A novel Intelligent Credit Scoring Method using MOPSO

Yan Guo l , Chao Dong2

I. Ningbo Institute of Technology, Zhejiang University, Ningbo 315100, China

E-mail: guoyanbox@126.com

2. Ningbo Dahongying University, Ningbo 315175, China E-mail: 2821625@sina.com

Abstract: We present an intelligent credit scoring method to categorize credit applicants. Then, a novel multi-objective credit scoring model is proposed in this paper. In term of the defects of linear discriminant analysis (LDA): lack of accuracy, a multi-objective particle swarm optimization for credit scoring is designed in this paper. Finally, through the experiments with two real-world data set and one benchmark data set, we compare our approach with NaiveBayes, Logistic Regression (LR), Sequential Minimal Optimization (SMO), Neural Networks (NN), and Decision Trees (DT), the results of experiments demonstrate our proposed method outperforms the abovementioned five data-driven counterparts in term of accuracy and specificity while maintaining acceptable sensitivity. Key Words: Credit scoring, Data classification, Particle swarm optimization

# 1. Introduction

Credit risk evaluation decision is a crucial issue for banking industry because even a one percent improvement in early detection of bad credit account may avoid huge amount of losses [1, 2]. Credit scoring model is the most successful method that helps financial institution to decide whether to grant or refuse a loan [3].

The popular methods used in building credit scoring models include Linear Discriminant Analysis (LDA) [2, 4], Logistic Regression (LR) [5, 6], Neural Networks (NN) [7,8], Support vector machines (SVM) [9,10] , Decision Tree (DT) [l l, 12] and Evolutionary computation techniques [13, 14].

We present a novel credit score method using LDA, and discriminate "good credit" group and "bad credit" group through comparing credit score with cutoff. Compared with NN and SVM, our credit scoring approach is more intelligent and easy to be implemented.

## 2. Mathematic model

Linear discriminant analysis (LDA) classifies two or more group using a set of independent variables. [15] LDA classifies credit applicants based on their discriminant score, which is calculated by a discriminant function. Scorq =A.X = ailXl +anX2 +...+aiRxR where Scorq is the credit score of record i, R is the number of attribution, X = (x x is the weight set of attribution, and f! = (a il' a a iR) is the value set of record i.

For two-group classification, if Scorq > b , record i belongs to class 0 (Negative); and if Scorq < b , record i belongs to class I (Positive), where b is a boundary (cutoff), i = I,...,N , and N is the sample size. The aim of

|  |
| --- |
| This work is supported by the Zhejiang Provincial Education  Department project (Y201636906) and Ningbo innovative team project |

credit scoring based on LDA is to determine the best coefficients of the variables, denoted by X = (x and value b (a scalar) to separate two classes: Good (Class O)and Bad (Class l). For credit scoring practice, Bad means a group of "bad credit" customers, Good means a group of "good credit" customers.

Traditional linear programming models of classification are usually based on distance, in these models, there are two kinds of objectives, one is to minimize the internal distance and the other is to maximize the external distance. But the solution with minimum internal distance or maximum external distance is not able to be proved to classify all samples correctly. In this paper, we insert an objective that is to minimize the number of misclassification to LP model of classification.

Then the credit scoring problem can be described as how to set the confident set X to achieve the following objectives:

(l) Minimize the number of misclassification

### Minimize MCN = E [WGdG (i) -FWBdB (i)] (1)

i=l where MCN is a coefficient that measure the quality of the classification for weight set X = (x dB(i) are the 0-1 variable that dG(i) = 0 if is a Bad record or J. is a Good record classified correctly and dG(i) = I if 4. is a Good record classified as Bad; dB(i) = 0 if J. is a Good record or 4. is a Bad record classified correctly and dB (i) = 1 if Ai is Bad record classified as Good.

In traditional data classification problem, accuracy is the main objective to be optimized. But to credit card customer classification, sensitivity is more important than other metrics. In this paper, we locate the trade-off between specificity and sensitivity through setting the weight WG,WB e (0, l) WG is the weight of

|  |
| --- |
| (2016C11024).  978-1-5090-4657-7/17/$31.00 @2017 IEEE 6717 |

misclassification number in Good records. Then bigger weight WG will result in bigger specificity. Similarly, is the weight of misclassification number in Bad records, and bigger weight WB will result in bigger sensitivity.

Obviously, MCN is equal to half of the number of the records misclassified if WG = WG = 0.5 , and MCN=O if all records are classified correctly.

1. Minimize the sum of deviations ofthe observations Minimize MSD = (2)

where MSD is the sum of the deviations of the observations.

1. Maximizing the minimal distances of observations from the critical value

Maximize MAIL) — (3)

where MMD is the minimal distances of observations from the critical value.

A graphical representation of , A is shown in Fig. 1.

## AIX=b

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  | a |  |
|
|  |

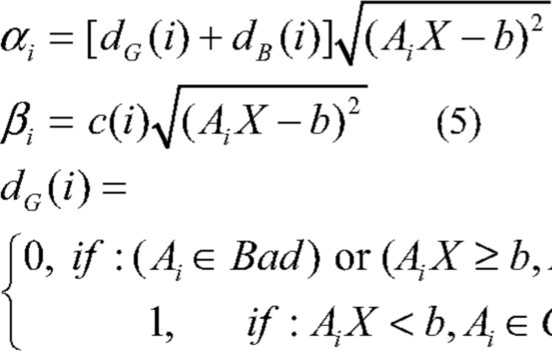
 Bad Good

Figure 1. Graphical representation of a

Based on the above analysis, the constraints of this model are listed as follows:

(4)

e Good) (6)

Good

|  |  |
| --- | --- |
| dB (i) — |  |
| 0, if : (Ai e Good) or (4X < b, 4 e Bad) | (7) |

O, if: (4X > Bad) or (4.X < b, Ai e Good) c(i) =(8) 1, Bad) or (4.X > b, 4. e Good) dG (i) + dB (i) + , Vie {l, 2,...,N} (9)

 (10)

where 14, b are given, X is unrestricted and is the 0-1 variable which c(i) = 1 if case Ai is classified correctly and c(i) = 0 if case 4. is misclassified.

The primary objective of classification problem is to classify all samples correctly, and then we define the solution which classifies all samples correctly as optimal solution.

Define 1 (Optimal solution): the solution xJ which classifies all records correctly.

 (11)

where P is the set of feasible solutions, and Q is the set of all solutions. According to (2) and (4), MSD of all feasible solutions is equal to 0.

Usually, it is not able to get an optimal solution if the data set is linear inseparable. Then the objective of classification problem is to find the feasible solution which MCN is the minimum.

Define 2 (Feasible solution): the solution with the minimum misclassification coefficient. s  e Q}, X/ Q} (12)

where S is the set of feasible solutions. Obviously, all optimal solutions are feasible solutions.

1. Multi-objective particle swarm optimization for credit scoring

According to three objectives for credit scoring problem, multi-objective particle swarm optimization (MOPSO) is used to deal with this problem. Multiple subswarms and co-evolutionary approach is able to improve the efficiency and effectiveness of MOPSO [16], in this paper a co-evolutionary MOPSO is adapted for Credit Scoring (MOPSO-CS). MOPSO-CS employs two subswarms ( Subswarntl and Subswar1112 ) with the same population J to probe the search space and information is exchanged between them.

3.1 Best position selection and updating process



value of MSD is minimum be the personal best position ofSubswarn11 Pbestl , and let the position of the particle in Subswar1112 whose value of MNID is maximum be the personal best position ofSubswarn12 Pbest2 .

In order to search optimal solution as soon as possible, we select two feasible solutions from set S, and let the position of the feasible solution with minimum MSD be the global best position of Subswarml Gbestl , and let the position of the feasible solution with maximum MNID be the global best position of Subswarn12 Gbest2 . Through this method, each swarm can share information from feasible solutions, so that MOPSO-CS is able to guide the particles to optimal solution as soon as possible.

|  |
| --- |
| 6718 2017 29th Chinese Control And Decision Conference (CCDC) |

MOPSO-CS is similar to co-evolutionary MOPSO and, in each generation t, particlej in the kth swarm updates its current position xo.r (t) and velocity (t) through each dimension r by the personal best position Pbest .r and



the global best position GbestkT using equation (14) and

(15):

Vor (t + I) = WVkp. (t) +

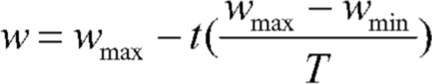
(14)

[Pbest&r — xor + —

 (15)

where w is inertia weight, q,C2 are the acceleration constants and h, h are random real numbers drawn from U(O,I) .

In this paper, the following weight function is used:

 (16)

where w : Initial weight, : Final weight.

T: Maximum generation number, t: Current generation number.

Because there are perhaps many particles with the same particle rank, MSD and MMD are also taken into consideration when all particles are sorted. The detailed criterion of rank for each particle is as follows:

Particle a is superior to particle b:

 {rank(a) < rank(b)}

Or {rank(a) = rank(b) and MSD(a) < MSD(b)}

3.2 The optimization algorithm

For MOPSO-CS, the scope of the solution is equal to R-dimensional search space where R is the total number of attributes of each sample (customer). The position of particlej xo. = kjl ' • • x kjR) corresponds to a solution for the problem, and the rth dimension of the position xor (k = 1, 2; j = 1,...,J;r = denotes the rth coefficient used by each sample. The coding design of particle velocity is similar to the design of particle position, which is composed of a component controlled set of coefficient: v ( kjl ' kjr

Firstly the algorithm randomly generates two subswarms. Then the particle position and the particle velocity of all particles in these subswarms are initialized. Each element xo.r of xo is randomly initialized within [-100, 100]. Initialize each element vor of particle velocity within [-10, 10].

Secondly evaluate these subswarms. According to xo which has been initialized, calculate MCN, MSD and

MAID for each particle using objective (1), (2) and (3). Then, evaluate the particle rank of each particle using Eqt. (13). The algorithm initializes the global best position of each subswarm and the personal best position of each particle.

In this paper, MOPSO-CS tries to find optimal solutions of this problem and terminates when the first optimal solution is found or the maximum iteration number T is reach. Before the algorithm finds optimal solution, all feasible solutions are used as the best found solutions. At each iteration of algorithm, in order to optimize the Objective (1), MOPSO-CS updates the best found solutions in term of MCN and reserves the solutions with minimum MCN found by optimization process.

Thirdly update the position and velocity of each particle according to Eqt. (14) and (15). Then, we select top 2J particles with minimum particle rank from old population and updated population, and use them as the new population for the next generation. The optimization algorithm can be formally described as follows:

Pseudocode of MOPSO-CS:

Stepl. Start procedure MOPSO-CS. Initialize the position of each particle of subswarml and subswarm2 Define MPG , .n,w and Cutoff

Step2. Evaluate Swarm

For each Subswarmk do

For to 2

For each particlej do

Forj=l to J

Compute MSD(j) and MMD(j)

Calculate rank(j)

Update PbestkJ

Nextj

End do

Update Gbestk

Nest k

End do

Step 3: Update the best found solution

Step 4: Check the stopping condition. If it is not met go to Step 5. Otherwise go to Step 7.

Step 5: Update the new velocities and positions for each particlej

Step 6: Select new population for the next generation.

Go to Step 2

Step 7: Finish the procedure MOPSO-CS

## 4. Computational experiments

4.1. The selection and preprocessing of credit data set

|  |
| --- |
| 2017 29th Chinese Control And Decision Conference (CCDC) 6719 |

In computational experiments, we use various data sets that represent the credit behaviors of people from different countries which include one real-world data set from UK and one benchmark German data set. All these data sets classify people as "Normal" or "Bankrupt" credit risks based on a set of attributes/variables. In order to investigate the effectiveness of the proposed model and algorithm in unbalance data set and balance data set, the real-world credit data from UK is separated to two date set. The first is UK-I data set [2], which collects 323 bankrupt and 902 normal customers with 14 variables. The second is balance data set UK-2, which consist of 323 normal customers and 323 bankrupt customers. The third data set comprises German Credit card records from UCI Machine Learning databases [10], which contains 1000 records (700 normal and 300 bankrupt).

In order to improve the accuracy of credit data classification, we have preprocessed data in credit records before classification. The data preprocessing includes the conversion of unstructured data into structured data and deletion of the attributions whose correlation with other attributions are higher (the correlation coefficient r.. < —0.5 or U. > 0.5 ). Table 1 shows the selection and preprocessing of credit data set.

Tablel. The selection and preprocessing of credit data set

|  |  |  |  |
| --- | --- | --- | --- |
| Data set  UK-I | Record No.  1225 | Attribution No.  of raw data  14 | The attributions that have been deleted |
| UK-2 |  | 14 |  |
| German | 1000 | 24 |  |

Sensitivity = TP/(TP+FN)

Specificity = TN/(FP+TN)

Table 4 shows the result of ten-fold cross-validation of UK-I Data Set (902 normal customers, 323 bankrupt customers). We can see from table 4, the sensitivity of our MOPSO-CS excel those of LR, SMO and NN, and only below the sensitivity of NaiveBayes and DT, but the accuracy of MOPSO-CS is superior to those of all the other counterparts.

Table 4. Ten-Fold Cross-Validation Result of UK- 1

4.2. Experimental evaluation

In this section, our proposed methods, MOPSO-CS, is compared with five classic classification methods: NaiveBayes, Logistics Regression (LR), SMO, Neural Networks (NN), and Decision Tree (DT). The implementation software of these counterparts is WEKA 3.6, and all experiments were performed on a PC Core i5 with 2.5GHz and 4GB RAM running under the Windows 10 operating system. Table 2 shows the implementation software that was explored to run the three data sets.

Table2. Employed Software

|  |  |
| --- | --- |
| Classifier | Software |
| MOPSO-CS | Visual Studio 6.0 |
| NaiveBayes | WEKA3.6 Bayes (NaiveBayes) |
|  | WEKA3.6 Functions (Logistics) |
| SMO | WEKA3.6 Functions SMO |
|  | WEKA3.6 Functions  (MultilayerPerceptron) |
|  | WEKA3.6 Trees(J48) |

The parameters used in MOPSO-CS are presented in Table 3. The detailed results of the experiments show as follows.

Table 3. The parameters used in MOPSO-CS

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Data set |  |  | T | Population | Cutoff |
| UK-I | 1/2 | 1/2 | 1000 | 40 | -1.1 |
| UK-2 | 1/2 | 1/2 | 1000 | 40 | 80 |
| German | 1/2 | 1/2 | 1000 | 40 | -1.1 |

There are several measures of classification performance commonly used in the credit industry. [17]

True Positives (TP): the number of actual positive records which predicts positive.

False Negatives (FN): the number of actual positive records which predicts negative.

True Negatives (TN): the number of actual negative records which predicts negative.

False Positives (FP): the number of actual negative records which predicts positive.

Accuracy

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accurac | S ecifici | Sensitivi |
| MOPSO-CS | 74.64% | 96.81% | 12.72% |
| NaiveBayes | 65.22% | 73.50% | 42.10% |
|  | 73.80% | 96.60% | 10.20% |
| SMO | 73.63% | 100.00% | 0.00% |
|  | 72.57% | 94.50% | 11.50% |
|  | 74.29% | 94.80% | 17.00% |

Table 5 shows the result of ten-fold cross-validation of balance UK-2 Data Set (323 normal customers, 323 bankrupt customers).

Table 5. Ten-Fold Cross-Validation Result of UK-2

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy | Specificity | Sensitivity |
| MOPSO-CS | 59.85% | 81.67% | 38.02% |
| NaiveBayes | 58.20% | 42.10% | 74.30% |
|  | 60.99% | 63.20% | 58.80% |
| SMO | 58.51% | 60.70% | 56.30% |
|  | 61.15% | 59.10% | 63.20% |
|  | 61.14% | 57.30% | 65.00% |

From table 5, we can see that the accuracy of our MOPSO-CS excel those of NaiveBayes and SMO, but the Specificity of MOPSO-CS is superior to those of all the other counterparts.

Table 6 shows the result of ten-fold cross-validation of balance German Data Set (700 normal customers, 300 bankrupt customers). The specificity of MOPSO-CS excel those of all other counterparts, and the accuracy of MOPSO-CS excel those ofNN, Rules and DT.

Table 6. Ten-Fold Cross-Validation Result of German

|  |  |  |  |
| --- | --- | --- | --- |
| Classifier | Accuracy | Specificity | Sensitivity |
| MOPSO-CS | 74.69% | 89.87% | 39.27% |
| NaiveBayes | 75.10% | 84.90% | 52.30% |
|  | 76.70% | 89.10% | 47.70% |
| SMO | 76.40% | 89.70% | 45.30% |
|  | 70.30% | 80.00% | 47.70% |
|  | 72.70% | 83.90% | 46.70% |

## 5. Conclusion

Traditional credit scoring model are usually based on distance, in these models there are two kinds of objectives, one is to minimize the internal distance and the other is to maximize the external distance. But the solution with minimum internal distance or maximum external distance is not able to be proved to classify the samples correctly. In

6720 2017 29th Chinese Control And Decision Conference (CCDC)

|  |  |  |  |
| --- | --- | --- | --- |
| this paper, a novel multi-objective credit scoring model is | | [7] | P Håjek, Municipal credit rating modelling by neural |
| proposed, minimum the number of misclassification, then | |  | networks. Decision Support Systems, Vol. 5 1, No. 1, |
| minimize the internal distance and maximize the external | |  | 108-118, 2011. |
| distance are all included in the objectives of our model. | | [8] | D West, Neural network credit scoring models. |
| To credit scoring problem, it is important for credit | |  | Computers and Operations Research, Vol. 27, No. |
| scoring model to provide the credit score which is easy to | |  | 11, 1131-1152, 2000. |
| be comprehended by credit applicants. Compared with | | [9] | T Bellotti, J Crook, Support vector machines for |
| black box technology such as NN and SVM, the credit | |  | credit scoring and discovery of significant features. |
| score function in our approach is comprehensible. Finally, | |  | Expert Systems with Applications, Vol. 36, No. 2, |
| our example and experimental studies based on benchmark | |  | 3302-3308, 2009. |
| data set and real world data sets, confirm that our proposed | | [10] | D Niklis, M Doumpos, C Zopounidis, Combining |
| method outperforms the abovementioned five data-driven | |  | market and accounting-based models for credit |
| counterparts in term of accuracy while maintaining | |  | scoring using a classification scheme based on |
| acceptable sensitivity. | |  | support vector machines. Applied Mathematics and Computation, Vol. 234, No. 15, 69-81, 2014. |
| References | | [11] | D.L Olson, D Delen, Y Meng, Comparative analysis |
|  | T.S Lee, I.F Chen, A two-stage hybrid credit scoring |  | of data mining methods for bankruptcy prediction. |
|  | model using artificial neural networks and |  | Decision Support Systems, Vol. 52, No. 2, 464—473, |
|  | multivariate adaptive regression splines. Expert |  | 2012. |
|  | Systems with Applications, Vol 28, 743—752, 2005. | [12] | G Wang, J Ma, L Huang, K Xu, Two credit scoring |
| [2] | J He, Y.C Zhang , Y Shi, G.Y Huang, |  | models based on dual strategy ensemble trees. |
|  | Domain-driven classification based on multiple |  | Knowledge-Based Systems, Vol. 26, 61—68, 2012. |
|  | criteria and multiple constraint-level programming | [13] | R Aliehyaei, S Khan. Ant Colony Optimization, |
|  | for intelligent credit scoring. IEEE transactions on |  | Genetic Programming and a hybrid approach for |
|  | knowledge and data engineering, Vol. 22, No. 6, |  | credit scoring: A comparative study. Software, |
|  | 826-838, 2010. |  | Knowledge, Information Management and |
| [3] | B. W Chi, C.C Hsu, A hybrid approach to integrate |  | Applications (SKIMA), 2014 8th International |
|  | genetic algorithm into dual scoring model in |  | Conference, 1-5, 2014. |
|  | enhancing the performance of credit scoring model. | [14] | J.J Huang, G.H Tzeng, C.S Ong, Two-stage genetic |
|  | Expert Systems with Applications, Vol. 39, No. 3, |  | programming (2SGP) for the credit scoring model. |
|  | 2650-2661, 2012. |  | Applied Mathematics and Computation, Vol. 174, |
| [4] | H Jo, I Han, H Lee, Bankruptcy prediction using |  | No. 2, 1039-1053, 2006. |
|  | case-based reasoning, neural networks, and | [15] | S Lessmann, B Baesens, H.V Seow, Benchmarking |
|  | discriminant analysis. Expert Systems with |  | state-of-the-art classification algorithms for credit |
|  | Applications, vol. 13, No. 2, 97-108, 1997. |  | scoring: An update of research. European Journal of |
| [5] | D. J Hand, W. E Henley, Statistical Classification |  | Operational Research, Vol. 247, No. l, 124—136, |
|  | Methods in Consumer Credit Scoring: A Review. |  | 2015. |
|  | Journal of the Royal Statistical Society. Series A, | [16] | Goh, c.K, Tan, K.c., Liu, D.S., Chiam, S.C. A |
|  | vol. 160, No. 3, 523-541, 1997. |  | competitive and cooperative co-evolutionary |
| [6] | A.C Bahnsen, D Aouada, Example-Dependent |  | approach to multi-objective particle swarm |
|  | Cost-Sensitive Logistic Regression for Credit |  | optimization algorithm design. European Journal of |
|  | Scoring. Machine Learning and Applications |  | Operational Research, Vol. 202, 42-54, 2010. |
|  | (ICMLA), 2014 13th International Conference, 263 | [17] | S Bhattacharyya, S Jha, K Tharakunnel. Data mining |
|  | - 269, 2014. |  | for credit card fraud: A comparative study. Decision Support Systems, Vol. 50, No. 3, 602—613, 2011. |

2017 29th Chinese Control And Decision Conference (CCDC) 6721