The Development of a Credit Scoring System - A CBR Approach with GA Applied

SHIHCHIEH CHOU, DOUG SHEU, TAI-PING HSING, PATRIC S. CHEN
Department of Information Management
National Central University
Dept. of Information Management, National Central University, Chung-Li 320
TAIWAN, R.O.C.

Abstract: - To improve the classifying accuracy for credit rating, many techniques have been applied in the credit scoring methods. This research has developed a credit scoring system based on case-based reasoning (CBR) with the genetic algorithm (GA) applied in the *k*-nearest-neighbor (*k*-NN) technique for case retrieving. Performance of the system has been presented and examined under human selected factors. Also, advamced research with the same techniques applied to support loan amount rating has been designed and suggested.

Key-Words: - Case-Based Reasoning, Nearest Neighbor, Genetic Algorithm, Knowledge Management, Credit Scoring, Banking

1 Introduction

The quantitative credit scoring methods have been developed to classify credit applicants into one of two groups: a “good credit” group that is likely to repay the financial obligation, or a “bad credit” group that has a high likelihood of defaulting on the financial obligation[1]. To improve the classifying accuracy, many techniques have been applied in the scoring methods including parametric and nonparametric statistical methods, decision tree, neural network, etc. But the comparison research showed that there existed no method that could peform always better than others. For a particular data set, there existed a particular optimal algorithm[2].

To extend the application of the techniques in the credit assessment situation, this research is aimed to develop a credit scoring system based on case-based reasoning (CBR) with the genetic algorithm (GA) applied in the *k*-nearest-neighbor (*k*-NN) technique for case retrieving. CBR is applied here since it can learn knowledge from the past cases with little assumption or limitation of the input data. The GA is applied here since it can search the best weights for the distance metrics withnin the *k*-NN technique without human intervention.

In section 2, we study the related techniques first. In section 3, the methods and techniques of our system are descibed. In section 4, the performance of the system with these techniques applied has been presented and examined under human selected factors. In section 5, advanced research with the same techniques applied is presented. Finally, some conclusion is made in section 6.

2 Related Techniques

Linear discriminant analysis (LDA), a simple parametric statistical model, is the first and commonly employed model for credit assessment. The appropriateness of LDA for credit assessment has been questioned in past literatures. This has led to the development of more sophisticated parametric statistical methods and nonparametric statistical methods. CBR approach as we proposed is one of the development efforts.

Case-based reasoning can use old cases to assess new situations. This property makes it suitable for the credit assessing problem. First, the non-parametric nature of the method enables the modeling of irregularities in the credit assessment situation. Second, it could be easily explained to the business manager who would need to assess the credit in the daily work since it is conceptually simple and straightforward [3].

In our CBR approach for credit assessment, the *k*-NN technique is used for case retrieving. The *k*-NN is a standard technique in nonparametric statistics. It can be applied to classify a new case on the basis of a majority vote among the *k* most similar training-set cases to the new case. In the application, similarity can be measured in the space of the measured attributes using an appropriate distance metric[4]. Traditionally, the most popular measure of distance between two points x and y in the *k*-NN is as following:

$$D(x,y) = \sqrt{w_i \cdot \sum_{i=1}^{n} \left(x_i - y_i\right)^2} \qquad (1)$$

The application of the simplest type of *k*-NN technique, in which each new case is assigned to the class of its single nearest neighbor, to the credit assessing problem can be referred in Fogarty and Ireson [5].

Weighting the attributes is usually the most difficult part in the *k*-NN method [6]. The difficulty is resulted from the huge of the space in the searching of the best weight. Taking advantage of the GA's optimiaing efficiency, this study has applied the GA to the searching for the best weight of an attribute [7].

3 Methodology

In this study, we have developed a credit scoring system based on CBR method as figure 1 shows. The major works include:

Case representation. The appropriate attributes are selected to characterize a case and determine how cases are stored in the case library. The major purpose of these attribures is to allow a CBR system to retrieve one or more cases that are similar to the new case.

Case retrieval. The system uses the *k*-NN technique to retrieve the cases. The attributes are the Euclidean metrics of the *k*-NN technique. The GA can search the best weight for the Euclidean metrics. The Euclidean metric as equation (1) shows can be used to calculate the similarity between each training-set cases and the new case. When a borrower applies a new loan , the *k* most similar training cases are determined with the *k* most short distances.

Case reuse. Usually a new case may not exactly match the old ones. Reuse of the old cases can have many ways. Our strategy of the system is to use the majority vote of the *k*-nearest old cases as the class of the new case. The old cases also can be presented as an inspiration for solving the new problem.

Case storage. Once the new problem is solved, it is stored in the case library for future use.

3.1 Genetic Algorithm

As aforementioned, this system uses the GA to search for the best weights of the attributes. The search space of all possible weights of *k*-NN will be mapped onto a set of finite strings (called chromosomes) and each string has a corresponding point in the search space. The encoded chromosome represents a set of possible weights of the attributes in a case. The range of each weight is from 0 to 1. The fitness function is used as the performance index to judge the classification accuracy for the GA. The chromosome with the higher classification accuracy will have a higher probability to reproduce the next generation [8].

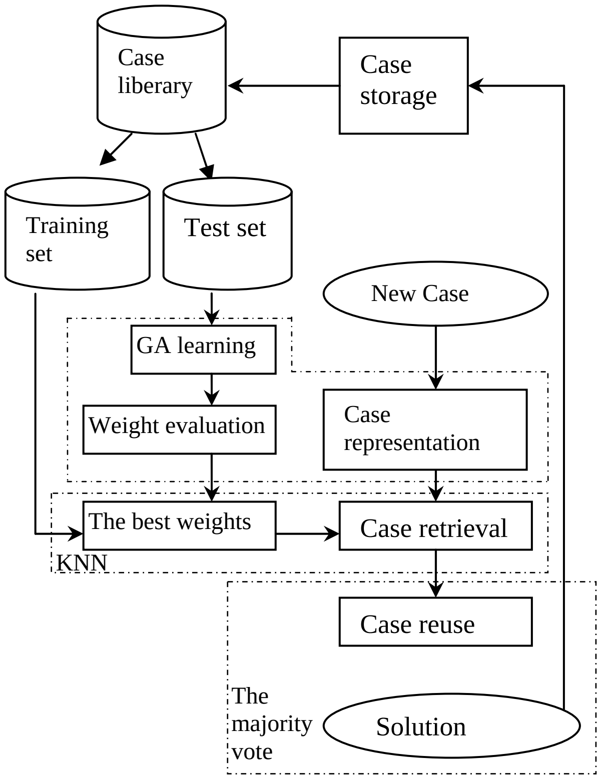

Fig.1 System infrastructure diagram

This study has formulated a fitness function to search for a chromosome which has the highest classification accuracy referring to repayment. The fitness function is defined as equation (2) shows:

$$Max \quad TR = \frac{\left[\sum_{i=1}^{n} TA_i\right]}{n} \qquad (2)$$

$$\begin{aligned}
s.t. \quad & TA_i = 1 \qquad if\ O(T_i) = O(S_{j-i})_k \\
& TA_i = 0 \qquad if\ O(T_i) \neq O(S_{j-i})_k \\
& S_{j-i} = \min\left[DIS_{RO}() + DIS_N()\right] \\
& DIS_{RO}() = \sqrt{\sum_{v=1}^{l} W_v (T_{iv} - L_{jv})^2} \\
& DIS_N() = \sqrt{\sum_{m=1}^{p} W_m \times D_m} \\
& D_m(T_{im}, L_{jm}) = 0 \quad if\ T_{im} = L_{jm} \\
& D_m(T_{im}, L_{jm}) = 1 \quad if\ T_{im} \neq L_{jm} \\
& i = 1, \cdots, n
\end{aligned}$$

TR, the fitness value in this system, represents the classification accuracy of the test set referring to repayment. TA_i represents the clasification accuracy of the ith case of the test cases. If it is accurate, then TA_i equals 1, else zero. *O*( ) represents the class of the test case or the class of the majority vote of the *k* most similar training cases. $O(T_i)$ is the class of the *i*th case among the test cases. This system get *k* training cases that are the most similar to the *i*th case among the test cases. $O(S_{j\text{-}i})_k$ represents the class of the majority vote of these *k* cases. $S_{j\text{-}i}$ indicates the similarity between the *j*th case of the training set and the *i*th case of the test set. DIS_{RO}( ) is the weighted distance function of quantitative or ordinal attributes. DIS_N( ) is the weighted distance function of nominal attributes. D_m( ) is the distance of nominal attributes. T_{ik} is the *k*th quantitative or ordinal attribute of the *i*th case in the test set. L_{jk} is the *k*th quantitative or ordinal attribute of the *j*th case in the training set. T_{im} is the *m*th nominal attribute of the *i*th case among the test set. L_{jm} is the *m*th nominal attribute of the *j*th case in the training set. W_v is the weight of the *v*th quantitative or ordinal attribute. W_m is the weight of the *m*th nominal attribute. *l* is the number of quantitative or ordinal attributes. *p* is the number of nominal attributes. *n* is the case number of the test set.

4 Evaluation

4.1 Data description

This study has simulated 1226 cases. 1000 cases of them are used as training and test cases. The other 226 cases are used to evaluate the performace of the system. Among training and test data, there are 860 good credit cases, and 140 bad credit ones. Among the evaluation cases, there are 199 good credit cases, and 27 bad credit ones. There are 1059 good credit cases, and 167 bad credit ones in all. Each case has 11 attributes including gender, age, marriage, ownership of the house, occupation, years of working, annual income, the number of banks and the amount of past loans, the amount of loan applying, the time duration of repayment, and guarantor. The repayment credit is classified into two classes as good or bad.

4.2 The implementation of the GA

Parameters of GA has been set as follows. The string length of the chromosome is set to be 77 bits with 7 bits to represent each of the 11 attributes. Genetic selecting method uses ‘the expected value method’ because its result is more stable than others. Crossover operation uses two-point crossover operation. The probability of crossover is set to be 0.7. Generally, there is a better evolution result when the rate of crossover is between 0.6~0.9. The probability of mutation is set to be 0.001. Setting the probability of mutation can avoid producing a local optimal solution. However, a higher probability of mutation may lead to the phenomenon of a random search. Hence, mutation should be set in a very low probability. The initial population size is set as 200. Generally, a larger population size will reduce the search speed of the GA, but it will increase the probability of finding a high quality solution. Referring to the generation, we have set 50 as the stop condition after the pilot test. The choice of the generation is a balance between the better result and the speed of the GA performance.

4.3 Experiment design

In this study, performance of the system could be affected by human selected factors including the proportion of the training and test cases and the number of voting cases. Therefore, the experiment is designed to present the performance and examine the human selected factors that may affect the performance of the system.

To test the affection of the proportion of the training and test cases on classification accuracy, 4 experiments are conducted with proportions 80:20, 70:30, 50:50, and 30:70 each. In each experiment, the affection of the number of voting cases is tested by setting the number to be 1, 3, 5, 7, 11, 13, 15, 17, 19, 21, 25, 31, 35, 41, 45, 51.

The experiment has evaluated the classification accuracy of the good and the bad credit groups respectively.

4.4 Results

Experiment 1: The training set is 800 cases and the test set is 200 cases (the proportion is 80:20). The number of voting cases are 1, 3, 5, 7, 11, 13, 15, 17, 19, 21, 25, 31, 35, 41, 45, 51 respectively. The evaluation set is 226 cases.

Referring to the good credit group as shown in Fig. 2, there will be the highest classification accuracy rate (92.96%) when the vote number is 1 and the lowest classification accuracy rate (56.28%) when the votenumber is 51. Referring to the bad credit group as shown in Fig. 2, there will be the highest classification accuracy rate (81.48%) when the vote number is 7 and the lowest classification accuracy rate (48.15%) when the vote number is 1. When the vote number is 5, the classification rates for the good and the bad credit groups are both over 76%.

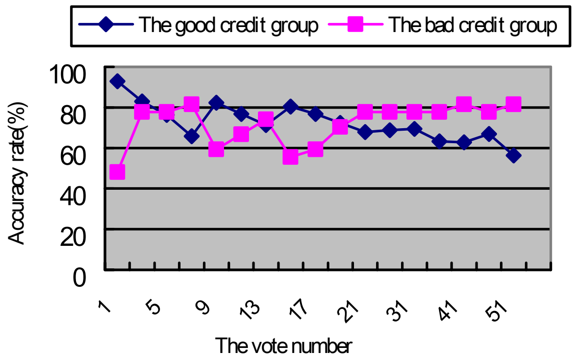

Fig.2 Accuracy rate at porprotion 80:20

Experiment 2: The training set is 700 cases and the test set is 300 cases (the proportion is 70:30). The number of voting cases are 1, 3, 5, 7, 11, 13, 15, 17, 19, 21, 25, 31, 35, 41, 45, 51 respectively. The evaluation set is 226 cases.

Referring to the good credit group as shown in Fig.3, there will be the highest classification accuracy rate (93.97%) when the vote number is 1 and the lowest classification accuracy rate (57.79%) when the vote number is 45. Referring to the bad credit group as shown in Fig. 3, there will be the highest classification accuracy rate (81.48%) when the vote number is 5 and the lowest classification accuracy rate (51.85%) when the vote number is 1. When the vote number is 5, the classification rates for the good and the bad credit groups are both over 71%.

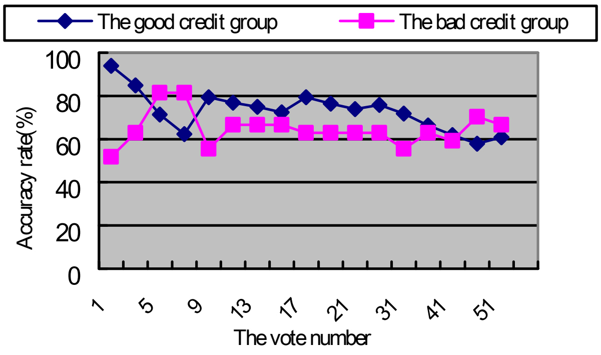

Fig.3 Accuracy rate at porprotion 70:30

Experiment 3: The training set is 500 cases and the test set is 500 cases (the proportion is 50:50). The number of voting cases are 1, 3, 5, 7, 11, 13, 15, 17, 19, 21, 25, 31, 35, 41, 45, 51 respectively. The evaluation set is 226 cases.

Referring to the good credit group as shown in Fig.4, there will be the highest classification accuracy rate (92.46%) when the vote number is 1 and the lowest classification accuracy rate (53.32%) when the vote number is 51. Referring to the bad credit group as shown in Fig.4, there will be the highest classification accuracy rate (92.59%) when the vote number is 17, 19, 25 and the lowest classification accuracy rate (33.33%) when the vote number is 1. When the vote number is 5, the classification rates for the good and the bad credit groups are both over 73%.

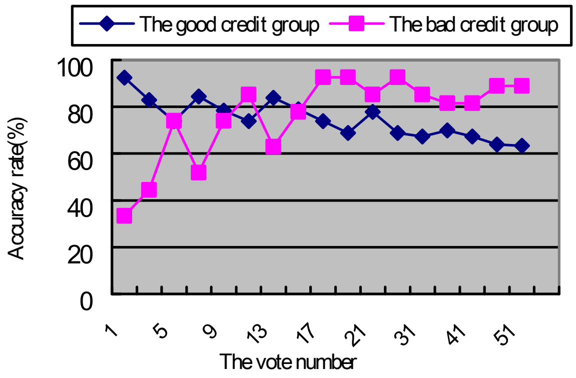

Fig.4 Accuracy rate at porprotion 50:50

Experiment 4: The training set is 300 cases and the test set is 700 cases (the proportion is 30:70). The number of voting cases are 1, 3, 5, 7, 11, 13, 15, 17, 19, 21, 25, 31, 35, 41, 45, 51 respectively. The evaluation set is 226 cases.

Referring to the good credit group as shown in Fig.5, there will be the highest classification accuracy rate (89.95%) when the vote number is 1 and the lowest classification accuracy rate (60.30%) when the vote number is 21, 45 and 51. Referring to the bad credit group as shown in Fig.5, there will be the highest classification accuracy rate (85.19%) when the vote number is 7 and the lowest classification accuracy rate (25.93%) when the vote number is 1. When the vote number is 5, the classification rates for the good and the bad credit groups are both over 69%..

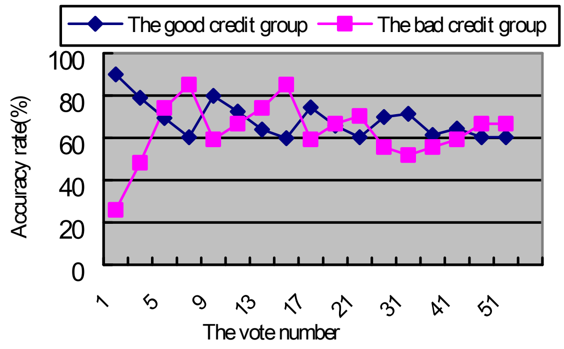

Fig.5 Accuracy rate at porprotion 30:70

4.5 Discussion

The experiment results are discussed as follows:

(1) Affection of the proportion of the training and test cases on system performance.

When a new case is classified by the most similar case, the accuracy rates of the classification for the good and the bad credit cases can be summarized in

Table 1. The results show that the classification accuracy for the good credit cases maintain a high rate over 89% for all four experiments. However, the classification accuracy for the bad credit cases have a low rate below 52% for all four experiments, and the accuracy rate of the classification has a sharp drop at experiment 3 and 4.

Table 1 The accuracy rates of classification with the most similar case.

Experiment	Good credit case(%)	Bad credit case(%)
1	92.96	48.15
2	93.97	51.85
3	92.46	33.33
4	89.95	25.93

When a new case is classified by cases voting, the average and the standard deviation of the accuracy rates of the classification for the good and the bad credit cases can be summarized in Table 2. The results show that the classification accuracy for the good and the bad credit cases both maintain a stable average rate between 65-78%. However, the standard deviations of the accuracy rates of the classification for the bad credit cases become bigger at experiment 3 and 4.

Table 2 The average and standard deviation for accuracy rates of classification with cases voting.

	Good credit case(%)		Bad credit case(%)	
Experiment	Average	Standard deviation	Average	Standard deviation
1	71.29	7.59	73.38	8.58
2	71.64	7.75	65.51	7.37
3	73.59	6.92	78.70	14.41
4	67.03	6.86	65.28	10.81

Observation of the data as previously described reveals that proportion of the training set cases could has affection on the performance of the system. When the training set cases has a low proportion under 50% (experiment 3 and 4), the accuracy rates of classification for the bad credit cases turn out to be lower and unstable.

(2) Affection of the number of voting cases on system performance.

Referring to the results of the 4 experiments, 1 voting case has the effect of high accuracy for classifying good credit cases and low accuracy for classifying bad credit cases; and 5 voting cases comes out to have the most stable classifying accuracy for both good and bad credit cases.

Observation of the data reveals that the selection of the number of voting cases could have some meaning to the credit assessment strategy. Since 1 voting case has high accuracy for classifying good credit cases and low accuracy for classifying bad credit cases, it would make more loans to be granted. This would be appropriate if the bank would implementd a high profit high risk strategy. On the other hand, 5 voting cases has a fairly high and stable accuracy for classifying both good and bad credit cases, it would make the bank more severely however more correctly in granting the loan. This would be appropriate if the bank would implementd a low profit low risk strategy.

5 Advanced Research

Together with the credit rating, CBR approach of our system can support the loan amount rating also. After credit rating, the good credit group retrived could be used. If we have the loan amount rating classified into several classes, the fitness function as equation (3) shows could be used in the system to perform loan amount rating.

$$
\begin{aligned}
&Max \quad TR = \frac{\left[\sum_{i=1}^{n} TA_i\right]}{n} \qquad (3)\\
&s.t. \quad TA_i = | O(T_i) - O(S_{j-i})_k |\\
&\qquad S_{j-i} = \min\left[DIS_{RO}() + DIS_N()\right]\\
&\qquad DIS_{RO}() = \sqrt{\sum_{v=1}^{l} W_v (T_{iv} - L_{jv})^2}\\
&\qquad DIS_N() = \sqrt{\sum_{m=1}^{p} W_m \times D_m}\\
&\qquad D_m(T_{im}, L_{jm}) = 0 \quad if\; T_{im} = L_{jm}\\
&\qquad D_m(T_{im}, L_{jm}) = 1 \quad if\; T_{im} \neq L_{jm}\\
&\qquad i = 1, \cdots, n
\end{aligned}
$$

TR represents the amount-class similarity of the test set. *O*( ) is the the class of the test case or the class of the majority vote of the *k* most similar training cases including (1) 100 ~ 190 thousand NT dollars granted, (2) 200 ~ 290 thousand NT dollars, (3) 300 ~ 390 thousand NT dollars, (4) 400~ 490 thousand NT dollars, (5) over 500 thousand NT dollars. $O(T_i)$ is the class of the *i*th case of the test set. This system get *k* cases from training set, they are the *k* most similar cases to the *i*th case of the test set. $O(S_{j-i})_k$ is the class of the majority vote of these *k* cases. TA_i is the similarity between $O(T_i)$ and $O(S_{j-i})_k$. The smaller the TA_i is, the closer the class of the test case and the class of the majority vote among the *k* most simlar training cases are. Other variables are described the same as equation(2).

6 Conclusion

This study takes a CBR approach to work on the credit assessment problem. It has developed a credit scoring system to demonstrate the using of the k-NN technique with the GA applied in the searching of the similar cases. Performance of the system reveals that the application of the techniques in solving the credit assessment problem could be feasible. However, the human selected factors should be carefully examined in the application of the techniques. As the experiments show, inappropriate proportion of the training and test cases could result in unstable performance; different strategies of using the most similar case or cases voting could be appropriate for different credit assessment strategies. Finally, this research has suggested that loan amount rating could also be performed using the same techniques.

References:

[1] West, D., Neural network credit scoring models, *Computers & Operations Research*, Vol. 27, 2000, pp. 1131-1152.

[2] Liu, Y., A framework of data mining application process for credit scoring, *Research paper*, Institute of Information Systems, University of Goettingen, Göttingen, 2002.

[3] Henley, W. E., Statistical aspects of credit scoring, *PhD Dissertation*, The Open University, Milton Keynes, UK, 1995.

[4] Henley, W. E., and Hand, D. J., Construction of a k-nearest-neighbor credit-scoring system, *IMA Journal of Mathematics Applied in Business & Industry*, Vol. 8, 1997, pp. 305-321.

[5] Fogarty, T. C , and Ireson, N. S., Evolving Bayesian classifiers for credit control—a comparison with other machine learning methods, *IMA Journal of Mathematics Applied in Business and Industry*, Vol. 5, 1993, pp. 63-75.

[6] Barletta, R., An introduction to case-based reasoning, *AI Expert*, Vol. 6, 1991, pp. 42-49.

[7] Davis, L., *Handbook of genetic algorithms*, Amsterdam: Van Nostrand Reinhold, New York, 1991.

[8] Huang, M. J., and Huang, H. S., Chen, M. Y., Constructing a personalized e-learning system based on genetic algorithm and case-based reasoning approach, *Expert Systems with Applications*, In Press, Corrected Proof, Available online, 9 June, 2006.