Fuzzy Systems and Data Mining VIII
A.J. Tallón-Ballesteros (Ed.)

doi:10.3233/FAIA220367

Credit Risk Assessment by a Comparison Application of Two Boosting Algorithms

Zhichao Si, Hongli Niu[1], Weiqing Wang
School of Economics and Management, University of Science and Technology Beijing, 30 Xueyuan Road, Haidian District, Beijing, 100083, China

Abstract The main purpose of credit risk assessment is to help financial institutions identify applicants with good credit and eliminate applicants with bad credit, minimizing the risk of capital loss and maximizing returns. Recent years have witnessed excellent performance of machine learning in the credit risk prediction. This paper extends the previous research by applying two boosting algorithms, namely AdaBoost and XGboost, to perform the credit scoring for real data from Lending Club. Compared with two statistical methods and three individual classifiers, the results show that (i) AdaBoost and XGBoost obtain higher forecasting accuracy for credit risk, providing stronger discrimination ability. (ii) AdaBoost has a greater ability to discriminate minority classes (defaulters), which can reduce capital losses for institutions. (iii) XGBoost is able to capture more potential benefits for institutions because it is more accurate in identifying majority classes, i.e., non-defaulters.

Keywords. Machine learning, AdaBoost, XGboost, Credit risk assessment, Financial default

1. Introduction

In several decades, with the rapid growth of digital science and internet finance, great changes have taken place in the credit industry. In particular, the emergence of P2P makes the interaction between borrowers and lenders more transparent, because there is no longer the participation of financial intermediaries. According to Ref. [1], credit risks could account for 60% of the total risk of banks, due to the emergence of P2P.

Credit risk prediction can a provide more accurate support for the decision-making of financial institutions. This process is typically constructed as a binary classification model. In the past, the most commonly-used credit assessment methods are expert discriminant method [2] and statistical models [3,4]. However, the former requires a very expensive cost, while the latter requires that the data must meet the assumptions. With the event of computing technology, machine learning algorithms have been proved to have excellent performance in the credit evaluation process [5-7], such as artificial neutral networks, support vector machines and decision tree. Compared with statistical models, these machine learning algorithms do not need the information to satisfy any prior assumptions. Nevertheless, individual classifiers have some limitations. For example, some classifiers are

[1] Corresponding Author, Hongli Niu, University of Science and Technology Beijing, China, Email: niuhongli@ustb.edu.cn

computationally expensive and not universal enough. Therefore, some researchers focus on using ensemble learning and hybrid method to evaluate the credit. As one of the classic algorithms of boosting, AdaBoost has some applications in credit scoring, and it has been verified the excellent performance of the algorithm [8-10]. Besides, Xgboost, as another representative algorithm, has also been considered and applied to credit risk assessment and good results have been obtained [11-13]. But the existing researches mainly focus on their respective performance. To our knowledge, there is little literature that applies AdaBoost and XGBoost to a same dataset for a detailed comparison to examine their respective strengths. This study deepens the application of AdaBoost and XGBoost in credit risk assessment.

This paper is organized as follows. Sect. 2 introduces the AdaBoost and XGBoost. The experiment on P2P dataset is discussed in Sect. 3. Sect. 4 summarizes the results.

2. Methodologies

2.1. Adaboost

AdaBoost is a prevalent boosting algorithm proposed by Freund and Schapire [14]. Suppose $S=\{(x_1,y_1),(x_2,y_2),\ldots,(x_T,y_T)\}$ is a collection of sample data, and $y_i\left(i=1,2,\ldots,\mathrm{t}\right)\in\{-1,1\}$ stands for two categories of labels. The achievement process of AdaBoost include the following steps:

First, we set initial weights for the dataset. Second, for *m=1,2,3,....M*, AdaBoost, based weight distribution W_m, learns and gets the base classifier $C_m(x)$, whose error rate is calculated for the training dataset as follows:

$$e_m=P\left(C_m(x_i)\neq y_i\right)=\sum_{i=1}^{T}w_{mi}I\left(C_m(x_i)\neq y_i\right) \tag{1}$$

The coefficient of $C_m(x)$ is calculated as follows:

$$\beta_m=\frac{1}{2}\log\frac{1-e_m}{e_m} \tag{2}$$

Weight distribution of training data set is updated as follows:

$$W_{m+1}=\left(w_{m+1,1},\ldots,w_{m+1,i},\ldots,w_{m+1,T}\right) \tag{3}$$

$$w_{m+1,i}=\frac{w_{mi}}{O_m}\exp\left(-\beta_m y_i C_m(x_i)\right),\quad i=1,2,\cdots,T \tag{4}$$

O_m is the normalization factor that is calculated as follows:

$$O_m=\sum_{i=1}^{T}w_{mi}\exp\left(-\beta_m y_i C_m(x_i)\right) \tag{5}$$

Finally, the linear combination of base classifiers is constructed as expresses in Eq.(6) and the final classifier is obtained as Eq.(7).

$$g(x)=\sum_{m=1}^{M}\beta_m C_m(x) \tag{6}$$

$$C(x) = \mathrm{sign}(g(x)) = \mathrm{sign}\left(\sum_{m=1}^{M} \beta_m C_m(x) \right) \tag{7}$$

2.2. eXtreme gradient boosting tree

In eXtreme gradient boosting tree (XGBoost), CART trees are constructed one at a time. By partitioning the feature values, a new function is learned for each new tree. In addition, the application of second-order derivative functions and the consideration of model complexity are important characteristics of XGBoost. The objective function of XGBoost model is:

$$obj = \sum_{i=1}^{m} l\left(y_i^t, \hat{y}_i^{(t-1)} + f_t(x_i)\right) + \sum_{k=1}^{K} \Omega(f_k) \tag{8}$$

where k represents the k th tree; Ω is the complexity of the model, transformed by the structure of the tree. The objective function is approximated by Taylor expansion.

The DT complexity is computed as:

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{9}$$

g_i and h_i are given as follows:

$$g_i = \frac{\partial l\left(y_i^t, \hat{y}_i^{(t-1)}\right)}{\partial \hat{y}_i^{(t-1)}}, \quad h_i = \frac{\partial^2 l\left(y_i^t, \hat{y}_i^{(t-1)}\right)}{\partial \left(\hat{y}_i^{(t-1)}\right)^2} \tag{10}$$

By combining Eqs(8) and (10) and formal transformation, we can obtain the objective function:

$$Obj \simeq \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{11}$$

For XGBoost, each leaf node has a prediction score, also known as prediction weight, expressed by $f_k(x_i)$ or ω . For each tree, it has its own unique structure. In this structure, $q(x_i)$ is used to represent the leaf node where the sample x_i is located, and ω is used to represent the score of the sample falling on the $q(x_i)$-th leaf node of the *t*-th tree. Therefore, the objective function can be transformed as follows:

$$obj = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T \tag{12}$$

A key point of the algorithm is to find the max score among the given parameters. The tree model of the algorithm will stop growing when the result of the loss function is less than some predefined threshold.

3. Empirical experiment

3.1. Data description and preprocessing

In this work, the credit data[2] for empirical study is collected from the Lending Club in the United States. In the dataset, the status of the loan (marked as loan status) has two

[2] API: kaggle kernels output faressayah/lending-club-loan-defaulters-prediction -p /path/to/dest

category labels: ‘Charged off’ and ‘Fully paid’. A charged off is a loan default. This status is marked as 1. In contrast, fully paid samples are marked as 0.

Next, feature engineering is performed. The initial data is described by 27 features. Firstly, a classification feature can be considered for direct deletion when the category that it contains exceeds a certain value. Besides, if the information provided by two or more features is repeated, the redundant feature will be deleted. In addition, samples pretreatment is also crucial. Specifically, samples with missing values will be removed directly if the null values of a feature account for less than 0.5%. Considering the negative impact that inconsistency in magnitudes can have on model, and also to circumvent the effect of extreme values, the numerical attributes need to be normalized.

Table 1. Sample data.

Data name	Number of observation(defaulter/non-defaulter)	Ratio of defaulters to non-defaulters	Number of variables (including the labels)
Loan data set of Lending Club	395,219 (77,523/317,696)	1:4.1	22

Table 2. Variables.

Features related to the applicant	Features related to loan characteristics
Home_ownership	Loan_amount
Annual_inc	Term
Verification_status	Int_rate
Purpose	installment
Zip_code	subgrade
Dti	
Earliest_cr_line	
Pub_rec	
Open_acc	
Revol_bal	
Revol_util	
Initial_listing_status	
Total_acc	
Application_type	
Mort_acc	
pub_rec_bankruptcies	

As Table 1 shows, there are 77,523 defaulters and 31,7696 non-defaulters in the total sample, and the ratio is 1:4.1. Table 2 reveals variables adopted, which is categorized into two categories: features regard to the applicants and loan characteristics. In the former, verification_status represents whether the borrower's income source has been officially verified by Lending Club. The debt-to-income ratio (Dti) is one measure of an individual's financial health and is calculated by dividing an individual's monthly debt, excluding mortgages and required letter of credit loans, by the borrower's monthly income. Earliest reported credit line (earliest_cr_line) is the time the line of credit was first opened in the borrower's self-report, specific to the month. The number of credit lines in the applicant's credit history that have not yet been fully liquidated is represented by Open_acc. Total credit revolving balance can be seen in Revol_bal. Pub_rec implies the number of derogatory public records. Revol_util is the utilization rate of revolving credit. Total_acc is the amount of credit available to the borrower totally. Initial_list_status indicates the starting status of the loan. Application_type is used to distinguish whether a loan is an individual application or a combined application of two people. Number of mortgage accounts opened and number of publicly recorded bankruptcies is marked as mort_acc and pub_rec_bankruptcies. In the latter, term is refers to the repayment duration, which is divided into 36 months or 60 months. Interest rate on the loan is represented by int_rate. Finally, Installment is the amount of loan repaid at each repayment point.

The total number of observations was divided into 70% and 30%. The former was used as the training set and the latter as the test set. Unbalanced data sets allow the model to learn the wrong data patterns, especially when the ratio of the two types of labels is very different, which can lead to an algorithm that performs well in the training set but very poorly in the test set. In order to avoid this unreasonable situation, this paper uses Synthetic Minority Oversampling Technique (SMOTE) [15] on training set to overcome the problem of imbalance data. As a simple and efficient oversampling method, SMOTE balances the sample labels by generating new samples.

After using SMOTE, the number of data entries is 444,774 for training, which including equal proportion of defaulters and non-defaulters, and 190,618 for testing.

3.2. Validation of different models

In this section, a number of machine learning models are constructed to verify whether AdaBoost and XGBoost relatively perform better in credit risk assessment, compared with two statistic technique including logistic regression (LR) and linear discriminant analysis (LDA) and several artificial intelligence methods including decision tree (DT), super vector classifier (SVC), k-nearest neighbor classification (KNN). The experiment is performed based on Python 3.8.8 in Windows 10. The main libraries used include sklearn 1.0.2 and xgboost1.5.2.

Table 3 gives the evaluation of prediction results. As is shown, the highest accuracy for XGBoost (0.926) indicates that XGBoost has a stronger discrimination ability on the whole, which is of great significance to the credit department of financial institutions. The result that accuracy of SVC (0.925) is lightly higher than that of AdaBoost (0.922) demonstrates an excellent ability of SVC. Among several other models, DT obtains the lowest accuracy (0.835), which demonstrates its relatively weak classification ability. LR, LDA and KNN obtained a compromise level of performance in turn, corresponding to the accuracy of 0.914, 0.905 and 0.902. Besides, F1-score reveals the same tendency: XGBoost is optimal (0.922) and SVC is slightly more preferable than AdaBoost (0.919 vs 0.918). As the harmonic average of precision and sensitivity, F1-score considers the balance between the requirements for classifying minority category (defaulters) correctly and the need to avoid the wrong classification of majority category. Therefore, XGBoost also wins in this performance. The result of DT on this indicator is still the worst (0.835). KNN and LDA get the same F1-score (0.898). In addition, for the comprehensive performance of the algorithm described by AUC, the highest value (0.97) is obtained for both AdaBoost and XGBoost. Then LR, SVC and LDA get same AUC (0.96) and are followed by KNN (0.95). Similar to the other two indicators, DT has the smallest result

(0.91).The corresponding receiver operative curves are shown in Figure 1. DT obtains the worst ROC curve, and among the rest of the models, AdaBoost and XGBoost acquire the optimal ROC.

In order to verify which boosting algorithm is more suitable for credit risk assessment, a more detailed comparison is made in Table 4 for AdaBoost and XGBoost. First, it is worth noting that AdaBoost has a higher sensitivity (0.873) than XGBoost (0.871), which indicates that AdaBoost classifies more defaulters correctly. This result is significant for the institution because identifying defaulters among applicants can reduce losses in principal and interest. Besides, XGBoost classifies more of the majority classes (non-defaulters) correctly because it has a higher precision and specificity. This outcome indicates that XGBoost can identify more non-defaulters and institution can gain more potential benefits from this. In addition, XGBoost is superior to AdaBoost on comprehensive indicators (accuracy, F1-score and AUC) according to Table 3.

Table 3. Indicators of different classifiers.

Classifier	DT	LR	SVC	KNN	LDA	AdaBoost	XGBoost
Accuracy	0.835	0.914	**0.925**	0.902	0.905	**0.922**	**0.926**
F1-score	0.827	0.909	**0.919**	0.898	0.898	**0.918**	**0.922**
AUC	0.91	0.96	0.96	0.95	0.96	**0.97**	**0.97**

Table 4. Indicators of AdaBoost and XGBoost on test set.

Classifier	Precision	Sensitivity	Specificity
AdaBoost	0.967	**0.873**	0.971
XGBoost	**0.979**	0.871	**0.981**

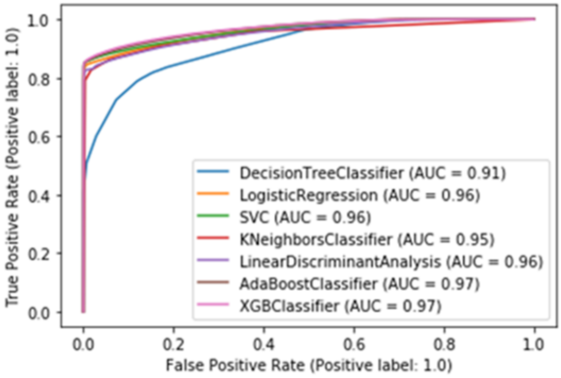

Figure 1. Receiver operative curves of different classifiers

4. Conclusions

In this work, AdaBoost and XGBoost from machine learning are constructed based on P2P data set and compared with two statistical methods (LR, LDA) and three individual classifiers (DT, SVC, KNN). The empirical results show that AdaBoost and XGBoost obtain higher accuracy, F1-score and AUC, which indicates that these two methods provide stronger discrimination ability. Besides, AdaBoost has a stronger ability to distinguish minority classes, i.e., defaulters. In addition, XGBoost can capture more potential benefits for institutions because it is more exact in identifying majority classes, i.e., non-defaulters. These results are instructive for real business scenarios. This has important implications for the healthy and sustainable development of the P2P market. In the future research, more loan data sets should be used to verify the advantages of boosting algorithm. Besides, further research can be devoted to designing new feature engineering ideas to improve the performance of the algorithm.

5. Acknowledgments

The work was partially supported by the Fundamental Research Funds for the Central Universities (No. FRF-BR-20-04B).

References

[1] K. Buehler, A. Freeman and R. Hulme, The new arsenal of risk management. *Harv. Bus. Rev.* 86 (2008) 92-100.

[2] A. B. Hens and M. K. Tiwari, Computational time reduction for credit scoring: An integrated

[3] E. Rosenberg and A. Gleit, Quantitative methods in credit management: a survey, *Oper. Res.* 42 (1994) 589-613.

[4] S. Y. Sohn, D. H. Kim and J. H. Yoon, Technology credit scoring model with fuzzy logistic regression, *Appl. Soft Comput.* 43 (2016) 150-158.

[5] P. Golbayani, I. Florescu and R. Chatterjee, A comparative study of forecasting corporate credit ratings using neural networks, support vector machines, and decision trees, *N. Am. J. Econ. Financ.* 54 (2020) 101251.

[6] V. J. Silvija and P. Marko, Data mining for assessing the credit risk of local government units in Croatia, *Croat. Oper. Res. Rev.* 8 (2017) 193-205.

[7] Z. Li, Y. Tian, K. Li, F. Zhou and W. Yang, Reject inference in credit scoring using Semi-supervised Support Vector Machines, *Expert Syst. Appl.* 74 (2017) 105-114.

[8] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim and A. K. Nandi, Credit Card Fraud Detection Using AdaBoost and Majority Voting, *IEEE Access* (2018) 1-1.

[9] T. Damrongsakmethee and V. E. Neagoe, C4.5 Decision Tree Enhanced with AdaBoost Versus Multilayer Perceptron for Credit Scoring Modeling, in Proceeding of *Computational Statistics and Mathematical Modeling Methods in Intelligent Systems*, Cham 2019.

[10] K. Randhawa, C. K. Loo, M. Seera, C. P. Lim and A. K. Nandi, Credit Card Fraud Detection Using AdaBoost and Majority Voting, *IEEE Access* (2018) 1-1.

[11] H. Li, Y. Cao, S. Li, J. Zhao and Y. Sun, XGBoost Model and Its Application to Personal Credit Evaluation, *IEEE Intell. Syst.* **35** (2020) 52-61.

[12] C. V. Priscilla and P. Prabha, Influence of Optimizing XGBoost to handle Class Imbalance in Credit Card Fraud Detection, in *IEEE- International Conference on Smart Systems and Inventive Technology*, 2020.

[13] H. Li, Y. Cao, S. Li, J. Zhao and Y. Sun, XGBoost Model and Its Application to Personal Credit Evaluation, *IEEE Intell. Syst.* 35 (2020) 52-61.

[14] Y. Freund and R. E. Schapire, A desicion-theoretic generalization of on-line learning and an application to boosting, in *Computational Learning Theory*, 1995.

[15] S. Wang, S. Liu, J. Zhang, X. Che, Y. Yuan, Z. Wang and D. Kong, A new method of diesel fuel brands identification: SMOTE oversampling combined with XGBoost ensemble learning, *Fuel* 282 (2020).