Proceedings of the 2007 IEEE Symposium on
Computational Intelligence and Data Mining (CIDM 2007)

A Modified Genetic Programming for Behavior Scoring Problem

CHEN Qing-Shan [1], ZHANG De-Fu [1+], WEI Li-Jun [1], CHEN Huo-Wang[2,3]
1(Department of Computer Science, Xiamen University, Xiamen 361005, China)
2(Longtop Group Post-doctoral Research Center, Xiamen, 361005, China)
3(School of Computer, National University of Defense Technology, Changsha, 410073, China)
+ Corresponding author: Phn: +86-592-5918207, Fax: +86-592-2183502, E-mail: dfzhang@xmu.edu.cn

***Abstract*-Behavior scoring is an important part of risk management in financial institutions, which is used to help financial institutions make better decisions in managing existing customers by forecasting their future credit performance. In this paper, a modified genetic programming (MGP) is introduced to solve the behavior scoring problems. A real life credit data set in a Chinese commercial bank is selected as the experimental data to demonstrate the classification accuracy of this method. MGP is compared with back-propagation neural network (BPN), and another GP that uses normalized inputs (NGP), the experimental results show that the MGP has slight better classification accuracy rate than NGP, and outperforms BPN in dealing with behavior scoring problems because of less historical samples of credit data in Chinese commercial banks.**

Keywords: Data Mining; Genetic Programming (GP); Behavior Scoring; Back-propagation Neural Network (BPN).

Ⅰ INTRODUCTION

Forecasting credit risk has become more and more important in financial institutions, and is one of the applications that have obtained serious attention over the past decades. Modeling techniques like traditional statistical analyses and artificial intelligence techniques have been developed in order to tackle this task [1]. A good model not only helps financial institutions make correct decisions quickly, but also helps them to avoid potential loss. Hence, developing a more proper model is an important task for researchers.

Credit scoring [2, 3] and behavior scoring [4, 5] are the techniques that help financial institutions decide whether or not to grant credit to applicants. There are two types of decisions financial institutions have to make. One is how to grant credit to a new applicant. Credit scoring can help them answer this question. The other is how to make credit limit or marketing strategies to existing customers. Behavior scoring is a tool designed to be special for this question. This paper is trying to deal with the latter decision making problem. Behavior scoring model is used to help decision-maker make better decisions in managing existing customers by forecasting their future credit performance.

Currently, researchers have developed a lot of traditional statistical methods and artificial intelligence tools for behavior scoring, such as rough sets, k-nearest neighbor, decision tree, and artificial neural network (ANN) [6-9]. We believe the application of GP in this field is a promising research area, since GP has the advantage of performing a global search in the space of candidate rules. In the context of classification rules discovery, in general GP makes it cope better with attribute interaction than greedy rule induction and decision trees. In fact, several papers have been proposed to discover intelligible classification rules using GP [10-12]. But, those papers cannot provide the concise and useful classification rules. In this paper, MGP for discovering the intelligible classification rules is developed; the computational results show that MGP is very efficient.

The paper is organized as follows: Section Ⅱ describes the basic concepts of the GP. MGP is proposed to solve behavior scoring problems in Section Ⅲ. The experimental results are reported in Section Ⅳ. Conclusion is provided in Section Ⅴ.

*This paper is supported by academician start-up fund (Grant No. X01109) and 985 information technology fund (Grant No. 0000-X07204) in Xiamen University.

II BASIC CONCEPTS OF GP

GP was first proposed by Koza [13], which had been used in a range of problems including classification [14], and symbolic regression [15]. When using GP, the aim is to automatically extract intelligible classification rules for each class in a database. The representation of GP can be viewed as a tree-based structure composed of the function set and terminal set. The function set is the operators, functions or statements such as arithmetic operators or conditional statements which are available in the GP. The terminal set includes both variables, and constants. For example to express $(x + (y \div 3))$, the GP-tree can be represented as Fig.1.

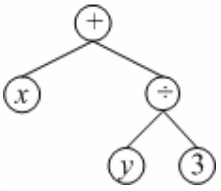

Fig.1. The representation of GP tree.

The major steps of genetic programming can be formalized as follows: generate at random an initial population of individuals representing potential solutions to the classification problem for the class at hand; evaluate each individual on the training set by a fitness function; on the basis of Darwin's evolutionary theory, select genetic operators (e.g. crossover, copy, mutation) to produce new individual, until an acceptable classification individual is found or the specified maximum number of generations has been reached. Next, we will introduce three main operators, crossover, mutation, and copy.

In GP, the crossover operates on two individuals, and produces two child individuals. Two random nodes are selected from within each individual and then the resultant sub-trees are swapped, generating two new individuals. These new individuals become a part of the next generation of individuals to be evaluated. An example of a crossover operator in GP is shown in Fig.2.

GP uses the mutation operator to avoid falling into the local optima. The mutation operator can be applied to either a function node or a terminal node. Randomly select a node in a sub-tree and replace it with a new created sub-tree randomly. Finally, the copy operator can choose an individual in the current population and copy it without changes into the new population.

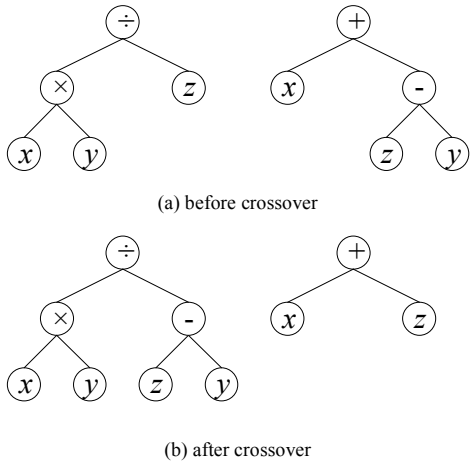

Fig.2. The effect of a crossover operation.

III MGP FOR BEHAVIOR SCORING

In this section, we depict the procedures of the MGP system which is used to solve the behavior scoring problems.

A. Encoding

In order to obtain the classification rules efficiently. First, discretization of continuous attributes should be employed before MGP. Many algorithms have been proposed to deal with it. In this paper, we select the Boolean reasoning algorithm [16]. Then, the maximum GP-tree depth of six is enforced to ensure for obtaining a simple rule. In addition, the function set only consists of the logical connectives {AND, OR, NOT}, and the relation operators $\{\leq, =, \geq\}$. The terminal set is simply predicting attributes A_i or values in the domain of A_i.

The procedures of creating the individuals in the initial population follow some constrains:

(1) Randomly select from the function or terminal set.

(2) A relational operator' right child can only be the predicting attributes A_i, and its left child can only be value in the domain of A_i.

(3) An internal node's parent can only be a logical connective.

(4) The root node can only be a logical connective.

For example, if the good credit customer can be represented using the GP-tree as shown in Fig.3, then the rule can be

interpreted as

IF ($A_1 \geq 3$ AND ($A_2 \leq 6$ OR $A_5 = 1$))

THEN customer = good credit.

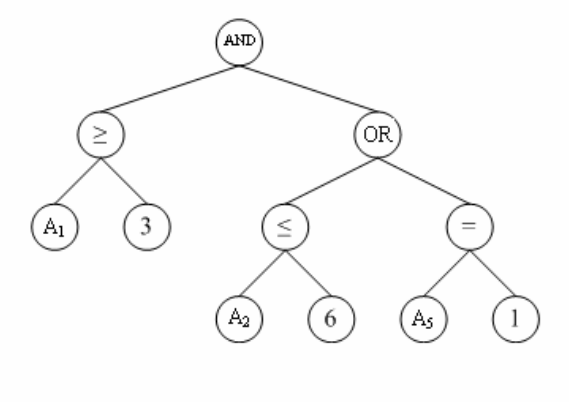

Fig.3. The representation of the classification rule using the GP-tree.

B. Genetic operators

Our model selects crossover, copy, and mutation operators. In order to avoid producing an invalid child, some restrictions have been imposed. An internal node does not swap with a leaf node. Only the compatible operators can be swapped.

C. Fitness function

The measurement of fitness is a rather nebulous subject since it is highly problem-dependent. In this paper, the fitness function is similar to that first proposed in [17].

Before the definition of the fitness function, a review of some basic concepts on classification-rule evaluation is necessary. When using a rule for classifying a data instance (a record of the data set being mined), depending on the class predicted by the rule and on the actual class of the instance, one of the following four types of result can be observed:

(1) True positive: The rule predicts that the instance has a given class and the instance does have that class.

(2) False positive: The rule predicts that the instance has a given class but the instance does not have it.

(3) True negative: The rule predicts that the instance does not have a given class, and in deed the instance does not have it.

(4) False negative: the rule predicts that the instance does not have a given class but the instance does have it.

Let t_p, f_p, t_n and f_n denote respectively the number of true positives, false positives, true negatives and false negatives observed when a rule is used to classify a set of instances, the fitness function of MGP can be described as

$$ff_i = \frac{t_p}{t_p + f_n} + \frac{t_n}{t_n + f_p}$$

where ff_i denotes the predictive accuracy.

D. Classification of newly entered instances

Now, we introduce the procedures of classifying newly entered instance. We derive the rules as simply as possible for each class so that only a few rules are derived to represent the general trend of each class. If we want to derive another rule in a class, the data which satisfy any rule should not be trained again. Then we can extract k types of classification rule set from the train data set, where k denotes the number of classes.

When a newly entered instance is fed in the model, the instance can be classified as the following situations: first, if the instance satisfies one of the k rule sets, the instance is simply assigned the class; second, if the instance satisfies more than one rule set, the instance is assigned the class predicted by the rule with the best fitness; finally, if the instance does not satisfy any rule in the rule sets, the instance is assigned to the default class which is the class of the majority of the instances in the sample.

Ⅳ The Experimental Results

A credit card data set provided by a Chinese commercial bank is used to demonstrate the feasibility and effectiveness of the proposed method. The data set is in recent eighteen months, and includes 599 instances. Each instance in the data set contains 17 independent variables. The decision variable is the customer credit: good, bad, and normal credit. The number of good, normal, and bad is 160, 225, and 214. The detailed descriptions of those variables are listed in TABLE Ⅰ.

In this paper, MGP is compared with BPN which has been successfully applied to credit analysis [18, 19]. The main purpose of this comparison is not trying to show that MGP is better than the BPN under all circumstances; we will simply focus on testing the advantages of MGP in the small data set at the situation of China. In addition, MGP also compared with

NGP proposed in Reference [11] which uses normalized inputs.

TABLE Ⅰ

DATA SET DESCRIPTION

Index	Description	Index	Description
1	Sex	10	Address
2	Age	11	Telephone
3	Card credit limits	12	Occupation
4	Last customer credit	13	Total asset
5	Last Card credit limits	14	Total saving
6	Customer type	15	Flow asset
7	Education level	16	Debt total
8	marriage status,	17	Average of saving
9	Month income		
	Decision attribute		Customer credit

The MGP, NGP, and BPN are developed by using the C++ language. The parameters of MGP can be described as follows: the population size is 1000; the maximum number of generations is 50, the crossover probability is 0.9; the mutation probability is 0.1; and the selection operator is lexicographic parsimony pressure tournament [20]. For the BPN, several options of the neural network configurations are tested, we select 17-32-1 [21]. The learning rate and momentum are set to 0.75 and 0.15, respectively. For the NGP, the detail design refers to Reference [11].

To provide a reliable estimate and minimize the impact of data dependency in developing behavior scoring models, k-fold cross-validation is used to generate random partitions of the credit data sets [22]. In this procedure, the credit data set is divided into *k* independent groups. MGP, NGP, and BPN are trained by using the (*k*-1) groups of samples and tested by using the remained group. This procedure is repeated until each of the groups has been used as a test set once. The overall scoring accuracy was reported as an average across all *k* groups. In this paper, the value of k was set to 5 and thus forms a 5-fold cross-validation. Since the training of MGP, NGP, and BPN is a stochastic process, five iterations of the MGP, NGP, and BPN in the same data sets, the final results in each group are an average of five iterations.

Next, we can use MGP to discover intelligible classification rules. For the first group, the classification rules are shown in TABLE Ⅱ. The classification accuracy rate obtained by MGP, NGP, and BPN are shown in TABLE Ⅲ.

TABLE Ⅱ

RULES ARE DERIVED FROM THE FIRST ITERATION

	Rule	Class
1	IF (Total asset >= 496,216) AND (Address = Register) THEN	Good credit
2	IF (Total asset >= 205,516) AND (Total asset <= 263,792) THEN	Normal credit
3	IF (Total asset <= 191,306) OR (Last customer credit = 0) THRN	Bad credit

TABLE Ⅲ

CLASSIFICATION ACCURACY RATE (%) ON THE TEST DATA SETS

	MGP	NGP	BPN
Group 1	89.94	89.17	86.59
Group 2	90.50	89.13	87.24
Group 3	91.62	90.32	88.92
Group 4	88.27	88.12	86.76
Group 5	88.83	87.75	84.31
Overall	89.83	88.73	86.76

From the TABLE Ⅲ, we can observe that the modified MGP method developed in this paper has higher classification accuracy rate than the NGP and BPN. In addition, MGP and NGP have better performance than BPN in both classification accuracy rate and rule comprehensive. The computational results further verify the conclusion in reference [23] which turns out ANN is only suited for lager data sets. Therefore, we can conclude that MGP outperforms BPN in dealing with behavior scoring problems in Chinese commercial banks which have small historical samples.

Ⅴ CONCLUSION

In this paper, a modified MGP is introduced to solve the behavior scoring problems in a Chinese commercial bank. The experimental results show that MGP performs well in the small samples. However, in China's situation, many commercial banks have no enough historical data. As a result, MGP has an extensive application in China.

In this paper, when a newly entered instance does not

satisfy any rule or satisfy more than one rule, we roughly classify it to a class, how to classify it precisely will be investigated in our future work. In addition, Combining MGP with other artificial intelligence technologies may also be our future research work.

REFERENCES

[1] Mehmed Kantardzic, *Data Mining: Concepts, Models, Methods and Algorithms*, IEEE Press, 2002.

[2] Capon N., "Credit scoring system: A critical analysis," Marketing, vol.46, pp.82-91, 1982.

[3] Thomas, Lyn C., Edelman, David B. Edelman, and Jonathan N Crook, *Credit Scoring and its Applications*, Philadelphia, USA, SIAM, 2002.

[4] Michael Banasiak, "Behavior Scoring," Business Credit, vol.103, pp.52-55, 2001.

[5] Lyn C. Thomas, "A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers", International Journal of Forecasting, vol.16, pp. 149–172, 2000.

[6] Beynon, M., and Peel, M., "Variable precision rough set theory and data discretization: an application to corporate failure prediction," Omega-International Journal of Management Science, vol.29, No.6, pp.561-576, 2001.

[7] Henley W E, and Hand D J, "A k-nearest neighbor classifier for assessing consumer credit risk," Statistician, vol.44, No.1, pp.77-5, 1996.

[8] Davis RH, Edelman DB, and Gammerman AJ, "Machine-learning algorithms for credit-card applications," IMA Journal of Mathematics Applied in Business and Industry, vol.4, pp.43-52, 1992.

[9] Huang, Z Chen, H., Hsu, C.J., Chen, W.-H, and Wu, S, "Credit Rating Analysis with Support Vector Machines and Neural Network: A Market Comparative Study", Decision Support Systems, Vol.37, no.4, pp.543-558, 2004.

[10] H.E. Johnson, R.J. Gilbert, M.K. Winson, R. Goodacre, A.R. Smith, J.J. Rowland, M. A. Hall, and D.B. Kell, "Explanatory analysis of the metabolome using genetic programming of simple, interpretable rules," Genetic Programming and Evolvable Machines, vol.1, no.3, pp.243-258, 2000.

[11] De Falco, A. Della Cioppa, and E. Tarantino, "Discovering interesting classification rules with genetic programming," Applied Soft Computing, vol.1,no.3,pp.257-269,2002.

[12] A.A. Freitas, "A genetic programming framework for two data mining tasks: classification and generalized rule induction," in: Proceedings of the Second Annual Conference on Genetic Programming, Morgan Kaufmann, San Francisco, pp. 96–101, 1997.

[13] J.R. Koza, *Genetic programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA: MIT Press, 1992.

[14] Yifeng Zhang, and Siddhartha Bhattacharyya, "Genetic programming in classifying large-scale data: an ensemble method," Information Science, vol.163, no.3, pp.85-691, 2004.

[15] J. W. Davidson, D. A. Savic, and G. A. Walters, "Symbolic and numerical regression: Experiments and applications," Information Sciences, vol.150, no.2, pp.95-117, 2003.

[16] Nguyen, and H. Son, "Discretization of real value attributes: Boolean reasoning approach," Dissertation, Warsaw University, Warsaw, 1997.

[17] Lopes HS, Coutinho MS, Heinisch R, Barreto JM, and Lima WC, "A knowledge-based system for decision support in chest pain diagnosis," Med Biol Eng Comput, vol.35, no.1, pp:514, 1997.

[18] Bart Baesens, Rudy Setiono, Christophe Muse, and Jan Vanthienen, "Using Neural network rule extraction and decision tables for credit-risk evaluation," Management Science, vol.49, No.3, pp.312-329, 2003.

[19] Mahlhotra, R., and Malhotra D.K., "Evaluating consumer loans using neural networks," OMEGA: The International of Management Science, Vol.3, No.2, pp.83-96, 2003.

[20] Sean Luke, Liviu Panait, "Lexicographic Parsimony Pressure," Proceedings of the Genetic and Evolutionary Computation Conference, pp.829-836, 2002.

[21] J. Dayhoff, *Neural-Network Architecture: An Introduction*. New York: Van Nostrand Reinhold, 1990.

[22] Nan-Chen Hsieh, "Hybrid mining approach in the design of credit scoring models," Expert Systems with Applications, vol.28, pp.655–665, 2005.

[23] Nath R., Rajagopalan B., and Ryker, R., "Determining the saliency of input variables in neural network classifiers," Computers and Operations Research, vol.24, no.8, pp.767–773, 1997.