ELSEVIER

Decision Support Systems 34 (2002) 167–175

Decision Support Systems

www.elsevier.com/locate/dsw

Combination of multiple classifiers for the customer's purchase behavior prediction

Eunju Kim [a,*], Wooju Kim [b,1], Yillbyung Lee [a]

[a] Department of Computer Science, Yonsei University, 134, Shinchon-dong, Seodaemoon-ku, Seoul 120-749, South Korea
[b] Department of Industrial Engineering, Chonbuk National University, 664-14 Deokjin, Chonju, Chonbuk 561-756, South Korea

Abstract

In these days, EC companies are eager to learn about their customers using data mining technologies. But the diverse situations of such companies make it difficult to know which is the most effective algorithm for the given problems. Recently, a movement towards combining multiple classifiers has emerged to improve classification results. In this paper, we propose a method for the prediction of the EC customer's purchase behavior by combining multiple classifiers based on genetic algorithm. The method was tested and evaluated using Web data from a leading EC company. We also tested the validity of our approach in general classification problems using handwritten numerals. In both cases, our method shows better performance than individual classifiers and other known combining methods we tried.

Keywords: Purchase behavior prediction; Multiple classifiers; Combination; Genetic algorithm

1. Introduction

Data mining is already becoming one of the popular terms in e-Commerce society because it is generally recognized as one of the most important enablers of EC. Anyone can see that it is very natural process, if they just briefly observe the background of the advent of data mining. Let's list some major necessities of data mining as follows: first, there is a great explosion of data that should be handled or analyzed by many organizations, but they do not have enough capability for such tasks. The second reason is that today's fierce market competition forces companies to compete with other companies to gain knowledge about their customers. The fact that one of the flourishing trends in marketing is one-to-one marketing is another reason for the necessity of data mining. An interesting thing is that all of these reasons for data mining represent exactly the current states of EC-related companies. This is the very reason why EC companies seriously consider data mining as their major competitive edge.

Today, there are already many ways to apply data mining to the business of EC companies [11]. Personalization, one-to-one marketing, and customer relationship management (CRM) are representative examples of such applications. The knowledge that EC companies especially want to know is how to figure out the customer's tendency in their web sites and how to react to and attract those customers to buy their products and services. Identifying the propensity of a specific customer to buy a product

* Corresponding author. Tel.: +82-2-365-4598; fax: +82-2-365-2579.
E-mail addresses: outframe@csai.yonsei.ac.kr (E. Kim), wjkim@chonbuk.ac.kr (W. Kim), yblee@csai.yonsei.ac.kr (Y. Lee).
[1] Tel. +82-63-270-2332; fax: +82-63-270-2333.

0167-9236/02/$ - see front matter © 2002 Elsevier Science B.V. All rights reserved.
PII: S0167-9236(02)00079-9

is categorized as a type of the knowledge mentioned above and it is also regarded as an important basis for driving personalization and one-to-one marketing effectively and dynamically. Therefore, making predictions of a customer's purchase behavior more accurate cannot be emphasized enough. So, we propose a methodology to enhance the accuracy in predicting the propensity of customer purchase by combining multiple classifiers based on genetic algorithm.

From the data mining perspective, prediction of customer's purchase propensity can be classified as a classification problem, and classification is also one of the most common tasks in data mining area. There are many techniques or algorithms available for solving such classification problems. Sometimes, a method might override the others in classification performance on a specific problem, but in general, it is not possible that one method always outperforms all the other methods for every possible situation. This usually depends on the characteristics of the training patterns and unfortunately it is so hard to know in advance exactly which technique or algorithm is best for the problem at hand.

Many researchers have realized that there exists limitations on using a single classification technique. This observation has motivated the relatively recent researches utilizing multiple classifiers for better accuracy [1,3–7,9,10,13,14,16,18,19]. The superiority of these approaches with multiple classifiers and features has already been proved in international recognition competitions [10,13,16].

There are two families of combining multiple classifiers: serial combination and parallel combination. Serial combination arranges classifiers sequentially and the result from the prior classifier is fed to the next classifier [3,7]. Parallel combination arranges classifiers in parallel. If an input is given, multiple classifiers classify it concurrently, and then the classification results from them are integrated by a combining algorithm [1,6,9]. In serial combination, the order of arrangement is crucial for the classification performance of the system and the individual performance of each classifier does not have as much an effect on the system performance. In parallel combination, system performance depends on the combination algorithm. The method described here is a family of parallel combinations. The structures of serial combination and parallel combination are shown in Figs. 1 and 2.

Some commonly used methods for combining classifiers include majority voting, Bayesian, BKS, and Borda Count. Besides these, there are some works that use neural network or fuzzy algorithm [1,6,19].

Since the outputs of individual classifiers are inputs to the combination module, it is therefore important to analyze what kinds of output information classifiers can support. The output information that various classifiers support can be divided into three levels: abstract level, rank level, and measurement level [18]. The abstract level classifiers output only the class label, and the rank level classifiers output the rank for each class. The measurement level classifiers assign each class a measurement value to indicate the possibility that the input pattern pertains to the class. Neural networks are representative examples of measurement level classifiers. The measurement level classifier is able to provide richer information than the abstract and the rank level classifiers. In this paper, we propose a GA-based approach for the combining of measurement level classifiers such as neural network.

The remainder of this paper is organized as follows. Section 2 treats some commonly used methods belonging to parallel combination. GA-based multiple classifier combination is proposed in Section 3. Experimental results on Web data and handwritten

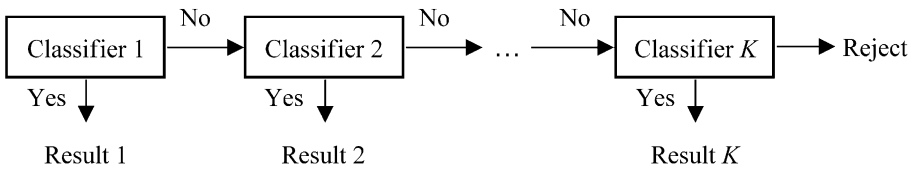

Fig. 1. The block diagram of serial combination.

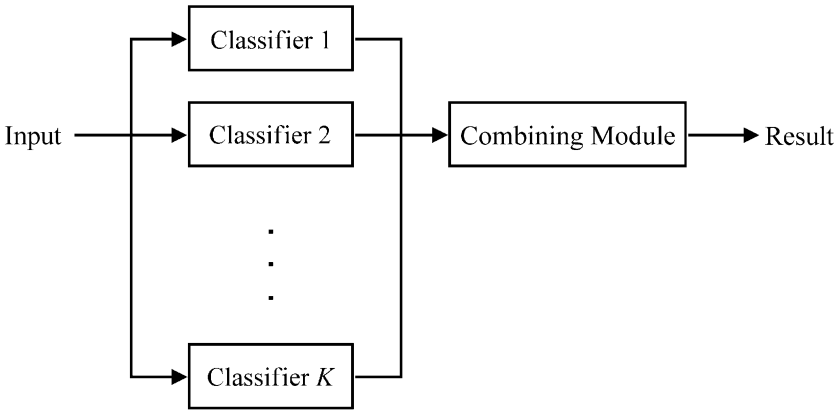

Fig. 2. The block diagram of parallel combination.

numerals are stated in Section 4. Finally in Section 5, conclusions and future research are given.

2. Multiple classifier combination methods

Various classifier combination schemes have been proposed and their usefulness has been experimentally demonstrated. Xu et al. [18] attempted to combine individual classifiers using methods such as Bayesian formalism, voting principle, and Dempster–Shafer theory. Ho et al. [4] suggested methods based on Borda Count and Logistic Regression to reduce and reorder the output set of individual classifiers.

In the remainder of this section, we describe five relatively well-known combining methods: majority voting, Bayesian method, BKS, Borda count, and neural network.

2.1. Majority voting

Voting is the most common method to combine more than one decision. There are various voting strategies such as unanimity, majority, and Borda Count. The majority voting method goes with the decision when there is a consensus for it or at least more than half of the classifiers agree on it. When there is no agreement among more than half of the classifiers, the input is rejected. This method is very simple and needs no extra memory. However, it has a demerit that all classifiers are treated equally regardless of the characteristic of each classifier.

2.2. Bayesian method

Whereas the voting method only considers the result of each classifier, the method using Bayesian formalism considers the error of each classifier. Assuming M classes labeled 1 through M exist, the error for kth classifier where $k=1,\ldots,K$, can be represented by two-dimensional confusion matrix as follows:

$$PT_k = \begin{bmatrix} n_{11} & n_{12} & \cdots & n_{1M} \\ n_{21} & n_{22} & \cdots & n_{2M} \\ \vdots & \vdots & & \vdots \\ n_{M1} & n_{M2} & \cdots & n_{MM} \end{bmatrix}. \quad (1)$$

The rows represent the true identity of input and the column labels represent classification by each classifier.

The probability of classifier selection of class j where $1 \le j \le M$ as its classified class when true class was class i where $1 \le i \le M$ is defined as:

$$P(x \in C_i \mid e_k(x) = j) = \frac{n_{ij}(k)}{\sum_{i=1}^{M} n_{ij}(k)} \quad (2)$$

where $e_k(x)$ is a class label selected by classifier k as the true class for an input x.

The belief function for class i can be expressed by the sum of conditional probabilities as follows:

$$\mathrm{BEL}(i) = \eta \prod_{k=1}^{K} p(x \in C_i \mid e_k(x) = j),$$
$$\text{for } i = 1, \ldots, M \tag{3}$$

where η is a normalization coefficient that satisfies $\sum_{i=1}^{M} \mathrm{BEL}(i) = 1$.

The belief function BEL(i) is the product of the contributions from all classifiers for class i, and represents the total validity for class i. Taking the class label whose BEL value is the largest makes the final decision. The combining rule is shown below,

$$F(x) = \begin{cases} j & \text{if } \mathrm{BEL}(j) = \max_{i \in \Lambda}(\mathrm{BEL}(i)) \text{ and} \\ & \mathrm{BEL}(j) \geq \alpha (0 < \alpha \leq 1) \\ \text{reject} & \text{otherwise} \end{cases} \tag{4}$$

2.3. Behavior–knowledge space method

One of the significant limitations of Bayesian method is that it requires mutual independencies among multiple classifiers, which doesn't usually hold in real application. An approach to overcome this limitation is the Behavior–Knowledge Space (BKS) [5]. A Behavior–Knowledge Space (BKS) means a K-dimensional space, where each dimension corresponds to the decision by one of the classifiers. Table 1 shows an example of two-dimensional BKS, where rows and columns mean the corresponding decision values generated by two different classifiers, 1 and 2, respectively. We also assume those decision values range from 1 to 11 in this table.

Therefore, each cell in the table means the intersection of the decision values from the individual classifiers and becomes a basic unit of computation in BKS approach. In general, each cell in K-dimensional space can be denoted as BKS($e(1),\ldots,e(K)$), where classifier 1 gives its decision as $e(1),\ldots$, and classifier K gives its decision as $e(K)$. To combine the decisions by each classifier, BKS method follows two phases, learning phase and decision phase. During the learning phase, it makes K-dimensional BKS for K classifiers, collecting information to be needed in the following decision phase. In decision phase, it decides the final result using the following rules:

Table 1
Two-dimensional behavior–knowledge space

$e(1)/e(2)$	1	...	j	...	11
1	(1,1)	...	(1,j)	...	(1,11)
:	:	:	:	:	:
i	:	:	(i,j)	:	:
:	:	:	:	:	:
11	(11,1)	...	(11,j)	...	(11,11)

$$F(x) = \begin{cases} R_{e(1)\ldots e(K)}, & \text{if } T_{e(1)\ldots e(K)} > 0 \text{ and} \\ & \frac{n_{e(1)\ldots e(K)}(R_{e(1)\ldots e(K)})}{T_{e(1)\ldots e(K)}} \geq \lambda \\ \text{reject}, & \text{otherwise} \end{cases} \tag{5}$$

where λ is a threshold ($0 \leq \lambda \leq 1$), which controls the reliability of the final decision, $R_{e(1)\ldots e(K)}$ is the best representative result class in BKS($e(1),\ldots,e(K)$), $T_{e(1)\ldots e(K)}$ is the total number of incoming samples in BKS($e(1),\ldots,e(K)$), and $n_{e(1)\ldots e(K)}(m)$ is the total number of incoming samples belonging to class m in BKS($e(1),\ldots,e(K)$).

2.4. Borda count

Borda count, which is a generalization of the majority vote is a useful group consensus function. The Borda count for a class is the sum of the number of classes ranked below it by each classifier. Assuming $B_j(i)$ is the number of classes ranked below the class i by jth classifier and K classifiers are existent, the Borda count for class i is defined as follows:

$$B(i) = \sum_{j=1}^{K} B_j(i). \tag{6}$$

The consensus ranking is given by arranging the classes so that their Borda counts are in descending order and the class label which has the largest Borda count is selected. The Borda count method is simple to implement but it does not consider the differences of individual classifiers in capability.

Weighted Borda count method is an approach assigning weights to the rank scores produced by

individual classifiers to consider each individual's capability [4].

2.5. Combination by neural network

This approach uses the learning ability of neural networks for combining multiple classifiers. In Ref. [6], Huang et al. proposed a combining method using both data transformation and a multi-layer perceptron with the generalized delta rule. The output values of each classifier are first transformed into a form of likelihood measurement. The transformed measurement values are fed to the input layer of the neural network and neural network produces a final classification decision.

3. A GA-based multiple classifier combination

Here, we propose a GA-based multiple classifier combination method that integrates the measurement level classification results generated by multiple classifiers into a single result.

Consider a pattern classification problem where pattern x is assigned to one of the N possible classes $C_1, C_2,\ldots,C_N$. Let $\Lambda=\{1, 2,\ldots,N\}$ be the set of class labels. Let us assume that we have K classifiers each representing the given pattern x by a measurement vector $M_k=\{m_{1k}, m_{2k},\ldots,m_{Nk}\}$, where $k=1,\ldots,K$, and m_{ik} is the measurement value of kth classifier for class i. Let $W_k=\{w_{1k}, w_{2k},\ldots,w_{Nk}\}$ be the weight vector representing the relative significance of kth classifier for all classes. The weight w_{ik} is the degree of importance of kth classifier for class i and implies the estimation of how important kth classifier is in the classification of the class i compared to the other classifiers.

Now, to obtain the final output o_i for class i, measurement values $m_{i1}, m_{i2},\ldots,m_{iK}$ supplied by K classifiers are each weighted by corresponding weight values $w_{i1}, w_{i2},\ldots,w_{iK}$. Then, the output o_i for class i is the summation of the weighted measurement values. This can be shown as:

$$o_i = \sum_{k=1}^{K} w_{ik} m_{ik}. \tag{7}$$

This expression can be written in matrix form.

The final decision is given by selecting the class label whose output value o_i is the highest as follows:

$$E(x) = \begin{cases} j & \text{if } o_j = \max_{i\in\Lambda} o_i \text{ and } o_j \geq \alpha \\ \text{reject} & \text{otherwise} \end{cases} \tag{8}$$

where α is a given threshold.

As shown in Fig. 3, the proposed combination scheme features a single-layer net. It has two layers, input layer and output layer. The measurement values produced from each classifier are fed to the input layer. $o_1, o_2,\ldots,o_N$ are the output values from the output layer. The number of nodes in output layer equals that of the total number of classes. The main difference between the presented method and a single-layer net is that GA is used to optimize the connection weights in the proposed combination scheme. Starting from randomized weight values, the weights gradually reflect the relative importance of each classifier.

3.1. Learning by genetic algorithm

GA is one of the optimization methods using a stochastic search algorithm based on the biological evolution process [12]. There have been many studies using GA for optimization of fuzzy membership

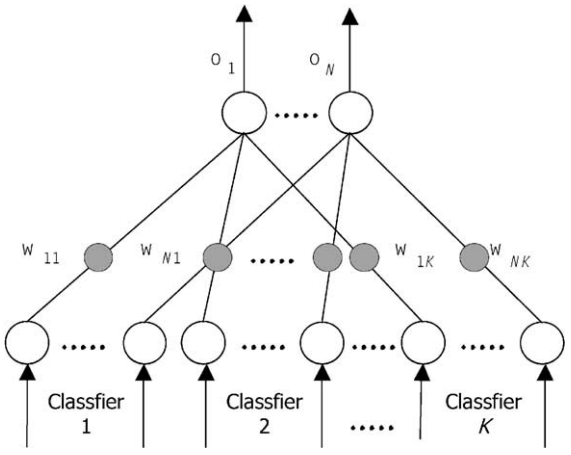

Fig. 3. The structure of the GA-based multiple classifier combination module.

function [17], TSP, and so on. Often, it is applied to the parameter optimization for some system. A key property of GA is that, occasionally, very large changes are introduced and the presence of such large changes and random variations implies that evolutionary methods such as GA can find good solutions even in extremely complex discontinuous spaces or ''fitness landscapes'' that are hard to address by techniques such as gradient descent.

In GA, the problem at hand should be encoded into a string called a *chromosome*, which is usually a binary string. In our research, it is encoded into a string of real values. The collection of candidate solutions known as individuals is called a population. Here, the problem to be solved is optimizing weight matrix in order to combine different measurement values given by K classifiers and determine the final decision. The proposed method maintains a population consisting of candidate weight matrices and uses the genetic algorithm for automatic optimization. Before entering the evolution procedure, our string representation of weight matrix must be defined.

We represent each weight matrix as shown in Fig. 4.

In Fig. 4, the representation of weights is divided into K parts for K classifier system: $W_1,\ldots,W_K$. W_k denotes the weight vector for kth classifier. w_{ik} is the ith weight of W_k, and represent the relative importance of the kth classifier for class i. It is a positive number between 0.0 and 1.0.

GA requires a population of feasible solutions to be initialized and updated during the evolution process. The initial population is generated by setting weights in each individual randomly. Once initial population is generated, the GA operates by iteratively updating the population. On each iteration known as a generation, all individuals of the population are evaluated according to the fitness function that measures their worth. A new population is then generated by probabilistically selecting the fittest individuals from the current population. Some of these selected individuals are carried forward into the next generation population intact. The others are used as a basis for creating new offspring by applying genetic operators, such as selection, crossover, and mutation.

3.2. Fitness function

On each iteration, an evaluation function called fitness function is used to qualify each individual and score it according to its performance on a classification task. Individuals are then ranked according to these scores called fitness values. As a preliminary task to formulate fitness function, hit function (HF) for a candidate weight matrix WS_q is defined as follows:

$$\mathrm{HF}(WS_q) = \begin{cases} 1 & \text{if correctly matched} \\ \left(o_j(WS_q) \Big/ \sum_{i=1}^{N} o_i(WS_q) \right) \xi & \text{otherwise} \end{cases} \tag{9}$$

where $o_i(WS_q) = \sum_{k=1}^{K} w_{ik} m_{ik}$, w_{ik} is the member of weight matrix WS_q, ξ is the constant to control the influence of potential hit on overall learning process, and j is the true class for the input.

If an individual classifies an input correctly, the score for the classifier is increased by 1. Otherwise it is increased by the ratio of the measurement for the true class to the sum of all measurement, shown in Eq. (10), which is the consideration for the potential hit of the individual in the next stage. Since the goal is to

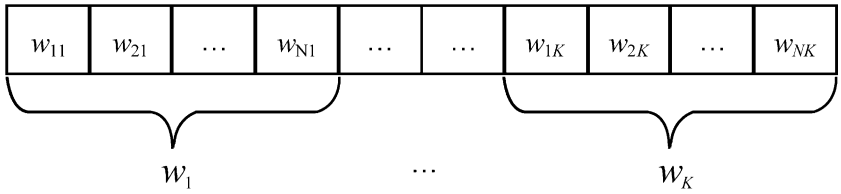

Fig. 4. The string representation of weight matrix.

increase the hit ratio, the fitness function is then defined as follows:

$$\text{Fitness}(WS_q) = \left[\frac{\sum_{i=1}^{S} \text{HF}(WS_q)}{\text{total number of training data}} \right] \quad (10)$$

where S is the total number of training data.

3.3. Selection operator

Individuals from the current population are selected for inclusion in the next generation according to their fitness. We employ the probabilistic model that selects the individuals in the population probabilistically. The probability of selecting candidate solution WS_q is given by

$$P(WS_q) = \text{Fitness}(WS_q) \Big/ \sum_{i=1}^{M} \text{Fitness}(WS_i) \quad (11)$$

where M is the constant denoting the population size.

We also employ the elite preserving strategy to keep weight matrices with high fitness value.

3.4. Crossover and mutation operators

In selection, the chosen individuals are merely reproduced, unchanged. In order to introduce variation into the new offspring, we apply the crossover and mutation operators to the individuals of the current population.

Crossover involves the mixing of two individuals to yield two new ones. We adopt two-point crossover, as shown in Fig 5. Split positions along the weight string are chosen randomly.

The mutation operator selects some elements of an individual at random based on the mutation rate and adds a random value to it. This operation ensures the diversity in the weight matrices over long periods of time and prevents stagnation in the convergence of the optimization.

4. Experimental results

Two data sets are used in the experiment for our combining approach. Firstly, the Web data from one of the leading EC companies in Korea is used to build a model for the customer's purchase behavior prediction. We extract 15 features from the database, which include 10 demographic features (age, gender, education, occupation, marital status, address, hobby, concerning part, customer class, and segmentation number) and 5 transactional features (purchasing pattern, web usage pattern, frequency of purchasing target product, items purchased frequently, items viewed frequently) during one year. It consists of 1602 cases.

Three neural networks with different numbers of hidden units, 10, 20, and 30 (called NN1, NN2, and NN3), are used to build three different classifiers and they are trained to predict the purchase propensity for

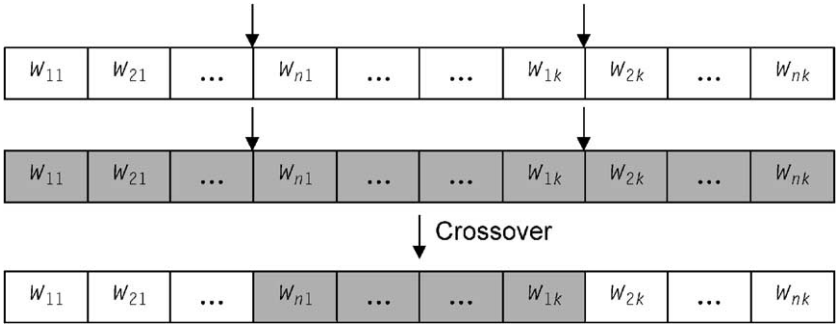

Fig. 5. Two-point crossover operator.

Table 2
The classification rates of individual classifiers and the proposed method

Classifier	Classification rate (%)	Error rate (%)
NN1	73.2	16.8
NN2	73.5	16.5
NN3	74.0	16.0
Proposed method	76.5	13.5

the target products using the back-propagation algorithm [15]. The results from the three classifiers are then integrated to a single unified prediction using the proposed GA-based combination algorithm.

Table 2 presents the 10-fold cross-validation results of each classifier and the GA-based combining method. The table shows that the proposed method has better performance than any individual classifiers.

Secondly, the CENPARMI data set is also used to test the general performance of the proposed method. The data set consists of 6000 handwritten digits, 4000 for training and 2000 for tests, respectively. We use 4000 digits to train the individual classifiers. Among the remaining 2000 digits, 1000 digits are used for combination module training and 1000 digits for combination module test.

Three different neural network classifiers are chosen: K-NN, C-NN, and N-NN. Each classifier is a multi-layer perceptron trained with back-propagation. K-NN uses a Kirsh Mask to detect edge [8]. C-NN uses Chain code representation for the extracted contour of a digit [2]. The N-NN uses a raw digit image without any transformation but normalization by 16×16. Table 3 shows the performances of each classifier and the GA-based combining method.

In the experiments, the population size is set to 150 and the mutation rate is 0.05. The set of individuals is evolved to 100 generations. After 40 generations, there is no significant change in fitness value. Table 4 compares the accuracy of our proposed approach with those of other combining algorithms.

Table 3
The classification rates of individual classifiers and the proposed method

Classifier	Classification rate (%)	Error rate (%)
K-NN	95.85	4.15
C-NN	94.80	5.20
N-NN	92.50	7.50
Proposed method	97.80	2.30

Table 4
Comparison of the proposed method and other combining algorithms

Combining algorithm	Classification rate (%)	Error rate (%)
Majority vote	96.65	2.00
Bayesian	97.55	2.45
BKS + Bayesian	97.50	2.50
Borda count	97.40	2.60
Weighted Borda count	97.40	2.60
Condorect	97.40	2.60
Sum of measurements	97.30	2.70
NN	97.60	2.40
Proposed method	97.80	2.30

5. Discussion and future work

We have proposed a GA-based multiple classifier combining method for the prediction of the EC customer's purchase behavior. Our main idea in the proposed method is based on the fact that different classifiers potentially offer complementary information about the patterns to be classified. The advantage of the proposed approach is derived from its capability to combine individual decisions by multiple classifiers, considering their relative competence in the various contexts.

Our experiment for the case from a leading EC company in Korea shows that the proposed combining method outperforms any individual classifiers. This is also validated for identification of the handwritten digit case.

In the first experiment, we show that the proposed algorithm can improve the prediction accuracy of the purchase propensity. And in the second experiment, we also show that this method has better performance than other combining methods we tried as well as any individual classifiers. Due to the encouraging results obtained from these experiments, the concluding remarks can be summarized as follows.

First, the proposed GA-based combining method can be successfully applied to the prediction task of the customer's purchase propensity in the real world case with more accuracy than the traditional data mining approaches. Second, this method is also an

effective strategy for multiple classifier combination in general classification problem domains.

In our future research, we will apply the method proposed here to additional classification problem domains and we also consider further extension of our approach to integrate different level classifiers such as rank level or abstract level.

References

[1] S.B. Cho, J.H. Kim, Multiple network fusion using fuzzy logic, IEEE Trans. Neural Netw. 6 (2) (1995) 497–501.

[2] H. Freeman, Boundary encoding and processing, in: B.S. Lipkin, A. Rosenfeld (Eds.), Picture Processing and Psychopictorics, Academic Press, 1970, pp. 241–266.

[3] P.D. Gader, D. Hepp, B. Forester, T. Peurach, B.T. Mitchell, Pipelined systems for recognition of handwritten digits in USPS ZIP codes, Proc. U.S. Postal Service Adv. Technol. Conf., (1990) 539–548.

[4] T.K. Ho, J.J. Hull, S.N. Srihari, Decision combination in multiple classifier systems, IEEE Trans. Pattern Anal. Mach. Intell. 16 (1) (1994) 66–75.

[5] Y.S. Huang, C.Y. Suen, The behavior–knowledge space method for combination of multiple classifiers, Proc. IEEE Conf. CVPR, (1993) 347–352.

[6] Y.S. Huang, K. Liu, C.Y. Suen, A neural network approach for multi-classifier recognition systems, Proc. of 4th IWFHR, (1994) 235–244.

[7] F. Kimura, M. Shridhar, Handwritten Numeral Recognition Based on Multiple Algorithms, Pattern Recogn. 24 (10) (1991) 969–983.

[8] R. Kirsch, Computer determination of the constituent structure of biomedical images, Comput. Biomed. Res. 4 (3) (1971) 315–328.

[9] J. Kittler, M. Hatef, R.P.W. Duin, Combining Classifiers, Proc. IEEE Conf. ICPR, (1996) 897–901.

[10] T. Matsui, T. Noumi, I. Yamashita, T. Wakahara, M. Yoshimuro, State of the art of handwritten numeral recognition in Japan—the results of the first IPTP character recognition competition, Proc. of the Second ICDAR, (1993) 391–396.

[11] J. Mena, Data Mining Your Web Site, Digital Press, Butterworth-Heinemann, Linacre House, Jordan Hill, Oxford OX2 8DP, UK, 1999.

[12] T. Mitchell, Machine Learning, The McGraw-Hill, 1221 Avenue of the Americas, New York, NY 10020, USA, 1997.

[13] T. Noumi, et al., Result of second IPTP character recognition competition and studies on multi-expert handwritten numeral recognition, Proc. of 4th IWFHR, (1994) 338–346.

[14] J. Paik, S. Jung, Y. Lee, Multiple combined recognition system for automatic processing of credit card slip applications, Proc. of the Second ICDAR, (1993) 520–523.

[15] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning internal representations by error propagation, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition, vol. 1, The MIT Press, Five Cambridge Center, Cambridge, MA 02142-1493, USA, 1986.

[16] H. Takahashi, T.D. Griffin, Recognition enhancement by linear tournament verification, Proc. of the Second ICDAR, (1993) 585–588.

[17] C.H. Wang, T.P. Hong, S.S. Tseng, Integrating fuzzy knowledge by genetic algorithms, IEEE Trans. Evol. Comput. 2 (4) (1998) 138–149.

[18] L. Xu, A. Krzyzak, C.Y. Suen, Method of combining multiple classifiers and their application to handwritten numeral recognition, IEEE Trans. Syst., Man Cybern. 22 (3) (1992) 418–435.

[19] F. Yamaoka, Y. Lu, A. Shaout, M. Shridhar, Fuzzy integration of classification results in handwritten digit recognition system, Proc. of 4th IWFHR, (1994) 255–264.

Eunju Kim is a PhD candidate in Computer Science at Yonsei University. She also received a BS and a MS in Computer Science at the Yonsei University. Her main research interests are Data Mining, CRM and Machine Learning.

Wooju Kim is an associate professor of Industrial and System Engineering at the Chonbuk National University in Korea. He received a BBA degree from Yonsei University in 1987, and a PhD in Management Science from KAIST in 1994. He has published many papers related to Neural Networks, Expert Systems. S/W Engineering, Knowledge Representation and Acquisition, Managerial Forecasting, and his current research areas are e-business architecture, data mining, web mining, e-CRM, semantic-based meta web search, knowledge management and intelligent systems.

Yillbyung Lee is a professor of Information and Industrial Engineering at the School of Engineering in Yonsei University where he is a director of the Artificial Intelligence Laboratory. He received a BE from Yonsei University, a MS from the University of Illinois and a PhD from University of Massachusetts. His main fields of interests are Document Recognition, Data Mining, Computational Models of Vision, and Biometrics.