



Two-stage genetic programming (2SGP) for the credit scoring model

Jih-Jeng Huang^a, Gwo-Hshiung Tzeng^{b,c,*},
Chorng-Shyong Ong^a

^a *Department of Information Management, National Taiwan University, No. 1, Sec. 4,
Roosevelt Road, Taipei 106, Taiwan*

^b *Institute of Management of Technology and Institute of Traffic and Transportation College
of Management, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan*

^c *Department of Business Administration, Kainan University, No. 1, Kai-Nan Road, Luchu,
Taoyuan 338, Taiwan*

Abstract

Credit scoring models have been widely studied in the areas of statistics, machine learning, and artificial intelligence (AI). Many novel approaches such as artificial neural networks (ANNs), rough sets, or decision trees have been proposed to increase the accuracy of credit scoring models. Since an improvement in accuracy of a fraction of a percent might translate into significant savings, a more sophisticated model should be proposed for significantly improving the accuracy of the credit scoring models. In this paper, two-stage genetic programming (2SGP) is proposed to deal with the credit scoring problem by incorporating the advantages of the IF–THEN rules and the discriminant function. On the basis of the numerical results, we can conclude that 2SGP can provide the better accuracy than other models.

© 2005 Published by Elsevier Inc.

* Corresponding author. Address: Institute of Management of Technology and Institute of Traffic and Transportation College of Management, National Chiao Tung University, 1001 Ta-Hsueh Road, Hsinchu 300, Taiwan.

E-mail address: u5460637@ms16.hinet.net (G.-H. Tzeng).

Keywords: Credit scoring model; Artificial neural network (ANN); Decision trees; Rough sets; Two-stage genetic programming (2SGP)

1. Introduction

Credit scoring models have been widely used by financial institutions to determine if loan customers belong to either a good applicant group or a bad applicant group. The advantages of using credit scoring models can be described as the benefit from reducing the cost of credit analysis, enabling faster credit decision, insuring credit collections, and diminishing possible risk [1,2]. Since an improvement in accuracy of a fraction of a percent might translate into significant savings [2], a more sophisticated model should be proposed to significantly improve the accuracy of the credit scoring model in this paper.

In order to obtain a satisfactory credit scoring model, numerous methods have been proposed. Roughly, we can divide these models into the function-based methods (e.g. discriminant analysis, logistic regression, and artificial neural network (ANN)) and induction-based (also called IF–THEN rule) methods (e.g. rough sets, classification and regression tree (CART) and C4.5/5.0). For function-based methods, ANN is the most popular tool used for credit scoring and has been reported that its accuracy is superior to that of traditional statistical methods in dealing with credit scoring problems, especially in regards to non-linear patterns [3–7]. On the other hand, however, ANN has been criticized for its poor performance when incorporating irrelevant attributes or small data sets [8–10].

For induction-based algorithms, the main advantage is that they can provide the intelligence rules for decision-makers. These intelligence rules can help decision-makers to understand the contents of the data sets and can be employed for other marketing strategies. Although these induction-based methods have been well developed and successfully used in many applications such as failure prediction [1,11,12], and association rules [13], the main problem of induction-based methods is the ability of forecasting. It is clear that if a newly entered object does not match any rule, it cannot be determined to which class it belongs [14]. However, on the other hand, the function-based methods can provide the capability of forecasting but suffer from the drawback of lacking intelligence rules.

In order to combine the advantages of both, in this paper, we propose two-stage genetic programming (2SGP) to integrate the function-based and the induction-based methods. First, the IF–THEN rules are derived using genetic programming (GP). Next, the reduced data are fed into GP again to form the discriminant function for providing the capability of forecasting. The reasons which we employ GP to propose the credit scoring model is that GP can auto-

matically and heuristically determine the adequate discriminant functions and the valid attributes simultaneously. In addition, unlike ANNs which are only suited for large data sets, GP can perform well even in small data sets [10]. In addition, two real credit scoring data sets are used to demonstrate the proposed method and compare with other conventional methods. On the basis of the numerical results, we can conclude that 2SGP can significantly improve the accuracy of the credit scoring model and is superior to the conventional methods.

The rest of this paper is organized as follows. Section 2 reviews the conventional credit scoring models. Two-stage genetic programming for the credit scoring problems is proposed in Section 3. Two real-world examples are used to demonstrate the proposed method and to compare with other models are in Section 4. Discussions are presented in Section 6 and conclusions are in Section 7.

2. Credit scoring models

In this section, we describe two popular models used in building credit scoring models. The first model is logistic regression, which is mostly used for classification problems in the area of statistics. The second model is ANN, which is known for its excellent ability of learning non-linear relationships in a system.

2.1. Logistic regression

Logistic regression model is one of the most popular statistical tools for classification problems. Logistic regression model, unlike other statistical tools (e.g. discriminant analysis or ordinary linear regression), can fit various kinds of distribution functions such as Gamble, Poisson, and normal distributions [15] and is more suitable for the fraud detection problems. In addition, in order to increase its accuracy and flexibility, several methods have been proposed to extend the traditional binary logistic regression model including multinomial logistic regression model [16–20] and logistic regression model for ordered categories [21]. Therefore, the generalized logistic regression model is the general form of binary logistic regression model and multinomial logistic regression model.

Let a p -dimensional explanatory variables $x' = (x_1, x_2, \dots, x_p)$ and Y be the response variable with categories $1, 2, \dots, r$. Then the multinomial logistic regression model be given by the equation

$$\text{logit}(\pi) = \ln \left[\frac{P(Y = j|x)}{P(Y = k|x)} \right] = x' \beta_j, \quad 0 \leq j \leq r, \quad j \neq k, \quad (1)$$

where β_j is a $(p + 1)$ vector of the regression coefficients for the j th variable.

Let the last response level be the reference level and then the response probabilities $\pi_1, \pi_2, \dots, \pi_r$ can be calculated by the equations

$$\pi_r \equiv P(Y = r|x) = \frac{e^{x\beta_r}}{\sum_{l=1}^r e^{x\beta_l}} = \frac{e^{x\beta_r}}{e^{x\beta_r} + \sum_{l=1}^{r-1} e^{x\beta_l}} = \frac{1}{1 + \sum_{l=1}^{r-1} e^{x\beta_l}}, \quad (2)$$

$$\pi_j \equiv P(Y = j|x) = \pi_r e^{x\beta_j}, \quad 1 \leq j \leq r-1, \quad (3)$$

where l is a response level, and

$$l = l(\beta_j, 1 \leq j \leq r, j \neq k) = \sum_{i=1}^n \ln(P(Y = y_i|x_i)), \quad l \in \{1, 2, \dots, r\} \quad (4)$$

is the ln likelihood for the multinomial logistic regression model and $\{(y_i, x_i) | 1 \leq i \leq n\}$ denotes the sample of n objects. When the category is equal to two, the multinomial logistic regression model reduces to a binary logistic regression model.

Although logistic regression model can perform well in many applications, when the relationships of the system are non-linear, the accuracy of logistic regression decreases and ANN has been proposed to deal with this problem.

2.2. Artificial neural network

Artificial neural networks were developed to mimic the neurophysiology of the human brain to be a type of flexible non-linear regression, discriminant, and clustering models. The architecture of ANN can usually be represented as a three layer system, named input, hidden, and output layers. The input layer first processes the input features to the hidden layer. The hidden layer then calculates the adequate weights by using the activation function such as hyperbolic tangent, softmax, or logistic function before sending to the output layer.

Combining many computing neurons into a highly interconnected system, we can detect the complex non-linear relationship in the data. The simple three-layer perceptron, which is most used in fraud detection problems, can be depicted as shown in Fig. 1.

The procedures of the back-propagation training algorithm can be described as follows. First, let the weights and threshold levels are randomly drawn from a uniform distribution inside a range [22]:

$$\left[-\frac{2.4}{F_i}, +\frac{2.4}{F_i} \right], \quad (5)$$

where F_i denotes the total number of inputs of the i th neuron in the network. Then, first calculate the actual outputs of the neurons in the hidden layer by the equation:

$$y_j(n) = \varphi \left[\sum_{i=1}^p x_i(n) \times w_{ij}(n) - \theta_j \right], \quad (6)$$

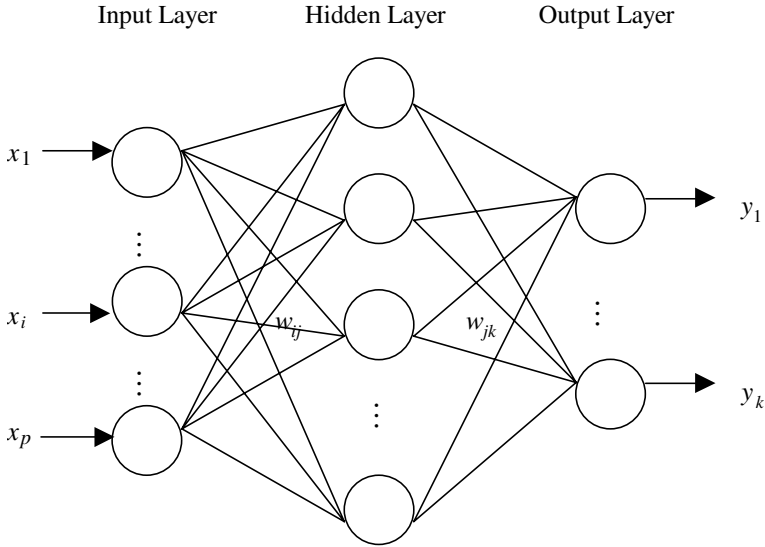


Fig. 1. Three-layer neural network.

where $\varphi(\cdot)$ denotes the activation function, x_i denotes the i th input, w_{ij} denotes the weight between the i th input and the j th hidden neuron, θ_j is the j th threshold, and p is the number of input of j th neuron in the hidden layer.

Second, calculate the actual output of the neurons in the output layer using the equation:

$$y_k(n) = \varphi \left[\sum_{j=1}^q x_{jk}(n) \times w_{jk}(n) - \theta_k \right], \quad (7)$$

where w_{jk} denotes the weight between the j th hidden layer and the k th output, and q is the number of inputs of the k th neuron in the output layer.

Next, update the weights in the output layer using the equation:

$$w_{jk}(n+1) = w_{jk}(n) + \eta \cdot \delta_k(n) \cdot y_j(n), \quad (8)$$

where η denotes the learning rate,

$$\delta_k(n) = y_k(n) \cdot [1 - y_k(n)] \cdot e_k(n), \quad (9)$$

$$e_k(n) = d_k(n) - y_k(n), \quad \text{where } d_k(n) \text{ denotes the actual output,} \quad (10)$$

and update the error gradient for the neurons in the hidden layer using the equation:

$$w_{ij}(n+1) = w_{ij}(n) + \eta \cdot \delta_j(n) \cdot x_i(n), \quad (11)$$

where

$$\delta_j(n) = y_j(n) \cdot [1 - y_j(n)] \cdot \left[\sum_{k=1}^r \delta_k(n) \cdot w_{ik}(n) \right]. \quad (12)$$

The above procedures will stop until the error criterion or the maximum iterations are satisfied.

Recently, ANN has been widely used in fraud detection problems, and it has been reported that its accuracy is superior to the traditional statistical methods such as discriminant analysis and logistic regression [3–7]. However, ANN has been criticized for its poor performance when existing irrelevant attributes or small data sets. Although many methods have been proposed to deal with the problem of variable selection [9,10], it is time waste and makes the model more complicated. In addition, other scholars are criticized the limitations of its long training process in designing the optimal network's topology in fraud detection problems [23,24]. In order to overcome the problem of ANNs and combine the advantage of induction-based algorithms, 2SGP is proposed in this paper. Next, we will first describe the concept of genetic programming (GP) in Section 3.

3. Genetic programming

Genetic programming was proposed by Koza [25] to automatically extract intangible relationships in a system and has been used in many applications such as symbolic regression [26], and classification [27,28]. 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 ($\{+, -, \times, \div\}$) or conditional statements (IF... THEN...) which are available in the GP. The terminal set contains all inputs, constants and other zero-argument in the GP tree. For example to express $xy + 3/x$, the GP tree can be represented in Fig. 2.

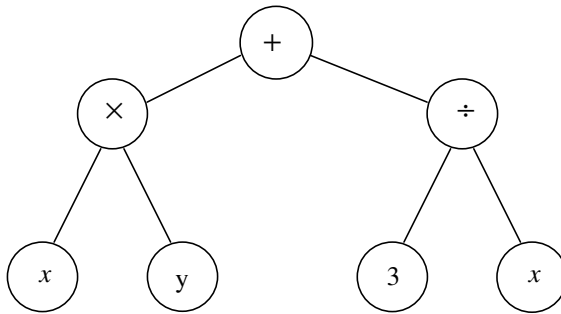


Fig. 2. The representation of a GP tree.

Once we initialize a population of the GP tree, the following procedures are similar to genetic algorithms (GAs) including defining the fitness function, genetic operators such as crossover, mutation and reproduction, and the termination criterion, etc. Next, we introduce three main operators, crossover, mutation and reproduction, to show the procedures of finding the (approximate) optimal generation.

In GP, the crossover operator is used to swap the subtree from the parents to reproduce the children using mating selection policy rather than exchanging bit strings as in GAs. An example of a crossover in GP is shown in Fig. 3.

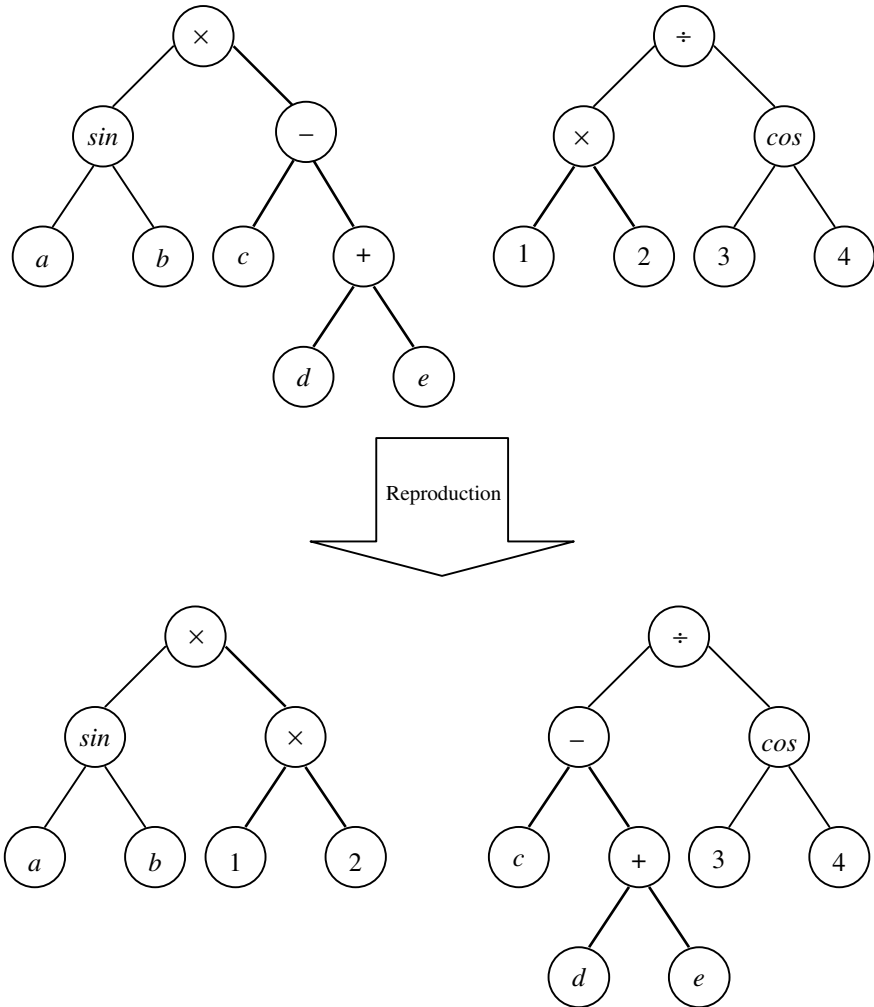


Fig. 3. The crossover operator of GP tree.

Similar to GAs, GP uses the mutation operator in order to avoid falling into the local optima. The mutation operator is used to randomly choose a node in a subtree and replace it with a new created subtree randomly. Finally, a new generation can be reproduced from two parents using the reproduction operator to represent a better solution.

It should be highlighted that the function set and the terminal set should be varied enough to represent the relationships among independent and response variables. Moreover, in order to satisfy the principle of parsimony, the depth of the GP-tree should also be limited. Next, we propose the procedures of the proposed method for the credit scoring problems using 2SGP.

4. Two-stage genetic programming (2SGP)

In this paper, the first-stage of GP is employed to derive the IF–THEN rules for the decision-maker. Although several papers have been proposed to discover the IF–THEN rules using GP [29–32], these papers cannot provide the concise and useful IF–THEN rules for the decision-maker [33]. The reason is that the IF–THEN rules, which are derived using GP, are usually too complex for the decision-maker to understand the meaning of the rules [33]. For example, let a decision-rule can be derived as

$$\text{IF} \left[\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \times (x_3 + x_4) \geq (x_5 - x_6) \times \begin{pmatrix} x_7 \\ x_8 \end{pmatrix} \right]$$

THEN creditworthy ELSE non-creditworthy,

where x_i denotes the i th attributes. It is hard for the decision-maker to interpret and understand the rule above.

In order to derive the interesting and useful IF–THEN rules for the decision-maker, a revised GP for discovering the IF–THEN rules are proposed. First, the maximum GP-tree depth of four is enforced to ensure for obtaining a simple GP-tree. In addition, we only employ the conditional operators (IF...THEN...), rational operators ($\geq, =, \leq$), and the Boolean operators (*and, or, not*) in this paper. Furthermore, in order to avoid producing an invalid child, only the compatible operators can be swapped.

Next, we can present the proposed method as follows. Instead of deriving the rules in the whole data set, we derive the rules as compact as possible for each class so that only a few rules are derived to represent the general trend of each class. For example, if the creditworthy customer can be represented using the GP-tree as shown in Fig. 4, then the rule can be interpreted as

$$\text{IF} (A_1 > 0.3 \text{ or } A_2 < 0.4) \text{ THEN customer} = \text{creditworthy.}$$

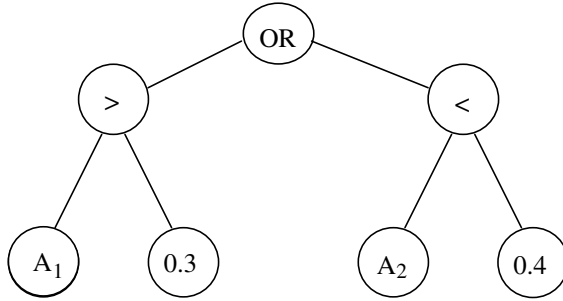


Fig. 4. The representation of the IF–THEN rule using the GP-tree.

Note that if the decision-maker wants to derive another rule in a class, the data which satisfy any rule should not be trained again. In addition, the fitness function of first-stage GP can be described as

$$ff_i = \frac{\sum_i^n \text{abs}(o_i - e_i)}{n}, \tag{13}$$

where $\text{abs}(\cdot)$ denotes the absolute operator, o_i denotes the observed class, e_i denotes the expected class, and n denotes the number of each class.

For the second stage of GP, the reduced data set are employed to build the discriminant function for providing the capability of forecasting. In this paper, the reduced data set are defined as the data which do not satisfy any rule or satisfy more than one rule. In addition, the labeled classes of the reduced data can be represented as

$$o_i = \begin{cases} +1 & \forall o_i \in \text{non-creditworthy}, \\ -1 & \forall o_i \in \text{creditworthy}, \end{cases} \tag{14}$$

where o_i denotes the status of customers. Next, the discriminant function can be built using GP and the forecasting outputs can be determined according to the following equation:

$$y(x) = \text{sign}(g(x)), \tag{15}$$

where $g(x)$ denotes the discriminant function and

$$\text{sign} = \begin{cases} +1 & \forall g(x) \geq 0, \\ -1 & \forall g(x) < 0. \end{cases} \tag{16}$$

In order to determine the appropriate discriminant function, the fitness function of GP can be described as

$$ff_j = \frac{\sum_j^m \text{abs}(o_j - e_j)}{m}, \tag{17}$$

where o_j denotes the observed reduced class, e_j denotes the expected reduced class, and m is the numbers of the reduced data.

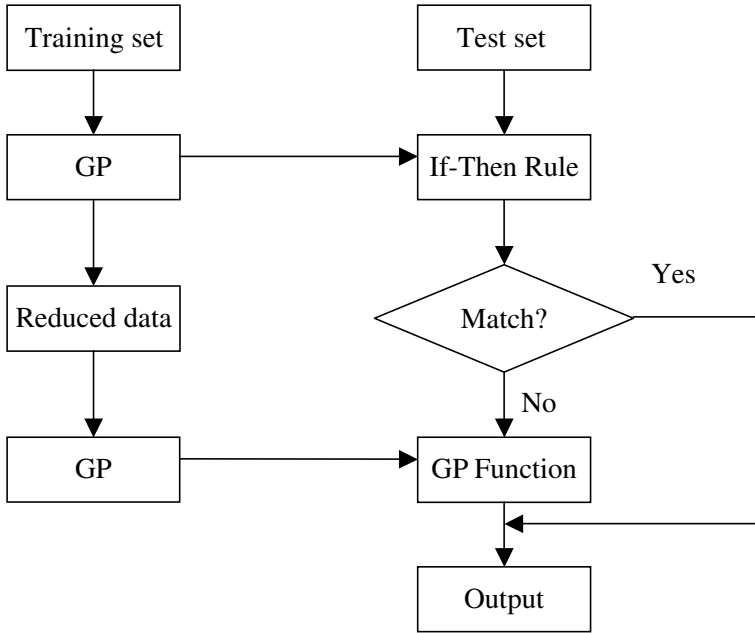


Fig. 5. The procedures of classifying newly entered data sets.

In summary the statements above, the whole procedures of 2SGP can be depicted as shown in Fig. 5.

In the next section, two numerical cases are use here to demonstrate the proposed method. In addition, several artificial intelligence and statistical approaches are also employed to compare with 2SGP.

5. Numerical analysis

In this section, 2SGP is compared with MLP, CART, C4.5, Rough sets, and logistic regression (LR) using the two real-world data sets. The first data set, called the German Credit Data Set, was provided by Prof. Hofmann in Hamburg. It includes customer credit scoring data with 20 features, such as age, gender, marital status, credit history records, job, account, loan purpose, other personal information, etc. There are 700 records judged to be creditworthy and 300 records judged to be non-creditworthy. The second data set includes Australian credit scoring data with 307 examples of the credit worthy customers and 383 examples for the non-creditworthy customers. It contains 14 attributes, where six are continuous attributes and eight are categorical attributes. Both data sets are made public from the UCI Repository of

Machine Learning Databases, and are mostly used to compare the accuracy with various classification models.

In this paper, the parameters used in 2SGP can be described as follows. The population size is 100, the maximum number of generations is 1000, the crossover rate is 0.9, and the mutation rate is 0.01. In addition, since GP is a heuristic tool, five iterations of the proposed method are used to consider the robust problem in the five different sampling data. Furthermore, the holdout method is employed for avoiding the problem of overfitting. In this numerical analysis, the scaling ratio of the train and the validation data sets is 7:3.

Next, we can derive the IF–THEN rules for each class using GP. Because the scaling of the Australian data set is relatively small, only one rule is derived using GP for each creditworthy and non-creditworthy class. In the first iteration, the IF–THEN rules, the accuracy, and the hit numbers can be presented as shown in Table 1.

Next, the reduced data are employed to form the discriminant function using the second stage of GP. The confusion matrix of the discriminant function can be described as shown in Table 2.

In order to compare the proposed method with other models, five different sampling samples are employed to test the error rate of the validation set. As shown in Table 3, the proposed method outperforms to other models.

With the same procedures above, the IF–THEN rules, the accuracy, the hit numbers, and the confusion matrix can be obtain as shown in Tables 4 and 5. Since the scaling of the German data is relatively large, we derive two IF–

Table 1
The decision rules of GP in the first iteration in the Australian data set

Australian	Antecedent	Consequent	Train (Accuracy/number)	Validation (Accuracy/number)
Rule for creditworthy	IF (A8 = 1 and A9 = 1), THEN	Good	90.3%/149	92.1%/58
Rule for non-creditworthy	IF (A8 = 0 and A3 > 0.165 and A5 ≤ 13), THEN	Bad	98.0%/194	93.6%/88

Table 2
The confusion matrix of the discriminant function in the Australian data set

Australaian	Train		Test	
	Class 1	Class 2	Class 1	Class 2
Class 1	50	16	17	7
Class 2	11	43	7	19

Table 3
The comparison of the credit scoring models in the Australian data set

Australian credit	Iteration 1 (%)	Iteration 2 (%)	Iteration 3 (%)	Iteration 4 (%)	Iteration 5 (%)	Overall
2SGP	0.1207	0.0973	0.1035	0.1207	0.0994	0.1083
GP	0.1111	0.1280	0.1304	0.1207	0.0966	0.1173
MLP	0.1352	0.1256	0.1352	0.1062	0.1014	0.1207
CART	0.1497	0.1256	0.1449	0.1400	0.1497	0.1419
C4.5	0.1594	0.1304	0.1400	0.1014	0.1159	0.1294
<i>k</i> -NN	0.3381	0.3381	0.2850	0.3236	0.2946	0.3158
LR	0.1497	0.1449	0.1304	0.1304	0.1352	0.1381

Table 4
The decision rules of the genetic programming in the German data set

German	Antecedent	Consequent	Train (Accuracy/number)	Validation (Accuracy/number)
Rule for creditworthy	1 IF (checking \geq 3), THEN	Good	86.12%/335	83.33%/145
	2 IF (duration < 22.5 and purpose = 3 or 8 or 9 or 10 and checking \leq 2), THEN	Good		
Rule for non-creditworthy	1 IF (checking \leq 2 and duration < 22.5 and purposed not (3 or 8 or 9 or 10) and history = 0 or 1), THEN	Bad	70.23%/92	67.39%/31
	2 IF (saving \leq 3 and duration > 22.5 and checking \leq 2 and purpose not (1 or 6)), THEN	Bad		

Table 5
The confusion matrix of the discriminant function in the German data set

Australian	Train		Test	
	Class 1	Class 2	Class 1	Class 2
Class 1	20	45	8	18
Class 2	4	111	4	50

Table 6

The comparison of the credit scoring models in the German data set

German credit	Iteration 1 (%)	Iteration 2 (%)	Iteration 3 (%)	Iteration 4 (%)	Iteration 5 (%)	Overall
2SGP	0.2200	0.1985	0.2057	0.1986	0.2028	0.2051
GP	0.2166	0.2266	0.2200	0.2433	0.2266	0.2266
MLP	0.2400	0.2382	0.2500	0.2433	0.2533	0.2449
CART	0.2765	0.2617	0.2435	0.3170	0.3721	0.2941
C4.5	0.2446	0.2500	0.2227	0.2926	0.3318	0.2683
<i>k</i> -NN	0.3049	0.3007	0.2849	0.3292	0.3587	0.3157
LR	0.2163	0.2421	0.1813	0.2479	0.2914	0.2358

THEN rules for each class. In addition, the comparison of the error rate of the five-validation sets can also be described as shown in Table 6.

On the basis of the results, we can conclude that the proposed method outperforms to other models in our empirical analysis. In addition, GP, ANN and logistic regression also perform well in this study and can be other alternatives for the credit scoring model. However, another advantage of the proposed method is the presentation of the intelligence rules for the decision-maker. Next, we provide the in-depth discussions based on our implementation.

6. Discussions

Due to the huge growth rate of the credit industry, building an effective credit scoring model have been an important task for saving amount cost and efficient decision making. Although many novel approaches have been proposed, more issues should be considered for increasing the accuracy of the credit scoring model.

First, the irrelevant variables will destroy the structure of the data and decreases the accuracy of the discriminant function. Second, the credit scoring model should determine the correct discriminant function (linear or non-linear) automatically. Third, the credit scoring model should be useful in both large and small data sets. Finally, the intelligence rules are very useful for the decision-maker to understanding the meaning of the data set. For above reasons, 2SGP is proposed to build the credit scoring models in this paper.

On this basis of the simulated results, we can conclude that 2SGP outperforms to other models. However, GP, ANN and logistic regression can also provide the satisfactory solutions and can be other alternatives. The accuracy of the induction-based approaches (decision trees and rough set) is inferior in this study. It is clear that the IF–THEN rules are derived from the training set. However, if a newly entered object within the test set does not match any rule, it cannot be determined to which class it belongs.

Compared with other models, it can be seen that 2SGP is more suitable for the credit scoring problems for the following reasons. Unlike the traditional statistical methods need the assumptions of the data set and the attributes, GP is a non-parametric tool and suitable for any situations and data sets. Compared with ANNs, 2SGP can determine the adequate discriminant function automatically rather than assigned the specific transfer function by decision-makers. In addition, 2SGP can also select the important variable automatically. Furthermore, the discriminant function which is derived by GP can provide the better forecasting accuracy than the induction-based algorithms. Finally, the decision-maker can obtain the intelligence rules.

7. Conclusions

Building a credit scoring model involves the problems of assigning a record into a correct class. Although many approaches have been proposed, a flexible and accurate method is limited. In this paper, 2SGP is proposed to incorporate the IF–THEN rules and the discriminant function for the credit scoring problems. On the basis of the empirical results, we can conclude that the proposed method is more flexible and significantly performs better accuracy in the credit scoring problems.

References

- [1] T.S. Lee, C.C. Chiu, C.J. Lu, I.F. Chen, Credit scoring using the hybrid neural discriminant technique, *Expert Systems with Applications* 23 (3) (2002) 245–254.
- [2] D. West, Neural network credit scoring models, *Computers & Operations Research* 27 (11–12) (2000) 1131–1152.
- [3] V. Desai, J. Crook, G. Overstreet, A comparison of neural networks and linear scoring models in credit union environment, *European Journal of Operations Management* 95 (1) (1996) 24–37.
- [4] R. Mahlotra, D.K. Malhotra, Evaluating consumer loans using neural networks, *OMEGA: The International Journal of Management Science* 31 (2) (2003) 83–96.
- [5] S. Piramuthu, Financial credit-risk evaluation with neural and neurofuzzy systems, *European Journal of Operational Research* 112 (2) (1999) 310–321.
- [6] H.L. Jensen, Using neural networks for credit scoring, *Managerial Finance* 18 (1) (1992) 15–26.
- [7] V. Desai, J. Crook, G. Overstreet, Credit scoring models in the credit union environment using neural networks and genetic algorithms, *IMA Journal of Mathematics Applied in Business and Industry* 8 (4) (1997) 324–346.
- [8] F. Castillo, K. Marshall, J. Green, A. Kordon, A methodology for combining symbolic regression and design of experiments to improve empirical model building, *Genetic and Evolutionary Computation Conference (2003)* 1975–1985.
- [9] R. Feraud, F. Cleror, A methodology to explain neural network classification, *Neural Network* 15 (2) (2002) 237–246.

- [10] R. Nath, B. Rajagopalan, R. Ryker, Determining the saliency of input variables in neural network classifiers, *Computers and Operations Research* 24 (8) (1997) 767–773.
- [11] A.I. Dimitras, R. Slowinski, R. Susmaga, C. Zopounidis, Business failure prediction using rough sets, *European Journal of Operational Research* 144 (2) (1999) 263–280.
- [12] M.J. Beynon, M.J. Peel, Variable precision rough set theory and data discretisation an application to corporate failure prediction, *OMEGA: The International Journal of Management Science* 29 (6) (2001) 561–576.
- [13] Y. Bi, T. Anderson, S. McClean, A rough set model with ontologies for discovering maximal association rules in document collections, *Knowledge-Based Systems* 16 (5–6) (2003) 243–251.
- [14] B.S. Ahn, S.S. Cho, C.Y. Kim, The integrated methodology of rough set theory and artificial neural network for business failure prediction, *Expert Systems with Applications* 18 (2) (2000) 65–74.
- [15] S.J. Press, S. Wilson, Choosing between logistic regression and discriminant analysis, *Journal of the American Statistical Association* 73 (4) (1978) 699–705.
- [16] A. Agresti, *Categorical Data Analysis*, Wiley, New York, 1990.
- [17] J.H. Aldrich, F.D. Nelson, *Linear Probability, Logit, and Probit Models*, Sage, California, 1984.
- [18] A. DeMaris, *Logit Modeling*, Sage, California, 1992.
- [19] D. Knoke, P.J. Burke, *Log-linear Models*, Sage, California, 1980.
- [20] T.F. Liao, *Interpreting Probability Model: Logit, Probit, and Other Generalized Linear Models*, Sage, California, 1994.
- [21] P. McCullagh, Regression model for ordinal data, *Journal of the Royal Statistical Society, Series B* 42 (2) (1980) 109–142.
- [22] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing Company, New York, 1994.
- [23] H.M. Chung, P. Gray, Special section: data mining, *Journal of Management Information Systems* 16 (1) (1999) 11–16.
- [24] M.W. Craven, J.W. Shavlik, Using neural networks for data mining, *Future Generation Computer Systems* 13 (2/3) (1997) 221–229.
- [25] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, MA, 1992.
- [26] J.W. Davidson, D.A. Savic, G.A. Walters, Symbolic and numerical regression: experiments and applications, *Information Sciences* 150 (1–2) (2003) 95–117.
- [27] Y. Zhang, S. Bhattacharyya, Genetic programming in classifying large-scale data: an ensemble method, *Information Science* 163 (1–3) (2004) 85–101.
- [28] C.D. Stefano, A.D. Cioppa, A. Marcelli, Character preclassification based on genetic programming, *Pattern Recognition Letters* 23 (12) (2002) 1439–1448.
- [29] H.E. Johnson, R.J. Gilbert, M.K. Winson, R. Goodacre, A.R. Smith, J.J. Rowland, M.A. Hall, D.B. Kell, Explanatory analysis of the metabolome using genetic programming of simple, interpretable rules, *Genetic Programming and Evolvable Machines* 1 (3) (2000) 243–258.
- [30] P.S. Ngan, M.L. Wong, K.S. Leung, Using grammar based genetic programming for data mining of medical knowledge, in: *Proceedings of the Third Annual Conference on Genetic Programming*, Morgan Kaufmann, San Francisco, 1998, pp. 304–312.
- [31] C.C. Bojarczuk, H.S. Lopes, A.A. Freitas, E.L. Michalkiewicz, A constrained-syntax genetic programming system for discovering classification rules: application to medical data sets, *Artificial Intelligence in Medicine* 30 (1) (2004) 21–48.
- [32] 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, 1997, pp. 96–101.
- [33] I. De Falco, A.D. Cioppa, E. Tarantino, Discovering interesting classification rules with genetic programming, *Applied Soft Computing* 1 (4) (2002) 257–269.