FRFE: Fast Recursive Feature Elimination for Credit Scoring

Van-Sang Ha[1(✉)] and Ha-Nam Nguyen[2]

[1] Department of Economic Information System, Academy of Finance, Hanoi, Vietnam
sanghv@hvtc.edu.vn
[2] Department of Information Technology, VNU-University of Engineering and Technology, Hanoi, Vietnam
namnh@vnu.edu.vn

Abstract. Credit scoring is one of the most important issues in financial decision-making. The use of data mining techniques to build models for credit scoring has been a hot topic in recent years. Classification problems often have a large number of features, but not all of them are useful for classification. Irrelevant and redundant features in credit data may even reduce the classification accuracy. Feature selection is a process of selecting a subset of relevant features, which can decrease the dimensionality, reduce the running time, and improve the accuracy of classifiers. Random forest (RF) is a powerful classification tool which is currently an active research area and successfully solves classification problems in many domains. In this study, we constructed a fast credit scoring model based on parallel Random forests and Recursive Feature Elimination (FRFE) . Two public UCI data sets, Australia and German credit have been used to test our method. The experimental results of the real world data showed that the proposed method results in a higher prediction rate than a baseline method for some certain datasets and also shows comparable and sometimes better performance than the feature selection methods widely used in credit scoring.

Keywords: Credit risk · Credit scoring · Feature selection · Random forests · RFE · Machine learning

1 Introduction

The main purpose of credit risk analysis is to classify customers into two sets, good and bad ones [1]. Over the last decades, there have been lots of classification models and algorithms applied to analyze credit risk, for example decision tree [2], nearest neighbor K-NN, support vector machine (SVM) and neural network [3–7]. One important goal in credit risk prediction is to build the best classification model for a specific dataset.

Financial data in general and credit data in particular usually contain irrelevant and redundant features. The redundancy and the deficiency in data can reduce the classification accuracy and lead to incorrect decision [8, 9]. In that case, a feature selection strategy is deeply needed in order to filter the redundant features. Indeed, feature selection is a process of selecting a subset of relevant features. The subset is sufficient to

P.C. Vinh et al. (Eds.): ICTCC 2016, LNICST 168, pp. 133–142, 2016.
DOI: 10.1007/978-3-319-46909-6_13

describe the problem with high precision. Feature selection thus allows decreasing the dimensionality of the problem and shortening the running time.

Credit scoring is a technique using statistical analysis data and activities to evaluate the credit risk against customers. Credit scoring is shown in a figure determined by the bank based on the statistical analysis of credit experts, credit teams or credit bureaus. In Vietnam, some commercial banks start to perform credit scoring against customers but it is not widely applied during the testing phase and still needs to improve gradually. For completeness, all information presented in this paper comes from credit scoring experience in Australia, Germany and other countries.

Many methods have been investigated in the last decade to pursue even small improvement in credit scoring accuracy. Artificial Neural Networks (ANNs) [10–13] and Support Vector Machine (SVM) [14–19] are two commonly soft computing methods used in credit scoring modelling. In order to achieve higher classification performance, SVM recursive feature elimination (SVM-RFE) filter relevant features and remove relatively insignificant feature variables. SVM-RFE uses numerical attribute but credit data sets has a lot of categorical attributes. How to deal with an SVM-RFE with categorical attributes? The conversion of categorical attributes into numerical attributes will lack information and reduce accuracy. Random forest is a popular classification method which deal with this problem. Recently, other methods like evolutionary algorithms [20], stochastic optimization technique and support vector machine [21] have shown promising results in terms of prediction accuracy.

This study proposed a new method for feature selection based on recursive feature elimination and integrated with a parallel Random Forest classifier in credit scoring tasks. The proposed method reduces the set of features via feature ranking criterion. This criterion re-evaluates the importance of features according to the Gini index and the correlation of training and validation accuracy which are obtained from RF algorithm. By that way, we take both feature contribution and correlation of training error into account. We applied the proposed algorithm to classify credit datasets. Integration with H2O parallel random forest, the FRFE showed better classification accuracy and faster than RF.

The rest of the paper is organized as follows: Sect. 2 presents the background of credit scoring, random forests and feature selection. Section 3 is the most important section that describes the details of the proposed model. Experimental results are discussed in Sect. 4 while concluding remarks and future works are presented in Sect. 5.

2 Feature Selection

Feature selection is the most basic step in data pre-processing as it reduces the dimensionality of the data. Feature selection can be a part of the criticism which needs to focus on only related features, such as the PCA method or an algorithm modeling. However, the feature selection is usually a separate step in the whole process of data mining.

There are two different categories of feature selection methods, i.e. filter approach and wrapper approach. The filter approach considers the feature selection process as a precursor stage of learning algorithms. The filter model uses evaluation functions to evaluate the classification performances of subsets of features. There are many evaluation functions such as feature importance, Gini, information gain, the ratio of information gain, etc. A disadvantage of this approach is that there is no relationship between the feature selection process and the performance of learning algorithms.

The wrapper approach uses a machine learning algorithm to measure the good-ness of the set of selected features. The measurement relies on the performance of the learning algorithm such as its accuracy, recall and precision values. The wrapper model uses a learning accuracy for evaluation. In the methods using the wrapper model, all samples should be divided into two sets, i.e. training set and testing set. The algorithm runs on the training set, and then applies the learning result on the testing set to measure the prediction accuracy. The disadvantage of this approach is highly computational cost. Some researchers proposed methods that can speed up the evaluating process to decrease this cost. Common wrapper strategies are Sequential Forward Selection (SFS) and Sequential Backward Elimination (SBE). The optimal feature set is found by searching on the feature space. In this space, each state represents a feature subset, and the size of the searching space for n features is $O(2^n)$, so it is impractical to search the whole space exhaustively, unless n is small.

3 H2O Parallel Random Forests

H2O is a platform for distributed in memory predictive analytics and machine learning. H2O uses pure Java which easy deployment with a single jar, automatic cloud discovery. H2O does in-memory analytics on clusters with distributed parallelized state-of-the-art Machine Learning algorithms. Figure 1 show H2O architecture:

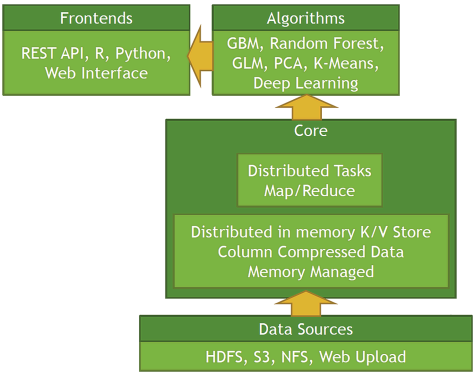

Fig. 1. H2O architecture

Random Forest is an ensemble classifier consisting of a set of CART classifiers using bagging mechanism. Each node of a tree only selects a small subset of features for a split, which enables the algorithm to create classifiers for highly dimensional data very quickly. One has to specify the number of randomly selected features *(mtry)* at each split. The default value is *sqrt(p)* for the classification where p is the number of features. The Gini index is used as the splitting criterion.

Random Forest is an ensemble classifier consisting of a set of CART classifiers using bagging mechanism. Each node of a tree only selects a small subset of features for a split, which enables the algorithm to create classifiers for highly dimensional data very quickly. One has to specify the number of randomly selected features (*mtry*) at each split. The default value is *sqrt(p)* for the classification where p is the number of features. The largest possible tree is grown and not pruned. The big enough number of trees (*ntree*) is chosen to ensure that every input feature is predicted at least several times. The root node of each tree in the forest keeps a set of bootstrapped samples from the original data as the training set to build a tree. The rest of the samples, called out-of-bag (OOB) samples are used to estimate the performance of classification. The out-of-bag (OOB) estimation is based on the classification of the set of OOB samples which is roughly one third of the original samples. H2O's Random Forest algorithm is parallel processing which produces a dynamic confusion matrix. As each tree is built, OOBE (out of bag error estimate) is recalculated. The expected behavior is that the error rate increases before it decreases, as it is a natural outcome of Random Forest's learning process. When there are only a few trees built on random subsets, the error rate is expected to be relatively high. As more trees are added, resulting in more trees "voting" for the correct classification of the OOB data, the error rate should decrease.

4 The Proposed Method

Our proposed method uses H2O parallel random forest (PRF) to estimate performance and reduce running time. We consider the proposed method has two phases. In the first phase, the training set was trained and tested by PRF in order to select the best features. The most important procedure in phase one is to estimate feature ranking value for each feature. A recursive elimination approach was applied to evaluated contribution of each feature to the classifier through one-by-one eliminating feature. The irrelevant feature(s) are eliminated and only the important features are survived by means of feature ranking value. Output of the phase one is a set of selected features. To deal with over-fitting problem, we apply n-fold cross validation technique to minimize the generalization error.

In the second phase, result of learning phase is used as a filter of test dataset. The detail of proposed algorithm will be presented in next section.

In wrapper approaches, they only focus on accuracies of the features when computing the ranking criteria, but not much on the correlation of the features. A feature with good ranking criteria may not turn out a good result. Also, the combination of several features with good ranking criteria may not give out a good result. On the other hand, Recursive

Feature Elimination takes a lot of time to run. To remedy this problem, we propose a procedure named Fast Feature Elimination based on parallel RF (FRFE).

1. Train data by Random Forest with the cross validation
2. Calculate the ranking criterion for all features F_i^{rank} where i = 1..n (n is the number of features).
3. Remove a feature by using *FastFeatureElimination* function (may be more efficient if we remove several features at a time)
4. Back to step 1 until reach the desired criteria.

In step 1, from the j[th] cross validation we get set of (F_j, A_j^{learn}, $A_j^{validation}$, AUC_j^{learn}, $Gini_j^{Learn}$) that are the feature importance. The learning accuracy, the validation accuracy, the area under curve (AUC). Those values will be used to compute the ranking criterion in step 2.

In step 2, we use the results from step 1 to build the ranking criterion which will be used in step 3. The ranking criterion of feature i[th] is computed as follow:

$$F_i^{rank} = \sum_{j=1}^{n} F_{i,j} \times \left(\frac{\left(A_j^{learn} + A_j^{validation} \right)}{\left| A_i^{learn} - A_j^{validation} \right| + \varepsilon} + AUC_j^{learn} \right) \tag{1}$$

where $j = 1, .., n$ is the number of cross validation folders;

$F_{i,j}$ is the feature importance in terms of the node impurity which can be computed by Gini impurity

A_j^{learn} the learning accuracy

$A_j^{validation}$ the validation accuracy of feature j[th] obtained from H2O Random Forest module, respectively.

ε is the real number with very small value.

AUC_j^{learn}: the area under curve (AUC)

The first factor $\left(F_{i,j}\right)$ is presented the Gini decrease for each feature over all trees in the forest when we train data by RF. Obviously, the higher decrease of $F_{i,j}$ is obtained, and the better rank of feature we have. We use the second factor to deal with the over fitting issue as well as the desire of high accuracy. The numerator of the factor presents for our desire to have a high accuracy. The larger value we get, the better the rank of the feature is. We want to have a high accuracy in learning and also want not too fit the training data which so called over fitting problem. To solve this issue, we apply the n-folder cross validation technique. We can see that the less difference between the learning accuracy and the validation accuracy, the more stability of accuracy. In the other words, the purpose of the denominator is to reduce over fitting. In the case of the learning accuracy is equal to the validation accuracy, the difference is equal to 0, we use ε with very small value to avoid the fraction to be ∞. We added AUC measure because the AUC is a commonly used evaluation metric for binary classification problems like predicting a Good (Buy) or Bad (Sell) decision (binary decision). The interpretation is

that given a random positive observation and negative observation, the AUC gives the proportion of the time you guess which is correct. It is more affected by sample imbalance than accuracy. A perfect model will score an AUC of 1, while random guessing will score an AUC of around 0.5. AUC is in fact often predicted over accuracy for binary classification for a number of different reasons.

In step 3: we execute the feature elimination strategy based on backward approach. The proposed feature elimination strategy depends on both ranking criterion and the validation accuracy. The ranking criterion makes the order of features be eliminated and the validation accuracy is used to decide whether the chosen subset of features is permanently eliminated. The new subset is validated by H2O Random Forest module. The obtained validation accuracy plays a role of decision making. It is used to evaluate whether the selected subset is accepted as a new candidate of features. If the obtained validation accuracy is lower than the previous selected subset accuracy, it tries to eliminate other features based on their rank values. This iteration is stopped whenever the validation accuracy of the new subset is higher than the previous selected subset accuracy. If there is either no feature to create new subset or no better validation accuracy, the current subset of features is considered as the final result of our learning algorithm. Otherwise the procedure goes back to step 1. The set of features, which is a result of learning phase, is used as a filter to reduce the dimension of the test dataset before performing predicting those samples in classification phase.

5 Experiment and Results

Our proposed algorithm was coded using R language (http://www.r-project.org), using H2O Random Forest package. This package is optimized for doing "in memory" processing of distributed, parallel machine learning algorithms on clusters. A "cluster" is a software construct that can be fired up on your lap-top, on a server, or across the multiple nodes of a cluster of real machines, including computers that form a Hadoop cluster. We tested the proposed algorithm with several datasets including two UCI public datasets, German and Australian credit approval, to validate our approach. The learning and validation accuracies were determined by means of 5-fold cross validation. In this paper, we used RF with the original dataset as the base-line method. The proposed method and the base-line method were executed on the same training and testing datasets to compare their efficiency.

5.1 Australian Credit

The Australian credit dataset is composed of 690 applicants, with 383 credit worthy and 307 default examples. Each instance contains eight numerical features, six categorical features, and one discriminant feature, with sensitive information being transferred to symbolic data for confidentiality reasons. The averages of classification results are depicted in Fig. 2.

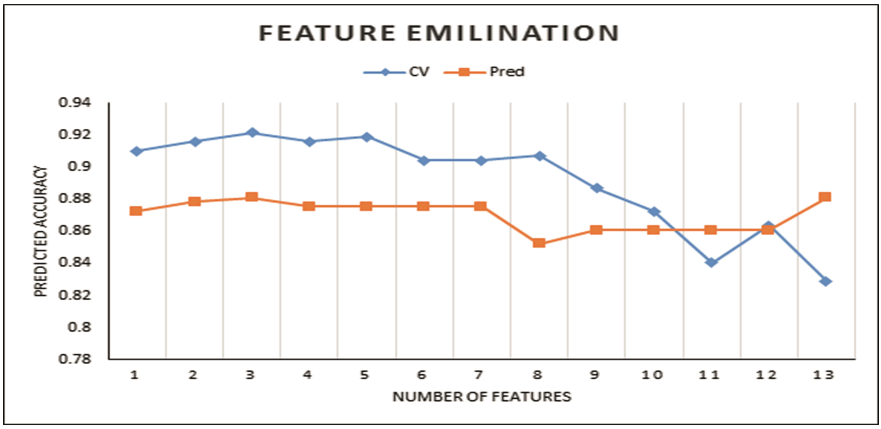

Fig. 2. Accuracy in case of Australian dataset

Table 1 shows the performances of different classifiers over the Australian credit datasets. Baseline is the classifier without feature selection. Classifiers used in [22] include: Linear SVM, CART, k-NN, Naïve Bayes, MLP. Filter methods include: t-test, Linear Discriminant analysis (LDA), Logistic regression (LR). The wrapper methods include: Genetic algorithms (GA) and Particle swarm optimization (PSO).

Table 1. Compare performances of different classifiers over the Australian credit dataset

Classifier	Filter methods			Wrapper methods		Baseline
	t-test	LDA	LR	GA	PSO	
Linear SVM	85.52	85.52	85.52	85.52	85.52	85.52
CART	85.25	85.46	85.11	84.85	84.82	85.20
k-NN	86.06	85.31	84.81	84.69	84.64	84.58
Naïve Bayes	68.52	67.09	66.74	86.09	85.86	68.55
MLP	85.60	86.00	85.89	85.57	85.49	84.15
Random forests						87.25
Our method	**89.16** (± 3.09)					

The prediction the performances of different classifiers over the Australian credit dataset. The table shows the classification accuracy of our method is much higher than these studies' one. Relying on parallel processing, time to run 20 trails with 5-fold cross validate taken by our method is only 2974 s (~50 min).

5.2 German Credit Dataset

The German credit approval dataset consists of 1000 loan applications, with 700 accepted and 300 rejected. Each applicant is described by 20 attributes. Our final results were averaged over these 20 independent trials. In our experiments, we use the default

value for the mtry parameter and the ntree parameter was tried with value of 100. The averages of classification results are depicted in Fig. 3.

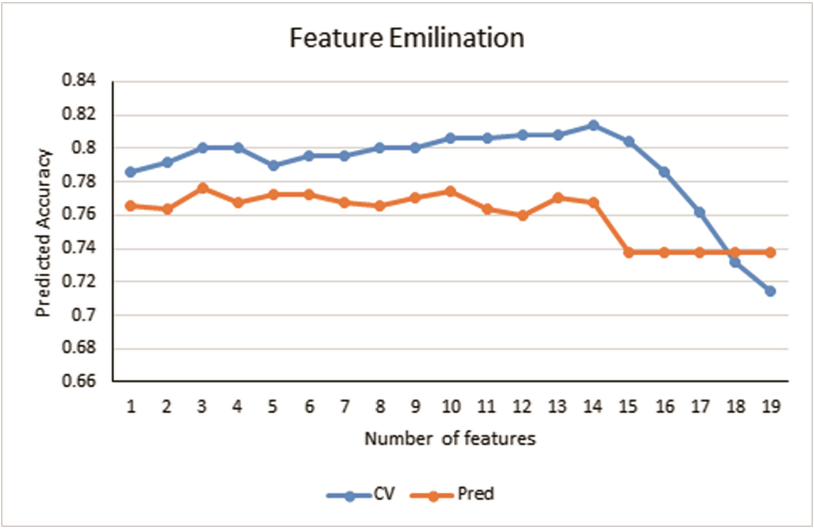

Fig. 3. Accuracy in case of German credit dataset

Table 2 shows the performances of different classifiers over the German credit datasets. Baseline is the classifier without feature selection. Classifiers used in [22] include: Linear SVM, CART, k-NN, Naïve Bayes, MLP. Filter methods include: t-test, Linear Discriminant analysis (LDA), Logistic regression (LR). The wrapper methods include: Genetic algorithms (GA) and Particle swarm optimization (PSO).

Table 2. Performances of different classifiers over the German credit dataset

Classifier	Filter methods			Wrapper methods		Baseline
	t-test	LDA	LR	GA	PSO	
Linear SVM	76.74	75.72	75.10	85.52	85.52	85.52
CART	74.28	73.52	73.66	84.85	84.82	85.20
k-NN	71.82	71.86	72.62	84.69	84.64	84.58
Naïve Bayes	72.40	70.88	71.44	86.09	85.86	68.55
MLP	73.28	73.44	73.42	85.57	85.49	84.15
Random forests						76.60
Our method	**78.95** (± 2.62)					

Moreover, relying on a parallel processing strategy, time to run 20 trails with 5-fold cross validate taken by our method is only 4311 s (~72 min) while other methods must run several hours. This result highlights the efficiency in terms of running time of our method when filtering the redundant features.

6 Conclusion

In this paper, we focused on studying feature selection and Random Forest method. Features selection involves in determining the highest classifier accuracy of a subset or seeking the acceptable accuracy of the smallest subset of features. We have introduced a new feature selection approach based on feature scoring. The accuracy of classifier using the selected features is better than other methods. Fewer features allow a credit department to concentrate on collecting relevant and essential variables. The parallel processing procedure leads to a significant decrement in runtime. As a result, the workload of credit evaluation personnel can be reduced, as they do not have to take into account a large number of features during the evaluation procedure, which will be somewhat less computationally intensive. The experimental results show that our method is effective in credit risk analysis. It makes the evaluation more quickly and increases the accuracy of the classification.

References

1. Altman, E.I., Saunders, A.: Credit risk measurement: developments over the last 20 years. J. Bank. Finance **21**(11–12), 1721–1742 (1997)
2. Davoodabadi, Z., Moeini, A.: Building customers' credit scoring models with combination of feature selection and decision tree algorithms **4**(2), 97–103 (2015)
3. Khashman, A.: A neural network model for credit risk evaluation. Int. J. Neural Syst. **19**(4), 285–294 (2009)
4. Bellotti, T., Crook, J.: Support vector machines for credit scoring and discovery of significant features. Expert Syst. Appl. **36**(2), 3302–3308 (2009)
5. Wen, F., Yang, X.: Skewness of return distribution and coefficient of risk premium. J. Syst. Sci. Complexity **22**(3), 360–371 (2009)
6. Zhou, X., Jiang, W., Shi, Y., Tian, Y.: Credit risk evaluation with kernel-based affine subspace nearest points learning method. Expert Syst. Appl. **38**(4), 4272–4279 (2011)
7. Kim, G., Wu, C., Lim, S., Kim, J.: Modified matrix splitting method for the support vector machine and its application to the credit classification of companies in Korea. Expert Syst. Appl. **39**(10), 8824–8834 (2012)
8. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers, Dordrecht (1998)
9. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. J. Mach. Learn. Res. **3**, 1157–1182 (2003)
10. Oreski, S., Oreski, D., Oreski, G.: Hybrid system with genetic algorithm and artificial neural networks and its application to retail credit risk assessment. Expert Syst. Appl. **39**(16), 12605–12617 (2012)
11. Saberi, M., Mirtalaie, M.S., Hussain, F.K., Azadeh, A., Hussain, O.K., Ashjari, B.: A granular computing-based approach to credit scoring modeling. Neurocomputing **122**, 100–115 (2013)
12. Lee, S., Choi, W.S.: A multi-industry bankruptcy prediction model using back-propagation neural network and multivariate discriminant analysis. Expert Syst. Appl. **40**(8), 2941–2946 (2013)
13. Ghatge, A.R., Halkarnikar, P.P.: Ensemble neural network strategy for predicting credit default evaluation **2**(7), 223–225 (2013)

14. Chaudhuri, A., De, K.: Fuzzy support vector machine for bankruptcy prediction. Appl. Soft Comput. J. **11**(2), 2472–2486 (2011)
15. Ghodselahi, A.: A hybrid support vector machine ensemble model for credit scoring. Int. J. Comput. Appl. **17**(5), 1–5 (2011)
16. Huang, L., Chen, C., Wang, J.: Credit scoring with a data mining approach based on support vector machines. Comput. J. Expert Syst. Appl. **33**(4), 847–856 (2007)
17. Eason, G., Li, S.T., Shiue, W., Huang, H.: The evaluation of consumer loans using support vector machines. Comput. J. Expert Syst. Appl. **30**(4), 772–782 (2006)
18. Martens, D., Baesens, B., Gestel, T., Vanthienen, J.: Comprehensible credit scoring models using rule extraction from support vector machines. Eur. Comput. J. Oper. Res. **183**(3), 1466–1476 (2007)
19. Wang, Y., Wang, S., Lai, K.: A new fuzzy support vector machine to evaluate credit risk. Comput. J. IEEE Trans. Fuzzy Syst. **13**(6), 25–29 (2005)
20. Oreski, S., Oreski, G.: Genetic algorithm-based heuristic for feature selection in credit risk assessment. Expert Syst. Appl. **41**(4), 2052–2064 (2014)
21. Ling, Y., Cao, Q.Y., Zhang, H.: Application of the PSO-SVM model for credit scoring. In: Proceedings of the 2011 7th International Conference on Computational Intelligent and Security, CIS 2011, pp. 47–51 (2011)
22. Liang, D., Tsai, C.-F., Wua, H.-T.: The effect of feature selection on financial distress prediction. Knowl. Based Syst. **73**, 289–297 (2015)