Determining the optimal number of cases to combine in a case-based reasoning system for eCRM

Hyunchul Ahn[a]*, Kyoung-jae Kim[b] and Ingoo Han[c]

[a] *Graduate School of Management, Korea Advanced Institute of Science and Technology*
207-43 Cheongrangri-dong, Dongdaemun-gu, Seoul 130-722, Korea
Tel: +82-2-958-3685, Fax: +82-2-958-3604, E-mail:hcahn@kaist.ac.kr

[b] *Department of Information Systems, Dongguk University*
3-26 Pil-dong, Chung-gu, Seoul 100-715, Korea
Tel: +82-2-2260-3324, Fax: +82-2-2260-8824, E-mail:kjkim@dongguk.edu

[c] *Graduate School of Management, Korea Advanced Institute of Science and Technology*
207-43 Cheongrangri-dong, Dongdaemun-gu, Seoul 130-722, Korea
Tel: +82-2-958-3613, Fax: +82-2-958-3604, E-mail:ighan@kgsm.kaist.ac.kr

Abstract

Case-based reasoning (CBR) often shows significant promise for improving effectiveness of complex and unstructured decision making. Consequently, it has been applied to various problem-solving areas including manufacturing, finance and marketing. However, the design of appropriate case indexing and retrieval mechanisms to improve the performance of CBR is still challenging issue. Most of previous studies to improve the effectiveness for CBR have focused on the similarity function or optimization of case features and their weights. However, according to some of prior researches, finding the optimal k parameter for k-nearest neighbor (k-NN) is also crucial to improve the performance of CBR system. Nonetheless, there have been few attempts which have tried to optimize the number of neighbors, especially using artificial intelligence (AI) techniques. In this study, we introduce a genetic algorithm (GA) to optimize the number of neighbors to combine. This study applies the new model to the real-world case provided by an online shopping mall in Korea. Experimental results show that a GA-optimized k-NN approach outperforms other AI techniques for purchasing behavior forecasting.

Keywords:

case-based reasoning, genetic algorithm, the number of neighbors to combine, customer classification

Introduction

Case-based reasoning (CBR) is a problem-solving techniques that is similar to the decision making process of the human beings used in many real world applications. It often shows significant promise for improving the effectiveness of complex and unstructured decision making. Due to its good adaptability for general purposes, it has been applied to various problem-solving areas including manufacturing, finance and marketing (see Yin et al., 2002; Chiu, 2002; Chiu et al. 2003; Shin and Han, 1999; Kim and Han, 2001).

Regardless of its many advantages, there are some problems must be solved to design effective CBR system. The fact that there are no mechanisms to determine appropriate similarity measures, the methods of case indexing and case retrieval in typical CBR systems are some examples of these problems. In this aspect, the selections of the appropriate similarity measures, feature subsets and their weights in the case retrieval step have been most popular research issues (see Wang and Ishii, 1996; Shin and Han, 1999; Kim and Han, 2001; Chiu et al. 2003).

According to some of prior studies, finding the optimal k parameter for k-NN may be crucial to improve the performance of CBR systems (Lee and Park, 1999; Garrell i Guiu, 1999; Jarmulak, 2000). Nonetheless, there have been few attempts which tried to optimize the number of neighbors (i.e. k parameter).

This paper proposes genetic algorithms (GA) to optimize the number of neighbors to combine in CBR system. This study applies the proposed model to the real-world case provided by an online shopping mall in Korea. In addition, this study presents experimental results on application.

The rest of this paper is organized as follows: The next section reviews prior research. Section 3 proposes the GA approach to optimize the number of combining cases and section 4 describes the research design and experiments. In the fifth section, the empirical results are summarized and discussed. In the final section, conclusions and the limitations of this study are presented.

Prior research

In this study, we propose the combined model of two artificial intelligence techniques, CBR and GA. First, in this section, we review the basic concepts of CBR. After that, we introduce prior studies that attempt to combine CBR and GA. Finally, we review the some of studies that tried to optimize the number of combining cases in CBR system.

An overview of CBR

CBR is a problem solving technique that reuses past cases and experiences to find a solution to the problems. While other major artificial intelligence techniques depend on generalized relationships between problem descriptors and conclusions, CBR utilize specific knowledge of previously experienced, concrete problem situations, so it is effective for complex and unstructured problems and easy to update (Shin and Han, 1999).

CBR is considered as a five-step reasoning process shown in Figure 1 (Bradley, 1994).

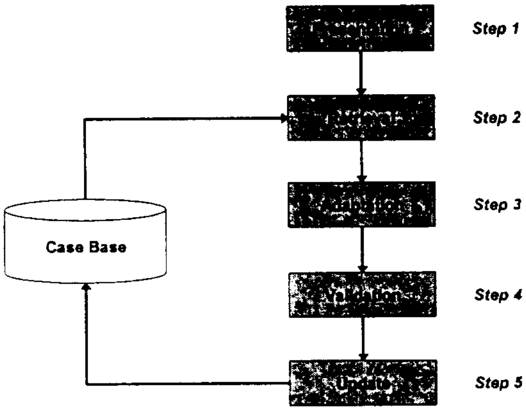


Figure 1 - The general CBR process

1. *Presentation*: a description of the current problem is inputted into the system

2. *Retrieval*: the system retrieves the closest-matching cases in a case base

3. *Adaptation*: the system uses the current problem and closest-matching cases to generate a solution to the current problem.

4. *Validation*: the solution is validated through feedback from the user or the environment.

5. *Update*: if appropriate, the validated solution is added to the case base for future use.

Among these five steps, case retrieval is most critical to determine the effectiveness of CBR system. During the retrieval step, similar cases that are potentially useful to the current problem are retrieved from the case base. So, how to measure similarity of the cases and how to combine the similar cases can be challenging issues in this step (Chiu, 2002).

The similarity can be determined in many ways. However, when cases are represented as feature vectors, calculating the weighted sum of feature distances (e.g. a Hamming distance or Euclidean distance) is common approach. Nearest-neighbor (NN) matching is the most popular method that uses a numerical function to compute the degree of similarity. Equation (1) is a typical numerical function for NN (Jarmulak, 2000; Chiu, 2002).

$$\frac{\sum_{i=1}^{n} W_i \times sim(f_i^I, f_i^R)}{\sum_{i=1}^{n} W_i} \qquad (1)$$

where W_i is the weight of the i th feature, f_i^I is the value of the i th feature for the input case, f_i^R is the value of the i th feature for the retrieved case, and $sim()$ is the similarity function (usually, Euclidean distance) for f_i^I and f_i^R.

Regarding case retrieval, many CBR systems use one-nearest neighbor (1-NN) method. It's the method to retrieve the most similar case from the case-base and make prediction based on it. However, to improve performance, some CBR systems retrieve several most similar cases simultaneously and make prediction by combining these all cases (e.g. voting or interpolation). This is called k nearest neighbor (k-NN) retrieval. The parameter, k, means the number of cases to combine. Values of k larger than 1 may be used to improve the generalization properties of the retrieval and reduce sensitivity to noise. That is, large k parameter can improve accuracy of the prediction results for CBR. However, if k is too large, the prediction accuracy may be lower because the selected similar cases would include many noisy cases. So, finding the optimal k parameter for k-NN is important to improve accuracy of this kind of retrieval systems.

Optimization approaches for CBR using GA

When we use NN or k-NN matching as case retrieval mechanism for CBR, there exist two critical issues regarding designing CBR system. One is how to select the appropriate features, known as feature selection, and the other is how to determine the weight of each feature, which is known as feature weighting. So, there have been many studies that attempt to resolve these problems. Among many methods of feature selection and feature weighting, the GA is increasingly being used in CBR system.

Genetic algorithms are stochastic search techniques that can search large and complicated spaces. It is based on the biological backgrounds including natural genetics and evolutionary principle. Especially, GAs are suitable for parameter optimization problems with an objective function subject to various hard and soft constraints (Shin and Han, 1999). The GA basically explores a complex space in an adaptive way, guided by the biological evolution of selection, crossover, and mutation. This algorithm uses natural selection, survival of the fittest, to solve optimization problems. (Kim, 2004)

Historically, the GA has been used to determine various feature selection and weighting problems in other AI algorithms including ANN, inductive learning, and linear regression. (Kim, 2004). For CBR, Siedlecki and Sklansky (1989) proposed a feature selection algorithm based on genetic search, and Kim and Han (2001) used GA for feature discretization. In the case of GA-optimized feature weighting, there are many examples. Shin and Han (1999) proposed GA-optimization for feature weights and applied it to corporate bond rating. Chiu et al. (2003) applied GA-optimized feature weighting to due-date assignment problem in a water fabrication factory. Chiu (2002) also used the same algorithm, but he applied it to the real case of CRM (Customer Relationship Management).

Optimization for the number of combining cases in k-NN algorithm

As we can see above, there have been many approaches to optimize features and their weights for CBR system so far. However, as mentioned at the bottom of section 2.1, determining optimal k parameter (i.e. the number of combining cases) is also important to improve the performance of k-NN CBR system. Nonetheless, there are few studies that tried to optimize it.

Lee and Park (1999) proposed three methods to optimize the number of cases to combine. These are (1) fixed number combining methods (conventional k-NN), (2) optimal spanning methods, and (3) mathematical programming (MP) model using similarity distribution. A simulation study was conducted to test the performance of each model and it proved that MP model using similarity distribution was the best among the suggested. Equation (2) and (3) are the objective function and constants in their MP model:

$$Max.\ \mathrm{SF} = \frac{\sum_{b=1}^{n} S_{tb} Z_b}{(\sum_{b=1}^{n}\sum_{q=1}^{n} S_{bq} Z_b Z_q)^p} \qquad (2)$$

$$\begin{aligned} s.t.\quad & (S_{tb} - S_{tq}) \times (Z_b - Z_q) \geq 0 \quad \forall\ b \text{ and } q \\ & Z_b = 0 \quad \text{or} \quad 1 \\ & 0 \leq p \leq 0.5 \end{aligned} \qquad (3)$$

where n is the number of cases to combine, s_{tb} is the similarity between target case (input case) t and base case (retrieved case) b, and s_{bq} is the similarity between base case b and another base case q. And, z_b is the binary sign variable which represents whether the base case b would be selected or not.

Their MP model is worthwhile because it is the first attempt to optimize k parameter and it is based on concrete science such as linear programming (LP) and statistics. However, their suggestion has several critical limitations. First of all, as we can infer from above equations, the optimal number of cases to combine wholly depends on each input case t. That is, the model computes different optimal k every time when it gets new input case. So, this model may not suggest the optimal number of k parameter that can be applied generally and it also causes too much computation time that may disable real-time prediction. Furthermore, this model still has a variable to optimize, parameter p. The authors explain parameter p as an adjusting factor to determine the number of cases to combine, but there is not precise definition for parameter p. It also suggests no mechanism to determine the appropriate value of parameter p.

GA-optimization for k parameter of k-NN

To mitigate the limitations of prior studies, this paper proposes GA as an optimization algorithm for k parameter of k-NN. This study names this model as GA k-NN (GA-optimized k-Nearest Neighbor algorithm). The framework of GA k-NN is shown in Figure 2.

The process of GA k-NN consists of the following three stages:

Stage 1. For the first stage, we search the search space to find an optimal or near-optimal k parameter. The population (seed points for finding optimal k) is initiated into random values before the search process. The parameter for searching must be encoded on a chromosome. The encoded chromosome is searched to maximize the specific fitness function. The objective of this paper is to determine appropriate k parameter of k-NN and it can be represented by the average prediction accuracy of the test data. Thus, this study applies it to the fitness function for GA. The fitness function can be expressed as Equation (4):

$$\begin{aligned} & \text{Fitness} = \frac{1}{n}\sum_{i=1}^{n} CR_i \quad (i = 1, 2, \ldots n) \\ & \textit{if } PO_i = AO_i\ ,\quad CR_i = 1 \\ & \textit{otherwise}\ ,\quad CR_i = 0 \end{aligned} \qquad (4)$$

where CR_i is the prediction result for the i th test case which is denoted by 0 or 1, PO_i is the predicted output from the model for the i th test case, and AO_i is the actual output from the model for the i th test case.

In this stage, the GA operates the process of crossover and mutation on initial chromosome and iterates it until the stopping conditions are satisfied.

Stage 2. The second stage is the process of case retrieval and matching for new problem in the CBR system. In this stage, k-NN matching is used as a method of case retrieval. In our study, we use weighted average of Euclidean distance for the each feature as a similarity measure. And, all the feature weights, w_i, are set to '1' as it is most common method for feature weighting (Chiu et al., 2002).

Stage 3. The third stage applies selected k, the optimal number of cases to combine, to the hold-out data. This stage is required because the GA optimizes the parameters to maximize the average predictive accuracy of the test data, but sometimes the optimized parameters are not generalized to deal with the unknown data.

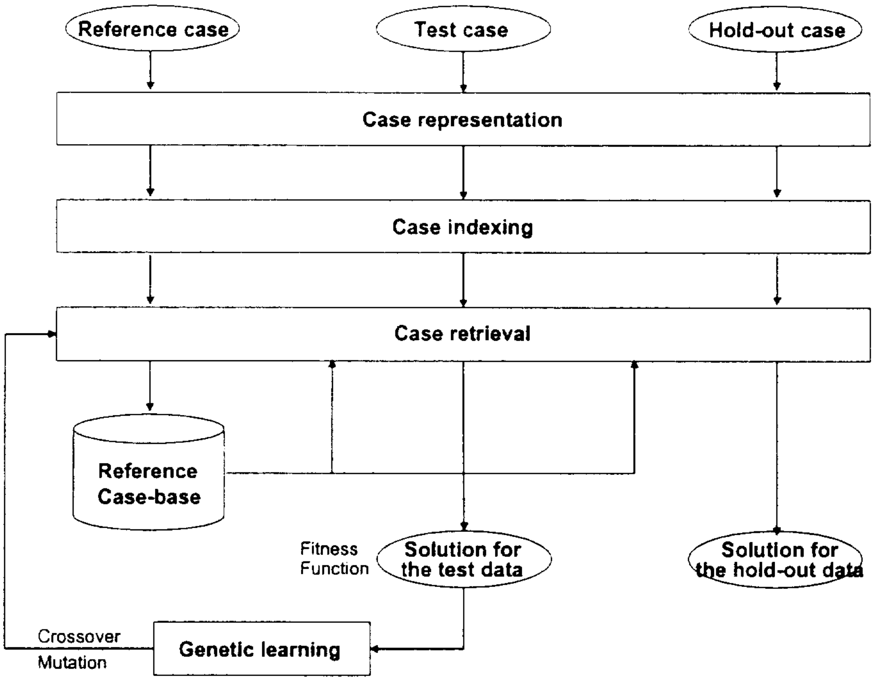


Figure 2 - Framework of GA k-NN

The research design and experiments

Research data

It is an important issue in real world marketing to find the potential buyers for the specific product, i.e. customer classification. It is believed that companies' knowledge about their customer behavior patterns would create many chances that more effective marketing strategies can be developed (Chiu, 2002). This research applies GA k-NN method to classify potential customers into either purchasing or non-purchasing categories.

Data is collected from an online diet shopping mall in Korea. The mall deals with 4 kinds of products, (1) total diet counseling service, (2) functional meals or recipes which are helpful for diet, (3) fitness equipments, and (4) various accessories. Among them, fitness equipments are the most beneficial product group for the mall because they are usually quite expensive and provide high margin to the company. So, the mall has interest in examining the potential buyers for the fitness equipments and we developed a model to classify the customers who were expected to purchase fitness equipments.

The experiment data includes 3,156 cases that consisted of the purchasing and non-purchasing customers from May 2001 to August 2001. The data are split into the three groups: reference, test and hold-out case-bases. The reference case-base is used to search optimal k parameter in genetic learning and also used as a case-base for retrieval. The test case-base is used to measure how well the candidate for k improves the accuracy of the CBR system. The final one, the hold-out case-base, is used to validate the generalizability of the model for the unseen data. The number of cases in each case-base is shown in Table 1.

Table 1 - The number of each case-base

Case-base	The number of cases	Proportions
Reference	1893	60%
Test	631	20%
Hold-out	631	20%
Total	3155	100%

In order to develop a model that is able to effectively differentiate purchasing customers from non-purchasing customers, 46 possible factors including demographic and other personal information are collected. After that, we adapt two statistical methods, the two-sample t-tests for ratio variables and chi-square variables for nominal variables in order to select relevant factors, and we select 17 candidate factors. Among them, finally, we select only 7 factors which are proved to be the most influential to the purchase of fitness equipments by using the forward selection procedure based on logistic regression. The probability for stepwise entry is set to 0.05 and that for stepwise removal is set to 0.10. Table 2 contains detail information about the selected factors.

Table 2 - Selected features and their descriptions

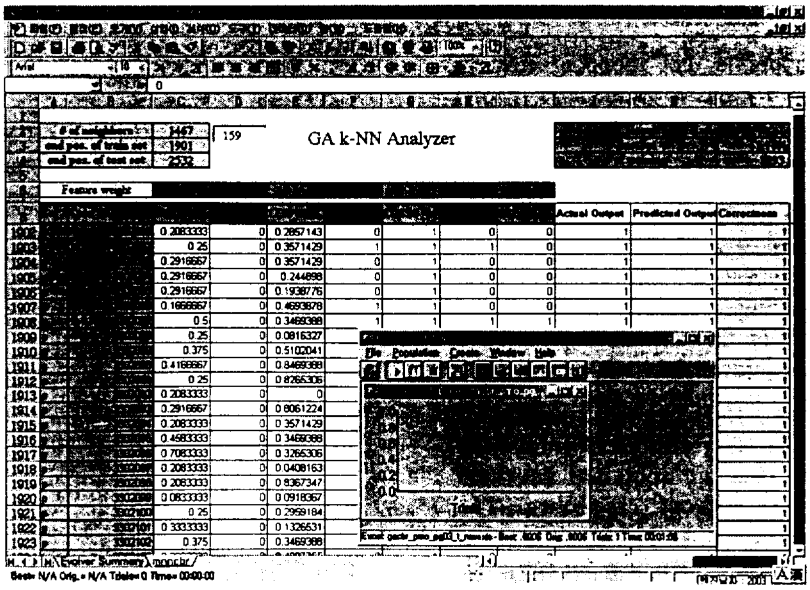
Feature name	Description	Range
MARRIED	Customer's marriage status	0 : Not married 1: Married
LOSS1	Customer's need to lose weight around their belly	0 : Not exist 1 : Exist
LOSS6	Customer's need to lose weight around their legs and thighs	0 : Not exist 1 : Exist
E01	Customer's prior experience of 'functional diet food'	0 : Not exist 1 : Exist
E02	Customer's prior experience of 'diet drug'	0 : Not exist 1 : Exist
BMI	Body mass index (BMI) is measure of body fat based on height and weight that applies to both adult men and women. It is calculated as follows: $BMI(kg/m^2)=\frac{weight(kg)}{(height(m))^2}$	Continuous (kg/m^2)
LENPUR	The length of the time from customer's last purchase to present.	Continuous $(days)$

Research design and system development

For the controlling parameters of GA search, the population size was set to 100 organisms and the crossover and mutation rate were set to 0.7 and 0.1. And, as the stopping condition, only 700 trials are permitted. The parameter k to be searched used only the information about the reference and the test case-base.

To compare the result of k-NN, we also applied other algorithms to the same data set. The comparing algorithms include 1-NN CBR, Conventional k-NN (Conv. K-NN), logistic regression (LOGIT), and artificial neural networks (ANN). 1-NN CBR is the nearest neighbor algorithm which selects just one closest neighbor. Conv. k-NN is k-NN algorithm but selects k as a fixed number that usually ranges from 1 to 10. In this experiment, we select k for the Conv. k-NN which shows the best performance in the range between 1 and 10. The experiment for LOGIT is performed by using the SPSS 11.0 for windows. ANN is designed as three-layer network whose learning rate and momentum rate are 0.1. We experiment ANN models by varying its number of hidden nodes from 4 to 14 and, among them, we have chosen the model whose performance is the best. To experiment ANN models, we apply Ward System Group's Neuroshell 4.0.

The GA k-NN system is developed by using Microsoft Excel 2002 and Palisade Software's Evolver version 4.06. k-NN algorithm was implemented in VBA (Visual Basic for Applications) of Microsoft's Excel 2002. In addition, GA-optimization for k parameter was done by Evolver. Figure 3 represents the working screen of the developed GA k-NN system.



GA k-NN Analyzer

# of neighbors	1467
end pos. of train set	1901
end pos. of test set	2532

159

Feature weight

									Actual Output	Predicted Output	Correctness
1902	[illegible]	0.2083333	0	0.2857143	0	1	0	0	1	1	1
1903	[illegible]	0.25	0	0.3571429	1	1	1	0	1	1	1
1904	[illegible]	0.2916667	0	0.3571429	0	1	0	0	1	1	1
1905	[illegible]	0.2916667	0	0.244898	0	1	0	0	1	1	1
1906	[illegible]	0.2916667	0	0.1938776	0	1	0	0	1	1	1
1907	[illegible]	0.1666667	0	0.4693878	1	1	0	0	1	1	1
1908	[illegible]	0.5	0	0.3469388	1	1	1	1	1	1	1
1909	[illegible]	0.25	0	0.0816327							1
1910	[illegible]	0.375	0	0.5102041							1
1911	[illegible]	0.4166667	0	0.8469388							1
1912	[illegible]	0.25	0	0.8265306							1
1913	[illegible]	0.2083333	0	0							1
1914	[illegible]	0.2916667	0	0.8061224							1
1915	[illegible]	0.2083333	0	0.3571429							1
1916	[illegible]	0.4583333	0	0.3469388							1
1917	[illegible]	0.7083333	0	0.3265306							1
1918	[illegible]	0.2083333	0	0.0408163							1
1919	[illegible]	0.2083333	0	0.8367347							1
1920	[illegible]	0.0833333	0	0.0918367							1
1921	[illegible]	0.25	0	0.2959184							1
1922	[illegible]	0.3333333	0	0.1326531							1
1923	[illegible]	0.375	0	0.3469388							1

Best= N/A Orig.= N/A Trials= 0 Time= 00:00:00

Figure 3 - Sample screen of the GA k-NN system

Experimental results

In this section, the prediction performances of GA k-NN and other alternative models are compared. As an experimental result, we get optimal k for the Conv. k-NN and GA k-NN as 6 and 159. Table 3 shows all the results of the Conv. k-NN ranging k from 1 to 10 and Table 4 describes the average prediction accuracy of each model.

In Table 3, GA k-NN achieves higher prediction accuracy than 1-NN, Conv. k-NN, LOGIT, and ANN by 7.92%, 6.02%, 4.75%, and 3.96% for the hold-out data.

The McNemar tests are used to examine whether the predictive performance of the GA k-NN is significantly higher than that of other algorithms. This test is used with nominal data and is particularly useful with before-after measurement of the same subjects (Kim, 2004). Table 5 shows the results of the McNemar test to compare the performances of five algorithms for the hold-out data.

As shown in Table 5, GA k-NN is better than 1-NN and Conv. k-NN at the 1% and better than LOGIT at the 5% statistical significance level. But, the performance of GA k-NN does not outperform ANN significantly.

In addition, the two-sample test for proportions is performed. This test is designed to distinguish between two proportions when the prediction accuracy of the left-vertical methods is compared with those of the right horizontal methods (Harnett and Soni, 1991). Table 6 shows *p* values for the pairwise comparison of performance between models. As shown in Table 6, GA k-NN outperforms LOGIT and 1-NN with the 1% statistical significance level and also outperforms Conv. K-NN with the 5% significance level. In addition, GA k-NN is better than ANN at the 10% significance level. Table 6 also shows ANN outperforms 1-NN with 10% statistical significance level.

Table 3 - Prediction accuracy of Conv. k-NN models ranging k from 1 to 10

k	1	2	3	4	5	6	7	8	9	10
Performance for hold-out case-base	56.42%	53.25%	57.05%	57.84%	57.84%	58.32%	57.05%	56.74%	57.53%	58.16%

Table 4 - Average prediction accuracy of the models

Case-base	LOGIT	ANN	1-NN	Conv. k-NN (k=6)	GA k-NN (k=159)
Training	58.94%	58.21%	-	-	-
Test		58.41%	-	-	62.44%
Hold-out	59.59%	60.38%	56.42%	58.32%	64.34%

Table 5 - McNemar values for the hold-out data

	ANN	1-NN	Conv. k-NN	GA k-NN
LOGIT	0.8797	0.1667	0.0302	5.3535*
ANN		1.9794	0.5353	2.1413
1-NN			0.6722	9.0263**
Conv. k-NN				6.8450**

* significant at the 5% level, ** significant at the 1% level

Table 6. p values for the hold-out data

	ANN	1-NN	Conv. k-NN	GA k-NN
LOGIT	0.1653	0.3346	0.4098	0.0076
ANN		0.0767	0.2282	0.0731
1-NN			0.2473	0.0020
Conv. k-NN				0.0139

Conclusions

This paper has suggested a new kind of hybrid system of GA and CBR to improve performance. In this paper, we use GA as a tool to optimize the number of cases to combine (k parameter) in k-NN. From the results of the experiment, we show that our GA k-NN outperforms other comparative algorithms such as LOGIT and ANN as well as typical CBR algorithms (1-NN and conventional k-NN)

However, this study has some limitations. First of all, our GA k-NN doesn't consider any efforts to optimize feature selection and feature weighting. In fact, many prior studies have pointed out that optimizing feature selection and weighting can improve the performance of the CBR system (Siedlecki and Sklansky, 1989; Shin and Han, 1999; Chiu, 2002; Chiu et al., 2003; Kim, 2004). So, the prediction performance of GA k-NN may be enhanced if the GA is employed for simultaneous optimization of k parameter and feature weights. In addition, other GA-optimization methods for CBR can be also considered. For example, GA can be applied to relevant instance selection. All of these remain interesting topics for the future research. And, of course, the generalizability of GA k-NN should be tested further by applying it to other problem domains.

References

[1] Bradley, P. (1994). "Case-based reasoning: Business applications," *Communication of the ACM, 37* (3), 40-43.

[2] Chiu, C. (2002). "A case-based customer classification approach for direct marketing," *Expert Systems with Applications, 22*, 163-168.

[3] Chiu, C., P. C. Chang and N. H Chiu. (2003). "A case-based expert support system for due-date assignment in a water fabrication factory," *Journal of Intelligent Manufacturing, 14*, 287-296.

[4] Fu, Y. and R. Shen. (2003). "GA based CBR approach in Q&A system," *Expert Systems with Applications*, Forthcoming.

[5] Garrell i Guiu, J. M., E. Golobardes i Ribé, E. Bernadó i Mansilla, X. Llorà i Fàbrega. (1999). "Automatic diagnosis with genetic algorithms and case-based reasoning," *Artificial Intelligence in Engineering, 13*, 367-372.

[6] Harnett, D. L. and A. K. Soni. (1991). *Statistical methods for business and economics*. Addison-Wesley: Massachusetts.

[7] Jarmulak, J., S. Craw and R. Rowe. (2000). "Self-optimizing CBR Retrieval," *Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence*. 376-383.

[8] Jeng, B. C. and T. P. Liang. (1994). "Fuzzy indexing and retrieval in case-based systems," *Expert Systems with Applications, 8*, 135-142.

[9] Kim, K. and I. Han. (2001). "Maintaining case-based reasoning systems using a genetic algorithms approach," *Expert Systems with Applications, 21*, 139-145.

[10] Kim, K. (2004). "Toward global optimization of case-based reasoning systems for financial forecasting," *Applied Intelligence*, Forthcoming.

[11] Lee, H. Y. and K. N. Park. (1999). "Methods for Determining the optimal number of cases to combine in an effective case based forecasting system," *Korean Journal of Management Research, 27*, 1239-1252.

[12] Shin, K. S. and I. Han. (1999). "Case-based reasoning supported by genetic algorithms for corporate bond rating," *Expert Systems with Applications, 16*, 85-95.

[13] Siedlecki, W. and J. Sklanski. (1989). "A note on genetic algorithms for large-scale feature selection," *Pattern Recognition Letters, 10*, 335-347.

[14] Wang, Y. and N. Ishii. (1996). "A method of similarity metrics for structured representations," *Expert Systems with Applications, 12*, 89-100.

[15] Yin, W. J., M. Liu and C. Wu. (2002). "A genetic learning approach with case-based memory for job-shop scheduling problems," *Proceedings of the First International Conference on Machine Learning and Cybernetics*, 1683-1687.