A Novel Hybrid Data Mining Framework for Credit Evaluation

Yatao Yang[1], Zibin Zheng[1,2], Chunzhen Huang[1], Kunmin Li[1],
and Hong-Ning Dai[3](✉)

[1] School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China
[2] Collaborative Innovation Center of High Performance Computing,
National University of Defense Technology, Changsha 410073, China
[3] Faculty of Information Technology,
Macau University of Science and Technology, Taipa, Macau SAR
hndai@ieee.org

Abstract. Internet loan business has received extensive attentions recently. How to provide lenders with accurate credit scoring profiles of borrowers becomes a challenge due to the tremendous amount of loan requests and the limited information of borrowers. However, existing approaches are not suitable to Internet loan business due to the unique features of individual credit data. In this paper, we propose a unified data mining framework consisting of feature transformation, feature selection and hybrid model to solve the above challenges. Extensive experiment results on realistic datasets show that our proposed framework is an effective solution.

Keywords: Credit evaluation · Data mining · Internet finance

1 Introduction

Internet finance has been growing rapidly in China recently. A number of online financial services, such as Wechat Payment and Yu'E Bao have receive extensive attentions. In addition to the payment services, Internet loan business has an explosive growth. On such platforms, borrowers request the loans online. The Internet loan service providers then help borrowers find proper loan agencies. However, it is critical for lenders to obtain the *credit worthiness* of borrowers so that they can minimize the loan risk (to avoid the loans to low credit users).

How to evaluate the *credit worthiness* of borrowers is one of challenges in Internet loan services. In conventional loan markets, banks (or other small firms) usually introduce credit scoring system [4] to obtain the credit worthiness of borrowers. During the credit evaluation procedure, the loan officer carefully checked the loan history of a borrower and evaluated the loan risk based on the officer's past experience (i.e., domain knowledge). However, the conventional credit evaluation procedure cannot be applied to the growing Internet loan markets due to the following reasons. First, the loan officers only have the limited information

S. Wang and A. Zhou (Eds.): CollaborateCom 2016, LNICST 201, pp. 16–26, 2017.
DOI: 10.1007/978-3-319-59288-6_2

of borrowers through Internet loan service platform. Second, there are a tremendous amount of requests for Internet loan business every day, which demands the prompt approval (or disapproval) for customers. Thus, the tedious and complicated procedure of convention credit evaluations is no longer suitable for the fast growth of Internet loan business. Third, the conventional credit evaluation heavily depends on the judgment of loan officers. For example, the credit evaluation is often affected by the knowledge, experience and the emotional state of the loan officer. As a result, there may exist misjudgments of loan officers. It is implied in [8] that computer-assisted credit evaluation approaches can help to solve the above concerns.

In fact, to distinguish the credit borrowers is equivalent to classifying all borrowers into two categories: the "good" borrowers who have good credits and are willing to pay their debts plus interest on time, and the "bad" users who may reject to pay their debts on time. Many researchers employ multiple supervised machine learning algorithms to solve the problem, such as Neural Network, Decision Tree and SVM. In particular, Huang et al. [6] utilize Support Vector Machine (SVM) and Neural Networks to conduct a market comparative analysis. Angelini et al. [1] address the credit risk evaluation based on two correlated Neural Network systems. Pang and Gong [9] also apply the C5.0 classification tree to evaluate the credit risk. Besides, Yap et al. [11] use data mining approach to improve assessment of credit worthiness. Moreover, several different methods have been proposed in [5,10,12].

Although previous studies exploit various models, there is no unified hybrid model that can integrate the benefits of various models. Besides, the existing models are not suitable for the growing Internet loan business due the following unique features of individual credit data: (i) *high dimension of features*, which can be as large as 1,000; (ii) *missing values*, which can significantly affect the classification performance; (iii) *imbalanced samples*, in which there are much more positive samples than negative samples. The above features result in the difficulties in analyzing credit data.

In light of the above challenges, we propose a unified analytical framework. The main contributions of this paper can be summarized as follows.

- We propose a novel hybrid data mining framework, which consists of three key phases: feature transformation, feature selection and hybrid model.
- We integrate various feature engineering methods, feature transformation procedures and supervised learning algorithms in our framework to maximize their advantages.
- We conduct extensive experiments on realistic data sets to evaluate the performance of our proposed model. The comparative results show that our proposed model has the better performance in terms of classification accuracy than other existing methods.

The remaining paper is organized as follows. We describe our proposed framework in Sect. 2. Section 3 shows experimental results. Finally, we conclude this paper in Sect. 4.

2 Our Framework

In order to address the aforementioned concerns, we propose a hybrid data mining framework for credit scoring. As shown in Fig. 1, our framework consists of three key phases: feature transformation, feature selection, hybrid model. We then describe the three phases in detail in the following sections.

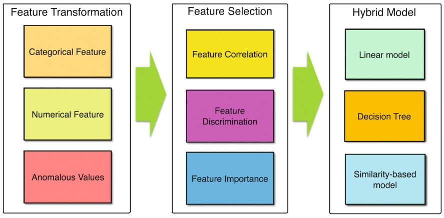


Fig. 1. Our proposed hybrid data mining framework consists of three phases

2.1 Feature Transformation

We categorize the features into two types: (i) *numerical* features are continuous real numbers, representing borrower's age, height, deposit, income, *etc.*; (ii) *categorical* features are discrete integers, indicating borrower's sex, educational background, race, *etc.* Since the two kinds of features cannot be treated as the same, we conduct a conversion so that they can be fit into an unified model.

Categorical Feature Transformation. Regarding to categorical features, we exploit a simple *one-hot encoding*. For example, we use a four-bit one-hot binary to represent four seasons in a year. Specifically, '1000', '0100', '0010' and '0000' denote spring, summer, autumn and winter, respectively. The one-hot encoding conversion is intuitive and easy to be implemented. It converts a categorical feature with the unknown range into multiple binary features with value 0 or 1.

Numerical Feature Transformation. The range of numerical features may differ vastly. For instance, the age is normally ranging from 1 to 100 while the deposit may vary from several hundred to several millions. We utilize the following mapping functions on original features and replace them with the mapped values so that we can reduce the differences between features.

$$Normalize(x_k) = \frac{x_k - mean}{std}, \tag{1}$$

$$Sigmoid(x_k) = \frac{1}{1 + e^{-x_k}}, \tag{2}$$

$$Tanh(x_k) = \frac{e^{x_k} - e^{-x_k}}{e^{x_k} + e^{-x_k}}, \tag{3}$$

$$Maxmin(x_k) = \frac{x_k - \min}{\max - \min}, \tag{4}$$

$$LogAll(x_k) = \log(x_k - \min + 1), \tag{5}$$

where $x_k = \{x_k^{(1)}, x_k^{(2)}, \ldots, x_k^{(n)}\}$ is a set of feature values indicating the kth dimension of the dataset, $x_k^{(i)}$ indicates its value for the ith sample, *mean* denotes the mean value, *std* represents the standard deviation of x_k, max denotes the maximum and min denotes the minimum value. Note that the above basic mapping functions can be nested. For example, a feature can be first transformed by *LogAll* function and can then be mapped into range $(0, 1)$ by *Sigmoid* function.

Anomalous Values Handling. Data sets may contain some values deviated from normal values (i.e., outliers) and some missing values. Specifically, we distinguish outliers by Eq. (6) according to the "3 sigma rules" in classical statistics:

$$Outlier(x_k^{(i)}) = \begin{cases} True, & \text{if } |x_k^{(i)} - mean| \geq 4 \times std \\ False, & \text{otherwise} \end{cases}. \tag{6}$$

Depending on the fraction of anomalous values in feature x_k, we first define the anomalous factor $f = \frac{N_{missing} + N_{outlier}}{N_{sample}}$, where $N_{missing}$ represents the number of missing values, $N_{outlier}$ denotes the number of outliers, and N_{sample} is the number of samples. We then propose three different methods to handle the outliers and the missing values: *replace*, *delete*, and *convert* based on different values of anomalous factor f,

$$Anomalous(x_k) = \begin{cases} Replace, & \text{if } f \leq \alpha \\ Delete, & \text{if } f \geq \beta \\ Convert, & \text{otherwise} \end{cases}. \tag{7}$$

Extra Feature Extraction. We also apply statistical methods to extract extra features. Specifically, we construct ranking features from numerical features and percentage features from categorical features. If the value of the kth numerical feature for the ith sample is $x_k^{(i)}$, the value of ranking feature for it is $a_k^{(i)} = r_k^{(i)}$, where $r_k^{(i)}$ represents $x_k^{(i)}$'s ranking in x_k. However, this simple extension of numerical features significantly increases the dimension, which leads to the extra computational cost. To solve the problem, we use percentiles of the expanded features to represent them in a more concise way. If the extra features are $A = \{a_1, a_2, \ldots, a_n\}$, we use 0th, 20th, 40th, 60th, 80th and 100th percentiles of A as final numerical extra features, which can be represented as $e^{num} = \{a_{0\%}, a_{20\%}, a_{40\%}, a_{60\%}, a_{80\%}, a_{100\%}\}$.

We use a similar method to obtain extra features from categorical features. Suppose $x_k^{(i)}$ represent the kth categorical feature for the ith sample, the value of extra feature for it is $b_k^{(i)} = p_k^{(i)}$, where $p_k^{(i)}$ represents the percentage of category $b_k^{(i)}$ in x_k. If the extra categorical features are $B = \{b_1, b_2, \ldots, b_m\}$, we use 0th, 20th, 40th, 60th, 80th and 100th percentiles of B as final categorical features as $e^{cat} = \{b_{0\%}, b_{20\%}, b_{40\%}, b_{60\%}, b_{80\%}, b_{100\%}\}$.

After feature conversion, each $x_k^{(i)}$ is within the same range, we then use statistics to describe them to capture a high level information $e^{sat} = \{mean, std, perc\}$, where $mean$, std and $perc$ represent the mean value, the standard deviation of $x^{(i)}$ and the percentage of missing values in $x^{(i)}$, respectively.

2.2 Feature Selection

After the feature transformation, the dimension of features can be significantly increased (e.g., 3,000 in our testing datasets), which lead to the high computational complexity. Thus, it is crucial for us to select the most important and informative features to train a good model. In this paper, we combine three different feature selection techniques to extract the most useful features.

Feature Correlation. If two features are correlated to each other, it implies that they convey the same information. Therefore, we can safely remove one of them. Consider an example that a person who has the higher income will pay the more tax. So, we can remove the tax feature and only keep the income feature during model training. There are many methods to measure the correlation (or similarity) between features. In this paper, we use the Pearson Correlation Coefficient (PCC), which is calculated by Eq. (8).

$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}}, \tag{8}$$

where $x = x_1, x_2, \ldots, x_n$ and $y = y_1, y_2, \ldots, y_n$ represent two features, x_i and y_i denote the corresponding values for features x and y in the ith sample, and $\overline{x}$ and $\overline{y}$ denote the means for x and y, respectively. In practice, for the feature pairs whose r_{xy} is higher than 0.95, we arbitrarily remove one of them.

Feature Discrimination. In model training, our goal is to discriminate different categories based on feature information. If a feature itself can distinguish positive and negative samples, implying that it has a strong correlation with the label, we shall include it in model training since it is an informative feature. For instance, F-score [3] is a simple technique to measure the discrimination of two sets of real numbers. Specifically, F-score is calculated by Eq. (9) as follows,

$$\frac{\left(\overline{x}^{+} - \overline{x}\right)^2 + \left(\overline{x}^{-} - \overline{x}\right)^2}{\frac{1}{n^{+}-1}\sum_{k=1}^{n^{+}}\left(x_k^{+} - \overline{x}^{+}\right)^2 + \frac{1}{n^{-}-1}\sum_{k=1}^{n^{-}}\left(x_k^{+} - \overline{x}^{-}\right)^2}, \tag{9}$$

where $\overline{x}$, $\overline{x}^{+}$, $\overline{x}^{-}$ are the average values of the whole sets, the positive and negative data sets, respectively, x_k^{+} is the kth positive instance and x_k^{-} is the kth negative instance. The larger F-score is, the more likely feature x is more discriminative.

Feature Importance. Before applying hybrid model training in Sect. 2.3, we need to evaluate the importance of every feature in training set. Specifically, we choose the features that contribute the most to our model. After each training, we assign a certain importance value v_k to each feature x_k. Taking all information into consideration, we use Eq. (10) to calculate Feature Importance index (FI),

$$FI_k = 0.6 \times v_k^{gbdt} + 0.2 \times v_k^{rf} + 0.2 \times f_k, \tag{10}$$

where v_k^{gbdt} and v_k^{rf} represent importance values given by Gradient Boosting Decision Tree (GBDT) and Random Forest (RF), respectively and f_k denotes F-score of x_k. Since v_k^{gbdt}, v_k^{rf} and f_k may not be within the same range, we use function $Maxmin$ defined in Eq. (4) to normalize them first.

Summary. We illustrate the whole feature selection procedure in Algorithm 1. In particular, after conducting feature transformation, we first remove features with large number of anomalous values. Then, we remove the highly correlated feature. Finally, we calculate Feature Importance Index and select the top K features based on the trained RF and GBDT values.

Algorithm 1. Feature Selection

```
Require: a set of features $X = \{x_1, x_2, \ldots, x_n\}$, selection threshold $K$
Ensure: a subset of $X$
for each $x_k$ in $X$ do
    conduct feature transformation
    if $Anomalous(x_k) == Delete$ then
        delete $x_k$ from $X$
    end if
end for
for each feature pairs $(x_i, x_j)$ in $X$ do
    calculate correlation $r_{ij}$
    if $r_{ij} \geq 0.95$ then
        delete $x_i$ or $x_j$ from $x$
    end if
end for
train a RF and a GBDT with $X$
for each $x_k$ in $X$ do
    obtain feature importance $v_k^{rf}$ and $v_k^{gbdt}$
    calculate F-score $f_k$
end for
$v_k^{rf} \leftarrow Maxmin(v_k^{rf})$
$v_k^{gbdt} \leftarrow Maxmin(v_k^{gbdt})$
$f_k \leftarrow Maxmin(f_k)$
$FI_k \leftarrow 0.6 \times v_k^{gbdt} + 0.2 \times v_k^{rf} + 0.2 \times f_k$
$X' \leftarrow K$ feature $x_k$ in $X$ with largest $FI_k$
return $X'$
```

2.3 Hybrid Model

We first present the models that we use as follows:

- **Linear model.** To reduce the generalization error, we train 10 different Logistic Regression (LR) models with various parameters and blend their results.

Table 1. Performance of AUROC on different methods during different phases

Factor		Classifier		
Data	Dimension	LR	RF	AdaBoost
Original	1138	0.6442 ± 0.0170	0.6549 ± 0.0313	0.6625 ± 0.0335
Extended	1984	0.6566 ± 0.0235	0.6558 ± 0.0314	0.6624 ± 0.0319
Refilled	1984	0.6755 ± 0.0198	0.6573 ± 0.0356	0.6649 ± 0.0329
Selected	200	0.7025 ± 0.0257	0.6635 ± 0.0362	0.6772 ± 0.0291
Data	Dimension	GBDT	XGBoost	LR+XGBoost
Original	1138	0.6574 ± 0.0350	0.6988 ± 0.0058	0.7048 ± 0.0065
Extended	1984	0.6539 ± 0.0431	0.7000 ± 0.0060	0.7103 ± 0.0058
Refilled	1984	0.6548 ± 0.0432	0.7025 ± 0.0059	0.7127 ± 0.0027
Selected	200	0.6722 ± 0.0362	0.7200 ± 0.0049	0.7248 ± 0.0053

- **Decision Tree model.** Gradient Boosting Decision Tree (XGBoost)[2] is a popular scalable Gradient Boosting approach. Like LR, we train 20 different XGBoost models and blend their prediction results.
- **Similarity-based model.** We use Pearson Correlation Coefficient (PCC) to evaluate the similarity between samples. Due the imbalance of samples, we identify the negative samples as many as possible. Therefore, we compare each sample in the test set with each negative sample in the training set and label those with high similarity as negative.

We then describe our proposed hybrid model. In particular, our model exploits one of ensemble classification algorithms - “bagging” [5,7]. More specifically, we average the predictions from various models. With regard to a single model (e.g., LR), we average predictions of the same model with different parameters. We then average results from LR and XGBoost. The bagging method often reduces overfit and smooths the separation board-line between classes. Besides, we also use PCC to identify the samples that are most likely to be negative.

Our hybrid model has a better performance than traditional single model due to the following reasons. Firstly, we exploit a diversity of models and the “bagging” method combines their results together so that their advantages are maximized and their generalization errors are minimized. Secondly, we utilize XGBoost library, which is an excellent implementation of Gradient Boosting algorithm, which is highly efficient and can prevent model from over-fitting.

3 Experiment

We use the sample dataset from CashBus[1], a micro credit company in China, which offers the online loaning service to individuals. The dataset contains 15,000

[1] http://www.cashbus.com/.

samples. Since all the samples are anonymous in order to protect user privacy, we cannot use any domain knowledge in problem analysis. The dataset has the following features: (1) **High Dimension**. The dataset contains 1,138 features, including 1,045 numerical features and 93 categorical features. (2) **Missing values**. There are a total of 1,333,597 missing values in our dataset, making the missing rate 7.81%. The number of missing values for each feature is ranging from 19 to 14,517 and the number of missing values for each sample is ranging from 10 to 1,050. (3) **Imbalanced samples**. There are 13,458 positive samples while only 1,532 negative samples in the dataset.

3.1 Experimental Setup and Evaluation Metrics

We predict the probability that a user has a good credit and evaluate the prediction results by Area under the Receiver Operating Characteristic curve (AUROC), i.e., $\text{AUROC} = \frac{\sum_i S_i}{|P| \times |N|}$, where P and N represent the positive samples and the negative samples in test set, respectively, and S_i is the score for the ith pairs between each positive sample and each negative sample, defined by

$$S_i = \begin{cases} 1, & score_{i-p} > score_{i-n} \\ 0.5, & score_{i-p} = score_{i-n} \\ 0, & score_{i-p} < score_{i-n}, \end{cases} \tag{11}$$

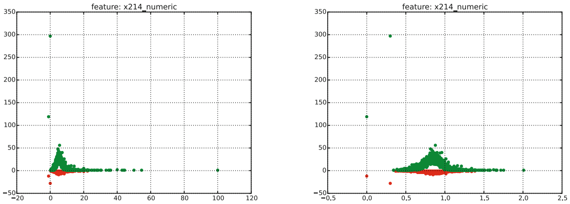
where $score_{i-p}$ and $score_{i-n}$ represent the scores for the positive and the negative sample, respectively. A higher value of AUROC means that the prediction result is more precise.

3.2 Performance Comparison

To investigate the prediction performance, we compare our proposed hybrid model (LR+XGBoost) with other five approaches (each with single model): Logistic Regression (LR), Random Forest (RF), AdaBoost, Gradient Boosting Decision Tree (GBDT) and XGBoost. Table 1 presents the comparative results of different models in different phases in terms of AUROC. Origin represents the raw data set. Extended represents feature transformation. Refilled represents the anomalous values handling process, where we set $\alpha = 0.1$ to choose feature to refill them. Selected represents feature selection process, where we select the top $K = 200$ features. We have the following observations: (1) in all four phases, our proposed hybrid model obtains a better AUROC score than any other methods; (2) our proposed model has a relatively small variation compared with other models, implying the stable performance; (3) LR + XGBoost outperform others, indicating that they are the right choices for constructing the hybrid model.

3.3 Impact of Feature Transformation

We then investigate the impact of feature transformation. Figure 2 shows the impact of *LogAll* function on one numerical feature. After the feature transformation, the distribution of the features becomes more smooth and the extremely large values are minimized.



(a) Before (b) After

Fig. 2. Impact of LogAll Transformation, where green points and red points represent positive and negative samples, respectively. (Color figure online)

3.4 Impact of Refilling Anomalous Values

To deal with the large amount of missing values and outliers, we propose a method to refill the anomalous values based on anomalous value rate under α. We set α to be 0.02 to 0.6 to investigate the impact of α. Table 2 presents the results, where Fill Features represent the number of features that are affected during this process. It is shown in Table 2 that AUROC values for both LR and XGBoost models first increase and then slowly decrease. This can be explained by the fact that filling the anomalous values can bring more information while too many extra filled values also cause noise. In fact, the best performance is obtained when $\alpha = 0.1$.

Table 2. Performance of AUROC on LR and XGBoost under different fill criterion

Factor		Classifier	
Criterion α	Fill features	LR	XGBoost
0.02	661	0.6560 ± 0.0243	0.6991 ± 0.0063
0.05	883	0.6563 ± 0.0240	0.7001 ± 0.0062
0.1	881	0.6755 ± 0.0198	0.7025 ± 0.0059
0.2	885	0.6567 ± 0.0234	0.7021 ± 0.0053
0.3	905	0.6567 ± 0.0227	0.7008 ± 0.0054
0.4	1018	0.6561 ± 0.0231	0.7016 ± 0.0052
0.5	1024	0.6562 ± 0.0238	0.7015 ± 0.0053
0.6	1026	0.6558 ± 0.0239	0.7013 ± 0.0072

3.5 Impact of Feature Selection

After feature transformation, the dimension of features is significantly increased due to the introduction of extra features. We then exploit the feature selection algorithm to reduce the dimension of features. Specifically, we investigate the impact of the feature importance values given by different models and we set

Table 3. Performance of different models under different feature selection methods

Factor		Classifier		
Importance calculation	Dimensions	LR	RF	AdaBoost
RF	200	0.6818 ± 0.0168	0.6626 ± 0.0372	0.6656 ± 0.0359
XGBoost	200	0.6995 ± 0.0263	0.6732 ± 0.0267	0.6695 ± 0.0367
FSore	200	0.6716 ± 0.0193	0.6586 ± 0.0373	0.6608 ± 0.0367
Ensemble	200	0.7025 ± 0.0257	0.6635 ± 0.0362	0.6772 ± 0.0291
Importance calculation	Dimensions	GBDT	XGBoost	LR+XGBoost
RF	200	0.6602 ± 0.0350	0.6888 ± 0.0073	0.6927 ± 0.0423
XGBoost	200	0.6641 ± 0.0333	0.7195 ± 0.0051	0.7214 ± 0.0031
FSore	200	0.6572 ± 0.0454	0.6701 ± 0.0133	0.6843 ± 0.0245
Ensemble	200	0.6722 ± 0.0362	0.7200 ± 0.0049	0.7248 ± 0.0053

the feature selection threshold to be Top $K = 200$. It is shown in Table 3 that our proposed LR+XGBoost achieve the best performance.

3.6 Impact of Selection Threshold

In addition to the feature importance, the threshold top K also contributes to the final quality of the selected features. To investigate the impact of K, we set K to be 100 to 1500 and conduct experiments based on LR and XGBoost models. It is shown in Table 4 that AUROC values first increase and then slowly decrease as K increases. The best performance is obtained when $K = 200$.

Table 4. Performance of LR and XGBoost under different thresholds

Factor	Classifier	
Threshold	LR	XGBoost
100	0.7019 ± 0.0235	0.7156 ± 0.0035
200	0.7025 ± 0.0257	0.7200 ± 0.0049
300	0.6996 ± 0.0184	0.7176 ± 0.0048
400	0.6912 ± 0.0217	0.7129 ± 0.0031
500	0.6853 ± 0.0186	0.7092 ± 0.0038
600	0.6810 ± 0.0157	0.7043 ± 0.0062
800	0.6775 ± 0.0174	0.7032 ± 0.0061
1200	0.6748 ± 0.0237	0.7003 ± 0.0051
1500	0.6721 ± 0.0246	0.7020 ± 0.0060

4 Conclusion and Future Work

In this paper, we propose a novel hybrid data mining framework for individual credit evaluation. To address the challenging issues in individual credit data, such as the high dimension, the outliers and imbalanced samples, we exploit various feature engineering methods and supervised learning models to establish a unified framework. The extensive experimental results show that our proposed framework has a better classification accuracy than other existing methods. There are several future directions in this promising area. For example, we can apply the unsupervised algorithms to utilize the unlabeled data. Besides, we shall use the domain knowledge in finance to further improve the feature transformation and the feature selection procedure.

Acknowledgment. The work described in this paper was supported by the National Key Research and Development Program (2016YFB1000101), the National Natural Science Foundation of China under (61472338), the Fundamental Research Funds for the Central Universities, and Macao Science and Technology Development Fund under Grant No. 096/2013/A3.

References

1. Angelini, E., di Tollo, G., Roli, A.: A neural network approach for credit risk evaluation. Q. Rev. Econ. Finan. **48**(4), 733–755 (2008)
2. Chen, T., Guestrin, C.: Xgboost: A scalable tree boosting system. arXiv preprint arXiv:1603.02754 (2016)
3. Chen, Y.W., Lin, C.J.: Combining svms with various feature selection strategies. In: Guyon, I., Nikravesh, M., Gunn, S., Zadeh, L.A. (eds.) Feature Extraction, pp. 315–324. Springer, Heidelberg (2006)
4. Gray, J.B., Fan, G.: Classification tree analysis using TARGET. Comput. Stat. Data Anal. **52**(3), 1362–1372 (2008)
5. Hsieh, N.C., Hung, L.P.: A data driven ensemble classifier for credit scoring analysis. Expert Syst. Appl. **37**(1), 534–545 (2010)
6. Huang, Z., Chen, H., Hsu, C.J., Chen, W.H., Wu, S.: Credit rating analysis with support vector machines and neural networks: a market comparative study. Decis. Support Syst. **37**(4), 543–558 (2004)
7. Koutanaei, F.N., Sajedi, H., Khanbabaei, M.: A hybrid data mining model of feature selection algorithms and ensemble learning classifiers for credit scoring. J. Retail. Consum. Serv. **27**, 11–23 (2015)
8. Lessmann, S., Baesens, B., Seow, H.V., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. Eur. J. Oper. Res. **247**(1), 124–136 (2015)
9. Pang, S.L., Gong, J.Z.: C5. 0 classification algorithm and application on individual credit evaluation of banks. Syst. Eng. Theory Pract. **29**(12), 94–104 (2009)
10. Wang, Y., Wang, S., Lai, K.K.: A new fuzzy support vector machine to evaluate credit risk. IEEE Trans. Fuzzy Syst. **13**(6), 820–831 (2005)
11. Yap, B.W., Ong, S.H., Husain, N.H.M.: Using data mining to improve assessment of credit worthiness via credit scoring models. Expert Syst. Appl. **38**(10), 13274–13283 (2011)
12. Yu, L., Wang, S., Lai, K.K.: Credit risk assessment with a multistage neural network ensemble learning approach. Expert Syst. Appl. **34**(2), 1434–1444 (2008)

Springer

http://www.springer.com/978-3-319-59287-9

Collaborate Computing: Networking, Applications and Worksharing
12th International Conference, CollaborateCom 2016, Beijing, China, November 10–11, 2016, Proceedings
Wang, S.; Zhou, A. (Eds.)
2017, XIV, 699 p. 279 illus., Softcover
ISBN: 978-3-319-59287-9