SIMULTANEOUS OPTIMIZATION OF FEATURE WEIGHTING AND INSTANCE SELECTION IN CASE-BASED REASONING SYSTEMS USING GENETIC ALGORITHMS

Hyunchul Ahn
Graduate School of Management, Korea Advanced Institute of Science and Technology, hcahn@kaist.ac.kr

Kyoung-jae Kim
Department of Information Management, Dongguk University, kjkim@dongguk.edu

Ingoo Han
Graduate School of Management, Korea Advanced Institute of Science and Technology, 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 retrieval mechanisms to improve the performance of CBR is still a challenging issue. Most of previous studies to improve the effectiveness for CBR have focused on the selection of appropriate instances and the optimization of case features and their weights. However, these approaches have been applied independently. Here we encode the feature weighting and instance selection within the same genetic algorithm (GA) and suggest simultaneous optimization model of feature weighting and instance selection. This study applies the new model to bankruptcy prediction. Experimental results show that simultaneously optimized CBR outperforms other CBR techniques.

KEYWORDS: case-based reasoning, genetic algorithm, feature weighting, instance selection

INTRODUCTION

Case-based reasoning (CBR) is a problem-solving technique that is similar to the

decision making process that human beings use in many real world applications. It often shows significant promise for improving the effectiveness of complex and unstructured decision making. Due to its high adaptability for general purposes, it has been applied to various problem-solving areas including manufacturing, finance and marketing (see Chiu, 2002; Kim & Han, 2001; Shin & Han, 1999; Yin, Liu, & Wu, 2002).

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

Recently, simultaneous optimization of several variables in CBR attracts researchers' interest due to its better performance. As a pioneering work, there exists the approach to combine feature selection and instance selection simultaneously (Kuncheva and Jain, 1999; Rozsypal and Kubat, 2003). Theoretically, feature weighting includes feature selection. For feature weighting determines weights of features from 0 to 1, but feature selection is just binary selection, 0 or 1. Consequently, feature weighting may improve the effectiveness of CBR systems better than feature selection. Nonetheless, there have been few attempts which tried to optimize feature weighting and instance selection simultaneously.

This paper proposes genetic algorithms (GA) to optimize the feature weights and instance selection simultaneously. This study applies the proposed model to the real-world case of bankruptcy prediction. In addition, this study presents experimental results from the application.

PRIOR RESEARCH

In this study, we propose the combined model of two artificial intelligence techniques, CBR and GA. So, in this section, we first review the basic concepts of CBR and GA. After that, we introduce prior studies that attempt to optimize CBR. Finally, we review the some of studies that tried to optimize several variables of CBR system simultaneously.

Genetic algorithms as an optimization tool for 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 utilizes specific knowledge of previously experienced, concrete problem situations, so it is effective for complex and unstructured problems and easy to update (Shin & Han, 1999). CBR is considered to be a five-step process shown in Fig. 1 (Bradley, 1994).

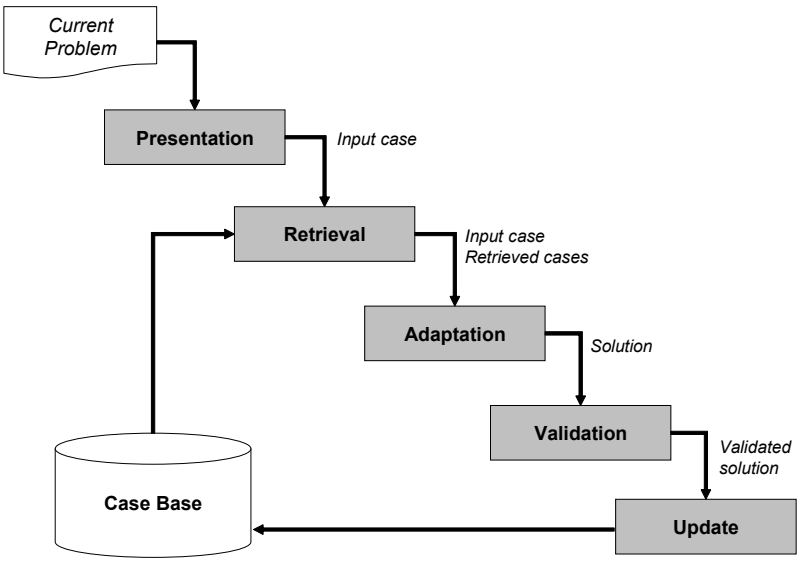

Fig. 1. The general CBR process

Among these five steps, step two, case retrieval, is most critical for determining the effectiveness of CBR system. During the retrieval step, similar cases that are potentially useful for solving the current problem are retrieved from the case base. So, how to measure the similarity of the cases and how to combine the similar cases are challenging issues in this step (Chiu, 2002). Especially, feature weighting (selection) and instance selection for measuring similarity have been controversial issues in designing CBR system. To determine these uncertain factors of CBR system, there have been many studies that attempt to resolve these problems. Among many methods of instance selection and feature weighting, GA is increasingly being used in the CBR system.

Genetic algorithms are stochastic search techniques that can search large and complicated spaces. It is based on biological backgrounds including natural genetics

and evolutionary principle. In particular, GAs are suitable for parameter optimization problems with an objective function subject to various hard and soft constraints (Shin & 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).

Feature selection and weighting approaches

Feature selection is the process to pick a subset of features that are relevant to the target concept and remove irrelevant or redundant features. And, feature weighting is assigning a weight to each feature according to the relative importance of each one. These are important factors that determine the performance of AI systems, so they have been the most popular research issues in designing most AI systems including CBR systems.

In the case of feature selection, Siedlecki and Sklanski (1989) proposed a feature selection algorithm based on genetic search and Cardie (1993) presented a decision tree approach to feature subset selection. Skalak (1994) and Domingos (1997) also proposed a hill climbing algorithm and a clustering method for feature subset selection. In addition, Cardie and Howe (1997) used a mixed feature weighting and feature subset selection method. They first selected relevant features using decision tree, then they assigned weights to the remained features using the value of information gain for each feature. Jarmulak et al. (2000) selected relevant features using decision tree algorithm including C4.5 and assigned feature weights using the GA.

Regarding feature weighting, Wettschereck et al. (1997) presented various feature weighting methods based on distance metrics in the machine learning literature. Kelly and Davis (1991) proposed a GA-based feature weighting method for k-nearest neighbor. Similar methods are applied to the prediction of corporate bond rating (Shin & Han, 1999) and to failure-mechanism identification (Liao et al., 2000). In addition, Kim and Shin (2000) presented feature weighting methods based on the GA and ANN.

Instance selection approaches

Instance selection is the technique that selects an appropriate reduced subset of case-base and applying the nearest-neighbor rule to the selected subset. It may increase the

performance of CBR systems dramatically if the systems contain many noises. So, it has been another popular research issues in CBR systems for long time.

There exist many different approaches to select appropriate instances. First of all, Hart (1968) suggested condensed nearest neighbor algorithm (Hart, 1968) and Wilson (1972) proposed Wilson's method (Wilson, 1972). Their algorithms that are based on simple information gain, so they are easy to apply. Recent studies for instance selection use mathematical tools or AI techniques for improving accuracy. For example, Sanchez, Pla, and Ferri (1997) propose proximity graph approach and Lipowezky (1998) suggests linear programming methods for instance selection. Yan (1993) or Huang, Chiang, Shieh and Grimson (2002) suggest ANN-based instance selection method and GA approach is also proposed by Babu and Murty (2001).

Simultaneous optimization approaches

The first simultaneous optimization approach is proposed in 1999, so there are few studies because of its short history. Kuncheva and Jain (1999) proposed simultaneous optimization of feature selection and instance selection using GA and they compared their model to sequential combining of traditional feature selection and instance selection algorithms. Rozsypal and Kubat (2003) also tried simultaneous optimization of feature and instance selection using GA, but they differentiated their model by different design of GA settings. They showed that their model outperforms one of Kuncheva and Jain.

As mentioned, feature weighting includes feature selection, since selection is a special case of weighting with binary weights. Consequently, simultaneous optimization model of feature weighting and instance selection model may improve the performance of the one of feature selection and instance selection. In this manner, Yu et al. (2003) proposed simultaneous optimization model of feature weighting and instance selection for collaborative filtering. Collaborative filtering is the algorithm that is very similar to CBR, but, it is different from CBR in essence. Furthermore, they applied not AI techniques, but an information-theoretic approach to the optimization model. So, in the strict sense of the word, their model is not simultaneous optimization model but sequential combining model of two approaches.

GENETIC ALGORITHMS FOR SIMULTANEOUS FEATURE WEIGHTING AND INSTANCE SELECTION

To mitigate the limitations of prior studies, this paper proposes GA as a simultaneous optimization tool of feature weighting and instance selection. To test the effectiveness of the proposed model, we compare the results of four different models.

The first model, labeled COCBR (COnventional CBR), uses a conventional approach for reasoning process of CBR. This model considers all initially available features as a feature subset. Thus, there is no special process of feature subset selection. In addition, relative importance of each feature is not considered because many conventional CBR models do not have general feature selection or weighting algorithm. For the feature transformation method, linear scaling is used. Linear scaling means linear scaling to unit variance in this study. It transforms a feature component x to a random variable with zero mean and unit variance (Jain & Dubes, 1988). It is usually employed to enhance the performance of the CBR system because it ensures the larger value input features do not overwhelm smaller value input features.

The second model assigns relevant feature weights via genetic search. This study names this model FWCBR (Feature Weighting using the GA for CBR). Similar models to it were previously suggested by Kelly and Davis (1991), Shin and Han (1999), Kim and Shin (2000), and Liao et al. (2000).

The third model uses the GA to select a relevant feature subset. This study names this model ISCBR (Instance Selection using the GA for CBR). This model also uses linear scaling for feature transformation. Babu and Murty (2001) proposed similar model to it.

The fourth model, the proposed model in this study, employs the GA to select a relevant instance subset and to optimize the weights of each feature simultaneously using the reference and the test case-base. This model is named as SOCBR (Simultaneous Optimization using the GA for CBR) in this study. The process of SOCBR consists of the following three stages:

Stage 1. For the first stage, we search the search space to find optimal or near-optimal parameters (feature weights and selection variables for each instance). The population (seed points for finding optimal parameters) is initiated into random values before the

search process. The parameter to be found 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 the feature weights and instance selection of CBR systems 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 (1):

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

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

In this study, the GA operates the process of crossover and mutation on the 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 a new problem in the CBR system using the parameters that are set in Stage 1. In this stage, 1-NN(one-nearest neighbor) matching is used as a method of case retrieval. And, we use the weighted average of Euclidean distance for the each feature as a similarity measure. This stage is repeated after the process of GA's evolution (crossover and mutation) and the value of the fitness function is updated.

Stage 3. The third stage applies the finally selected parameters - the optimal weights of features and selection of instances - to the hold-out data. This stage is required because 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.

THE RESEARCH DESIGN AND EXPERIMENTS

Application data

The application data used in this study consists of financial ratios and the status of

bankrupt or non-bankrupt for corresponding corporate. The data was collected from one of largest commercial banks in Korea. The sample of bankrupt companies was 1335 companies in heavy industry which filed for bankruptcy between 1996 and 2000. The non-bankrupt companies were 1335 ones in heavy industry which filed between 1999 and 2000. Thus, the total number of samples is 2670 companies.

The financial status for each company is categorized as "0" or "1" and it is used as a dependent variable. "0" means that the corporate is bankrupt, and "1" means that the corporate is solvent. For independent variables, we first generate 164 financial ratios from the financial statement from each company. Finally, we get 15 financial ratios as independent variables through the two independent sample t-test and the forward selection procedure based on logistic regression. Table 1 gives selected features and some statistics from outputs of descriptive statistics and logistic regression analysis.

Table 1. Selected features and their statistics

Name of feature	Mean	Standard deviation	Sig.
Financial expenses to liabilities	7.082	3.551	0.000
Cost of sales to net sales	81.879	8.167	0.027
Net worth to total assets	24.183	16.821	0.000
Financial expenses growth	-0.007	0.035	0.038
Payables turnover	13.373	21.388	0.000
Solvency ratios	38.553	36.620	0.043
Window coefficient	0.964	1.547	0.017
Financial expenses & normal profit to total assets	17.054	23.104	0.000
Cash flow to total liabilities	0.094	0.323	0.000
Total asset change ratios	20.390	20.739	0.000
Financial expenses growth rate to assets	0.262	2.845	0.000
Non-operating expenses growth rate to assets	-0.080	3.903	0.000
Cost of sales X Cost of sales growth ratio	142.358	152.350	0.000
Inventories growth rate to sales	1.597	7.146	0.002
Total assets turnover X Sales growth rate	1.991	1.837	0.000

Research design and system development

For the controlling parameters of GA search for SOCBR, the population size was set at

100 organisms and the crossover and mutation rate were set at 0.7 and 0.1. And, as the stopping condition, only 1500 trials (15 generations) are permitted.

To compare the result of SOCBR, we also applied other algorithms to the same data set. The compared algorithms include COCBR(Conventional CBR), FWCBR(Feature weighted CBR by GA), and ISCBR(Instance selected CBR by GA). COCBR is 1-NN algorithm whose feature weights are set to 1. FWCBR is 1-NN algorithm whose feature weights are optimized by GA. In the case of COCBR and FWCBR, all the instances are used for reference case-base. However, ISCBR uses only subset of total reference case-base which is selected by GA. The controlling parameters of GA search for FWCBR and ISCBR, the population size was set at 50 organisms and the crossover and mutation rate were set at 0.7 and 0.1. And, as the stopping condition, about 500 trials (10 generations) are permitted.

All CBR systems are developed by using Microsoft Excel 2002 and Palisade Software's Evolver Version 4.06. The 1-NN algorithm was implemented in VBA (Visual Basic for Applications) of Microsoft Excel 2002. In addition, GA-optimization for the parameters was done by Evolver.

EXPERIMENTAL RESULTS

In this section, the prediction performances of SOCBR and other alternative models are compared. Table 2 describes the average prediction accuracy of each model.

In Table 2, SOCBR achieves higher prediction accuracy than COCBR, FWCBR, and ISCBR by 7.92%, 6.02%, 4.75%, and 3.96% for the hold-out data.

Table 2. Average prediction accuracy of the models

Model	COCBR	FWCBR	ISCBR	SOCBR
Test data set	-	84.83%	83.71%	85.96%
Hold-out data set	80.75%	83.18%	82.62%	86.17%

The McNemar tests are used to examine whether the predictive performance of the SOCBR 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 3 shows the results of the McNemar test to compare the

performances of five algorithms for the hold-out data.

Table 3. . McNemar values for the hold-out data

	FWCBR	ISCBR	SOCBR
COCBR	2.361	3.115**	10.453***
FWCBR		0.062	3.309*
ISCBR			4.208**

* significant at the 10% level, ** significant at the 5% level, *** significant at the 1% level

As shown in Table 3, SOCBR is better than COCBR at the 1% and better than ISCBR at the 5% statistical significance level. But, SOCBR outperforms FWCBR at only 10% statistical significance level.

CONCLUSIONS

We have suggested a new kind of hybrid system of GA and CBR to improve the performance of the typical CBR system. This paper used GA as a tool to optimize the feature weights and instance selection simultaneously. From the results of the experiment, we show that SOCBR, our proposed model, outperforms other comparative algorithms such as COCBR and FWCBR as well as ISCBR.

However, this study has some limitations. First of all, the number of generations (trial events) in our GA experiments is too small. In fact, the search space for simultaneous optimization of feature weights and feature selection is very wide area, so we need to increase the number of populations and generations. Secondly, it takes too much computational time for SOCBR. As mentioned, SOCBR iterates case retrieval process whenever genetic evolution occurs. And, in general, case retrieval process in CBR takes much computational time because it should search whole case-base to make just one solution. Consequently, the efforts to make SOCBR more efficient should be followed in future. Moreover, the generalizability of SOCBR should be tested further by applying it to other problem domains.

REFERENCES

Babu, T. Ravindra and Murty, M. Narasimha (2001). Comparison of genetic algorithm based prototype selection schemes, *Pattern Recognition*, 34, 523-525.

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

Cardie, C. (1993). Using decision trees to improve case-based learning. *Proceedings of the Tenth International Conference on Machine Learning*, Morgan Kaufmann: San Francisco, CA, 25-32.

Cardie, C. and Howe, N. (1997). Improving minority class prediction using case-specific feature weights. *Proceedings of the Fourteenth International Conference on Machine Learning*, Morgan Kaufmann: San Francisco, CA, 57-65.

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

Domingos, P. (1997). Context-sensitive feature selection for lazy learners. *Artificial Intelligence Review*, 11, 227-253.

Hart, P.E. (1968). The condensed nearest neighbor rule, *IEEE Transactions on Information Theory*, 14, 515-516.

Huang, Y.S., Chiang, C.C., Shieh, J.W. and Grimson, E. (2002). Prototype optimization for nearest-neighbor classification, Pattern Recognition, 35, 1237-1245.

A.K. Jain and R.C. Dubes, *Algorithms for clustering data*, Prentice Hall: NJ, 1988.

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

Kelly, J.D.J. and Davis, L. (1991). Hybridizing the genetic algorithm and the k nearest neighbors classification algorithm, *Proceedings of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann: San Diego, CA, 377-383

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

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

Kim, S.H. and Shin, S.W. (2000). Identifying the impact of decision variables for nonlinear classification tasks, *Expert Systems with Applications*, 18, 201-214.

Kuncheva, Ludmila I. & Jain, Lakhmi C. (1999). Nearest neighbor classifier: Simultaneous editing and feature selection. *Pattern Recognition Letters, 20*, 1149-1156.

Liao, T.W., Zhang, Z.M., and Mount, C.R. (2000). A case-based reasoning system for identifying failure mechanisms, *Engineering Applications of Artificial Intelligence*, 13, 199-213.

Lipowezky, Uri (1998). Selection of the optimal prototype subset for 1-NN classification, *Pattern Recognition Letters*, 19, 907-918.

Rozsypal, Antonin & Kubat, Miroslav. (2003). Selecting representative examples and

attributes by a genetic algorithm. *Intelligent Data Analysis, 7*, 291-304.
Sanchez, J.S., Pla, F. and Ferri, F.J. (1997). Prototype selection for the nearest neighbour rule through proximity graphs, *Pattern Recognition Letters*, 18, 507-513.
Shin, K. S. & Han, I. (1999). Case-based reasoning supported by genetic algorithms for corporate bond rating. *Expert Systems with Applications, 16*, 85-95.
Siedlecki, W. & Sklanski, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters, 10*, 335-347.
Skalak, D.B. (1994). Prototype and feature selection by sampling and random mutation hill climbing algorithms. *Proceedings of the Eleventh International Conference on Machine Learning*, New Jersey, 293-301.
Wang, Y. & Ishii, N. (1996). A method of similarity metrics for structured representations. *Expert Systems with Applications, 12*, 89-100.
Wettschereck, D., Aha, D.W., and Mohri, T. (1997). A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, *Artificial Intelligence Review*, 11, 273-314.
Wilson, R.L. and R. Sharda (1994). Bankruptcy prediction using neural networks, *Decision Support Systems*, 11, 545-557.
Yan, Hong (1993). Prototype optimization for nearest neighbor classifier using a two-layer perceptron, Pattern Recognition, 26, 317-324.
Yin, W. J., Liu, M., & Wu, C. (2002). A genetic learning approach with case-based memory for job-shop scheduling problems. *Proceedings of the First International Conference on Machine Learning and Cybermetics*, 1683-1687.
Yu, Kai, Xu, Xiaowei, Ester, Martin and Kriegel, Hans-Peter (2003). Feature weighting and instance selection for collaborative filtering: an information-theoretic approach, *Knowledge and Information Systems*, 5, 201-224.