given the pattern (A,B,C) with matching sequences:

(A,C,B,C)
 (A,C,D,B,D,C)
 (A,B,C)

the pattern could be complemented with histograms describing the size of the gaps in the sequences. In this case, the histogram for the subsequence of the pattern (A,B) would be 1/0 (meaning 1 sequence with gap size 0), 1/1 and 1/2, and for the subsequence (B,C), 2/0 and 1/1. Additionally, information about the items in the gaps would also be useful. In this case, concerning subsequence (A,B), the gaps in the sequences contain C twice immediately after A and both C and D appear once each immediately before B. Although this kind of information could be collected in the sequence mining process, given the large amount of different statistics that may be interesting, it would better be implemented as a tool for post-processing sequential patterns. A related approach has been proposed earlier, in which patterns are represented as aggregate trees [11].

The use of an alternative representation for items based on meta-data introduces another issue. Many redundant patterns of the form (A,B), (A,A,B), (A,B,B), (A,A,B,B), etc. were obtained. Again, results could be simpler to understand by post-processing the patterns. For instance, these patterns could be represented by a single regular pattern $(A\{1,2\},B\{1,2\})$.

5.2 Suggestions for Sequence Mining Research

We have identified four alternative sequence mining tasks which could be useful for the analysis of access log data but are possibly also applicable in other domains.

Mining frequent repetitive subsequences The goal is to find subsequences that repeatedly appear within sequences. An example the subsequence (A,B,C) is a frequent repetitive pattern in the following data:

sequence 1: ... A,B,C ... $\{e_1 \text{ elements}\}$... A,B,C ... $\{e_2 \text{ elements}\}$... A,B,C ...
 sequence 2: ... A,B,C ... $\{e_3 \text{ elements}\}$... A,B,C ...
 ...

One application in access log mining would be to find repetitive behaviour of users that may, for instance, indicate that different organization of the content or a recommendation system could decrease the number of clicks required to reach necessary information. This method would require two new parameters: minimum number of repetitions and range of valid e_i values.

Mining frequent periodic subsequences The goal is the same as the previous one but imposing constraints on the variation of the values of e_i for each pattern. For all e_i of all sequences matching a pattern, $max(e_i) - min(e_i)$ must be smaller than a given threshold, which is a parameter to the algorithm. Alternatively, the function assessing the quality of patterns, which usually takes only the support into account, could also include a term that reflects the amount of variation of the e_i values.

Mining frequent subsequences of elements represented as sets In this case, the goal is to find subsequences of elements which are described using sets of properties [1]. As an example, (A_1, C_1) is a frequent pattern in the following data, in which each item is represented by a set of fields $\langle XY \dots \rangle$:

sequence 1: $\dots \langle A_1 B_1 \rangle, \langle A_2 C_1 \rangle \dots$

sequence 1: $\dots \langle A_1 B_2 \rangle, \langle B_3 C_1 \rangle \dots$

...

Its application in access log mining could be to identify navigation patterns combining different representations for content items (e.g., content id, category, keyword, author).

Mining frequent subsequences of structured elements The goal is the same as in the previous task but the patterns are built taking into account constraints on the values of sets. As an example, assume the constraint $C \leftarrow A$, which means that including in a pattern a value concerning the field C is possible only if the pattern already restrains the values of field A for the same item. Then, in the following data, $(\langle B_1 \rangle \langle A_2 \rangle)$, $(\langle A_1 \rangle \langle A_2 \rangle)$ and $(\langle A_1 C_2 \rangle \langle A_2 \rangle)$ are frequent patterns but not $(\langle C_2 \rangle \langle A_2 \rangle)$

sequence 1: $\dots \langle A_1 B_1 C_1 \rangle \langle A_2 \rangle \dots$

sequence 2: $\dots \langle A_1 B_1 C_2 \rangle \langle A_2 \rangle \dots$

sequence 3: $\dots \langle A_1 B_2 C_2 \rangle \langle A_2 \rangle \dots$

...

In access log analysis, this could be used to identify time-references navigation patterns, e.g., if A is the category and C is the weekday, we would be able to obtain patterns such as “users who consult the stock exchange section on friday” but not patterns such as “users who access something on friday”.

6 Conclusions

This paper describes the application of frequent sequence mining methods to web access log data. First, we exploit an alternative representation for the items that constitute the sequences. We compare the common approach, in which items are represented by their ids, with a higher-level representation, which indicates the type of content (e.g., navigation page, article, news item). Although we have focused on an alternative representation which is specific to the application at hand, similar approaches can be followed in other access log analysis problems, because many web sites are currently supported by content management systems which characterize content using this kind of meta-data.

Rather than proposing a new algorithm, we apply existing algorithms. However, we provide an analysis of the results from an application perspective. We observe that, if the goal is to understand user behavior, it is possible to obtain better patterns with the alternative representation. We also discuss the applicability of frequent sequence mining to web access log data in light of that analysis. We identify some advantages and shortcomings. Although this is not a novelty in the community, this application confirms that better post-processing methods are required.

Finally, based on this application, we identify variants of frequent mining tasks which can be useful. We expect that our observations will trigger research on methods to address those tasks. Although they are identified in the context of access log mining, we believe that they may be relevant in different sequence mining applications.

Acknowledgements

The first author thanks the financial support of Fundação Calouste Gulbenkian, POSC/EIA/-58367/2004/Site-o-Matic Project (Fundação Ciência e Tecnologia) co-financed by FEDER and the European Research Training Networks SegraVis (HPRN-CT-2002-00275). The second author is financed by the Netherlands Organization for Scientific Research (NWO) MISTA Project (grant no. 612.066.304).

References

- [1] R. Agrawal and R. Srikant. Mining sequential patterns. In *ICDE*, pages 3–14, 1995.
- [2] C. Antunes and A. L. Oliveira. Generalization of pattern-growth methods for sequential pattern mining with gap constraints. In *MLDM*, pages 239–251, 2003.
- [3] W. Baker, M. Marn, and C. Zawada. Price smarter on the net. *Harvard Business Review*, 79(2):122–127, 2001.
- [4] E. de Graaf and W.A. Kusters. Efficient feature detection for sequence classification in a receptor database. In *BNAIC*, pages 81–88, 2005.
- [5] C.I. Ezeife and Y. Lu. Mining web log sequential patterns with position coded pre-order linked wap-tree. *Data Min. Knowl. Discov.*, 10(1):5–38, 2005.
- [6] M. Leleu, C. Rigotti, J.-F. Boulicaut, and G. Euvrard. Constraint-based mining of sequential patterns over datasets with consecutive repetitions. In *PKDD*, pages 303–314, 2003.
- [7] S. Nijssen and J.N. Kok. Multi-class correlated pattern mining. In *KDID*, pages 165–187, 2005.
- [8] J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M. Hsu. Prefixspan: Mining sequential patterns by prefix-projected growth. In *ICDE*, pages 215–224, 2001.
- [9] D. Pierrakos, G. Paliouras, C. Papatheodorou, and C.D. Spyropoulos. Web usage mining as a tool for personalization: A survey. *User Model. User-Adapt. Interact.*, 13(4):311–372, 2003.
- [10] F.F. Reichheld and P. Schefter. E-loyalty. *Harvard Business Review*, 78(4):105–113, 2000.
- [11] M. Spiliopoulou and C. Pohle. Data mining for measuring and improving the success of web sites. *Data Min. Knowl. Discov.*, 5(1/2):85–114, 2001.
- [12] J. Srivastava, R. Cooley, M. Deshpande, and P.-N. Tan. Web usage mining: Discovery and applications of usage patterns from web data. *SIGKDD Explorations*, 1(2):12–23, 2000.
- [13] W.M.P. van der Aalst, T. Weijters, and L. Maruster. Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering*, 16(9):1128–1142, 2004.
- [14] T. Weijters and W.M.P. van der Aalst. Process mining: Discovering workflow models from eventbased data. In *BNAIC*, pages 283–290, 2001.
- [15] A. Zimmermann and L. De Raedt. Corclass: Correlated association rule mining for classification. In *Discovery Science*, pages 60–72, 2004.

Autonomous Color Learning in an Artificial Environment

Dave van Soest Mark de Greef Jürgen Sturm Arnoud Visser

Intelligent Autonomous Systems, Universiteit van Amsterdam

<http://www.science.uva.nl/research/ias/>

Abstract

1 Introduction

1.1 Context

Low level computer vision tasks are deceptively difficult. Promising results obtained in a laboratory environment are hard to reproduce when a new environment is encountered. Yet, a robot that relies for its world modeling on computer vision should be able to operate in different environments, where the lighting conditions can only partly controlled.

The computer vision system of a robot should be able to reliably provide the significant features that are visible in the environment. Which features are significant is application dependent, but in principle it is represented with characteristics clusters in a multi-dimensional feature space.

In our application, the RoboCup soccer competition, we are interested if certain objects (ball, beacons, goal) are present in the field of view. RoboCup soccer is a color-coded environment, where highly saturated colors are used to ease the recognition of those objects essential for RoboCup soccer. So the color of the object, as classified by the human, is the natural feature space to be used as the first step of processing the stream of images.

Natural to the robots used in the 4-Legged League is the 3-dimensional YUV color space (see figure 1). Transformations to other color spaces are possible, but this transformation will cost a fair amount of computation time. Our method is also applicable for other color spaces, as all major color spaces are represented with three components (e.g. RGB, YUV, HSI). A color coded object in the image, like the orange ball, should be seen a group of pixels of the same color class, although the actual values of the YUV color components will not be the same. Think of shadows, highlights, reflections of nearby colored surfaces. Yet, a human is able to classify a ball as orange under a wide variety of circumstances, and a robot should be able to learn the same mapping from the 3-dimensional color space to an 1-dimensional color classes.

To find the mapping from the 3-dimensional color space to the 1-dimensional color classes an analysis of the feature space is needed. Objects which humans would classify as the same color should be visible as denser regions, indicating that pixels that are 'similar'. It is the task of the algorithm to separate those regions into clusters. While there is a wide variety clustering techniques available (see for an overview [8]), complex algorithms are out of scope due to the limited computational resources of the RoboCup robots.

Partitional clustering algorithms are attractive for applications where computational costs are an issue. Fortunately the number of color classes at the RoboCup field is priori knowledge, which allows us to use a mixture density model for the clusters, without the need to resolve the issue how many clusters are present in the data. The data only has to be 'fitted' against the free parameters

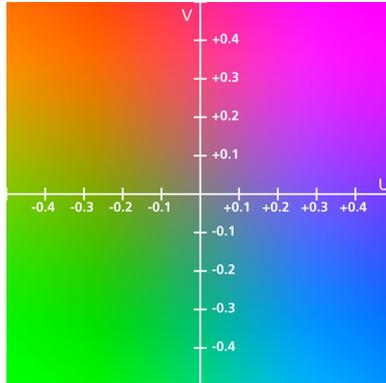


Figure 1: A cross-section of the YUV color space at $Y = 0.5$.

of the density distributions in the mixture model. For the density distribution we have chosen a Mixture of Gaussian (MoG), which means that the mean $\vec{\mu}_s$ and the standard deviation Σ_s for each of the clusters s has to be estimated, together with w_s which represent the weighting factor between the clusters. The free parameters $\vec{\mu}_s$, Σ_s and w_s are optimized in the maximization step of an expectation-maximization (EM) algorithm [2], while the training data is redistributed over the clusters in the expectation step.

Once the mapping from the 3-dimensional color space to the 1-dimensional color classes is learned, this color map can be stored in a lookup-table. The color-map is then used to segment the images. The segmented images can be scanned fast to identify useful objects (e.g. the beacons, the ball). The quality of the mapping has a strong influence on the reliability of the world modeling and therefore the performance of the robots in the soccer games. Currently, all participants prefer to manually generated this color-map, in spite of all efforts to automate this process.

The method that has the greatest resemblance with our method is applied by the Austin Villa team [7]. They introduce complete automatic way to learn the relevant colors on the field, by explicitly visiting the color coded objects in a certain order (center circle, yellow goal, yellow-pink beacon, blue goal, pink-blue beacon, ball, red opponent). In a sense, this is a greedy EM algorithm [9], initiated with two color clusters, and continued by adding a new cluster (and new training data) one after each other. The results were good for the goals and beacons, but dropped fast after the inclusion of orange in the color map. Based on this result they decided to adapt the LAB color space [5].

Another successful method applied in the RoboCup is the layered color precision method [3]. Here the color map is learned with increasing precision level. On each level, the learned colors are defined relative to green, which makes this method strongly depended on the initial green assignment. In [6] a hierarchical Bayesian model is described, which slowly adjust the mean of the clusters with the aid of a Kalman filter. Fast changes in the lighting conditions are accommodated by using multiple Kalman filters, and to switch between filters when appropriate. This approach requires that multiple filters are initialized with data collected under a variety of circumstances. Because the location of the clusters is constantly drifting, it is difficult to assign an absolute meaning to such a color cluster.

So, playing under natural illumination stays a challenge, as indicated by the survey performed by Mayer et. al [4].

2 Approach

2.1 Clustering

In this section we will outline our learning method to learn the correspondence between a color observed by the robot’s camera with a color class which is a property of one the objects on the soccer field. Let each pixel p of an image I have a color \vec{c} , where \vec{c} is a vector with the YUV components. Color segmentation is assigning a pixel p to a color class C_i , where C_i has to be selected from the list of RoboCup color-codings (C_1, \dots, C_k). C_1 represents for instance ‘white’, C_k ‘black’. The number k is small; in total 9 different color-codings are used in the 4 Legged League. For each value of \vec{c} we can calculate the posterior probability using Bayer rule:

$$p(C_i|\vec{c}) = \frac{p(\vec{c}|C_i)p(C_i)}{p(\vec{c})} \quad (1)$$

This relationship cannot be learned directly. When we collect training data, the RoboCup color-codings should be visible as clusters in the color space. The location and spread of these clusters can be learned by optimizing a Mixture of Gaussians. The result is the conditional probability $p(\vec{c}|CC_s)$ where CC_s is one cluster from the set of the clusters that is visible in the image (CC_1, \dots, CC_{kk}). When all RoboCup objects, and no other objects, are visible in the collected training data, the number of color-codings k should be equal to the number of clusters kk . This one-to-one assignment of each color cluster CC_s to a color class C_i is not trivial, but we have found a robust heuristic for this assignment $i(s)$, as described in section 2.2. So, under the assumption that all RoboCup objects are visible ($p(C_i) = 1$), we can learn the relationship:

$$p(C_{i(s)}|\vec{c}) = \frac{p(\vec{c}|CC_s)}{p(\vec{c})} \quad (2)$$

If we model the color space with a Mixture of Gaussians, each cluster is described with a normal distribution with a mean $\vec{\mu}_s$ and a variance of Σ_s .

$$\mathcal{N}(\vec{c}; \mu_s, \Sigma_s) \equiv (2/\pi)^{-D/2} |\Sigma_s|^{-1/2} \exp[-\frac{1}{2}(\vec{c} - \vec{\mu}_s)^\top \Sigma_s^{-1}(\vec{c} - \vec{\mu}_s)] \quad (3)$$

The probability to see a specific color value $p(\vec{c})$ is than a weighted sum of all clusters

$$p(\vec{c}) \equiv \sum_{s=1}^{kk} w_s \mathcal{N}(\vec{c}; \mu_s, \Sigma_s) \quad (4)$$

where w_s are a set of non-negative weights which are partition of unity $\sum_{s=1}^{kk} w_s = 1$. The parameters w_s , $\vec{\mu}_s$ and Σ_s of the Mixture of Gaussians define the importance, location and form of the clusters. These are the free parameters that are updated during the maximalization step of the EM-algorithm.

In the normal setting of the RoboCup it is difficult to guarantee that only RoboCup color-coded objects are visible. In the background many other colors can be present. To demonstrate the working of our algorithm, we simplified our test circumstances by putting two goals on top of each other (see 2 which removes the background from sight. The camera view now contains exactly those colors that are to be classified. We explicitly choose to not model this setup in the algorithm in order to maintain generality. This means that every setup is allowed, as long as all predefined colors are visible and no background is visible. This constraint can be gradually released, by adding slowly some views of the background in the training-set. The algorithm has to be carefully checked for robustness against the introduction of this type of noise in the color space.

We decided to only use five of the predefined colors, namely white, yellow, sky-blue, green, and pink. We only use these colors because the separation of orange/pink/red and dark-blue/skyblue is



Figure 2: The test-image used to generate the training-set, where only RoboCup color-coded colors are visible.

a task on itself, especially in the YUV color space. If we only use these five colors we do not have the problem of mixing up colors as these colors are well separated in the YUV color space. We can scale our work up later on with the rest of the predefined colors.

2.2 Color class assignment

Once the clusters of the typical colors of the objects are found, it seems to trivial to assign the correct label to each of them. Yet, when different lighting conditions are encountered it is clear that the clusters do not have a fixed size or location. If the light changes then the colors will be shifted. To be able to deal with different lighting condions we can not use absolute boundaries to assign colors to specific areas in the color space. The only constant is the relative location of the clusters in relationship to each other. If sufficient relations between the clusters are defined, then a unique color class can be assigned to each cluster or combination of clusters.

In the YUV color space it is hard to define the relations between the color clusters; e.g. these relations all need to be multi-dimensional. It is easier to define the relations in the HSI color space, because the predefined colors of the soccer field (except black and white) are well saturated, and mainly differ on their hue value H ¹ This allow the specify the relationship between the color clusters in a single dimension (see 3. To define the relations in the HSI color space we have to convert the means of the clusters from YUV values to HSI values. After this we can do define relations in the HSI color space.

For each set of colors and for each color space a different set of relations is needed in order to be able to uniquely identify the color class for each cluster. Therefore the choice of the color space is important for reducing the amount and complexity of the needed relations. As stated before, the HSI space achieves this for the subset of five predefined colors which we use. The set of relations implies an algorithm which uniquely identifies all five clusters. The algorithm for our setting is as follows. Here \mathcal{CC} initially is the set of clusters. One by one clusters are removed this set based on a heuristic rule and added to the set of known color classes \mathcal{C} . The following rules are applied:

$$C_{white} = \arg \min_{CC_s \in \mathcal{CC}} S_s, \mathcal{CC} := \mathcal{CC} - \{C_{white}\} \quad (5)$$

$$C_{sky-blue} = \arg \max_{CC_s \in \mathcal{CC}} H_s, \mathcal{CC} := \mathcal{CC} - \{C_{sky-blue}\} \quad (6)$$

¹In the hue-saturation-intensity (HSI) color space the hue H represents the dominant wavelength of the color; the saturation S refers to the purity of the color, i.e. how much light of the other wavelengths is mixed in the color; the intensity I corresponds to the amount of light. Hue H can be calculated from the RGB colorspace with the formula: $H = s \cdot (\frac{1}{2}\pi + \arccos(\frac{R-\mu}{\sigma\sqrt{2}}))$. $\mu = \frac{R+G+B}{3}$ is the brightness of the color, $\sigma = \sqrt{\frac{(R-\mu)^2+(G-\mu)^2+(B-\mu)^2}{3}}$ is the standard deviation of the color channels and s is a scaling factor.

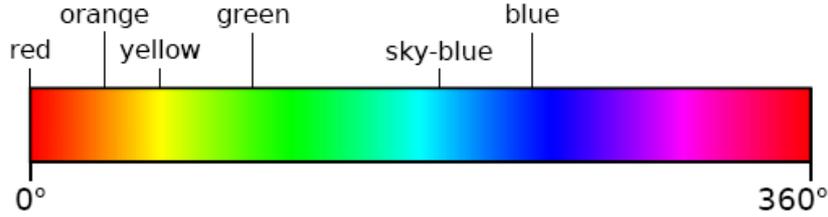


Figure 3: The hue dimension, tagged with the approximate positions of the subset of predefined colors which are theoretically pure.

$$C_{green} = \arg \max_{CC_s \in CC} H_s, CC := CC - \{C_{green}\} \quad (7)$$

$$C_{yellow} = \arg \max_{CC_s \in CC} H_s, CC := CC - \{C_{yellow}\} \quad (8)$$

$$C_{yellow} := CC \quad (9)$$

Adding rules for the other predefined colors is quite trivial. Black can be distinguished from white by its intensity; blue and orange can be distinguished by their hue values, and finally red can be distinguished from pink by its saturation value.

3 Results

Having the method we tested it using different lighting conditions.

- curtains open/closed
- lights on/off
- tl-lights on/off

In figure 4-6 the results of the clustering and color assignment are shown. We show the original image, the color classes of the image, the YU space and the VU space of the YUV color space and finally the hue of the colors that are sorted on their hue (pink, yellow, green and sky-blue). The results show that despite the variety of lighting conditions the cluster assignment is often completely correct and the greater part of the pixels is assigned to the correct cluster. We can see that even with natural light we can distinguish most of the colors.

We have tested all eight circumstances, and the results were good. As an example of a good clustering and assignment we show figure 4. See [1] for more good examples. In this article we highlight the few cases where the limits of the method were encountered.

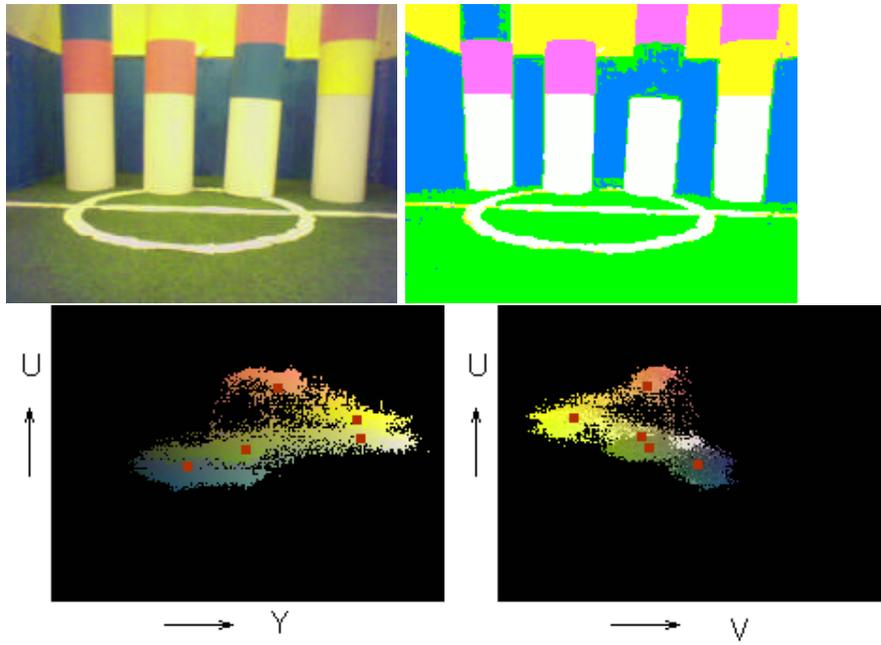


Figure 4: Curtains closed, lights on, tl-lights on

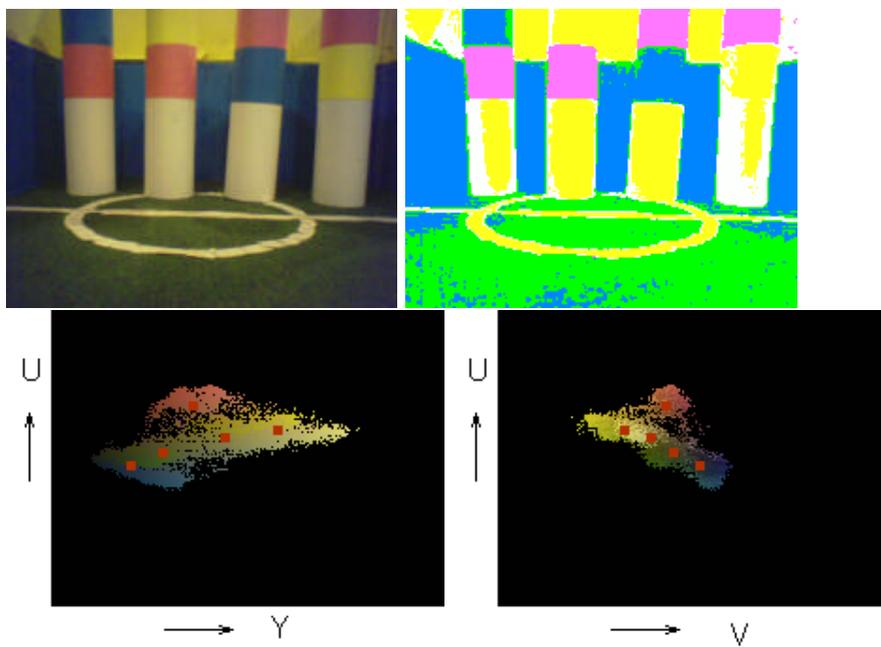


Figure 5: Curtains closed, lights on, tl-lights off

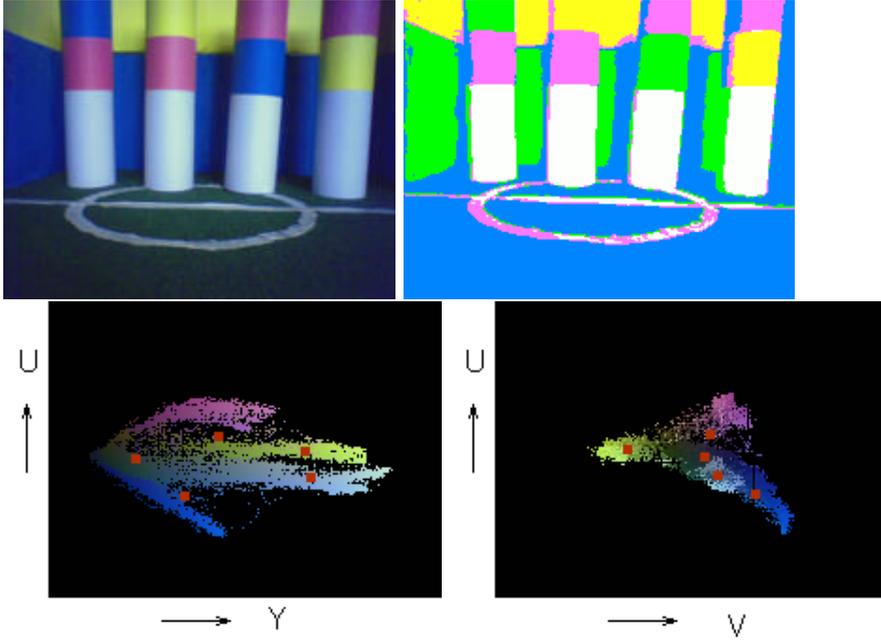


Figure 6: Curtains open, lights off, tl-lights off

In two cases the automatic color calibration algorithm did not work flawlessly. The first case is in figure 5 where the sky-blue and green clusters are switched. This can be explained by the fact that there is very little light, making the green very dark. Also the light is in this case blueish (which can be seen from the reflections from the white pillars). We have seen that the hue of the cluster means of green and sky-blue are very close to each other. Because of this the field is classified as being sky-blue and the sky-blue of the goal and beacons is classified as green.

The second problem we see in figure 5, this is due to the fact that we only have the big lights on and not the tl-lights making the light have a yellowish color. Because of the light being slightly yellowish, the yellow and white clusters get too close to each other, which results in some pixels assigned to the wrong cluster.

4 Conclusions

By creating an ideal test-image, which made sure that the camera view contains only objects with predefined colors, the color classes were clearly visible as clusters. Our algorithm could separate these clusters without problems and we can find the right clusters in a wide variety of circumstances.

For the assignment of color classes to clusters we have used the fact that in HSI the relations between the clusters can be very well defined. In this way we can just walk through a list of clusters and assign the right color classes to them. To do this the centres of the clusters have to be converted from YUV to HSI first. After that the color classes can be assigned relatively easily. The only time that our approach was tested to its limits is when the light is too colored (in our cases blue or yellow).

Acknowledgements

The Dutch AIBO Team is financially supported by DECIS Lab, University of Amsterdam, Technical University of Delft, Saxion University of Professional Education, University of Groningen and University of Utrecht. We are grateful for the contributions by all of our team members, working on RoboCup related research and other research.

References

- [1] M. de Greef and D.A. van Soest. Automatic color calibration on a robosoccer-field. Technical report, Universiteit van Amsterdam, June 2006.
- [2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B 39:1–38, 1977.
- [3] M. Jünger. Using layered color precision for a self-calibrating vision system. In *Proceedings of the 8th International Workshop on RoboCup 2004*, Lecture Notes on Artificial Intelligence. Springer Verlag, 2005.
- [4] Gerd Mayer, Hans Utz, and Gerhard K. Kraetzschmar. Playing robot soccer under natural light: A case study. In *Proceedings of RoboCup-2003 Symposium*, Lecture Notes in Artificial Intelligence, Berlin, Heidelberg, Germany, 2004. Springer-Verlag.
- [5] B. Minten, R. Murphy, J. Hyams, and M. Micire. Low order complexity vision-based docking. *IEEE Transactions on Robotics and Automation*, 17(6):922–930, December 2001.
- [6] D. Schulz and D. Foz. Bayesian color estimation for adaptive vision-based robot localization. In *Proceedings of IROS*, 2004.
- [7] Mohan Sridharan and Peter Stone. Autonomous color learning on a mobile robot. In *AAAI*, pages 1318–1323, 2005.
- [8] J.J. Verbeek. *Mixture models for clustering and dimension reduction*. PhD thesis, Universiteit van Amsterdam, December 2004.
- [9] J.J. Verbeek, N. Vlassis, and B.J.A. Kröse. Efficient greedy learning of Gaussian mixture models. *Neural Computation*, 15(2):469–485, 2003.

Apple Defect Segmentation by Artificial Neural Networks

Devrim Unay ^a Bernard Gosselin ^a

^a *TCTS Lab., Faculté Polytechnique de Mons, Belgium*

Abstract

This paper presents a defect segmentation work for bi-colored apple fruits performed by several artificial neural networks. Pixel-wise classification approach is employed to realize segmentation. Quantitative and qualitative evaluations showed that competitive networks were more erroneous while feed-forward and recurrent networks tested were more accurate in segmenting apple defects.

1 Introduction

Quality of apple fruits depends on size, color, shape and presence-type of defected skin according to the marketing standard of European Commission [1]. Visual inspection of apples with respect to size and color by machine vision is already automated in the industry. However, detection of defects is still problematic due to high variance of defect types, presence of stem/calyx concavities and natural variability of skin color. Defect detection, in particular, requires their precise segmentation.

Majority of the works performing defect segmentation of apples used simple thresholding [4, 5, 13, 14, 17]. Wen and Tao [20] applied an adaptive spherical transform before simple thresholding to eliminate boundary light reflectance effect caused by the spherical shape of the fruit. Kim et al. [8] used globally adaptive thresholding (modified version of Otsu's algorithm) to segment fecal contamination defects on apples.

Region-based techniques segment images by finding coherent, homogeneous regions subject to a similarity criterion. Yang [21] used a region-based technique (flooding algorithm) to segment patch-like defects on apples.

Classification-based techniques attempt to partition pixels into several classes using different classification methods. Bayesian classification is the most used method by researchers [3, 10, 12, 15], where pixels are compared to a pre-calculated model and classified as defected or healthy. On the other hand, Nakano [16] introduced a neural network based system to classify pixels of apple skin into six classes, one of which was 'defect'. Unsupervised classification does not benefit any guidance in the learning process due to lacking target values. Such an approach was used by Leemans et al. [11] for defect segmentation.

Above literature survey reveals that in segmenting surface defects of apple fruits, researchers have mainly focused on global thresholding-based approaches and Bayesian-based classification methods. Furthermore, defect segmentation on apples is a problematic task due to several facts. Hence, in this paper we focus on segmentation of skin defects of apples using sophisticated classifiers; namely artificial neural networks (ANN).

2 Methodology

We propose to segment skin defects of bi-colored apples as in Figure 1. After describing the image acquisition and the database, we will explain this system as individual subsections.

2.1 Image Acquisition and Database

Imaging of fruits is performed in a diffusely illuminated environment by a multispectral system based on a high-resolution monochrome camera and four bandpass filters (centered at 450, 500,

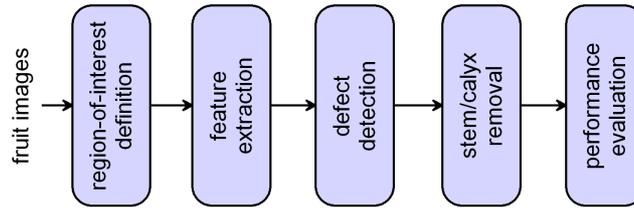


Figure 1: Architecture of the system used for defect segmentation.

750, and 800 nm with respective bandwidths of 80, 40, 80, and 50 nm). Figure 2 displays an illustration of the image acquisition system, which is capable of one-view inspection only. Each filter image is composed of 430x560 pixels with 8 bits-per-pixel resolution.

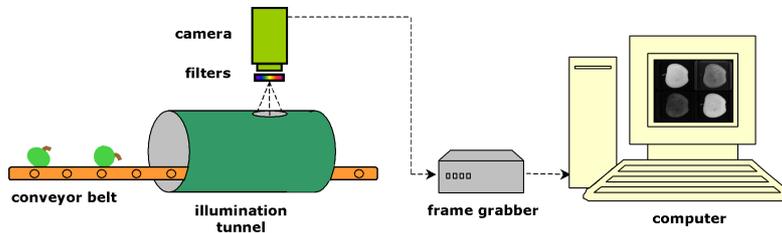


Figure 2: Illustration of the image acquisition system used.

246 Jonagold apples with various naturally and artificially created defects (russet, bruise, rot, scald,...) are tested in this study. Jonagold variety is selected, because it has a bi-colored skin causing more difficulties in defect segmentation due to color transition areas. Defected skins in the image database are manually segmented by experts and used as reference in this work. Figure 3 displays some examples from the database with related manual segmentations.

Assembly of the image acquisition system and collection of the database were done in the Mechanics and Construction Department of Gembloux Agricultural University of Belgium. Therefore, for more details please refer to [9, 10].

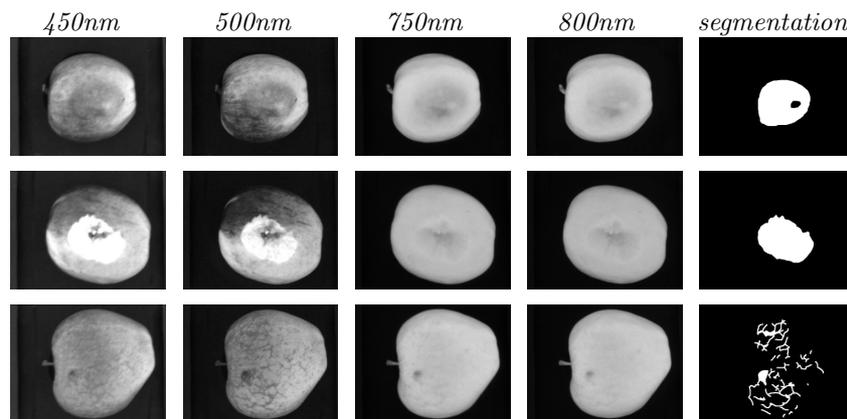


Figure 3: Examples of apple images and related manual segmentations. First four columns present images from different filters, while the last one shows corresponding manual segmentations. Rows display apples with different defect types (Top to bottom: bruise, flesh damage and russet).

2.2 Region-of-Interest Definition

As observed in Figure 3, background is lower than the fruit area in intensity. Therefore, fruit area can be separated from background by thresholding the 750nm filter image at intensity value of 30. However, this fixed thresholding can falsely remove some defects, stems or calyxes that are lower in intensity. Hence, morphological filling is applied to eliminate this error.

Initial observations revealed that segmentation was problematic at the far edges of fruits probably due to illumination artifacts. Hence, after background removal, fruit area is eroded by a rectangular structuring element with size adaptive to fruit size (15 % of fruit’s bounding-box). Result of this erosion step gives the region-of-interest (*roi*) that defines the fruit area to be inspected.

2.3 Feature Extraction

Segmentation of defects at pixel level requires each pixel to be represented by features. Thus, intensity values of each pixel from four filter images form its local features. In our previous work [18] we showed that an additional local feature related to pixels’ location relative to geometric center of *roi* improved segmentation of defects on the same database.

In addition to the local features, average and standard deviation of intensity values over the *roi* are also calculated from each filter image, making up the global features. Hence, each pixel is represented by 13 features in the feature space (Table 1). Feature values are also normalized to fall into the range of [-1,+1] before being introduced to the defect detection step.

category	description	quantity
local	intensity of pixel	4
	pixel’s location relative to <i>roi</i> center	1
global	average of intensities in <i>roi</i>	4
	standard deviation of intensities in <i>roi</i>	4

Table 1: Details of features extracted for defect segmentation.

2.4 Defect Detection

Emerging from the studies on how animal brains work, ANNs are composed of layers of neurons gathered in a parallel architecture with a high degree of interconnection between them [2, 6, 7]. Among various ANN algorithms provided in the literature, we selected the following ones for defect segmentation.

2.4.1 Feed-Forward Networks

In a feed-forward network data propagates in the forward (from input layer to output layer) direction, thus its neurons have only unidirectional connections (no feedback or same layer neuron-to-neuron connections).

Perceptron

Perceptron (or McCulloch-Pitts neuron) is the simplest form of neural networks that can only solve linearly separable problems. It is a single neuron with threshold activation function.

Multi-Layer Perceptrons (MLP)

MLP is composed of several perceptrons in layered structure. It performs back-propagation learning, where two passes of signals through the network are employed. Forward pass: Input signals are propagated in forward direction, while weights at each layer are fixed and actual output of the network is produced. Error between the actual output and the desired output (label) is calculated. In backward pass, this error signal is propagated backward and weights are adapted a second time.

Cascade Forward Neural Networks (CFNN)

CFNN is similar to MLP, except that the neurons of each subsequent layer have inputs coming from not only the previous layers but also input layer. This results in more interconnections than MLP.

2.4.2 Competitive Networks

In competitive networks, the output neurons compete among themselves for being the one to be activated. Competitive networks can learn to detect correlations between inputs and adapt themselves for that. These networks can also include lateral connections (same layer neuron-to-neuron connections) among themselves.

Competitive Neural Networks (CNN)

The most straightforward example of competitive networks is the CNN, which is an unsupervised method that can learn to recognize groups of similar inputs. CNN has lateral connections among neurons.

Learning Vector Quantizers (LVQ)

LVQ is composed of a competitive layer followed by a linear layer (a layer of neurons with linear activation function), where the former learns to classify inputs and the latter transforms the outputs of the former into labels defined by user. Therefore, these networks benefit from supervised learning.

Self Organizing Feature Maps (SOM)

SOM (Kohonen maps) is a competitive network composed of one or two-dimensional lattice of neurons. In SOM, presentation of similar inputs lead to the response of the same group of neurons (physically close to each other), while others stay quiet. It is an unsupervised algorithm.

2.4.3 Recurrent Networks

Feed-forward networks allow progression of inputs only in the forward direction. However, introduction of a feedback path between layers can unveil exclusive information, which is implemented by recurrent networks. Due to the feedback path, information flows in both forward and backward directions, which makes recurrent networks dynamic systems.

Elman Neural Networks (ENN)

ENN is a two-layered recurrent network with feedback from the first layer output to the first layer input. In addition to the feedback connection, it also permits supervised learning.

All the ANNs used in this study have 2-layered architecture with 5 neurons in the hidden layer except for SOM, which has a 5x8 hexagonal topology. Sigmoid neurons are used as long as architecture permitted. Levenberg-Marquardt algorithm, learning rate of 0.01 and maximum epoch number of 25 are used for their training. Above parameters are found optimum after several trials.

2.4.4 Training of ANNs

Pixel-wise segmentation leads to excessive amount of data to be processed, which is computationally very expensive. Thus we performed our tests by *hold-out* method, where $\frac{1}{2}$, $\frac{1}{6}$ and $\frac{1}{3}$ of the fruits of each defect type are placed in training, validation and test databases, respectively, providing 116 fruits for training, 38 for validation and 92 for testing. This separation is found optimum after several trials.

Training set is composed of a set of pixels that are randomly selected from fruits in the training database. Furthermore, defect types and classes (healthy-defected) are equally represented in the training set. We paid special attention on discarding the pixels in the training set that belonged to the fruit being tested, in order to prevent possible forced training.

Cross-Validation is a training method for supervised classifiers, where a portion of training set is separated as validation data and training of the classifier is done on training set while evaluation on validation set. Perceptron, MLP, CFNN and ENN permitted cross-validation, thus they are trained with this method.

2.5 Stem/Calyx Removal

As orientation of fruits were not controlled during image acquisition, stem and calyx parts of fruits are also visible in the images. Discrimination of these natural parts from real defects by image processing is necessary to achieve more precise segmentation. Recently, we have introduced a highly accurate support vector machines-based method to solve this problem [19]. Hence, we benefit from this method to remove stem/calyx regions from segmented areas and provide refined segmentation (Figure 4).

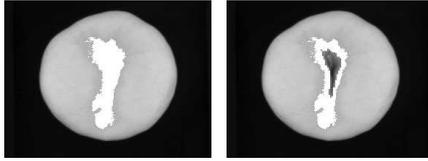


Figure 4: Example of stem/calyx removal. Before the removal on the left, and stem/calyx removed on the right. Defected area displayed in white.

2.6 Performance Evaluation

The popular recognition error assumes that classes are equally represented, which is not true for our case where defect sizes highly vary within the database. Hence, we measure performance of segmentation by calculating class-specific recognition error (CSRE) as follows:

$$CSRE = \frac{\frac{FN}{TP+FN} + \frac{FP}{TN+FP}}{2} \quad (1)$$

where

TP (True Positives): number of defected pixels correctly detected

FP (False Positives): number of healthy pixels incorrectly detected as defect

TN (True Negatives): number of healthy pixels correctly detected

FN (False Negatives): number of defected pixels incorrectly detected as healthy.

If a class does not exist (e.g. no defected skin, i.e. fruit in perfect quality), then the error is computed using the other class only. Note that this measure is calculated for each test image, whereas error of a test is estimated as the average of measures of all test images.

3 Results and Discussion

We will first introduce quantitative analysis of segmentation results estimated by CSRE measure. Figure 5 displays segmentation performances of all the artificial neural networks. We observe that generally competitive networks are more erroneous than the others. Perceptron is the most inaccurate method with its 0.526 error rate, probably because it can only solve linear problems by definition. Feed-forward networks slightly outperform the recurrent one (ENN) in accuracy, which may be due to the feedback connection. Concerning learning, supervised methods surpass unsupervised ones in general, which shows that supervision through labelled training samples results in more accurate segmentation of defects.

We also performed qualitative analysis by visual comparison of segmentation results with respect to manual segmentations. Figure 6 displays some examples of segmentations executed by ANNs with corresponding manual segmentations. As observed from segmentations, Perceptron cannot find the defected skin at all and this is why its error rates were relatively very high. Coherent with the observations of quantitative analysis, segmentations of competitive networks (CNN, SOM and LVQ) are quite erroneous, whereas those of feed-forward (MLP and CFNN) and recurrent (ENN) networks are very promising in general. When we focus on the segmentations performed by feed-forward and recurrent networks we observe over-segmentation in limb rub; under-segmentation in russet, rot and bruise; and very accurate segmentation in flesh damage defects. Precision in the latter is probably due to the defect that is clearly distinct from healthy skin in 450 and 500nm images (see Figure 3). As seen in its manual segmentation russet defect is relatively very complex in shape. Despite this complexity, MLP, CFNN and ENN achieve very encouraging segmentations for this defect type. We observe false segmentations (e.g. quasi-circular ring in limb rub), which are due to the concavities around the stem or calyx areas. Pixels at these parts have relatively low intensities because of varying illumination caused by the curvature.

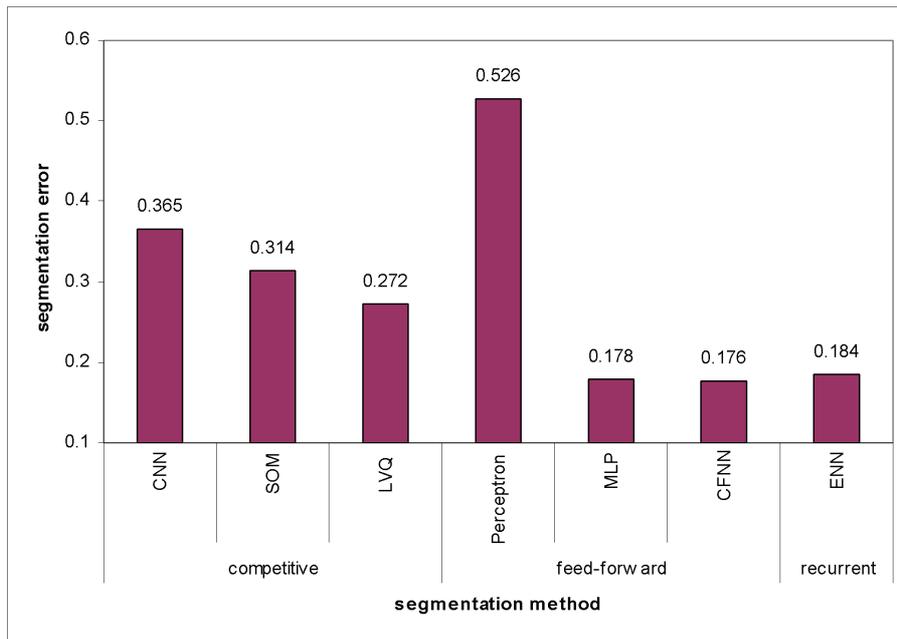


Figure 5: Performances of artificial neural networks for defect segmentation.

4 Conclusion

Detection of defects on bi-colored apple fruits by image processing is a problematic task due to stem/calyx areas and diverse types of defects present, as well as natural variability of skin color.

In this paper we addressed this problem using an approach based on pixel-wise classification of fruit skin by different artificial neural networks. Following classification, segmentations are refined by an SVM-based stem/calyx removal method. Final segmentations are examined by qualitative and quantitative analysis, both of which showed that competitive networks were more erroneous while feed-forward and recurrent networks were quite accurate for apple defect segmentation. Some of the false segmentations were due to the natural variability of fruit shape, while others were probably because of the complexity of defects.

A machine vision-based inspection system has to assign fruits to the corresponding quality categories. Therefore, a necessary future work is to perform fruit grading based on defect segmentations achieved here. Furthermore, evaluation of segmentations are accomplished with respect to a single set of manual segmentations in this work. Hence, repeatability of segmentations should be tested with a new database.

5 Acknowledgements

This research is funded by the General Directorate of Technology, Research and Energy of the Walloon Region of Belgium with Convention No 9813783. We thank Prof. M.-F. Destain, O. Kleyen and V. Leemans from Gembloux Agricultural University of Belgium for their support.

References

- [1] Anonymous. Commission regulation (ec) no 1619/2001 of 6 august 2001 on marketing standards for apples and pears. *Official Journal L 215*, pages 0003–0016, August 2001.
- [2] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press Inc., New York, USA, 1995.

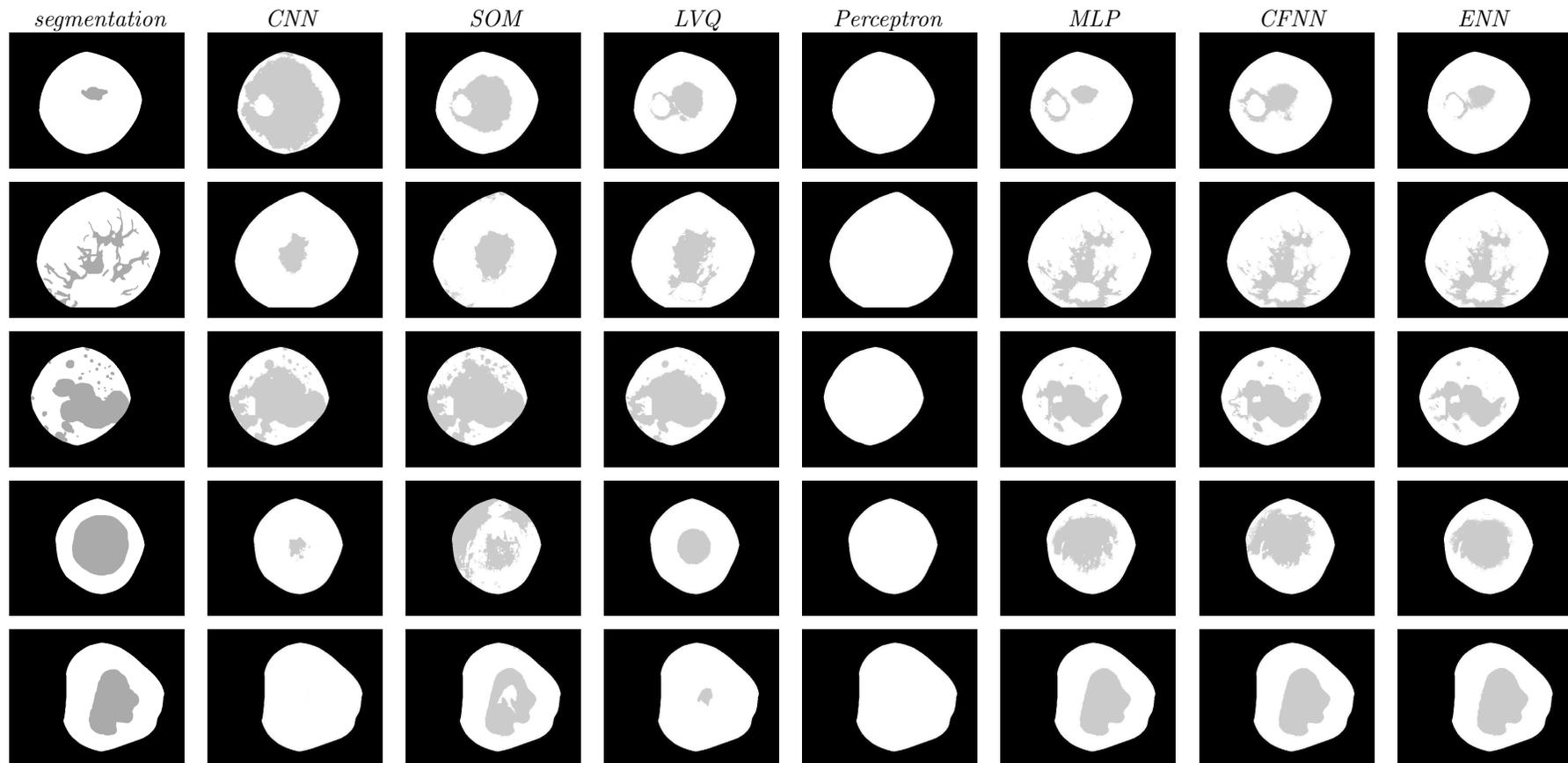


Figure 6: Examples of segmentations achieved by artificial neural networks, where the first column displays manual segmentations. In each image healthy skin is displayed in white color, while defected areas in gray. From top-to-bottom, examples are of fruits defected by limb rub, russet, rot, bruise and flesh damage.

- [3] J. Blasco, N. Aleixos, and E. Moltó. Machine vision system for automatic quality grading of fruit. *Biosystems Engineering*, 85(4):415–423, 2003.
- [4] T.G. Crowe and M.J. Delwiche. Real-time defect detection in fruit - part ii: An algorithm and performance of a prototype system. *Transactions of ASAE*, 39(6):2309–2317, 1996.
- [5] A. Davenel, Ch. Guizard, T. Labarre, and F. Sevilla. Automatic detection of surface defects on fruit by using a vision system. *Journal of Agricultural Engineering Research*, 41:1–9, 1988.
- [6] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern classification*. John Wiley & Sons, Inc., 2nd edition, 2001.
- [7] S. Haykin. *Neural Networks: A Comprehensive Foundation*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 1994.
- [8] M.S. Kim, A.M. Lefcourt, Y.-R. Chen, and Y. Tao. Automated detection of fecal contamination of apples based on multispectral fluorescence image fusion. *Journal of food engineering*, 71(1):85–91, 2005.
- [9] O. Kleynen, V. Leemans, and M.-F. Destain. Selection of the most efficient wavelength bands for ‘jonagold’ apple sorting. *Postharvest Biology and Technology*, 30(3):221–232, December 2003.
- [10] O. Kleynen, V. Leemans, and M.-F. Destain. Development of a multi-spectral vision system for the detection of defects on apples. *Journal of Food Engineering*, 69(1):41–49, 2005.
- [11] V. Leemans, H. Magein, and M.-F. Destain. Defect segmentation on ‘golden delicious’ apples by using colour machine vision. *Computers and Electronics in Agriculture*, 20(2):117–130, July 1998.
- [12] V. Leemans, H. Magein, and M.-F. Destain. Defect segmentation on ‘jonagold’ apples using colour vision and a bayesian classification method. *Computers and Electronics in Agriculture*, 23(1):43–53, June 1999.
- [13] Q. Li, M. Wang, and W. Gu. Computer vision based system for apple surface defect detection. *Computers and Electronics in Agriculture*, 36(2-3):215–223, November 2002.
- [14] P.M. Mehl, K. Chao, M. Kim, and Y.R. Chen. Detection of defects on selected apple cultivars using hyperspectral and multispectral image analysis. *Applied Engineering in Agriculture*, 18(2):219–226, 2002.
- [15] E. Moltó, J. Blasco, and J.V. Benlloch. Computer vision for automatic inspection of agricultural produces. In *SPIE Symposium on Precision Agriculture and Biological Quality*, Boston, MA, USA, November 1998.
- [16] K. Nakano. Application of neural networks to the color grading of apples. *Computers and Electronics in Agriculture*, 18(2-3):105–116, August 1997.
- [17] J.A. Throop, D.J. Aneshansley, W.C. Anger, and D.L. Peterson. Quality evaluation of apples based on surface defects: development of an automated inspection system. *Postharvest Biology and Technology*, 36:281–290, 2005.
- [18] D. Unay and B. Gosselin. A quality sorting method for ‘jonagold’ apples. In *Proc. Int. Agricultural Engineering Conf. (AgEng)*, Leuven, Belgium, September 2004.
- [19] D. Unay and B. Gosselin. Stem and calyx recognition on ‘jonagold’ apples by pattern recognition. *Journal of Food Engineering*, to appear, DOI: 10.1016/j.jfoodeng.2005.10.038.
- [20] Z. Wen and Y. Tao. Building a rule-based machine-vision system for defect inspection on apple sorting and packing lines. *Expert Systems with Applications*, 16(3):307–313, April 1999.
- [21] Q. Yang. An approach to apple surface feature detection by machine vision. *Computers and Electronics in Agriculture*, 11(2-3):249–264, March 1994.

A Generalized Matching Framework : Combining Matchmaking and Coalition Formation

Jeroen M. Valk ^a

Duco N. Ferro ^{ab}

^a *Almende B.V., Westerstraat 50, 3016 DJ Rotterdam*

^b *Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam*

Abstract

This paper presents a matchmaking framework for a new matchmaking component in the ASK communication platform (<http://www.ask-cs.nl/>). The framework allows a variety of matchmaking techniques to be applied in a single automated matchmaking system. A variety of settings are generalized in a way that combines coalition formation with typical matching settings such as perfect/maximal matching and bi/tri-partite matching. Our framework turns out to be useful to categorize existing work on coalition formation and matching, but at the same time allows us to identify new domains to be studied.

1 Introduction

The problem of matching together individuals in an effective way has been studied in a variety of different settings, e.g., perfect/maximal matching [5, 6], bi/tri-partite matching [14, 9, 10], or coalition formation [8, 1, 7, 11]. Originally, matchmaking is concerned with bringing together individuals into couples [19, 2]. This particular setting is commonly represented using a graph. The nodes of the graph are individuals and two individuals that are connected by an edge could potentially form a match. In particular, a special case where the matching graph is bipartite is studied extensively. This so-called bipartite matching problem has many applications.

Many situations occur in everyday life where one group must be mapped on another, e.g., employees/jobs, patients/doctors, consumers/products. At a first glance, it seems logical to model these matching problems using bipartite graphs. However, even in typical bipartite circumstances it is often useful to take additional dimensions into account. A patient, for example, goes to the doctor for certain reason, i.e., a certain expertise is required. This extra dimension should be taken into account, because the quality of the match depends on it. While the latter type of matching is known as 3-dimensional matching, theory on matching in more than three dimensions can, as far as we know, only be found in the area of coalition formation.

Coalition formation was originally studied in cooperative game theory. However, coalitions are not formed to bring together individuals on a certain occasion, but rather to suggest possible cooperations between players in a non-cooperative game. If two players are in the same coalition this means that they are fully cooperative in order to maximize their collective payoff. That is, the members of a coalition act as a single player in the game. Since players in a non-cooperative game can always improve on their collective payoff by cooperating, it was traditionally assumed that players will always form the largest possible coalition ¹. The main concern of cooperative game theory is therefore how to divide the collective gain over the members of the coalition.

More recently, there is an interesting trend in research on coalition formation to dispute the traditional arguments. For reasons such as, e.g., bounded rationality, incomplete information, or communication overhead in large coalitions, it is argued that it is sometimes undesirable to form large coalitions. As a result, there is an increasing interest in games that are not superadditive. In our opinion, this trend makes that matchmaking and coalition formation become more and more related. Some researchers recognized this relationship between matching and coalition formation.

¹The precise argument in traditional game theory is based on the observation that all games that are put in coalitional form are superadditive.

In [16], for example, pair partnership matching of elements from an unordered group is treated as a variant of the general coalition formation process. In pair coalition formation an agent is satisfied with a coalition of itself and only one other partner.

A comprehensive framework that combines the many aspects of matchmaking and coalition formation is still missing however. The aim of this paper is to fill in this gap. We present a matching framework that concerns matching structures which contain variable-sized groups of individuals and support different (possibly conflicting) opinions on the effectiveness of groups. The framework (i) generalizes existing matchmaking and coalition formation models, (ii) allows new domains to be studied, and (iii) is used for the implementation of a matchmaking system developed at Almende B.V., a research company situated in Rotterdam. The outline of the paper is as follows. First, Section 2 presents the generalized matchmaking framework. Next, Section 3 shows how the framework is related to existing models from the literature. Then, in section 4 we give an illustration of how the framework can be applied in a healthcare domain and we end in Section 5 with the conclusions.

2 A Generalized Matching Framework

In this section we present a matching framework that generalizes existing approaches to match-making and cooperative game theory. First we provide some basic notions in our model, i.e. the matching environment, matching evaluation, the matching structure and control mapping. Then we define a notion of (un)stability. Finally we give some properties each representing different aspects of matching.

2.1 The Basic Framework

In many applications, matchmaking is driven by the fact that individuals (e.g., users or other entities) request to be matched to other individuals or items. In the health care domain, for example, matchmaking concerns users requesting a particular medical treatment, preferably, by the best doctor available.

In its most general form, the matching problem we are interested in consists of finding an effective grouping of elements. We distinguish *unordered* and *ordered* groups. Given a countable set E of elements, an unordered E -group is simply a finite subset of E . This means that the group has no structure at all. An ordered E -group, on the other hand, is a vector $(e_1, \dots, e_i, \dots, e_n)$ such that each $e_i \in E$. This means that each element has a certain role; the total order which is associated with an ordered group is used to establish the roles of the elements in the group, e.g. buyer or seller, patient or care provider, etc.

Matching elements are either *passive* elements, e.g. a book or a service, or *active* elements that can be considered parts of which a particular individual (agent) has full *control*. Some individuals may be represented by more than one element. These individuals are said to be in control of these elements. The idea is that active elements, in contrast to passive elements, take into account the opinion, which their controlling individual has about the groups that are formed in the matching process. The active elements of which an individual has control represent the different roles that the individual possibly has, e.g., employee, patient, owner, seller, father, etc. The quadruple, containing all individuals, active elements, passive elements and the control function for active elements, is called the *matching environment*, or sometimes simply called the *environment*.

Definition 2.1 (Matching Environment) *A (matching) environment $M = (\mathcal{A}, A, B, U)$ is a quadruple with:*

- (i) \mathcal{A} the set of agents that exercise control, and
- (ii) A a countable set of all active elements, and
- (iii) B a countable set of all passive elements, and
- (iv) U a control mapping, given by the surjective function $U : A \rightarrow \mathcal{A}$ that maps active elements to agents.

Note, we will write E to refer to the set of all elements, both active and passive. If the control function U is injective as well then it is said to be *localized*. In this situation, we will assume U to be the identity mapping on A , i.e., $\mathcal{A} = A$ and $U(a) = a$ for all $a \in A$. This means that every agent is represented by one active element exactly.

Given a matching environment, matchmaking is concerned with finding a particular configuration of elements into groups. The configuration must be such that the commitments of the elements with respect to the group are clear. For this reason, we adopt an assumption that is common in the literature: i.e., each element can belong to only one group at the same time [17, 13]. In this way, the element can be fully committed to the group to which it belongs. Note that, in our framework, this means that agents can still have control over elements in different groups.

Definition 2.2 (Matching Configuration) *Given a matching environment $M = (\mathcal{A}, A, B, U)$, with $E = A \cup B$, a (matching) configuration in M is a set \mathcal{C} of E -groups such that every $e \in E$ occurs in exactly one member of \mathcal{C} .*

In the literature, the concept of a matching configuration appears frequently, although under different names. Cooperative game theory, for example, often uses the term *coalition structure* [20]. In combinatorics the matching configuration is known under the name *matching* or *perfect matching* [6].

Not all matching configurations are equally good. In general, there may be different opinions on the value of a matching configuration. We define an M -opinion as follows, assuming that an opinion on a matching configuration is formed by the opinions on the individual groups in the configuration.

Definition 2.3 (M -opinion) *Let $M = (\mathcal{A}, A, B, U)$ be a matching environment. An M -opinion is a pair (\mathcal{V}, v) where $\mathcal{V} \subseteq \bigcup \Gamma$ and $v : \mathcal{V} \rightarrow \mathbb{R}$.*

The part of cooperative game theory that deals with so-called transferable utility (TU) games assumes that there is only one opinion and that this single opinion is complete [12]. The reason being that in a TU game all utilities are assumed to be monetary and therefore any disagreement about the value of a coalition can be eliminated using side-payments, i.e., changing the distribution of the collective gain over the members of a coalition.

Not everything can be compensated by side-payments however. In such settings, different opinions on the values of coalitions can be distinguished. In an environment where different opinions exist, what actually happens depends on how control is distributed over the different opinions. We define an M -evaluation as a family of M -opinions where the index set, represented by the set of agents, of the family is used to identify the different opinions.

Definition 2.4 (M -Evaluation) *Let $M = (\mathcal{A}, A, B, U)$ be a matching environment. An M -evaluation is a finite family $V = \{(\mathcal{V}_a, v_a)\}_{a \in \mathcal{A}}$ of M -opinions.*

It is not always possible that all matching configurations may form. There may be all sort of constraints on possible combinations of coalitions. The collection of all possible combinations of matching configurations that might be formed is called the *skill set* for the environment.

Definition 2.5 (Matching Skills) *A collection Γ of matching configurations in M is called a skill set for M .*

We can now describe the matching problem in terms of matching environments M , skill sets Γ for these environments and M -evaluations.

Definition 2.6 (Matching Problem) *The matching problem is: given a matching instance $I = (M, \Gamma, V)$ with $M = (\mathcal{A}, A, B, U)$ a matching environment, Γ a skill set, and V a M -evaluation, find a matching configuration $C \subseteq \Gamma$ with some desirable properties.*

In the next section, we will introduce one such property, which we left unspecified in the above definition.

	v_X	v_Y	v_x	v_y		v_X	v_Y	v_x	v_y
$\{X\}$	4	2	0	0	$\{X, y\}$	9	4	4	9
$\{Y\}$	2	4	0	0	$\{Y, y\}$	4	8	3	6
$\{x\}$	0	0	4	2	$\{X, x\}$	6	3	8	4
$\{y\}$	0	0	2	4	$\{Y, x\}$	0	2	2	0

Table 1: an M -evaluation

2.2 Strongly Unstable Matching Configurations

An interesting property of a matching configuration is its (un)stability. Intuitively, a matching configuration is unstable, if there exists a coalition of agents that has enough influence over the agents outside the coalition to deviate from the matching configuration in such a way that each agent in the coalition is individually better off. This means that unstability depends on the possibility of a transition $\mathcal{C} \xrightarrow{\mathcal{A}_0} \mathcal{C}'$ influenced by $\mathcal{A}_0 \subseteq \mathcal{A}$ from matching configuration \mathcal{C} to \mathcal{C}' . It is difficult to provide an exact characterization of the above transition relation, because it depends on the strategy adopted by the agents outside the coalition whether the transition will succeed or not. Nevertheless, sufficient conditions can be given for which the transition is guaranteed to succeed under the assumption of rational agents.

A sufficient condition on unstability that is often provided in the literature can be formulated as follows. Let $\mathcal{A}_0 \subseteq \mathcal{A}$ be a coalition of agents. We now only consider transitions to matching configurations where only isolated regroupings among the active elements under control of \mathcal{A}_0 are allowed. That is, if $\mathcal{C} \xrightarrow{\mathcal{A}_0} \mathcal{C}'$ then there exists a subset $\mathcal{C}_0 \subseteq \mathcal{C}'$ such that for every $e \in E$, e occurs in \mathcal{C}_0 iff e is an active element under control of \mathcal{A}_0 , i.e., $U(e) \in \mathcal{A}_0$. Under the above restriction, \mathcal{A}_0 can guarantee that agents outside the coalition can only regroup the partial matching configuration $\mathcal{C}' \setminus \mathcal{C}_0$. This means that \mathcal{A}_0 can guarantee a new coalition structure that belongs to $\Gamma_0 = \{\mathcal{C} \mid \mathcal{C}_0 \subseteq \mathcal{C}\}$. If in all these possibilities, each member of \mathcal{A}_0 is better off then \mathcal{C} must have at least one outgoing transition, and therefore, \mathcal{C} is unstable.

The above considerations can be summarized in the following very strong notion of unstability.

Definition 2.7 (Strong Unstability) *Let $I = (M, \Gamma, V)$ be a matching instance with $M = (\mathcal{A}, A, B, U)$. A matching configuration $\mathcal{C} \in \Gamma$ is called I -unstable if there exists a subset $\mathcal{A}_0 \subseteq \mathcal{A}$ and a partition \mathcal{C}_0 of $A_0 = \{a \in A \mid U(a) \in \mathcal{A}_0\}$ such that $\Gamma_0 = \{\mathcal{C}' \in \Gamma \mid \mathcal{C}_0 \subseteq \mathcal{C}'\}$ is non-empty and:*

$$\sum_{\mathcal{C} \in \mathcal{C} \cap \mathcal{V}_a} v_a(\mathcal{C}) < \min_{\mathcal{C}' \in \Gamma_0} \sum_{\mathcal{C} \in \mathcal{C}' \cap \mathcal{V}_a} v_a(\mathcal{C}),$$

for every $a \in \mathcal{A}_0$.

Let us give a simple example to illustrate that stability definitions in existing literature [4, 15] provide a strong notion of unstability of matching configurations.

Consider a kind of stable marriage problem on a deserted island with only two men X and Y , and two women x and y . The possible couples that can be formed are $\{X, x\}$, $\{X, y\}$, $\{Y, x\}$, and $\{Y, y\}$. Man X , for instance, has different preferences for $\{X, x\}$ and $\{X, y\}$, but considers Y to be his friend as well. This means that, although X prioritizes its own partner selection, he also cares about whether $\{Y, x\}$ or $\{Y, y\}$ will be formed. Now we assume that: (i) the same holds for Y , (ii) the women's situation is symmetric to the men's situation, and (iii) each man/woman considers the situation of his/her friend to be at most half as important as his/her own situation. Under these assumptions, an interesting example of a matching model is as follows.

Example 2.8 (Stable Marriage) *Consider the matching instance $I = (M, \Gamma, V)$ where (i) $M = (\mathcal{A}, A, B, U)$ with $A = B = (\{X, Y, x, y\})$ and $\Gamma = \{\{X\}, \{Y\}, \{x\}, \{y\}, \{X, x\}, \{X, y\}, \{Y, x\}, \{Y, y\}\}$, (ii) U is the identity mapping on A , and (iii) V is the family of M -opinions shown in Table 1. Note that for every $\mathcal{C} \in \Gamma$, if x_1 and x_2 are friends and x_1 occurs in \mathcal{C} then $v_{x_1} \geq 2 \cdot v_{x_2}$.*

Now, consider the matching configuration $\mathcal{C} = \{\{A, a\}, \{B, b\}\}$. Note that, $v_A(\mathcal{C}) = v_A(\{A, a\}) + v_A(\{B, b\}) = 6 + 4 = 10$, and $v_b(\mathcal{C}) = v_b(\{A, a\}) + v_b(\{B, b\}) = 4 + 6 = 10$. To check whether A and b have an incentive to deviate from \mathcal{C} , according to Definition 2.7, we need take into account

the *minimum* of A 's and b 's evaluations of $\Gamma_0 = \{\{\{A, b\}, \{a, B\}\}, \{\{A, b\}, \{a\}, \{B\}\}\}$. This yields $\min\{v_A(\{A, b\}) + v(\{a, B\}), v(\{A, b\}) + v(\{a\}) + v(\{B\})\} = \min\{9 + 0, 9 + 0 + 2\} = 9$. But $10 \geq 9$, which means that the formation of $A_0 = \{A, b\}$ is not sufficient to show \mathcal{C} 's unstability. However, this conclusion is based on the formation of $\{\{A, b\}, \{a, B\}\}$, and A could know that B and a would prefer to stay alone then to join together. This, $\{\{A, b\}, \{a\}, \{B\}\}$ is actually the only matching configuration that is feasible within Γ_0 . Based on this observation, it is not difficult to see that \mathcal{C} should be classified as unstable. Therefore, we may conclude that Definition 2.7, in some circumstances, provides a notion of unstability that is too strong.

2.3 Properties of the Framework

Given a triple $I = (M, \Gamma, V)$ with M a matching environment, Γ the skill set, and V an M -evaluation, the matching problem in general searches for a matching configuration with some desirable properties. It depends on the structure of I which are the relevant properties that the matching problem should take into account:

- (i) a matching instance I is called *ordered/unordered*, if all the E -groups in the members of Γ are *ordered/unordered*; otherwise I is called *mixed*.
- (ii) a matching instance I is called n -partite, if there exists a partition $\{E_i\}_{1 \leq i \leq n}$ of E such that every E -group belonging to one of the members of Γ contains exactly one element from every E_i for $1 \leq i \leq n$;
- (iii) given subset $N \in \mathbb{N}$, a matching instance I is called N -sized, if for every E -group that occurs in Γ has a size that belongs to N ;
- (iv) a matching instance I is *globalized* if for every $a_1, a_2 \in A$, $U(a_1) = U(a_2)$, and I *localized* if U is injective;
- (v) a matching instance I is *bounded*, if for every E -group C in V_a , there exists a $b \in C \cup A$, such that $U(b) = a$; alternatively if I is not bounded, we call I *social*.

As a convention we assume the n -partite matching instance is either ordered or unordered. Note that a strict n -partite matching instance is also singleton N -sized, where $N = \{n\}$. These above properties allow us to categorize existing approaches to matching, but at the same time allows us to identify new domains to be studied.

3 Matching Framework & Related Work

As mentioned earlier, in existing literature, different terms are used to refer to the matching framework we defined. It is interesting to see that the model we present is able to capture essentials of different approaches to matching. In this section, we give a description of existing approaches to matching and coalition forming and show how it relates to our work. First, we present the relation between stable marriage and our framework. Second, we discuss matching as it is often approached in combinatorial mathematics. Third, we illustrate how cooperative game theory commonly approaches the matching problem.

3.1 Matching : Stable Marriage

Stable marriage can be considered as a special case of a matching instance $I = (M, \Gamma, V)$. A stable marriage setup is typically represented by two finite sets A_1 and A_2 of n elements each, and for each member in the one set, a preference relation on the members in the other set. In terms of our framework this boils down to a matching instance I that is bipartite, localized, bounded and where E is finite.

Traditionally stable marriage uses total orders instead of M -opinions to represent the preferences of individuals [3]. This means that a family $\{\preceq_x\}_{x \in A}$ is given, where $y_1 \preceq_x y_2$ iff the $v_x(\{x, y_1\}) \leq v_x(\{x, y_2\})$. Using the total orders, the stability condition is defined as follows.

Definition 3.1 (Unstable Marriage) *Given a matching instance $I = (M, \Gamma, V)$, a matching configuration C in I is called unstable, if there exist two E -groups $\{X, x\}$ and $\{Y, y\}$, such that $x \preceq_X y$ and $Y \preceq_y X$.*

It can be proven that the above definition is a special of Definition 2.7.

A variant on stable marriage is the, non-bipartite, stable roommate problem. We can represent this variant by a matching instance that is $\{2\}$ -sized but not bipartite. For the stable roommate problem there does not always exist a stable matching configuration.

3.2 Matching : Maximum Cardinality

Combinatorial matching algorithms usually deal with matching problems of forming unordered pairs as groups. Garey and Johnson, for example, define the matching problem as follows [6].

Definition 3.2 (Matching Problem(Garey and Johnson)) *Given a graph $G = (V, E)$, find a set $E' \subseteq E$ of edges such that every vertex $v \in V$ is incident with at most one edge in E' ; if every vertex is incident with at least one edge in E' as well, then a matching M of E' is called perfect.*

It can be seen that the matching problem can be modelled using an unordered, $\{2\}$ -sized matching instance. Since the matching problem searches a so-called perfect matching, where every individual is matched in a pair, there are no M -opinions at all. That is, every element is passive.

An optimization variant of the matching problem is called maximal matching. In the maximal matching, the matching instance is unordered as well, but $\{1, 2\}$ -sized: we have both the singleton sets and unordered pairs as E -groups. The maximum matching searches for a maximum cardinality set of matched pairs. This is modelled by a globalized control mapping and a single M -opinion that assigns 1 to every unordered pair and 0 to every singleton set.

Another variant on this matching problem is known as weighted matching [18]. A graph can be extended by labeling each edge E of the graph G with a weight. Here, the weights can express a distribution over the preference of a vertex for its incident edges. The goal is to (globally) optimize the sum of weights. Similarly to the maximum cardinality matching, the weights can easily be incorporated in the M -opinions.

3.3 Matching : Coalition Formation

In game theory, the approach to the matching problem is different from the combinatorial view. In cooperative game theory, for example, games are considered in their so-called *coalitional form*. The relation of matching to coalition formation is already noticed in [16] where pair partnership matching of elements from an *unordered* group is essentially treated as a variant to the general coalition formation process. In pair coalition formation an agent is satisfied with a coalition of itself and only one other partner.

Let $N = \{1, 2, \dots, n\}$ be the set of players. Any non-empty subset S of N is called a coalition. A coalition S can obtain a total utility v . This utility can be distributed among the members(players) of S . Collectively the utilities of all coalitions are represented by a value function.

Definition 3.3 (Value Function) *The value function in a n -person game is given by $v : 2^A \rightarrow \mathbb{R}$; For each coalition $S \subseteq N$, $v(S)$ representing the amount of utility that the members of S , collectively, can gain from the coalition.*

The idea is to find a partition $\mathcal{C} = \{A_1, \dots, A_n\}$ of A (i.e., the coalition structure) such that the sum of all corresponding values are maximized, i.e., $v(A_1) + \dots + v(A_n)$ is maximized.

A game in coalitional form can be represented by a unordered, globalized, bounded matching instance such that E is finite and Γ contains all partitions of 2^A .

4 Application Scenario: Home Health Care

The generalized matching framework presented in this paper has been applied to further develop a matchmaking component, called the Matcher, within the ASK communication platform. The development and implementation of this novel matching component is carried out as a part of

the research within Almende. In particular, we focus on a health care scenario that is studied in the context of the “Luister” project. In “Luister”, the aim is to realize a personalized and flexible health care service at Humanitas, a national member-based institute for social services and community development. The matchmaking of health care demand and provision is dependent on the availability, skills and preferences of *patients* and *carers*. Patients can articulate their health care demand through a speech interface, i.e., voice recognition on a telephone connection. Consequently the patient uttering a demand is matched to a suitable and available carer. This results in an appointment with the carer, who will be notified of this commitment automatically.

The participants in a match are enabled to express their preferences on the actual experienced health care provision. After the scheduled service is delivered, feedback from both patient and carer is requested. One of the projects research questions is how to use feedback to improve service quality through personalization.

Initially, we use an ordered bipartite matching instance I to model this domain where the partition $\{A_p, A_c\}$ consists of a *finite* set of patients A_p and a *finite* set of carers A_c . We assume that (i) I is localized, (ii) the patients are active elements, and (iii) the carers can be both active and passive. We could further refine the model of the scenario by partitioning the carers into two groups: professionals $A_{c,p}$ and volunteers $A_{c,v}$; the Matcher is designed to make these kind of refinements on-the-fly while the system is running.

Now let us say that professionals are not allowed to have preferences for patient (professional ethics does not allow them to refuse treatment to an individual in need for care), i.e. every element $a_i \in A_{c,p}$ is *passive*, while preferences of volunteers are taken into account in the matching procedure, i.e. every element $a_j \in A_{c,v}$ is *active*, since a pleasant match may motivate them to provide as much care as possible.

Using the flexibility of our framework, the matching component can dynamically add other matching classes or individuals. Additional context that is relevant for the matching can be included whenever this information becomes available. The additional context can be practically everything: e.g., place, time, language, etc. In this way, the system is able to deal dynamically with multi-dimensional matching in variable dimensions.

5 Conclusion

In this paper, we presented a general matching framework covering some representative matching approaches in literature. The main goal of this model is to combine the properties of different matching approaches. The need for such a general model arises in the light of a real-life application scenarios such as presented in section 4.

We have defined a generalized model and criteria for determining the quality of a matching configuration. We related these criteria to the ones used in combinatorial matching algorithms and cooperative game theory. One of the interesting aspects of our model is that our matching configuration can contain groups of variable size. It allows contextual matching, where certain dimensions of the context can be included or discarded depending on the type of match. In addition, our approach carefully takes into account different, possibly conflicting, opinions on the quality of a match, i.e., the effectiveness of a group.

Several aspects of our model are interesting for further research. In the near future, we want to explore new domains with unbounded control mapping. Research will focus on the effect of opinions of an individual about matches in the matching configuration in which it is not involved.

Next to this, we would like to explore techniques for forecasting of missing evaluations. In society people can base their (local) decision making on external information. Friends, colleagues, like-minded people or others they trust can make recommendations or suggestions for good restaurants, interesting books or a pleasant dentist. An individual consults this group of acquaintances for recommendations on possible matches. In particular, we are interested in systems where the forecasting is localized and distributed over a (possibly large) collection of personal agents.

References

- [1] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms, Second Edition*. MIT Press and McGraw-Hill, 2001.
- [2] K. Decker, K. Sycara, and M. Williamson. Middle-agents for the internet. In *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, Nagoya, Japan, 1997.
- [3] R. Diestel. *Graph Theory*, volume 173 of *Graduate Texts in Mathematics*. Springer, New York, third edition edition, 1997.
- [4] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *American Mathematical Monthly*, 69(1):9–15, 1962.
- [5] Z. Galil. Efficient algorithms for finding maximum matchings in graphs. *ACM Computing Surveys*, 18:23–38, 1986.
- [6] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York, NY, USA, 1979.
- [7] M. Klusch and A. Gerber. Dynamic coalition formation among rational agents. *IEEE Intelligent Systems*, 17(3):42–47, 2002.
- [8] M. Klusch and O. Shehory. Coalition formation among rational information agents. In *Seventh European Workshop on Modelling Autonomous Agents in a Multi-Agent World*, Eindhoven, The Netherlands, 1996.
- [9] D.E. Knuth. Stable marriage and its relation to other combinatorial problems. In *CRM Proceedings and Lecture Notes*, volume 10. American Mathematical Society, 1997.
- [10] C. Ng and D. Hirschberg. Three-dimensional stable matching problems. *SIAM Journal On Discrete Mathematics*, 4:245–252, 1991.
- [11] E. Ogston, B. Overeinder, M. van Steen, and F. Brazier. A method for decentralized clustering in large multi-agent systems. In *AAMAS Proceedings of the second international joint conference on Autonomous agents and multiagent systems*, pages 789–796, New York, NY, USA, 2003. ACM Press.
- [12] M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. The MIT Press, 1994.
- [13] T. Rahwan and N. R. Jennings. Distributing coalitional value calculations among cooperative agents. In *Proceedings of the Third European Workshop on Multi-Agent Systems*, page 499, Brussels, Belgium, 2005. Koninklijke Vlaamse Academie van Belie voor Wetenschappen en Kunsten.
- [14] M. Riedel. Online matching for scheduling problems. *Lecture Notes in Computer Science*, 1563:571–580, 1999.
- [15] A. Roth and M. Sotomayor. Two sided matching. *Econometric Society Monographs*, 18, 1989.
- [16] D. Sarne and S. Kraus. Time-variant distributed agent matching applications. In *AAMAS*, pages 168–175, 2004.
- [17] O. Shehory and S. Kraus. Methods for task allocation via agent coalition formation. *Artificial Intelligence*, 101(1-2):165–200, May 1998.
- [18] J. Silc and B. Robic. Processor allocation based on weighted bipartite matching scheduling. Technical Report CSD-TR-97-1, 9, 1997.
- [19] F. Stolzenburg, J. Murray, and K. Sturm. Multi-agent matching algorithms with and without coach. In *Proceedings First German conference on Multiagent System Technologies(MATES)*, pages 192–204. Springer-Verlag, Berlin Heidelberg, Germany, 2003.
- [20] F. Tohme and T. Sandholm. Coalition formation processes with belief revision among bounded-rational self-interested agents. *Journal of Logic and Computation*, 9(6):793–815, 1999.

Reliable Instance Classifications in Law Enforcement

Stijn Vanderlooy Eric Postma Karl Tuyls
Ida Sprinkhuizen-Kuyper

MICC-IKAT, Universiteit Maastricht, PO Box 616,
6200 MD Maastricht, The Netherlands

Abstract

Machine-learning classifiers are gaining interest in the domain of law enforcement. However, when classifiers are applied in this domain two fundamental difficulties are encountered. First, the cost of an incorrect classification is extremely high. Second, the error-cost distribution is not static. In this paper we relate these difficulties to the reliability of instance classifications. Accordingly, we define two requirements to be met by classifiers. It is shown that an approach based on isometrics in Receiver Operator Characteristic (ROC) space is able to construct classifiers that satisfy both requirements. Due to the efficiency and generality of the approach, classifiers can be safely used to solve law enforcement tasks.

1 Introduction

In the last years, terrorist acts and other criminal activities had a strong influence on the politics of nations worldwide. Preventing crime and providing security for civilians have become high-priority goals. To achieve these goals, governments allow more data storage and international data transfers for law enforcement. This has resulted in an overwhelming amount of information that needs to be processed and interpreted in an efficient way. The new form of law enforcement guided by data analysis is known as intelligence led policing [5].

In this paper we focus on law enforcement tasks that can be suitably solved by a machine-learning classifier. Two domain-specific problems arise when a classifier is applied. First, incorrect classifications have serious consequences, e.g., waste of limited resources and privacy violations when personal data are involved. Clearly, the cost of an incorrect classification (error cost) is high. A second problem is the dynamics of the error-cost distribution. This distribution describes the balance between the costs of false positives and false negatives. It varies with the societal and legal context. For example, when terrorist threats are high, ensuring public safety will become more important than securing civilians' privacy. If crime rates are low for a sustained time period, then securing privacy will be more important. The former implies that the error cost of failing to identify an upcoming terrorist attack is substantially higher than the error cost of violating the privacy of innocent civilians. The latter implies a smaller difference between the error costs.

To apply a classifier in law enforcement it has to deal with these two problems.¹ Accordingly, we define two requirements that a classifier has to meet. First, the classifier has to guarantee a minimum level of performance on each class. Second, in order to cope with a dynamic error-cost distribution, it needs to have the ability to adjust its performance in a timely fashion. We will show that so-called reliable classifiers [8, 9] meet both requirements.

The rest of this paper is organized as follows. Section 2 briefly outlines reliable classifiers. Section 3 shows how to construct these classifiers and explains why they satisfy the requirements. In Section 4 we apply the approach on a real-life legal classification task. Section 5 discusses benefits of applying reliable classifiers in law enforcement. Section 6 concludes the paper.

2 Reliable classifiers

We consider classification tasks with two classes: positive (p) and negative (n). To solve a task we construct a scoring classifier. This is a classifier that outputs two positive values $l(x|p)$ and $l(x|n)$ indicating the likelihood that an instance x is positive and negative, respectively. The score of an instance combines these values as follows:

$$l(x) = \frac{l(x|p)}{l(x|n)} \quad (1)$$

and is used to rank instances from most likely positive to most likely negative.

The classification of a new instance is obtained by applying a numerical threshold on the score. Instances with a score higher than or equal to this threshold are classified as positive. The remaining instances are classified as negative. Unfortunately, a scoring classifier is not optimal, i.e., there exists negative instances with a higher score than some positive instances. Therefore, the use of a threshold always results in (too) many incorrect classifications. This implies that we do not know if the classification assigned to a particular instance is correct.

To overcome this problem, an approach to reliable classifiers is defined [8, 9]. A reliable classifier is able to guarantee a specific performance on each class.² To ensure this guarantee, it may be needed to abstain from classifying ambiguous instances. This abstention is implemented as a filtering mechanism with two numerical thresholds $a > b$. An instance x is classified as positive if $l(x) \geq a$. If $l(x) \leq b$, then x is classified as negative. Otherwise, the instance is left unclassified since its score does not indicate the correct classification with sufficient certainty.

3 Reliable Classifiers via ROC Isometrics

Recently, an approach was introduced to construct a reliable classifier with a desired precision on both classes [9]. In subsequent work [8] analysis revealed that the approach can be used for other performance metrics as well, e.g., the

¹Note that the two identified problems can also occur in other domains, e.g., medical diagnosis.

²Thus, by definition, a reliable classifier satisfies the first requirement that we have defined.

F -measure. Thus, performance on each class can be measured with a variety of performance metrics. We will restrict to a classification task for which the minimum achievable performance is denoted in terms of precision on each class. We identify precision on each class with reliability. The reliability is said to be high when the precisions for both classes are high.

The key idea of the approach is to identify instances with uncertain classification. This is done on the basis of a scoring classifier and its representation in terms of a Receiver Operator Characteristic (ROC) curve. If the uncertain instances are left unclassified, then the corresponding classifier obtains the desired reliability. The next three Subsections 3.1, 3.2 and 3.3 discuss in detail this approach and how it allows us to handle error-cost distributions.

3.1 ROC Isometrics

Given a scoring classifier, its ROC curve shows the trade-off between false positive rate fpr and true positive rate tpr for each possible threshold on the score $l(x)$. Figure 1(a) shows an ROC curve.³ A concavity in this curve indicates local sub-optimality of the scoring classifier [3]. To remove concavities the convex hull of the ROC curve (ROCCH) is constructed.

It can be shown that for any point (fpr, tpr) on an ROCCH, a classifier can be constructed that has the performance represented by that point [7]. For simplicity of presentation, we assume that ROC curves are convex and all points can be obtained by a threshold.

Reliability is measured by the precision on the positive classifications ($prec_p^c$) and precision on the negative classifications ($prec_n^c$):

$$prec_p^c = \frac{tpr}{tpr + c fpr} \quad (2)$$

$$prec_n^c = \frac{tnr}{tnr + \frac{1}{c} fnr} \quad (3)$$

Here, $c = \frac{N}{P}$ with N and P the number of negative instances and positive instances, respectively. ROC isometrics are collections of points in the (fpr, tpr) plane with the same value for a performance metric [2]. Precision isometrics are defined by rewriting each precision equation to that of a line in ROC space:

$$tpr = \frac{prec_p^c}{1 - prec_p^c} c fpr \quad (4)$$

$$tpr = \frac{1 - prec_n^c}{prec_n^c} c fpr + 1 - \frac{1 - prec_n^c}{prec_n^c} c \quad (5)$$

The isometrics for positive precision are lines that go through $(0, 0)$. Isometrics for negative precision go through point $(1, 1)$. Varying the precision value results in a line with different slope. The higher the precision value, the closer the isometric

³Implicitly, an ROC curve also shows the trade-off between true negative rate tnr and false negative rate fnr since $tnr = 1 - fpr$ and $fnr = 1 - tpr$.

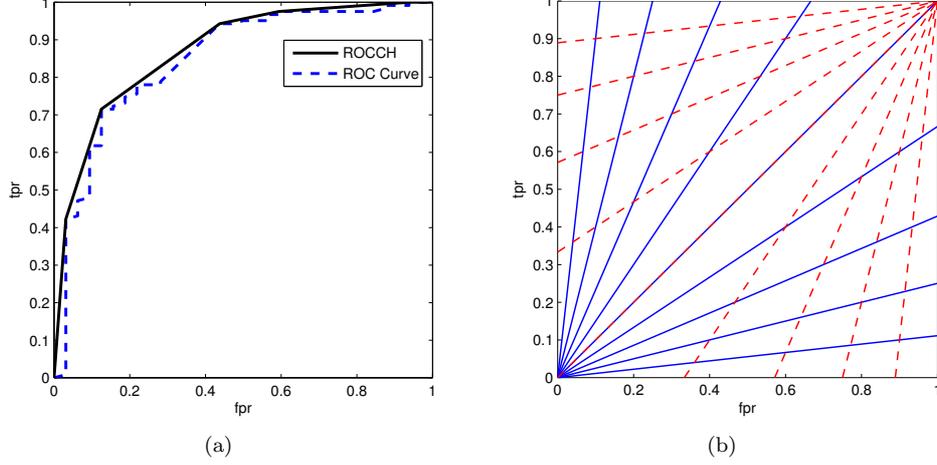


Figure 1: Constructions in ROC space: (a) an ROC curve with corresponding ROCCH, and (b) $prec_p^c$ -isometrics (solid lines) and $prec_n^c$ -isometrics (dashed lines).

approaches point $(0, 1)$. Figure 1(b) shows $prec_p^c$ -isometrics and $prec_n^c$ -isometrics. The precision value is varied from 0.1 to 0.9 in steps of 0.1. In the next Subsection 3.2 we show how these lines can be used to construct a reliable classifier, i.e., a classifier with guaranteed reliability.

3.2 Design Framework

Given a ROCCH and a desired positive precision and negative precision, a positive precision isometric and a negative precision isometric are constructed. The intersection point of the $prec_p^c$ -isometric and the ROCCH is denoted as (fpr_a, tpr_a) . By definition, this intersection point represents a classifier with the $prec_p^c$ used to construct the isometric. In case of multiple intersection points, we choose the intersection point with the highest tpr value. Similarly, the intersection point (fpr_b, tpr_b) of the $prec_n^c$ -isometric and the ROCCH represents a classifier with negative precision defined by that isometric. In case of multiple intersection points, we choose the intersection point with the lowest fpr value. If either $(fpr_a, tpr_a) = (0, 0)$ or $(fpr_b, tpr_b) = (1, 1)$, then the desired classifier cannot be constructed. Otherwise, we can distinguish the following three cases as shown in Figure 2:

- **Case 1:** the isometrics intersect on the ROCCH
The classifier corresponding to this point has by definition the precisions defined by both isometrics. Therefore, this classifier has the desired reliability.
- **Case 2:** the isometrics intersect below the ROCCH
This classifier also has the desired reliability. However, the classifiers corresponding to any point on the ROCCH between (fpr_b, tpr_b) and (fpr_a, tpr_a) have higher reliability.

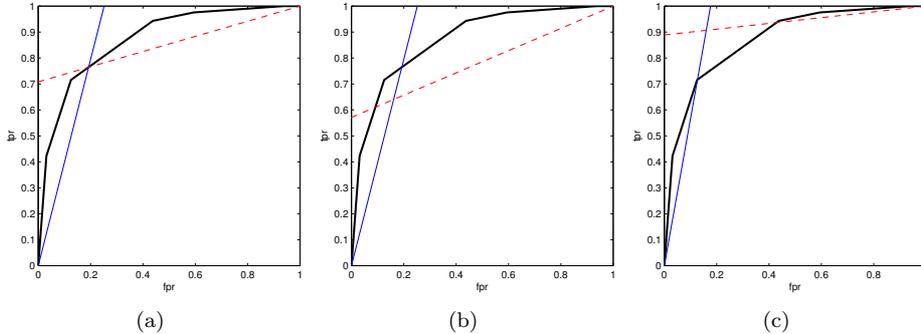


Figure 2: Location of the intersection point between a positive precision isometric and a negative precision isometric: (a) Case 1, (b) Case 2, and (c) Case 3.

- **Case 3:** the isometrics intersect above the ROCCH

There is no classifier with the desired reliability. Therefore, a number of ambiguous instances needs to be left unclassified. We define thresholds a and b of the abstention mechanism (see Section 2) to correspond with points (fpr_a, tpr_a) and (fpr_b, tpr_b) , respectively. It was shown that the reliable classifier with these thresholds has the desired reliability [8]. The construction of this classifier can be visualized by removing the instances in the ROCCH between points (fpr_a, tpr_a) and (fpr_b, tpr_b) . The removed instances are those left unclassified.

From the three cases we can conclude that a reliable classifier satisfies the first requirement, i.e., it is able to guarantee a desired level of reliability. In the next Subsection 3.3 we show that the second requirement is also satisfied.

3.3 Changing error costs

Having established a framework for designing classifiers with a desired reliability, we now extend it to deal with changing error costs (the second requirement). We define $c(p, n)$ and $c(n, p)$ as the costs of a false positive and false negative, respectively. For example, if the task is to classify a person as guilty (class p) or non-guilty (class n), then $c(p, n)$ represents the cost of classifying a non-guilty person as guilty. In a similar way the error cost $c(n, p)$ can be defined. Clearly, both costs are extremely high from both societal and legal perspective.

Error costs are incorporated in the precision metrics and corresponding isometrics by setting $c = \frac{c(p, n) N}{c(n, p) P}$. Thus, changes in the error-cost distribution are incorporated by adjusting the class distribution. The effect is a change in the slope of the isometrics.⁴ Therefore, the design framework of the previous Subsection 3.2 remains valid.

⁴For a detailed discussion see [2, 8].

4 Application to Law Enforcement

We apply a reliable classifier to solve the task of detecting credit card abuse in a timely fashion. Credit card transaction can be classified as fraudulent (class p) or non-fraudulent (class n). The score $l(x)$ of transaction x can be seen as a suspicion score: the higher the score, the more unusual the transaction. The error cost of a false positive includes a waste of time and money. Moreover, since the card holder of a possibly fraudulent transaction is contacted by the agency, the false positive cost also includes a privacy violation. The error cost of a false negative includes non-detection of criminal activities.

Credit card fraud detection is gaining popularity in law enforcement because the number of fraudulent transactions is increasing significantly and a high level of (organized) crime activity is involved [1]. For these reasons, law enforcement agencies are prepared (and forced) to cope with the problem. However, the limited number of human resources implies that only a very small subset of the suspicious transactions can be investigated. A reliable classifier can be used to find a small number of instances with very high reliability. Thus, the used resources and obtained benefits are optimized.

To solve the outlined task, we reproduced the experiment in [6] that resulted in the best overall classifier (Bayesian belief network).⁵ The minimum reliability for the classifier to be applied is defined by $prec_p^c = 0.9$ and $prec_n^c = 0.95$. The ROCCH of the network is shown in Figure 3(a). From the location of the intersection point of the isometrics can be seen that abstention is needed to obtain the desired reliability. The point (fpr'_a, tpr'_a) that represents the resulting reliable classifier is marked by a solid disc. The classifier abstains from classifying 2531 instances. The output of the classifier for the remaining 2469 instances are considered as reliable.

If we remove the unclassified instances in the ROCCH, then we obtain a new ROCCH, i.e., ROCCH2 shown in Figure 3(b). This ROCCH has a larger area under the ROC curve (AUC), more specifically 0.96 in contrast to 0.88 for the original ROCCH. Although ROCCH2 is constructed with only a subset of the data of ROCCH1, the increase in AUC shows that the approach successfully identified instances that would be classified incorrectly.

5 Discussion

The law enforcement domain is changing constantly and it has a large impact on the politics of nations worldwide. Moreover, decisions and corresponding actions need to be correct since it concerns security and safety for civilians. For these reasons, it is difficult to apply machine-learning classifiers. Moreover, in the broader domain of legal practice, most of the intelligent applications concern legal text retrieval systems or expert systems for administrative tasks. It has been shown that even these systems make a substantial number of errors in their decisions [4].

⁵An instance (transaction) is described by ten features and its class. The high error costs are assumed to be equal and the skew ratio is $c = \frac{3405}{1595}$ (the class distribution of the training set).

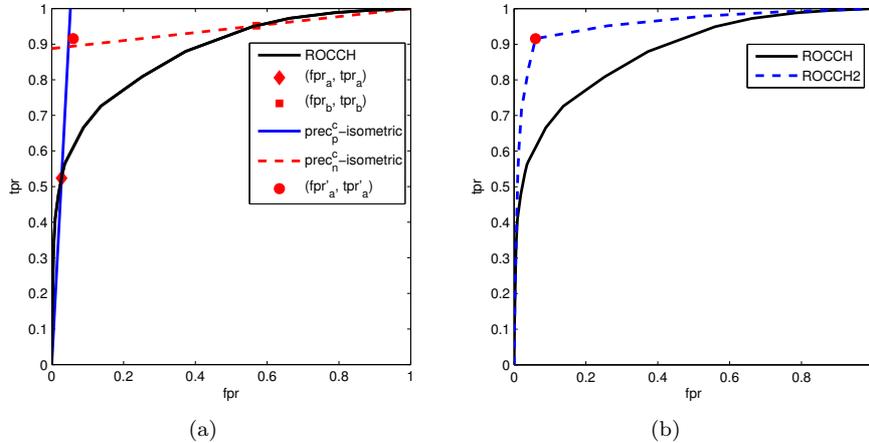


Figure 3: Fraud detection with a Bayesian belief network: (a) ROCCH with two isometrics defined for $c = \frac{3405}{1595}$, and $prec_p^c = 0.9$ and $prec_n^c = 0.95$, respectively. The disc symbol represents the reliable classifier, and (b) ROCCH2 is obtained by removing instances with uncertain classification.

Therefore, one cannot guarantee legally correct decisions and a correct treatment of civilians.

We argue that reliable classifiers are an appropriate solution to solve law enforcement tasks. Due to the massive amount of available data, reliable classifiers can improve the quality and reliability of information and support a more effective and efficient execution of law enforcement than was previously possible.

Our real-life application illustrated that reliable classifiers can be successfully used to cope with credit card fraud. In addition, the approach can be readily applied to other types of fraud detection with the same efficiency. We mention two examples: the detection of money laundering in an attempt to stop funds received by terrorist networks, and detecting computer intrusions to secure confidential data. In general, any application domain with high error costs benefits from reliable classifiers.

6 Conclusion

In this paper we focused on classification problems in law enforcement. We used an approach based on ROC isometrics to construct reliable classifiers. We have seen that this approach is generally applicable to construct a classifier with a predefined reliability. In the present context, reliability was measured by precision on each class but other performance metrics could be used as well, e.g., the F -measure. Furthermore, the approach has the ability to adjust its reliability in response to a changing error-cost distribution. Finally, we argued why reliable classifiers are desired in law enforcement. In conclusion, we state that reliable

classification significantly improves the viability of applying machine learning to the law enforcement domain.

Acknowledgments

The first author is sponsored by the Dutch Organization for Scientific Research (NWO), grant nr: 634.000.435. The third author is sponsored by the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

References

- [1] Richard Bolton and David Hand. Statistical fraud detection: a review. *Statistical science*, 17(3):233–255, 2002.
- [2] Peter Flach. The geometry of ROC space: Understanding machine learning metrics through ROC isometrics. In *Proceedings of the 20th International Conference on Machine Learning (ICML-2003)*, pages 194–201, Washington, USA, August 21-24 2003.
- [3] Peter Flach and Shaomin Wu. Repairing concavities in ROC curves. In *Proceedings of the 2003 UK Workshop on Computational Intelligence (UKCI 2003)*, pages 38–44, Bristol, UK, September 1-3 2003.
- [4] Marga Groothuis and Jörgen Svensson. Expert system support and juridical quality. In *Proceedings of the 13th Foundation for Legal Knowledge Based Systems Conference (JURIX 2000)*, pages 1–11, Enschede, the Netherlands, December 14-15 2000.
- [5] Paul De Hert, Wim Huisman, and Thijs Vis. Intelligence led policing ontleed. *Tijdschrift voor Criminologie*, 47(4):365–376, 2005. (in Dutch).
- [6] Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. Credit card fraud detection using Bayesian and neural networks. In *Proceedings of the 1st International NAISO Congress on Neuro Fuzzy Technologies (NF 2002)*, Havana, Cuba, January 16-19 2002.
- [7] Foster Provost and Tom Fawcett. Robust classification for imprecise environments. *Machine Learning*, 42(3):203–231, 2001.
- [8] Stijn Vanderlooy, Ida Sprinkhuizen-Kuyper, and Evgueni Smirnov. An analysis of reliable classifiers through ROC isometrics. In *Proceedings of the ICML 2006 Workshop on ROC Analysis (ROCML 2006)*, pages 55–62, Pittsburgh, USA, June 29 2006.
- [9] Stijn Vanderlooy, Ida Sprinkhuizen-Kuyper, and Evgueni Smirnov. Reliable classifiers in ROC space. In *Proceedings of the 15th Annual Machine Learning Conference of Belgium and the Netherlands (BENELEARN 2006)*, pages 113–120, Ghent, Belgium, May 11-22 2006.

Single-Player Games: Introduction to a New Solving Method

Combining State-Space Modelling with a Multi-Agent Representation

François Van Lishout

Pascal Gribomont

University of Liège, Montefiore Institute, B28, 4000 Liège, Belgium
{vanlishout, gribomont}@montefiore.ulg.ac.be

Abstract

Classically, games are described as search problems. Each game situation is considered as a node in a graph. The arcs represent legal moves from one position to another. Solving a single-player game requires to solve a state-space problem, i.e. find one path that leads to a solution-state of the game. The heart of this paper consists in exploring the idea that most single-player games can also be modelled as multi-agent systems. The agents are no longer the players of the game as for multi-player games, but primitive game elements depending on the particular game. Furthermore, instead of facing each other, the agents collaborate to achieve a common objective. This new representation leads to new interesting resolution techniques, even more when both modelling methods are combined. It is demonstrated on the game of Sokoban, a challenging one-player puzzle for which mankind still dominates the machine.

1 Introduction

Games have always fascinated mankind and attracted the attention of the AI research community. Writing game-playing programs is not just a diverting activity, it also has many applications in real-life problems. Some people consider even life as a big game, where each living creature tries to maximize its well-being.

In many games, the machine has become stronger than the best human players. Machines have already beaten the human World Champion in famous games like Checkers, Chess, Scrabble and Othello. However, mankind has not been humbled by chips in all games. The best human players are still stronger than computers in games like Go, Poker, Chinese Chess and Hex [8].

This paper introduces a new solving method for single-player games, which will be demonstrated on the game of Sokoban. We have chosen this game for three reasons. First, it is a one-player puzzle which has not been solved yet by the AI community. Furthermore, man still dominates machine in this domain. For now, the best documented solver called *Rolling Stone* uses single-agent search techniques with a lot of problem-dependent improvements, and is able to solve 59 problems of a difficult 90-problem test suite [4].

Second, an unlimited set of different starting positions can be created by varying the size and the difficulty of the component problems. Different solving methods are therefore easy to compare, as there will always exist a test-set that can highlight the limits of the solving strategies. In this paper, we will compare our results to *Rolling Stone*'s ones on the same 90-problem benchmark.

Finally, Sokoban is a very challenging game. The main reason for this is the size of the search-space: it has been estimated at 10^{98} for 20×20 Sokoban mazes [3]. Other reasons are the large branching factor, the presence of deadlock states, the fact that the problem is PSPACE-complete, the length of the solutions, etc. For more details about the properties that make Sokoban a challenging research domain, consult [1, 2, 3]. Let us just add that Sokoban is not only challenging, it is also of practical interest as an instance of robot motion planning problems [2].

This is still an ongoing research, and further problem-dependent enhancements in the implementation are planned. On the other hand, problems that are unsolvable by *Rolling Stone* with single-player methods without other enhancement become solvable with the new strategy presented here.

2 The Game of Sokoban

A Sokoban maze is a grid composed of unmovable walls, free squares, exactly one man, and as many stones as goal squares. The player controls the man and the man can only push stones (not pull). Furthermore, only one stone can be pushed at a time. The objective of the game is to push all stones on goal squares.

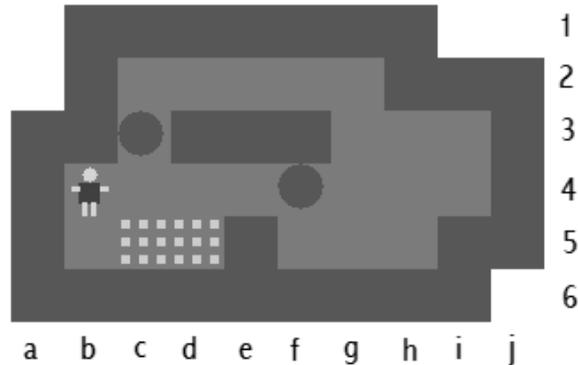


Figure 1: A Sokoban maze and a particular solution: b4-c4-d4-e4-f4-g4-g3-g2-f2-e2-d2-c2-c3-c4-b4-b5-c5-c4-d4-e4-f4-g4-g3-h3-i3-i4-h4-g4-f4-e4-d4-e4-f4-g4-g3-g2-f2-e2-d2-c2-c3-c4. The corresponding stone-moves notation of the solution: f4-g4-h4, c3-c4-c5-d5, h4-g4-f4-e4-d4-c4-c5.

For a better understanding of the game, a solution to the problem of Figure 1 is given. It is first described as an ordered list of coordinates of the squares that the man must pass by, to bring all the stones on goal areas (starting with the initial square of the man). A shorter notation giving only the lists of the stone-moves is also provided. Note that this notation makes the implicit hypothesis that each stone-move of the solution is valid, i.e. that the man can reach the square adjacent to the stone to effectively make the push.

State-of-the-art solvers model Sokoban as a state-space problem. The states of the graph are all the possible states of the game which can be obtained by varying the position of the man and the stones. The arcs represent legal one-square stone-moves from a position to another. *Rolling Stone* uses the Iterative Deepening A* (IDA*) [5] as basis for exploring the state-space.

In [4], the fact that only trivial problems can be solved by using the IDA* without other enhancements is demonstrated (even with a clever heuristic function, no maze of the difficult 90-problem test suite used can be solved). A lot of enhancements like transposition table, move ordering, deadlock table, pattern search, and further problem-dependent improvements were implemented to achieve good performances (59 problems solved). However, this strategy seems to have reached its limits as the most difficult instances are still far from being solved by such methods.

3 The New Multi-Agent Modelling Approach

The idea of the new multi-agent modelling approach is that every stone of the maze can be seen as an agent whose aim is to reach one of the goal squares, and the global goal is to find a solution for which everyone achieves his objective. It is also possible to impose some kind of optimality like minimizing the global number of agent moves. In this view of the problem, the man is only a puppet which can be called by the stones when they want to be pushed.

The classical state-space techniques can be reused in this approach, as tools which can be used by agents to solve sub-problems of the game. Agents can for example use the A* algorithm [6, 7] to determine if they can reach some of the goal squares without needing the collaboration of the other agents. However, an agent that is solvable cannot start moving without the permission of the other agents. Otherwise, it could reach a goal square but generate a configuration in which other agents are in deadlock states. The difficulty of the problem is therefore reported to the communication process between agents.

3.1 Solving a Particular Subclass of Sokoban Problems

3.1.1 Definition of the Subclass

Let us define a particular subclass of Sokoban mazes which have been completely solved by the new method with a rather simple protocol. To be more precise, we will write a protocol that solves mazes if (but not only if) they belong to the class. We can thus apply this protocol to any Sokoban maze but we have the assurance to find a solution only if it is in the class. Intuitively, a maze is in this class if the stones are solvable one by one. To be more precise, the maze must satisfy the following conditions:

- *Goal-ordering-criterium*: it must be possible to determine in advance the order in which the goal squares will be filled without introducing deadlocks, independently from the position of the stones and the man.
- *Solvable-stone-existence*: it must be possible to bring at least one stone to the first selected goal square without having to move other stones.
- *Recursive-condition*: for each stone which satisfies the previous condition, the maze obtained by removing that stone and replacing the selected goal square by a wall must also be in the class.

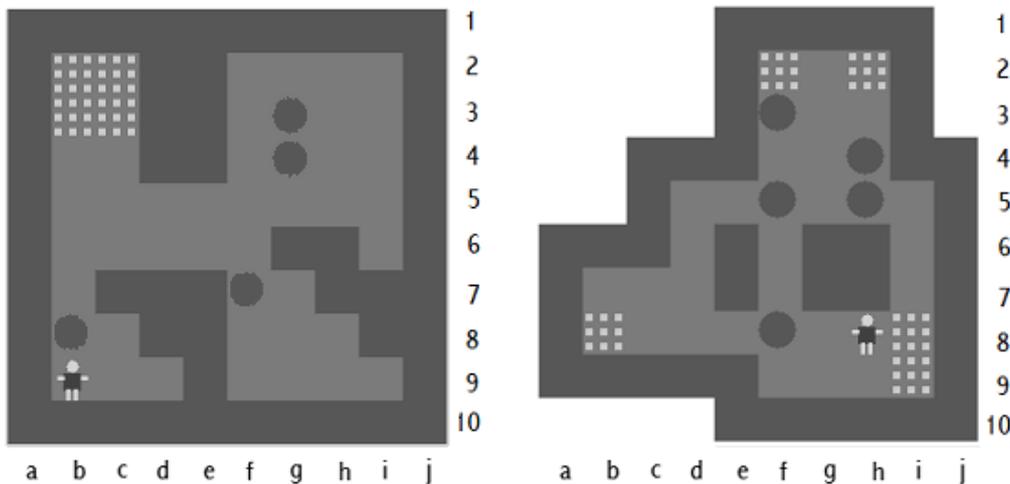


Figure 2: The Sokoban maze on the left side is an example of maze that is in the defined subclass. The one on the right side is a counterexample.

The maze on the left side of Figure 2 satisfies all the conditions:

- Filling the goal squares in the following order is adequate: b_2 , c_2 , b_3 and finally c_3 .
- The b_8 stone is the only one that can reach the square b_2 without requiring other stone-moves.
- The maze obtained by removing the stone b_8 and replacing the square b_2 by a wall is obviously also in the class. This comes from the fact that each remaining stone can reach each remaining goal. The order in which the stones will be solved can thus be chosen freely.

The maze on the right side of Figure 2 is not in the class. It is a good compilation of non-satisfied conditions. The goal squares are divided into several goal areas and it is not possible to determine in advance the order in which the goal squares should be filled without considering the position of the stones and the man. Furthermore, in the initial position, only one stone can reach some goals without other stone-moves (the stone f_8 can reach b_8 and i_8). However, removing the stone and replacing the square i_8 by a wall would make the goal square i_9 unreachable. Even by replacing the square b_8 by a wall, the corresponding maze would still not be in the class. Indeed, none of the four remaining stones can reach a goal without other stone moves.

3.1.2 Protocol for Solving the Mazes of the Subclass

Due to the *recursive-condition* of the defined subclass, solving an agent cannot introduce a deadlock situation. This facilitates the elaboration of the algorithm. Indeed, we do not have to write a communication protocol that an agent must initiate when he can reach the first goal square. The agent can in fact not cause damage to other agents. From this, a simple sequential program can be used for finding a non-optimal solution:

1. If no agent (stone) exists, the maze is already solved and the solution path is the empty list.
2. Else, select the first goal square.
3. Select an agent that was not chosen before and use the A* to compute if it can reach the selected goal square.
 - If it cannot, return to step 3.
 - If it can, move the agent to the goal square, replace the goal square reached by the agent by a wall and delete the agent. Go back to step 1. The solution path is the solution path of the deleted agent appended to the recursive solution path produced by the result of step 1.

In the first version of *Talking Stones* presented here, a very simple rule for ordering the goal squares has been chosen. The first goal square will be the one with the maximum adjacent walls (in case of a tie any of them). The next goal squares are chosen by following the same strategy, but by considering that the previously chosen goal squares are walls. Thus, the goal squares which can only be reached from a single adjacent square will be filled first and we will avoid to block their entrance. This simple criterion has been chosen because it is easily implemented and works for most of the mazes that satisfy the *goal-ordering-criterion*. However, counterexamples can be constructed and more complicated ordering rules will be considered in later versions.

This protocol is easy to modify for obtaining shorter solutions at the cost of more computation time. In step 3, the first solvable agent will no longer be chosen directly. Instead, the solvability of all the agents will be computed and the one that has found the shortest path to the goal square be selected. By this means, a situation in which a stone has to make a detour around another stone to reach a goal square cannot occur anymore. The stone that would have been on the way would have already been solved. This strategy does not guarantee optimality (counterexamples can be constructed easily) but practice shows that in most real cases it leads to optimal or near-optimal solutions.

The termination of the algorithm is assured for mazes that are in the subclass, as the *solvable-stone-existence* condition assures that it will always exist at least one agent that can reach the first goal square at step 3. The mazes of the subclass can thus all be solved by this protocol.

The algorithm could also find an answer for some mazes that are not in the class. Consider for example a maze for which the stones are solvable one-by-one but only in a precise order. The algorithm could try this order first by chance and find the solution.

It is easy to modify the protocol so that it terminates also for mazes that are not solvable by this protocol. We can simply answer that no such easy multi-agent exists when no more agents that have not been selected before can be chosen at step 3.

3.2 Results

Only one problem of our 90-problem test suite is in the subclass just defined (problem 78) and can be solved by this protocol. This is not surprising, as problems that are directly solvable stone by stone are uncommon in difficult benchmarks. Instead of elaborating other protocols for solving more general problems with the pure new multi-agent modelling approach, another idea has been developed.

As it is often the case in AI, trying to understand how the human player solves problems helps to find new algorithms. In the case of Sokoban, one of the talents of the human player consists in recognizing very soon in the resolution process that he can reach a configuration that is easy to solve (stone-by-stone). This suggests a new solving method for difficult games.

4 Embedding the Multi-Agent Modelling Approach into a Classical State-Space Algorithm

The method consists in using a classical state-space algorithm, but one in which the nodes whose corresponding state of the game is solvable by the new multi-agent modelling approach are defined as success nodes. When the search reaches such a node, the search terminates successfully. The solution is then obtained by appending the solution path found by the state-space algorithm to the solution found by the multi-agent modelling approach.

The offspring of success nodes are no longer reachable and can be considered to have been pruned out of the state-space. In practice, the size of the state-space will decrease substantially. On the other hand, more computation time is needed at each node as the multi-agent modelling approach is called for each node to determine whether it is a success node. However, the time lost by these calls is largely compensated by the time won by having less nodes to visit. Our program *Talking Stones* implements this idea.

4.1 Choosing the Right State-Space Algorithm

The size of the search space remains huge for difficult problems. As the branching factor is rather high, a good memory management is required. Therefore, an iterative deepening algorithm is a good choice. We have thus to choose between the pure iterative deepening algorithm and the IDA* algorithm. The latter is naturally the best choice if we can find a good heuristic.

However, the heuristics commonly used for games are functions that try to minimize a particular distance to the objective. For Sokoban, it is generally the distance between the stones and the goals (using a minimum matching algorithm to assign the stones to the goals). This does not help much in the context of the new method. Indeed, almost every Sokoban mazes start in a configuration where the situation seems to be nearly blocked. The initial strategy consists therefore in finding a few moves to make more space for the man. Those moves have no reasons to be moves that diminish the global stone-goal distance. Defining a good heuristic function is thus far from trivial. For this reason, we have simply used the original iterative deepening algorithm in our first version of *Talking Stones*. We decided to limit the number of generated nodes to 20000.

4.2 Results

The algorithm seems rather naive; it simply tests all the possible moves until a position in which the stones are solvable one-by-one is reached. Surprisingly, this coarse strategy is already able to solve 9 problems of the 90-problem test suite.

Problem	Generated Nodes	Depth of the multi-agent solution	Time
1	76	3	6 sec
2	7155	4	8 min 23 sec
3	31	2	5 sec
5	67	2	25 sec
6	2849	4	4 min 20 sec
51	972	4	31 sec
54	2761	3	19 min 4 sec
78	0	0	< 1 sec
82	173	3	12 sec

Table 1: Results obtained by Talking Stones on the 90-problem benchmark

4.3 Comparison with *Rolling Stone*

Rolling Stone has demonstrated that a pure IDA* approach without other enhancements cannot solve any problem of the benchmark and that immense progress can be obtained with them. Implementing some of those enhancements within *Talking Stones* looks thus promising for solving more instances.

One enhancement needs a special comment: Move Ordering. In *Rolling Stone*, moves which preserves the *inertia* of the stones are favored. This means that if the previous move was performed on a particular stone, moving this stone again will be considered first. From this, if *Rolling Stone* reaches a node that is solvable by the multi-agent modelling approach, it will find a solution similar to the one found by *Talking Stones* (solving the stones one-by-one, but possibly in another order).

However, the big difference is that our program will in this case require much less time and space to find the solution. Indeed, if we consider a maze composed of N stones, *Talking Stones* will simply call the A* algorithm at most N times to find a first solvable stone, $N - 1$ times for a second one, and so on. This gives a complexity of N^2 calls to the A* algorithm. Each call is relatively fast as the size of a Sokoban maze is small and so is the number of possibilities. To be more precise, for a 20×20 maze the stone can be placed on at most 18×18 squares (the borders are walls). A stone can move in at most 4 directions. The number of possibilities is thus clearly bounded by $18 \times 18 \times 4 = 1296$. A smaller bound can even be found by considering that a stone cannot move if it is in a corner and can at most move in 2 directions if it is against a wall.

An implementation based on the IDA* will do much more. At each step, it will generate all the possible moves for the N stones. For each position that can be obtained by performing one of those moves, it will compute its heuristic distance to the final goal. Then, it will select the most promising move not tried so far (in case of a tie one that preserves the inertia) and continue from that point. As the branching factor of the game of Sokoban is important, the IDA* algorithm will rapidly have to keep a large amount of candidate nodes to expand in memory. In the case of a maze solvable stone-by-stone, this is not needed as most of the candidate moves will never be tried.

5 Generalization of the Method to Other Games

We can decompose the new solving method for difficult single-player games in three layers:

- The high-level layer is a classical state-space algorithm where a node is a success node if the medium layer can solve it. The choice of the algorithm depends on the precise game. If the branching factor is important an iterative deepening algorithm is appropriate. The IDA* should be preferred to the pure iterative deepening algorithm when expert-knowledge of the game is provided and a good heuristic function can be constructed. If the branching factor is small, the memory is not critical and the A* algorithm is indicated. When no good heuristic function can be constructed, the breadth-first search algorithm is the best suited one at this level. See [9] for more information on the most adapted state-space algorithms according to the branching factor.
- The medium-level is a protocol based on a multi-agent representation of the game. The agents are primitive game elements depending on the particular game. The agents have to communicate together to find a common solution. They can use the algorithms of the low-level as tools for solving sub-problems.
- The low-level is a set of algorithms for solving subproblems of the game. Classical state-space algorithms can be used but not exclusively.

We believe that it is possible for almost all games to determine primitive game elements that have to reach some goal. In puzzles like the 24-tile puzzle, the agents could be defined as the tiles. In this representation, each tile aims to reach its final destination but cannot move without altering the position of other agents. In the game of Sokoban, all the agents are instances of stones of the maze and have thus the same characteristics. For other games however, we could define agents that have their own personality. For the game of solitaire for example, the agents could be the 52 cards. Each agent is now unique. Note that for such imperfect information game, we must consider that only a subset of the agents is visible. The other agents can thus be seen as being in an unknown queue, waiting for entering into play. The new multi-agent modelling method could also be extended to multi-player games. In this case, teams of agents would face each other. Take the game of Go as an example. Here the goal of the white and black teams of stones would be to control as many areas of the game as possible.

6 Conclusion and Future Work

In this work we have presented a new modelling method for single-player games. Our idea has been to model the game as a multi-agent system where the agents are primitive game elements depending on the particular game. We have demonstrated the method on the game of Sokoban. It is important to note that the multi-agent notion which has been introduced is only conceptual and that it does not imply multi-agent programming. Our program is a solver for a single player puzzle. The method presented here requests a central solver to decide the order of the stones to be solved. This work presents thus solely a new way to view the problem which leads to new interesting resolution ideas.

We have defined a particular subclass of Sokoban mazes that has been completely solved by a protocol based on our pure multi-agent representation. It is of course possible to write an algorithm which determines efficiently if a maze is in the class. This has not been done here since it was not necessary. Our protocol solves indeed all the mazes of the subclass but can also solve some out of it. Furthermore, we have discovered that even when a Sokoban problem is not in this very particular class, it can often become so after a few moves.

We propose thus a new method. It consists in using a classical state-space algorithm, but one in which the nodes whose corresponding state of the game are solvable by the multi-agent modelling approach are defined as success nodes. This means that when the search reaches such a node, it terminates successfully. The solution is then obtained by appending the solution path found by the state-space algorithm to the solution found by the multi-agent modelling approach. Thus, we have not to test if a maze is in the class, but merely search for a configuration that is solvable by an algorithm based on a multi-agent representation.

At the time being, our program *Talking Stones* solves only 9 mazes of the benchmark whereas *Rolling Stone* solves 59. However, the latter is based on the IDA* algorithm with a lot of really interesting problem-dependent enhancements. These are presented in [4] and it is well explained why each of them contribute to a substantial decrease of the search-tree size. On the other hand, the fact that no problem of the benchmark can be solved with the pure IDA* approach without these enhancements, even with a clever heuristic, is also demonstrated in [4]. We have not implemented these enhancements yet and we plan to inject them within our new method in future works. As *Rolling stone* has enjoyed tremendous progress by adding them to its initial pure IDA* approach (from 0 mazes solved to 59), we hope to benefit from the same kind of progression.

References

- [1] J. Culberson. Sokoban is pspace-complete. In *Informatics 4 Fun with Algorithms*, pages 65–76, Waterloo, 1999. Carleton Scientific.
- [2] D. Dor and U. Zwick. Sokoban and other motion planning problems. *Computational Geometry: Theorie and Applications*, 13(4):215–228, 1999.
- [3] A. Junghanns. *Pushing the limits: New developments in single-agent search*. PhD thesis, University of Alberta, 1999.
- [4] A. Junghanns and J. Schaeffer. Sokoban: Enhancing general single-agent search methods using domain knowledge. *Artificial Intelligence*, 129(2):219–251, 2001. Republished in [8].
- [5] R.E. Korf. Depth-first iterative-deepening: An optimal admissible tree search. *Artificial Intelligence*, 26(1):35–77, 1985.
- [6] N.J. Nilsson P.E. Hart and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.
- [7] N.J. Nilsson P.E. Hart and B. Raphael. Correction to [6]. *SIGART Newsletter*, 37:28–29, 1972.
- [8] J. Schaeffer and J. van den Herik, editors. *Chips Challenging Champions*. Elsevier, 2002.
- [9] P.H. Winston. *Artificial Intelligence*. Addison-Wesley, third edition, 1992.

PROGNOSTIC BAYESIAN NETWORKS: DEFINITION, CONSTRUCTION, AND USE

Marion Verduijn ^{a,e} Niels Peek ^a Peter Rosseel ^d Evert de Jonge ^b
Bas de Mol ^{c,e}

^a *Dept. of Medical Informatics*, ^b *Dept. of Intensive Care Medicine*, ^c *Dept. of Cardio-thoracic Surgery, Academic Medical Center – University of Amsterdam*
^d *Dept. of Anesthesia and Intensive Care, Amphia Hospital, Breda*
^e *Dept. of Biomedical Engineering, University of Technology, Eindhoven*

Abstract

Prognostic models are tools to predict the future outcome of disease and disease treatment, one of the fundamental tasks in clinical medicine. These models are traditionally built with supervised machine learning techniques, and then consider prognosis as a static, one-shot activity. This paper presents a new type of prognostic model that builds on the Bayesian network methodology, and that implements a dynamic, process-oriented view on prognosis. In contrast to traditional prognostic models, prognostic Bayesian networks explicate the scenarios that lead to disease outcomes, and can be used to update predictions when new information becomes available. The paper presents an algorithm for inducing prognostic Bayesian networks from medical datasets, and describes the prognostic reasoning tasks of these models in clinical practice. Prognostic Bayesian networks and the prognostic tasks are illustrated with an extensive application in the field of cardiac surgery.

1 Introduction

Prognostic models have become important instruments in medicine. Given a set of patient specific parameters, they predict the future occurrence of a medical event or outcome. Example events are the occurrence of specific diseases (e.g., cardiovascular diseases and cancer) and death. These models are used for prediction purposes at levels that range from individual patients (where their predictions help doctors and patients to make treatment choices) to patient groups (where they support health-care managers in planning and allocating resources) and patient populations (where they provide for case-mix adjustment) [1, 19]. They provide as such useful information for the clinical staff and management that are involved in health care processes to identify high risk patients and to evaluate the quality of provided health care.

Building a prognostic model is generally taken as a supervised learning problem in which the relationship between predictive variables and the outcome variable is described. In this standard modeling approach, prediction is assumed to be a one-shot activity at a single, predefined point in time. The resulting model does not reflect that the variables come from a process and may be measured at different time points or in different phases. The model cannot be used to update one's prognostic expectations during the process, when new information has become available. Furthermore, most models predict a single outcome and ignore the course of the process that precedes that outcome.

This paper introduces a new type of prognostic model, based on the Bayesian network methodology, that extends the functionality of traditional prognostic models. Since their introduction in the 1980s, a large number of Bayesian networks have been developed for medical diagnosis and therapy selection, e.g., [3, 16, 17]. Prognostic applications of Bayesian networks have however been rare [14]. The *prognostic Bayesian network* (PBN) that is introduced here, is used to provide a structured representation of a health care process. The key part of this health care process is a medical intervention (e.g., surgery) that is generally preceded by a phase of pre-assessment and

followed by a phase of care for recovery. The mutual relationships between variables that come into play in these phases and the outcome are modeled as Bayesian network. As such, it makes explicit how variables from early phases of the care process are related to an outcome variable through a process of intermediate variables.

Although Bayesian networks provide a powerful and general methodology for knowledge representation and reasoning, we will argue that the task of medical prognosis adds specific requirements. First, general algorithms for learning Bayesian networks from data are not prepared to take into account, and exploit, the structure of health care processes where prognosis is important. Second, the “raw” results of probabilistic inferences on a Bayesian network are not meaningful to clinicians, and require post-processing. We have developed these extensions, in the form of (i) a dedicated algorithm for inducing PBNs from clinical databases, and (ii) the definition of a set of structured clinical tasks where PBNs are useful.

The remainder of the paper is structured as follows. In Section 2 we elaborate on the proposed type of model and architecture. Section 3 describes and evaluates our algorithm for inducing PBNs from data. The clinical tasks of PBNs are described in Section 4. Section 5, finally, provides a general discussion of the work presented. Throughout the paper, we illustrate our ideas with an application in the field of cardiac surgery.

2 Prognostic Bayesian Networks

Medical prognostic models describe the relationship between patient characteristics and outcome variables. The standard methodology to obtain an objective description of this relationship is to build a predictive model from a set of observed patient data and outcomes, using supervised learning techniques. The predominant technology in the medical domain is (generalized) linear regression analysis, but also other techniques, such as tree models and neural networks are applied.

The general supervised learning paradigm requires that a model can only be learned, and used, for a single time point of prediction. A common example of such time points is the moment of decision making for a specific therapy. All patient data that are available at this time point are then taken into account for model development; these can be data that are measured at the time of prediction, but also data from the clinical history of the patient. Subsequently, variables that are found to have predictive value for the outcome are selected for inclusion in the model (feature subset selection), and the relation between these predictors and the outcome variable is induced from the data.

This approach of predictive modeling has in our view some shortcomings, as a result of which the traditional model has limited functionality. First, medical predictions are usually needed at multiple time points during a care process, each time taking more (new) information into account. Furthermore, variables that are observed beyond the predefined prediction time are not included in the model; therefore we cannot express that predictor variables are related to the outcome variable through a process of intermediate variables. Finally, when a strong correlation exists between two predictive variables, only one of them is usually included in the model, while the other variable is left out; this is counter-intuitive for domain experts who would always name both variables as important prognostic markers.

PBNs solve these problems by (i) explicitly modeling the entire care process in a structured, graphical fashion, where the directions of arcs represent the flow of time, (ii) including all variables with predictive value in the Bayesian network. As a result, a comprehensive domain model is provided and outcome predicting at multiple time points and with different types of information is allowed by exploiting Bayesian reasoning methods. We will describe and illustrate some prognostic tasks of PBNs in Section 4.

As an example, we present a PBN that we developed for the domain of cardiac surgery, shown in Figure 1. This network distinguishes three subsequent phases of care, and has the variable hospital mortality (`hospmort`) as final outcome. The network includes 8 variables from the preoperative phases of pre-assessment, 3 operative variables, and 12 physiological and complication variables from the postoperative phases of recovery. The network was learned from data of 10,114 patients who underwent cardiac surgery in the Amphia Hospital in Breda, the Netherlands; 277 of these patients died during hospitalization (2.74%). The learning algorithm that was used for inducing this network from these data will be described in the next section.

In Figure 1, the network is applied to the case of a 62-year-old non-diabetic patient who will

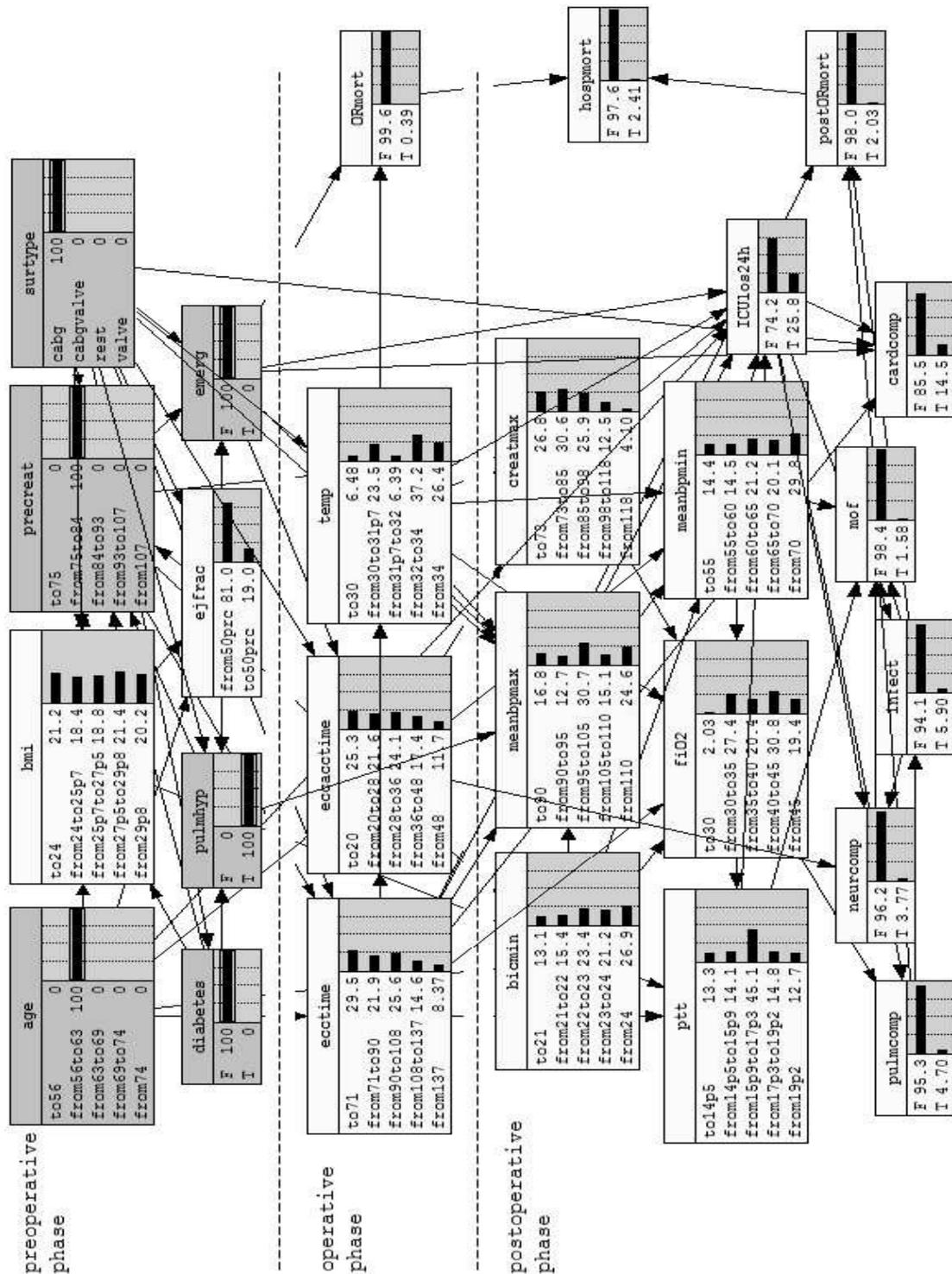


Figure 1: A PBN for cardiac surgery. The model distinguishes three phases of care: a preoperative, operative, and postoperative phase. The model has here been applied to a 62-year-old patient who will undergo a non-emergent coronary artery bypass grafting (CABG) operation. The estimated risk of death during hospitalization for this patient is 2.41%.

undergo an elective (i.e., non-emergency) coronary artery bypass grafting (CABG) operation. The patient has pulmonary hypertension and a preoperative serum creatinine value of 80 $\mu\text{mol/l}$. Six network variables (`age`, `diabetes`, `surtype`, `emerg`, `pulmhyp`, and `precreat`) have been instantiated with the values that represent this information. For the other variables in the network, no value is known (yet), and the conditional probability distribution over their possible values has been computed. This patient case is used as running example in Section 4.

3 Learning algorithm

This section presents a new algorithm that was developed to induce PBNs from medical datasets. The algorithm takes explicitly account of several important characteristics of PBNs, and is therefore more suited to learn these networks from data than general Bayesian network learning algorithms. In particular, the algorithm exploits the temporal structure of the health care process being modeled, optimizes performance of the network’s primary task, outcome prediction, and adequately handles *patient dropout*, i.e., the phenomenon that patients may die during this process.

In the past 15 years, several algorithms for learning Bayesian networks from data have been developed, [6, 9, 10, 13]. In these standard algorithms, learning a Bayesian network is considered to be an unsupervised learning problem. That is, no variable is considered to be more important than any other variable, and the network structure is built up by recursively adding arcs between pairs of variables that appear most strongly correlated in the data. Furthermore, these algorithms assume that the same amount of observations is available for all network variables (i.e., the network is learned from a ‘flat table’).

Both these assumptions fail for PBNs. In prognostic tasks, the outcome variable is far more important than all other variables, and its accurate prediction for future cases must be the primary focus of optimization. Furthermore, patients who die halfway during the process (e.g., in the operating room) will not reach subsequent phases of care. For these patients, therefore, data on subsequent phases is systematically lacking.

In our algorithm, Bayesian network learning is considered as a collection of local supervised learning problems. Local models are recursively learned in a top-down approach, starting at the outcome variable of the health care process; in our application, local tree models were built. These local models are subsequently combined to obtain the Bayesian network. This proceeds as follows. Let $\mathbf{X} = \{X_1, \dots, X_m\}$ denote the set of domain variables, where X_m denotes the outcome variable of the process described by \mathbf{X} ; X_m is therefore also denoted by Y . We use $\sigma_G^*(X_i)$ to denote the set of all *descendants* of node X_i in graph G . The algorithm now consists of the following steps:

1. Define a preorder \preceq on the domain variables, based on their occurrence in time.
2. Let G be an empty graph with nodes corresponding to \mathbf{X} and no arcs.
3. Let Q be a queue with the outcome variable $Y \in \mathbf{X}$ as its only element.
4. $X_i \leftarrow \text{dequeue}(Q)$.
 - (a) Define $R_{X_i} \leftarrow \{X_j \in \mathbf{X} \mid X_i \not\preceq X_j, X_j \notin \sigma_G^*(X_i)\}$.
 - (b) Select $S_{X_i} \leftarrow \text{argmax}\{\mathbf{X}' \subseteq R_{X_i} \mid I(X_i, \mathbf{X}')\}$.
 - (c) For each $X_j \in S_{X_i}$, add an arc $X_j \rightarrow X_i$ to graph G .
 - (d) Build a model f_{X_i} for variable X_i using S_{X_i} .
 - (e) Enqueue all variables from S_{X_i} into Q .
5. Repeat from step 4 until $Q = \emptyset$.

The preorder \preceq that is defined in step 1 ensures that the resulting network mirrors the flow of time through the care process that is being modeled. In the cardiac surgery domain, all variables from the preoperative phase (`age`, `bmi`, `diabetes`, `ejfrac`, `precreat`, `pulmhyp`, `surtype`) are known first, in no specific order. Then, the variables related to the surgical procedure (`emerg`, `ecctime`, `eccacctime`, `temp`) obtain their values, in that order. Subsequently, we observe the variables from the first 24h at the ICU (`bicmin`, `creatmax`, `fio2`, `meanbmax`, `meanbmin`, `ptt`), in no specific order,

whether the patient remained longer than 24h at the ICU (`ICUlos24h`), whether complications occurred during the entire postoperative phase (`cardcomp`, `infect`, `mof`, `neurcomp`, `pulmcomp`), and finally we know whether the patient survived the entire care process (`hospmort`).

At each iteration of the algorithm, the set R_{X_i} consists of all non-descendants of node X_i in the network that do not occur later than X_i in time. So, we can freely select parent nodes for X_i from this set without creating cycles in the graph and while respecting the order defined by \preceq . From these candidate parent nodes we select a feature subset that has maximal predictive value (in terms of information gain) for variable X_i using heuristic search, and build a local model for X_i . The collection of local models eventually defines the numerical part of the PBN.

As mentioned earlier, patients may die halfway during the process, in the operation room; in our case, 66 patients in the dataset died during surgery. For these patients, no postoperative information is recorded, and the postoperative variables become irrelevant. To correctly capture this phenomenon of patient dropout in the Bayesian network, we must separate operative and postoperative mortality in our representation. We therefore included separate variables, `ORMort` and `postORMort`, for these two types of mortality in the network, and defined the global outcome hospital mortality in terms of them:

$$\text{hospmort} = \begin{cases} \text{T}, & \text{if } \text{ORMort} = \text{T} \text{ or } \text{postORMort} = \text{T}, \\ \text{F}, & \text{otherwise.} \end{cases}$$

We refer to the variables `ORMort` and `postORMort` as *subsidiary outcomes*. They become the parent nodes of the global outcome `hospmort` in the network.

Between the subsidiary outcome operative mortality (`ORMort`) and each postoperative variable, including the subsidiary outcome `postORMort`, a simple deterministic relationship exists. When category ‘I’ denotes irrelevancy of the variable in question, it formally holds that $P(X_i = \text{I} \mid \text{ORMort} = \text{T}) = 1$ for each postoperative variable $X_i \in \mathbf{X}$, independent of any other variable. These relationships can be modeled in the network by drawing arcs from the subsidiary outcome `ORMort` to each postoperative variable X_i and extending the corresponding local conditional probability models with the deterministic relation. Figure 2 provides a schematic depiction of the graphical part of the PBN. In Figure 1, both subsidiary outcomes are included in the PBN to represent patient dropout; the deterministic relations, however, are not represented in this network.

In the learning algorithm, the analyses of feature subset selection and local model building for the subsidiary outcome `postORMort` and all other postoperative variables are based on a subgroup of patients that survived the operation, and the analyses of the subsidiary outcome `ORMort` and all the other (pre)operative variables are based on the dataset of which no patients are excluded.

We applied this learning algorithm to data of 6778 randomly chosen patients. The method of class probability trees from the tree building methodology *Classification and Regression Trees* (CART) [4] was used as supervised learning method for feature subset selection and local model building. Figure 1 shows the resulting PBN. Data of the remaining 3336 patient was used for network evaluation. We calculated the area under the ROC curve (AUC) [15] and the standardized

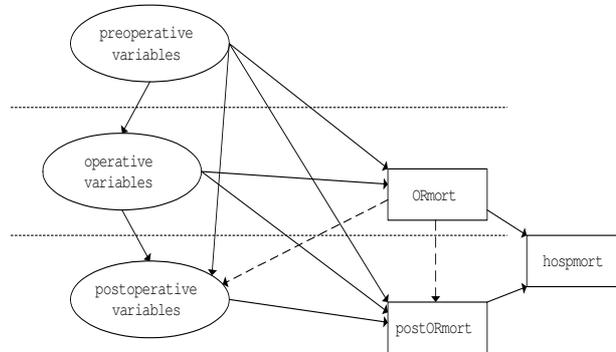


Figure 2: Schematic network structure that represents patient dropout by the subsidiary outcome variable `ORMort` (operative mortality) and `postORMort` (postoperative mortality); the dotted arcs represent the deterministic relationships that exist between operative mortality and the postoperative (subsidiary outcome) variables.

Table 1: The performance of the PBN developed with (a) the new learning algorithm and (b) BN PowerConstructor for predicting future events in the preoperative stage.

variable	(a)		(b)		variable	(a)		(b)	
	AUC	R^2	AUC	R^2		AUC	R^2	AUC	R^2
hospmort	0.767	0.011	0.693	0.004	pulmcomp	0.654	0.009	0.609	0.006
cardcomp	0.602	0.024	0.498	0.002	mof	0.792	0.005	0.689	0.006
neurcomp	0.707	0.014	0.674	0.007	infect	0.643	0.008	0.602	0.006

mean squared error or R^2 [11] for the outcome variable hospital mortality (`hospmort`) and the postoperative variables (cardiac, neurological, and pulmonary complication, multiple organ failure, and infection) at two prediction times: (i) during the preoperative stage, and (ii) at ICU admission. We compared the PBN to a network that was learned using the conventional software package BN PowerConstructor¹. The performance of both PBNs for prediction during the preoperative stage is listed in Table 1.

4 Prognostic tasks

In this section, we discuss a number of prognostic reasoning tasks that are frequently carried out by medical practitioners, and describe how these tasks are supported by PBNs in clinical practice.

The primary task supported by PBNs is *prognosis*, i.e., estimating the distribution of variables that represent future events. These events may pertain to conditions that occur during the process or to endpoints of the process. For example, the 62-year-old CABG patient has a .0039 probability of dying during the operation (`ORMort`), a .0158 probability of postoperative multiple organ failure (`mof`), and a .0203 postoperative probability of dying (`postORMort`), yielding a total death risk of 2.41% (`hospmort`); these probabilities are shown in Figure 1.

Sometimes it is not possible to collect all the information of a case at hand, while a prediction would still be useful. In an emergency setting, for instance, one may not know whether a patient is diabetic or not. Bayesian networks can perform probabilistic inference with any number of observed variables; this property allows us to make predictions with PBNs on the basis of incomplete information, and to apply PBNs for *quick prognostic assessment*.

Furthermore, a patient’s prognosis is not a static concept, but changes as the process progresses and new information becomes available. The Bayesian methodology that underlies PBNs allows us to implement a dynamic notion of prognosis, by employing probability updating based on this new information. The PBN thus provides clinicians who are involved in later phases of the process with predictions that are adjusted for the course of the preceding phases. In addition to the adjusted risk estimations, the *change* in a patient’s prognosis contains important information about risk progress. For example, a complicated surgical procedure is generally accompanied by a long duration of the operation (`ecctime`, `eccacctime`). In this case, the optimistic prognosis that could be assessed preoperatively for the running example will change to a rather poor prognosis for the further course of the process. The estimated probability to die during hospitalization increases for instance from 2.41% to 4.12%.

A death risk of 4.12% may seem small for a serious intervention such as cardiac surgery. However, the a priori risk in this health care process is 2.74%, and in the early phases, the estimated death risks range from 1 to 10%, where a risk of 10% indicates a very high risk. The risk of 4.12% is a factor of 1.7 higher than the initial estimated risk of dying for the example patient (2.41%). So, the *relative risk* indicates more clearly the risk progress than an absolute risk frequency does. The results of probabilistic inference in a PBN therefore need to be related to a priori risks or risks that were calculated for the general patient population.

We briefly mention two other tasks of PBNs in this paper: (i) analysis of *what-if scenarios*, and (ii) *risk factor analysis*. In a what-if scenario analysis, the impact of a clinical event that may occur during the health care process (e.g., a particular complication) on the expectations for the

¹<http://www.cs.ualberta.ca/~jcheng/bnpc.htm>

future part of the process is analyzed. In risk factor analysis, the predictive value of variables that precede a clinical event in the process is quantified to provide insight into important predictors of the event that probably can be influenced by the clinical staff.

The prognostic tasks can be accomplished by performing ‘conventional’ probabilistic queries on the PBN, but they generally require that multiple queries be performed and the results be aggregated. To support the use of PBNs in medical practice, the PBN is equipped with a prognostic task layer that holds procedures to perform these tasks. We implemented the task layer in the cardiac surgical application of a PBN, the Procarsur system; a demonstration summary of this system has been accepted for the BNAIC’06 Conference.

5 Discussion and conclusions

This paper presents the PBN as a new type of prognostic model that builds on the Bayesian network methodology. Due to the expressiveness of Bayesian networks, PBNs can answer many different questions that are related to medical prognosis. These answers are generally not provided directly by the Bayesian network, though. A layer that holds knowledge of these tasks is therefore necessary as intermediary between the PBN and the clinical user to support the use of PBNs in practice. The two-layered architecture of PBNs separates domain knowledge and task knowledge, a common strategy in AI. For Bayesian networks, this was earlier proposed for diagnosis in [18], and a similar type of separation is found in the diagnostic Promedas system [12]. To date, the concept was not applied to Bayesian networks for *prognosis*, though.

In addition, knowledge of the prognostic application of the network is used in the algorithm for learning PBNs that is presented in this paper. The algorithm is based on the correspondence that building the graphical part of a Bayesian network boils down to finding the Markov blanket for each variable, i.e., the set of variables that separates this variables from all others, which in turn corresponds to selecting the best predictive feature subset for this variable in a representative dataset. In the algorithm, tree induction was used for feature subset selection. In principle, however, any feature subset selection method can be plugged into our algorithm.

In the learning algorithm, the tree models were also used to represent the local conditional probability distribution (cpd) of each variable given its parents in the Bayesian network. As such, our work is related to the work of N. Friedman *et al.* [9] and D.W. Chickering *et al.* [5]. In these studies, tree models were used for compact representation of the local cpds, and it was shown how such representations can be exploited K2-type methods [6] for learning Bayesian networks from data. The number of parameters that are required to encode the cpd is lower in these local models than in a tabular representation, which results in more reliable estimation of the distributions.

The health care processes that are modeled in PBNs are composed of a sequence of substantially different phases, and have no recurring character such as a Markov process. Therefore, although the notion of time is important in this type of modeling problem, the data are not suitable to be modeled as a dynamic Bayesian network [7].

To conclude, the main contribution of this paper is the introduction of the concept of prognostic Bayesian networks. We presented a dedicated algorithm for learning PBNs from data, and described the prognostic tasks that can be supported by these networks. In the learning algorithm, the network is built up from local models that are learned with a supervised learning algorithm. The algorithm takes explicitly account of the outcome variable of the medical process being modeled and of the phenomenon of patient dropout. The PBN extends the functionality of the traditional prognostic model. To realize the prognostic tasks, the PBN is equipped with a task layer that serves as intermediary between the PBN and the user. As such, we adapted the Bayesian network methodology for prognostic applications in medicine.

Acknowledgments

Niels Peek receives a grant from the Netherlands Organization of Scientific Research (NWO) under project number 634.000.020. The authors wish to thank Ameen Abu-Hanna and Frans Voorbraak for their comments on the ideas presented.

References

- [1] A. Abu-Hanna and P.J.F. Lucas. Prognostic models in medicine. *Methods of Information in Medicine*, 40:1–5, 2001.
- [2] J. R. Beck and S. G. Pauker. The Markov process in medical prognosis. *Medical Decision Making*, 3:419–458, 1993.
- [3] R. Bellazzi, C. Berzuini, et al. Cytotoxic chemotherapy monitoring using stochastic simulation on graphical models. In M. Stefanelli, A. Hasman, and M. Fieschi, editors, *Proceedings of the Third Conference on Artificial Intelligence in Medicine*, pages 227–238, Berlin, 1991. Springer-Verlag.
- [4] L. Breiman, J. H. Friedman, et al. *Classification and Regression Trees*. Wadsworth & Brooks, Monterey, 1984.
- [5] D. M. Chickering, D. Heckerman, and C. Meek. A Bayesian approach to learning Bayesian networks with local structure. In *Proceedings of the thirteenth Conference on Uncertainty in Artificial Intelligence*, pages 80–89, 1997.
- [6] G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.
- [7] P. Dagum and A. Galper. Time series prediction using belief network models. *International Journal of Human-Computer Studies*, 42:617–632, 1995.
- [8] L. Frey, D. Fisher, et al. Identifying Markov blankets with decision tree induction. In *Proceedings of the third IEEE International Conference on Data Mining*, pages 59–66, 2003.
- [9] N. Friedman and M. Goldszmidt. Learning Bayesian network with local structure. In *Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence*, San Francisco, CA, 1996. Morgan Kaufmann.
- [10] D. E. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: the combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.
- [11] D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression*. John Wiley & Sons, New York, 1989.
- [12] H. J. Kappen and J. P. Neijt. Promedas, a probabilistic decision support system for medical diagnosis. Technical report, SNN Foundation for Neural Networks, Nijmegen, and University Medical Center Utrecht, Utrecht, the Netherlands, 2002.
- [13] W. Lam and F. Bacchus. Learning Bayesian belief networks. an approach based on the mdl principle. *Computational Intelligence*, 10:269–293, 1994.
- [14] P. J. F. Lucas, L. C. van der Gaag, and A. Abu-Hanna. Bayesian networks in biomedicine and health care. *Artificial Intelligence in Medicine*, 30:201–214, 2004.
- [15] C. E. Metz. Basic principles of ROC analysis. *Seminars in Nuclear Medicine*, 8:283–298, 1978.
- [16] M. A. Shwe, B. Middleton, et al. Probabilistic diagnosis using reformulation of the INTERNIST-1/QMR knowledge base. the probabilistic model and inference algorithm. *Methods of Information in Medicine*, 30:241–255, 1991.
- [17] L. C. van der Gaag, S. Renooij, et al. Probabilities for a probabilistic network: a case study in oesophageal cancer. *Artificial Intelligence in Medicine*, 25:123–148, 2002.
- [18] L. C. van der Gaag and M. L. Wessels. Selective evidence gathering for diagnostic belief networks. *AISB Quarterly*, (86):23–34, 1993.
- [19] J. Wyatt and D. G. Altman. Prognostic models: clinically useful or quickly forgotten? *British Medical Journal*, 311:1539–1541, 1995.

Robot companion localization at home and in the office

Arnoud Visser Jürgen Sturm Frans Groen

Intelligent Autonomous Systems, Universiteit van Amsterdam

<http://www.science.uva.nl/research/ias/>

Abstract

The abilities of mobile robots depend greatly on the performance of basic skills such as vision and localization. Although great progress has been made to explore and map extensive public areas with large holonomic robots on wheels, less attention is paid on the localization of a small robot companion in a confined environment as a room in office or at home. In this article, a localization algorithm for the popular Sony entertainment robot Aibo inside a room is worked out. This algorithm can provide localization information based on the natural appearance of the walls of the room. The algorithm starts making a scan of the surroundings by turning the head and the body of the robot on a certain spot. The robot learns the appearance of the surroundings at that spot by storing color transitions at different angles in a panoramic index. The stored panoramic appearance is used to determine the orientation (including a confidence value) relative to the learned spot for other points in the room. When multiple spots are learned, an absolute position estimate can be made. The applicability of this kind of localization is demonstrated in two environments: at home and in an office.

1 Introduction

1.1 Context

Humans orientate easily in their natural environments. To be able to interact with humans, mobile robots also need to know where they are. Robot localization is therefore an important basic skill of a mobile robot, as a robot companion like the Aibo. Yet, the Sony entertainment software contained no localization software until the latest release¹. Still, many other applications for a robot companion - like collecting a news paper from the front door - strongly depend on fast, accurate and robust position estimates. As long as the localization of a walking robot, like the Aibo, is based on odometry after sparse observations, no robust and accurate position estimates can be expected.

Most of the localization research with the Aibo has concentrated on the RoboCup. At the RoboCup² artificial landmarks as colored flags, goals and field lines can be used to achieve localization accuracies below six centimeters [6, 8].

The price that these RoboCup approaches pay is their total dependency on artificial landmarks of known shape, positions and color. Most algorithms even require manual calibration of the actual colors and lighting conditions used on a field and still are quite susceptible for disturbances around the field, as for instance produced by brightly colored clothes in the audience.

The interest of the RoboCup community in more general solutions has been (and still is) growing over the past few years. The almost-SLAM challenge³ of the 4-Legged league is a good example of the state-of-the-art in this community. For this challenge additional landmarks with bright colors are placed around the borders on a RoboCup field. The robots get one minute to walk around and explore the field. Then, the normal beacons and goals are covered up or removed, and the robot must then move to a series of five points on the field, using the information learnt during the first

¹Aibo Mind 3 remembers the direction of its station and toys relative to its current orientation

²RoboCup Four Legged League homepage, last accessed in May 2006, <http://www.tzi.de/4legged>

³Details about the Simultaneous Localization and Mapping challenge can be found at <http://www.tzi.de/4legged/pub/Website/Downloads/Challenges2005.pdf>

minute. The winner of this challenge [6] reached the five points by using mainly the information of the field lines. The additional landmarks were only used to break the symmetry on the soccer field.

A more ambitious challenge is formulated in the newly founded RoboCup @ Home league⁴. In this challenge the robot has to safely navigate toward objects in the living room environment. The robot gets 5 minutes to learn the environment. After the learning phase, the robot has to visit 4 distinct places/objects in the scenario, at least 4 meters away from each other, within 5 minutes.

1.2 Related Work

Many researchers have worked on the SLAM problem in general, for instance on panoramic images [1, 2, 4, 5]. These approaches are inspiring, but only partially transferable to the 4-Legged league. The Aibo is not equipped with an omni-directional high-quality camera. The camera in the nose has only a horizontal opening angle of 56.9 degrees and a resolution of 416 x 320 pixels. Further, the horizon in the images is not a constant, but depends on the movements of the head and legs of the walking robot. So each image is taken from a slightly different perspective, and the path of the camera center is only in first approximation a circle. Further, the images are taken while the head is moving. When moving at full speed, this can give a difference of 5.4 degrees between the top and the bottom of the image. So the image seems to be tilted as a function of the turning speed of the head. Still, the location of the horizon can be calculated by solving the kinematic equations of the robot. To process the images, a 576 Mhz processor is available in the Aibo, which means that only simple image processing algorithms are applicable. In practice, the image is analyzed by following scan-lines with a direction relative the calculated horizon. In our approach, multiple sectors above the horizon are analyzed, with in each sector multiple scan-lines in the vertical direction. One of the general approaches [3] divides the image in multiple sectors, but this image is omni-directional and the sector is analyzed on the average color of the sector. Our method analyzes each sector on a different characteristic feature: the frequency of color transitions.

2 Approach

The main idea is quite intuitive: we would like the robot to generate and store a 360° circular panorama image of its environment while it is in the learning phase. After that, it should align each new image with the stored panorama, and from that the robot should be able to derive its relative orientation (in the localization phase). This alignment is not trivial because the new image can be translated, rotated, stretched and perspectively distorted when the robot does not stand at the point where the panorama was originally learned [11].

Of course, the Aibo is not able (at least not in real-time) to compute this alignment on full-resolution images. Therefore a reduced feature space is designed so that the computations become tractable⁵ on an Aibo. So, a reduced circular 360° panorama model of the environment is learned. Figure 1 gives a quick overview of the algorithm's main components.

The Aibo performs a **calibration** phase before the actual learning can start. In this phase the Aibo first decides on a suitable camera setting (i.e. camera gain and the shutter setting) based on the dynamic range of brightness in the **autoshutter** step. Then it collects color pixels by turning its head for a while and finally clusters these into 10 most important color classes in the **color clustering** step using a standard implementation of the Expectation-Maximization algorithm assuming a Gaussian mixture model [9]. The result of the calibration phase is an automatically generated lookup-table that maps every YCbCr color onto one of the 10 color classes and can therefore be used to segment incoming images into its characteristic color patches (see figure 2(a)). These initialization steps are worked out in more detail in [10].

⁴RoboCup @ Home League homepage, last accessed in May 2006, <http://www.ai.rug.nl/robocupathome/>

⁵Our algorithm consumes per image frame approximately 16 milliseconds, therefore we can easily process images at the full Aibo frame rate (30fps).

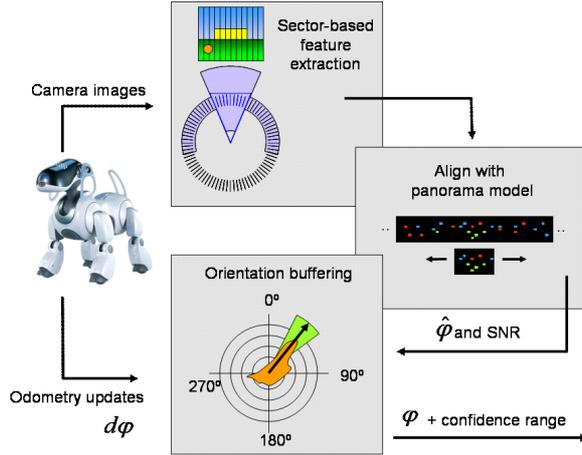
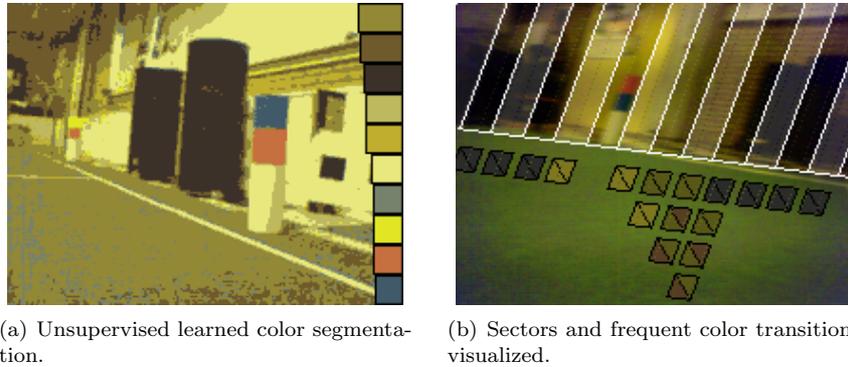


Figure 1: Architecture of our algorithm



(a) Unsupervised learned color segmentation.

(b) Sectors and frequent color transitions visualized.

Figure 2: Image processing: from the raw image to sector representation. This conversion consumes approximately 6 milliseconds/frame on a Sony Aibo ERS7.

2.1 Sector signature correlation

Every incoming image is now divided into its corresponding sectors⁶. The sectors are located above the calculated horizon, which is generated by solving the kinematics of the robot. Using the lookup table from the unsupervised learned color clustering, we can compute the sector features by counting per sector the transition frequencies between each two color classes in vertical direction. This yields the histograms of 10x10 transition frequencies per sector, which we subsequently discretize into 5 logarithmically scaled bins. In figure 2(b) we displayed the most frequent color transitions for each sector. Some sectors have multiple color transitions in the most frequent bin, other sectors have a single or no dominant color transition. This is only visualization; not only the most frequent color transitions, but the frequency of all 100 color transitions are used as characteristic feature of the sector.

In the **learning phase** we estimate all these 80x(10x10) distributions⁷ by turning the head and body of the robot. We define a single distribution for a currently perceived sector by

$$P_{current}(i, j, bin) = \begin{cases} 1 & \text{discretize}(freq(i, j)) = bin \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where i, j are indices of the color classes and bin one of the five frequency bins. Each sector is seen multiple times and the many frequency count samples are combined into a distribution learned

⁶80 sectors corresponding to 360°; with an opening angle of the Aibo camera of approx. 50°, this yields between 10 and 12 sectors per image (depending on the head pan/tilt)

⁷When we use 16bit integers, a complete panorama model can be described by (80 sectors)x(10 colors x 10 colors)x(5 bins)x(2 byte) = 80 KB of memory.

for that sector by the equation:

$$P_{learned}(i, j, bin) = \frac{count_{sector}(i, j, bin)}{\sum_{bin \in frequencyBins} count_{sector}(i, j, bin)} \quad (2)$$

After the learning phase we can simply multiply the current and the learned distribution to get the correlation between a currently perceived and a learned sector:

$$Corr(P_{current}, P_{learned}) = \prod_{\substack{i, j \in colorClasses, \\ bin \in frequencyBins}} P_{learned}(i, j, bin) \cdot P_{current}(i, j, bin) \quad (3)$$

2.2 Alignment

After all the correlations between the stored panorama and the new image signatures were evaluated, we would like to get an alignment between the stored and seen sectors so that the overall likelihood of the alignment becomes maximal. In other words, we want to find a diagonal path with the minimal cost through the correlation matrix. This minimal path is indicated as green dots in figure 3. The path is extended to a green line for the sectors that are not visible in the latest perceived image.

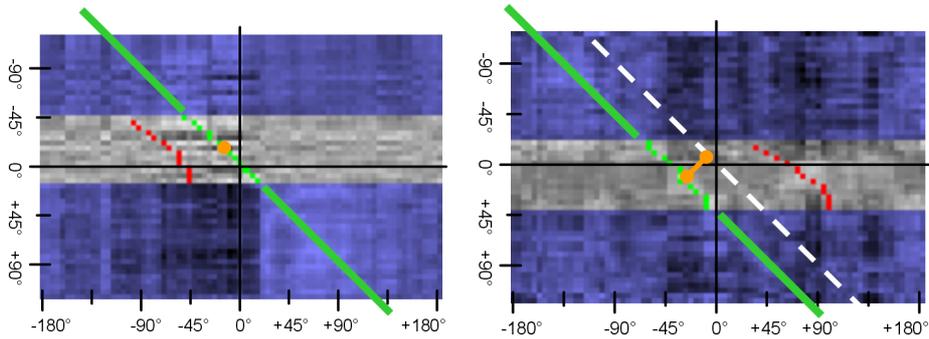
We consider the fitted path to be the true alignment and extract the rotational estimate φ_{robot} from the offset from its center pixel to the diagonal ($\Delta_{sectors}$):

$$\hat{\varphi}_{robot} = \frac{360^\circ}{80} \Delta_{sectors} \quad (4)$$

This rotational estimate is the difference between the solid green line and the dashed white line in figure 3, indicated by the orange halter. Further, we try to estimate the noise by fitting again a path through the correlation matrix far away from the best-fitted path.

$$SNR = \frac{\sum_{(x,y) \in minimumPath} Corr(x, y)}{\sum_{(x,y) \in noisePath} Corr(x, y)} \quad (5)$$

The noise path is indicated in figure 3 with red dots.



(a) Robot standing on the trained spot (matching line is just the diagonal) (b) Robot turned right by 45 degrees (matching line displaced to the left)

Figure 3: Visualization of the alignment step while the robot is scanning with its head. The green solid line marks the minimum path (assumed true alignment) while the red line marks the second-minimal path (assumed peak noise). The white dashed line represents the diagonal, while the orange halter illustrates the distance between the found alignment and the center diagonal ($\Delta_{sectors}$).

2.3 Position Estimation with Panoramic Localization

The algorithm described in the previous section can be used to get a robust bearing estimate together with a confidence value for a single trained spot. As we finally want to use this algorithm to obtain full localization we extended the approach to support multiple training spots. The main idea is that the robot determines to which amount its current position resembles with the previously learned spots and then uses interpolation to estimate its exact position. As we think that this approach could also be useful for the RoboCup @ Home league (where robot localization in complex environments like kitchens and living rooms is required) it could become possible that we finally want to store a comprehensive panorama model library containing dozens of previously trained spots (for an overview see [1]).

However, due to the computation time of the feature space conversion and panorama matching, per frame only a single training spot and its corresponding panorama model can be selected. Therefore, the robot cycles through the learned training spots one-by-one. Every panorama model is associated with a gradually changed confidence value representing a sliding average on the confidence values we get from the per-image matching.

After training, the robot memorizes a given spot by storing the confidence values received from the training spots. By comparing a new confidence value with its stored reference, it is easy to deduce whether the robot stands closer or farther from the imprinted target spot.

We assume that the imprinted target spot is located somewhere between the training spots. Then, to compute the final position estimate, we simply weight each training spot with its normalized corresponding confidence value:

$$position_{robot} = \sum_i position_i \frac{confidence_i}{\sum_j confidence_j} \quad (6)$$

This should yield zero when the robot is assumed to stand at the target spot or a translation estimate towards the robot's position when the confidence values are not in balance anymore.

To prove the validity of this idea, we trained the robot on four spots on regular 4-Legged field in our robolab. The spots were located along the axes approximately 1m away from the center. As target spot, we simply chose the center of the field. The training itself was performed fully autonomously by the Aibo and took less than 10 minutes. After training was complete, the Aibo walked back to the center of the field. We recorded the found position and kidnapped the robot to an arbitrary position around the field and let it walk back again.

Please be aware that our approach for multi-spot localization is at this moment rather primitive and has to be only understood as a proof-of-concept. In the end, the panoramic localization data from vision should of course be processed by a more sophisticated localization algorithm, like a Kalman or particle filter (last not least to incorporate movement data from the robot).

3 Results

3.1 Environments

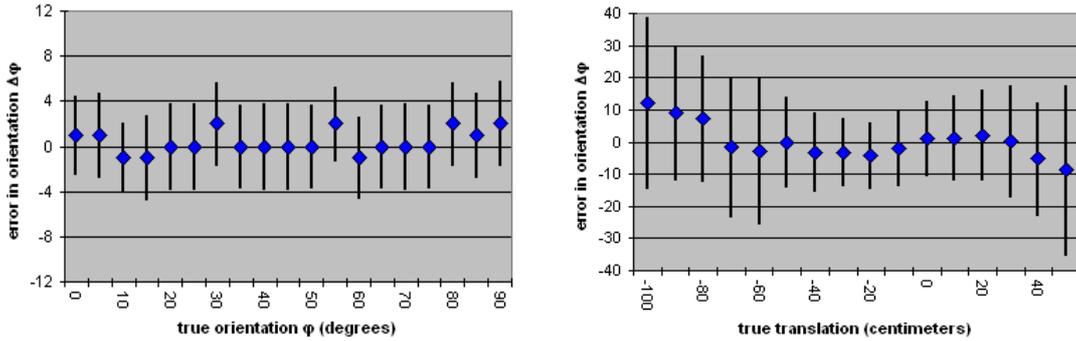
We selected four different environments to test our algorithm under a variety of circumstances. The first two experiments were conducted at home and in an office environment⁸ to measure performance under real-world circumstances. The experiments were performed on a cloudy morning, sunny afternoon and late in the evening. Furthermore, we conducted exhaustive tests in our laboratory. Even more challenging, we took an Aibo outdoors (see [7]).

3.2 Measured results

Figure 4(a) illustrates the results of a rotational test in a normal living room. As the error in the rotation estimates ranges between -4.5 and +4.5 degrees, we may assume an error in alignment of a single sector⁹; moreover, the size of the confidence interval can be translated into maximal two sectors, which corresponds to the maximal angular resolution of our approach.

⁸XX office, DECIS lab, Delft

⁹full circle of 360° divided by 80 sectors



(a) Rotational test in natural environment (living room, sunny afternoon)

(b) Translational test in natural environment (child's room, late in the evening)

Figure 4: Typical orientation estimation results of experiments conducted at home. In the rotational experiment on the left the robot is rotated over 90 degrees on the same spot, and every 5 degrees is estimated. The robot is able to find its true orientation with an error estimate equal to one sector of 4.5 degrees. The translational test on the right is performed in a child's room. The robot is translated over a straight line of 1.5 meter, which covers the major part of the free space in this room. The robot is able to maintain a good estimate of its orientation; although the error estimate increases away from the location where the appearance of the surroundings was learned.

Figure 4(b) shows the effects of a translational dislocation in a child's room. The robot was moved onto a straight line back and forth through the room (via the trained spot somewhere in the middle). The robot is able to estimate its orientation quite well on this line. The discrepancy with the true orientation is between +12.1 and -8.6 degrees, close to the walls. This is also reflected in the computed confidence interval, which grows steadily when the robot is moved away from the trained spot. The results are quite impressive for the relatively big movements in a small room and the resulting significant perspective changes in that room.

Figure 5(a) also stems from a translational test (cloudy morning) which has been conducted in an office environment. The free space in this office is much larger than at home. The robot was moved along a 14m long straight line to the left and right and its orientation was estimated. Note the error estimate stays low at the right side of this plot. This is an artifact which nicely reflects the repetition of similarly looking working islands in the office.

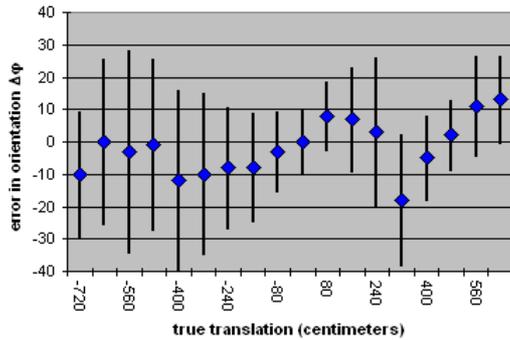
In both translational tests it can be seen intuitively that the rotation estimates are within acceptable range. This can also be shown quantitatively (see figure 5(b)): both the orientation error and the confidence interval increase slowly and in a graceful way when the robot is moved away from the training spot.

Finally, figure 6 shows the result of the experiment to estimate the absolute position with multiple learned spots. It can be seen that the localization is not as accurate as traditional approaches, but can still be useful for some applications (bearing in mind that no artificial landmarks are required). We recorded repeatedly a derivation to the upper right that we think can be explained by the fact that different learning spots don't produce equally strong confidence values; we believe to be able to correct for that by means of confidence value normalization in the near future.

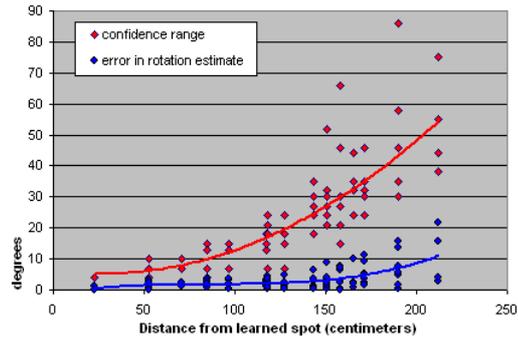
4 Conclusion

Although at first sight the algorithm seems to rely on specific texture features of the surrounding surfaces, in practice no dependency could be found. This can be explained by two reasons: firstly, as the (vertical) position of a color transition is not used anyway, the algorithm is quite robust against (vertical) scaling. Secondly, as the algorithm aligns on many color transitions in the background (typically more than a hundred in the same sector), the few color transitions produced by objects in the foreground (like beacons and spectators) have a minor impact on the match (because their sizes relative to the background are comparatively small).

The lack of an accurate absolute position estimates seems to be a clear drawback with respect to the other methods, but bearing information alone can already be very useful for certain applications.



(a) Translational test in natural environment (office, cloudy morning)



(b) Signal degradation as a function of the distance to the learned spot (measured in the laboratory)

Figure 5: Challenging orientation results. On the left a translational test in office environment over 14 meters along a line 80 centimeters from the learned spot (only one). A translation to the left of the office increases the error estimate increases, as expected. When translating to the right of the office to the orientation estimate oscillates, but the error estimate stays low. This is due to repeating patterns in the office, after 4 meters there is another group of desks and chairs which resembles the learned appearance. Normally the error estimate increases as a second order polynome of the distance, as demonstrated with our qualitative measurements on the right. For this measurements the rotation estimate was recorded on a grid of 99 locations regularly distributed over our laboratory.

For example, a robot companion can highly benefit from a robust bearing estimation towards the front door. With this bearing estimating the robot can walk into the right direction. Only when the front door is nearly reached, and the robot has to search for a newspaper, additional distance estimation could be advantageous.

The requirements of this approach for both the robot as well as for its environment are quite moderate (on a Sony Aibo ERS7, the computation time is below 20ms/frame). The robot itself needs only a simple camera and medium computational power, while most natural environments carry, as shown, enough panoramic information the algorithm can lock on to. Therefore, this method becomes interesting for the newly established RoboCup @ Home league, where fast localization information is needed in natural but completely unknown environments.

As the training on a single spot can be completed in less than one minute on a Sony Aibo in an arbitrary place, robot research can move outside the laboratory into non-prepared places. Progress in this domain facilitates the advancement of mobile robots - and thereby robotics research itself - into more natural environments.

Acknowledgements

The Dutch AIBO Team is financially supported by DECIS Lab, University of Amsterdam, Technical University of Delft, Saxion University of Professional Education, University of Groningen and University of Utrecht. We are grateful for the contributions by all of our team members, working on RoboCup related research and other research. Special thanks to Olaf Booij, David Knibbe and Niek Wijngaards.

References

- [1] R. Bunschoten. *Mapping and Localization from a Panoramic Vision Sensor*. PhD thesis, Universiteit van Amsterdam, November 2003.
- [2] J. Gonzalez and S. Lacroix. Rover localization in natural environments by indexing panoramic images. In *International Conference on Robotics and Automation 2002*, Washington, DC (USA), 2002.

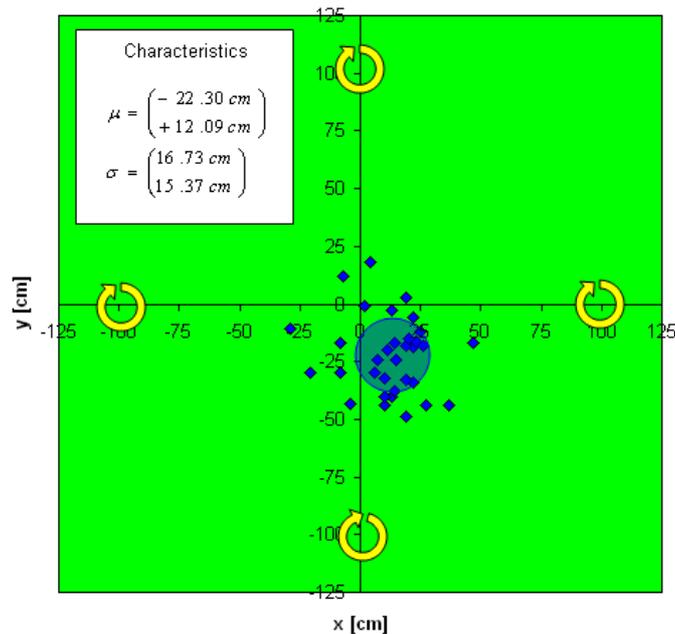


Figure 6: The position estimation after learning the appearance of the surroundings on 4 spots (indicated with yellow arrows). The small dots indicate the distribution of positions reached every time the robot tried to return to the central spot after been placed randomly somewhere on the field.

- [3] Horst-Michael Gross et al. Omnivision-based probabilistic self-localization for a mobile shopping assistant continued. In F.C.A. Groen, editor, *IEEE/RSJ Int. Conf. on Intell. Robots and Systems (IROS 2003)*, pages 1505–1511. IEEE omnipress, 2003.
- [4] Matjaz Jogan and Leonardis Ales. Robust localization using panoramic view-based recognition. In *Proceedings 15th International Conference on Pattern Recognition ICPR'00*, pages 136–139, Barcelona, Spain, 2000.
- [5] T. Pajdla and V.Hlavac. Zero phase representation of panoramic images for image based localization. In *Proceedings of the 8th International Conference on Computer Analysis of Images and Patterns: CAIP'99*, volume 1689 of *Lecture Notes on Computer Science*, pages 550–557. Springer Verlag, September 1999.
- [6] T. Röfer et al. *GermanTeam RoboCup 2005*. 247 pages, 2005.
- [7] J. Sturm, P. van Rossum, and A. Visser. Panoramic localization in the 4-legged league. In *Proc. 10th RoboCup International Symposium*, Bremen, June 2006. To be published in the *Lecture Notes on Artificial Intelligence* series, Springer Verlag, Berlin.
- [8] J. Sturm, A. Visser, and N. Wijngaards. Dutch aibo team: Technical report robocup 2005. Technical report, Dutch Aibo Team, October 2005.
- [9] J.J. Verbeek. *Mixture models for clustering and dimension reduction*. PhD thesis, Universiteit van Amsterdam, December 2004.
- [10] Arnoud Visser, Paul van Rossum, Joost Westra, Jürgen Sturm, Dave van Soest, and Mark de Greef. Dutch aibo team at robocup 2006. In *Proceedings CD RoboCup 2006, Bremen, Germany*, June 2006.
- [11] A. Zomet, D. Feldman, S. Peleg, and D. Weinshall. Mosaicing new views: The crossed-slits projection, June 2003.

Compressed Contributions

Inference of Concise DTDs from XML Data¹

Geert Jan Bex ^a Frank Neven ^a Thomas Schwentick ^b Karl Tuyls ^c

^a *Hasselt University and Transnational University of Limburg*

^b *Dortmund University*

^c *Maastricht University and Transnational University of Limburg*

1 Motivation

XML is the lingua franca for data exchange on the Internet. Within applications or communities, XML data is usually not arbitrary but adheres to some structure possibly imposed by a schema. The advantages offered by the presence of such a schema are numerous. The most direct application is of course automatic validation of the document structure. Input validation, for instance, not only facilitates automatic processing but also ensures soundness of the input data. Indeed, unvalidated input from web requests is considered as the number one vulnerability for web applications. The presence of a schema allows for automation and optimization of search, integration, and processing of XML data. Further, the existence of schemas is imperative when integrating (meta) data through schema matching and in the area of generic model management. A final advantage of a schema is that it assigns meaning to the data. That is, it provides a user with a concrete semantics of the document and aids in the specification of meaningful queries over XML data. Although the examples mentioned here just scrape the surface of current applications, they already underscore the importance of schemas accompanying XML data. References are provided in the original paper.

2 Problem setting

Given a collection of XML documents, a schema should be inferred without user intervention and can therefore solely be based on the XML data at hand. Like for any effective inference algorithm, the generated schema should strike a good balance between (1) specialization, i.e., covering all XML documents in the sample in a minimal way; and, (2) generalization, i.e., covering all documents satisfying the target schema but which are not necessarily present in the sample.

In this respect, schema inference problems come in two flavors. First, there is the setting when only little XML data is present, for instance, when XML is returned as answers to queries or Web service requests. In such a case, a schema inference algorithm should balance more towards generalization than to specificity as it is unlikely that a rich class of schemas can be learned from few data instances. In the other scenario, a huge amount of XML data is available, for instance, when the data resides in a native XML databases or is generated in bulk from existing (say relational) data. In this case, there usually is enough information to derive a highly specific schema in a rich class and a learning algorithm should therefore favor specialization over generalization.

We consider the inference of concise Document Type Definitions (DTDs) in both of the above settings rather than XML Schema. The most important reason is that DTD inference has not adequately been addressed yet: existing systems do not perform well when tested on real world or sparse XML data. Secondly, DTD inference is a subcase of XSD inference [3].

As DTDs can be abstracted by context-free grammars with regular expressions (REs) at their right-hand sides, DTD inference reduces to learning of REs from positive example strings. Unfortunately,

¹To appear in Proceedings of the 32nd International Conference on Very Large Data Bases, Seoul, South Korea, September 12–15, 2006

Gold showed that the class of *all* REs cannot be learned from positive data only [2]. As the framework for schema inference from XML data is exactly such that only positive example strings are provided, it is unrealistic to develop inference algorithms for the class of all DTDs. One of the main challenges is therefore to identify subclasses of REs which (1) include the large majority of REs occurring in practical DTDs, (2) which are concise, and (3) which can be learned efficiently from positive data only. We present two such classes:

1. The class of **single occurrence REs (SOREs)**, these are REs in which every element name can occur at most once, e.g., $((b?(a+c))^+d)^+e$ is SORE but $a(a+b)^*$ is not: a occurs twice.
2. The class of **chain regular expressions (CHAREs)** which are those SOREs consisting of a sequence of factors $f_1 \cdots f_n$ where every factor is an expression of the form $(a_1 + \cdots + a_k)$, $(a_1 + \cdots + a_k)?$, $(a_1 + \cdots + a_k)^+$, or, $(a_1 + \cdots + a_k)^*$, where $k \geq 1$ and every a_i is an alphabet symbol. For instance, $a(b+c)^*d^+(e+f)?$ is a CHARE, while $(ab+c)^*$ and $(a^*+b^*)^*$ are not.

These classes certainly satisfy the relevance criteria mentioned above: an examination of the 819 DTDs gathered from the Cover Pages as well as from the web at large, reveals that more than 99% of the REs occurring in practical schema's are CHAREs (and therefore also SOREs) [3]. Furthermore, they are succinct by definition: every element name can occur only once.

3 Approach

We introduce the algorithm iDTD that first infers an automaton from a set of strings and then rewrites the latter into an equivalent SORE when one exists, and into a SORE that is a super-approximation otherwise. The size of the produced regular expression is always linear in the number of different alphabet symbols. As every alphabet symbol needs to occur at least once, a SORE can be seen as the most concise representation.

Furthermore, we introduce the algorithm CRX that derives CHAREs (a subclass of SOREs) directly without going through an automaton representation. Whereas iDTD derives more specific REs, the strength of CRX is its strong generalization ability. As a consequence only very small data sets are necessary to infer an optimal CHARE.

Both iDTD and CRX are validated on real world data, incorporating small and large data sets, and on real world DTDs containing REs in and outside the target classes. Our experiments show that iDTD and CRX outperform existing systems such as XTRACT [1] on this type of data. Some results are shown in Table 1. Further, we asses the strong generalization ability of CRX by establishing on average the minimal number of strings needed to derive optimal REs.

Finally extensions of iDTD and CRX are presented to incrementally compute the inferred RE when new data arrives, how to address noise, and how to deal with numerical predicates. We briefly explain how our inference algorithms can be used to infer simple XSDs.

original	$(a_1a_2?a_3?)?a_4?(a_5 + \cdots + a_{18})^*$	$a_1?a_2a_3?a_4?(a_5^+ + ((a_6 + \cdots + a_{61})^+a_5^*))$
CRX	$a_1?a_2?a_3?a_4?(a_5 + \cdots + a_{18})^*$	$a_1?a_2a_3?a_4?(a_6 + \cdots + a_{61})^*a_5^*$
iDTD	$(a_1a_2?a_3?)?a_4?(a_5 + \cdots + a_{18})^*$	$a_1?a_2a_3?a_4?(a_6 + \cdots + a_{61})^*a_5^*$
XTRACT	an expression of 252 tokens	an expression of 185 tokens
original	$a_1?(a_2a_3?)?(a_4 + \cdots + a_{44})^*a_{45}^+$	$a_1(a_2 + a_3)^*(a_4(a_2 + a_3 + a_5)^*)^*$
CRX	$a_1?a_2?a_3?(a_4 + \cdots + a_{44})^*a_{45}^+$	$a_1(a_2 + a_3 + a_4 + a_5)^*$
iDTD	$a_1?(a_2a_3?)?(a_4 + \cdots + a_{44})^*a_{45}^+$	$a_1((a_2 + a_3 + a_4)^+a_5^*)^*$
XTRACT	an expression of 142 tokens	an expression of 85 tokens

Table 1: Results on real world DTDs: original DTD, inferred DTDs by CRX, iDTD and XTRACT.

References

- [1] Garofalakis, Gionis, Rastogi, Seshadri, Shim. XTRACT: learning document type descriptors from XML document collections. *Data mining and knowledge discovery*, 7:23–56, 2003.
- [2] Gold. Language identification in the limit. *Information and Control*, 10(5):447–474, 1967.
- [3] Martens, Neven, Schwentick, Bex. Expressiveness and Complexity of XML Schema. *ACM TODS*, 31(3), 2006.

Automated Evaluation of Coordination Approaches[†]

Tibor Bosse Mark Hoogendoorn Jan Treur

Vrije Universiteit Amsterdam, Department of Artificial Intelligence
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{tbosse, mhoogen, treur}@cs.vu.nl

1 Introduction

Coordinating processes in a complex software (or agent) system is a nontrivial issue. By a component-based approach to software systems, a divide and conquer strategy can be used to address the various aspects involved. This may lead to a possibly large number of components, which each can be analysed and designed independently. However, a designer may still be left with the problem how all these fragments can be combined into a coherent system. To solve such a problem, many different coordination approaches have been proposed, each having its advantages and drawbacks. Important questions when choosing such a coordination approach are the suitability, correct functioning, and efficiency of the approach for the particular component-based system. This paper presents a methodology to enable a comparison of such factors for the different coordination approaches in a series of test examples.

2 Comparison Methodology

To explore possibilities to address the coordination problem, an evaluation methodology, supported by a software environment, has been created which follows the following steps: (a) a number of *coordination approaches* are selected, (b) a number of *test examples* representing specific software component configurations are chosen, (c) based on each of these coordination approaches a *simulation model* is formally specified, (d) related to the test examples, relevant *requirements* are formally specified in the form of relevant dynamic properties, (e) *simulations* are performed where selected coordination approaches are applied to the chosen test examples, resulting in a number of

[†] The full version of this paper appeared in: *Proceedings of the Eighth International Conference on Coordination Models and Languages, Coordination'06*. Lecture Notes in Computer Science, vol. 4038. Springer Verlag, 2006, pp. 44-62.

simulation traces, and (f) the simulation traces are *evaluated* (automatically) for the specified requirements.

To enable a formal specification of the simulation model, and an evaluation of the resulting traces, the Temporal Trace Language (TTL) [1] is used. TTL first of all allows the specification of executable properties for each of the coordination approaches. After such properties have been specified and test examples are given as input, a simulation engine is used to execute the properties. The execution results in a formal trace with sequences of events that occurred during the simulation of the coordination approach for a particular test example. With such a formal trace as input, and properties on a non-executable level specified in TTL that ought to be fulfilled by the coordination approach (e.g. successfulness, efficiency), a verification tool is used to automatically verify whether these properties are indeed satisfied for the given trace.

3 Results

Based on the approach presented above, the following well-known coordination approaches have been compared: (1) Behaviour networks introduced by Pattie Maes [2]; (2) the pandemonium model [4], and (3) voting [3]. Since the approach also requires test examples to be specified, a choice has been made to use relatively simple workflow patterns. These patterns can be seen as building blocks for more complex patterns occurring in real-life component-based systems. In total, seven such test examples have been used. All approaches turned out effective in finding the solution in all cases. However, none of the approaches is always efficient for all test examples. The behaviour networks and pandemonium approaches perform equally well; they succeed for the “simple” cases and sometimes fail to be efficient for two complicated cases. Surprisingly, the voting approach always finds an efficient solution for one of the complicated cases but fails in a rather trivial case. Finally, the overall methodology turned out to be very useful in comparing the different coordination approaches.

References

- [1] Bosse, T., Jonker, C.M., Meij, L. van der, Sharpanskykh, A, and Treur, J. A Temporal Trace Language for the Formal Analysis of Dynamic Properties. Technical Report, Vrije Universiteit Amsterdam, Dept. of Art. Int., 2006.
- [2] Maes, P. How to do the right thing. *Connection Science* 1(3): pp. 291-323.
- [3] Ordeshook, P. *Game theory and political theory: An Introduction*. Cambridge: Cambridge University Press, 1986.
- [4] Selfridge, O. G. Pandemonium: a paradigm for learning in mechanization of thought processes. In *Proceedings of a Symposium Held at the National Physical Laboratory*, pages 513-526, London, November 1958.

Formal Analysis of Damasio's Theory on Core Consciousness (extended abstract)[†]

Tibor Bosse^a Catholijn M. Jonker^b Jan Treur^a

^a Vrije Universiteit Amsterdam, Dept. of Artificial Intelligence
{tbosse, treur}@cs.vu.nl, <http://www.cs.vu.nl/~{tbosse, treur}>

^b Radboud Universiteit Nijmegen, NICI
C.Jonker@nici.ru.nl, <http://www.nici.ru.nl/~catholj>

1. Introduction

The neurologist Antonio Damasio [2] describes his theory of consciousness in an informal manner, and supports it by a vast amount of evidence from neurological practice. Damasio's theory is described, on the one hand, in terms of the occurrence of certain neural states (or neural patterns), and temporal or causal relationships between them. To formally analyse these relationships, a modelling format is required that is able to express direct temporal or causal dependencies. On the other hand, Damasio gives interpretations of most of these neural states as representations, for example as 'sensory representation', or 'second-order representation'. This requires an analysis of what it means that a neural state is a representation for something. This paper specifically analyses Damasio's notions of 'emotion', 'feeling', and 'core consciousness' or 'feeling a feeling'. This is done by formalising, simulating, and specifying representation of these notions.

2. Simulation

First Damasio's notion of *emotion* is addressed. He describes an *internal emotional state* as a collection of neural dispositions in the brain, which are activated as a reaction on a certain (visual, acoustic, or tactile) stimulus. Once such an internal emotional state occurs, it entails modification of both the body state and the state of other brain regions. By these events, an *external emotional state* is created, which is accessible for external observation.

Next, Damasio's notion of *feeling* is considered. According to Damasio, a feeling emerges when the collection of neural patterns contributing to the emotion lead to mental images. In other words, the organism senses the consequences of

[†] The full version of this paper appeared in: *Proceedings of the Seventh International Conference on Cognitive Modelling, ICCM'06*. Edizioni Goliardiche, 2006, pp. 68-73.

the internal emotional state. Damasio distinguishes two mechanisms by which a feeling can be achieved:

- Via the *body loop*, the internal emotional state leads to a changed state of the body, which is next, after sensing, represented in somatosensory structures of the central nervous system.
- Via the *as if body loop*, the state of the body is not changed. Instead, based on the internal emotional state, a changed representation of the body is created directly in sensory body maps. Consequently, the organism experiences the same feeling as via the body loop: it is ‘as if’ the body had really been changed but it was not.

Finally, Damasio’s notion of (*consciously*) *feeling a feeling* is addressed. This notion is based on the organism detecting that its representation of its own (body) state (the *proto-self*) has been changed by the occurrence of a certain object. The proto-self is described as “a coherent collection of neural patterns which map, moment by moment, the state of the physical structure of the organism”. [2], p. 177. The conscious feeling occurs when the organism detects the transitions between the following moments: (1) the proto-self exists at the inaugural instant, (2) an object comes into sensory representation, (3) the proto-self has been modified by the object.

In the full version of this paper, a simulation model is described of the neural dynamics leading via emotion and feeling to core consciousness, using a temporal language [1]. Moreover, by means of a special software environment, a number of simulation traces have been generated on the basis of the model.

3. Representational Content

In addition to simulation, the notion of representation used by Damasio is formally analysed against different approaches to representational content from the literature on the Philosophy of Mind, e.g., [3]. It is shown that the classical causal/correlational approach to representational content, e.g., [3], pp. 191-193, is inappropriate to describe the notion of representation for core consciousness used by Damasio, as this notion essentially involves more complex temporal relationships describing histories of the organism’s interaction with the world. An alternative approach is shown to be better suited: representational content as relational specification over time and space, cf. [3], pp. 200-202. Criteria for this approach are formalised, and it is shown that the formalisation of Damasio’s notions indeed fit these criteria.

References

- [1] Bosse, T., Jonker, C.M., Meij, L. van der, and Treur, J. LEADSTO: a Language and Environment for Analysis of Dynamics by SimulaTiOn. *Proc. of the Third German Conf. on Multi-Agent System Technologies, MATES'05*. Springer LNAI, vol. 3550, 2005, pp. 165-178.
- [2] Damasio, A. *The Feeling of What Happens: Body, Emotion and the Making of Consciousness*. MIT Press, 2000.
- [3] Kim, J. *Philosophy of Mind*. Westview Press, 1996.

Computationally Intelligent Online Dynamic Vehicle Routing by Explicit Load Prediction in an Evolutionary Algorithm

Peter A.N. Bosman

Han La Poutré

*Centre for Mathematics and Computer Science (CWI),
P.O. Box 94079, 1090 GB Amsterdam, The Netherlands
{Peter.Bosman,Han.La.Poutré}@cwi.nl*

The full version of this paper has been accepted for publication at the *Parallel Problem Solving from Nature (PPSN-IX)* conference.

1 Problem definition

A fleet of vehicles must be controlled to pick up loads at customers and bring these loads to a central depot. As the loads are made available while the vehicles are on route, the problem is dynamic and must be solved online, i.e. as time goes by.

Each customer has a location defined by 2D coordinates. The distance between two locations is taken to be the Euclidean distance. The goal in solving the problem is to deliver as many loads as possible. Each load has a certain announcement time (i.e. the time from which it is available for pickup). Each load must be delivered to the depot within a certain delivery window, starting from the moment of announcement. All vehicles have the same capacity. All loads have the same size. Both the size of the loads and the capacity of the vehicles is integer. Initially, all vehicles are located at the depot.

2 Optimization approach

Evolutionary Algorithms (EAs) are known to be a good problem-solving methodology for dynamic optimization with respect to the efficiency of tracking optima that shift through the search space as time goes by [1]. Often this is a virtue of properly maintaining diversity in the population.

Intuitively, a more efficient routing can be made if it is known beforehand when the loads will become ready for transport. However, this optimal information about future introduction of new loads is not readily available. To still be able to make use of it, the only option is to learn it.

We opt for an approach in which the vehicles keep driving, unless they are at a centrally located depot. The rationale behind this idea is the principled notion that as long as there are loads to be transported, we don't want to have any vehicles waiting around. To move the vehicles as efficiently as possible, we propose to learn the distribution of load announcements at the customers online. We use this information to predict the number of future expected loads at a certain customer at a future point in time. By directly integrating this expected value into the fitness of solutions, i.e. vehicle routes, the EA is able to make informed decisions about anticipated moves (i.e. moves to customers that currently don't have a load ready).

* This work is part of DEAL (Distributed Engine for Advanced Logistics) supported as project EETK01141 under the Dutch EET programme.

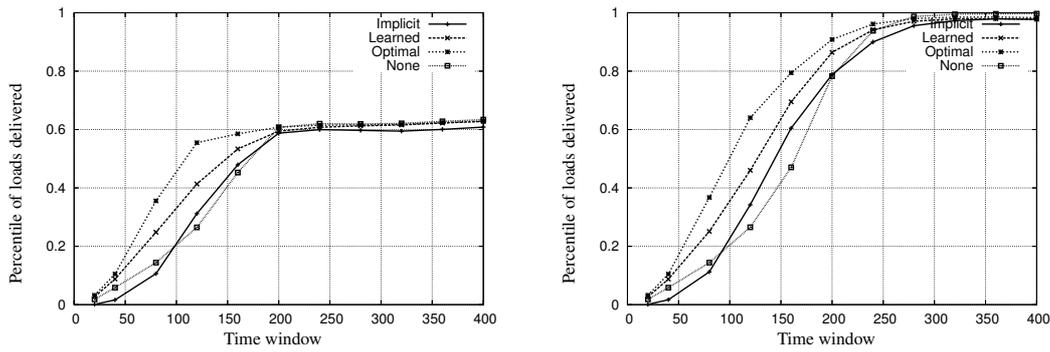


Figure 1: Routing efficiency in percentage of loads delivered as a function of the delivery window for all tested algorithms and a standard deviation of the time spread of the loads of $\sigma^{\text{Loads}} = 40$. Vehicle capacity is 1 in the left graph and 5 in the right graph.

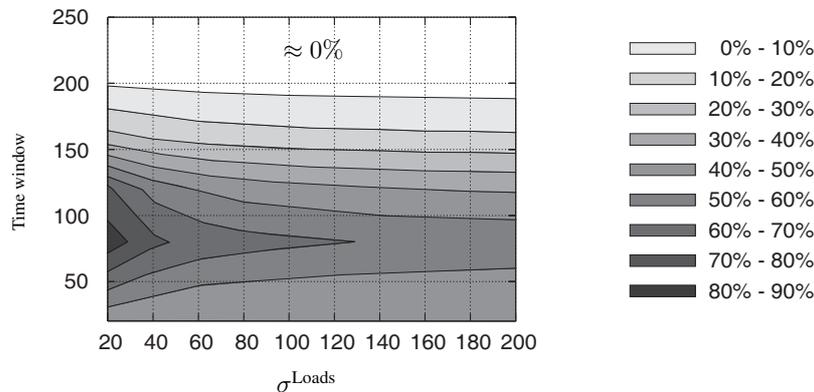


Figure 2: Relative performance increase of our anticipation-by-explicit-learning EA over the non-anticipatory EA for varying delivery window and varying st. dev. of the time spread of the loads.

3 Experiments

We randomly generated 25 problem instances and ran the EA without anticipation, an existing EA that uses anticipation without explicit learning [2], the EA with optimal-information anticipation and the EA with learned-information anticipation 30 times. We varied the standard deviation of the time between subsequent loads, the delivery window and the capacity of the vehicles.

Figures 1 and 2 show routing efficiency and relative performance increase, respectively. Not only do the results show an improvement compared to building routes solely on the basis of currently available loads, our explicit-learning approach also outperforms the existing EA that uses anticipated moves without explicit learning.

4 Conclusions

The use of anticipatory moves to customers where currently no load is available can improve the quality of routing. To facilitate anticipatory moves we have integrated learning techniques to predict future events in an EA. By analyzing carefully what it is that should be predicted, learning this information from past experience and explicitly incorporating it in anticipating the consequences of current decisions, better results can be obtained than when using no prediction or when using an implicit means of prediction.

References

[1] J. Branke. *Evolutionary Optimization in Dynamic Environments*. Kluwer, Norwell, MA, 2001.
 [2] J. I. van Hemert and J. A. La Poutré. Dynamic routing problems with fruitful regions: Models and evolutionary computation. In X. Yao et al., editors, *Parallel Problem Solving from Nature — PPSN VIII*, pages 690–699. Springer-Verlag, 2004.

Enhancing cooperative transport using negotiation of goal direction

Alexandre Campo Shervin Nouyan Mauro Birattari
Roderich Groß Marco Dorigo

IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium

Swarm robotics is a relatively new approach to the coordination of a system composed of a large number of autonomous robots. The coordination among the robots is achieved in a self-organised manner: the collective behaviour of the robots is the result of *local* interactions among robots, and between the robots and the environment. Each single robot typically has limited sensing, acting and computing abilities. The strength of swarm robotics lies in the properties of robustness, adaptivity and scalability of the group [5].

Foraging is a typical task considered in swarm robotics [4]. It can be decomposed in an exploration subtask followed by a transport subtask. The robotic metaphor consists in the search and retrieval of an object. Therefore, the nest is the metaphorical term for the goal and the prey is synonymous of the object to transport. Examples of applications of foraging are toxic waste cleanup, search and rescue, demining and collection of terrain samples.

In our work, we address the case in which all robots completely lose sight of the nest during the exploration subtask of foraging. We assume that the robots have partial knowledge of the goal direction. For instance, they may have perceived the nest earlier and kept track of its approximate direction by means of odometry [2]. Using this approximate knowledge, if several robots attempt to transport a heavy prey in different directions they may fail to move the prey at all. Therefore, we introduce a mechanism to let the robots negotiate the goal direction. In order to meet the general principles of swarm robotics [5], this system is fully distributed and makes use of local communication only.

The mechanism we introduce is strongly inspired by a natural mechanism that has been long studied by biologists. We rely on a particular property of models designed to explain and reproduce the behaviour of fish schools and bird flocks [1, 7]. The models available in the literature are usually composed of three behaviours: an attraction behaviour that makes the individuals stick together, a repulsion behaviour that prevents collisions among individuals, and an orientation behaviour that coordinates the individuals' motion. It is the last of these three behaviours that we transfer and implement in our robots. Informally, the orientation behaviour lets every individual advertise locally its own orientation and update it using the mean orientation of its neighbours. In the robotics experiment, the orientation advertised by the robots at the beginning is their initial estimate of the goal direction. A robot transmits orientation information using LEDs and perceives it thanks to an omnidirectional camera, as seen in Figure 1.

We conducted experiments with a group of four robots called *s-bots*, designed and implemented in the SWARM-BOTS project¹. The robots have to transport a prey moving in a direction about which they have noisy knowledge. We assess quantitatively the performance of the negotiation mechanism implemented with respect to different levels of noise and different control strategies. Furthermore, we define three strategies involving negotiation at different stages of the experiments, and a strategy that does not make use of negotiation at all. These strategies are implemented and their performances are compared using a group of four robots, varying the goal direction and the level of noise in perception.

We identify a strategy in which robots negotiate and move at the same time that enables efficient coordination of motion of the robots. Moreover, this strategy lets the robots improve their knowledge

¹For more details, see the project website: <http://www.swarm-bots.org/>

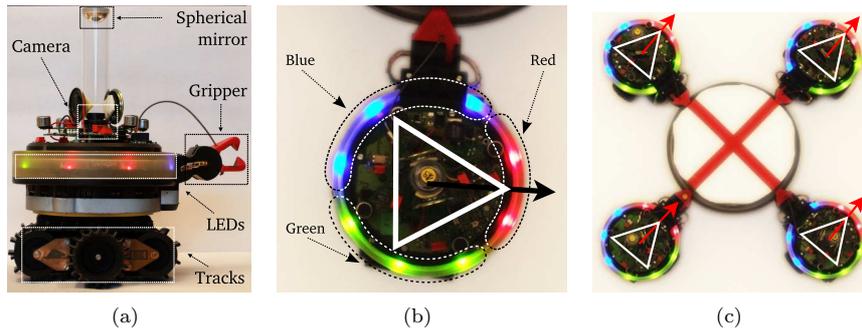


Figure 1: (a) The *s-bot*. (b) An *s-bot* displaying a direction using a triangular LED pattern. (c) Star-like formation of four *s-bots* around the prey as used in the experiment.

of the goal direction. Despite significant noise in the robots' communication, we achieve effective cooperative transport towards the goal and observe that the negotiation of direction entails interesting properties of robustness. The negotiation mechanism we have introduced is not only able to supply a group of robots with collective motion, but also to let each individual improve its own estimate of the goal direction by sharing knowledge with its neighbours. This mechanism may also be used to correct measures of odometry in multi-robot experiments, in a fully distributed fashion. This self-organised negotiation is likely to display properties of scalability besides the robustness shown in this paper. The mechanism makes use of visual communication and has several interesting properties. In particular, it is available when the prey is not in motion, and it is not distracted if the prey moves in irregular steps. Additionally, visual communication opens the door to collective motion with or without transport or physical connections. The topology of the communication network is also likely to be very flexible, allowing the robots to school in very diverse patterns.

We plan to integrate the cooperative transport in a more complex and challenging scenario of foraging, such as for instance the one used by Nouyan *et al.* [6]. This scenario would include an exploration phase preliminary to transport, in which robots lose sight of the nest before finding the prey. In this context, robots have a rough estimate of the direction of the nest by means of odometry. Improvement of this knowledge using negotiation negotiation is a critical feature of the scenario, necessary to let the robots transport the prey efficiently to the nest, even in presence of noisy communications and robot failure.

This work [3] has been presented at the Fifth International Workshop on Ant Colony Optimization and Swarm Intelligence (ANTS 2006).

References

- [1] I. Aoki. A simulation study on the schooling mechanism in fish. *Bulletin of the Japanese Society of Scientific Fisheries*, 48(8):1081–1088, 1982.
- [2] J. Borenstein and L. Feng. Measurement and correction of systematic odometry errors in mobile robots. *IEEE Trans. on Robotics and Automation*, 12(5):869–880, 1996.
- [3] A. Campo, S. Nouyan, M. Birattari, R. Groß, and M. Dorigo. Negotiation of goal direction for cooperative transport. In *Lecture Notes in Computer Science : Ant Colony Optimization and Swarm Intelligence*, volume 4150, pages 191–202. Springer, Berlin / Heidelberg, 2006.
- [4] Y. U. Cao, A. S. Fukunaga, and A. B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–27, 1997.
- [5] M. Dorigo and E. Şahin. Swarm robotics – special issue editorial. *Autonomous Robots*, 17(2–3):111–113, 2004.
- [6] S. Nouyan, R. Groß, M. Bonani, F. Mondada, and M. Dorigo. Group transport along a robot chain in a self-organised robot colony. In *Proc. of the 9th Int. Conf. on Intelligent Autonomous Systems*, pages 433–442. IOS Press, Amsterdam, The Netherlands, 2006.
- [7] C. W. Reynolds. Flocks, herds, and schools: a distributed behavioral model. *Computer Graphics*, 21(4):25–34, 1987.

Two Description-Logic-Based Methods for Auditing Medical Ontologies

Ronald Cornet ^a

Ameen Abu-Hanna ^a

^a *Dept. of Medical Informatics, AMC - Universiteit van Amsterdam*

1 Introduction

Knowledge modelling and auditing in very large ontologies are complicated tasks. The need arises for systematic, reproducible methods to support them. In medicine Description logics (DL) are getting increasing attention as formalisms for representing and reasoning (e.g., GALEN and SNOMED-CT). This paper, abstracted from [1], presents two methods that aim at applying DL-based inference for auditing medical ontologies in order to better understand the potential of DL-based methods.

The methods are based on migration from frame-based representation to DL-based representation. Consider the definition “A parent is a person who has at least 1 child that is a person”. Using frames, a frame `Parent` is defined, having `Person` as superframe. `Parent` has a slot `hasChild` with an AT-LEAST 1 restriction on it. The filler for the `hasChild` slot is `Person`.

Using DL, this can be represented as $\text{Parent} \equiv \text{Person} \sqcap \exists \text{hasChild} \cdot \text{Person}$

2 Detection of Equivalent Definitions

Concepts with equivalent definitions indicate either duplicately defined concepts, or underspecified concepts. For example, in the Foundational Model of Anatomy (FMA) the concepts “Paraganglion” and “Paraaortic body” are defined in exactly the same way. Consequently, either they represent equivalent, hence duplicately defined, concepts (as is considered correct according to some textbooks) or they represent actually different concepts, but the distinction between them is not represented in FMA. It is impossible to represent every characteristic for many concepts, but studying concepts with equivalent definitions can help bringing about better distinctions between definitions.

Our method to detect equivalent definitions comprises the following procedure. In the procedure, frames are expressed as DL statements and assumptions about primitive concepts play a central role in detecting equivalences. All frames that contain a reference to exactly one superframe and have no specified slot-fillers, are assumed primitive and hence represented as primitive concept definitions, $B \sqsubseteq A$. All other frames are represented as concepts with non-primitive definitions with a logical conjunction of superframes and slot-fillers. Slot-fillers are interpreted as existential quantifications of the role values. Primitive concepts are easily recognizable by their definitional form. To detect concepts with equivalent (non-primitive) definitions, the resulting model is classified using a DL reasoner such as RACER ¹. This will result in sets of concepts with equivalent (represented) definitions. For example, both “Paraganglion” and “Paraaortic body” are defined equivalent to: $\text{HomogeneousOrgan} \sqcap \exists \text{systemPartOf} \text{EndocrineSystem}$.

3 Detection of Inconsistencies

The process of migrating a frame-based to an expressive DL-based representation involves the incorporation of closure axioms to concept definitions. We will assume that definitions contain only necessary properties. The following closure axioms are added in order to be able to detect as many potential inconsistencies as

¹www.racersistems.com

possible. (1) All sibling frames are represented as mutually disjoint concepts, (2) Slot-fillers are represented as conjunctions of existential quantifications of the role values and universal quantification of the disjunction of the role values, and (3) The anatomy taxonomy is treated as a partition, i.e. parts are represented as non-overlapping.

As an example, “Paraganglion” is defined as:

```
Paraganglion  $\sqsubseteq$  HomogeneousOrgan  $\sqcap$   $\exists$  systemPartOf EndocrineSystem  
 $\sqcap$   $\forall$  systemPartOf EndocrineSystem
```

Classification of the resulting model may result in a number of unsatisfiable concepts, i.e. concepts that have an inconsistent definition. As the definition is based on an interpretation of the original definition, one needs to determine whether the interpretation or the original definition is incorrect.

4 Results

Application of the first method (equivalence detection) to FMA resulted in an ontology which contained 35425 (=52%) primitive and 33356 (=48%) non-primitive definitions. Classification was performed on the “Organ” taxonomy, having 1167 (=31%) primitive and 2659 (=69%) non-primitive definitions. 28 sets contained concepts that referred to laterality (e.g. Left phrenic nerve, Right phrenic nerve), without explicit reference to laterality in the definition. In general, many of the equivalent concepts contained positional information, e.g. distal/middle/proximal, or posterior/anterior.

The language used for the representation was *EL* (allowing for \sqcap , \exists).

When applying the second method (inconsistency detection) to the “Organ” taxonomy of FMA, the third closure axiom had to be ignored to render the model classifiable. 307 inconsistent concepts were found. 230 inconsistencies originate from two characteristics of “Organ”, respectively “regional part of” Organ system, and “part of” Organ system. In many cases, fillers of these slots are not an organ system.

The language used for the representation was *ALC* (allowing for \sqcap , \sqcup , \neg , \exists and \forall).

5 Discussion and conclusion

A major advantage of the methods described in this paper is that they use readily available reasoning capabilities of DL reasoners. This makes it possible to find concepts with logically equivalent or inconsistent definitions, with relatively little effort. One drawback of the methods is the fact that a large ontology, such as FMA, must be partitioned in order to be classifiable, and the methods need to be applied to the resulting parts. Another drawback (mainly of the second method) is the lack of support for processing the results of the classification, e.g., methods to pinpoint sources of inconsistencies. It must be stressed that the models resulting from our methods are useful for auditing purposes, but the underlying assumptions by which they are generated may not be in correspondence with the actual semantics, hence these models are by no means a replacement for the original ontology. As demonstrated for the FMA, the methods described provide guidance in finding concepts for which the definition can be enhanced, and concepts for which the definition should be revised. In this way, they contribute to the auditing of terminological systems.

References

- [1] R. Cornet and A Abu-Hanna. Two DL-based methods for auditing medical terminological systems. In Charles P. Friedman, Joan S. Ash, and Peter Tarczy-Hornoch, editors, *Proceedings of the 2005 AMIA Annual Symposium, Washington, DC, USA*, pages 166–70. Hanley and Belfus Inc., Philadelphia, PA, USA, 2005.

Efficient Hierarchical Entity Classification Using Conditional Random Fields

Koen Deschacht Marie-Francine Moens

*Interdisciplinary Centre for Law & IT, K.U.Leuven
Tiensestraat 41, 3000 Leuven, Belgium*

1 Introduction

In this paper we describe work carried out in the CLASS project¹. The central objective of this project is to develop advanced learning methods that allow images, video and associated text to be analyzed and structured automatically. We will, for example, learn the correspondence between faces in an image and persons described in surrounding text. The role of the authors in the CLASS project is mainly on information extraction from text. In the first phase of the project we build a classifier for automatic identification and categorization of entities in texts. This classifier extracts entities from text, and assigns a label to these entities chosen from an inventory of possible labels.

2 WordNet

WordNet [1] is a lexical database that organizes nouns, verbs, adjectives and adverbs in synsets. A synset is a collection of words that have a close meaning and that represent an underlying concept. An example of a synset is “person, individual, somebody, mortal, human being”. WordNet defines a number of relations between synsets. For nouns the most important relation is the hypernym/hyponym relation. A noun X is a hypernym of a noun Y if Y is a subtype or instance of X . For example, “bird” is a hypernym of “penguin” (and “penguin” is a hyponym of “bird”). This relation organizes the synsets in a hierarchical tree. We make the assumption that every synset has exactly one hypernym. This is sensible since only a minority of the synsets has two or more hypernyms. In such a case we choose the most common hypernym. We will build a classifier that tags every noun phrase in a sentence with its WordNet synset using Conditional Random Fields.

3 Conditional Random Fields

Conditional random fields (CRFs) [2] is a statistical method based on undirected graphical models. Let X be a variable over input sequences to be labeled and Y a variable over corresponding label sequences. We define $G = (V, E)$ to be an undirected graph such that there is a node $v \in V$ corresponding to each of the random variables representing an element Y_v of Y . If each random variable Y_v obeys the Markov property with respect to G (e.g., in a first order model the transition probability depends only on the neighboring state), then the model (Y, X) is a Conditional Random Field. Although the structure of the graph G may be arbitrary, we limit the discussion here to graph structures in which the nodes corresponding to elements of Y form a simple first-order Markov chain. CRFs can be thought of as a generalization of both the Maximum Entropy Markov model (MEMM) and the Hidden Markov model (HMM).

When using CRFs to build a classifier for the WordNet synsets, we’ve experienced that the computational complexity of both training and labeling a new sentence is an important delimiting factor when using a very big collection of labels. Employing CRFs to learn the 95076 WordNet synsets was not feasible on current hardware.

¹<http://class.inrialpes.fr/>

4 Reducing computational complexity

We reduce the complexity of both labeling and training using CRFs by taking into account the hierarchical tree of nouns in WordNet. We will first explain how we do this when labeling a new sentence. The standard method to label a sentence with CRFs is the Viterbi algorithm which has a computational complexity of $O(TM^2)$, where T is the length of the sentence and M the number of labels. To reduce this computational complexity we select the best labeling in a number of iterations. In the first iteration, we label every noun phrase in a sentence with a synset chosen from the synsets at the top of the hierarchical tree (“physical entity”, “abstract entity” or “thing”). After choosing the best labeling, we refine our choice (choose a hyponym of the previous chosen synset) in subsequent iterations until we arrive at a synset which has no hyponyms. In every iteration we only have to choose from a very small number of labels, thus breaking down the problem of selecting the correct label from a large number of labels in a number of smaller problems. This results in a computation complexity of $O(T \log_q(M)q^2)$ where q is the average number of hyponyms per synset.

To train the parameters of a CRF we employ the forward-backward algorithm. This algorithm has a computational complexity of $O(TM^2NG)$, where N is the number of training examples and G the number of iterations needed for training. We reduce the complexity of training by making the same assumption as during labeling: at every level of the tree, we select the correct label among the q possible hyponym synsets. All other synsets are left out of consideration. Since the number of levels in the tree is (on average) $\log_q M$, we only have to take into account $q \log_q M$ synsets during training, resulting in a complexity of $O(T(q \log_q M)^2 NG)$.

5 Results

We used the Sencor corpus [1] for training. In this corpus, that contains almost 700,000 words, every sentence is noun phrase chunked. The chunks are tagged with their part-of-speech tag and their WordNet synset. We implemented the described method (by adapting the Mallet² package) and trained for approximately 102 hours. Testing on unseen data resulted in an accuracy of 77.82%. We must note that a baseline approach that ignores context completely but simply assigns the most likely sense (according to WordNet) to a given word already achieves an accuracy of 67%. Furthermore, Mihalcea and Moldovan [3], who use the semantic density between words to determine the word sense, achieved an 86.5% accuracy on the same data.

An important weakness of our algorithm is the fact that, to label a sentence, we have to traverse the hierarchy and choose the correct synset at every level. An error at a certain level can not be recovered. In the future, we plan to improve our method by implementing a beam-search, keeping a number of possible synsets at every level.

Acknowledgments

The work reported in this paper was supported by the EU-IST project CLASS (IST-027978).

References

- [1] C. Fellbaum, J. Grabowski, and S. Landes. Performance and confidence in a semantic annotation task. In C. Fellbaum, editor, *WordNet: An Electronic Lexical Database*. The MIT Press, 1998.
- [2] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, 2001.
- [3] R. Mihalcea and D.I. Moldovan. A method for word sense disambiguation of unrestricted text. In *Proceedings of the 37th conference on Association for Computational Linguistics*, pages 152–158. Association for Computational Linguistics Morristown, NJ, USA, 1999.

²<http://mallet.cs.umass.edu>

COORDINATED COMMUNICATION, A DYNAMICAL SYSTEMS PERSPECTIVE

Bart De Vylder

Free University of Brussels

Abstract

Over the past years, several computational models have been introduced to study the coordination of communication between agents. Although these models have given valuable insights into the mechanisms required for letting agents develop a successful communication system, few theoretical results have been obtained which substantiate these findings. In this paper we introduce a theoretical framework which allows us to analyze and compare different existing models in a uniform way.

In general we assume a population of agents which have random, pairwise linguistic interactions and in this way try to develop a shared communication system. In the literature, several models of this kind have been described which differ in the agent architecture (representation of an agent's state, update rules, etc.) and in the interaction style between two agents (often a speaker/hearer setup but still with a variety of possibilities in the type and timing of the information transmitted between two agents).

In order to make abstraction of these differences, we explicitly make a distinction between an agent's externally observable linguistic behavior and its internal state which causes this behavior. This linguistic behavior consists of the way an agent produces words to express different meanings and the way in which he interprets these words. Hence, at a particular point in time, the way an agent influences another agent in an interaction only depends on its linguistic behavior, not on its internal state.

We then treat an agent as a black box and define its *response function*. This function gives the average linguistic behavior an agent exhibits when repeatedly interacting with agents with a fixed linguistic behavior. As each agent is part of the population, the output of this function indicates the direction to which the average linguistic behavior of the population will shift. Therefore of particular interest are the fixed points of this function as they predict no change on average. But what is more, a linear analysis of the response function around its fixed points also reveals the stability of these behaviors when adopted by all agents in the population.

The ability of agents to converge to an optimal communication system relates to whether agents, while traversing their behavioral landscape, are attracted to optimal behaviors and repelled from suboptimal behaviors. We thus argue this can be examined solely based on the response function.

Finally we verify and apply this new framework by analyzing several models described in the literature. In each case, we first calculate the agent response function, analytically if possible, otherwise by estimation. Then we determine the fixed points of this function and analyze their stability. In most cases, our analysis confirmed the convergence results obtained in the publications. In one example however a suboptimal but stable fixed point was found. A simulation indeed showed that a population with an initial average behavior near this fixed point indeed encounters severe difficulties to escape from the suboptimal behavior.

Extending NK Landscapes to the Mixed-Integer Domain

Jeroen Eggermont^b Ernst G.P. Bovenkamp^b Rui Li^a
Michael T.M. Emmerich^a Thomas Bäck^a Jouke Dijkstra^b
Johan H.C. Reiber^b

^a *Natural Computing Group, Leiden University,
P.O. Box 9500, 2300 RA Leiden, The Netherlands*

^b *Division of Image Processing, Department of Radiology C2S,
Leiden University Medical Center,
P.O. Box 9600, 2300 RC Leiden, The Netherlands*

1 Introduction

NK landscapes (NKL), introduced by Stuart Kauffman [2], were devised to explore how epistasis controls the ‘ruggedness’ of an adaptive fitness landscape. Frequently NKL are used as test problem generators for Genetic Algorithms. NKL have two advantages. First, the ruggedness and the degree of interaction between variables of NKL can easily be controlled by two tunable parameters: the number of genes N and the number of epistatic links of each gene to other genes K . Second, for given values of N and K , a large number of NK landscapes can be created at random.

As NKL have not yet been generalized for continuous, nominal discrete, and mixed-integer decision spaces, they cannot be employed as test functions for a large number of practically important problem domains. To overcome this shortcoming, we introduced an extension of the NKL model, *mixed-integer NKL (MI-NKL)* [1], which captures these problem domains. They extend traditional NKL from the binary case to a more general situation, by taking different parameter types (continuous, integer, and nominal discrete) and interactions between them into account.

2 NK Landscapes

Kauffman’s NK Landscapes model defines a family of pseudo-boolean fitness functions $F : \{0, 1\}^N \rightarrow \mathbb{R}^+$ that are generated by a stochastic algorithm. It has two basic components: a structure for gene interaction (using an *epistasis matrix* E), and a way this structure is used to generate a fitness function for all the possible genotypes. The gene interaction structure is created as follows: The genotype’s fitness is the average of N fitness components F_i , $i = 1, \dots, N$. Each gene’s fitness component F_i is determined by its own allele x_i , and also by K alleles at K ($0 \leq K \leq N - 1$) epistatic genes distinct from i . The fitness function reads:

$$F(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N F_i(x_i; x_{i_1}, \dots, x_{i_k}), \quad \mathbf{x} \in \{0, 1\}^N \quad (1)$$

where $\{i_1, \dots, i_k\} \subset \{1, \dots, N\} - \{i\}$.

The computation of $F_i : \{0, 1\}^K \rightarrow [0, 1)$, $i = 1, \dots, N$ is based on a *fitness matrix* F . For any i and for each of the 2^{K+1} bit combinations a random number is drawn independently from a uniform distribution over $[0, 1)$. Accordingly, for the generation of one (binary) NK landscape the setup algorithm has to generate $2^{K+1}N$ independent random numbers. The setup algorithm also creates an epistasis matrix E which for each gene i contains references to its K epistatic genes. Table 1 illustrates the *fitness matrix* and *epistasis matrix* of an NKL.

$E_1[1]$	$E_1[2]$	$E_1[K]$
$E_2[1]$	$E_2[2]$	$E_1[K]$
...	$E_i[j]$
$E_N[1]$	$E_N[2]$	$E_N[K]$

$F_1[0]$	$F_1[1]$	$F_1[2^{K+1}-1]$
$F_2[0]$	$F_2[1]$	$F_2[2^{K+1}-1]$
...	$F_i[j]$
$F_N[0]$	$F_N[1]$	$F_N[2^{K+1}-1]$

Table 1: Epistasis matrix E (left) and fitness matrix F (right)

3 Mixed-Integer NK-Landscapes

As mentioned in the previous section, Kauffman’s NKL model is a stochastic method for generating fitness functions on binary strings. In order to use it as a test model for mixed-integer evolution strategies, we extend it to a more general case such that the fitness value can be computed for different parameter types. Here we consider continuous variables in \mathbb{R} , integer variables in $[z_{min}, z_{max}] \subset \mathbb{Z}$, and nominal discrete values from a finite set of L values. In contrast to the ordinal domain (continuous and integer variables), for the nominal domain no natural order is given.

For the continuous variables, we choose an extension of binary NKL to an N -dimensional hypercube $[0, 1]^N$. Therefore, all continuous variables are normalized between $[0, 1]$. Whenever the continuous variable takes values at the corners of the hypercube, the value of the corresponding binary NKL is returned. For values located in the interior of the hypercube or its delimiting hyperplanes, we employ a *multi-linear interpolation technique* that achieves a continuous interpolation between the function values at the corners. Note that a higher order approach is also possible but we chose a multi-linear approach for simplicity and ease of programming. Moreover, the theory of multi-linear models introduces intuitive notions for the effect of single variables and interaction between multiple variables of potentially different types.

Based on our design, NKL on integer variables can be considered to be a special case of continuous NKL. The integer variables can be normalized as follows: Let $z_{min} \in \mathbb{Z}$ denote the lower bound for an integer variable, and $z_{max} \in \mathbb{Z}$ denote its upper bound. Then, for any $z \in [z_{min}, z_{max}] \subset \mathbb{Z}$ we compute the value of $x = (z - z_{min}) / (z_{max} - z_{min})$ in order to get the corresponding continuous parameter in $[0, 1]$, which can then be used in the continuous version of F to compute the NKL.

To introduce nominal discrete variables in an appropriate manner a more radical change to the NKL model is needed. In this case it is not feasible to use interpolation, as this would imply some inherent neighborhood defined on a single variable’s domain $x_i \in \{d_1^i, \dots, d_L^i\}$, $i = 1, \dots, N$, which, by definition, is not given for the nominal discrete case. Therefore we propose an extension of NKL that takes into account the special characteristics of nominal discrete variables.

Let the domain of each nominal discrete variable x_i , $i = 1, \dots, N$ be defined as a finite set of maximal size $L \geq 2$. Then for the definition of a function on a tuple of $K + 1$ such values we would need a table with L^{K+1} entries. Again, we can assign all fitness values randomly by independently drawing values from a uniform distribution. The size of the sample is upper-bounded by L^{K+1} . For $L = 2$ this corresponds to the binary case.

Initial experiments demonstrate the applicability of the mixed-integer NKL problem generator. The results obtained with a mixed-integer evolution strategy indicate the increase in difficulty for finding the global-optimum as K grows.

Future work will focus on exploring more of the characteristics of the MI-NKL, including its specializations: continuous, integer and discrete NKL.

Acknowledgements: This research is supported by the Netherlands Organisation for Scientific Research (NWO).

References

- [1] R. Li and M.T. Emmerich and J. Eggermont and E.G.P. Bovenkamp and T. Bäck and J. Dijkstra and J.H.C. Reiber: Mixed-Integer NK-Landscapes. Parallel Problem Solving from Nature - PPSN IX, LNCS 4193, Springer, 2006.
- [2] S.A. Kauffman: "The origins of order". Self-organization and selection in evolution. Oxford University Press, New-York, 1993

Optimizing Parameters of Coronary Vessel Image Analysis using Evolution Strategies

Michael T.M. Emmerich^a Rui Li^a Jeroen Eggermont^b
Ernst G.P. Bovenkamp^b Thomas Bäck^a Jouke Dijkstra^b
Johan H.C. Reiber^b

^a *Natural Computing Group, Leiden University,
P.O. Box 9500, 2300 RA Leiden, The Netherlands*

^b *Division of Image Processing, Department of Radiology C2S,
Leiden University Medical Center,
P.O. Box 9600, 2300 RC Leiden, The Netherlands*

1 Introduction

In [1] we compare the ability of Mixed-Integer Evolution Strategies (MI-ES) and standard Evolution Strategies (ES) to find optimal parameter settings for coronary vessel image segmentation problems. MI-ES are special instantiations of standard ES that can solve optimization problems with different objective variable types (continuous, integer, and nominal discrete). To find reasonable algorithm settings for this application, we first tested the Evolution Strategies on comparable, but much less time consuming, artificial test problems. With these settings, the Evolution Strategies were then tried on the real image analysis problem.

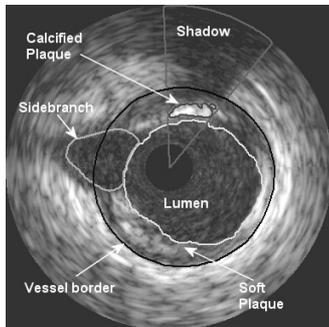


Figure 1. An IntraVascular UltraSound (IVUS) image with detected features. The black circle in the middle is where the catheter was located. The dark area surrounding the catheter is the *lumen*, which is the part of the artery where the blood flows. Above the catheter *calcified plaque* is detected which blocks the ultrasound signal causing a dark *shadow*. Between the vessel border and the lumen there is some soft plaque, which does not block the ultrasound signal. The dark area left of the catheter is a sidebranch.

The practical image processing problem is the detection of the lumen boundary in IntraVascular UltraSound (IVUS) images. IVUS images show the inside of coronary or other arteries and are acquired with an ultrasound catheter positioned inside the vessel. An example of an IVUS image with several detected features can be seen in Figure 1. IVUS images are difficult to interpret, which causes manual segmentation to be highly sensitive to intra- and inter-observer variability.

In addition, manual segmentation of the large number of IVUS images per patient is very time consuming. Therefore an automatic system is needed. However, feature detectors consist of a large number of parameters that are hard to optimize manually. Moreover, these parameters are subject to change when something changes in the image acquisition process, and there are continuous as well as different types of discrete parameters involved. Encouraged by previous work on optimization of image segmentation algorithms in the medical domain, we consider Evolutionary Strategies (ES) as a solution method.

2 Mixed-Integer Evolution Strategies (MI-ES)

From the point of view of numerical analysis, the problem described above can be classified as a black box parameter optimization problem. The evaluation software is represented as an objective function to be minimized. Typically, also constraints are defined.

One of the main reasons why standard optimization methods cannot be effectively used in our application, is that different types of parameters are involved in the lumen feature detector. More precisely, those types can be classified as the following:

- **Continuous variables:** variables that can change gradually in arbitrarily small steps.
- **Integer variables:** variables that can change gradually, but there are smallest steps.
- **Nominal discrete variables:** variables with no reasonable ordering (e.g. red, green, blue).

An ad-hoc approach for using a “standard” continuous ES for integer optimization is to just truncate the continuous values after their mutation to the next integer value. However, a conceptual drawback of this approach is that the step-size might reduce to a value that is too small to generate any improvement. Even worse, in case of nominal discrete values the implicit assumption of a neighborhood could cause the ES to converge to an artificial local optimum.

Mixed-integer evolution strategies (MI-ES) are a special instantiation of evolution strategies that can deal with different parameter types, i.e. can tackle problems as described above. MI-ES can deal simultaneously with continuous, integer, and nominal discrete variables by using a mutation operator that takes into account the different types of decision variables.

3 Experiments and Results

In order to select a favorable variant of the MI-ES for the time-consuming runs on the IVUS image analysis problem, we study the behavior of the MI-ES and standard ES (continuous variables are truncated to integer values) using a new problem generator that produces mixed-integer optimization problems with a scalable degree of ruggedness.

To determine if (MI-)ES can be used as an optimizer for the parameters of feature detectors in the image analysis system we tested ES and MI-ES algorithms on the lumen feature detector. The fitness function used in the experiments is based on the difference between an expert drawn contour and the contour found by the lumen feature detector.

We trained the MI-ES and ES algorithms on 5 datasets of 40 images each, for which we had expert-drawn ground-truth contours. We compared the results of the MI-ES and ES algorithms with each other and with a set of expert default parameters. The MI-ES always produced better or equal results than the default parameter settings chosen by an expert. The standard ES failed to do so in some of the cases. Moreover, on all five data sets the results of the MI-ES were significantly better or equal to those obtained with the standard ES, trained on the same data set.

Visual inspection of the contours detected by the MI-ES and ES algorithms shows that they are very similar to the expert-drawn contours. In some cases the (MI-)ES contours actually seem to follow the lumen boundary more closely. A major difference between the (MI-)ES found contours and the default parameter settings is that the (MI-)ES solutions are more smooth.

We conclude that MI-ES is a promising technique for optimizing image segmentation algorithms and generally performs better than standard ES.

Acknowledgements: This research is supported by the Netherlands Organisation for Scientific Research (NWO) and the Technology Foundation STW.

References

- [1] R. Li and M.T. Emmerich and J. Eggermont and E.G.P. Bovenkamp and T. Bäck and J. Dijkstra and J.H.C. Reiber: Mixed-Integer Optimization of Coronary Vessel Image Analysis using Evolution Strategies. In M. Keijzer et al, editors, Proceedings of the 8th annual conference on Genetic and Evolutionary Computation (GECCO), pages 1645–1652, 2006.

LEARNING EFFECTIVE SURFACE TEXT PATTERNS FOR INFORMATION EXTRACTION

Gijs Geleijnse Jan Korst

Philips Research Laboratories
High Tech Campus 34, 5656 AE Eindhoven, the Netherlands
{gijs.geleijnse,jan.korst}@philips.com

Abstract

When we extract information from the Web using a search engine, we want to formulate effective queries to generate usable search results. We query a combination of a pattern expressing some relation (e.g. *was born in*) with some instance (e.g. *Albert II*) and identify related instances (*Brussels* or *Belgium*) in the excerpts found by the search engine. We present a novel method to identify effective patterns [1]. A pattern is effective, if it links many different instance-pairs in the excerpts found with a search engine. Precision is only one of the criteria to identify the most effective patterns among the candidates found. The learned surface text patterns are applied in an ontology population algorithm.

1 Selecting Effective Patterns

We present an algorithm to learn text patterns expressing relations [1]. We query Google™ with a training set $T_{\mathcal{R}}$ of instance pairs that are \mathcal{R} -related. The training set $T_{\mathcal{R}}$ should be chosen such that the instance-pairs are typical for relation \mathcal{R} . We first discover how relation \mathcal{R} is expressed in natural language texts on the web. For example, one of the patterns that relates instances of *person* and *city* shows to be the pattern ‘*was born in*’. By querying pairs in $T_{\mathcal{R}}$ we thus obtain a list of patterns that link the related instances. From this list we select the most effective ones.

Earlier work [2], describes the identification of *precise* patterns. However, precision is not the only criterion for a pattern to be effective in information extraction from the web.

We have formulated three criteria for selecting effective relation patterns.

1. The patterns should *frequently* occur on the web, to increase the probability of getting any results when querying the pattern in combination with an instance.
2. The pattern should be *precise*. When we query a pattern in combination with an instance, we want to have many search results containing \mathcal{R} -related instances.
3. If relation \mathcal{R} is not functional, the pattern should be *wide-spread*, i.e. among the search results when querying a combination of the pattern and an instance there must be as many distinct \mathcal{R} -related instances as possible.

We formulate scoring functions based on these criteria. In order to reduce the amount of queries, we use the frequency of patterns in the list found as an indication for their occurrence on the web. We therefore only evaluate the most frequently found patterns. We do so, since the evaluation of precision and wide-spreadness requires additional queries per pattern.

2 Using Patterns in Information Extraction from the Web

Having a method to identify effective patterns, we now focus on utilizing these patterns in information extraction from texts found by a search engine. We use an ontology to represent the information extracted.

Suppose we have an ontology O with classes (c_1, c_2, \dots) (e.g. *Actor*) and corresponding instance sets (I_1, I_2, \dots) (e.g. *Jean-Claude van Damme* is an instance of *Actor*). On these classes, relations $\mathcal{R}_{(i,j)}$ are defined, with i and j the index number of the classes. The non-empty sets $T_{(i,j)}$ contain the training set of instance-pairs of the relations $\mathcal{R}_{(i,j)}$. For example the instance pair $(Tom\ Cruis\ e, Top\ Gun)$ can be part of the training set to learn patterns for the relation *acts in*.

We combine instances with learned patterns into queries. From the retrieved excerpts, we extract related instances. In this way, we simultaneously extract instances of some class and instance-pairs of some relation.

- **Step 1:** Select a relation $\mathcal{R}_{(i,j)}$, and an instance v from either I_i or I_j such that there exists at least one pattern expressing $\mathcal{R}_{(i,j)}$ we have not yet queried in combination with v .
- **Step 2:** Construct queries using the patterns with v and send these queries to Google.
- **Step 3:** Extract instances from the excerpts.
- **Step 4:** Add the newly found instances to the corresponding instance set and add the instance-pairs found (thus with v) to $T_{(i,j)}$.
- **Step 5:** If there exists an instance that we can use to formulate new queries, then repeat the procedure.
Else, learn new patterns using the extracted instance-pairs and then repeat the procedure.

Note that instances of class c_x learned using the algorithm applied on relation $\mathcal{R}_{(x,y)}$ can be used as input for the algorithm applied to some relation $\mathcal{R}_{(x,z)}$ to populate the sets I_z and $T_{(x,z)}$.

We recognize instances in the Google excerpts using regular expressions. For example, two or three capitalized words describe person names. We check the extracted terms by querying the term in combination with a class-instance relation pattern. For example, if we are interested in movies and we have extracted *The Godfather*, we check whether the number of hits to the query “*The movie The Godfather is*” exceeds a given threshold.

3 Conclusion

Our main contributions are the following.

1. We have developed a method to effectively access relevant web data using a search engine by querying combinations of learned patterns and instances.
2. We use a double bootstrapping mechanism to extract information from the web. On the one hand we use newly identified instances to retrieve other instances, on the other hand we learn new relation patterns by adding learned instance pairs to the training set.

A first experiment, the identification of hyponym patterns, showed that the patterns identified indeed intuitively reflect the relation considered. Moreover, we have generated a ranked list of hyponym patterns. An experiment with an ontology describing restaurants and their locations (countries) illustrated that a small training set suffices to learn effective patterns and populate an ontology with good precision (80%) and recall (85%) providing the countries where Burger King is located. The algorithm performs well with respect to recall of the instances found. The identification of the instances however is open to improvement, since the additional check does not filter out all falsely identified candidate instances.

References

- [1] Gijs Geleijnse and Jan Korst. Learning effective surface text patterns for information extraction. In *Proceedings of the EACL 2006 workshop on Adaptive Text Extraction and Mining (ATEM 2006)*, pages 1–8, Trento, Italy, April 2006. <http://acl.ldc.upenn.edu/W/W06/W06-2201.pdf>.
- [2] D. Ravichandran and E. Hovy. Learning surface text patterns for a question answering system. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 41–47, Philadelphia, PA, 2002.

Semantic Web Techniques for Multiple Views on Heterogeneous Collections: a Case Study

Marjolein van Gendt Antoine Isaac Lourens van der Meij
Stefan Schlobach

*Vrije Universiteit Amsterdam
Koninklijke Bibliotheek, Den Haag
Max Planck Institute for Psycholinguistics, Nijmegen*

Abstract

Integrated digital access to multiple collections is a prominent issue for many Cultural Heritage institutions. The metadata describing diverse collections must be interoperable, which requires aligning the controlled vocabularies that are used to annotate objects from these collections. In this paper, we present an experiment where we match the vocabularies of two collections by applying the Knowledge Representation techniques established in recent Semantic Web research. We discuss the steps that are required for such matching, namely formalising the initial resources using Semantic Web languages, and running ontology mapping tools on the resulting representations. In addition, we present a prototype that enables the user to browse the two collections using the obtained alignment while still providing her with the original vocabulary structures. This paper is an abbreviated version of a paper accepted at the 10th European Conference on Digital Libraries, ECDL 2006.

1 Introduction

Ontology alignment to facilitate data integration is a prominent issue within the Semantic Web (sw) community. Thesaurus mapping is important in the same fashion for collection integration within the Cultural Heritage domain. The general aim of our project (STITCH¹, funded by NWO, Netherlands Organisation for Scientific Research) is to determine to what extent the current sw techniques can solve heterogeneity issues in the CH sector. Here we report on how we took off-the-shelf, state-of-the-art ontology mappers and evaluated those in a case study concerning Cultural Heritage (CH) thesaurus mapping. In this case study we aligned two thesauri, Iconclass² and ARIA³, to integrate two collections annotated using these vocabularies: the Illuminated Manuscripts collection from the Dutch National Library and the Masterpieces collection from the Rijksmuseum, respectively. Our approach for integrating these collections was first to formalise the vocabularies using SKOS, then to align the thesauri using Falcon [3] and S-Match [1], and finally to visualize the resulting mapping in a faceted browser. A full account of this work can be found in [4].

2 Formalisation

Our case study supplied two controlled vocabularies which needed transformation. For each vocabulary we first performed an analysis of its original structure and its idiomatic elements. This showed that CH thesauri can have very specific features – multiple labels, associative relationships – which could and should be exploited in mapping. Second we constructed a standard representation of the thesauri using SKOS, an RDF vocabulary that is currently being developed within the w3C Semantic Web activity. Because we opted for a standard formalisation, some idiomatic information

¹<http://stitch.cs.vu.nl/index.html>

²<http://www.iconclass.nl>

³http://www.rijksmuseum.nl/aria/aria_catalogs/index?lang=en

from Iconclass was lost in this step, which is an obvious drawback of using generic methods as proposed by the SW community. Finally, we performed application-specific formalisation in order to let tools like reasoning engines and browsers interpret the semantics of the SKOS models.

3 Collection integration

The formalised versions of ARIA and Iconclass were fed into Falcon and S-Match. Falcon is one of the best performing⁴ tools for aligning complex RDFS/OWL ontologies. S-Match has been developed for mapping tree-like structures, and required us to rewrite the SKOS files into a non-standard input format. Mapping thesauri proved to be difficult for both mappers, and the overall results were less than satisfactory. S-Match showed 46% correct mappings for a selected subset of Iconclass (1500 concepts) and the complete ARIA thesaurus (500 concepts). Falcon reached a precision of only 16% and was not scalable regarding complete Iconclass (28.000 concepts, that is matrices of that order of dimension as starting points for Falcon in-memory computation). The main cause for these results is that CH controlled vocabularies have features that make them really different from ontologies. The most prominent differences in our experiment were that some CH thesauri have glosses for describing concepts instead of simple terms and that thesauri structure and semantics are loosely defined compared to full-fledged, formal ontologies.

4 Collection visualization

We implemented a multi-faceted browsing framework to evaluate and explore the results of our mapping effort. This user interface is operational⁵ and provides a way to browse the original CH thesauri and retrieve images from both collections. The design of our browser was inspired by the Flamenco search interface framework [2]. Its implementation uses SWI-Prolog and the Sesame RDF repository⁶ for storage and querying.

5 Conclusion

The main goal of our research was to evaluate to what extent SW techniques can solve heterogeneity issues when integrating multiple CH collections. The general conclusion is positive: in a relatively short time we managed to implement an integrated browsing environment that was built purely on accepted standards for representing data, and which used existing tools for storage, querying and mapping.

In our research we did notice some problems, though. The most important issues are the balance between standardization and information loss and the need for CH thesauri specific adaptation of ontology mapping techniques: it is for example necessary to have alignment tools properly exploiting rich labeling information while still remaining independent from the case of specific thesauri.

References

- [1] Giunchiglia, F., Shvaiko, P., and Yatskevich, M.: Semantic Schema Matching. 13th International Conference on Cooperative Information Systems (CoopIS 2005), Cyprus, 2005.
- [2] Hearst, M., English, J., Sinha, R., Swearingen, K. and Yee, P.: Finding the Flow in Web Site Search. Communications of the ACM, 45 (9), 2002.
- [3] Jian, N., Hu, W., Cheng, G., and Qu, Y.: Falcon-AO: Aligning Ontologies with Falcon. K-CAP Workshop on Integrating Ontologies, Banff, Canada, 2005.
- [4] van Gendt, M., Isaac, A., van der Meij, L., and Schlobach S.: Semantic Web Techniques for Multiple Views on Heterogeneous Collections: a Case Study. In 10th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2006), Alicante, Spain, 2006.

⁴In the OAEI - Ontology Alignment Evaluation Initiative 2005

⁵See <http://stitch.cs.vu.nl/demo.html>

⁶Available at <http://www.openrdf.org>

Identity Management in Agent Systems¹

David R.A. de Groot and Frances M.T. Brazier

IIDS Group, Computer Science Department, Faculty of Sciences, Vrije Universiteit
Amsterdam, de Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
{dra.de.groot, fmt.brazier}@few.vu.nl, <http://www.iids.org/>

Agent technology is a promising and enabling technology in open distributed environments. Agents are autonomous entities that interact with each other and with (web-) services, for which digital identity management (DIDM) is a prerequisite: the rights and obligations of all entities in an agent system need to be secured.

A framework for evaluation of DIDM in agent systems is proposed. Four computational entities in an agent system are distinguished: agent platforms, hosts, agents and services. These four entities are analysed with respect to four aspects of DIDM: representation, confidentiality, integrity and availability.

For each computational entity the following identity information is specified: a) name(s), b) address(es), c) look-up services, d) related principals, e) meta-data, and f) access regulation. *Names* are associated with each entity, e.g., unique identifiers and/or pseudonyms. *Addresses* specify an entity's point(s) of access and message delivery [2]. *Look-up services* provide information about other entities. *Related principals* are roles associated to a computational entity: Admin, Auditor, Creator, Developer, Owner, Publisher and User. These roles are often not explicitly represented. *Meta-data* describes the characteristics of an entity and its functions, e.g. information stored in a look-up service. *Access regulation* information, e.g. access rights.

The four aspects of DIDM of information stored and maintained in an agent system, are comparable to those found in other systems. A *representation* is a creation that is a visual or tangible rendering of someone or something. Representations are useful to store identity information. *Confidentiality* is related to information privacy. Any information that could possibly lead to the identification of a specific entity needs to be protected [3], e.g. user profiles, traces of transactions, and logs of interactions. In agent systems both the entities and their data need to be protected so that "only authorized entities

¹ The full version of this paper appeared in the Proceedings of the First International Workshop on Privacy and Security in Agent-based Collaborative Environments (PSACE), by Foukia, N., Seigneur, J., and Purvis, M. (editors) at the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-06), Future University, Hakodate, Japan, pp. 23-34

can see protected data” [3, 4]. Maintaining *integrity* includes [5]: a) Preventing unauthorized users from making modifications; b) Maintaining internal and external consistency; c) Preventing authorized users from making improper modifications. *Availability* can be defined as “ensuring that information and information processing resources both remain readily accessible to their authorized users” [5]. In agent systems, agents, for example, need to be able to find other agents and services. White- and Yellow page services can serve this purpose.

Two agent system development frameworks, JADE-S and AgentScape, are used to illustrate the potential of this framework with respect to analysis, evaluation, design and development of agent systems. The framework provides a basis for further consideration of issues concerning privacy, anonymity, traceability and accountability.

Acknowledgements

The authors thank Stichting NLnet and the Vrije Universiteit for their support and the Computer Law Institute of the Faculty of Law at the Vrije Universiteit (CLI) for their input. This research is conducted in the context of ALIAS project [6], in particular the project "The Virtual World requires New Bridges" (in Dutch: "De Virtuele Wereld vraagt om Nieuwe Bruggen"). In these projects legal implications of the use of software agents and agent systems are investigated from both a legal and technological perspective.

References

1. Luck, M., McBurney, P., Shehory, O., Willmott S. and the AgentLink Community (2005), "Agent Technology: Computing as Interaction - A Roadmap for Agent-Based Computing", ISBN 085432 845 9, <http://www.agentlink.org>
2. Tanenbaum, A.S. and Van Steen, M. (2002), Distributed Systems: Principles and Paradigms, Prentice Hall, New Jersey.
3. Subirana, B., Bain, M., (2005), "Legal Programming: Designing Legally Compliant RFID and Software Agent Architectures for Retail Processes and Beyond", Springer's Integrated Series in Information Systems, Vol. 4, ISBN: 0-387-23414-4.
4. Pfleeger P. (1997), Security in Computing, international edition Prentice Hall, pp. 158, ISBN 0-13-185794-0
5. Mayfield, T., J. E. Roskos, S. R. Welke and J. M. Boone, (1991) "Integrity in Automated Information Systems". National Computer Security Center (NCSC), TR 79-91
6. <http://www.iids.org/alias>

Eliminating Interdependencies between Issues for Multi-Issue Negotiation¹

Koen Hindriks Catholijn M. Jonker Dmytro Tykhonov

*Radboud University Nijmegen, Montessorilaan 3, Nijmegen, The Netherlands
{k.hindriks, c.jonker, d.tykhonov}@nici.ru.nl*

Negotiation is a process by which a joint decision is made by two or more parties. The parties first express contradictory demands and then move towards agreement by a process of concession making or search for new alternatives. Negotiation is a key capability of human beings operating within a social setting. Within multi-agent systems the importance of negotiation is understood as an important way for agents to achieve their own goals as well as possible. Representing agent's preferences in terms of mathematical formulae expressing relationships between values of issues and the utility of bids allows the development of software support for negotiations. The complexity of these relationships determines the computational complexity of the negotiation process.

Mathematically, the preferences of an agent can be represented as a utility function. The overall utility of a set of independent issues can be computed as a weighted sum of the evaluation values associated with each of the separate issues (cf. also [3]):

$$u(x_1, x_2, \dots, x_n) = w_1 ev_1(x_1) + w_2 ev_2(x_2) + \dots + w_n ev_n(x_n) \quad (1)$$

In equation (1), the (weighted) contribution of each issue to the overall utility only depends on the value associated with that issue and the contribution of a single issue to overall utility can be modeled independently from any other issues. Issue evaluation for independent issues thus can be modeled as a function with exactly the same properties as the utility function associated with the bids that consist of multiple issues: a mapping of issue values on a closed interval [0; 1]. This setup can be used for issue values that are numeric (e.g., price, time) as well as for issue values taken from ordered, discrete sets (e.g., colors, brands).

Multi-issue bid utility functions such as (1) cannot be used to model dependencies between issues. In that case, the contribution to overall utility value associated with one issue may depend on that of another. Dependencies between these issues give rise to a generalization of equation (1) to:

$$\begin{aligned} u(x_1, x_2, \dots, x_n) = \\ = w_1 ev_1(x_1, x_2, \dots, x_n) + w_2 ev_2(x_1, x_2, \dots, x_n) + \dots + w_n ev_n(x_1, x_2, \dots, x_n) \end{aligned} \quad (2)$$

In general, negotiation strategies assume that it is computationally feasible to search for bids $\langle x_1, x_2, \dots, x_n \rangle$ that have a given (target) utility value u , to be able to make a (counter)proposal in the negotiation. According to [1], however, solving this search problem for utility functions with issue dependencies that can only be modelled by equation (2). In that case, the complexity increases exponentially with an increasing number of issues.

One of the better known methods to tackle this problem is to introduce a mediator to which both negotiating agents reveal their respective utility functions and solve the corresponding optimization problem to find a near-optimal outcome for both agents [3]. In that case, e.g. non-linear programming

¹ The full paper: "Eliminating Interdependencies between Issues for Multi-issue Negotiation" is to appear in the *Proceeding of the "Cooperative Information Agents" workshop (CIA 2006)*.

techniques or evolutionary algorithms can be used to approximate an efficient solution. Revealing utility functions is not always acceptable, however, and therefore we propose an alternative way to deal with the computational complexity involved in searching through utility spaces with non-linear issue dependencies. The proposal is to transform the initial utility space before starting the negotiation into a utility space that approximates the original one but does not have any non-linear issue dependencies..

The main feature of the technique proposed here thus is to eliminate interdependencies between issues by approximating a utility function defined by equation (2) by a utility space defined by equation (1). Mathematically, the idea is to “average out” in a specific way the influence of other issues on a particular issue. A weighted averaging method can be used for this purpose that takes as input a utility space with non-linear interdependencies between issues. The weighted averaging technique to compute the approximated space without dependencies consists of the following steps:

1. As a first step, estimate the utility of an expected outcome that is reasonable (given available knowledge).
2. Select a type of weighting function. The selection of a weighting function is based on the amount of uncertainty about the estimated expected outcome in the previous step.
3. Calculate an approximation of the original utility space based on non-linear issue dependencies using the expected outcome and the weighting function determined in the previous step.
4. Perform an analysis of the difference of the original and approximated utility space by means of a χ^2 -function to assess the range of the error for any given utility level.

Finally, the resulting approximated utility space can be used in combination with a particular negotiation strategy for (independent) multi-issue bid spaces.

Experiments between two agents negotiating about two issues have been performed to validate the approach. In this setup agents’ evaluation functions were modelled with 2nd-order polynomials with two variables to model interdependencies between both issues. The ABMP negotiation strategy from [2] has been used in the experimental setup. Negotiations were performed using this strategy for the original utility space and the approximated one. The results show that the negotiation outcomes obtained using the original and approximated utility spaces are close and with a standard deviation that is close to 0.05. The main advantage of the proposed method is that it makes it computationally feasible to use a wider range of negotiation strategies without introducing a mediator into the negotiation process.

References

- [1] Bar-Yam, Y., (1997). *Dynamics of complex systems*. Reading, Mass., Addison-Wesley.
- [2] Jonker, C.M., Treur, J., An Agent Architecture for Multi-Attribute Negotiation. In: B. Nebel (ed.), *Proceedings of the 17th International Joint Conference on AI, IJCAI’01*, 2001, pp. 1195 – 1201.
- [3] Lai, G., Li, C., Sycara, K., and Giampapa, J., (2004,) *Literature Review on Multi-attribute Negotiations*, Technical Report CMU-RI-TR-04-66, Carnegie Mellon University, Robotics Institute.

An Agent Program Logic with Declarative Goals

K.V. Hindriks ^a

J.-J.Ch. Meyer ^b

^a *Radboud University Nijmegen, P.O. Box 9104, 6500 HE Nijmegen*

^b *Utrecht University, P.O. Box 80.089, 3508 TB Utrecht*

Abstract

It has been argued that declarative goals provide for a natural conceptual tool for designing as well as programming agents. This has given rise to various proposals for integrating declarative goals into programming languages. It is not always clear, however, how to establish a precise relation to logical agent theories to reason about such agents. In this paper, we propose an agent programming theory that provides both an agent programming language as well as a corresponding agent program logic to verify agent programs. The agent programming language and agent program logic are developed in parallel to ensure the existence of a mathematically precise relation between the program and logical semantics. To this end, a modal agent logic including the core agent concepts of action, knowledge and goals is introduced. Consecutively, an equivalent state-based semantics is introduced that can be used to relate the logical and operational program semantics. Additionally, it is shown how to integrate goal adoption into the programming theory using the concept of second-order goals.

Agent programming is based on the metaphor of agents that derive their choice of action from their knowledge and goals. Most agent programming languages that have been proposed therefore include programming constructs that operate on agent states consisting of knowledge and (declarative) goals (cf. [1]). Our aim in this paper is to introduce an *agent programming theory* for such agents. A programming theory in our sense consists of two components: a programming language and an associated program logic that can be used to axiomatically verify programs written in that programming language. This requires that we establish a formal relation between an agent programming language and the semantics of the program logic. To this end, we introduce an agent logic with a formal semantics and show how to derive a corresponding agent programming language with an operational semantics from this logic.

As a first step, we introduce a program logic called DEL based on a modal dynamic logic extended with an epistemic operator. The syntax of this logic for formulas $\varphi \in \mathcal{L}_{DEL}$ and programs $\pi \in \Pi_{DEL}$ is given by the following inductive BNF definitions:

$$\begin{aligned}\varphi & ::= \text{any element of } At \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi \mid K\varphi \\ \pi & ::= \text{any element of } Act \mid \varphi? \mid \pi; \pi \mid \pi + \pi \mid \pi^*\end{aligned}$$

The semantics of the logic is provided in the usual way by means of a modal model $\langle W, R, K, V \rangle$ with R_π modeling (complex) actions and K modeling knowledge relations on the set of worlds W and V a propositional valuation function (cf. [2, 3]). A state-based semantics that is equivalent with the modal semantics is constructed next. The states that replace the abstract modal worlds in this semantics are defined as pairs $\langle v, k \rangle$ with v a set of propositional atoms and $k \subset \mathcal{L}_0$ a consistent set of objective propositional formulas, i.e. without occurrences of modal operators. The first component v is called a *world state* and the second component k a *knowledge base*. It is clear that with any world w in a modal DEL model M a unique DEL state $s = \langle v, k \rangle$ can be associated, with $v = \{p \in At \mid M, w \models p\}$ and $k = \{\varphi \in \mathcal{L}_0 \mid M, w \models K\varphi\}$. This state-based semantics can be used to define a structural operational semantics for the associated programming language (cf. [4]).

In a second step, the DEL logic is extended to an agent logic with a motivational operator G to model the agent's goals that is called DAL. The goal operator is modeled as a normal modal operator satisfying the axiom schemas K and D and the modal semantics is extended accordingly. Again we construct an equivalent state-based semantics that is equivalent with the modal semantics.

One of our objectives is to keep this extended state-based semantics simple which is achieved by assuming full introspection of the agent of its goals. Given introspection, the construction of a state-based semantics for the logic DAL can be provided that is a modular extension of the state-based semantics for DEL. By this we mean that it incorporates the state-based semantics for DEL without modifications.

The states in the state-based semantics for DAL are triples $\langle v, k, g \rangle$ that extend DEL states with a goal base g . The goal base cannot be defined as simple as the knowledge base k due to the different logical properties of the G operator. In particular, we do not have $G\varphi \rightarrow GG\varphi$ and cannot eliminate multiple occurrences of the G operator. A natural proposal for a goal base structure is to define it is a linear hierarchy of goals, where goals φ at each level are knowledge propositions (i.e. $\varphi \in \mathcal{L}_k$). Unfortunately, this does not quite work due to the branching structure of the modal semantics for goals (we may have $G(G\varphi \vee G\psi)$ without either $GG\varphi$ or $GG\psi$ being the case in the modal semantics). To remain faithful to the modal semantics we thus would have to introduce a branching structure in goal bases as well. This would not be cognitive plausible, however, and there does not seem to be much use for such structures from an agent programming perspective. Therefore, to exclude such counterintuitive structures and to avoid unnecessary complications in the agent programming theory, we require that goals satisfy the axiom $G^i G\varphi \vee G^i \neg G\varphi$ for $i \geq 0$, with $G^0\varphi = \varphi$ and $G^{(i+1)}\varphi = G(G^i\varphi)$. Goals then can be modeled operationally as hierarchies of goals including first order goals, second order goals, etc.

Finally, we show that the state-based semantics of goals is useful to define a mechanism for goal adoption by using second-order goals. The idea is that second-order goals $GG\varphi$ can be viewed as *dispositions* to adopt goals. An agent that has a first-order goal φ , expressed by $G\varphi$, signifies that the agent wants to change the world to satisfy φ . A second-order goal of the same agent, expressed by $GG\varphi$, however, signifies that the agent in some sense wants to change its current set of (first-order) goals. Interestingly, Sloman et al. in [5] have also pointed to this role of higher-order goals as *motive generators*. Since the state-based semantics is equivalent to the standard modal semantics, moreover, we can also reason about this goal adoption mechanism in the agent logic by introducing a corresponding axiom for program verification.

References

- [1] Rafael H. Bordini, Mehdi Dastani, Jrgen Dix, and Amal El Fallah-Seghrouchni, editors. *Multi-Agent Programming: Languages, Platforms and Applications*. Springer, 2005.
- [2] D. Harel, D. Kozen, and J. Tiuryn. *Dynamic Logic*. MIT Press, 2000.
- [3] J.-J. Ch. Meyer and W. van der Hoek. *Epistemic Logic for AI and Computer Science*. Cambridge: Cambridge University Press, 1995.
- [4] G. Plotkin. A structural approach to operational semantics. Technical report, Aarhus University, Computer Science Department, 1981.
- [5] Aaron Sloman and Monica Croucher. Why robots will have emotions. In *Proceedings of the 7th IJCAI*, pages 197–202, 1981.

Action-Reaction Learning in Multi-Agent Games *

P.J. 't Hoen, S.M. Bohte, and J.A. La Poutré

*CWI, The Netherlands Centre for Mathematics and Computer Science,
hoen@cw.nl,sbohte@cw.nl,hlp@cw.nl*

Abstract

Recent work in Multi-Agent Reinforcement Learning (MARL) has proposed a number of extensions from the single agent setting to the competitive multi-agent domain (see [2] for a State-of-the-art overview). In a Multi-Agent setting, an agent pursuing individual reward is faced by changing opponents that learn. We note however that current algorithms do not sufficiently incorporate this co-learning aspect of multi-agent systems and that agents essentially treat their opponents as moving targets against which they must optimize their own behaviour/policy. We develop the StrOPM algorithm that models opponent policy and, as novel component, learns how an opponent changes as a function of the agent's own actions. The StrOPM algorithm chooses actions to maximize its own reward, given the learned model of adaptation of the opponent. StrOPM learns optimal play for representative matrix games [3]. Additionally, StrOPM learns to cooperate through a learned Tit-For-Tat strategy in self-play on the two-player iterated Prisoner's Dilemma. In contrast, current MARL algorithms will converge to the low level of reward for mutual defection [1] as these algorithms do not take into account that exploitation of opponents will lead to reciprocal exploitation by the opponent. Furthermore, the StrOPM algorithm achieves a high level of learned cooperation for up to nine players for larger scale social dilemma's modelled by the generalised iterated n-player Prisoner's Dilemma [4].

Multi-Agent Reinforcement Learning (MARL) is a thriving research area. Agents through Reinforcement Learning adapt their policies to maximise their own reward in the face of competing agents that strive to maximize their own gains. Sophisticated algorithms and opponent models have been developed that achieve excellent results (see [2] for a State-of-the-art overview). However, a neglected issue in current research is how did the behaviour of the opponents arise? More to the point, is the learned policy of the opponents, at least in part, determined by an agents own past actions? The iterated Prisoner's Dilemma (iPD) is a case in point where ignoring this issue leads to poor outcomes in terms of individual reward.

In the PD two competing agents can each choose from actions 'Cooperate' or 'Defect' (C or D), and the maximum *joint* payoff is achieved when both agents choose to Cooperate. However, an individual agent obtains a larger payoff by playing Defect, provided that the other agent plays Cooperate. From a game-theoretic perspective, playing $\{D, D\}$, the joint action with the lowest joint and individual payoff, is the dominant strategy and a Nash-Equilibrium in the single shot game. In iterated Play of the PD, current MARL algorithms learn to defect [1]. Through modelling of the opponent, the agents estimate that an increase in defection is profitable, and ignore the fact that symmetrical reasoning is the road to mutual defection and loss in possible individual reward. The developed Strategic Opponent Policy Modelling (StrOPM) algorithm [3] is able to overcome this myopic reasoning and achieves cooperation, and higher reward, in self play.

The StrOPM algorithm maintains an estimate of the policy of its opponents. Additionally, and importantly, it also maintains an estimate of how the opponent policy changes as a result of the past actions taken. The StrOPM algorithm is able to learn good optimal policies for representative matrix games as zero-sum games, and coordination games [3]. Importantly, the StrOPM algorithm is also able to coordinate on the iPD without prior knowledge of the game or access to precoded fallback strategies.

In essence, the StrOPM algorithm learns that an increase in defection is retaliated by an increase in defection by the opponent. On the other hand, for a suitable opponent like another StrOPM agent, more cooperation can result in more cooperation and less exploitation by the opponent. In Figure 1, in results from [3], we show the learned level of cooperation for two agents playing the iPD. The agents learn to avoid exploitative defections as these decrease the long term reward through mutual increase in defections. The greedy agents learn to cooperate from a self-interested viewpoint.

Furthermore, in [4] we have shown that StrOPM agents can learn a high level of cooperation for the n-player iterated Prisoner's Dilemma (the nIPD). The agents use an adapted state space that models

*This paper originally appeared as [4].

the aggregate behaviour, and the impact of one's own action, for a set of opponents. The nIPD is a representative model of a social dilemma, a thorny real-life problem. Each agent can benefit from a global good, but also has an incentive to not contribute to the global welfare and exploit the other agents. The n-IPD is the generalised case of the two player iPD. In Figure 1 we show that three agents using the StrOPM algorithm and playing the 3-IPD can achieve a high level of cooperation, and hence individual reward. For four to nine agents, a high level of cooperation is still achieved, whereas classic MARL algorithms would converge to full defection already for the two agent case.

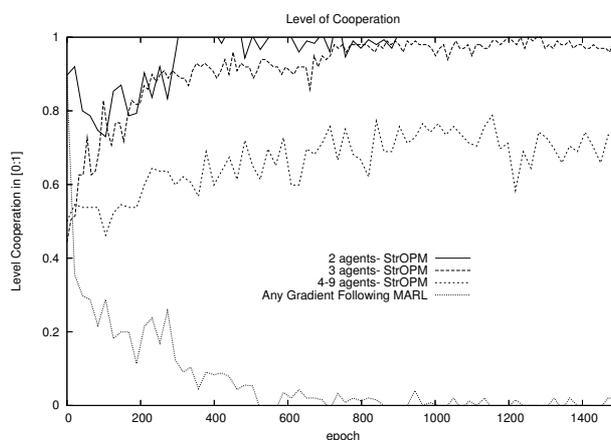


Figure 1: Cooperation for 2, 3, and 4 to 9 players in the nIPD using the StrOPM.

In summary, we have presented the Strategic Opponent Modelling (StrOPM) algorithm that extends the Multi-Agent learning agenda by stressing that good opponent models may not be enough. For example, for problem settings with an n-IPD in the core, better individual reward can be achieved through reasoning not only about the current estimate of the opponent, but also how the opponent will change due to one own actions. Sophisticated agents should not be too greedy in the short term if this reduces higher long term gain. This is common sense for a shop keeper (at least most) who does not exploit a customer now as this will reduce the odds of repeated sales. Such reasoning is introduced to competitive selfish agents through the StrOPM algorithm to induce seemingly cooperative behaviour if beneficial.

References

- [1] J. W. Crandall and M. A. Goodrich. Learning to compete, compromise, and cooperate in repeated general-sum games. In *Proc. 22nd ICML*, 2005.
- [2] R. Powers and Y. Shoham. New criteria and a new algorithm for learning in multi-agent systems. In *NIPS*, 2004.
- [3] P.J. 't Hoen, S.M. Bohte, and J.A. La Poutré. Learning from induced changes in opponent (re)actions in multi-agent games. In *AAMAS'06*, 2006.
- [4] P.J. 't Hoen, S.M. Bohte, and J.A. La Poutré. Strategic farsighted learning in competitive multi-agent games. In *ECAI'06*, 2006.

Modelling Trade and Trust across Cultures (extended abstract)

Gert Jan Hofstede^a Catholijn M. Jonker^b Sebastiaan Meijer^a Tim Verwaart^a

^a Wageningen UR, P.O. Box 9109, 6700 HB Wageningen

^b Radboud Universiteit Nijmegen, Montessorilaan 3, Nijmegen

1. Introduction

Misunderstandings arise in international trade due to difference in cultural background of trade partners. Culture influences trust and the role it plays in trade. Considering that trade always involves working on the relationship with the trade partner, understanding the behaviour of the other is of the essence. This extended abstract summarises the full paper [1]. It advocates the incorporation of culture scripts in the modelling of trade and associated aspects such as trust. As an example, the paper presents a model of the effects of the Cooperation-Performance orientation index of the culture scripts in the models of trustworthiness and acceptability of trade partners in negotiation settings.

2. Culture and Trust

In a trading situation, culture of the trader will manifest itself in four ways. First, culture filters observation. It determines the salience of clues about the acceptability of trade partners and their proposals. Second, culture sets norms for what constitutes an appropriate partner or offer. Third, it sets expectations for the context of the transactions, e.g. enforceability of regulations and possible sanctions. Fourth, it sets norms for the kind of action that is appropriate given the other three, and in particular, the difference between the actual situation and the desired situation.

Culture is that what distinguishes one group of people from another. It is an attribute of a group that manifests itself through the behaviours of its members. Culture at the national level is concerned with five big issues of social life: hierarchy, identity, cooperation-performance orientation, the unknown, and the gratification of needs. Hofstede conceptualizes each of these issues as a bipolar continuum ranging from about 0 to about 100 [2].

This case study models one of Hofstede's dimensions, that of cooperation-orientation versus performance-orientation. The meaning of trust across cultures is related to this dimension. Although singling out one dimension is a deliberate distortion of reality, there is a look-alike real-world case. As figure 1 shows, the Netherlands and the USA differ considerably on the Cooperation-Performance orientation dimension (MAS) and little on the other four.

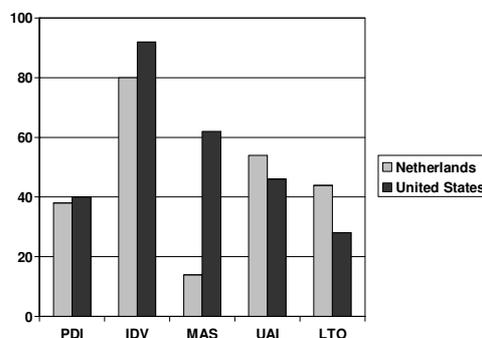


Fig. 1. The cultures of the Netherlands and the USA compared. (data from [2])

3. Modelling Cooperation versus Performance-Orientation

The Trust and Tracing game for human players is a research tool for supply chain studies. The players are assigned roles of producers, middlemen, retailers and consumers in a supply network. The producers receive products some of which are of high, some of low quality. The producers are informed about the quality of their products, but the quality is invisible for the buyers. However, buyers can have the quality tested at the expense of a fee. The game is used as a data gathering tool about the role of reputation and trust in various types of business networks. [3] proposes a research method that combines the human games with multi-agent simulation.

Between human games played with groups of Dutch players and groups of American players, systematic differences were observed. This motivates the modelling of the dimension of performance orientation versus cooperation orientation in the agent model.

The dimension of cooperation versus performance-orientation has its effect on the way people behave in the Trust and Tracing Game. E.G., performance oriented traders are interested in fast trades, with as many goods as possible in one trade. These traders are rather impatient, and if bids are too far off from their profiles, they will walk away quickly. Performance oriented traders will not trust but always trace the goods after buying, since they expect the possibility of deception. They stick to the contracts of the deals, and will deceive their trade partners to the limits of the contracts without any compunction. As a consequence, performance oriented traders see no problems in dealing again with traders that conned him in the past: "It's all in the game".

[1] elaborates on informal descriptions of cooperation-oriented and performance-oriented behaviour, and proposes a set of production rules that model these behaviours for implementation in artificial Trust and Tracing game agents. An example of such a rule:

```
If cultural_script_contains(pc_orientation(F: Real))
    And minimum_utility(M: Real)
    And personality_factor(impatience, I: Real)
Then impatience_factor(F: Real * (I: Real + 0.5))
    And preferred_relative_deal_size(F: Real)
    And allowed_relative_gap_size(F: Real)
    And cut_off_value(M: Real * F: Real);
```

4. Conclusion

Trade situations in the real world can be better understood by taking into account the cultural background of the traders. Concepts like trust and honesty do not mean the same in different cultures, nor do practical aspects such as cheating, negotiation time and good relationships. To be able to model and test agents with culture scripts a comparable data set from real world trade is needed. The Trust and Tracing game provides a conceptualisation of trust in a well-defined laboratory trade environment to compare artificial agent behaviour with.

Future research should test our scripts against data from human games to validate the approach and find plausible values for the parameters in the models. Then the model can be extended to take into account other dimensions of culture as well, increasing validity.

References

- [1] G.J. Hofstede, C.M. Jonker, S. Meijer, T. Verwaart. Modelling Trade and Trust Across Cultures. In: *Trust Management. iTrust 2006, Proceedings*. LNCS 3968:120-134, 2006.
- [2] G.Hofstede and G.J.Hofstede. *Cultures and Organizations: Software of the Mind, Third Millennium Edition*. McGraw-Hill, 2005.
- [3] C.M. Jonker, S. Meijer, D. Tykhonov, T. Verwaart. Multi-agent Model of Trust in a Human Game. In: P. Mathieu, B. Beaufile, O. Brandouy(Eds.). *Artificial Economics: Agent-Based Methods in Finance, Game Theory and Their Applications*. LNMES 564: 91-102, 2006.

Frequent Subgraph Mining in Outerplanar Graphs

Tamás Horváth ^a Jan Ramon ^b Stefan Wrobel ^a

^a *Dept. of Computer Science III, University of Bonn, Germany
Fraunhofer AIS, Sankt Augustin, Germany*

^b *Dept. of Computer Science, Katholieke Universiteit Leuven, Belgium*

Abstract

In recent years there has been an increased interest in frequent pattern discovery in large databases of graph structured objects. While the frequent connected subgraph mining problem for tree datasets can be solved in incremental polynomial time, it becomes intractable for arbitrary graph databases. Existing approaches have therefore resorted to various heuristic strategies and restrictions of the search space, but have not identified a practically relevant tractable graph class beyond trees.

In this paper, we define the class of so called *tenuous outerplanar graphs*, a strict generalization of trees, develop a frequent subgraph mining algorithm for tenuous outerplanar graphs that works in incremental polynomial time, and evaluate the algorithm empirically on the NCI molecular graph dataset.

The discovery of *frequent patterns* in a database is one of the central tasks considered in data mining. In addition to be interesting in their own right, frequent patterns can also be used as features for predictive data mining tasks (see, e.g., [2]). For a long time, work on frequent pattern discovery has concentrated on relatively simple notions of patterns and elements in the database as they are typically used for the discovery of association rules (simple sets of atomic items). In recent years, however, due to the significance of application areas such as the analysis of chemical molecules or graph structures in the WWW, there has been an increased interest in algorithms that can perform frequent pattern discovery in databases of structured objects such as *trees* or arbitrary *graphs*.

While the frequent pattern problem for trees can be solved in *incremental polynomial time*, i.e., in time polynomial in the *combined size* of the input and the set of frequent tree patterns *so far* computed, the frequent pattern problem for graph structured databases in the general case cannot be solved in *output polynomial time*, i.e., in time polynomial in the *combined size* of the input and the set of *all* frequent patterns. Indeed, this can easily be seen by considering a database consisting of two examples: one simple cycle of length n with identical vertex labels and one arbitrary graph of size n . One can then observe that the size of the output is at most $n + 1$ (n paths of length $0 \dots n - 1$ and one cycle), while the problem of deciding whether the cycle is a subgraph of (is isomorphic to a hamiltonian cycle in) the other example is NP-hard.

Existing approaches to frequent pattern discovery for graphs have therefore resorted to various heuristic strategies and restrictions of the search space (see, e.g., [1, 2, 4, 7]), but have not identified a practically relevant tractable graph class beyond trees.

In [3], we define the class of so called *tenuous outerplanar graphs*, which is the class of planar graphs that can be embedded in the plane in such a way that all of its vertices lie on the outer boundary, i.e. can be reached from the outside without crossing any edges, and which have a fixed limit on the number of inside diagonal edges (k -tenuous means at most k diagonals). This class of graphs is a strict generalization of trees, and is motivated by the kinds of graphs actually found in practical applications. In fact, in one of the popular graph mining data sets (the NCI data set), 94.3% of all elements are outerplanar graphs and 99.99% of these are 5-tenuous. At the same time, in the paper, we develop an algorithm for enumerating frequent tenuous outerplanar graph patterns which is guaranteed to work in incremental polynomial time.

Our approach is based on a canonical string representation of outerplanar graphs which may be of interest in itself, and further algorithmic components for mining frequent biconnected outerplanar graphs and candidate generation in an Apriori style algorithm. To map a pattern to graphs in the database, we define a special notion of *block and bridge preserving* (BBP) subgraph isomorphism, which does not allow tree components to be mapped onto cyclic (biconnected) components. This notion is motivated by application and complexity considerations, and we show that it is decidable in polynomial time for outerplanar graphs. Our algorithm is based on a transformation of the outerplanar graphs to a tree and the application of a tree subgraph isomorphism algorithm [6] and a subgraph isomorphism for outerplanar graphs [5].

We note that for trees, which form a special class of outerplanar graphs, BBP subgraph isomorphism is equivalent to subtree isomorphism. Thus, BBP subgraph isomorphism generalizes subtree isomorphism to graphs, but is at the same time more specific than subgraph isomorphism. Since in many applications, subgraph isomorphism is a non-adequate matching operator (e.g., when pattern matching is required to preserve certain type of fragments in molecules), by considering BBP subgraph isomorphism we take an important first step towards the direction of studying the frequent graph mining problem w.r.t. non-standard matching operators as well.

Beside complexity results including both an analysis of our own algorithm and a discussion of the complexity of related graph matching problems, we present also empirical results. These show that in the large NCI database of 250251 molecules our assumptions are satisfied by a vast majority of the instances, and that the favorable theoretical properties of the algorithm and pattern class also translate into efficient practical performance. In particular we observe that, when pattern matching operations share computations, the time needed to evaluate the frequency of patterns does not increase significantly with the pattern size. Even though our current implementation runs in main memory, the algorithm does not need to store results between passes over the database and therefore can run on large databases.

Besides working on optimization of the algorithm, in particular, on improving the time complexity of the coverage testing, it is natural to ask whether the positive result of this paper can be generalized to arbitrary outerplanar graphs. Our algorithm exploits the constant bound on the number of diagonals only in the computation of the set of frequent biconnected graphs in its first step. Therefore, to generalize the result of this paper to arbitrary outerplanar graphs, it is sufficient to consider the following special problem: Given a finite set of biconnected outerplanar graphs and a non-negative integer t , compute the set of t -frequent patterns in w.r.t. BBP subgraph isomorphism. We do not know whether this problem can be solved in output polynomial time.

References

- [1] D. J. Cook and L. B. Holder. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231–255, 1994.
- [2] M. Deshpande, M. Kuramochi, N. Wale, and G. Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1036–1050, 2005.
- [3] Tamas Horvath, Jan Ramon, and Stefan Wrobel. Frequent subgraph mining in outerplanar graphs. In *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, August 2006. To appear.
- [4] Akihiro Inokuchi, Takashi Washio, and Hiroshi Motoda. Complete mining of frequent patterns from graphs: Mining graph data. *Machine Learning*, 50(3):321–354, 2003.
- [5] A. Lingas. Subgraph isomorphism for biconnected outerplanar graphs in cubic time. *Theoretical Computer Science*, 63:295–302, 1989.
- [6] Ron Shamir and Dekel Tsur. Faster subtree isomorphism. *Journal of Algorithms*, 33(2):267–280, November 1992.
- [7] Xifeng Yan and Jiawei Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, Japan, 2002. IEEE Computer Society.

A computational model of intention reading in imitation

Bart Jansen ^a

Tony Belpaeme ^b

^a *Vrije Universiteit Brussel, Pleinlaan 2, 1050 Brussel, BE*

^b *University of Plymouth, A318 Portland Square, PL4 8AA Plymouth, UK*

Original paper: Bart Jansen and Tony Belpaeme, A computational model of intention reading in imitation, *Robotics and Autonomous Systems*, Volume 54, Number 5, pp. 394–402, 2006.

1 Introduction

Imitation in artificial systems involves a number of important aspects, such as extracting the relevant features of the demonstrated behaviour, inverse mapping observations, and executing motor commands. In this article we focus on how an artificial system can infer what the demonstrator intended to do. The model that we propose draws inspiration from developmental psychology and contains three crucial features. One is that the imitating agent needs repeated trials, thus stepping away from the one-shot learning by demonstration paradigm. The second is that the imitating agent needs a learning method in which it keeps track of intentions to reach goals. The third feature is that the model does not require an external measure of equivalence; instead, the demonstrator decides whether an attempted imitation was equivalent to the demonstration or not. We present a computational model and its simulation results, which underpin our theory of goal-directed imitation on an artificial system.

2 The model

In the article we introduce computer simulations of artificial agents which are living in a discrete blocks world they can observe and manipulate. The actions the agents can perform are sequences of any of the four primitive actions which can move a block in any of the four directions. Besides the capability to observe exact block positions, the agents are endowed with the ability to detect some spatial relations over blocks: is a given block positioned at the left of another block, is a block on top of another one, etc. This very limited environment allows the agents to exhibit and imitate intentional behaviour. In the article, the intentions agents can develop express the desire to obtain a blocks world in which some spatial relations hold.

In a typical imitation scenario, a demonstrator possesses a set of goals. The demonstrator performs sequences of actions which will lead to those goals. As the imitator observes those sequences of actions, he tries to identify the goal of this behaviour. The identified goal is imitated by the imitator. Attributing goals to observed behaviour is non trivial, as there is no direct nor unique mapping from actions onto goals. A given sequence of actions can have originated from different goals. Therefore, the enormous search space must be constrained somehow in order to make goal attributing algorithms computationally efficient. The main contribution of the article is that we have identified and motivated four major assumptions which will allow our simulated agents to attribute goals to actions. The four assumptions are the following:

1. All agents perform the same action primitives, or at least the agents know the actions the others can perform.
2. All agents have the ability to categorize the blocks world in terms of a set of spatial relations. Agents are not required to have the same spatial relations. However, in order to be able to learn the same goals, the sets of spatial relations of all agents must have the same expressiveness.
3. The actions the demonstrator performs in order to teach a goal are chosen such that every action which is performed is directly relevant for making the current goal overt to the imitator, i.e. no actions which are irrelevant (and could possibly confuse the imitator) are performed. As a consequence, the plans proposed by the planning algorithm must be optimal. Note that this does not entail that all agents must have the same planner.

4. Agents learn goals by maintaining a repertoire of goals. Attributing intentions to (sequences of) goals is done by categorizing the actions as one of the goals in the repertoire. By consequence, goal level imitation is accomplished by learning and maintaining a repertoire of goals.

The four assumptions are exploited in the interaction pattern which is defined between the demonstrating and the imitating agent (demonstrator and imitator). The interaction is called an *imitation game* and comprises the following six steps:

1. The demonstrator randomly selects a goal g from its repertoire, builds a plan p for it and executes the plan p .
2. The imitator observes this sequence of actions and finds the best matching goal g' from its own repertoire.
3. The imitator builds a plan p' for this goal g' and executes the plan p' .
4. The demonstrator observes this sequence of actions and verifies whether its initial goal g holds in the blocks world arranged by the imitator. If that is the case, the game succeeds; in all other cases, it fails.
5. The demonstrator sends binary feedback to the imitator.
6. The imitator adapts its repertoire.

In the article, every step of the game is explained in depth. Particular attention is paid to the second step, which involves the actual goal categorization process. Crucial in the goal categorization process is the ability of the imitator to take the perspective of the demonstrator by verifying what actions the demonstrator could have performed when pursuing any goal in the imitator's repertoire.

3 Experiments

The paper introduces several measures suited for evaluating the quality of the imitative attempts of the agents. Using those measures, four main experiments are discussed. In every experiment, the same experimental setup is considered: a single demonstrator repeatedly demonstrates goals which are randomly selected from its repertoire. The imitator observes and tries to imitate. In a first experiment, the number of goals in the repertoire of the demonstrator is varied. It is investigated whether the quality of imitation is influenced when more goals are to be transmitted. In a second experiment, the complexity of the environment is varied by changing the size of the blocks world the agents can operate on. The third experiment investigates whether the specificity of the demonstrated goals affect imitation quality. The last experiment is added to investigate how the third assumption (expressiveness of the agents' spatial relations) shapes the intentions which can be learnt from imitation. In that experiment, the set of spatial relations used by the imitator in the categorization of the board state, is made different from the set used by the demonstrator. The experiment clearly shows that as long as the expressiveness of both sets remains the same, imitation is not affected.

In all four experiments, the imitator succeeds in learning the repertoire of goals of the demonstrator and hence, it gradually gets better in imitating its demonstrations.

4 Conclusion

The article proposes an actual implementation of an agents' simulation in which intentional behaviour can be imitated. This is accomplished by maintaining a repertoire of goals. Using information obtained from observing a demonstrator and feedback from ones own imitative attempts, the imitator succeeds in shaping its repertoire of goals such that they become similar to those in the repertoire of the demonstrator. By consequence, the quality of the imitative attempts improves as the demonstration of a limited set of behaviours is repeated.

DECA: Dimension Extracting Coevolutionary Algorithm

Edwin D. de Jong^a Anthony Bucci^b

^a *Utrecht University, PO Box 80.089, 3508 TB Utrecht, The Netherlands*

^b *Brandeis University, 415 South Street, Waltham MA 02454, USA*

Full paper: Proceedings of GECCO 2006.

1 Introduction

Test-based coevolution algorithms use the outcomes of interactions between individuals to perform evaluation and selection. Historically, coevolutionary algorithms have used an aggregate measure of these outcomes during selection, but averaging does not guarantee overall progress over time.

Several recent approaches to coevolution have used archives to develop a stable basis for evaluation and subsequent selection. The criterion used to decide which individuals should reside in the archive depends on the *solution concept* [4] the experimenter wishes to approximate.

The solution concept adopted in this article is the Pareto-optimal equivalence set. This concept specifies a set of candidate solutions that are *non-dominated* in the sense of evolutionary multi-objective optimization. The objectives used in determining this set are given by the set of all possible *tests*, where tests are the co-evolving individuals with which candidate solutions interact. An issue with using all tests as objectives, is that the number of possible tests in a problem is typically very large. The approach taken in this work can provide a way to reduce the set of all possible tests to a smaller set of objectives that provide equivalent evaluation.

Recent work on test-based problems has shown the existence of *underlying objectives* which compress evaluation information into a possibly small set of dimensions [3, 2]. Rather than representing the raw outcome against a single opponent, dimensions provide structure over the outcomes against many opponents. Evaluation information can then be derived from this structure. Theoretically, it has been shown that all test-based problems can be structured in this way [2]. The question then arises whether this theoretical structure can be approximated by the search algorithm, and used simultaneously to evaluate individuals. We develop the Dimension Extracting Coevolutionary Algorithm, DECA, which extracts dimensions and utilizes this information for evaluation and selection.

DECA constitutes a novel approach to evaluation in coevolution. The algorithm constructs and updates a multi-dimensional evaluation function that aims to approximate the true underlying objectives of the problem. If this underlying multi-objective evaluation function can be approximated with sufficient accuracy, then DECA can in principle provide stable evaluation using only a possibly small subset of all tests.

To test the viability of DECA, comparison experiments with several recent reliable coevolutionary methods are performed on COMPARE-ON-ONE, a numbers game problem that is likely to induce over-specialization, and on a game called Tartarus [5, 1].

1.1 Results for Compare-on-one

In this section, we report experimental results with DECA and the comparison methods. Figure 1 compares DECA with LAPCA. The best performance for LAPCA is obtained with 5 or 10 layers. DECA performs comparably to or better than these best instances of LAPCA.

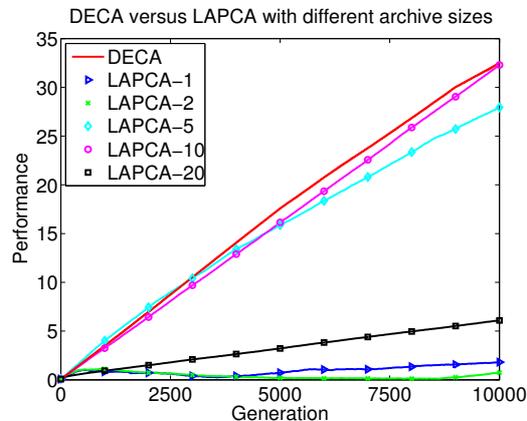


Figure 1: DECA versus LAPCA.

2 Conclusions

We have described DECA, a coevolutionary algorithm that extracts the underlying objectives of a problem and simultaneously uses these to evaluate individuals. The number of tests on the dimensions of an evaluation space can be vastly less than the total number of tests. Due to this reduction, evaluation in DECA can be efficient and yet provide accurate and reliable information.

In contrast to algorithms inspired by biological principles, DECA derives from a theoretical understanding of problem structure. We take the performance of this algorithm as evidence that its theoretical grounding is both sound and useful.

3 Acknowledgements

The authors wish to thank Dan Ashlock for kindly providing his Tartarus code.

References

- [1] Daniel Ashlock, Stephen Willson, and Nicole Leahy. Coevolution and tartarus. In *Proceedings of the 2004 IEEE Congress on Evolutionary Computation*, pages 1618–1624, Portland, Oregon, 20–23 June 2004. IEEE Press.
- [2] Anthony Bucci, Jordan B. Pollack, and Edwin D. De Jong. Automated extraction of problem structure. In *Proceedings of the Genetic and Evolutionary Computation Conference, GECCO-04*, pages 501–512, 2004.
- [3] Edwin D. De Jong and Jordan B. Pollack. Ideal evaluation from coevolution. *Evolutionary Computation*, 12(2):159–192, 2004.
- [4] Sevan G. Ficici. *Solution Concepts in Coevolutionary Algorithms*. PhD thesis, Brandeis University, 2004.
- [5] Astro Teller. The evolution of mental models. In Kenneth E. Kinnear, editor, *Advances in Genetic Programming*, Complex Adaptive Systems, pages 199–220, Cambridge, 1994. MIT Press.

Application of Hybrid Metaheuristics in Sensor Management

Jeroen L. de Jong^a Wilbert L. van Norden^b

^a Thales Research & Technology, P.O. Box 90, 2600 AB Delft

^b CAMS - Force Vision, P.O. Box 10000, 1780 CA Den Helder

1. Introduction: Hybrid metaheuristic scheduling

In the application field of scheduling, computers have gradually been introduced to enhance the search for a good schedule or timetable. This study [1] presents the case of the scheduling problem in current radar systems as applied on naval vessels. These radars can cover a tremendous area where thousands of objects could be visible. Scheduling algorithms that are fast and accurate should get the best out of the sensor.

Numerous schedulers have been developed (and abandoned), all claiming their superiority in at least some special cases. However, *AI* Research moves away from endlessly looking for the Holy Grail. Instead, pragmatic thinking takes over. How can progress be booked by smartly combining the building blocks we already have? Research on *Hybrid Metaheuristics* tries to answer such questions.

2. Case: Sensor management

In a warfare setting, different sensors may be used to obtain a good situational awareness. New Multifunction Radars (MFR) enable the user to track a number of flying objects, perform horizon searches and ‘illuminate’ targets to be destroyed by radar-guided missiles – all at the same time. The capacity, however, is limited. In this research, 3 scheduling approaches to optimize sensor performance are tested in a simulated environment, in comparison to a hybrid approach.

The first approach is a simple straightforward heuristic approach, comparable to the system currently in use on the newest Dutch naval frigates. The second approach is based on fuzzy Lyapunov synthesis (FL). This heuristic boils down to sorting all tasks in different buffers – one for each task type. Whenever a sensor becomes available, the weight for each buffer is calculated. The buffer weight incorporates the priorities of tasks, the (still) available time window for a

task and how much the available sensor is suited to the task. The sensor takes the heaviest buffer and picks the task that adds most to the calculated weight. This heuristic guarantees graceful degradation in times of saturation. However, the risk of sticking to so-called 'local optima' is present: sometimes it is better to sacrifice a big fish, in order to catch an even bigger fish on the longer term.

The third approach is an Evolutionary Algorithm (EA). Swapping Representation, a specially developed chromosome mapping, is used. This mapping overcomes typical order-related problems occurring in popular mappings like Partially Matched Crossover and Ordinal Representation.

In the hybrid setting, the best performing approaches (FL & EA) are combined. This is straightforwardly done by translating solutions brought up by FL into the chromosome mapping, subsequently injecting this solution in the 'current' population of the running EA. As the EA is set in an elitist approach, the best chromosome is always preserved over the next generations. The EA uses the 'ideas' brought in by the other heuristic to build upon.

3. Results

The schedulers are evaluated using a difficult test case in which the system is brought into saturation. As it appears, both Fuzzy Lyapunov and Evolutionary Algorithms perform quite well on their own. They outperform the simple heuristic approach by far. Combining the strength of the straightforward, good performing FL algorithm with EA led to a hybrid scheduler that, as expected, combines the strength of both constituents and yields the best overall performing scheduler.

Interestingly, comparison of the approaches shows that the hybrid scheduler sometimes scores a little worse than one of its constituents. This is due to the fact that the EA which is also present in the hybrid scheduler can better look in the future: sometimes it is advantageous to sacrifice a high-scoring episode on the short term, to gain an even bigger reward in the future. Indeed, detailed observation shows that less good values for the hybrid scheduler are always followed by a higher reward shortly afterwards.

In order to operate within military environments, being able to give performance guarantees is important. This study shows how hybrid approaches can benefit from the good performance of non-deterministic heuristics, while still delivering at least the same performance guarantees of the best deterministic algorithm in the approach.

References

- [1] J.L. de Jong, W.L. van Norden: "*Application of Hybrid Metaheuristics in Sensor Management*", Proceedings COGIS conference, Paris, 2006.

DESIGN OPERATORS TO SUPPORT ORGANIZATIONAL DESIGN*

Catholijn M. Jonker^a Alexei Sharpanskykh^b Jan Treur^b
pInar Yolum^c

^a Radboud Universiteit Nijmegen, Montessorilaan 3, 6525 HR Nijmegen

^b Vrije Universiteit Amsterdam, De Boelelaan 1081a, 1081 HV Amsterdam

^c Bogazici University, TR-34342 Bebek, Istanbul, Turkey

Organizations play a key role in the modern society. To a large extent vitality and productivity of an organization situated in the environment depend on kinds of structure and behaviour of the organization, and how they conform to the environmental conditions. Organizational structures are studied in social sciences, where organizational design is a special topic. Galbraith stated in [2] that organization design "is conceived to be a decision process to bring about a coherence between the goals or purposes for which the organization exists, the patterns of division of labor and interunit coordination and the people who will do the work". In literature, a range of theories and guidelines concerning the design of organizations are present [2, 4]. In particular, Minzberg described in [4] guidelines applicable mostly for designing hierarchical organizations that function in a relatively stable environment. However, despite the abundance of organizational design theories no general principles applicable to organizational design in all times and places can be identified. Moreover, theories are typically informal and not detailed enough for immediate formalisation, which is required for the development of software tools for organization design and analysis.

The overall aim of this research is to create a software support environment for organization designers with operational automated tools for creating, analyzing, and revising organizations. As a first step a formal representation of an organization model as a design object description has been developed, see [3], that is based on predicate logic. Elaborated descriptions of a formal organization modelling approach and of analysis techniques based on the introduced organization model are given in [1], whereas this paper focuses on the process of formal organization design.

A design process can be described by a sequence of design operators to transform the current design object description into the next one. The design operators introduced in the paper are divided into specialized groups (e.g., operators for roles, links and groups) and formalized using the predicate logic. Examples of simple design operators are adding or deleting an element of a design object description (e.g., operators for creating and modifying roles and different types of links, operators for composing and modifying

* The full version of this paper will appear in the Proceedings of the 2nd International Conference on Design Computing and Cognition (DCC'06), 2006, see [3].

groups). More sophisticated design operators involve, for example, the introduction of role refinement, see Table 1.

TABLE 1. Sample complex operators for creating and manipulating organizations

NAME	PATTERN FOR	DESCRIPTION
Role refinement	Role Retraction. Interlevel link deletion*. Interaction link deletion*. Interaction dynamic property addition*. Interlevel link addition*. Interaction link introduction*. Role dynamic property addition*. Role introduction*	Divides a role into several roles such that the role properties of the first role are distributed over the newer roles
Role join	Role Retraction*. Interlevel link deletion*. Interaction link deletion*. Interaction dynamic property addition*. Interlevel link addition*. Interaction link introduction*. Role dynamic property addition*. Role introduction	Joins several roles into a single role

The symbol * denotes that an operator can be applied zero, one or multiple times.

The design operators can be combined into composite operators that can serve as design patterns. Furthermore, the identified set of operators is independent of any organization theory or sociological methodology: they can be used for formalizing design principles from different theories. Finally, a designer can choose any sequence of operators for creating designs of organizations of most types (e.g., functional and organic [2]). The operators offer both top-down refinements, as well as bottom-up grouping options.

The formal representations of the organization model and the design operators provide a solid basis for the development of a software environment supporting interactive organization design processes. A prototype implementation for such a tool is demonstrated by an example in [3].

References

- [1] E. Broek, C.M. Jonker, A. Sharpanskykh, J. Treur, and P. Yolum. Formal Modeling and Analysis of Organizations. In: Boissier et al. (eds.) *Coordination, Organization, Institutions and Norms in Agent Systems I, Proceedings of the ANIREM'05/OOOP'05*, LNAI 3913, Springer Verlag, 2006, pp.18-34.
- [2] J. Galbraith. *Organization design*. Addison-Wesley Publishing Company, London Amsterdam Sydney, 1978.
- [3] C.M. Jonker, A. Sharpanskykh, J. Treur, and P. Yolum. Design Operators to Support Organizational Design. In: J. Gero (ed.), *Proceedings of the Second International Conference on Design Computing and Cognition, DCC'06*. Springer Verlag, 2006, pp. 203-222.
- [4] H. Mintzberg. *Structure in Fives: Designing Effective organizations*. Prentice-Hall, NJ, 1993.

Learning Biological Interactions from Medline Abstracts

Sophia Katrenko ^a M. Scott Marshall ^b Marco Roos ^b Pieter Adriaans ^a

^a *Human-Computer Studies Lab, Informatics Institute, University of Amsterdam,
Kruislaan 419, 1098VA Amsterdam, The Netherlands*

^b *Integrative Bioinformatics Unit, University of Amsterdam,
Kruislaan 318, 1098SM Amsterdam, The Netherlands*

1 Introduction

Extracting interactions from texts is an active field of research in the biomedical domain ([3], for a recent review). Interactions between proteins and genes are often considered essential in the description of biomolecular phenomena, and networks of (protein) interactions are considered as an entre for a Systems Biology approach [5]. Interaction networks extracted from literature [2] complement interaction data obtained from high-throughput laboratory experiments [7]. The approach we describe in this paper falls into a class that explores the application of semantic models [8, 4].

2 Approach

In this paper, we address the problem of learning gene-protein interactions within the Genic Interaction Extraction Challenge. The aim of the challenge is to find such interactions given two data sets (the training and test data consisting of three types of interactions, *action*, *binding* and *regulon families*). The data contains the sentences extracted from the Medline collection; there is therefore no discourse analysis possible. It does, however, include two subsets which are distinguished based on the linguistic notion of co-referentiality. It is assumed that sentences including ellipsis or co-references are more difficult to analyze. An example below includes two interactions, *interaction(SigK,ykvP)* and *interaction(GerE,ykvP)*.

- (1) Both **SigK** and **GerE** were essential for **ykvP** expression, and this gene was transcribed from T5 of sporulation.

Our solution follows the close world assumption, in what we consider all combinations of genes and proteins which have not been annotated as true positives to be the negative instances. Our system combines several elements: 1) a domain theory about the interaction between language, semantics and syntax [1], 2) a biological ontology identifying amongst other things biomolecular entities and directed interaction verbs in the lexicon, 3) the notion of lexical-semantic-syntactic unification, 4) the notion of partial unification of lexical-semantic-syntactic trees and 5) the application of the standard RIPPER algorithm to the unification results.

The domain theory [1] we use provides a framework for syntactic and semantic language learning under the assumption of compositionality. The data sets we use have already been syntactically analyzed. In addition to this, to cope with the data sparseness, we developed a limited domain-specific ontology. It includes such concepts, as BioMolecularEntity, BiologicalProcess, etc. We have also identified verbs expressing relations between genes and proteins (*DirectedActionVerb*). The use of ontology makes it possible to assign semantic labels to the words in each sentence. Together with the syntactic information, it leads to the notion of lexical-semantic-syntactic unification. We have carried out unification of syntactically analyzed sentences

enriched with the semantic information on several levels. These include a root of a dependency tree, ancestors and descendants of the possible arguments of a relation and some others. Finally, we used a rule learner RIPPER [6] to acquire classification rules based on the unification scores.

3 Experiments

Using this approach on the very limited training and test data from the Challenge we show results that are promising. Our method observes a clear separation between domain-independent and domain-specific components. It can therefore easily be extended to other domains. Our analysis also shows that despite a small training set, the method we proposed is able to cover different interaction types (Table 1). Although the type *regulon families* occurred in the training set only twice, we identified interactions of this type in the test data. Moreover, our method works equally well on the sentences with co-references and without them. 10-fold cross-validation on the training set yields better results than testing on the additional data set provided by the organizers of the Challenge. We assume it is due to the fact that the test data has been gathered in a slightly different way than the training data. For instance, the test set includes sentences with the negative interactions, whereas the training data is missing them.

Table 1: Distribution of the interactions found in the test set

Co-references	Type of interaction		
	Action	Binding	Regulon families
Yes	8	2	1
No	5	5	1
Total	13	7	2

4 Discussion

In our future research, we plan to investigate how additional semantic information influences learning relations from the biomedical corpora. In particular, it would be useful to test our approach using information from UMLS which provides information on types of biomolecular entities. We also plan to investigate indirect relations, such as transitive and collective interactions.

References

- [1] P.W. Adriaans. *Language Learning from a Categorical Perspective*. 1992.
- [2] H. Chen and B. M. Sharp. Content-rich biological network constructed by mining pubmed abstracts. *BMC Bioinformatics* 5: 147, 2004.
- [3] M. Krallinger, R. A. Erhardt, and A. Valencia. Text-mining approaches in molecular biology and biomedicine. *Drug Discov Today* 10, pages 439–445, 2005.
- [4] H. M. Muller, E. E. Kenny, and P. W. Sternberg. Textpresso: an ontology-based information retrieval and extraction system for biological literature. *PLoS Biol* 2: e309, 2004.
- [5] P. Uetz and Jr. R. L. Finley. From protein networks to biological systems. *FEBS Lett* 579, pages 1821–1827, 2005.
- [6] I. H. Witten and E. Frank. *Data Mining: Practical machine learning tools with Java implementations*. San Francisco: Morgan Kaufmann, 2000.
- [7] I. Xenarios and D. Eisenberg. Protein interaction databases. *Current Opinion in Biotechnology* 12, pages 334–339, 2001.
- [8] M. D. Yandell and W. H. Majoros. Genomics and natural language processing. *Nat Rev Genet* 3, pages 601–610, 2002.

A Versatile Approach to Combining Trust Values for Making Binary Decisions¹

Tomas Klos^a

Han La Poutré^{a,b}

^a *Center for Mathematics and Computer Science (CWI), Amsterdam*

^b *Faculty of Technology Management, Technical University Eindhoven*

1 Introduction

In open multi-agent systems such as grid-based virtual organizations [1], agents typically need to rely on others for the delivery of information or resources or for the execution of tasks. Since trustworthiness can not be taken for granted, however, an agent needs to build up a measure of her trust in other agents in her environment, and to update it on the basis of her experiences with those other agents.

Many different computational trust models have been proposed in the literature, based on a wide variety of techniques (see, e.g., [2] for a review of computational trust models in multi-agent systems). In the current paper, we let the agents use Bayes' rule to update beliefs about other agents' capabilities. Once an agent has established a trust or reputation value for other agents, resources, services, etc., the agent needs to act *on the basis of* those values. This typically involves estimating the value of a certain, often binary, random variable. If different agents are providing conflicting information about this value, and some are trusted (highly) and some are not, then how should one combine these trust values with the information provided? Which of the agents are more important, and how should conflicting claims be weighted? We investigate the relative effectiveness of a variety of methods for combining trust values.

2 Establishing and Combining Trust Values

For establishing trust values, we let agent i assume that each other agent j 's behavior ($j = 1, \dots, J$) consists of a sequence of Bernoulli trials with 'success' and 'failure' as possible outcomes, and is governed by an agent-specific parameter θ_j : the 'bias,' or probability of success. Agent i 's trust in agent j refers to i 's beliefs about the value of θ_j , which i updates using Bayes' rule. When i uses the Beta(a, b) distribution as a (conjugate) prior distribution for θ_j , then the posterior distribution given our binomial likelihood is also a Beta distribution. Trust is then usually taken as the expected value of this posterior Beta distribution which is easily expressed using its parameters a and b as $E[\theta_j] = \frac{a}{a+b}$. Updating is then simply accomplished by taking $a = u + 1$ and $b = v + 1$, where u and v are the previous positive and negative experiences i has had with j , and where the values of 1 are the parameters of the Beta(1, 1) distribution, the uniform prior. Some example Beta distributions with different numbers of positive and negative experiences are shown in Fig. 1(a), with the corresponding expected values depicted using vertical lines.

We propose 3 methods for combining trust values when the observations of the value $a \in \{0, 1\}$ of a binary variable A , by all J observers in the collection \mathcal{J} need to be combined.

majority (m) This method lets agent i simply report the value for A which is observed by the majority of agents in \mathcal{J} . (We only allow odd values for J .)

¹This paper was originally published in Ketil Stølen, William H. Winsborough, Fabio Martinelli, and Fabio Massacci, editors, *Proceedings 4th International Conference on Trust Management, May 16-19 2006, Pisa, Italy*, volume 3986 of Lecture Notes in Computer Science. Springer, 2006.

evidence (e) Agent i adds her positive and negative experiences across all agents claiming each of the values $a \in \{0, 1\}$ and estimates the expected capability in each of the 2 groups of agents.

$$E(\theta_{\mathcal{J}_a}) = \frac{\sum_{j \in \mathcal{J}_a} \text{positive}_i^j + 1}{\sum_{j \in \mathcal{J}_a} \text{positive}_i^j + \sum_{j \in \mathcal{J}_a} \text{negative}_i^j + 2}$$

She chooses the value for a which maximizes $E(\theta_{\mathcal{J}_a})$.

likelihood (l) Using this method, assuming each value in turn to be correct, agent i calculates the joint probability of the observations, which is the same as the likelihood of each of the two values for a , given the observations. For each value of $a \in \{0, 1\}$ (where $\bar{a} = 1 - a$), the likelihood of it being the true value of A is equal to

$$L(A = a | \text{observations}) = p(\text{observations} | A = a) = \prod_{j \in \mathcal{J}_a} \text{trust}_i^j \prod_{j \in \mathcal{J}_{\bar{a}}} (1 - \text{trust}_i^j)$$

where \mathcal{J}_b (for $b \in \{a, \bar{a}\}$) is the subset of \mathcal{J} claiming $A = b$. Agent i reports the value $b \in \{a, \bar{a}\}$ which maximizes the likelihood $L(A = b | \text{observations})$.

3 Results

Figure 1(b) shows a sample of our results. In this case, all $J = 7$ observers have the same probability of

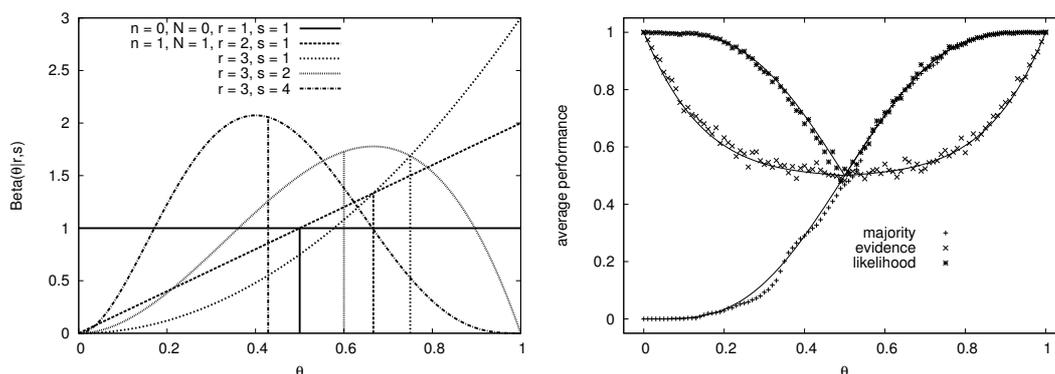


Figure 1: 1(a): example Beta distributions; 1(b): performance of the methods.

success, θ_j , shown on the x -axis. The graph shows the performance at time $t = 1000$ of agent i using each of the 3 methods, averaged over 1000 replications of each experiment (the crosses and plusses). In this simple case, each method's expected performance can be calculated analytically (the solid lines). More interesting results arise when different values for θ_j are distributed less evenly across the different observers. The various methods show differential sensitivity to such changes, but the likelihood method is typically optimal.

References

- [1] Foster, I., Jennings, N.R., Kesselman, C.: Brain meets brawn: Why grid and agents need each other. In: Proc. AAMAS 2004, ACM (2004) 8–15
- [2] Sabater-Mir, J., Sierra, C.: Review on computational trust and reputation models. AI Review **24** (2005) 33–60

MULTIAGENT CONTROL OF ELECTRICITY DEMAND AND SUPPLY

Koen Kok ^{ab} Cor Warmer ^a René Kamphuis ^a

^a *Energy research Center of the Netherlands (ECN), Intelligent Energy Grids,
P.O. Box 1, Petten. Email: (j.kok, warmer, kamphuis)@ecn.nl*

^b *Free University Amsterdam, Faculty of Sciences, Business Informatics Section.*

This *compressed contribution* to the BNAIC 2006 is an adapted version of a short article in IEEE Intelligent Systems [4]. This article was part of a compilation of the best industry papers of AAMAS 2005 held in Utrecht [5]. The original awarded AAMAS paper is [3].

Electricity Infrastructure Evolution. *Distributed generation* (DG) of electricity is providing an increasing part of the worldwide energy supply. DG consists of different sources of electric power connected to the distribution network or to a customer site. This approach is distinct from the traditional central-plant model for electricity generation and delivery. Examples of DG are photovoltaic solar systems, small and medium-scale wind turbine farms, and the *combined generation of heat and power* (CHP).

When the share of DG increases in a geographical area, clustered control of DG by common ICT (information and communication technology) systems can add value. As a result, distribution networks are expected to evolve from a hierarchically controlled structure into a network of networks, in which a vast number of system parts communicate with and influence each other. The number of components actively involved in coordination will be huge. Centralized control of such a complex system will reach the limits of scalability and communication overhead.

A key technology for solving this problem is market-based control. In market-based control, many control agents competitively negotiate and trade on an electronic market to optimally achieve their local control action goals. Use of market-based control in the electricity infrastructure opens the possibility for distributed coordination in addition to the existing central coordination.

The PowerMatcher. The PowerMatcher method provides market-based control for *supply-and-demand matching* (SDM) in electricity networks with a high share of DG. It is based partly on earlier research by Fredrik Ygge and Hans Akkermans [6], Hans Akkermans, Jos Schreinemakers, and Koen Kok [1] and Per Carlsson [2]. In this method, a control agent represents each device. The agent tries to operate the device process in an economically optimal way, within the process's constraints. The agents negotiate their electricity consumption or production on an electronic exchange market. The resulting market price determines the power volume allocated to each device.

From the viewpoint of controllability, devices that produce or consume electricity fall into six classes, each having a specific agent strategy. We look at three in this article. The first class are non-controllable devices, either being generators (e.g. wind energy systems) or energy consumers (like audio and video devices). The second class is shiftable-operation devices, which must run for a certain amount of time regardless of the exact moment and thus are shiftable in time. An example of such a device is a ventilation system in a utility building that needs to run for 20 minutes each hour. The third class comprises devices operating thermal processes like heating, cooling or freezing. Here, operational flexibility is determined by the allowed temperature band.

Local agents' self-interested behavior causes electricity consumption to shift toward moments of low electricity prices and causes production to shift toward moments of high prices. So, SDM emerges on the global-system level.

A Simulation. To investigate distributed SDM's impact for a residential area, we simulated a cluster of 40 houses, all connected to the same segment of a low-voltage distribution network.

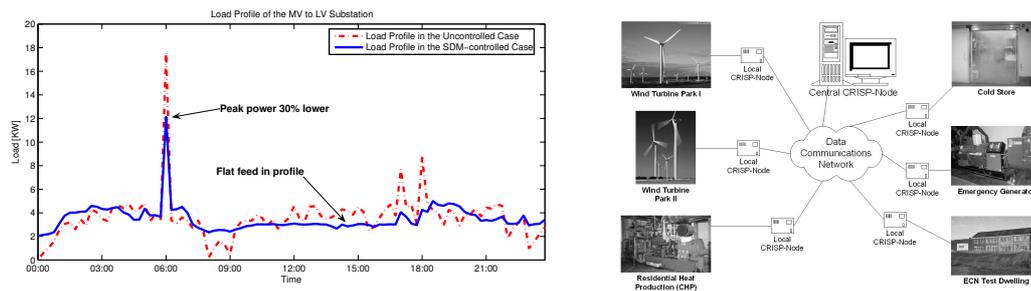


Figure 1: *Left*: Multiagent control leads to peak load reduction and power profile smoothing. *Right*: Market-based control field experiment set-up for imbalance reduction for electricity trade.

Heat pumps (electricity consumers) heated 20 of the dwellings; micro-CHP units heated the other half. The simulation treated washing machines as shiftable operation devices with a predefined operational time window, photovoltaic solar cells as stochastic-operation devices, and lighting as user action devices.

The left side of figure 1 shows the result of a typical simulation run. The graph shows the consumed power in the cluster, both when all devices are free running and when the market-based control agents match supply and demand. This simulation shows that our method can exploit flexibility in device operation through agent bids in an electronic power market. The peak in electricity demand is substantially lower in the controlled case. From the viewpoint of network operations, this result is important, because the highest expected peak demand determines the needed network capacity (transformers and cables). Reducing this peak reduces network investments. Furthermore, introducing SDM results in a flatter, smoother profile of the electricity fed in from the midvoltage network. This result is interesting from the viewpoint of electricity trading, where increased predictability of both production and consumption adds value.

Field Testing. We're investigating the PowerMatcher in real-life environments for two different business cases. One aims to automatically reduce the imbalance in a commercial trader's real world portfolio by aggregating medium sized industrial electricity producing and consuming installations (see figure 1, right). In this experiment, overproduction and underproduction of wind parks induce price changes on the cluster's electronic market. The other devices' control agents react to this with counteractions, which restore the cluster's energy balance. The first test results indicate a decrease of the total power imbalance by more than 40 percent. Reduction of unpredictability in the trade portfolio reduces imbalance costs charged to the trader by the independent transmission system operator.

The other field test, on a cluster of micro-CHP units operating as a virtual power plant, demonstrates their ability to contribute to a common control goal. This experiment uses 15 domestic heating systems at consumer premises. The virtual power plant can provide value through electricity trading or local grid-operation support.

References

- [1] Hans Akkermans, Jos Schreinemakers, and Koen Kok. Emergence of control in a large-scale society of economic physical agents. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1232–1234, 2004.
- [2] Per Carlsson. *Algorithms for Electronic Power Markets*. PhD thesis, Uppsala University, Sweden, 2004.
- [3] Koen Kok, Cor Warmer, and René Kamphuis. PowerMatcher: multiagent control in the electricity infrastructure. In *AAMAS '05: Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*, volume industry track, pages 75–82, New York, NY, USA, 2005. ACM Press.
- [4] Koen Kok, Cor Warmer, and René Kamphuis. The PowerMatcher: Multiagent control of electricity demand and supply. *IEEE Intelligent Systems*, 21(2):89–90, March/April 2006.
- [5] Michal Pechoucek and Simon G. Thompson. Agents in industry: The best from the AAMAS 2005 industry track. *IEEE Intelligent Systems*, 21(2):86–95, March/April 2006.
- [6] Fredrik Ygge and Hans Akkermans. Resource-oriented multicommodity market algorithms. *Autonomous Agents and Multi-Agent Systems*, 3(1):53–71, 2000.

A Cross-Language Approach to Historic Document Retrieval

Marijn Koolen^{ab} Frans Adriaans^c Jaap Kamps^{ab} Maarten de Rijke^a

^a *ISLA, University of Amsterdam*

^b *Archives and Information Studies, University of Amsterdam*

^c *Utrecht Institute of Linguistics OTS, Utrecht University*

Abstract

The full paper appeared as: M. Koolen, F. Adriaans, J. Kamps and M. de Rijke. A Cross-Language Approach to Historic Document Retrieval. In: M. Lalmas et al. (eds.), *Advances in Information Retrieval: 28th European Conference on Information Retrieval (ECIR 2006)*, LNCS 3936, pp. 407–419, Springer Verlag, 2006.

Introduction Our cultural heritage, as preserved in libraries, archives and museums, is made up of documents written many centuries ago. Large-scale digitization initiatives, like DigiCULT [2], make these documents available to non-expert users through digital libraries and vertical search engines. For a user, querying a historic document collection may be a disappointing experience. Natural languages evolve over time, changing in pronunciation and spelling, and new words are introduced continuously, while older words may disappear out of everyday use. For these reasons, queries involving modern words may not be very effective for retrieving documents that contain many historic terms. Although reading a 300-year-old document might not be problematic because the words are still recognizable, the changes in vocabulary and spelling can make it difficult to use a search engine to find relevant documents. To illustrate this, consider the following example from our collection of 17th century Dutch law texts. Looking for information on the tasks of a lawyer (modern Dutch: *advocaat*) in these texts, the modern spelling will not lead you to documents containing the 17th century Dutch spelling variant *advocaet*. Since spelling rules were not introduced until the 19th century, 17th century Dutch spelling is inconsistent. Being based mainly on pronunciation, words were often spelled in several different variants, which poses a problem for standard retrieval engines. We therefore define Historic Document Retrieval (HDR) as the retrieval of relevant historic documents for a modern query. Our approach to this problem is to treat the historic and modern languages as different languages, and use cross-language information retrieval (CLIR) techniques to translate one language into the other.

Earlier research has seen similar approaches. In 1992, Robertson & Willett [4] used spelling correction techniques and phonetic substitutions for retrieving 17th century English spelling variants of modern words. The phonetic substitutions were constructed manually. They found that phonetic substitutions (i.e., replacing a typically historic sequence of characters by a modern sequence with the same pronunciation) have very little effect, while spelling correction techniques increased performance in finding spelling variants. Braun [1] used (again, manually constructed) rewrite rules, similar to phonetic substitutions, to rewrite historic character sequences to modern character sequences. In this case, they turned out to be very effective in an IR experiment, making a modern Dutch stemmer [3] more effective after rewriting the historic documents. However, the manual construction of rewrite rules is very time-consuming, and each set of rules that is created, is language dependent. Rules created for 17th century Dutch will probably not work for 17th century English, nor for 14th century Dutch. Constructing rule sets automatically would save a lot of time and effort. But is it possible to construct such translation resources automatically? Furthermore, is this cross-language approach (translating historic language into modern language or vice versa) a suitable approach to HDR?

Table 1: Evaluating translation and stemming effectiveness, using the title field of the topic statement (top half) or its description field (bottom). Best scores are in boldface, significance * = $p < .05$, ** = $p < .01$.

Method	MRR	% Change
<i>Baseline (titles)</i>	0.1316	–
<i>RNF-all + RSF + PSS</i>	0.2780**	+111.2
<i>RNF-all + RSF + PSS + Stemming</i>	0.2766**	+110.2
<i>Baseline (descriptions)</i>	0.1840	–
<i>RNF-all + RSF + PSS</i>	0.2842*	+54.5
<i>RNF-all + RSF + PSS + Stemming</i>	0.3410**	+85.3

Results We have developed algorithms to construct rewrite rules which replace historical character combinations with modern variants. Using N-gram frequencies, typical historic character combinations are selected for rewriting (like *ae* in *advocaet*). A character combination is typically historic if it is much more frequent in a historic corpus than in a modern corpus. By replacing these N-grams in the historic words with a wildcard *, the resulting word (*advoc*t*) is matched with modern words. The character combination in a modern word (*advocaat*) that matches the wildcard is a possible substitute for the historical character combination (resulting in *ae* → *aa*). Since the rules are based on the relative frequency of n-grams, we call this algorithm the Relative N-gram Frequency (RNF) algorithm. (A variant of this is the RSF algorithm, based on the relative frequency of vowel sequences and consonant sequences.) Another approach to construct rules is to use phonetic transcriptions of historic and modern words, and match those words that are pronounced the same. By aligning sequences of vowels, and sequences of consonants, differences in spelling can be transformed into rewrite rules. (*ae* is aligned with *aa*, resulting in *ae* → *aa*.) This method is the Phonetic Sequence Similarity (PSS) algorithm. Multiple modern character combinations can be found for a historic character combination. The modern combination that is found most often, is used. The rules are applied on the historic collection to construct a translation resource, which is used in the retrieval experiments.

Our experimental evidence is based on a collection of 17th century Dutch documents and a set of 25 known-item topics in modern Dutch. We have experimented with both query translation (adding historic variants to modern words) and document translation (modernizing the historic documents), and found that document translation outperforms query translation. Table 1 shows the retrieval results for our most successful run, which uses a combination of the RNF, RSF, and PSS rules. We see that, after rewriting, we can use a modern stemming algorithm. This improves the performance on the description fields.

Conclusions Our main findings are as follows: First, we are able to automatically construct rules for modernizing the historic language using algorithms that compare historic and modern words on the phonetic and orthographic level, and use statistics to bridge the gap. Second, modern queries are not very effective for retrieving historic documents, but the historic language tools lead to a substantial improvement in retrieval effectiveness. The improvements are above and beyond the improvement due to using a modern stemming algorithm (whose effectiveness actually goes up when the historic language is modernized). However, our approach only addresses the spelling gap. The problems caused by changes in vocabulary are unresolved. We’re working on this specific problem as well. Currently, we are investigating the possibilities of mining annotations from 17th century literary texts to construct a translation dictionary.

Acknowledgments Thanks to Margariet Moelands (National Library of the Netherlands) for drawing our attention to historic document retrieval. Thanks to Loes Braun for making available the *Antwerpse Compilatae* and *Gelders landen landrecht* corpus. This research was supported by the Netherlands Organization for Scientific Research (NWO, grants # 016.054.616, 639.072.601, 017.001.190, 220-80-001, 264-70-050, 365-20-005, 612.000.106, 612.000.207, 612.013-001, 612.066.302, 612.066.513, 612.069.006, 640.001.501, and 640.002.501), and the E.U.’s 6th FP for RTD - project MultiMATCH contract IST-033104.

References

- [1] L. Braun. Information retrieval from Dutch historical corpora. Master’s thesis, Maastricht University, 2002.
- [2] DigiCULT. Technology challenges for digital culture, 2005. <http://www.digicult.info/>.
- [3] M.F. Porter. An algorithm for suffix stripping. *Program*, 14:130–137, 1980.
- [4] A.M. Robertson and P. Willett. Searching for historical word-forms in a database of 17th-century English text using spelling-correction methods. In *Proceedings ACM SIGIR ’92*, pages 256–265, 1992.

OWL AND QUALITATIVE REASONING MODELS

Jochem Liem Bert Bredeweg

*Human Computer Studies Laboratory, Informatics Institute, Faculty of Science,
University of Amsterdam*

Abstract

The Naturnet-Redime project needs to share qualitative knowledge models of issues relevant to sustainable development and the Web Ontology Language (OWL) seems the obvious choice for representing such models to allow search and other activities relevant to sharing knowledge models. However, although the design choices made in OWL are properly documented, their implications for Artificial Intelligence (AI) are part of ongoing research. This paper explores the expressiveness of OWL by formalising the vocabulary and models used in Qualitative Reasoning (QR), and the applicability of OWL reasoners to solve QR problems. To create the OWL definitions of the QR vocabulary and models, existing OWL patterns are used as much as possible. However, some new patterns, and pattern modifications, had to be developed in order to represent the QR vocabulary and models using OWL. A parser has been developed and implemented to export (and import) the QR representations to (and from) OWL representations. These exported model files are shared via an online repository. For details see [1].

1 Introduction

The Qualitative Reasoning and Modelling (QRM) community has the desire to share, search for, and reuse qualitative models. The qualitative modelling and simulation workbench Garp3¹ normally saves models to a binary format, which makes it hard to search for models with specific content or structures. The Semantic Web initiative proposes that the solution to this problem is to make content machine accessible. Therefore, within the NaturNet-Redime project², functionality has been implemented in Garp3 to export models to, and import models from, a Web Ontology Language (OWL) format. A central online repository¹ for (OWL) qualitative models is created to allow easy interchange, searching and reuse of qualitative model content.

One of the main problems encountered during the formalisation of qualitative models was the uncertainty whether OWL is expressive enough to represent the complex knowledge representation used in Garp3. In general, the consequences of the design choices in OWL for advanced Artificial Intelligence applications are still unclear. Therefore, a focus of our research is to find out whether the OWL classifier can be used to replace one of the typical reasoning tasks done by a Qualitative Reasoning (QR) engine.

2 Qualitative Reasoning Models

QR models consist of a large set of possible model ingredients which can be specialised by the user. There are strict rules concerning which model ingredients can be connected to others. Two types of aggregates of model ingredients exist: scenarios and model fragments. Both consist of a specific set of model ingredients. In scenarios model ingredients represent facts about the system to be simulated. In model fragments the model ingredients can be incorporated as either conditions or as consequences. They model the causal dependencies (consequences) which hold in certain situations (conditions). Model fragments themselves can be reused within other model fragments.

¹<http://hcs.science.uva.nl/QRM/>

²<http://www.naturnet.org>

Part of QR reasoning is a classification task. Given a scenario, which describes a specific situation, all the model fragments which match the scenario are determined. A model fragment matches a scenario if the conditions in the model fragment match the facts in the scenario. The consequences of the matching model fragments are incorporated into the scenario. Consequently, other model fragments could match the scenario. The final result is an augmented scenario in which all the causal dependencies are present. From this augmented scenario the successor states of the system are determined, and the same algorithm is applied to these new states. The end result is a state graph which describes the possible behaviours of the simulated system.

3 Results

We have developed a generic ontology of the QR vocabulary in which all the model ingredients and their usage restrictions are formalised. Based on this ontology, a formalisation of qualitative models has been designed. The qualitative models, which are in OWL Full, refer to the model ingredients defined in the generic ontology, which is OWL DL. We have successfully implemented OWL export and import functionality, which has been added to the Garp3 qualitative reasoning and modelling tool¹ using the SWI-Prolog Semantic Web Library³. This functionality is currently used to automatically formalise QR models as domain ontologies, which can be shared using an online model repository¹. These model ontologies were checked for consistency using the Triple20⁴ and Protégé⁵ ontology editors and the RacerPro reasoner⁶. Examples of qualitative models in OWL (which reference the QR vocabulary ontology) can be found in the model repository.

We found that it is impossible to formalise model fragments and scenarios in a way that allows the use of an OWL reasoner to perform the QR classification task. The model fragments would have to be formalised as classes with the conditions as necessary and sufficient restrictions. The scenarios would be formalised as instances, allowing them to be classified to model fragments. The main problem is that it is impossible to distinguish between two model ingredients of the same type using restrictions (e.g. two populations). To make the formalisation possible, the model fragments are formalised as classes with relations to instances (making the ontology OWL Full), preventing OWL reasoners to perform the classification task.

Existing knowledge representation patterns were used and adapted to succeed in the formalisation of QR models. The relations in QR models had to be reified, as they have a position on screen which has to be stored, and some of them are tertiary. The existing patterns for relation reification makes reuse of the relations impossible, since they fix either both the type of the target and the source of the relation, or just the type of the target. Our new pattern solves this reusability issue by moving the relation restrictions to the source class.

The existing patterns to formalise sequences of values are not sufficient to represent the sequences of qualitative values (called quantity spaces) as found in qualitative models. Sets of individuals and sets of classes have no fixed order. An rdf:List representation makes it impossible to classify a value depending on the owner of the list. Furthermore, a list has no semantic meaning. We created our own representation using values and inequalities to address these issues.

Finally, we developed a new pattern to impose relation usage restrictions on classes with certain conditions. New classes are defined with the conditions as necessary and sufficient restrictions. The necessary restrictions are imposed on these classes, and will apply when instances are classified as belonging to the class. This avoids information redundancy, as instances do not have to be explicitly represented as belonging to certain classes.

References

- [1] Jochem Liem and Bert Bredeweg. Owl and qualitative reasoning models. In *29th Annual German Conference on Artificial Intelligence (KI2006)*, 2006. Lecture Notes on Artificial Intelligence. Springer 2006. (To appear).

³<http://www.swi-prolog.org/>

⁴<http://www.swi-prolog.org/packages/Triple20/>

⁵<http://protege.stanford.edu>

⁶<http://www.racer-systems.com/>

Parallel Multicolony ACO Algorithm with Exchange of Solutions

Max Manfrin Mauro Birattari Thomas Stützle Marco Dorigo

IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium

The availability of parallel architectures at low cost, e.g. clusters of PCs connected through fast local networks like Gigabit Ethernet, has widened the interest for the parallelization of algorithms [1]. There are two reasons for parallelizing a metaheuristic if one is interested in performance: (i) given a fixed time to search, the aim is to increase the quality of the solutions found in that time; (ii) given a fixed solution quality, the aim is to reduce the time needed to find a solution not worse than that quality.

We use the Traveling Salesman Problem (TSP), an \mathcal{NP} -hard problem, as a case study for testing the impact on the final solution quality reached, given a fixed run time, of the exchange of solutions among multiple colonies on different interconnection topologies. To solve the TSP we use *MAX-MIN* Ant System (*MMAS*) [5], currently one of the best-performing ant colony optimization (ACO) algorithms [3]. Our implementation of *MMAS* is based on the publicly available ACOTSP code (<http://www.aco-metaheuristic.org/aco-code/>). To have a version that is easily parallelizable, we removed the occasional pheromone re-initialization applied in the *MMAS*, and we use only a best-so-far pheromone update. Our version uses the 3-opt local search and quadrant nearest neighbor lists.

The topologies we consider are: fully-connected, replace-worst, hypercube, and unidirectional ring. For each topology we have developed two versions of the algorithm: a first one in which the communication is synchronous, and a second one in which the communication is asynchronous. We consider also the parallel independent runs (PIR) model in which k copies of the same sequential *MMAS* algorithm are simultaneously and independently executed using different random seeds. The final result is the best solution among all the k runs. These topologies allow us to consider decreasing communication volumes, moving from more global communication, as in fully-connected, to more local communication, as in ring, to basically no communication, as in PIR.

The communication strategy we adopt involves the exchange of best-so-far solutions every r iterations, after an initial period of “solitary” search. A colony *injects* in his current solution-pool a received best-so-far solution if and only if it is better than its current best-so-far solution, otherwise it disregards it. The main advantage of using best-so-far solutions over pheromone matrices is that less data has to be exchanged: for the smallest instance that we consider, each pheromone matrix requires several megabytes of memory space, while a solution requires only some kilobytes.

All algorithms are coded in C using LAM/MPI 7.1.1 communication libraries. Experiments were performed on a homogeneous cluster of 4 computational nodes running GNU/Linux Debian 3.0 as Operating System. Each computational node contains two AMD Opteron 244 CPUs, 2 GB of RAM, and one 1 GB Ethernet network card. The nodes are interconnected through a 48-ports Gbit switch.

Computational experiments are performed with 8 colonies of 25 ants each that exchange the best-so-far solution every 25 iterations, except for the first 100 iterations. We considered 10 instances from TSPLIB with a termination criterion based on run time, dependent on the size of the instance. For each of the 10 instances, 10 runs were performed. In order to have a reference algorithm for comparison, we also test the equivalent sequential *MMAS* algorithm. We considered two cases: in the first one (SEQ), it runs for the same overall wall-clock time as a parallel algorithm (8-times the wall-clock time of a parallel algorithm), while in the second one (SEQ2), it runs for the same wall-clock time as one CPU of the parallel algorithm.

To compare and aggregate results across different instances, we normalize the results with respect to the distance from the known optimal value. We refer the reader interested in the raw data to the URL: <http://iridia.ulb.ac.be/supp/IridiaSupp2006-001/> and to [4].

Our hypothesis is that the exchange of best-so-far solutions among different colonies speeds up the search for high quality solutions, having a positive impact on the performance of the algorithms. Our experimental setup allows us to use statistical techniques for verifying if differences in solutions quality found by the algorithms are statistically significant.

The computational results indicate that all the parallel models perform on average better than the equivalent sequential algorithm, but that the best performing approach is PIR. The differences in performance of all the parallel models with information exchange from those of PIR are statistically significant w.r.t. the *Wilcoxon rank sum* test [2] with *p-values* adjusted by Holm's method, while differences in performance among interconnection topologies are not statistically significant.

The impact of communication on performance seems, therefore, negative. One reason might be that the run times are rather high, and *MMAS* easily converges in those times.

The modification we implemented to have a version of *MMAS* that is easily parallelizable result in a stagnation behavior of the sequential algorithm; this stagnation behavior can be avoided to a large extent by parallel independent runs, which also explains its overall good behavior, biasing the performance in favor of PIR over all the other parallel models. An apparent problem of our communication scheme is that communication is too frequent. To better understand the impact that the frequency of communication has on performance, we change the communication scheme to an exchange every $n/4$ iterations, except during the first $n/2$, where n is the size of the instance. The computational results of the new communication scheme on the parallel models *replace-worst* and *ring* show that the reduced frequency in communication has indeed a positive impact on the performance of the two parallel algorithms, even though this is not sufficient to achieve better performance w.r.t. PIR.

We believe that better performance than PIR can be obtained by the parallel models either adding the restarting feature, or implementing communication schemes that avoid early convergence. This second approach could be achieved implementing the acceptance of solutions from other colonies only when they "differ" less than a certain number of components, leading to the creation of groups of colonies that search in different areas of the search space, or by exchanging the solutions with a frequency that depends on both, instance size and run time.

Acknowledgments

This work is supported by 'COMP²SYS' and by the 'ANTS' project. 'COMP²SYS' is a Marie Curie Early Stage Training Site, funded by the European Community's Sixth Framework Programme under contract number MEST-CT-2004-505079. The 'ANTS' project is an Action de Recherche Concertée funded by the Scientific Research Directorate of the French Community of Belgium. M. Dorigo and T. Stützle acknowledge support from the Belgian National Fund for Scientific Research (FNRS), of which they are a Research Director and a Research Associate, respectively.

References

- [1] Enrique Alba, editor. *Parallel Metaheuristics: A New Class of Algorithms*. Wiley Series on Parallel and Distributed Computing. Wiley-Interscience, Hoboken, NJ, 2005.
- [2] W. J. Conover. *Practical Nonparametric Statistics*. John Wiley & Sons, New York, NY, third edition, 1999.
- [3] M. Dorigo and T. Stützle. *Ant Colony Optimization*. MIT Press, Cambridge, MA, 2004.
- [4] Max Manfrin, Mauro Birattari, Thomas Stützle, and Marco Dorigo. Parallel ant colony optimization for the traveling salesman problem. In M. Dorigo, L. M. Gambardella, M. Birattari, A. Martinoli, R. Poli, and T. Stützle, editors, *Ant Colony Optimization and Swarm Intelligence, 5th International Workshop, ANTS 2006*, volume 4150 of *Lecture Notes in Computer Sciences*, Berlin, Germany, Sep 2006. Springer-Verlag.
- [5] T. Stützle and H. H. Hoos. *MAX-MIN* Ant System. *Future Generation Computer System*, 16(8):889–914, Jun 2000.

MULTI-CLASS CORRELATED PATTERN MINING

Siegfried Nijssen ^{ab}

Joost N. Kok ^c

^a *Albert-Ludwigs-Universität, Georges-Köhler-Allee, Gebäude 097, D-79110, Freiburg im Breisgau, Germany*

^b *Katholieke Universiteit Leuven, Celestijnenlaan 200A, 3001, Heverlee, Belgium*

^c *LIACS, Leiden University, Niels Bohrweg 1, 2333 CA, Leiden, The Netherlands*

Frequent pattern mining algorithms, such as the frequent itemset mining algorithm APRIORI, are often used to construct features for classification algorithms, such as decision trees, support vector machines, etc. To focus the search for patterns in such a context more effectively, it has been proposed not to mine for frequent patterns, but to search for correlated patterns. Correlated patterns are patterns that achieve a sufficiently high correlation with a desired target class in a database. If correlation is measured with a convex measure, of which the χ^2 test is an example, it was shown, for example, by Morishita et al. [2] that it is feasible to perform a branch-and-bound search for all patterns that achieve a high correlation value.

It is already known for several years that frequent itemset mining, which deals with the discovery of all sets of items that are a subset of a sufficient number of example (supermarket) transactions, is also feasible in practice. Given the efficiency of many frequent pattern mining algorithms, it is natural to ask what the relation between correlated pattern mining and frequent itemset mining is. If there is such a relation, this is useful in the context of *inductive databases* [1]. An inductive database can be thought of as a database that also supports data mining operations. One key idea of an inductive database is to store patterns also in a database, such to allow reuse and efficient retrieval of previously discovered patterns.

In our paper [3], we provide new theoretical insight in the connection between correlated pattern mining and frequent pattern mining. We first study the two-class classification problem of Morishita et al., and show that the branch-and-bound search is equivalent to the following algorithm:

1. From a threshold on the correlation value, derive minimum supports $minsup_1$ and $minsup_2$, for each of the two classes a threshold;
2. Mine frequent itemsets with threshold $minsup_1$ in the examples of the first class;
3. Mine frequent itemsets with threshold $minsup_2$ in the examples of the second class;
4. Scan the data once to fill in missing supports for those itemsets that are infrequent in one of the two classes; output only those itemsets for which the correlation is high enough.

Thus, there is an obvious link between Morishita et al.'s approach and frequent itemset mining, and it is clear that we can exploit one of the many efficient implementations for frequent itemset mining to deal with the correlated pattern mining problem. Furthermore, our method performs what an inductive database is supposed to provide: reuse of patterns for different tasks.

In this paper, we obtain several results. First, we provide formulas to compute minimum support threshold values for correlation tests. These formulas allow an inductive database to automatically determine if reuse of patterns is possible. The main idea is to transform a correlation threshold into support thresholds, and to determine if frequent patterns for these thresholds, or lower ones, have already been computed. Only if not, they are computed, and added to the database. In this way, multiple correlation queries and minimum frequency queries can reuse each other's results.

Next, we show that our observation extends to the setting in which there are more than two classes. We illustrate this for the χ^2 test on higher dimensional contingency tables, and provide a proof that a threshold on such a correlation test, for any dimension, can be transformed into a

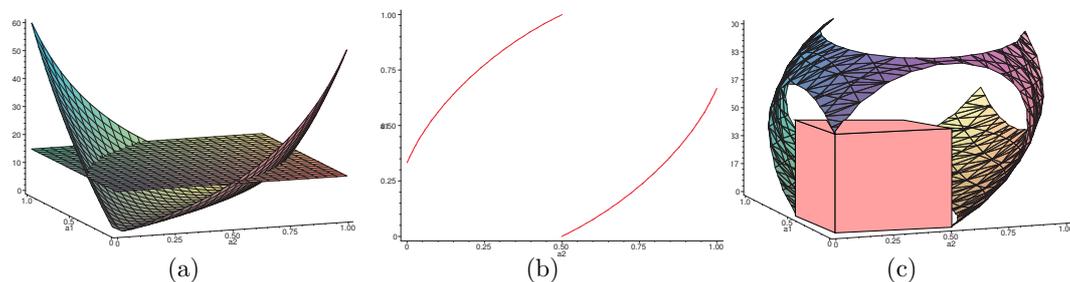


Figure 1: Illustrations of the χ^2 correlation measure: (a) given on the x and y axis are the numbers of examples covered by a hypothesis in two classes of a dataset, on the z axis are the corresponding χ^2 correlation values; clearly, if only examples in one class are covered, the correlation is highest; the horizontal plane depicts a threshold on the correlation value; (b) depicted is where the correlation measure and the threshold of (a) cross each other, as seen from above; the points where the lines in this figure cross the x and y axis, respectively, correspond to the minimum support thresholds that should be used in our approach; (c) shown is the same kind of figure as (b), but then for three classes (resulting in three dimensions); we prove that the ‘box’ can be put into the three dimensional body, which means that three minimum support thresholds can be used to find correlated patterns.

disjunction of minimum frequency thresholds for the individual classes. Our observations for the two class problem, therefore extend to higher dimensions, for this test. Additionally, we investigate the possibilities for reusing frequent patterns computed for higher dimensions when grouping multiple classes together to reduce the number of dimensions.

A nice benefit of our method is that, given the reduction to a linear number of minimum frequency thresholds, we can prune relatively efficiently. In previous work, it was suggested that an exponential number of tests would be required, but this turns out not to be the case.

We conclude our work by doing experiments. In these experiments, we find no significant benefits of our approach for a single run on a dataset with two classes, other than that it is simpler to run an existing frequent pattern mining algorithm than to implement a new algorithm. On the other hand, if particular kinds of consecutive queries are considered, they are more efficiently answered if previous results are reused, and our approach thus makes interactive data mining more feasible. Furthermore, for high numbers of target classes, our strategy is several orders of magnitude faster than naive approaches.

In this paper, we mostly concentrate on the χ^2 correlation test. We believe that many of our observations also apply to other heuristics, such as the information gain heuristic commonly used when learning decision trees, but we leave an investigation of further heuristics as future work.

Acknowledgements This work was partly supported by the EU FET IST project IQ (“Inductive Querying”), contract number FP6-516169.

References

- [1] T. Imielinski and H. Mannila. A database perspective on knowledge discovery. In *Communications of the ACM*, volume 39, pages 58–64, 1996.
- [2] S. Morishita and J. Sese. Traversing itemset lattices with statistical metric pruning. In *ACM SIGACT-SIGMOD-SIGART*, pages 226–236, 2000.
- [3] S. Nijssen and J.N. Kok. Multi-class correlated pattern mining. In *Knowledge Discovery in Inductive Databases (KDID)*, LNCS 3933, pages 165–187, 2005.

Audio Indexing Technology for the Exploration of Audiovisual Heritage Collections

extended abstract¹

Roeland Ordelman

Franciska de Jong
Arjan van Hessen

Willemijn Heeren

Human Media Interaction, University of Twente, The Netherlands

1 Introduction

A number of techniques from the AI-realm have proven to have added value for spoken document retrieval. Browsing tools for audio and/or video archives not only benefit from speech recognition, but also from techniques such as clustering, topic detection, speaker classification and segmentation. This paper will discuss audio indexing tools that have been implemented for the disclosure of Dutch audiovisual cultural heritage collections, and will analyze the specific requirements imposed by the nature and formats of the collections from a technological point of view. Moreover, the paper argues that research is needed to cope with the varying information needs for different types of users.

The number of digital audio collections in the cultural heritage domain is growing rapidly. Whereas the growth of storage capacity is in accordance with widely acknowledged predictions, the possibilities to index and access these archives is lagging behind. As a result, particular information may only be accessible via manual browsing of a collection of files, which is extremely time-consuming. Recent years have shown that automatic speech recognition can successfully be deployed for equipping spoken-word collections with search functionality. This is especially the case in the broadcast news domain. For that domain speech transcripts approximate the quality of manual transcripts for several languages. In other domains, a similar recognition performance is usually harder to obtain due to (i) a lack of domain-specific training data, in addition to (ii) a large variability in audio quality, speech characteristics and topics that are addressed. This applies to historical, audio(visual) data in particular. The application of audio indexing to Dutch historical audio collections, however, may greatly improve their accessibility.

2 Audio indexing for Dutch oral history collections

Full text transcription of historical audio data from a number of national audiovisual archives showed that search technology based on speech recognition might easily collapse. This is due to shockingly high word error rates caused by the typical characteristics of historical material (a wide variety in audio quality, background noise, overlapping and spontaneous speech, topics that are unknown beforehand, etc.). Therefore, requirements for successful tuning and improvement of available tools for indexing the heterogeneous A/V collections from the cultural heritage domain are reviewed through a number of pilot projects. Both the adaptation of language models to historical settings, and the adaptation of acoustic models for a homogeneous audio collection are discussed. Audio indexing for the historical domain is complicated, however, by the fact that speech training databases for that particular domain are only minimally available.

For homogeneous oral history collections, speaker adaptation of acoustic models is capable of reducing the word error rate considerably as is illustrated by the W.F. Hermans project, [2]. To overcome the mismatch of statistical language models based on contemporary text with the old-fashioned language and unknown

¹The full paper appeared in the Proceedings of the First European Workshop on Intelligent Technologies for Cultural Heritage Exploitation at ECAI 2006, [1].

words in the task domain, historical in-domain text data are needed - preferably in large amounts. At least some information on the topics of the particular documents is wanted, e.g. to reduce the numbers of out-of-vocabulary words. In case collateral text data is available for an audio collection, it is worthwhile to investigate whether synchronization of text data with the audiovisual data using alignment techniques is an option. This will result in a time-aligned index suitable for subtitling, search and cross-media browsing (by linking semantic representations from different media). In the Radio Oranje project a number of speeches from Queen Wilhelmina (1880-1962), broadcast from England and addressed to the Dutch people during World War II, were thus aligned with their written versions.

A number of techniques described above have been implemented in two separate demos that illustrate how the concept of cross-media browsing for a multimedia archive can be realized. The first demonstrator, the cross-media news browser, was initially a demonstrator for on-line access to an archive of Dutch news broadcasts (NOS 8 uur Journaal). It shows how either available collateral data sources (subtitling information for the hearing-impaired) or full-text speech recognition transcripts can be used as linguistic content for the generation of time-coded indexes for searching within audio archives. Next to the broadcast news browser, at TNO a news browser for heterogeneous media archives has been developed: Novalist, [3]. It aims to facilitate the work of information analysts by (i) clustering related news stories to create dossiers, (ii) analysing and annotating dossiers with several types of metadata, and (iii) providing a browsing screen with multiple views on the dossiers and their metadata. These technologies can also be deployed for the disclosure of audio archives from the cultural heritage domain.

3 Variance in information needs

Multiple levels of annotation will become available and collection fragments can be linked to internal or external multimedia sources via cross-media linking. Different types of users (e.g. archivists, information analysts, researchers, teachers and the general public) are expected to have varying information needs with respect to those annotation levels. In research, for instance, questions can be asked that apply to any of the levels of metadata information and particularly new insights emerging from combining a multitude of views on the data will be interesting. The general public, however, is more likely to search for information related to their personal interests. These user needs in the cultural heritage domain must therefore be investigated more elaborately. Moreover, professional users of audiovisual archives could contribute knowledge, for example in a personalized peer-to-peer set-up that stimulates the exchange of content.

Given the greatly varying information needs from different types of users, interface requirements for search options and data presentation are also likely to differ between user groups. For eventual successful deployment of the tools to be built, the development of methodologies for the use of historical multimedia collections is a prerequisite. In the ideal case, tools for adequate navigation and selection may even unfold new information.

4 Conclusion

The challenge is to facilitate access to heterogeneous audio collections from the cultural heritage domain. For such collections, extraction of speech transcripts and metadata calls for robust audio indexing technology that performs well irrespective of speaker, bandwidth or audio quality. In addition, user research is needed to gain insight into users' information needs and system requirements that will optimally serve those needs.

References

- [1] R. Ordelman, F. de Jong, and W. Heeren. Exploration of audiovisual heritage using audio indexing technology. In *Proceedings of the First European Workshop on Intelligent Technologies for Cultural Heritage Exploitation at ECAI 2006*, pages 36–39, 2006.
- [2] M.A.H. Huijbregts, R.J.F. Ordelman, and F.M.G. de Jong. A spoken document retrieval application in the oral history domain. In *Proceedings of the 10th International Conference Speech and Computer*, pages 699–702, 2005.
- [3] W. Kraaij. Novalist: the multimedia news browser. <http://twentyone.tpd.tno.nl/druid/folders/novalist.pdf>.

Immune Anomaly Detection Enhanced with Evolutionary Paradigms

Marek Ostaszewski ^a Franciszek Seredynski ^{b c} Pascal Bouvry ^a

^a *Faculty of Sciences, Technology and Communication, University of Luxembourg,
6 rue Coudenhove Kalergi, L-1359 Luxembourg-Kirchberg, Luxembourg*

^b *Polish-Japanese Institute of Information Technology,
Koszykowa 86, 02-008 Warsaw, Poland*

^c *Institute of Computer Science, Polish Academy of Sciences,
Ordonia 21, 01-237 Warsaw, Poland*

1 Introduction

The amounts of data exchanged between computers over the Internet are growing every year. The detection of intrusions in computer networks is a difficult task mostly because the methods and tools used for unauthorized accessing to various network resources keep evolving. Many approaches have been applied to the problem of recognition of malicious activities appearing in network traffic, however new and more effective ways of classification of behavior are still being pursued.

An artificial immune system (AIS) is a computational paradigm based on the abstraction of natural immunological processes [2], which can, among others, be applied to solve computer security problems, including the detection of intrusions and anomalies. AIS offers innovative mechanisms to deal with unwanted activities in computer networks, including the recognition of previously unknown attacks. Our work relies on a recent and promising network anomaly detection approach described in [1]. We construct a model of the network behavior based on the analysis of the legitimate traffic by Nature-inspired approaches. The network traffic model uses hyperrectangular structures as subspaces of behavior patterns and coevolutionary-based mechanisms to enhance the process of anomaly detection.

2 Immune anomaly detection

The AIS methodology called *Self-Nonself* paradigm is applied as guideline for constructing a network behavior classification system. The *self* space corresponds to the organism protected by its natural immune system, which consequently cannot consider as enemy any defended cells. In the network domain, the *self* space is defined on the basis of a normal traffic and the *nonself* space contains all possible threats and deviations. Self-Nonself space principles are coupled with a negative selection algorithm, which is used to construct detectors focused on *nonself* space [1].

The construction of an appropriate network model requires to record regular values of the network traffic in order to characterize the *self* space. A set of k parameters describing network traffic has to be selected. In addition to the recording of values, a generalization ability of the system is desired. It means the ability to recognize monitored values that are similar to *self* at a given level of deviation from given data. Intervals of normal behavior can have different values for every parameter taken into consideration and allow to construct accurate and flexible *self* description. Everyone of the recorded regular patterns is transformed into a structure defining acceptable values by changing it into a hyperrectangle. This hyperrectangle has its center in the recorded pattern and low and high boundaries defined by each allowed deviation level. The *self* space structure is complex because it consists of numerous (several thousands in our experiments) of uniform structures. The same set of deviation levels is applied to every recorded pattern.

The *nonself* space is complementary to *self*, therefore it should contain all anomalous patterns of the network traffic. Our objective is to cover *nonself* space with a small number of structures of different sizes. A *nonself* detector will also be a hyperrectangular structure based on intervals, but constructed in a more flexible way, different than the development of *self*. [1] proposes a genetic algorithm with a sequential niching (NGA) for this purpose, that is solving a multimodal and multiobjective problem, and this approach is also applied in the present work. NGA manages to develop detectors, that cover the *nonself* space efficiently with relatively small number to *self* structures (several dozens in our experiments).

3 Coevolution in anomaly detection

Our proposition for enhancing the detector generation process is based on coevolutionary mechanisms. Coevolution is a relatively new research paradigm in the field of evolutionary computation. The basic idea is taken from the world of Nature, where two or more coexisting species are constraining one other to evolve better features. Among many coevolution models, one seems to be useful to the detector generation problem. *Predator - prey* paradigm [3] describes a model, where individuals of one type (predators) are trying to catch individuals of another type (preys). The population of the first species develops features that allow it to catch its preys easily, and attributes of the second one evolve to make escape from a predator possible. Applying this process to the detector generation mechanism could improve it by providing a certain goal in search space. Coevolution controls the process by enforcing on generated detectors certain features, indicating areas to cover. In this way, a coevolving population of fixed points in the *nonself* space is constructed, as areas to focus. The classical NGA and its coevolutionary version (cNGA) were compared and their results are illustrated in Figure 1. The number of encounters corresponds to the number of interactions between detectors and anomalies taking place during development of detectors. The number of intercepted anomalies shows how successful were particular methods against a 1000-element anomaly set.

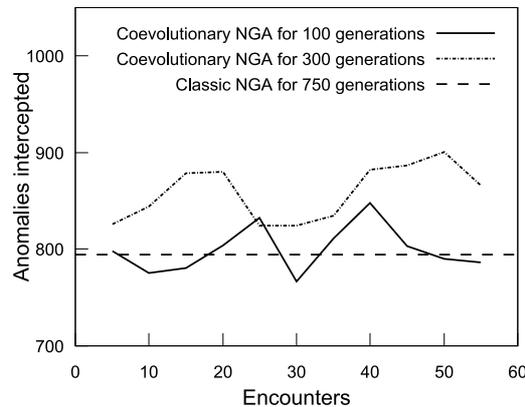


Figure 1: Comparison of coevolution efficiency with classic NGA approach

The experiments featuring coevolution show a significant advantage of the cNGA compared to the previous version of the algorithm (NGA), which implies that the additional population can indeed stimulate the detector generation process. Even relatively small number (100) of generations allows the cNGA to obtain better results than the classic NGA with 750 generations and with a smaller computational cost.

References

- [1] D. Dasgupta and F. Gonzalez. An immunity-based technique to characterize intrusions in computer networks. *IEEE Transactions On Evolutionary Computation*, 6(3), 2002.
- [2] S. M. Garret. How do we evaluate artificial immune systems? *Evolutionary Computation*, 13(2), 2005.
- [3] J. Paredis. Constraint satisfaction with coevolution. In *New Ideas in Optimization*. McGraw-Hill, 1999.

A Fast Implementation of the EM algorithm for Mixture of Multinomials¹

Jan Peter Patist

Computational Intelligence, Vrije Universiteit Amsterdam

Abstract

We propose several simple techniques that dramatically reduce both the memory demand and computational effort in building multinomial mixture models using the EM [1] algorithm. The reason of the dramatic improvement in performance is that the techniques make use of certain properties of the data. These properties are: the data is sparse, there are many repeating records and the number of different values per attribute is small. We claim that particular sources of data consistently satisfy these properties. Excellent examples are click-stream and retail data, which are sparse and consist of many repetitions. Using simple techniques, huge speed-ups and compression rates, on real life click-stream data sets, are observed compared to the standard implementation of the EM algorithm.

1 Introduction

Every day a huge amount of data is generated in the form of click streams, telephone records, multimedia data, sets of retail chain transactions, et cetera. With the increase of the amount of data the need grows for fast algorithms to analyze these huge datasets.

We propose some simple techniques for efficiently implementing the EM for the estimation of a multinomial mixture model on large data sets. The effectiveness of these techniques is based on certain assumptions about the data. We assume that the data is discrete and sparse, (i.e. that the data contains lots of zero-values), the data contains repeating records and the number of different values per attribute is relatively small compared to the number of records.

In practice we observed that data sets from particular sources often satisfy the prerequisites. This makes our techniques important for the analysis, using multinomial mixtures, of domain specific data. For example, data containing page visits within sessions from web-log-files are often highly sparse and redundant. The effectiveness of our techniques is tested on data sets originating from page visits. Mixture of multinomials is commonly used in this domain either for the purpose of representing the data density or defining clusters.

It is not surprising that click-stream data is sparse and contain lots of repetitive records. This is because visitors are usually focused only on a few pages.

We propose three techniques, which make use of repetition of records and sparsity. These techniques are: We transform the data into a weighted data set. In the weighted data set duplicate records are replaced with the unique record and its frequency. We introduce a sparse representation, and propose a lookup table to prevent recalculations of exponentials or products, in case of the use of logarithms. The modified implementation of the EM algorithm (“enhanced EM”) does not change the local maximum found.

We report on the speed-up and compression obtained on real life web-access log data sets.

2 Experiments

We compared the running time and memory requirements of standard EM and the enhanced EM on several real world data sets. The data sets are: webshop1, webshop2, cs.vu.nl and msnbc. All

¹Based on “J. P. Patist. A Fast Implementation of the EM algorithm for Mixture of Multinomials.” to be published in LNAI 4093, International Conference on Advanced Data Mining and Applications (ADMA 2006)

DataSet	#records	#variables	Mean Unique Val.	% Non-Zeros	Compr. Repl.	Total Comp.
cs.vu.nl	249,280	16	30	0.083 %	32	148
Msnbc	989,818	17	87	0.101 %	8.7	48
Webshop1	203,023	22	24	0.147 %	3.5	23.5
Webshop2	2,950,708	51	55	0.044 %	5.4	67

Table 1: Statistics of several data sets. The size is expressed in the total amount of numbers. “Compression rate replicates” is calculated by dividing the number of records in standard representation of the data set by the number of unique records. “The mean unique values” is the average number unique values per variable or dimension. The “Total Compression” is the compression obtained by combining replication compression and the sparse data representation.

Figure 1: Speed-up- and Compression ratio of enhanced EM as a function of the number of components on the data sets webshop1, webshop2, msnbc and cs.vu.nl.

the data sets originate from web-log-files of user behavior. Basic summary statistics can be found in Table 1. The speed-up and compression factor in building mixtures of multinomials of the enhanced EM is shown in Figure 1.

3 Conclusion

We proposed some simple techniques to boost the performance of the standard implementation of EM for mixture of multinomials in terms of computation effort and memory load.

These simple techniques were: removal of duplicate records, not storing of and no calculations over zero-values, and building a reference table of all encountered combinations of counts and parameter values to prevent recalculations of exponentials or products (in the case of the use of logarithms).

We have shown that using these techniques huge speed-up, in the order of 50 to 400, and memory compression 20 up to 150, could be achieved. We conclude that these simple techniques are very effective in the context of click-stream data.

References

- [1] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society B*, 39:1–38, 1977.

MONOTONE CLASSIFICATION BY FUNCTION DECOMPOSITION

Viara Popova^a

Jan C. Bioch^b

^a *Department of Artificial Intelligence, Vrije Universiteit,
De Boelelaan 1081a, 1081HV Amsterdam, The Netherlands.
popova@few.vu.nl*

^b *Econometric Institute, Erasmus University Rotterdam,
P.O. Box 1738, 3000 DR Rotterdam, The Netherlands.
bioch@few.eur.nl*

The full version of this paper was published in: A. G. Hoffmann, H. Motoda and T. Scheffer, editors, Proceedings of The 8th International Conference on Discovery Science (DS'05), volume 3735 of Lecture Notes in Computer Science, pp. 203-214, Springer, 2005.

Problem decomposition approaches are used in many areas of science, e.g. switching theory, game theory, reliability theory, machine learning. One of the applications in machine learning is in structured induction which aims at splitting a concept to be learnt in a hierarchy of sub-concepts which can be used separately to generate classification rules. The methods vary but the majority of them involve a human expert who provides domain knowledge about the underlying structure of the problem.

The contribution of this paper is within the research in automating the decomposition process. We look at the problem from the point of view of classification for monotone data sets. A data set with a set of attributes A and a labelling λ is called *monotone* if the values of each attribute are ordered and for each two data points x, y such that $x \leq y$ (y dominates x on all attributes, i.e. $\forall i x_i \leq y_i$) it is true that $\lambda(x) \leq \lambda(y)$. Monotone classification has been studied in the context of logical analysis of data, decision trees, decision lists, neural networks, rough sets theory, instance-based learning, etc. (see [2] for a list of references) In this paper we aim at building a decomposition hierarchy that preserves the monotonicity property with the ultimate goal of generating a monotone classifier.

Let us have a monotone data set D with an attribute set A and a discrete monotone labelling $\lambda : D \rightarrow \{0, m\}$. Let S_0 and S_1 be two disjoint subsets of attributes such that $S_0 \cap S_1 = \emptyset$, $S_0 \cup S_1 = A$. A scheme of the type $f = g(S_0, h(S_1))$ is called *positive* if the functions g, h are positive.

Monotone decomposition looks at the question whether for given S_0 and S_1 there exists an extension of the positive scheme $f = g(S_0, h(S_1))$. The requirement that $h(S_1)$ should be positive implies that the data set generated by S_1 and the corresponding values given by h should satisfy the monotonicity constraint. We denote this data set by $S_1|h$. Similarly the requirement that g should be positive implies that the resulting set (denoted by $S_0h|\lambda$) after replacing S_1 in D with the corresponding values of h should also satisfy the monotonicity constraint. The problem has so far been investigated in the context of *Boolean functions* in [1]. There a criterion is given for the existence of an extension of positive schemes of a number of different types.

In this paper we investigate the corresponding problem in the context of *discrete functions*. We build partly on the algorithm presented in [4, 5] for the general (non-monotone) case. This algorithm recursively decomposes a discrete function $y = f(A)$ into $y = g(S_0, h(S_1))$ where S_0 and S_1 are disjoint and $S_0 \cup S_1 = A$. The functions g and h are not predefined in the application of the method and are induced during the decomposition process. The requirement for them is to have joint complexity which is lower than the complexity of f and that is determined using some complexity measure. By applying the method recursively on h and g we can generate a hierarchy of concepts.

Our approach uses the following lemma stating that there exists a positive extension for the scheme $f = g(S_0, h(S_1))$ if and only if there exists an assignment of values $\{h_i\}_{i=1}^k$ such that the two new data sets $S_1|h$ and $S_0h|\lambda$ are monotone.

Based on this constraint on the two data sets we generate two sets of constraints on the values of h of the type $h_i \leq h_j$ and $h_i > h_j$. A natural way of representing the constraints is in a directed graph with vertices corresponding to $\{h_i\}_{i=1}^k$ and two types of directed edges. This representation can be used for finding a consistent assignment for h . Intuitively, such assignment cannot be found if a cycle is present. The paper formulates and proves a criterion for whether there exists an assignment for the values $\{h_i\}_{i=1}^k$ and gives two possible assignments for h .

We apply a graph theory algorithm called topological sorting to order the vertices so that all edges point in the same direction. We can now find an assignment as follows. If no edges start from the vertex, assign h_{\min} . Otherwise for each such edge: extract the label of the end vertex; for solid edges add 1 to the corresponding number; find the maximal among the numbers for all edges ending in x ; assign this maximal number to the current vertex.

Our algorithm inherits the characteristic of the general decomposition algorithm that it cannot guarantee coverage of the whole input space. A default rule is needed that results in a monotone classifier. We propose two alternatives for the labelling function which were previously used in the Monotone Decision Trees algorithm ([3]) and are proven to give consistent labels when the data set is monotone.

In order to investigate the successfulness of the proposed algorithm, some experiments were conducted on the Nursery data set. It is a real-world monotone data set generated using a hierarchical model developed by experts and therefore has *known* underlying structure. The final decomposition structure extracted by the algorithm is identical to the one developed by the experts and used to produce the data set.

References

- [1] E. Boros, V. Gurvich, P.L. Hammer, T. Ibaraki, and A. Kogan. Decomposability of partially defined Boolean functions. *Discrete Applied Mathematics*, 62, 51-75, 1995.
- [2] V. Popova. *Knowledge Discovery and Monotonicity*, PhD Thesis, Erasmus University Rotterdam, The Netherlands, 2004.
- [3] R. Potharst and J.C. Bioch. Decision Trees for Ordinal Classification. *Intelligent Data Analysis*, 4, 1-15, 2000.
- [4] B. Zupan. *Machine learning by function decomposition*. PhD Thesis, University of Ljubljana, 1997.
- [5] B. Zupan, M. Bohanec, J. Demsar and I. Bratko. Learning by Discovering Concept Hierarchies. *Artificial Intelligence*, 109, 211-242, 1999.

A COLLABORATIVE FILTERING METHOD FOR CONSTRUCTING UTILITY GRAPHS USED IN MULTI-ISSUE NEGOTIATION ¹

Valentin Robu

Han La Poutré

*CWI, Center for Mathematics and Computer Science
Kruislaan 413, NL-1098 SJ Amsterdam, The Netherlands*

Abstract

Graphical utility models represent powerful formalisms for modeling complex agent decisions involving multiple issues. In the context of negotiation, our previous work [3] has shown that using utility graphs enables agents to reach Pareto-efficient agreements with a limited number of steps in high-dimensional negotiations involving non-linear dependencies. This paper considerably extends the results of [3], by proposing a method for constructing the utility graphs of buyers automatically, based on previous negotiation data. The proposed method is based on techniques inspired from item-based collaborative filtering, used in on-line recommendation algorithms. Experimental results show that our approach is able to retrieve the structure of utility graphs on-line, with a relatively high degree of accuracy, even if a relatively small amount of data about concluded negotiations is available. Furthermore, we provide a rigorous method for determining the cut-off number of edges for a wide category of random graphs.

1 Introduction

Negotiation represents a key form of interaction between providers and consumers in electronic markets. One of the main benefits of negotiation in e-commerce is that it enables greater customization to individual customer preferences, and it supports buyer decisions in settings which require agreements over complex contracts. Multi-issue (or multi-item) negotiation models are particularly useful for this task, since they enable agents to reach mutually beneficial ("win-win") contracts. We consider the problem of a seller agent negotiating bilaterally with a customer about selecting a subset from a collection of goods or services, viz. a bundle, together with a price for that bundle. A bottleneck in most existing approaches to automated negotiation is that they only deal with linearly additive utility functions, and do not consider the problem of interdependencies between evaluations for different items. However, accounting for substitutability/complementarity effects between different items can be crucial in reaching mutually profitable deals.

2 Overview of the proposed method

In our previous work [3], in order to model buyer preferences in high-dimensional negotiations, we have introduced the concept of utility graphs. Intuitively defined, a utility graph (UG) is a structural model of a buyer, representing a buyer's perception of dependencies between two items (i.e. whether the buyer perceives two items to be as complementary or substitutable). An estimation of the buyer's utility graph can be used by the seller to efficiently compute the buyer's utility for a "bundle" of items, and propose a bundle and price based on this utility. The main result presented in [3] is that Pareto-efficient agreements can be reached, even for high dimensional negotiations with a limited number of negotiation steps, but provided that the seller starts the negotiation with a reasonable approximation of the *maximal structure* of the true utility graph of the type of buyer he is negotiating with (i.e. he has a reasonable idea which issues or items may be complimentary or substitutable in the evaluation of buyers in his domain).

¹This is an abstract of work first presented at PRIMA'05 [2], Kuala Lumpur, Malaysia, and further extended and improved for RSS'06 [1], Hakodate, Japan. The full version of the paper is available for download at: <http://homepages.cwi.nl/~robu/rss2006.pdf>

An important issue left open in [3] is how does the seller acquire this initial graph information. One method would be to elicit it from human experts (i.e. an e-commerce merchant is likely to know which items are usually sold together or complimentary in value for the average buyer and which items are not). For example, if the electronic merchant is selling pay-per-item music tunes, the tunes from the same composer or performer can be potentially related.

In this paper, we show they can also be retrieved automatically, by using information from completed negotiation data. The implicit assumption we use here is that buyer preferences are in some way clustered, i.e. by looking at buyers that have shown interest for the same combinations of items in the past, we can make a prediction about future buying patterns of the current customer. Note that this assumption is not uncommon: it is a building block of most recommendation mechanisms deployed in Internet today. Thus, we do not need to collect negotiation data which could be traceable back to specific customers. Collecting such personalized data may prove difficult anyway in a large, open electronic environment (such as the Internet), where most most buyer/seller encounters are likely to be first time. Furthermore, by not storing personalized information, our approach protects customer privacy. In order to generate the initial structure of our utility graph, in this paper we propose a technique inspired by collaborative filtering.

There are two main stages of our approach:

1. Using information from previously concluded negotiations to construct the maximal super-graph of dependencies likely to be encountered in negotiation. In this phase the information used (past negotiation data) refers to a class of buyers and is not traceable to individuals.
2. The actual negotiation, in which the seller, starting from a super-graph for a class (population) of buyers, will negotiate with an individual buyer, drawn at random from the buyer population. In this case, learning occurs based on the buyer's previous bids during the negotiation, so information is buyer-specific. The learning at this stage is guided by the structure of the super-graph extracted in the first phase.

3 Conclusions

Our approach, in contrast to other work on negotiation, is based on using information from previous negotiations in order to aid buyer modeling in future negotiation instances. As shown, this does not mean that personalized negotiation information about specific customers needs to be stored, only aggregate information about all customers. The main intuition behind our model is that we explicitly utilize, during the negotiation, the clustering effect between the structure of utility functions of a population of buyers. This is an effect used by many Internet product recommendation engines today, in order to shorten the period required for customers to search for items (though it comes under different names: collaborative filtering, social filtering etc.). When adapted and used in a negotiation context, such techniques enable us to handle high dimensional and complex negotiations efficiently. In this paper, this is achieved by using techniques derived from collaborative filtering (widely used in current e-commerce practice) to learn the structure of utility graphs which model the buyer utilities over different bundles. Thus, we show that the link between collaborative filtering and negotiation is a fruitful research area, which, we argue, can lead to significant practical applications of automated negotiation systems.

References

- [1] V. Robu and J.A. La Poutré “Constructing the Structure of Utility Graphs Used in Multi-Item Negotiation through Collaborative Filtering of Aggregate Buyer Preferences”. In *Proceedings of the 2nd International Workshop on Rational, Robust and Secure Negotiations in Multi-Agent Systems RSS'06, Hakodate, Japan, 2006* (to appear in Springer Lecture Notes in Computational Intelligence series).
- [2] V. Robu and J.A. La Poutré “Learning the Structure of Utility Graphs Used in Multi-Issue Negotiation through Collaborative Filtering”. In *Proc. of the 8th Pacific Rim Int. Workshop on Agents PRIMA'05, Kuala Lumpur, Malaysia, Sept. 2005* (to appear in Springer-Verlag LNAI series).
- [3] V. Robu and D.J.A. Somefun and J.A. La Poutré “Modeling Complex Multi-Issue Negotiations Using Utility Graphs”. In *Proceedings of the 4th International Conference on Autonomous Agents and Multi-Agent Systems AAMAS'05, Utrecht, The Netherlands, 2005*.

Decision Trees for Hierarchical Multilabel Classification: A Case Study in Functional Genomics (Extended Abstract)¹

Leander Schietgat ^a Hendrik Blockeel ^a Jan Struyf ^{ad} Sašo Džeroski ^b
Amanda Clare ^c

^a *Department of Computer Science, Katholieke Universiteit Leuven
Celestijnenlaan 200A, 3001 Leuven, Belgium*

^b *Department of Knowledge Technologies, Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia*

^c *Department of Computer Science, University of Wales Aberystwyth
SY23 3DB, UK*

^d *Department of Computer Science, University of Wisconsin
1210 West Dayton, Madison, WI 53706, USA*

Hierarchical multilabel classification (HMC) is an extension of binary classification where an instance can be labelled with multiple classes that are organised in a hierarchy. A well-known application of this kind of problem is gene function prediction. A gene can have multiple functions at the same time, and these functions are hierarchically organised: a gene predicted to have a certain class should also be predicted to have all its superclasses, as given by the hierarchy.

A straightforward approach to solve this problem would be to learn a binary classifier for each class separately and then to combine the predictions. However, this has several disadvantages: (1) learning is not very efficient, since a separate classifier has to be learned for each class, (2) binary classifiers have known problems with skewed class distributions and (3) the hierarchy constraint, implying that a class should be predicted along with all its superclasses, is not automatically fulfilled. The obvious alternative is to learn a single model that predicts all the different classes at once. In this paper we propose a method for learning decision trees that predicts for each instance a set of classes instead of a single class. We experimentally compare this method with the approach of learning a separate decision tree for each class (predicting for each instance whether it belongs to the class or not).

We use a decision tree learner called Clus, which is based on the principles of predictive clustering trees [1]. These trees represent a different view on decision trees where the examples in a leaf can be seen as clusters. Clus builds such trees top-down, choosing in each node the test that scores best according to some heuristic. Clus's heuristic is based on "variance reduction". Each instance is associated with a vector that has one component per class, which is 1 if the instance belongs to the class and 0 otherwise. The variance of a set of vectors is defined as the average squared Euclidean distance between any vector in the set and the mean vector. For a single target variable, this heuristic boils down to the Gini-index used by CART [3]. In addition, in the HMC setting, it makes sense to give a higher cost to misclassifications higher up in the hierarchy than lower down.

For the experiments, we use 12 datasets² representing different aspects of the genes in the genome of *Saccharomyces cerevisiae*. Five types of bioinformatic data are considered: sequence statistics (D_1), phenotype (D_2), predicted secondary structure (D_3), homology (D_4), and expression as measured with microarray chips ($D_5 - D_{12}$). The number of examples in each dataset ranges from 1592 to 2932, the number of attributes from 27 to 47034. Every instance is labeled with one or more functions from the MIPS FunCat classification scheme. On 4/24/2002, this hierarchy contained 250 classes, spread over 4 levels.

¹The full paper of this abstract can be found in [2].

²Available at <http://www.aber.ac.uk/compsci/Research/bio/dss/yeastdata/>.

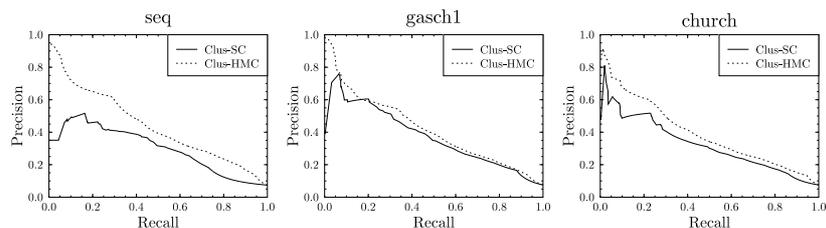


Figure 1: Representative PR-curves for Clus-SC and Clus-HMC.

In order to validate this new approach to HMC decision tree learning, we compared it to the current state-of-the-art, which is C4.5H [4]. Evaluating the precision and recall (accuracy isn't a suitable measure in a problem setting where classes have different frequencies), Clus scored a better result in almost all cases. The following is an example of a rule returned by Clus-HMC:

```
IF Nitrogen_Depletion_8_h <= -2.74 AND Nitrogen_Depletion_2_h > -1.94 AND
  1point5_mM_diamide_5_min > -0.03 AND 1M_sorbitol__45_min_ > -0.36 AND
  37C_to_25C_shock__60_min > 1.28
THEN 40, 40/3, 5, 5/1
```

The rule identifies conditions when a gene has classes 40/3 and 5/1 (and hence also their superclasses). This rule has a precision/recall of 0.97/0.15 for class 40/3 and 0.94/0.37 for class 5/1.

In comparing HMC tree learning to learning multiple binary trees, we looked at the size of the trees, runtimes, and predictive performance. The latter was evaluated using precision-recall curves. It turns out that a single HMC tree is smaller in size (HMC trees contained on average 24 nodes, binary trees predicting a single class 33 nodes), takes less time to build (with an average speedup factor of 37) and, as Fig. 1 shows, has a higher area under the curve (and thus a better predictive performance) than the combination of multiple binary trees. The latter is somewhat less expected. Further investigation revealed that single trees tend to overfit more easily.

These results confirm that the development of decision tree methods that predict multiple classes from a hierarchy at once (as opposed to learning a separate tree for each class) is a worthwhile undertaking: it leads to the discovery of models that are more interpretable and more efficient without suffering accuracy loss.

Acknowledgements

Leander Schietgat is supported by a PhD grant of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT-Vlaanderen). Hendrik Blockeel and Jan Struyf are post-doctoral fellows of the Fund for Scientific Research of Flanders (FWO-Vlaanderen).

References

- [1] H. Blockeel, L. De Raedt, and J. Ramon. Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning*, pages 55–63, 1998.
- [2] Hendrik Blockeel, Leander Schietgat, Jan Struyf, Sašo Džeroski, and Amanda Clare. Decision trees for hierarchical multilabel classification: A case study in functional genomics. In *Proceedings of the 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2006.
- [3] L. Breiman, J.H. Friedman, R.A. Olshen, and C.J. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.
- [4] A. Clare. *Machine learning and data mining for yeast functional genomics*. PhD thesis, University of Wales, Aberystwyth, 2003.

Exploration and Exploitation Bias of Crossover and Path Relinking for Permutation Problems ¹

Dirk Thierens

Department of Computer Science, Universiteit Utrecht, The Netherlands

For many combinatorial optimization problems the computational complexity of evaluating a complete solution is typically an order $O(n)$ higher than the cost of evaluating an incremental change made by a local perturbation operator to the current solution. Metaheuristic search algorithms like iterated local search or simulated annealing can benefit of this efficiency gain since they only apply small local changes to the current solution. When applying crossover a much larger step in the search space is taken, and the offspring has to be evaluated from scratch. This puts crossover at a potential disadvantage when compared to more local search operators. In this paper we note that crossover can however be implemented in a way that it also only needs incremental fitness evaluation updates. The trick is to construct a path through the search space that connects a parent solution with the offspring solution through a series of intermediate solutions. By calculating the incremental fitness updates along the path we can compute the fitness of the offspring without the need for a full fitness evaluation. Implementing crossover this way actually makes it very similar to the path relinking approach found in scatter search. In fact, the similarities become larger than the differences. In this paper we compare the search bias of crossover and path relinking for permutation problems where the absolute position of the elements is decisive. Calculations show that uniform permutation crossover can reach many more permutations from a given parent couple than path relinking. Uniform permutation crossover is therefore more exploratory than random path relinking, which is itself more exploratory than greedy path relinking.

We also compared the three search operators experimentally on 5 instances of the quadratic assignment problem with 50 facilities and locations. The QAP is a permutation problem where the absolute position of the elements is important. It is a notoriously hard combinatorial optimization problem and is relevant for several real world problems. We applied the three search operators in a steady-state genetic local search framework. The local search is performed by a best-improvement 2-swap neighborhood search. The experiments indicated that on 4 of the 5 instances greedy path relinking was more successful – within the given computational time – than random path relinking, which was itself more successful than uniform permutation crossover. On the 5th instance however greedy path relinking was not able to find solutions as good as the other two operators could. Although the higher exploitation bias of the greedy path relinking allowed it to be more successful in most instances, it also made it more vulnerable to being misled toward suboptimal solutions.

It is important for users to understand the differences in search bias of the operators they use. In general, applying operators with a higher focus on exploitation than on exploration will result in a more efficient search process but it will also be less robust. It is up to the user to decide how much computation time she is willing to invest to find a balance on the exploitation-exploration trade-off.

References

- [1] D. Thierens. Exploration and Exploitation Bias of Crossover and Path Relinking for Permutation Problems. In *Proceedings of the 9th International Conference on Parallel Problem Solving from Nature (PPSN IX)*. T.P. Runarsson et al. (eds.). LNCS 4193, pp. 1028-1037, 2006.

¹The full version of this paper has been published in the [1].

Predicting Mortality in the Intensive Care Using Episodes*

Tudor Toma^a

Ameen Abu-Hanna^a

Robert-Jan Bosman^b

^a *Academic Medical Center, University of Amsterdam, Department of Medical Informatics, PO Box 22700, 1100 DE Amsterdam, The Netherlands*

^b *Department of Intensive Care, Onze Lieve Vrouwe Gasthuis, 1e Oosterparkstraat 279, PO box 10550, 1090 HM Amsterdam, The Netherlands*

Patient outcome prediction lies at the heart of various medically relevant tasks such as quality assessment and decision support, and is an important research issue in medical informatics and AI in medicine. In the Intensive Care (IC) there are various prognostic models in use today that predict patient mortality. These are logistic regression models that predict the probability of death of an IC patient based on severity of illness scores that are calculated from information that is collected within the first 24 hours of patient admission. For example the SAPS (simplified acute physiology score) quantifies the patient's condition at admission and is used as the only covariate in the SAPS logistic regression model.

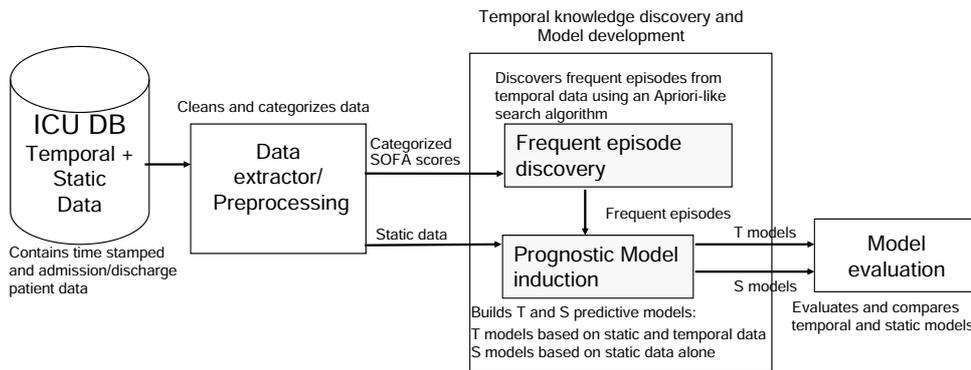


Figure 1: Workflow showing the primary tasks involved in our approach

Recently, IC units started, in addition, collecting sequential organ failure assessment (SOFA) scores that quantify the degree of organ derangement for each patient on *each day* of the IC stay [1]. A SOFA score varies between 0 to 24 where greater values imply worse condition of the patient's organ systems. For example 7-4-2-1 is a SOFA sequence for a patient that stayed for 4 days and steadily recovered. Although SOFA scores are primarily meant for recording incidence of organ derangement and failures, the hypothesis is that they contribute to better prediction of mortality. There is virtually no systematic way in the literature to exploit the temporal character of SOFA scores for prediction. This work, originally appearing in [2], adapts ideas from temporal datamining for discovery of sequential episodes and suggests a way to put them into use in the problem of mortality prediction. In particular our approach consists of the following steps

*This research was supported by the Netherlands Organisation for Scientific Research (NWO) under the I-Catcher project, number 634.000.020.

(Figure 1): 1. data preprocessing, 2. frequent episodes discovery, 3. prognostic models development, and 4. model evaluation.

Data preprocessing is applied on the temporal SOFA scores and deals with imputing missing values and removing artifacts introduced by medical data collection. SOFA scores were also categorized in three qualitative categories according to minimal entropy [3] with respect to mortality. Three SOFA categories were obtained: $SOFA \leq 10$, $11 \leq SOFA \leq 13$ and $SOFA \geq 14$. For the sake of interpretability we denote these categories, respectively, as L (LOW), M (MEDIUM) and H (HIGH). Although we look at coarse representation of the data, our approach preserves the inter-relationship between the values in the sequence.

Discovering frequent episodes (temporal patterns) in the SOFA sequences is achieved by implementing and using an adaptation of the Apriori-like algorithm described in [4]. M-H-H-M-L is an example of a temporal SOFA pattern corresponding to a 5 day evolution in the categorized SOFA scores. The data support threshold for an episode to be considered frequent was set to 5%.

For the **development of prognostic models** we use the logistic regression framework. A frequent pattern's presence in a patient is coded by a binary (indicator) variable used as a possible covariate in a logistic regression model. A fixed covariate in this model is the SAPS. We have created five (temporal) models for predicting hospital mortality after 1,2,3,4 and 5 days of stay in the ICU. In each model the probability of death is given by the formula: $P(died = 1|Ep_i) = \frac{e^{\beta_0 + \beta_1 SAPS + \sum \beta_i I[Ep_i]}}{1 + e^{\beta_0 + \beta_1 SAPS + \sum \beta_i I[Ep_i]}}$ where $I[Ep_i]$ is an indicator variable which codes for the presence of an episode Ep_i selected during the model fitting strategy.

In **model evaluation** on the test sets we relied on calibration-oriented (rather than discrimination-oriented) performance measures including the Brier score, defined as $1/N \sum_{i=1}^N (P(Y_i = 1 | x_i) - y_i)^2$ (where N denotes the number of patients in the test set, and y_i denotes the actual outcome for patient i), and the logarithmic score, $\sum_{i=1}^N LS_i$ where $LS_i = -\ln P(Y_i = y_i | x_i)$. Calibration accounts for the difference between the predicted and true probabilities. The performance of each temporal model was then compared to its respective reference (SAPS) model (a logistic regression model using the SAPS as the only covariate) on the same part of the test set. All the temporal models 1, 2, 3, 4 and 5 outperformed their respective reference model.

In summary, we introduced a new method for integrating temporal patterns in outcome prediction and showed the added value of the resulting predictive models compared to the logistic regression models which are in use today.

References

- [1] Vincent, J.-L. and Ferreira, F.L. Evaluation of organ failure: we are making progress, *Intensive Care Med.*, 26:1023–1024, 2000.
- [2] Toma, T., Abu-Hanna, A. and Bosman, R. Predicting Mortality in the Intensive Care Using Episodes. *IWINAC, LLNCS*, 3561:447-458, 2005.
- [3] Kohavi, R. and Sahami, M. Error-based and entropy-based discretization of continuous features, *KDD*, 114–119, 1996.
- [4] Mannila, H. and Toivonen, H. and Verkamo, A.I. Discovering frequent episodes in sequences, *Data Min. Knowl. Discov.*, 1(3):259–289, 1997.

Bagging using Statistical Queries

Anneleen Van Assche

Hendrik Blockeel

Department of Computer Science, Katholieke Universiteit Leuven, Celestijnenlaan
200A, 3001 Leuven, Belgium

The full version of this paper appeared in: *Proceedings of The 17th European Conference on Machine Learning*. Berlin, Germany, 2006 [6].

Introduction Bagging [1] operates by building several decision trees on replicate training sets that are produced by sampling with replacement from the original training set. By combining the votes of the different trees it is able to improve the predictive accuracy.

In some learning settings, direct access to individual examples of the data is not available; instead, the learning algorithm has access only to summary statistics about the data. Several authors have described how various predictive models (such as decision trees) can be learned from precomputed statistics such as itemset frequencies [5] or AD-trees [4], and what the advantages are of doing so.

When trying to apply bagging in the context of learning from statistics, one is confronted with the fact that statistics are needed for several resampled versions of the dataset, rather than for the dataset itself. Since no direct access to the data is available, resampling the dataset by randomly selecting (with replacement) individual examples from it is not possible, so the straightforward approach of resampling the dataset and computing the statistics for these resampled versions is not possible. The obvious alternative is to simulate the computation of statistics from randomly resampled versions of the dataset by computing the distribution (under random resampling of the dataset) of the statistics themselves, and then sampling values for these statistics from those distributions. The latter approach, applied to the specific case of bagging decision trees, is the subject of this paper.

Sampling the statistics The key idea to bagging without bootstrapping is the following. In classical bagging, the information gain IG_A (or some other heuristic) of an attribute A is computed from a random resample E_i of the original data set E . Since the resample is chosen randomly, it might just as well have been another one, which would have lead to a different value for IG_A . Clearly, computing the exact IG_A on a random resample E_i is equivalent to sampling IG_A from its own distribution. Put differently: since IG_A is a function of the data set, and considering E_i a stochastic variable of which the distribution is known, the distribution of $IG_A(E_i)$ can be computed. Generating values for IG_A from this distribution is equivalent to generating E_i from its own distribution and computing IG_A from it. We call this procedure “resampling the statistics” as opposed to “resampling the data sets” (and computing the statistics from them).

In our approach, the statistic that we resample is not the information gain itself but the relative proportions of the different classes in the subsets created by an attribute test. Information gain, but also other heuristics, can be computed efficiently from this.

How do we obtain sampled statistics from the original statistics? Suppose there are n_c class values and the test attribute has n_a values. Denoting with X_{ij} the number of instances that have class i and attribute value j , the vector containing all the X_{ij} follows a multinomial distribution. So for each iteration we need to draw for each test a new random vector X'_{ij} according to the following multinomial distribution:

$$P(X'_{11} = x_1, \dots, X'_{n_c n_a} = x_k) = \frac{n!}{x_1! \dots x_k!} p_{11}^{x_1} \dots p_{n_c n_a}^{x_k}$$

Table 1: Accuracy results of simulated bagging (SB) and normal bagging (Bag) on 12 UCI data sets.

	Accuracy											
SB	65.7	99.6	39.8	91.7	94.3	79.1	82.7	99.5	100.0	95.3	99.1	79.6
Bag	66.0	99.6	36.4	90.1	93.8	77.4	82.6	99.4	100.0	94.6	99.0	79.3

where $n = \sum X_{ij}$ (the total number of instances), $k = n_c * n_a$ and $p_{ij} = X_{ij}/n$. We use the method proposed by [2] to generate this vector; this method consists of repeatedly generating a number for each separate component X_{ij} (so $n_c * n_a$ times) according to a binomial distribution, using the BTPE algorithm from [3].

Note that this procedure of simulating bagging will not be exactly the same as bagging, because the new statistics that are drawn from the original distributions for each of the tests are taken independently from one another. As such, they will not necessarily correspond to one and the same bootstrap sample, which is the case for bagging. This results in somewhat less strict choice options: while for bagging certain tests may never occur as the best test, some of these might occur with a low probability as a best test in simulated bagging. Another point where bagging and the simulation differ is that while in bagging one complete tree is built on one particular bootstrap this is not the case in this simulated bagging, because new samples are drawn at each node.

Despite these differences, we find that in practice the bagged trees of both methods turn out to be highly similar.

Experimental results and conclusion We implemented this simulated bagging method in the WEKA data mining system [7]. We compared the accuracy of simulated bagging (using resampling of the statistics) to original bagging (using resampling the data) on 12 UCI data sets. As can be seen from Table 1¹, accuracy results of simulated bagging turn out to be as good as those obtained using normal bagging. From these results we can conclude that ignoring dependencies between tests when sampling the statistics does not have a detrimental effect and by doing so we can efficiently simulate bagging when learning from statistical properties of the data. As such, the method proposed makes bagging more generally applicable. Moreover, the described technique is not restricted to bagging only, it can be applied to all methods using resampling, such as certain pruning methods, some methods for error assessment and random forests among others.

References

- [1] L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
- [2] L. Devroye. *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.
- [3] V. Kachitvichyanukul and B. Schmeiser. Binomial random variate generation. *Communications of the ACM*, 31:216–222, 1988.
- [4] A. W. Moore and M. S. Lee. Cached sufficient statistics for efficient machine learning with large datasets. *Journal of Artificial Intelligence Research*, 8:67–91, 1998.
- [5] P. Panov, S. Džeroski, H. Blockeel, and S. Loškovska. Predictive data mining using itemset frequencies. In *Proceedings of the 7th International Multiconference Information Society*, 2005.
- [6] A. Van Assche and H. Blockeel. Bagging using statistical queries. In *Proceedings of The 17th European Conference on Machine Learning, Berlin, Germany*, 2006. accepted.
- [7] I. Witten and E. Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005. 2nd Edition.

¹for more info concerning experimental settings and results see [6]

INFLUENCE-BASED AUTONOMY LEVELS IN AGENT DECISION-MAKING

B. van der Vecht^{a,b} A.P. Meyer^a R.M. Neef^a F. Dignum^b
J-J. Ch. Meyer^b

^a *TNO Defense, Security and Safety, The Hague*

^b *Department of Information and Computing Sciences, Universiteit Utrecht, Utrecht*

1 Extended Abstract

Autonomy is a powerful feature of agents and it is the subject of much research in the agent field. In agent organizations, the ability of an agent to make decisions autonomously can be crucial, for example, in a distributed setting with no or limited communication. In other circumstances agents need to collaborate and make decisions together. Controlling the autonomy of agents is a way to coordinate the behavior of groups of agents. We believe that we can make the concept of agent organizations more powerful by designing agents that are able to handle different levels of autonomy in their decision-making process, and are capable of altering their own level of autonomy based on external influences. We analyze the autonomy of an agent as a gradual property that is related to the degree of intervention of other agents in the decision process. The aim of this research is to define a reasoning model for agents that includes the concept of autonomy [1].

In the agent deliberation process we distinguish four sub-processes: 1. do observations and receive messages, 2. process the observations and messages, and determine their semantics, 3. decide on the next action and 4. perform the selected action. These processes can be recognized in the four phases of agent deliberation: Observe, Orient, Decide and Act (OODA). We use the OODA loop to illustrate how different kinds of inter-agent influences take part in the agent deliberation.

There are three types of influence between agents to be distinguished: influence by environmental modification, influence by belief alteration and influence on goal/task determination. By linking the three types of influences to the first three phases of the OODA loop, we define a reasoning model that includes influence-based autonomy levels. We can implement reasoning profiles for agents, that model the level of intervention of other agents in the decision-making process of the agent explicitly. For example, a *solipsistic* profile in which the agent ignores other agents as much as possible, a *trusting* profile that allows for belief influences between agents, or a *obedient* profile in which influence on task determination has been made explicit.

In an experimental setting we have implemented a firefighter organization situated in a world with fires. The organizational goal is to extinguish the fires as fast as possible. We have defined different organizations by varying the reasoning profiles of the agents. The results show difference in performance between the organizations. This shows that our model can be used to implement organizational relations and roles in individual agents. As future work we want to extend our experiments to find rules for an agent to decide on switching between reasoning profiles. The reasoning model should be extended with a mechanism for agents to reason about and adjust their autonomy levels.

References

- [1] B. van der Vecht, A.P. Meyer, R.M. Neef, F. Dignum, and J-J. Ch. Meyer. Influence-based autonomy levels in agent decision-making. *Workshop on Coordination, Organization, Institutions and Norms in Agents Systems (COIN) at ECAI2006*, 2006.

Representing Causal Information about a Probabilistic Process

Joost Vennekens, Marc Denecker, and Maurice Bruynooghe

{joost,marcd,maurice}@cs.kuleuven.be

Dept. Computerscience, K.U. Leuven

Celestijnenlaan 200A

B-3001 Leuven, Belgium

The goal of this paper [4] is to reinvestigate the role of causality in probabilistic modeling. We do this based on the observation that causal information is inherently information about probabilistic processes. For instance, if one says that smoking causes cancer, then this means that the act of having a cigarette will initiate some sequence of events within the human body, and that one of the possible outcomes of this process will be that a cancer develops. Typically, we are not interested in the details of this actual process itself, but simply care about the probability of it leading to a certain result, such as the development of a cancer. The central thesis of our paper is that causal information is in essence a compact and robust way of specifying just enough properties of the behaviour of a non-deterministic process to uniquely characterize the probability distribution that it generates.

Let us first clarify some terminology. A probabilistic process is a succession of some ‘atomic’ events. Of course, which events are considered atomic depends on the granularity of one’s world view, i.e., one could, for instance, consider the detailed interactions between cigarette smoke and the cells of the human lung tissue, or, more abstractly, simply observe that smoking tends to make people sick. Each such atomic event affects the state of the world in some, possibly non-deterministic, way. An event occurs at some particular time and, typically, also for a certain reason. It should be clear that events and processes of this kind play an important role in our intuitions about causality. From a knowledge representation point-of-view, this now naturally begs the question of where such events occur in our formal models of causal information.

One of the most popular approaches to modeling causal information about a probabilistic domain is that of causal Bayesian networks [2]. In this language, the causal structure of the domain is described by a directed acyclic graph, in which every node corresponds to some unique random variable. The intuitive reading of such a network is that the value of every node is causally determined by the values of its parents in the graph. As such, from our point of view, the probabilistic processes described by such a network are those in which once the values of all parents of a node have been determined, there will occur an event that propagates these values to the node itself. In this way, a network specifies some properties of a probabilistic process, but not all. Indeed, for two nodes with no path between them, it does not say which of the events associated to these nodes will happen first. However, this information is not relevant for the probability distribution that will be generated, i.e., all processes that are possible according to the network will generate the same probability distribution. In this way, causal Bayesian networks fit into the claim we made above.

The kind of events admitted by a Bayesian network are obviously rather restricted. Firstly, only a single event can be involved in determining the value of any particular node. Clearly, in real life, many events might exert an influence on the same property. For instance, not only smoking might cause cancer, but also exposure to radioactivity, excessive sun bathing, and so on. Secondly, because of the acyclic structure of the graph, events can only propagate values in one fixed direction. In reality, this too is not always the case. For instance, if a patient has contracted pneumonia, then this might cause angina. Vice versa, however, if the patient first has angina, then this might also cause pneumonia. As such, in this case, the propagation of values can happen in both directions.

In our paper, we study a representation of causal events that goes beyond these two restrictions. Concretely, we consider what we call *causal probabilistic events*, or *CP-events* for short. Such an event is of the form $E \leftarrow \phi$, which is read as “property ϕ causes event E .” Here, ϕ is a formula of first-order logic and E is some formal representation of a probabilistic event. We then consider the language of *CP-logic*, which consists of all sets of such CP-events, called *CP-theories*.

According to our claims, a CP-theory describes some aspects of a probabilistic process. We make this more precise by specifying exactly what such a theory says about a process, or, to put it the other way around, which processes can be seen as corresponding to such a theory. Informally speaking, a probabilistic process corresponds to a CP-theory if and only if it satisfies the following three fundamental principles:

- The principle of no *deus ex machina* events: Events occur if and only if there is a cause for them to occur.
- The principle of independent causation: Every event represents an independent causal mechanism.
- The principle of temporal precedence: Whenever a property ϕ might cause an event E , then the part of the process that is involved in determining the truth of ϕ happens *before* the event E itself can happen.

In our paper, we formally define what it means for a process to satisfy these three principles. We are then able to mathematically support our claims by proving that all such processes do indeed generate precisely the same probability distribution.

If, as we claim, causality is indeed inherently tied to the notion of probabilistic processes, then one would expect to encounter concepts similar to our CP-events in other causal logics. Moreover, by studying the relation of such logics to CP-logic, we will also learn how to represent causal probabilistic processes in these logics, which provides a valuable piece of knowledge representation methodology for them. Motivated by these observations, our paper relates CP-logic to causal Bayesian networks and various Logic Programming based approaches to causal and probabilistic reasoning, such as Poole's Independent Choice Logic [3] and the causal logic of McCain and Turner [1].

As already mentioned above, CP-logic allows a less restricted kind of causal events than Bayesian networks do. This offers two significant knowledge representation advantages. Firstly, our principle of independent causation allows a better representation of effects that can be caused by a number of independent causes. The main advantage of CP-logic is that each cause for such an effect corresponds to a different CP-event, which makes the independence between these causes a *structural* property of the theory, rather than a numerical property of some probabilistic values in Bayesian networks. This improves the compactness and elaboration tolerance of the representation. Secondly, because of the principle of no *deus ex machina* effects, it is possible to directly and straightforwardly model cyclic causal relations (e.g., “pneumonia might cause angina” and “angina might cause pneumonia”) in CP-logic. In Bayesian networks, on the other hand, such cycles have to be encoded in a more complicated way, by the introduction of additional random variables.

CP-logic is also related to a number of Logic Programming approaches. This relation can be established through a probabilistic way of deriving a normal logic program from a CP-theory. Our main technical result on this topic states that the probability of deriving a logic program that has a certain interpretation I as its model (according to the well-founded semantics for logic programs) is precisely the probability with which I will be the outcome of a process corresponding to the CP-theory. In our paper, we discuss the implications of this result with respect to normal and disjunctive logic programs, the Independent Choice Logic and McCain & Turner's causal logic.

In summary, the aim of our paper is to investigate the claim that causal information is in essence a compact way of specifying enough properties of the behaviour of a probabilistic process to uniquely characterize the probability distribution that it will generate. We do this by constructing the formal language of CP-logic, for which we devise a semantics based on three fundamental principles. We then show that every process that satisfies these principles does indeed generate the same distribution. Finally, we also relate this logic to a number of other causal approaches.

References

- [1] N. McCain and H. Turner. Causal theories of action and change. In *Proc. 13th AAAI/8th IAAI*, 1996.
- [2] J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2000.
- [3] D. Poole. The Independent Choice Logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, 94(1-2):7–56, 1997.
- [4] J. Vennekens, M. Denecker, and M. Bruynooghe. Representing causal information about a probabilistic process. In *Proc. JELIA 2006*, 2006.

Refining Aggregate Conditions in Relational Learning

Celine Vens

Jan Ramon

Hendrik Blockeel

Department of Computer Science, Katholieke Universiteit Leuven,
Celestijnenlaan 200A, 3001 Leuven, Belgium

Introduction In relational learning, predictions for an individual are based not only on its own properties but also on the properties of a set of related individuals. Many systems use aggregates to summarize this set. Features are then constructed by comparing the result of the aggregate function to a threshold, we call such a feature an aggregate condition. For example, in the context of a data set on parents and children, a possible feature could be “*the maximum age of the person’s children is larger than 10*”.

Many learning systems rely on a general-to-specific ordering of the hypotheses to traverse the hypothesis space in an efficient way. Only few of the existing systems that learn hypotheses with aggregates extrapolate this generality ordering to the aggregate conditions, and when they do, they do it in a restricted way. For instance, the feature just mentioned could be refined in three ways: by changing the aggregate function (e.g., change *maximum* into *average*), the subset aggregated over (e.g., specialize *children* into *daughters*) or the threshold compared with (e.g., increase *10* to *15*). No current relational learners consider all three kinds of refinements, and indeed the effect of such refinements on the generality of a rule, and the interaction between these effects, are non-trivial and have not been studied up till now.

This paper presents the first comprehensive study of these effects and interactions. This study leads to the description of a refinement operator that enables relational learners to traverse the hypothesis space more efficiently. Although in this paper we have chosen the inductive logic programming (ILP) formalism for expressing relational concepts, the theory behind it can be applied to other relational representations as well.

Specializing Aggregate Conditions ILP systems learn sets of logic clauses, typically by learning one clause at a time. The ILP system usually finds a clause by starting with the empty clause $h \leftarrow$ and gradually refining it, adding literals to the body of the clause, until it is consistent with the data. We call refinements that never increase the coverage of a clause, valid refinements.

Now assume that one of the literals is an aggregate condition of the form $F(\{V|Q\})\theta R$, where F is an aggregate function, Q a conjunctive query, V a variable occurring in Q , θ one of $\{\leq, \geq\}$ and R a numeric value. For instance, the clause

$$person(P, pos) \leftarrow max(\{A|child(P, C), age(C, A)\}) \geq 10$$

classifies a person as positive if the maximum age of his children is higher than 10. Such a clause could now be refined not only by extending the clause itself with a literal, but also by extending the query Q , or by changing F or R . The question is then under what conditions such refinements are valid. For instance, changing the preceding aggregate condition into

$$max(\{A|child(P, C), age(C, A), male(C)\}) \geq 10$$

is a valid refinement, but with *min* instead of *max* this would not be the case. Also with \leq instead of \geq the above refinement would not be valid.

In order to provide an answer to which refinements of aggregate conditions are valid, we investigate the monotonicity properties of aggregate conditions. In general, we say that a function that maps an ordered domain onto an ordered range is monotone if it preserves the order, anti-monotone if it reverses the order, and non-monotone otherwise. An aggregate condition can be seen as a function mapping an aggregate function, a set, and a threshold value onto a boolean value. For sets,

numeric values, and boolean values an order relation is easy to define. For aggregate functions this is less trivial. In the paper we present several classes of aggregate functions with an order relation. One of these is the class of generalized averages, which is defined as follows:

$$avg_k(S) = \left(\frac{\sum_i (x_i^k)}{n}\right)^{1/k} \text{ with } S = \{x_1, \dots, x_n\}$$

If $S \subseteq \mathbb{R}^+$, then the following order relation holds: $i \leq j \Rightarrow avg_i \preceq_{\mathbb{F}} avg_j$. While this relation holds for all k , in practice only some of these k -values are commonly used: $avg_1(S) = avg(S)$, $\lim_{k \rightarrow \infty} avg_k(S) = max(S)$, and $\lim_{k \rightarrow -\infty} avg_k(S) = min(S)$. Moreover, the order relation $min \preceq_{\mathbb{F}} avg \preceq_{\mathbb{F}} max$ also holds for sets S that contain negative numbers.

Having defined an order relation on each component of an aggregate condition, it is possible to write down the monotonicity properties, from which an aggregate refinement operator can be derived that ensures valid refinements. This refinement operator can specialize an aggregate condition along three dimensions: the aggregate function F , the set of values (or equivalently, the query Q used to generate the set), and the threshold value v . The whole set of hypotheses spanned by these three dimensions can be visualized in what we call a *refinement cube*. Every discrete point in the cube represents a hypothesis and can be constructed in a finite number of steps, starting from one aggregate condition. A chain of refinements in the cube will be called a *path*. For simplicity, we only consider refinements along one direction at a time. We are only interested in valid refinements, therefore we only allow *monotone paths* in the refinement cubes, i.e. paths that consist only of valid refinements. Figure 1 shows the refinement cubes for the generalized averages. There is a cube for each comparison operator and the arrows define the monotone paths.

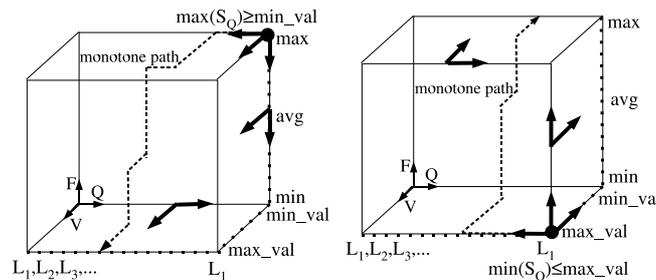


Figure 1: The refinement cubes for the generalized averages.

Application The framework we presented can be beneficial for any relational learning system that learns aggregates and makes use of a general-to-specific ordering of the hypotheses to guide the search. As an illustration we have applied the framework to the relational decision tree learner TILDE. To split a node in TILDE, a refinement operator generates all allowed tests using a given hypothesis language specification. The tests in this generated search space are then executed against the set of examples corresponding to the node. The test yielding the highest information gain is chosen to split the node.

Learning aggregates with TILDE is not new. However, since the generality ordering on aggregate conditions was insufficiently understood, no structure was imposed on the aggregate search space, i.e., the tests in it were executed in random order. Using the monotonicity properties discussed in this paper, we know that whenever a test T fails for an example, none of the tests that can be obtained from T via monotone paths of the refinement cubes has to be executed against that example. Exploiting this knowledge would obviously result in an efficiency gain, while the same search space as before would be searched, and hence, the same tree would be obtained. Our experiments confirmed this: on the datasets tested, speedup factors up to 3.2 were obtained.

Reference to full paper

C. Vens, J. Ramon, and H. Blockeel. *Refining Aggregate Conditions in Relational Learning*. Proceedings of European Conference on Principles and Practice of Knowledge Discovery in Database, Berlin, Germany (Frnkranz, J. and Scheffer, T. and Spiliopoulou, M., eds.), 2006, accepted.

Improving Patient Activity Schedules by Multi-agent Pareto Appointment Exchanging¹

Ivan Vermeulen Sander Bohte Koye Somefun Han La Poutré

*CWI, Centre for Mathematics and Computer Science
Amsterdam, The Netherlands*

1 Introduction

In hospital patient scheduling, the scheduling problem is dynamic and always in flux: operations take more or less time than anticipated, crucial staff may not be available, equipment breaks down, and new patients arrive, some with urgent requirements. Not unimportantly, hospitals are also organized around different autonomous departments (wards, ancillary units) each with their own specialty, and each department essentially has authority over their own schedule. Thus, hospital scheduling has strong decentralized characteristics. Here, we take a first step towards developing a multi-agent hospital patient scheduling solution that respects the current distribution of scheduling authority, and that is capable of continuously adjusting the different schedules in response to the dynamic environment.

Given the distributed and decentralized nature of hospital patient scheduling, the use of a market mechanism for scheduling seems an appropriate fit: markets can efficiently distribute scarce resources, they can facilitate dynamic environments, and only price-quotes need to be exchanged between participants, rather than complex constraints and preferences. In [2] a first step is taken in developing a framework for using virtual markets to solve distributed scheduling problems. Different types of auction mechanisms are analyzed, and the results show that it is hard to find a general solution.

Clearly, when optimizing a schedule the question is what metric to optimize against in a health-care setting, and patient well-being is an obvious choice [1]. Quantifying *relative* patient well-being however is notoriously hard and any choice will be controversial, with both doctors and patients. Here, we develop a scheduling method that offers substantial gains without having to consider this difficult issue. We observe that often patients have multiple appointments and note that the time of the *last* appointment effectively determines the patient's "waiting-time" and a scheduling algorithm can potentially move his/her *other* appointments without any negative effect on the waiting time for the patient. Guaranteeing "not-worse" for schedule changes means that patients actually have an incentive to cooperate. In any practical implementation of a (re)scheduler, patient cooperation will be essential to make sure that patients are actually willing and able to come to the hospital at the new appointment time.

We research a multi-agent scheduling method that exploits this opportunity. Each patient-agent acts on behalf of his patient and tries to get the best possible schedule. Agents can bilaterally exchange single appointments between each other to improve individual patient activity schedules, while no patient is worse off than before. In economics such "nobody-worse" improvement is called a Pareto improvement, thus we have a Multi-agent Pareto Appointment EXchanging algorithm: MPAEX.

2 Experiments

Based on practical cases, we develop our hospital patient scheduling model. It represents autonomous departments and resources, as well as individual patients and their activities. We use a semi-dynamic setting

¹This is an abstract of the paper published in the Proceedings of the IEEE Conference on E-Commerce Technology (CEC'06), June 26-29 2006, San Francisco, California. The full paper is available at <http://www.cwi.nl/~vermeule/mpaex.pdf>

to gain fundamental insights and allow comparison with more standard centralized static techniques. A robust simulation of the distribution of workloads over the various resources in a hospital is a crucial aspect in evaluating scheduling solutions for hospital patient scheduling. To assess accurately how useful different scheduling solutions are, we have to be able to consider a large distribution of different workloads. We introduce the use of the Theil index within the hospital scheduling setting to capture the inequality of the workload distribution.

We use initial schedules constructed by either a First Come First Served (FCFS) or First Come Randomly Served (FCRS) dispatching rule. Current practice will fall somewhere between FCFS and FCRS, depending on the actual hospital situation. We compare the scheduling performance of our decentralized dynamic MPAEX approach to the best centralized static heuristics (set to 1) for a range of Theil indices, see Figure 1.

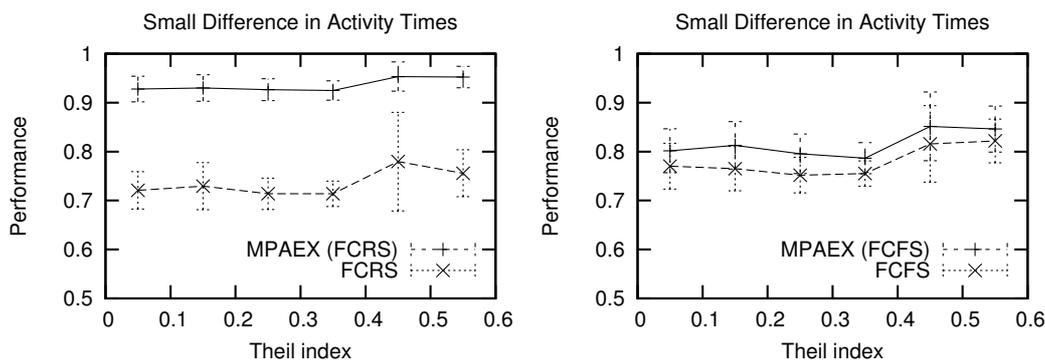


Figure 1: Averages and standard deviation of the relative performances compared to the best centralized static heuristic, for different Theil ranges.

The decentralized, dynamic approach of MPAEX performs very close to the best centralized static heuristics. We find that the MPAEX(FCRS) distributed scheduling approach obtains between 92% and 95% of the best performance of the best centralized heuristic. We find that MPAEX(FCRS) outperforms MPAEX(FCFS), although FCFS is the more efficient initial schedule, it also leaves fewer opportunities for MPAEX to improve upon. FCFS leaves more improvement opportunities, and the quality of the final schedule MPAEX(FCRS) is (almost) similar to that of the centralized heuristics. In our opinion, this improvement over MPAEX(FCFS) demonstrates the rescheduling quality of the MPAEX approach.

3 Conclusions

Unlike centralized heuristics for solving scheduling problems, our multi agent approach can straightforwardly be used in a dynamic environment. Agents can interact concurrently, with local dynamic information (such as: cancellations, disruptions, expired resource constraints), information does not need to be centrally collected before decisions are made. In health care, preferences on resource utilization and preferences of patients are inherently distributed. Multi agent systems can capture distributed preferences, such that agents act within the rules of scheduling, but try to optimize the schedule for their owner according to the preferences. They will not accept a schedule that is worse than its current one. The setup we developed also provides a proper setting for future approaches and solutions.

References

- [1] T. O. Paulussen, N. R. Jennings, K. Decker, and A. Heinzl. Distributed patient scheduling in hospitals. In *Proc. 18th IJCAI*, 2003.
- [2] M.P. Wellman, W. Walsh, P. Wurman, and J.K. MacKie-Mason. Auction protocols for decentralized scheduling. *Games and Economic Behaviour* 35, 2001.

IDENTIFYING AN AUTOMATON MODEL FOR TIMED DATA

Sicco Verwer Mathijs de Weerd Cees Witteveen
Delft University of Technology, P.O.Box 5031 2600GA Delft

The full version of this paper appeared in: Proceedings of the Fifteenth Annual Machine Learning Conference of Belgium and the Netherlands (Benelearn 2006).

1 Introduction

In our paper we focus on learning systems of which the execution is determined by a finite set of discrete events. Such a system is known as a discrete event system (DES) [2]. A common way to model discrete event systems is by using deterministic finite state automata (DFA). The problem we study can be stated as follows: *Given a data sample of observations, how to identify the correct model for a specific DES.*

When observing a real-world system, however, there often is information in addition to the system events, namely, their times of occurrence. In other words, the data consists of *timed strings* of events, instead of just strings of events. If this time information is important, a DFA is too limited. For example, it is impossible to distinguish between events that occur quickly after each other, and events that occur after each other with a significant delay between them.

Therefore, we propose a simple type of timed automaton [1] to model DES where the timing between two consecutive events is important. We call this type of automaton a delay automaton (DA). The structure of a DA is different from a DFA only in the transitions it uses:

Definition 1.1 *A transition $t \in T$ of a DA is a tuple $\langle q, q', s, \phi \rangle$, where q, q' are the source and target states, s is a symbol, and ϕ is a delay guard defined by an interval in \mathbb{R}^+ . A delay guard ϕ is satisfied by a time delay $d \in \mathbb{R}^+$ if $d \in \phi$.*

A transition $\langle q, q', s, \phi \rangle$ of a DA is interpreted as follows: whenever the automaton is in state q , reading s , and the delay guard ϕ is satisfied by the current delay, then the machine will move to state q' . Thus, in a DA it is not only possible to activate a transition to another state, but it is also allowed to remain in the same state for some time (delay).

The DA in Figure 1 models a ‘smart’ automatic bike light that does not turn the light off when the bike stops for a traffic light. The execution of this DA starts in the state *off*. A stop at a traffic light should not last longer than one minute. This is modeled by a *light off* event only occurring at time 60. A *start* event before 60 seconds will make the automaton return to the *on* state.

In our paper we prove that Learning a DA is NP-complete by a reduction from the problem of learning a DFA. Based on this reduction, we show how the currently best learning algorithm for DFAs (state merging [3]) can be adapted to deal with time information. The algorithm works as follows:

- Construct a timed augmented prefix tree acceptor (TAPTA) from an input sample. An APTA is a DA, shaped like a tree, that accepts the positive strings from the input sample, and rejects the negative strings.

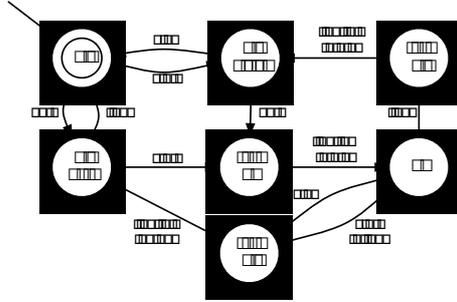


Figure 1: This DA models a ‘smart’ bike light that leaves the light on when the bike stops for less than 60 seconds.

- Each transition in the TAPTA is given as its delay guard the degenerate interval (of length zero) of exactly the delay value of the timed sample string used to create the transition.
- Continuously merge the states of the TAPTA in the normal way. But, also merge the intervals into larger intervals: If two states are merged and this causes two transitions with the same label to point to the same state, then merge the intervals into one. The result of a merge of two intervals i and i' is an interval from the lowest lower-bound to the highest upper-bound of i and i' .

This algorithm can be used to learn a DA from positive and negative data. As far as we know, currently no other learning algorithm exists to identify a timed automaton from a timed sample.

References

- [1] Rajeev Alur. Timed automata. In *International Conference on Computer-Aided Verification*, pages 8–22. Springer-Verlag, 1999.
- [2] Christoc G. Cassandras and Stephane Lafortune. *Introduction to Discrete Event Systems*, volume 11 of *The Kluwer International Series on Discrete Event Dynamic Systems*. Springer Verlag, 1999.
- [3] Kevin J. Lang, Barak A. Pearlmutter, and Rodney A. Price. Results of the abbadingo one DFA learning competition and a new evidence-driven state merging algorithm. In *ICGI*, volume 1433 of *Lecture Notes in Computer Science*, pages 1–12, 1998.

PROCESSING-BASED CONCEPT KINDS FOR ACTOR-AGENT COMMUNITIES

Manuela Viezzer^{ab} Niek Wijngaards^b Masja Kempen^b

^a *The University of Birmingham, B15 2TT Birmingham, UK*

^b *DECIS Lab, Thales R&T Netherlands, P.O. Box 90, NL-2600 AB Delft*

1 Extended abstract

The extended version of this paper has been accepted at the Context Representation and Reasoning Workshop of ECAI06. In press.¹

In the near future, humans and artificial systems are foreseen to engage in (even closer) collaboration. These organisations are named *actor-agent communities*, in which collaboration between multiple participants, including humans and artificial systems, takes place for the realization of a common mission or for the support of a shared process [10].

Actor-agent communities (AACs) require analysis as a whole, including the triple: actors, agents and their ‘niche’ [5, 8], where niche is characterised by requirements, constraints and opportunities. The intense collaboration between actors and agents places additional requirements on the development of artificial agents: the agents need to ‘understand’ humans - as communication is a pre-requisite for collaboration. The identification of concept kinds that can be common to both human and artificial minds is a means via which shared situation awareness and shared meaning can be facilitated. For example, the cognitive architecture of an agent can be designed to support the use and manipulation of such concept kinds.

Unfortunately, despite the wide agreement on the importance of concepts as major constituents of human cognition, no unified, comprehensive and well-established theory of concepts exists, and different empirical findings seem to support different, when not incompatible, views on the matter [6]. Moreover, even though research on concepts has been abundant in many fields, the question of whether there are distinct kinds of concepts has been rarely addressed, because most of the research on concepts has focused almost exclusively on natural object concepts (chair, bird, tool, etc.), within categorisation tasks. One of the few attempts to bring together candidates for kinds of concepts is to be found in [7], where Medin *et al.* propose criteria based on structural differences, processing differences, and content differences for distinguishing concept kinds.

The purpose of this paper is to further our understanding of conceptual structures by addressing the question of whether distinct kinds of concepts can be individuated, and on which basis. The importance of a framework for distinguishing among different concept kinds, including models for acquisition, reasoning, and other manipulations, thereof should not be underestimated. When developing (intelligent) agents in general, it is imperative to choose the right processing mechanisms for the right purposes. If it is known which concept kinds can be distinguished and if the distinction is based upon the existence of processing differences between those kinds, important parts of an agent’s cognitive architecture can be specified. Although it is compelling to look for only one processing mechanism ‘which suits all needs’, pragmatic experience dictates that multiple processing mechanisms within one agent are mandatory as the ‘needs’ can be extremely diverse. It is expected that the same holds for processing mechanisms for manipulation of concepts. Our analysis shows that processing-based distinctions facilitate the identification of concept kinds, basically affordance concepts and goal-based concepts on one side, and taxonomic concepts on the other. The processing mechanisms taken into consideration include reasoning with and acquisition of concepts.

In the paper we examine the categories proposed in [7] and based on structural and processing differences, and show that the only acceptable structural-based distinction actually reduces to a distinction based

¹Contact author: Manuela.Viezzer@decis.nl

on how the involved concept kinds are processed. We then turn to the processing-based criterion, and account for two distinct kinds of concepts: taxonomic concepts on the one hand, and goal-derived concepts on the other. Finally, we describe affordance concepts [4, 3, 2] as a special kind of concepts, and show how affordance concepts can be characterised in processing-based terms, along similar lines as goal-derived concepts [1, 9]. More precisely, we show that affordance concepts and goal-derived concepts can be grouped together as concepts that are learnt in a mediated way, in the sense that clusters of exemplars are not available before the learning process; instead, a prior formation of clustering criteria is required. Both affordance and goal-derived concepts can therefore be opposed to taxonomic concepts, which are exemplar-based. On the other hand, affordance concepts and goal-derived concepts are distinct with respect to the intermediate clustering step that is involved: goal-derived concepts require a process of conceptual combination, while affordances require direct interaction with the environment.

Our main research effort is focused on actor-agent teams AACs. With respect to the role and use of concepts and concept kinds in AACs, we plan the following subsequent research activities. On the one hand we aim to experiment with affordance-based learning approaches to enable an agent to adapt its behaviour to its (changing) simulated environment, albeit at the level of interaction with (physical) objects and properties thereof. In addition, we intend to explore the notion of affordances, to the extent that it can apply to strategic notions, such as those found in the management of teams. This is to be grounded by experimentation with human teams, as well as by experimenting with agents capable of applying affordances at a strategic level.

2 Acknowledgements

The research reported here is part of the Interactive Collaborative Information Systems (ICIS) project (www.icis.decis.nl), supported by the Dutch Ministry of Economic Affairs, grant nr: BSIK03024.

Part of this work was conducted at the School of Computer Science, The University of Birmingham, under the supervision of Prof. A. Sloman, to whom goes a special thanks for his guidance and advice.

References

- [1] I. Cos-Aguilera, L. Ca namero, and G. Hayes. Using a sofml to learn object affordances. In *Proceedings of the Workshop on Physical Agents (WAF04)*, University of Girona, 2004.
- [2] E. J. Gibson. Perceptual learning in development: some basic concepts. *Ecological Psychology*, 12(4):295–302, 2000.
- [3] J. J. Gibson. *The senses considered as perceptual systems*. Houghton Mifflin, Boston, 1966.
- [4] J. J. Gibson. *The ecological approach to visual perception*. Laurence Erlbaum, Hillsdale, NJ, 1979.
- [5] R. R. Hoffman, P. Hayes, K. M. Ford, and P. Hancock. The triples rule. *IEEE Intelligent Systems*, 17(3):62–65, 2002.
- [6] S. Laurence and E. Margolis. Concept and cognitive science. In E. Margolis and S. Laurence, editors, *Concepts: core readings*. The MIT Press, Cambridge, Massachusetts, 1999.
- [7] D. L. Medin, E. B. Lynch, and K. O.Solomon. Are there kinds of concepts? *Annu. Rev. Psychol.*, 51:121–147, 2000.
- [8] A. Sloman. Interacting trajectories in design space and niche space: a philosopher speculates about evolution. In *Lecture Notes in Computer Science*, volume 1917, pages 3–16. Springer-Verlag, Berlin, 2000.
- [9] M. Viezzer. Autonomous concept formation: an architecture-based analysis. Draft PhD. Thesis, 2003.
- [10] N.J.E. Wijngaards, M. Kempen, A. Smit, and K. Nieuwenhuis. Towards sustained team effectiveness. In G. Lindemann, editor, *Selected revised papers from the workshops on Agent, Norms and Institutions for Regulated Multi-Agent Systems (ANIREM) and Organizations and Organization Oriented Programming (OOP) at AAMAS'05*, volume 3913 of *LNCS*, pages 33 – 45. Springer-Verlag, Berlin, 2006. In press.

Stochastic Optimal Control of Continuous Space-Time Multi-Agent Systems

Wim Wiegerinck Bart van den Broek Bert Kappen

*SNN, Radboud University Nijmegen
6525 EZ Nijmegen, The Netherlands*

{w. wiegerinck, b. vandenbroek, b. kappen}@science.ru.nl

A collaborative multi-agent system (MAS) is a group of agents that autonomously control their behavior to achieve a common goal or to maximize the performance of the group. Examples are teams of soccer-robots and teams of unmanned rescue vehicles in a hazardous disaster area. In practical applications, a MAS often has to deal with uncertainty in the environment and limitations of its resources.

In our full paper [7], we are interested in optimal control in such systems. We focus on systems in which agents in a stochastic environment have to distribute themselves efficiently over a number of targets. For example, consider a system of n firefighter-agents and fires. The agents are at some initial positions and should reach the fires positions in the most efficient way, see Figure 1. The MAS should continuously control itself such that in the end one of these $n!$ configurations is realized at minimal expected effort. The additional complexity is that due to the noise in the dynamics, a configuration that seems optimal from the initial positions may become suboptimal in a later stage.

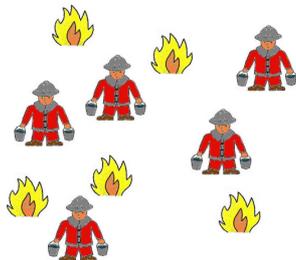


Figure 1: The firemen problem. A number of firemen go to a number of fires, each to a different one. How should the agents coordinate when it is not yet decided to which fire each agent should go, and when the actions of the agents are subject to noise?

A common approach is to model such a system as a Markov Decision Process (MDP) in discrete space and time: the optimal actions in an MDP optimization problem are in principle solved by backward dynamic programming. Since both the joint action space and the joint state space of the agents are assumed to be large in the discretization, and increase exponentially in the number of agents, simply taking a basic dynamic programming approach to solve the MDP will generally be infeasible [1].

Typically one can describe the system more compactly as a factored MDP. In such systems both the transition probabilities and reward functions have some structure. Unfortunately, this structure is not conserved in the value functions and exact computation remains exponential in the system size. Recently, a number of advanced and powerful approximate methods have been pro-

posed. The common denominator of these approaches is that they basically assume some predefined approximate structure of the value functions [2, 3].

In our paper, we take a different starting point. Rather than discretizing, we will consider the stochastic optimal control problem in continuous space and time. As in discrete MDPs, this optimization problem is in principle solved by backward dynamic programming. Usually this optimization is intractable. However, if (1) both the noise and the control are additive to the (non-linear) dynamics, (2) the increment in cost is quadratic in the control, and (3) the noise satisfies certain additional conditions, then it can be shown that the stochastic optimization problem can be transformed into a linear partial differential equation, which can be solved by forward stochastic integration of a diffusion process [4, 5]. This formalism contains linear-quadratic control as a special case [6].

An interesting observation in [4, 5] is the phenomenon of symmetry breaking in multi-modal systems, i.e. in problems where several local minima of the cost co-exist. This symmetry breaking manifests itself as a delayed choice, keeping options open and using the fact that the noise may help to come closer to one of the options at no additional cost.

Formally the extension of this formalism to cooperative MAS is straightforward. The question that we ask ourselves, is how the formalism scales from the single-agent single-target situation (e.g. one fireman has to go to a given fire) to a collaborative system of n agents and m targets. Although the dynamics of the agents is assumed to be independent, with optimal control the behavior of the agents will be coupled in a non-trivial way in order to reach an end-configuration at minimal cost. We show that the solution can be expressed in terms of a partition sum, hence the problem is equivalent to a graphical model inference problem. The graphical model consists of n variables, with n the number of agents, and each variable can be in m states, where m the number of targets. We illustrate the framework with simulation results of stochastic optimal control in two toy problems.

Acknowledgments

This research is part of the Interactive Collaborative Information Systems (ICIS) project, supported by the Dutch Ministry of Economic Affairs, grant BSIK03024.

References

- [1] C. Boutilier. Planning, learning and coordination in multiagent decision processes. In *TARK*, volume 6, pages 195–210, 1996.
- [2] Carlos Guestrin, Daphne Koller, and Ronald Parr. Multiagent planning with factored MDPs. *Proc. NIPS*, 14:1523–1530, 2002.
- [3] Carlos Guestrin, Shobha Venkataraman, and Daphne Koller. Context-specific multiagent coordination and planning with factored MDPs. *Eighteenth National Conference on Artificial Intelligence (AAAI 2002)*, pages 253–259, 2002.
- [4] H. J. Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and experiment*, page P11011, November 2005.
- [5] Hilbert J. Kappen. Linear theory for control of nonlinear stochastic systems. *Physical Review Letters*, 95(20):200201, November 2005.
- [6] R. Stengel. *Optimal Control and Estimation*. Dover, New York, 1993.
- [7] W. Wiegerinck, B. van den Broek, and B. Kappen. Stochastic optimal control in continuous space-time multi-agent systems. In *UAI'06*, 2006.

Project-Join-Repair: An Approach to Consistent Query Answering Under Functional Dependencies

Jef Wijsen

University of Mons-Hainaut, Belgium

Abstract

With the aim of getting more consistent information out of an inconsistent database, a new approach to consistent query answering under functional dependencies is proposed. Conditions are given under which consistent answers can be computed in tractable time.

1 Motivation

In the Internet age, data are ubiquitous and cannot be expected to be globally consistent. The database systems in which such data are stored and queried, should be capable of handling this *inconsistency* phenomenon. A way to deal with the problem is to rectify the database before proceeding to queries. Since there is usually no single best way to solve an inconsistency, we will generally end up with a set of possible *database repairs*. Such set of possible databases defines an *incomplete database*, a concept that has been studied for a long time [4]. When a query is asked on an incomplete database, the *certain query answer* is defined as the intersection of the answers to the query on every possible database. The motivation for intersecting query answers should be clear: although we do not know which database is the “right” one, we do know that it will (at least) return this certain answer. If the incomplete database is made up of repairs, the certain answer has also been called the *consistent answer*.

The problem of computing consistent query answers gained much attention since the seminal paper by Arenas et al. [2]. We address this problem for universal relations that are inconsistent with respect to a set of functional dependencies (fd’s). Repairing with respect to functional and key dependencies is commonly done by tuple deletion (inserting new tuples will not take away the inconsistency anyway). Unfortunately, when we delete a tuple because it contains an error, we also lose the correct information contained in the tuple. To solve this information loss problem, a novel approach, termed “project-join-repair,” is proposed for repairing a universal relation subject to a set of fd’s. Intuitively, the idea is to replicate correct values over different tuples so that they will not be lost in a single tuple deletion. Technically, the replication is achieved by applying a join dependency [1]. We illustrate the approach by an example.

The first row in the following relation states that on January 7, twenty units of product P1 have been shipped to customer C1 called A. Jones. The constraints are $Cid \rightarrow CName$ (the same identifier cannot be used for different customers) and, since quantities are daily totals per product and client, $\{Date, Pid, Cid\} \rightarrow Qty$. The first fd is violated, because C1 appears with two different names. We could repair this relation by deleting either the first or the second tuple. However, it may be more meaningful to assume that names are mistaken, and that “A. Johnson” should read “A. Jones,” or *vice versa*.

I	Date	Pid	Qty	Cid	CName
	7 Jan	P1	20	C1	A. Jones
	8 Feb	P2	15	C1	A. Johnson

We propose a novel way to make the intended rectification. First, we apply the join dependency $\bowtie [\{Date, Pid, Qty, Cid\}, \{Cid, CName\}]$, that is, we take the join of the projections on $\{Date, Pid, Qty, Cid\}$ and $\{Cid, CName\}$, which results in the following table:

$\pi_{\text{Date,Pid,Qty,Cid}}(I)$	Date	Pid	Qty	Cid	CName	
\bowtie	7 Jan	P1	20	C1	A. Jones	
$\pi_{\text{Cid,CName}}(I)$	8 Feb	P2	15	C1	A. Johnson	
	7 Jan	P1	20	C1	A. Johnson	(*)
	8 Feb	P2	15	C1	A. Jones	(*)

This join dependency (jd) corresponds to a lossless-join decomposition in third normal form (3NF). The lossless-join property means that applying the jd will not insert new tuples into consistent relations. However, since one fd is violated in our example, the join contains two new tuples (followed by *). Next, we take as repairs the maximal (under set inclusion) consistent subsets of the join relation. This gives us the two intended repairs:

J_1	Date	Pid	Qty	Cid	CName	J_2	Date	Pid	Qty	Cid	CName
	7 Jan	P1	20	C1	A. Jones		8 Feb	P2	15	C1	A. Johnson
	8 Feb	P2	15	C1	A. Jones		7 Jan	P1	20	C1	A. Johnson

The effect of the project-join-repair resembles database repairing by value modification, proposed in [5].

2 Problem Statement

The formal problem statement is as follows. Let Σ be a set of fd's and I a database relation. A *repair* of I under Σ is a maximal (under set inclusion) relation $J \subseteq I$ such that $J \models \Sigma$. The *consistent query answer* to a query q on I under Σ , denoted $q_\Sigma(I)$, is defined as follows:

$$q_\Sigma(I) = \bigcap \{q(J) \mid J \text{ is a repair of } I \text{ under } \Sigma\} .$$

Given a set Σ of fd's and a query q , consistent query answering is the complexity of (testing membership of) the set [3]:

$$\text{CQA}(\Sigma, q) = \{I \mid I \text{ is a ground relation and } q_\Sigma(I) = \emptyset\} .$$

Clearly, computing the consistent query answer is at least as difficult as testing its emptiness. Note that the set Σ of constraints and the query q are fixed, and the complexity is in the size of the input relation, known as "data complexity." We study a novel related problem. Let Σ be a set of fd's and σ a jd such that $\Sigma \models \sigma$:

$$\text{CQAJD}(\Sigma, \sigma, q) = \{I \mid I \text{ is a ground relation and } q_\Sigma(\sigma(I)) = \emptyset\} .$$

Here, $\sigma(I)$ is the smallest (under set inclusion) relation that contains I and satisfies σ . Intuitively, applying the jd can be thought of as repairing by tuple insertion relative to the weaker constraint σ (weaker in the sense that $\Sigma \models \sigma$). Next, we apply standard consistent query answering relative to Σ using tuple deletion.

The extended version of this two-page abstract [6] identifies restrictions on Σ , σ , and q under which $\text{CQAJD}(\Sigma, \sigma, q)$ is in **P**. It is also shown that the same restrictions do not guarantee tractability of $\text{CQA}(\Sigma, \sigma)$. This is an interesting result: by applying the project-join-repair approach, we can preserve more information and gain tractability at the same time.

References

- [1] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.
- [2] M. Arenas, L. E. Bertossi, and J. Chomicki. Consistent query answers in inconsistent databases. In *Proc. 18th ACM Symp. on Principles of Database Systems*, pages 68–79. ACM Press, 1999.
- [3] J. Chomicki and J. Marcinkowski. Minimal-change integrity maintenance using tuple deletions. *Information and Computation*, 197(1-2):90–121, 2005.
- [4] T. Imielinski and W. Lipski Jr. Incomplete information in relational databases. *J. ACM*, 31(4):761–791, 1984.
- [5] J. Wijsen. Database repairing using updates. *ACM Trans. Database Syst.*, 30(3):722–768, 2005.
- [6] J. Wijsen. Project-join-repair: An approach to consistent query answering under functional dependencies. In *FQAS*, volume 4027 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

Demonstrations
and
Applications

The Agent Reputation and Trust (ART) Testbed

Karen K. Fullam^a Tomas Klos^b Guillaume Muller^c Jordi Sabater-Mir^d
K. Suzanne Barber^a Laurent Vercouter^c

^a *The University of Texas at Austin, USA*

^b *Center for Mathematics and Computer Science (CWI),
Amsterdam, The Netherlands*

^c *École Nationale Supérieure des Mines, Saint-Étienne, France*

^d *Artificial Intelligence Research Institute (IIIA-CSIC), Barcelona, Spain*

1 The Agent Reputation and Trust (ART) Testbed

The Agent Reputation and Trust (ART) Testbed [1] provides functionality for researchers of trust and reputation in multi-agent systems. As a versatile, universal experimentation site, the ART Testbed scopes relevant trust research problems and unites researchers toward solutions via unified experimentation methods. Through objective, well-defined metrics, the testbed provides researchers with tools for comparing and validating their approaches. As such, the testbed operates in two modes: competition and experimentation. In **competition** mode, each participating researcher controls a single agent, which works in competition against every other agent in the system. At the BNAIC 2006 conference, we will demonstrate the ART Testbed with a variety of agents (i.e., participants in the First International ART Testbed Competition, held at AAMAS 2006). To utilize the testbed's **experimentation** mode, the Testbed is downloadable for researcher use independent of the competition [4]: results may be compared among researchers for benchmarking purposes, since the testbed provides a well-established environment for easily-repeatable experimentation.

1.1 Testbed Domain Problem

The testbed operates in an art appraisal domain (see [2] for a detailed justification), where researchers' agents function as painting appraisers with varying levels of expertise in different artistic eras. Clients request appraisals for paintings from different eras; if an appraiser does not have the expertise to complete the appraisal, it may purchase opinions from other appraisers. Other appraisers estimate the accuracy of opinions they send by the cost they choose to invest in generating an opinion, and opinion providers may lie about the estimated accuracy of their opinions. Appraisers produce appraisals using their own opinion and opinions received from other appraisers, receiving more clients, and thus more profit, for producing more accurate appraisals. They may also purchase reputation information from each other about third-party agents. Appraisers attempt to accurately value their assigned paintings; their decisions about which opinion providers to trust directly impact the accuracy of their final appraisals. In competition mode, the winning agent is selected as the appraiser with the highest bank account balance.

1.2 Testbed Architecture

The testbed architecture, implemented in Java, consists of several components (see [3] for a detailed description of the ART Testbed architecture). The Testbed Server manages the initiation of all games by starting a Simulation Engine for each game. The Simulation Engine is responsible for controlling the simulation environment by enforcing chosen parameters. In each timestep, the Simulation Engine assigns clients with paintings to each appraiser. Then appraisers conduct reputation and opinion transactions with each other if they so desire. Finally, the Simulation Engine assesses each appraiser's accuracy based on the opinions

the appraiser purchases and the ‘weights’ the appraiser places on those opinions. Weights are real values between zero and one that an appraiser assigns, based on its trust model, to another’s opinion.

Through the Simulation Engine, the Database collects environment and agent data, such as true painting values, opinions, transaction messages, calculated final appraisals, client share allocations, and bank balances. With access tools for navigating, downloading/uploading, and replaying Database logs, data sets are made available to researchers after each game session for game re-creation and experimental analysis.

User Interfaces permit researchers to observe games in progress and access information collected in the Database by graphically displaying details. Figure 1 shows the Game Monitor Interface, by which observers

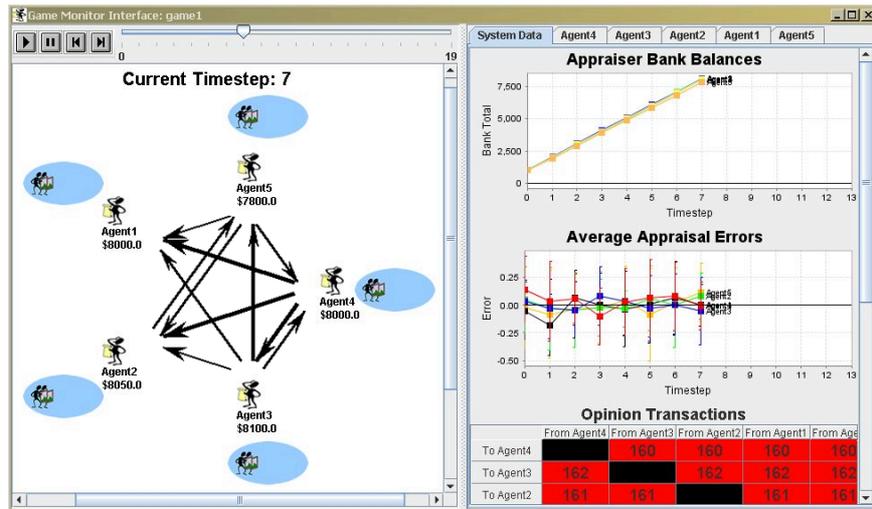


Figure 1: The Game Monitor Interface for viewing game data.

can view opinion and reputation transactions between agents on the left and detailed statistics, such as bank balance, about each appraiser agent on the right. The Game Monitor Interface’s play-pause buttons permit games to be played and replayed, regardless of whether the game is in progress or completed.

Finally, the abstract Agent class is designed to allow researchers to easily implant customized internal trust representations and trust revision algorithms while permitting standardized communication protocols with entities external to the appraiser agent. Users simply create a class inheriting from the Agent class, implementing a method for each of the agent’s necessary strategic decisions. The abstract class Agent handles all required inter-agent communication, as well as communication between agents and the Simulation Engine.

References

- [1] ART Testbed Team. *Agent Reputation and Trust Testbed Website*. <http://www.art-testbed.net/>, 2006.
- [2] K. Fullam, T. Klos, G. Muller, J. Sabater-Mir, A. Schlosser, Z. Topol, K. S. Barber, J. S. Rosenschein, L. Vercouter, and M. Voss. A Specification of the Agent Reputation and Trust (ART) Testbed In *Proc. AAMAS*, 2005, pp. 512–518.
- [3] K. Fullam, T. Klos, G. Muller, J. Sabater-Mir, Z. Topol, K. S. Barber, J. S. Rosenschein, and L. Vercouter. The Agent Reputation and Trust (ART) Testbed Architecture In *Proc. Trust Workshop at AAMAS*, 2005, pp. 50–62.
- [4] ART Testbed Team. *ART Testbed SourceForge project page*. <https://sourceforge.net/projects/art-testbed>, 2006.

Faceted Access to Heterogeneous Cultural Heritage Collections using Semantic Web Techniques

Marjolein van Gendt Antoine Isaac Lourens van der Meij
Stefan Schlobach

Vrije Universiteit Amsterdam
Koninklijke Bibliotheek, Den Haag
Max Planck Institute for Psycholinguistics, Nijmegen

Abstract

Integrated digital access to multiple collections is a prominent issue for many Cultural Heritage institutions. The metadata describing diverse collections must be interoperable, which requires aligning the controlled vocabularies that are used to annotate objects in these collections. We demonstrate an interface prototype presenting two collections whose vocabularies have been matched applying Knowledge Representation techniques as established in recent Semantic Web research. This prototype features three different views that enable the user to browse the two collections using the alignment, while still providing her with the original vocabulary structures.

1 STITCH general goals and pilot experimentation

STITCH – Semantic Interoperability To access Cultural Heritage – is a research project within CATCH, a programme funded by the Netherlands Organisation for Scientific Research – NWO. The objective of the project is to evaluate methods and tools from the Semantic Web research area for integrating Cultural Heritage collections.

Our first experiment and implementation aimed at providing integrated access to two heterogeneous collections, the Illuminated Manuscript collection¹ from the Dutch National Library in The Hague, and the ARIA Masterpieces collection² from the Rijksmuseum in Amsterdam.

First a conversion to generic Semantic Web formats, such as RDF(S)³ and SKOS⁴, was required. Second, having computer-readable representations, we could align them. We turned to two off-the-shelf ontology mappers (S-Match[1] and Falcon[3]) and evaluated their use for aligning CH thesauri. Third, automatically found correspondences were used in an interface we implemented to browse different vocabularies and to retrieve documents from several collections.

In this demonstration, we will recall the first steps⁵, but the emphasis will be on the browsing features offered by our interface.

2 Three faceted views on integrated collections

We implemented a multi-faceted browsing (MFB) interface to evaluate and explore the results of our mapping effort. MFB involves constraining search criteria along – usually orthogonal – aspects of a collection called *Facets*. Here we adapted this paradigm in an atypical way, since we used one category (the subject annotation) for defining several facets.

¹<http://www.kb.nl/kb/manuscripts/>

²<http://www.rijksmuseum.nl/collectie/index.jsp?lang=en>

³<http://www.w3.org/RDF/>

⁴<http://www.w3.org/2004/02/skos/>

⁵More details on these experiments and the lessons we learned from them can be found in [4].

For searching through the integrated collections we explored three different views on integrated collections: *single*, *combined*, and *merged*. All views take into account the links between the vocabularies established in the automatic alignment process.

The *Single View* presents the integrated collections from the perspective of only one of the collections. Elements of the other collection are found through the links between their subject annotations and the concepts of the current view.

The *Combined View* provides simultaneous access to the collections through their respective vocabularies. This allows us to browse through the integrated collections as if it was a single collection indexed against two vocabularies.

The *Merged View* provides access to the collections through a merged thesaurus combining both original vocabularies into a single one.

3 Technical details

The design of our browser was inspired by the Flamenco search interface framework [2]. It is implemented as a web server written in SWI-Prolog, using its HTTP server and HTML code generation libraries. All data is stored in an external RDF Sesame repository⁶ which is accessed using SerQL queries. The server software runs on Linux and MS Windows operating systems.

The focus for the prototype was on easy configurability and adaptability for experimentation using Semantic Web Techniques. In order to easily adapt configurations to new ideas concerning data modeling, all collection, vocabulary and even website configuration information is stored in the RDF repository. As basic representation blocks, we have instances of a `Facet` class, that we introduced to define all aspects of a single facet, such as the SKOS `ConceptScheme` it uses as structured vocabulary. Website configuration objects – instances of the class `SiteConfiguration` – then group together `Facets` into separate websites. New configurations can thus be specified very easily, which enabled us to define sites that use different sets of alignments between the two vocabularies. This also allows us to seamlessly plug in new collections, vocabularies and alignments. Our prototype thus is a proof-of-concept of the way Cultural Heritage institutes can benefit from integrating their collections on the basis of recently established Artificial Intelligence techniques.

The browser and the underlying collections can be found at <http://stitch.cs.vu.nl/demo>. The presentation, a brief explanation of the collection background and the formalisation and alignment process followed by the demonstration of the prototype, will not exceed 25 minutes.

References

- [1] Giunchiglia, F., Shvaiko, P., and Yatskevich, M. Semantic Schema Matching. In *13th International Conference on Cooperative Information Systems (CoopIS 2005)*, Agia Napa, Cyprus, 2005.
- [2] Hearst, M., English, J., Sinha, R., Swearingen, K., and Yee, P. Finding the Flow in Web Site Search. *Communications of the ACM*, 45(9), 2002.
- [3] Jian, N., Hu, W., Cheng, G., and Qu, Y. Falcon-AO: Aligning Ontologies with Falcon. In *K-CAP Workshop on Integrating Ontologies*, Banff, Canada, 2005.
- [4] van Gendt, M., Isaac, A., van der Meij, L., and Schlobach S. Semantic Web Techniques for Multiple Views on Heterogeneous Collections: a Case Study. In *10th European Conference on Research and Advanced Technology for Digital Libraries (ECDL 2006)*, Alicante, Spain, 2006.

⁶Available on <http://www.openrdf.org>.

Darwin at Home

A Hybrid Fitness Function

Gerald de Jong

Almende BV, Westerstraat 50, 3016 DJ Rotterdam, NL

Abstract

This demonstration will show the evolution of both the morphology and the muscle coordination of elastic bodies by means of a hybrid fitness function involving both an algorithm and human intuition. The project intends to make this experience accessible to a potentially large number of people on the Internet so that it might also reach outside of the academic community, hopefully generating new interest for evolutionary algorithms, and a broader understanding of what they can do.

1. An Experiment in Co-Design

Much progress has been made in our efforts to create software that exhibits some form of intelligence. Interpreting and mimicing biological evolution with our own genetic algorithms has given us software capable of discovering solutions in domains where we find ourselves unable to search painstakingly enough. The degree of richness and effectiveness of the algorithms depends on the robustness of their fitness function and the appropriate tuning of their mutation process and perhaps crossover mechanism.

As we gather more hands-on experience working with these genetic algorithms to accomplish particular goals, we often encounter similar challenges such as tendencies to get caught in local maxima. In such cases, we tend to return to the drawing board in order to invent algorithmic ways to avoid stagnation, and algorithmic stagnation-avoidance tends to conflict with the gradual refinement that works so well locally.

The fact that biological evolution seems to have managed the balance between stagnation-avoidance and refinement does not necessarily imply that our virtual algorithms must also be centered around straddling this divide. We can also try to harness the subtle proficiency of the human mind and combine it with the tireless optimization abilities of simpler genetic algorithms.

The Darwin at Home project proposes that we explore the effectiveness of separating these two concerns into a hybrid fitness function: part algorithm, part human. The software is available to everyone on the Internet [1] so that the optimization can be performed in parallel, and special attention has been paid to making the user interface intuitive and accessible, since the human participants make up an important part of the fitness function.

2. Demonstration

Darwin at Home represents further development and a new direction for the Fluidiom project [2], winner of the BNVKI best demonstration award at BNAIC 2005. The Fluidiom software allowed users to create bodies made of muscular elastic intervals and then submit them to a genetic algorithm to evolve locomotive ability. Darwin at Home introduces structural growth mutations and thereby mixes the creation of body structures with the algorithmic optimization of their behavior.

The demonstration of Darwin at Home will involve audience participation. A remote bluetooth mouse will be passed around among audience members so that they can take part in the evolution of the running bodies that they observe on the projection screen by making survival choices. At the same time that these choices are being made, the genetic algorithm is working to ensure the survival of the most accomplished runners. Interesting results flow from the interplay between the human and the algorithm because each of these two informs the other, and especially because both are active simultaneously.

A fitness function informed by human choices pays attention to subtle issues that can be extremely difficult to express in terms of software instructions. Based on how the algorithm is perceived to be working, humans can make survival choices that assist the algorithm to escape local maxima on the fitness landscape. This demonstration provokes audience members to make choices that can assist the algorithm to accomplish its goal, while the algorithm provides immediate visual feedback about the effects of those choices.

3. Future Work: A-Life

Darwin at Home has expanded its presentation strategy with video offerings as well as an audio podcast, both freely available at <http://www.darwinathome.org>, and the efforts to reach out to new people will continue to be refined.

Future work will be to mold the software into a more complete artificial life scenario in which the faculty of locomotion will play a key role in a competitive/cooperative environment involving a population sharing the same finite resources. Since the bodies are composed of “elastic intervals” it may, for example, be interesting to have them “eat” intervals to survive, and only be able to replicate if they have eaten enough intervals.

References

- [1] <http://www.darwinathome.org>
- [2] Gerald de Jong. Fluidiom: Evolution of Locomotion. *BNAIC 2005 Proceedings*.

SML: A Stream Mining Library

(Demonstration Proposal)

Wojtek Kowalczyk

Vrije Universiteit Amsterdam, De Boelelaan 1081A, 1081HV Amsterdam

Abstract

We will present a library of modules for real-time analysis of data streams that are generated by internet or intranet traffic. Our library, called SML, can be used for rapid development of prototype systems for detecting anomalies in web access, detection of misuse of information systems, fraud detection, or, in general, detection of cybercrime attacks. It uses several very efficient data structures, such as: hash tables, balanced binary search trees, and priority queues, that allow for very fast processing of incoming records. Systems developed with help of SML can monitor in real-time millions of entities (accounts, IP-addresses, users, etc.), processing hundreds of thousands of records per second.

1 Introduction

Computer networks such as the Internet, intranets, wireless networks, sensor networks, company networks, etc., generate very large streams of data. These data streams can be analyzed “on-the-fly”, providing useful information that could be used for various purposes, such as network intrusion detection, detection of fraudulent electronic transactions, real-time marketing, or detection of misuse of information systems. Building systems that are able to analyze such data streams is very challenging and recently a new area of research, Data Stream Mining, emerged.

To contribute to this new field of research, the Computational Intelligence Group of the Vrije Universiteit Amsterdam, has launched, together with several industrial partners, a research project DIANA, **D**ata **I**nterception and **A**NALysis, see www.cs.vu.nl/ci/DataMine/DIANA for details.

The main objective of this project is to develop a new technology for building adaptive systems that intercept huge streams of data, analyze them in real-time and provide some useful feedback. One element of this technology is a software toolbox, SML, for rapid prototyping of stream mining systems. The toolbox has been implemented in Python and C/C++. At the moment it is mainly used for research purposes and for developing prototype systems that are tested in business environments.

During the presentation we will explain the main components of SML and demonstrate a system that was build with help of SML. This system is used for detecting anomalies in access patterns of several big internet sites.

2 Stream Mining Library

The library consists of several modules that address various aspects of stream mining. They are grouped into the following categories.

2.1 Data Converters

Usually, most fields of input records is in the form of strings of characters: url's, cookies, user names, time stamps, etc. Processing strings is very inefficient: they consume lots of memory and string matching is much slower than comparing numeric data. Therefore, it is very beneficial, both with respect to the required cpu-time and memory, to convert input strings into numbers as soon as they enter the system, reducing in this way the remaining processing steps to number crunching.

The Data Converters module provides various methods for converting strings into numbers and numbers into strings.

2.2 Aggregators

Systems for stream mining often maintain some statistical models, called *profiles* or *signatures*, of the “typical behavior” of every individual user, IP-address, credit card holder, etc. These models are usually represented by a collection of some basic statistics: counts, averages, sums, standard deviations, or by histograms, quantiles, or other, non-parametric representations of probability distributions. The module Aggregators contains an implementation of numerous statistics that can be maintained over a sliding window of data.

2.3 Modelers

In some domains, statistical properties of data are modeled with help of Bayesian Networks, Markov Models, Mixture Models, etc. A collection of such modeling techniques is implemented in the Modelers module.

2.4 Deviation Detectors

The essence of deviation detection is in measuring the difference between “typical behavior” and “recent behavior”. This can be done with help of some statistical tests, such as Kolmogorov-Smirnov test, Kullback-Leibler statistic, Chi^2 test, Kendall’s *tau* statistic, etc. A collection of such tests is implemented in the Deviation Detectors module.

2.5 Timeout-Manager

Some entities have a limited life time. For example, an internet session which contains a number of clicks made within a short time interval, that is followed by a long period of inactivity can be considered to be closed; in such situation some action should be taken (updating statistics, closing the session, freeing memory). The Timeout-Manager takes care of all actions, such as closing an inactive session, that should be performed at a specific time. It uses a priority queue to process most urgent tasks first.

2.6 Monitors

When monitoring millions of entities one is interested only in most unusual ones. Therefore, it is very important to be able to produce, at any moment, the “Top K” most extreme entities, like the top 10 most frequent IP-numbers that were used within the last hour. Instead of (inefficient) re-sorting the IP-frequency table, the Monitors module uses a balanced binary search tree structure (AVL-tree) to maintain the ordered list of frequencies.

2.7 Auxiliary

Additionally, a number of functionalities is implemented in the Auxiliary module. They include functions for communication with external data bases, (MySQL and SQLite), communication with graphical or numeric engines (Matlab), backup and restore utilities, and many others.

Visualization of Agent-Mediated Bargaining over Bundles of Goods

Han Noot, Koye Somefun, Tomas B. Klos, Valentin Robu, Han La Poutré

Center for Mathematics and Computer Science (CWI)

PO Box 94079, 1090 GB Amsterdam, The Netherlands

Abstract

We present a visualization of bargaining processes between a seller and a buyer (or buyers) where negotiations take place over the composition and price of bundles of goods. The negotiations themselves are simulated by software specific for the kind of negotiations. The visualization collects information from the simulation steps in a simulation-specific way and visualizes those steps in a general way. We demonstrate the visualization for two cases: a seller who offers bundles of linearly dependent items to many customers and a seller who repeatedly offers a bundle of non linearly dependent goods to a specific customer meanwhile learning this customer's preferences.

1 The two Bargaining Situations

In the context of the projects ASTA (Autonomous Systems of Trade Agents in E-Commerce) and DEAL (Distributed Engine For Advanced Logistics), research has been performed on agent mediated bargaining over bundles of interdependent goods. We discuss the 2 different approaches taken in these projects below.

1.1 Modeling Customer Preferences Using a Hypercube of Bundles

We consider a seller who sells bundles of goods (or services) to many customers [1]. The seller bargains with each customer individually. A negotiation concerns the selection of a subset from a collection of goods, viz. the bundle, together with a price for that bundle. The negotiation is conducted in an alternating exchange of offers and counter offers, typically initiated by a customer. An example of such a practice may be a seller of bundles of news categories (e.g., politics, finance, economy, sports, arts, etc.) who bargains with his customers.

We developed a procedure that a seller can use to search for mutually beneficial alternative bundles during the negotiation; i.e., the procedure attempts to find *Pareto improvements* by changing the bundle content. It uses both negotiation specific information and aggregate anonymous knowledge (e.g., obtained through aggregate past sales data or expert knowledge) and it automatically outputs an alternative bundle content for the customer's (counter) offer. Bundles are modeled as the vertices of a hypercube, and improvements are sought in the bundles reachable along the hypercube's edges emanating from the vertex representing the bundle currently under negotiation. The procedure uses the ongoing bargaining process to determine *when* to output an alternative bundle (i.e., for which offer), and it uses the aggregate knowledge to assess *which* bundle to output.

1.2 Modeling Customer Preferences Using a Utility Graph of Goods

Here we consider an automated negotiation between a buyer and a seller agent over the composition of a bundle of goods with inter-dependent valuations (more specifically k -additive valuations) [2]. The basic idea of our model is to explore agent strategies which enable (self-interested) buyers and sellers in electronic commerce to find jointly profitable agreements (contract configurations), but without directly revealing too much information about their own preferences.

In this model, dependencies between different items in the utility function of the buyer are modeled in the form of a utility graph. The seller will maintain a model of the buyer in the form of such a graph and use it to compute counter-offers

during the negotiation. The negotiation is however with incomplete information: the buyer does not know the costs for the seller to provide any of the items, while the seller does not know the true preference graph of the buyer: this preference information remains hidden throughout the negotiation. However, the seller will attempt to learn a model of the buyer's preferences from the counter offers he makes.

We assume that the seller does start with an appropriate (maximal) structure of the utility graph of the buyer, meaning he knows where dependencies may occur, but he does not know the strength of these interdependencies (this is learned online). In an extension of this work, we show that the maximal structure for a group (population) of buyers can also be learned based on collaborative filtering. However this demo is for the bilateral negotiation part of our model.

2 The Demonstration, System Requirements

The visualization tool resembles a media player which visualizes successive simulation data in successive frames (see fig 1). Information shown can be bundle composition, selling/asking prices, gains from trade, reservation values and utility graphs with indicated strength of interdependence. The visualization tool is implemented in Java, so it is platform independent and runs on a stand-alone platform. It is intended to be interfaced with bargaining simulation software written in Java too. The duration of the demonstration is 10 minutes.

We will demonstrate the two kinds of bargaining processes, discussed in section 1.

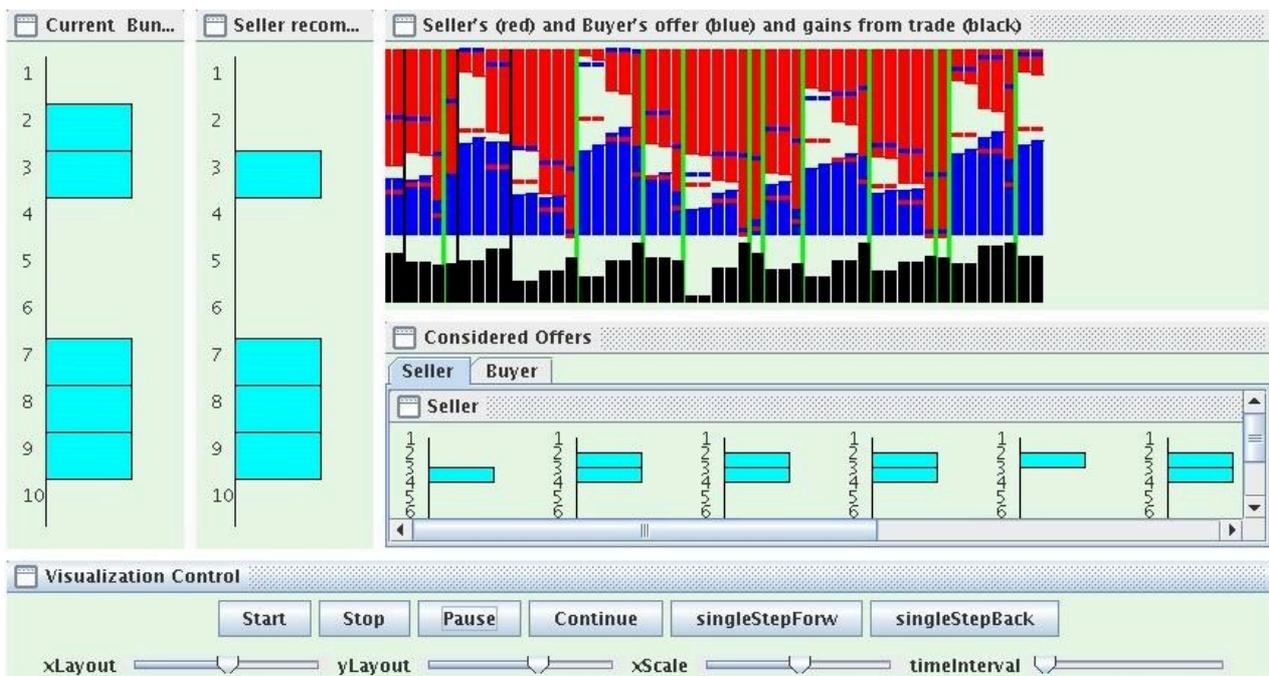


Fig 1 Visualizing bargaining about bundles of goods.

References

- [1] Koye Somefun, Tomas B. Klos, and Han La Poutré. Online learning of aggregate knowledge about nonlinear preferences applied to negotiating prices and bundles. In Marijn Jansen, Henk G. Sol, and René W. Wagenaar, editors, *ICEC04: Proceedings of the Sixth International Conference on Electronic Commerce (ICEC'04)*, pages 361-370. ACM Press, 2004.
- [2] Valentin Robu, Koye Somefun, and Han La Poutré. Modeling Complex Multi-Issue negotiations Using Utility Graphs, *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'05)*, pages 280-287, ACM Press 2005.

Demonstration of Wrapper Learning for Information Extraction Using (k,l) -Contextual Tree Languages

Stefan Raeymaekers

K.U.Leuven, Dept. of Computer Science, Celestijnenlaan 200A, B-3001 Leuven

Abstract

Our system learns a wrapper for extraction of information from web pages, given interactive input from the user. The requested input are counterexamples to equivalence queries that are presented graphically by the system, resulting in a very intuitive system. The learner is based upon (k,l) -contextual tree languages as described in [2]. Our system is optimized towards the task of wrapper induction with the same techniques as described in [2].

1 Introduction

The World Wide Web gives access to a lot of information. Websites allow databases to be queried, resulting in temporary documents generated by scripts. As these documents are formatted in HTML, which is primarily focused on presentation, extracting their content for further processing is difficult. A wrapper is a general name for a procedure that extracts data from a specific set of web pages (often script generated) based on the structure of the pages, commonly without the use of linguistic knowledge. Given that script generated pages are often the only accessible interface into some databases, the demand for suitable wrappers is high. Various tools have been designed to facilitate wrapper building, but the process remains tedious. Hence the research interest to create algorithms that learn wrappers from examples.

We demonstrate our system, an augmented web browser, that allows wrapper induction with a small number of user interactions. Initially the user clicks with the mouse on an example of the data he wants. The system learns a hypothesis and colors all data accordingly. The user can then click on false positives or negatives, until the wrapper is perfected. Typically this requires less than 5 interactions.

The learning algorithm is based on the tree structure of the document instead of its flattened string representation (HTML code). Learning an arbitrary tree/string language is impossible from positive examples only, and expensive from positive and negative examples. Solutions that enable to learn from positive examples only, learn within a subclass of the regular languages. We use as subclass the (k,l) -contextual tree languages. They can be learned very efficiently, but need parameters. An algorithm is used to learn these parameters from negative examples. This combination allows for efficient induction using positive and negative examples, while the class of languages stays sufficiently expressive (as shown in [2]). Below we give a brief explanation of the learning algorithm, which is described in more detail in [2].

2 The Learning Algorithm

The (k,l) -contextual tree languages are defined in similarly as k -contextual languages [1] for strings. Instead of k -grams of a string, (k,l) -forks of a tree are defined, which are local parts of the tree, with a maximal height l and a maximal width k . A set of (k,l) -forks defines a (k,l) -contextual tree language such that a tree is contained in that language iff all its (k,l) -forks are elements of the given set of (k,l) -forks. Learning a (k,l) -contextual tree language from a set of example trees consists in simply collecting all (k,l) -forks of the example trees and use the resulting set to define the learned language. For the same examples, languages with bigger k or l values are subsets of those with smaller k or l values (anti-monotonicity).

For extraction the algorithm learns marked trees, given trees in which one of the target nodes is marked. In the extraction phase, the system can use a language that accepts only correctly marked trees (only target nodes are marked) to retrieve all the required information. To generalize over all possible text strings, all text nodes are replaced by single symbol, except for those strings that the algorithm learns to be necessary context to disambiguate targets.

Given the anti-monotonicity property we can easily find the pair of parameters that produces the smallest language that accepts all positive examples and rejects all negative examples. The search is performed in a front through the parameter space. When an extra example is added, this front can be updated allowing for an efficient incremental algorithm.

This incremental algorithm provides a hypothesis after each interaction. In our implementation, all nodes satisfying the hypothesis are colored. When the user clicks on a colored node, a false positive is indicated, and that node is given as a negative example. When the user clicks on an uncolored node, a false negative is indicated, and that node is given as a positive example. In the figure below, the user clicked 'Call for papers', and the system colored all similar nodes. If the required information is the options in the menu of BNAIC06, the wrapper is finished.



3 Demonstration

Our demo can be run continuously allowing participants to learn their own wrapper for an arbitrary web site. Given the intuitive interface this takes only seconds per wrapper. Extra tools are build into the application that enable the visualisation of trees and their (k,l)-forks. This allows to answer and illustrate specific questions about the learning algorithm. Alternatively these extra tools can be used to give an exposition of a couple of minutes about the learning algorithm and the system. We require only a regular computer, preferably with Internet connection to allow participants to try out arbitrary sites.

References

- [1] S. Muggleton. *Inductive Acquisition of Expert Knowledge*. Addison-Wesley, 1990.
- [2] Stefan Raeymaekers, Maurice Bruynooghe, and Jan Van den Bussche. Learning (k, l)-contextual tree languages for information extraction. In *ECML*, pages 305–316, 2005.

PROCARSUR: A SYSTEM FOR PROGNOSTIC REASONING IN CARDIAC SURGERY DEMONSTRATION

Marion Verduijn ^{a,e} Niels Peek ^a Winston Tjon Sjoer Sjoer ^a
Peter Rosseel ^d Evert de Jonge ^b Bas de Mol ^{c,e}

^a *Dept. of Medical Informatics*, ^b *Dept. of Intensive Care Medicine*, ^c *Dept. of Cardio-thoracic Surgery, Academic Medical Center – University of Amsterdam*

^d *Dept. of Anesthesia and Intensive Care, Amphia Hospital, Breda*

^e *Dept. of Biomedical Engineering, University of Technology, Eindhoven*

We present the *ProCarSur* system for medical prognostic reasoning in the domain of cardiac surgery. The system consists of a prognostic Bayesian network, implemented using off-the-shelf Bayesian network software, and a *task layer*, implemented in Java. The task layer performs a number of prognostic reasoning tasks, and operates as an intermediary between the prognostic Bayesian network and the user; a theoretical paper on this topic has been accepted for a presentation at the BNAIC'06 Conference.

Conventional prognostic systems in medicine are usually based on logistic regression, and their functionality is limited to making one prediction at a single moment in time using a fixed set of predictor variables. Bayesian networks allow for implementing a more general concept of prognosis where variables change from *predictee* (i.e., that what is being predicted) to *predictor* (i.e., that what predicts) as time progresses. This concept of prognosis is realized by prognostic Bayesian networks.

A prognostic Bayesian network is equipped with a predefined set of prognostic reasoning tasks. *Prognostic assessment* and *prognosis updating* are examples of these tasks for (repeated) prediction of the future course and outcomes of the health care process being modeled in the Bayesian network; the impact of future events on the patient's prognosis is simulated in an analysis of *what-if scenarios*. The prognostic tasks can be accomplished by performing 'conventional' probabilistic queries on the Bayesian network, but they generally require that multiple queries be performed and the results be aggregated. The task layer holds procedures to perform these tasks. In general, it translates the physician's prognostic question to probabilistic inference queries for the Bayesian network, and presents the aggregated results of probabilistic inference that are relevant for answering the question to the user. This two-layered modeled supports the prognostic use of Bayesian networks in medical practice.

The ProCarSur system was developed for the Amphia Hospital in Breda, the Netherlands, in a multidisciplinary team, consisting of a medical informatician, a computer scientist, software engineers, and clinical specialists that are involved in the cardiac surgical process: a cardiac surgeon, an anaesthetist, and an ICU physician. Currently, the system has a prototype status and is not used in routine medical care. The intended users are physicians and management staff of departments that are involved in the cardiac surgical process. ProCarSur was developed within the Medicast project in which is worked on a generic platform for the development and implementation of advanced, real-time expert systems to support medical professionals in making clinical decisions; this project was funded by the Ministry of Economic Affairs, the Netherlands.

Figure 1 shows the prognostic Bayesian network that is part of ProCarSur. The network includes 8 variables from the preoperative phase of pre-assessment, 3 operative variables, and 12 physiological and complication variables from the postoperative phase of recovery. In addition, the network contains an outcome variable that represents death during hospitalization (*hospmort*). The network

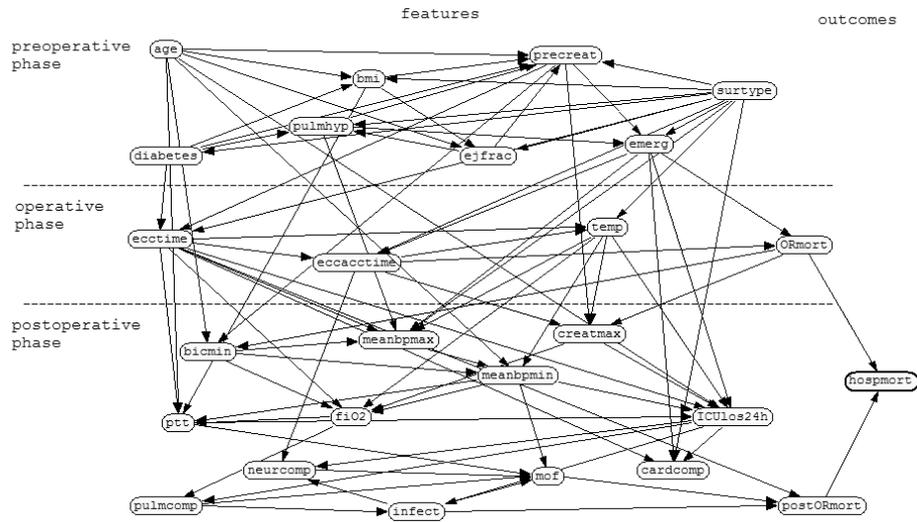


Figure 1: The prognostic Bayesian network that is part of ProCarSur.

was induced from data of 10,114 patients who underwent cardiac surgery in the Amphia Hospital; the learning algorithm that was used is described in the aforementioned theoretical paper. We implemented the network in the Netica software.

The task layer of ProCarSur was written in Java, and the Netica Java-API was used to access the Bayesian network. The user interface is made up of HTML-forms. Figure 2 shows the output screen that presents results for the task of prognosis updating for the variable 'ICU length of stay longer than 24h'. The left pane of the screen shows the menu of the system, the patient profiles as entered by the user is shown in the right upper pane, and the right lower pane shows the results of probabilistic inference. The ProCarSur system runs under Windows XP and requires Netica 1.12, Java Runtime Environment 1.5.0_04, Apache Tomcat 5.5.16, and Internet Explorer 6.0.

During a 15-minute demonstration we will present the main functionalities of the ProCarSur system, using various example cases from the Amphia patient database.

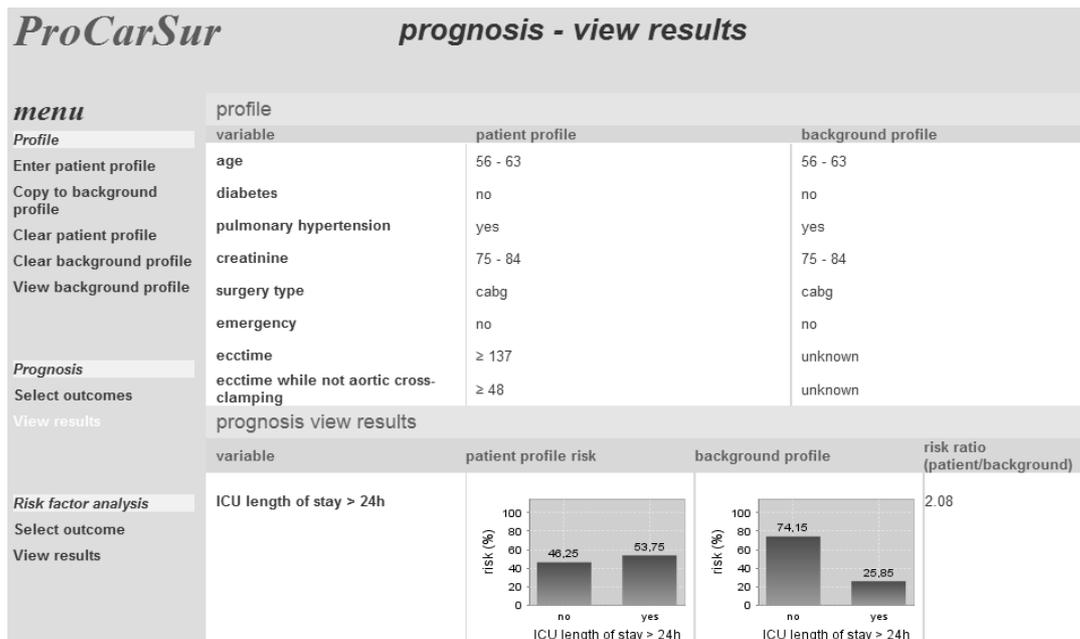


Figure 2: The output screen of the ProCarSur system with results for prognostic updating.