A Hybrid Model of AdaBoost and Back-Propagation Neural Network for Credit Scoring

Feng Shen[1(✉)], Xingchao Zhao[1], Dao Lan[2], and Limei Ou[2]

[1] School of Finance, Southwestern University of Finance and Economics, Chengdu 611130, People's Republic of China
shenfeng1213@gmail.com

[2] Southwestern University of Finance and Economics, Chengdu 611130, People's Republic of China

Abstract. Owing to the development of internet finance in China, credit scoring is growing into one of the most important issues in the field of financial risk management. Quantitative credit scoring models are widely used tools for credit risk assessment in financial institutions. In this study, an AdaBoost algorithm model based on back-propagation neural network for credit scoring with high accuracy and efficiency is proposed. We first illustrate the basic concepts of back-propagation neural network and AdaBoost algorithm and propose a hybrid model of AdaBoost and back-propagation neural network, then two real-world credit data sets are selected to demonstrate the effectiveness and feasibility of the proposed model. The results show that the proposed model can get higher accuracy compared to other classifiers listed in this study.

Keywords: Credit scoring · AdaBoost model · Back-propagation neural network

1 Introduction

Credit scoring has grown an increasingly important issue of financial risk management in financial institutions since the 2008 financial crisis. It often calculates by following a set of decision models and other underlying technologies, does a great favor for lenders' judging whether an application of credit should be approved or rejected [27]. When some applicants fail to repay their debt, it leads to a direct economic loss for the lenders. In addition, the sub-prime mortgage crisis occurred in the USA has caused some financial institutions loss billions of dollars due to customers' default. However, if a credit-granting-institution rejects all applicants of loans even with good credit scores, it will suffer the potential revenues it can earn from the applicants in the future. Therefore, an efficient decision support with high accuracy becomes a clear need for financial institution.

J. Xu et al. (eds.), *Proceedings of the Eleventh International Conference on Management Science and Engineering Management*, Lecture Notes on Multidisciplinary Industrial Engineering, DOI 10.1007/978-3-319-59280-0_6

Due to the great importance of credit risk assessment, an increasing research stream is focusing upon credit risk assessment and credit scoring using different methods and models, approaches such as linear discriminant analysis (LDA) [1], logistic regression analysis LRA [26], k-nearest neighbor (KNN) [12,13] and decision tree [5] which uses the statistics disciplines. With the development of artificial intelligence(AI) techniques, artificial neural network (ANN) [3,8,14,21, 30,31], support vector machines (SVM) [2,11,25,34], genetic algorithm (GA) [7,22,28], rough set [4] and case-based reasoning [15–19] are used for credit scoring. Also some combined and ensemble approaches perform quite well with high accuracy and efficiency, including fuzzy system and artificial neural network [20,23], rough set and support vector machine [33], fuzzy system and support vector machines [29], case-based reasoning and support vector machines [19], neural network ensemble [32] etc.

Inspired by the combined and ensemble theories, this study attempts to purpose an ensemble AdaBoost model based on BP neural network for credit scoring. The main idea of AdaBoost algorithm is to maintain a distribution of weights over the training samples and adjust them after each basic classifier sorting cycle adaptively. In our study, we use ten single back propagation neural networks as the weak learners contends of a three-layer feed-forward network each, and the final strong ensemble classifier is constructed by the method of weighted voting associated with the performances of the weak learners. Based on the experimental results achieved from two public available credit data sets, our proposed AdaBoost model based on BP neural network obtains a good performance with higher accuracy efficiency compared to other models used in this study, which indicates a wide prospect usage in financial risk management.

The rest of this paper is organized as follows. Section 2 gives a brief formulation concerns about back propagation neural network and AdaBoost algorithm. Section 3 presents a hybrid model of AdaBoost and BP neural network. To verify the accuracy and effectiveness of the purposed model, empirical validation results of the model using the German credit data set and Australian credit data set are analyzed in Sect. 4. Finally, a short conclusion and discussion are presented in Sect. 5.

2 Methodology Formulation

2.1 Back-Propagation Neural Network Theory

Back-propagation neural network model was proposed by Rumelhart and Mccelland in 1985 [24], it is a kind of multi-layered forward feed type error counterbiography neural network. This model usually consists of input layer, output layer and hidden layer. Each layer includes certain nodes, and each node expresses a neuron. There exists interconnection among the nodes of different layers, but there is no connection among nodes in the same layer. Among them, the single hidden layer BP network's application is the most common. The typical structure of BP neural network is as shown in Fig. 1.

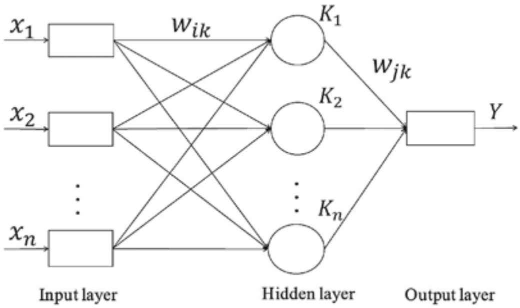

Fig. 1. Structure of single hidden layer BP neural network

The core thought of BP neural network can be described as following: training signal forward-propagating and error signal reverse dissemination. During the process of forward-propagating, the signal spreads from input layer, dealt with connection weights between input layer and hidden layer, then is transformed by the active function of hidden layer and transmitted to the output layer. If expected output can't be achieved from the output layer, then another process of error back propagation begins, which would continuously adjust the weights and bias of each layer in order to ultimately minimize the error of the network. This kind of signal forward-propagating and error back-propagating is to go iteration and iteration till the system output error is reduced to an acceptance degree or obtains the network constraints. Its concrete process is shown in Fig. 2.

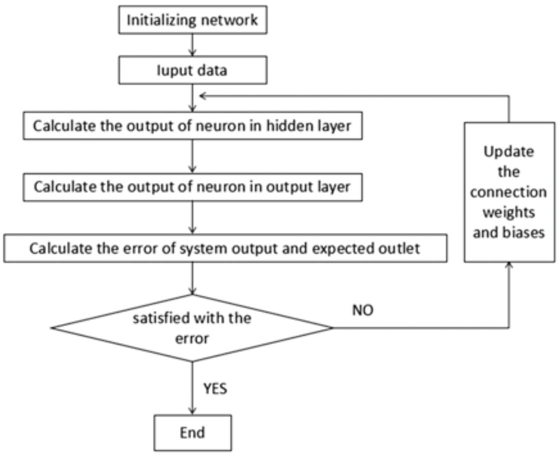

Fig. 2. Process of BP neural network model

2.2 Basic Theory of AdaBoost Algorithm

AdaBoost is a very popular boosting algorithm for binary classification, and it solved many practical difficulties of the earlier boosting algorithms. It was initially proposed by Yoav Freund and Robert Schapire in 1997 [10]. This algorithm constructs a high-quality compositive classifier by combining trained weak classifiers sequentially while putting more attention on those weak classifiers accompanied with good performance. There are several methods for combining results from base weak learners into one stronger predictor. Uniform voting, distribution summation, Bayesian combination, etc. are widely used for ensemble algorithms. Adaptive boosting algorithm has been applied in many fields such as speech recognition [9], moving vehicle classification based on images [6] and etc.

The main ideas of AdaBoost algorithm is to maintain a distribution of weights over the training samples and adjust them after each basic classifier sorting cycle adaptively. The weights of training data which are wrongly classified by current weak learner will be increased, otherwise, decreased if the samples are correctly classified. During the training process, the prediction error of the weak classifiers should be less than 0.5. And the voting weights of base classifiers will be increased while the decreasing of the prediction error, which means a larger voting weight this weak learner will take in the final ensemble output. Pseudocode for AdaBoost algorithm is shown in Fig. 3. As a series of base learners have been achieved, AdaBoost calculates a value α_t that is assigned to h_t, and the final hypothesis H is constructed via T weak classifiers using a weighted voting ensemble method.

AdaBoost Algorithm:

Given a training data set: $T = \{(x_1,y_1),\dots(x_i,y_i),\dots(x_n,y_n)\}$ $where\ x_i \epsilon X, y_i \epsilon Y = \{-1,+1\}$

Initialize Distribution: $D_1(i) = \frac{1}{n}$, $i = 1,2,\dots,n$

For t = 1,2, … T:

•Train weak classifier using samples with distribution D_t

•Get weak hypothesis $h_t : X \to \{-1,+1\}$

•Calculate the error of h_t:

$$\varepsilon_t = \sum_{i=1}^{n} D_t(i)\, I_t(\mathrm{i})$$

$$\text{Where } I_t(\mathrm{i}) = \begin{cases} 1 & if\ h_t \neq y_i \\ 0 & if\ h_t = y_i \end{cases}$$

• Calculate the weight of current hypothesis: $\alpha_t = \frac{1}{2} * \ln(\frac{1-\varepsilon_t}{\varepsilon_t})$

•Update distribution: $D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp(-\alpha_t * y_i * h_t(xi))$

$$\text{Where } Z_t = \sum_{i=1}^{n} D_t(i) * \exp(-\alpha_t * y_i * h_t(xi))$$

Output final classifier:

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t * h_t(x_i))$$

Fig. 3. Pseudocode for AdaBoost algorithm

3 Hybrid Model of AdaBoost and BP Neural Network

In this section, a hybrid model is constructed based on AdaBoost and BP neural network for credit scoring. According to the AdaBoost algorithm, the efficiency and accuracy of the ensemble predictor based on a series of weak classifiers largely depends on the accuracy of those base learners, the correlation coefficient of the ensemble predictor and the base classifiers is directly proportional. As BP neural network model is very fledged both in theory and practical applications with high accuracy and efficiency compared to other widely used models. So, in this study, BP neural network is selected as the weak classifier, and as the final ensemble classifier requires a certain method to combine those weak learners into an efficient ensemble system, we adopt the weighted voting method to combine the outputs of each weak classifier for the final outputs. The frame and pseudocode of hybrid model based on AdaBoost and BP neural network are shown in Figs. 4 and 5. The main steps used to construct ensemble predictor are shown as follows.

Step 1. Select the training samples randomly from the chosen databases with the method of 10-fold cross-validation, and each group training sample is assigned with the same weights:

$$\mathrm{T} = \{(x1, y1), \cdots, (xi, yi), \cdots, (xn, yn)\} \text{ while } xi \in X, yi \in Y = \{-1, +1\}.$$

The weights of this distribution on training samples on round t is denoted $D1(i)$. Initially, all weights are assigned with equal value $D1(i)$:

$$D1(i) = \frac{1}{n}, i = 1, 2, \cdots, n, \tag{1}$$

where n represents the total number of samples in the training data set;

Step 2. Parameters of the BP neural network, which include the structure of the network, neurons of each layer, expected goal, are set up according to the attributes of the input and output training samples. Then initialize the weights and biases of BP neural network.

Step 3. Training the BP neural network with the processed databases:

Step 3.1 Calculate input and output values of hidden layer: net_j, b_j

$$\mathrm{net}_j = \sum_{i=1}^{n} w_{ij} \times x_i - \theta_j, \tag{2}$$

$$b_j = f(\mathrm{net}_j), \tag{3}$$

where w_{ij} is the connection weights between input layer and hidden layer and θ_j represents the Threshold values of hidden layer, $f(x)$ represents the transfer function Sigmoid. Namely:

$$f(\mathrm{net}) = \frac{1}{1 + e^{-\mathrm{net}}}. \tag{4}$$

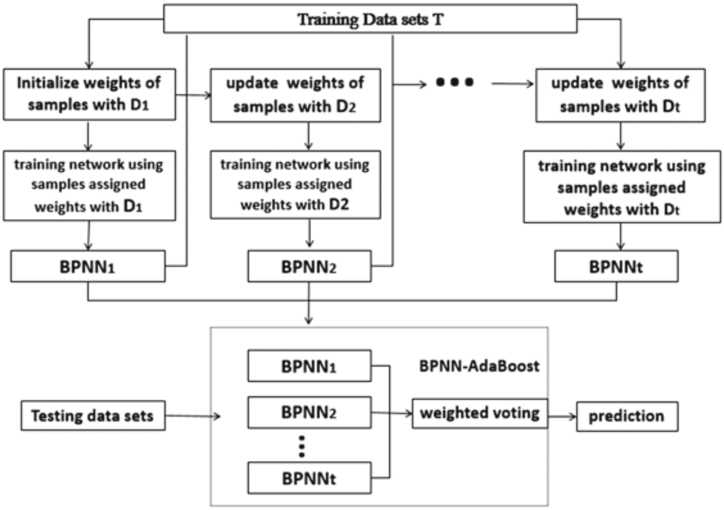

Fig. 4. The frame of hybrid model of AdaBoost and BP neural network

AdaBoost Model Based on BP neural network

Input: Training dataset $T = \{(x_1,y_1),\ldots(x_i,y_i),\ldots(x_n,y_n)\}$, where $x \in R^n, y_i \in \{-1,+1\}$

Weak classifier training algorithm (Back-Propagation algorithm)

Number of Learning Round: T

Initialize Distribution: $D_1(i) = \frac{1}{n}$, $i = 1,2,\ldots,n$

For t=1: T Do

1. Select the subset training data drawn from the distribution $D_1(i)$.
2. Train the base classifier with the samples and obtain the hypothesis $h_t, h_t \in \{-1,+1\}$.
3. Calculate the classifying error of $h_t(x)$:

   $\varepsilon_t = \sum_{i=1}^{n} D_t(i) I(h_t(x_i)$ where $I(h_t) = \begin{cases} 1 & if\ h_t(x_i) \neq y_i \\ 0 & if\ h_t(x_i) = y_i \end{cases}$
4. Calculate the weight of current hypothesis h_t

   $\alpha_t = \frac{1}{2}\ln(\frac{1-\varepsilon_t}{\varepsilon_t})$
5. Update distribution $D_t(i)$

   $D_{t+1}(i) = \frac{D_t(i)}{Z_t}\exp(-\alpha_t * y_i * h_t(x_i))$

   Where Z_t is normalization constant for$D_{t+1}(i)$ become a proper distribution.

Output: Construct the final classifier based on the weak classifiers that have been trained using weighted voting method.

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t * h_t(x))$$

Fig. 5. Pseudocode of hybrid model of AdaBoost and BP neural network

Step 3.2 Calculate input and output values of output layer:

$$\text{net}_i = \sum_{j=1}^{m} v_{jt} \times b_j - \gamma_i, \tag{5}$$

$$o_i = f(\text{net}_i), \tag{6}$$

where v_{jt} is the connection weights between hidden layer and output layer and γ_i represents the Threshold values of output layer. In this study, the hidden layer contains m neurons.

Step 3.3 Calculate the error d_i of each neuron of output using network's goal vector and its response output:

$$d_i = y_i - o_i, \quad (i = 1, 2, \cdots, n). \tag{7}$$

Step 3.4 Calculate the total error of the network $e(k)$:

$$e(k) = \frac{1}{2}\sum_{i=1}^{n} (d_i)^2. \tag{8}$$

Step 3.5 If the total error above is acceptable, the process stops. Otherwise, revise and. There are many ways of weight changes, we use the most common gradient descent method:

$$w_{ij}^{k+1} = w_{ij}^{k} + \Delta w_{ij}^{k} + \partial \Delta w_{ij}^{k-1}, \tag{9}$$

$$w_{jt}^{k+1} = w_{jt}^{k} + \Delta w_{jt}^{k} + \partial \Delta w_{jt}^{k-1}. \tag{10}$$

This process continuous till the system output error is reduced to an acceptance degree or obtains the network constraints.

Step 4. When training epochs go to round t, a base classifier $h_t(x)$ could be achieved with the weights distribution D_t and then calculate the weighted error ε_t from model:

$$\varepsilon_t = \sum_{i=1}^{n} D_t(i) \times I(h_t(x_i)), \tag{11}$$

where I($h_t(x_i)$) is the indicator function, and its mathematical expression shows as below:

$$I(ht) = \begin{cases} 1, & \text{if } h_t(x_i) \neq y_i, \\ 0, & \text{if } h_t(x_i) = y_i. \end{cases} \tag{12}$$

Step 5. Calculate the weight of base classifier $h_t(x)$

$$a_t = \frac{1}{2} \ln \left(\frac{1 - \varepsilon_t}{\varepsilon_t} \right). \tag{13}$$

As is shown above, if the prediction error of the samples is less than 0.5, the weight is bigger than zero, and with the decreasing of the prediction error, will be increased meanwhile, which means a larger voting weight of this weak learner in the final ensemble output.

Step 6. Update the weights of each samples in the training databases for the next iteration:

$$D_{t+1}(i) = \frac{D_t(i)}{Z_t} \exp(-a_t \times y_i \times h_t(x_i)), \tag{14}$$

where Z_t is normalization constant for $D_{t+1}(i)$ become a proper distribution:

$$Z_t = \sum_{i=1}^{n} D_t(i) \times \exp(-a_t \times y_i \times h_t(x_i)). \tag{15}$$

In this way, the weights of training data which are wrongly classified by current weak learner will be increased or decreased if the samples are correctly classified.

Step 7. As the training iteration goes on, a series of base classifiers will be obtained, then combine these classifiers during the processes:

$$f(x) = \sum_{t=1}^{T} a_t \times h_t(x). \tag{16}$$

A series of base classifiers with diversity are collected with loops from step 2 to step 5. Finally, an AdaBoost model based on the weak learners trained above is constructed:

$$H(x) = \mathrm{sign}(\sum_{t=1}^{T} a_t \times h_t(x)) \tag{17}$$

4 Empirical Analysis

In order to test the performance of the proposed AdaBoost algorithm based on BP neural network model, two available real world credit data sets with detailed input attributes description (German and Australian credit data sets) are used in this study. These two credit databases are available from open access UCI Machine Learning Repository. The German credit data set consists of 1000 instances with 700 samples labeled as creditworthy and 300 samples classified as poor credit. For each instance, there are 24 input variables described 19 attributes with 4 attributes changed to dummy variables. Also, the 25th variable is the label of the instance with two numerical descriptions, and label 1 stands for a worthy credit assessment of current instance, label 2, on the contrary, represents a bad credit evaluation. The meaning of original attributes are described in Table 1; The Australian credit data set is interesting because there is a good mix of attributes – continuous, nominal with small numbers of values, and nominal with larger numbers of values, which includes 690 instances in total with 383 samples classified as trustworthy and 307 samples labeled as poor credit. For each instance, 15 variables described 15 features of personal information and financial history of applicants, the last feature is labeled as approved (marked as 0) or rejected (marked as 1). All attribute names and values have been changed to meaningless symbols to protect confidentiality of the data.

Table 1. Original attributes in the German credit data set

Number	Description	Class
Attribute 1	Status of existing checking account	Qualitative
Attribute 2	Duration in month	Numerical
Attribute 3	Credit history	Qualitative
Attribute 4	Purpose	Qualitative
Attribute 5	Credit amount	Numerical
Attribute 6	Savings account/bonds	Qualitative
Attribute 7	Present employment since	Qualitative
Attribute 8	Instalment rate in percentage of disposable income	Numerical
Attribute 9	Personal status and sex	Qualitative
Attribute 10	Other guarantors	Qualitative
Attribute 11	Present residence since	Numerical
Attribute 12	Property	Qualitative
Attribute 13	Age in years	Numerical
Attribute 14	Other instalment plans	Qualitative
Attribute 15	Housing	Qualitative
Attribute 16	Number of existing credits at this bank	Numerical
Attribute 17	Job	Qualitative
Attribute 18	Number of people being liable to provide maintenance for	Numerical
Attribute 19	Telephone	Qualitative
Attribute 20	Foreign worker	Qualitative

4.1 Experiment Design

In the experiment, the AdaBoost algorithm model based on BP neural network is used. The final classifier contents ten weak learners constructed with a three-layer feed-forward BP neural network model. As to the neurons in hidden of each weak learner, 9 nodes in hidden layer dealt with German data set and 20 nodes in hidden layer in Australian data set are set. For comparison purpose, some commonly used models, such as linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), logistic analysis (Log A), K-Nearest neighbor (KNN), artificial neural network (ANN), back propagation neural network (BPNN), and least squares support vector machine (LSSVM) are also applied. In this study, training samples are selected by the method of 10-fold cross-validation to determine the model parameters. In BP neural network model, a common three-layer feed forward net trained by Levenberg-Marquartdt algorithm is employed, the transfer function in hidden layer is applied with Sigmoid function, and with Purelin function in output layer. In addition, three evaluation criteria, Type I Accuracy, Type II Accuracy and Total Accuracy are used, which are defined as follows [34]:

$$\text{Type I Accuracy} = \text{Specificity} = \frac{\text{number of both observed bad and classified as bad}}{\text{number of observed bad}}, \tag{18}$$

$$\text{Type II Accuracy} = \text{Sensitivity} = \frac{\text{number of both observed good and classified as good}}{\text{number of observed good}}, \tag{19}$$

$$\text{Total Accuracy} = \frac{\text{number of correct classification}}{\text{number of evaluation sample}}. \tag{20}$$

4.2 Experiment Result

As is shown in Tables 2 and 3, validation results for the German credit data set and Australian data set are achieved using the different algorithms and settings described above. The classification results of the first six classifiers in Tables 2 and 3 are from another article [35]. From Tables 2 and 3, several interesting findings can be drawn.

First of all, the total accuracy of the proposed AdaBoost model based on BP neural network is much better than other classifiers in both German and Australian data sets. Followed by the least squares support vector machine. Probable explaining reasons of the high accuracy achieved by AdaBoost model includes two aspects. For one thing, the accuracy of AdaBoost model depends much on the accuracy of the weak learners which it ensembles, and as we can see from Tables 2 and 3, BP neural network do a good credit classification compared to other commonly used models. For another reason, the main idea of AdaBoost system gives larger weights to those base classifiers with small prediction error, and as a result, the total accuracy of AdaBoost model get the best performance among all the classifiers compared in this study.

Second, referring to Type I Accuracy, the presented AdaBoost model based on BP neural network performs the best, compared to other classifiers employed in this study both in German credit data set and Australian credit data set. Followed by the linear discriminant analysis model in German data set and logistic analysis model in Australian data set.

Third, in terms of Type I Accuracy and Type II Accuracy, all models used in this study perform better with higher accuracy in Type II Accuracy than Type I Accuracy in general, which indicates that it's more difficult to classify customers with bad credit conditions from those of good credit evaluation cause of the complexity of credit risk.

Fourth, according to the Type II Accuracy shown in Table 2, our proposed model only ranks the fourth compared to other models for comparison. Supposed reason of this phenomenon may be the default of the AdaBoost algorithm, there is no exotic existence of noisy data in German data set, which will attract more attention of our AdaBoost model based on BP neural network on these noisy signals, as a result, the accuracy of our proposed model may not be satisfied as we wish and thus performs ordinarily in German credit data set.

Finally, compared with Tables 2 and 3, it's easy to find out that the performance of the German data set is worse than that of the Australian data set.

Table 2. Credit evaluation results comparison of different models for German credit dataset

Models	Type I Accuracy		Type II Accuracy		Total Accuracy	
	%	Rank	%	Rank	%	Rank
LDA	72	2	74.57	6	73.8	5
QDA	66.57	3	69.33	8	67.4	8
Log A	50.33	5	88.14	3	76.8	3
KNN	27	8	90.57	1	71.5	6
ANN	46.89	7	73.46	7	69.43	7
BPNN	63.83	4	79.36	5	75.8	4
LSSVM	49.67	6	88.86	2	77.1	2
BPNN_AdaBoost	78.7	1	84.19	4	83	1

Table 3. Credit evaluation results comparison of different models for Australian credit dataset

Models	Type I Accuracy		Type II Accuracy		Total Accuracy	
	%	Rank	%	Rank	%	Rank
LDA	80.94	6	92.18	2	85.94	5
QDA	66.12	8	91.38	3	80.14	6
Log A	85.9	2	86.32	6	86.09	4
KNN	81.72	5	54.4	8	69.57	8
ANN	72.56	7	83.61	7	78.94	7
BPNN	83.95	4	87.65	5	86.23	3
LSSVM	85.12	3	89.25	4	86.96	2
BPNN_AdaBoost	89.06	1	94.59	1	92.03	1

There are two possible reasons. On one hand, the credit market in Germany is more complex than that in Australia. On the other hand, there is more non-linearity in the German data set than in the Australian data set. Overviewing the above performance, one interesting phenomenon shows that there is a great capability raising of all three evaluation criteria both in the German data set and Australian data set compared BP neural network with the AdaBoost model based on BP neural network, from the viewpoint of Total Accuracy, there is a 9.5% raising in the German data set and a 6.7% raising in the Australian data set.

5 Conclusion

In this paper, a hybrid model of AdaBoost and BP neural network is proposed for credit risk evaluation. According to the empirical results, we find that our proposed hybrid model is the best one compared with other seven models for two publicly available credit data sets, which indicates that our proposed hybrid model of AdaBoost and BP neural network has a good practicability for credit scoring. In the future, we will use other method as base classifiers, for example, support vector machine and decision tree, and research other ensemble algorithm like bagging algorithm for credit risk assessment.

Acknowledgements. This research was supported by the project of Research Center for System Sciences and Enterprise Development (Grant No.Xq16C03).

References

1. Altman EI (1968) Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. J Finan 23(4):589–609
2. Baesens B, van Gestel T et al (2003) Benchmarking state-of-art classification algorithms for credit scoring. Oper Res Soc 54(6):627–635
3. Baesens B, Setiono R et al (2003) Using neural network rule extraction and decision tables for credit-risk evaluation. Manag Sci 49(3):312–329
4. Beynon MJ, Peel MJ (2001) Variable precision rough set theory and data discretisation: an application to corporate failure prediction. Omega 29(6):561–576
5. Carter C, Catlett J (1987) Assessing credit card applications using machine learning. IEEE Expert 2(3):71–79
6. Chen M, Ma L, Gao Y (2010) Vehicle detection segmentation based on adaboost and grabcut. In: IEEE International Conference on Progress in Informatics and Computing, pp 896–900
7. Chen MC, Huang SH (2003) Credit scoring and rejected instances reassigning through evolutionary computation techniques. Expert Syst Appl 24(4):433–441
8. Desai VS, Crook JN, Overstreet J (1996) A comparison of neural networks and linear scoring models in the credit union environment. Oper Res 95(2):24–37
9. Essa EM, Tolba AS, Elmougy S (2008) A comparison of combined classifier architectures for arabic speech recognition. In: International Conference on Computer Engineering & Systems, pp 149–153
10. Freund Y, Schapire RE (1995) A decision-theoretic generalization of on-line learning and an application to boosting. In: European Conference on Computational Learning Theory, pp 119–139
11. Gestel TV, Baesens B et al (2003) A support vector machine approach to credit scoring. Banken Financiewezen 2:73–82
12. Henley WE, Hand DJ (1996) A k-nearest-neighbour classifier for assessing consumer credit risk. J Roy Stat Soc 45(1):77–95
13. Henley WE, Hand DJ (1997) Construction of a k-nearest-neighbour credit-scoring system. IMA J Manag Math 8(4):305–321
14. Khashman A (2009) A neural network model for credit risk evaluation. Int J Neural Syst 19(4):285–294
15. Li H, Sun J (2008) Ranking-order case-based reasoning for financial distress prediction. Knowl Based Syst 21(8):868–878
16. Li H, Sun J (2009) Gaussian case-based reasoning for business failure prediction with empirical data in China. Inf Sci 179(1):89–108
17. Li H, Sun J (2009) Predicting business failure using multiple case-based reasoning combined with support vector machine. Expert Syst Appl 36(6):10085–10096

18. Li H, Sun J (2010) Business failure prediction using hybrid2 case-based reasoning (h2cbr). Comput Oper Res 37(1):137–151
19. Li H, Sun J, Sun BL (2009) Financial distress prediction based on or-cbr in the principle of k-nearest neighbors. Expert Syst Appl 36(1):643–659
20. Malhotra R, Malhotra DK (2002) Differentiating between good credits and bad credits using neuro-fuzzy systems. Eur J Oper Res 136(1):190–211
21. Malhotra R, Malhotra DK (2003) Evaluating consumer loans using neural networks. Soc Sci Electron Publishing 31(2):83–96
22. Ong CS, Huang JJ, Tzeng GH (2005) Building credit scoring models using genetic programming. Expert Syst Appl 29(1):41–47
23. Piramuthu S, Piramuthu S (1999) Financial credit-risk evaluation with neural and neurofuzzy systems. Eur J Oper Res 112(2):310–321
24. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. Nature 323(6088):533–536
25. Schebesch KB, Stecking R (2005) Support vector machines for classifying and describing credit applicants: detecting typical and critical regions. J Oper Res Soc 56(9):1082–1088
26. Steenackers A, Goovaerts MJ (1989) A credit scoring model for personal loans. Insur Math Econ 8(1):31–34
27. Thomas LC, Edelman DB, Crook JN (2002) Credit scoring and its applications. SIAM, Philadelphia
28. Varetto F (1951) Genetic algorithms in the analysis of insolvency risk. University of Illinois Press, Champaign
29. Wang Y, Wang S, Lai KK (2006) A new fuzzy support vector machine to evaluate credit risk. IEEE Trans Fuzzy Syst 13(6):820–831
30. West D (2000) Neural network credit scoring models. Comput Oper Res 27(11):1131–1152
31. Yobas MB, Crook JN, Ross P (2000) Credit scoring using neural and evolutionary techniques. IMA J Manag Math 11(2):111–125
32. Yu L, Wang S, Lai KK (2008) Credit risk assessment with a multistage neural network ensemble learning approach. Expert Syst Appl 34(2):1434–1444
33. Yu L, Wang SY et al (2008) Designing a hybrid intelligent mining system for credit risk evaluation. Syst Sci Complex 21(5):527–539
34. Yu L, Wang S, Cao J (2009) A modified least squares support vector machine classifier with application to credit risk analysis. Int J Inf Technol Decis Making 08(4):697–710
35. Zhou L, Lai KK, Yu L (2009) Credit scoring using support vector machines with direct search for parameters selection. Soft Comput 13(2):149–155