A random forest algorithm based on similarity measure and dynamic weighted voting

ZHAO Shu-xu, MA Qin-jing, LIU Li-jiao

(*School of Electronic and Information Engineering, Lanzhou Jiaotong Univercity, Lanzhou 730070, China*)

Abstract: The random forest model is universal and easy to understand, which is often used for classification and prediction. However, it uses non-selective integration and the majority rule to judge the final result, thus the difference between the decision trees in the model is ignored and the prediction accuracy of the model is reduced. Taking into consideration these defects, an improved random forest model based on confusion matrix (CM-RF) is proposed. The decision tree cluster is selectively constructed by the similarity measure in the process of constructing the model, and the result is output by using the dynamic weighted voting fusion method in the final voting session. Experiments show that the proposed CM-RF can reduce the impact of low-performance decision trees on the output result, thus improving the accuracy and generalization ability of random forest model.

Key words: random forest; confusion matrix; similarity measure; dynamic weighted voting

CLD number: TP312.8 **Document code**: A

Article ID: 1674-8042(2019)03-0277-08 **doi**: 10.3969/j.issn.1674-8042.2019.03.011

0 Introduction

The random forest algorithm is a hybrid model proposed by Leo Breiman from the University of California in 2001, which is also called the original random forest model[1-2]. The model can process discrete, continuous and mixed large data sets, has the ability to efficiently estimate out-of-bag (OOB) error and analyze feature importance. Especially, it is diffcult for this model to over-fit. As a typical representative of integrated learning algorithm, random forest is widely used in pattern recognition, text classification, product recommendation, etc. In order to carry out data mining and analysis more accurately and efficiently, people intend to optimize the algorithm by improving the correct rate of the output of random forest model.

Since the introduction of random forest, scholars have carried out subsequent research and optimization on this method. Kulkarni, et al. optimized the random forest by dividing data dimension into two parts, which improves the accuracy rate to some extent[3-4]. Oshiro, et al. proved mathematically that if a random forest selects different data sets for sample subset establishment during repeated sampling, a higher accuracy rate can be obtained[5]. Jian, et al. applied variable eigenvectors to the splitting of tree nodes, and the classification effect was improved compared with the structured random forest algorithm[6]. To some extent, Li, et al. improved the classification accuracy of the model by changing the number of training samples and not putting samples back[7]. Guo, et al. introduced the histogram of oriented gradient (HOG) multi-feature fusion method into random forests[8], and the results show that the classification accuracy rate of weighted random forest model is higher than that of the general random forest algorithm and traditional classification algorithm. These above improved methods optimize the random forest from different aspects, but there is still room for improvement.

In this paper, at first, the decision trees in the original random forest are screened. Then, the decision trees in the random forest are clustered by similarity measure. Finally, the output of the decision tree cluster is dynamically weighted, and the results are fused and output. In the experimental part, based on the actual desensitization data and the

Received date: 2019-06-18

Foundation items: Science Research Project of Gansu Provincial Transportation Department (No. 2017-012)

Corresponding author: MA Qin-jing (765324908@qq.com)

public dataset, comparing CM-RF with other improved random forests and some common hybrid models, it can be found that the proposed CM-RF has higher accuracy.

1 Random forest

1.1 Overview of algorithm

The random forest algorithm model is a mixed model composed of multiple decision trees $\{t(\boldsymbol{X},\boldsymbol{\Theta}_1), t(\boldsymbol{X},\boldsymbol{\Theta}_2),\cdots,t(\boldsymbol{X},\boldsymbol{\Theta}_k)\}$, where $\boldsymbol{\Theta}_i$ $(i=1,2,\cdots,k)$ respresents independent random variables with the same distribution. The result of the model is voted by all decision trees in the random forest and the final prediction output function is expressed as

$$T=\arg\max_{\boldsymbol{Y}}\sum_{i=1}^{k}F(t(\boldsymbol{X},\boldsymbol{\Theta}_i)=y_j), \tag{1}$$

where T represents the random foresst model; $\boldsymbol{X}=\mathbf{R}^n$ is the input vector; $t(\boldsymbol{X},\boldsymbol{\Theta}_i)$ is a single decision tree model; $\boldsymbol{Y}$ represents the output space; y_j is the input vector category label, $y_j\in\boldsymbol{Y}=\{y_1,y_2,\cdots,y_m\}$; and $F(\circ)$ is the indicator function. Eq. (1) shows that the random forest model uses a simple voting strategy obeying the majority rule to determine the final result. The algorithm of flow chart constructed by integrating all decision trees is shown in Fig. 1, which represents the model construction process.

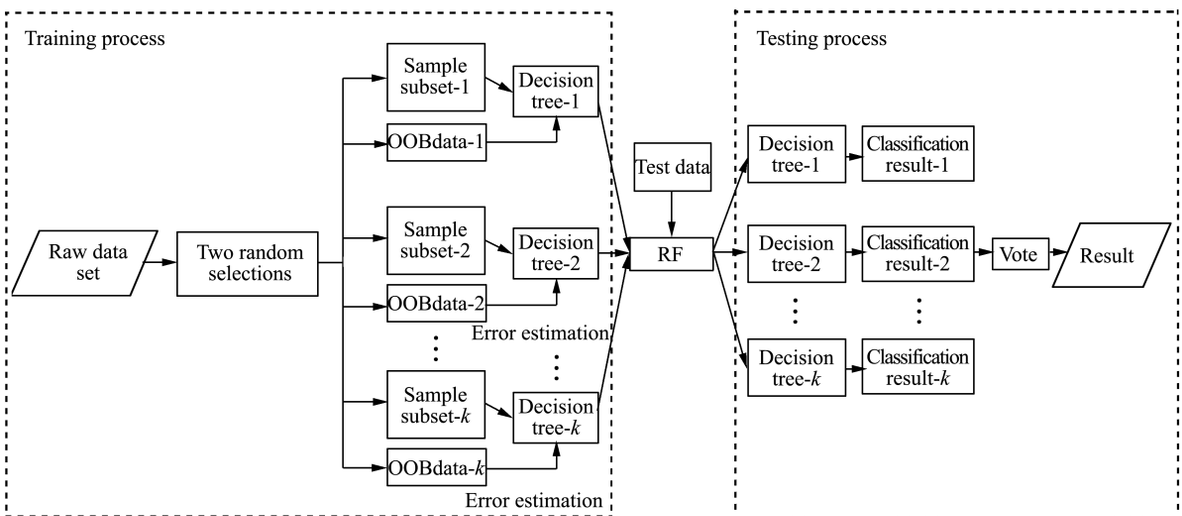

Fig. 1 Flow diagram of random forest algorithm

The specific implementation steps of the algorithm are as follows:

1) Given the original data set, $\boldsymbol{D}=\{(x_i,y_j)\,|\,x_i\in\boldsymbol{X},y_j\in\boldsymbol{Y}\}$, the number of samples is M, and the number of feature attributes is N. The bootstrap aggregating method (also called the Bagging method)[9] is used to randomly select M_t $(M_t=M)$ samples from the original data set to form a training sample subset. The unselected data are included in a test sample, also called OOB[10] data, for error estimation;

2) Based on the selected subset of training samples, the decision tree is constructed by the decision tree generation algorithm such as C4.5[11], CART[12], etc., and N_t $(N_t\ll N)$ feature attributes are used for complete splitting of the internal nodes without pruning, finally a decision tree is generate;

3) Repeating the above steps 1) and 2) for a total of $\boldsymbol{k}$ times, $\boldsymbol{k}$ decision trees are constructed to form a random forest model;

4) Using the test sample to test the random forest model, $\boldsymbol{k}$ decision trees generate $\boldsymbol{k}$ prediction results, and the final prediction results are calculated by a voting strategy obeying the majority rule.

1.2 Analysis of algorithm

As mentioned above, the random forest algorithm consists of three main steps. Firstly, a large number of decision trees are constructed. Secondly, these decision trees are integrated to form a random forest model. Finally, the input samples are voted and the prediction results are output.

For the random forest model, it usually adopts non-selective integration methods in the decision tree integration process, that is, all the constructed $\boldsymbol{k}$ decision trees are selected to form the random forest model. But when the training sample contains too much noise data, there may be too many noise trees

in the process of constructing a decision tree. Since the random forest model integrates all decision trees, the robustness of the model will be affected by these noise trees, which will reduce the prediction accuracy. In addition, when outputting the prediction results, the rule that the minority is subject to the majority is applied to the voting strategy. Although this voting strategy is simple and effective, it does not consider the difference in performance between the decision trees, which reduces the prediction accuracy of the model.

2 CM-RF model

Aiming at the shortcomings of random forest model in decision tree integration and predition accuracy, it is optimized based on the binary classification by selectivly structuring decision trees with similarity measure and dynamic weighted voting method. The basic block diagram of the optimization process is shown in Fig. 2 and the specific implementation method is given below.

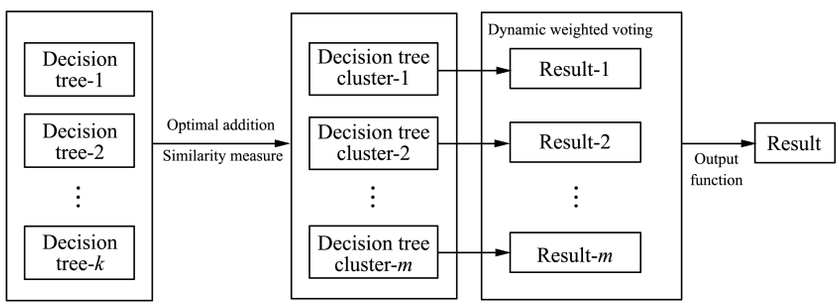

Fig. 2 Basic block diagram of model optimization based on similarity measure and dynamic weighted voting

2.1 Random forest optimization based on similarity measure

The binary classification is one of the most common classification prediction methods in reality. Its output space is $\boldsymbol{Y}=\{y_1, y_2\}$, where y_1 means positive samples and y_2 means negative samples. The error estimation confusion matrix[13] of the training data corresponding to the ith decision tree $t(\boldsymbol{X},\boldsymbol{\Theta}_i)$ in the random forest model $\{t(\boldsymbol{X},\boldsymbol{\Theta}_1), t(\boldsymbol{X},\boldsymbol{\Theta}_2), \cdots, t(\boldsymbol{X},\boldsymbol{\Theta}_k)\}$ is recorded as

$$\boldsymbol{C}_i=\begin{bmatrix} a_{tp}^i & a_{fn}^i \\ a_{fp}^i & a_{tn}^i \end{bmatrix}, \tag{2}$$

where a_{tp}^i and a_{tn}^i are the numbers that decision tree $t(\boldsymbol{X},\boldsymbol{\Theta}_i)$ can correctly identify class samples y_1 and y_2; a_{fn}^i and a_{fp}^i indicates the numbers of class samples y_1 and y_2 that are misjudged as y_2 and y_1.

Assuming that the total number of training samples is M, and dividing $\boldsymbol{C}_i$ by M, the meaning of each element in the matrix becomes its corresponding number of proportions, and then the confusion matrix after the transformation is recorded as

$$\boldsymbol{C}_{i/M}=\begin{bmatrix} a_{tp}^{i/M} & a_{fn}^{i/M} \\ a_{fp}^{i/M} & a_{tn}^{i/M} \end{bmatrix}. \tag{3}$$

It can be seen from the conclusion of Ref. [14] that the row vector of the confusion matrix represents the tendency of each predication result given when classifying the y-type sampes, which defines a measure of correlation between the decision trees and describes the degree of similarity between them. Firstly, the $\boldsymbol{C}_{i/M}$ is transformed into $[a_{tp}^{i/M} \quad a_{fn}^{i/M} \quad a_{fp}^{i/M} \quad a_{tn}^{i/M}]$. Since the size of each training sub set sampled by the Bagging method is still M, the confusion matrix deformed by k decision trees is combined into a measure matrix, which is recorded as

$$\boldsymbol{C}_{\mathrm{RF}}=\begin{bmatrix} a_{tp}^{1/M} & a_{fn}^{1/M} & a_{fp}^{1/M} & a_{tn}^{1/M} \\ a_{tp}^{2/M} & a_{fn}^{2/M} & a_{fp}^{2/M} & a_{tn}^{2/M} \\ \vdots & \vdots & \vdots & \vdots \\ a_{tp}^{k/M} & a_{fn}^{k/M} & a_{fp}^{k/M} & a_{tn}^{k/M} \end{bmatrix}. \tag{4}$$

With the measure matrix, the L2 measure is used to describe the similarity between different decision trees, and $\boldsymbol{C}_{\mathrm{RF}}$ can be calculated by L2 measure definition. The corresponding measure matrix $\boldsymbol{L}$ is a symmetric matrix whose main diagonal elements are all 0 and its size is $\boldsymbol{k}\times\boldsymbol{k}$. The corresponding relationship between the elements of $\boldsymbol{L}$ and $\boldsymbol{C}_{\mathrm{RF}}$ is

$$lm_{ij}=\begin{cases} 0, & i=j, \\ (a_{tp}^{i/M}-a_{tp}^{j/M})^2+(a_{fn}^{i/M}-a_{fn}^{j/M})^2+(a_{fp}^{i/M}-a_{fp}^{j/M})+(a_{tn}^{i/M}-a_{tn}^{j/M})^2, & 1\leqslant i\neq j\leqslant k. \end{cases} \tag{5}$$

Then normalization is performed as

$$lm_{n,ij} = \frac{lm_{ij}}{l_{\max}}, \tag{6}$$

where lm_{ij} represents the element in $\boldsymbol{L}$, $l_{\max} = \max(l_{ij})$, and subscripts i and j represent the positions of the element in $\boldsymbol{L}$. The smaller the value in $lm_{n,ij}$, the stronger the similarity between the two decision trees, otherwise the weaker.

2.2 Random forest optimization based on optimal addition

Before using the similarity measure to selectively integrate decision trees, in order to reduce the negative impact of noise trees on the model, the noise trees are eliminated as much as possible based on the principle of optimal addition. The method is shown in Fig. 3. The decision tree in the figure is referred as DTree, which is described as follows.

Assuming that the decision tree $\boldsymbol{t}$ is constructed, firstly, a decision tree with the smallest OOB error estimate is selected, and then the decision tree is combined with the remaining $\boldsymbol{t}-1$ decision trees to form $\boldsymbol{t}-1$ random forest sub-models and their respective OOB error measure estimates are calculated. The sub-model with the smallest error continues the next round of screening. Repeated iterations are performed. Eventually, $\boldsymbol{t}$ random forest sub-models are formed and their corresponding OOB error estimates are obtained, and the sub-model with the smallest OOB error estimate is taken as the initial model.

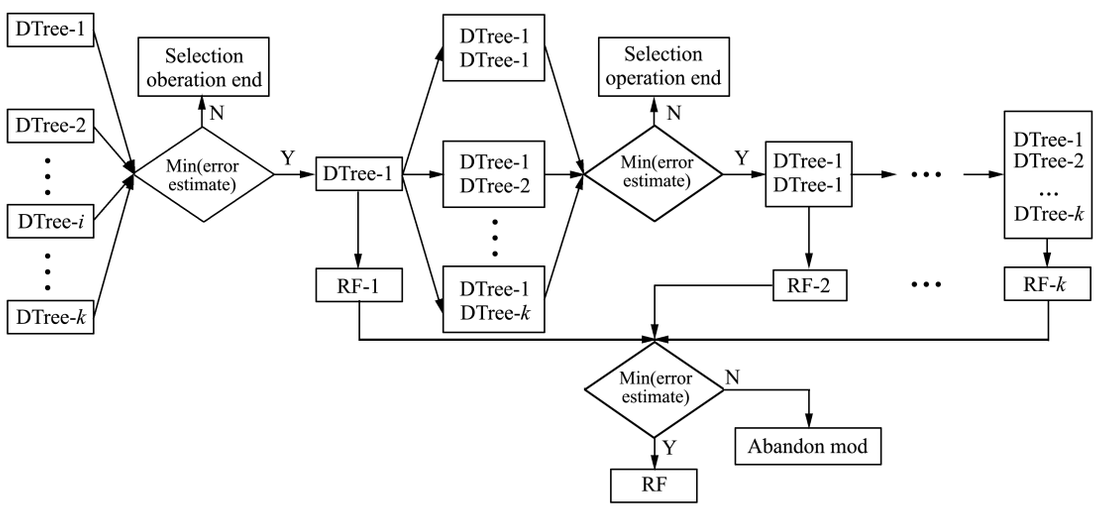

Fig. 3 Schematic diagram of a decision tree selection process based on principle of optimal addition

2.3 Decision tree cluster partitioning method based on similarity measure

In order to build decision tree clusters with large difference for the construction of random forest, we divide the decision tree set into subsets by using the optimal addition. The decision trees with strong correlation are taken as a decision tree cluster whereas these with weak correlation are included into different decision tree clusters. The specific implementation steps of the algorithm are as follows.

1) Given the training set $\boldsymbol{D}$, $\boldsymbol{k}$ decision trees are constructed by the Bagging method and the decision tree generation algorithm, then the confusion matrix $\boldsymbol{C}_{i/M}$ of each tree is obtained;

2) m decision trees are selected from $\boldsymbol{k}$ decision trees by optimal addition to form an initial random forest model;

3) Calculating the correlation measure matrix according to Eqs. (4)—(6), setting similar threshold $\boldsymbol{a}$ and distinct threshold $\boldsymbol{b}$, and comparing the relationship between the elements $\boldsymbol{a}$ and $\boldsymbol{b}$ and lm_{ij} in correlation measure matrix $\boldsymbol{C}_{\mathrm{RF}}$, the division of the decision tree cluster is determined. The division rules are as follows:

① If $lm_{n,ij} \leqslant a$, decision trees $\boldsymbol{i}$ and $\boldsymbol{j}$ are combined and recorded as combinable decision tree cluster $\boldsymbol{T}_m = \{t_i, t_j\}$;

② If $lm_{n,ij} \geqslant b$, decision trees $\boldsymbol{i}$ and $\boldsymbol{j}$ are recorded as uncombinable set $\boldsymbol{T}_m^{'} = \{t_i, t_j\}$;

③ For two combinable decision tree clusters $\boldsymbol{T}_{m1}$ and $\boldsymbol{T}_{m2}$, if $\boldsymbol{T}_{m1} \bigcap \boldsymbol{T}_{m2} \neq \varnothing$, they are merged;

④ When $a < lm_{n,ij} < b$, it can be seen that the decision trees $\boldsymbol{i}$ and $\boldsymbol{j}$ are between the related and uncorrelated relationship, thus the division needs to be determined according to the existing $\boldsymbol{T}_m$ and $\boldsymbol{T}_m^{'}$.

There are two cases:

Case 1 When $a<lm_{n,ij}<b$ as well as there being a combinable decision tree cluster $\boldsymbol{T}_m=\{t_i,t_k\}$ and a uncombinable set $\boldsymbol{T}_m^{'}=\{t_k,t_j\}$, the decision tree $\boldsymbol{j}$ cannot be added to $\boldsymbol{T}_m$;

Case 2 When $a<lm_{n,ij}<b$ as well as there being combinable decision tree clusters $\boldsymbol{T}_{m1}=\{t_j,t_k\}$ and $\boldsymbol{T}_{m2}=\{t_i,t_h\}$, combination can only be made on condition that both $lm_{n,hj}$ and $lm_{n,hk}$ are less than b.

2.4 Dynamic weighted voting

By calculating the prediction accuracy values of decision trees in the decision tree clusters divided by similarity measure, the average of the prediction accuracy values of all decision trees in the cluster is taken as the prediction accuracy values of the cluster. Assuming that the prediction accuracies of the ith decision tree cluster, the optimal decision tree cluster and the worst decision tree cluster are p_i, $p_{\max}$ and $p_{\min}$, respectively, the voting weight of the ith decision tree cluster composed of $\boldsymbol{n}$ decision trees in the random forest model can be calculated by

$$W_i=\begin{cases}\dfrac{100}{k}\displaystyle\sum_{i=1}^{k}p_i/\exp\left(\dfrac{p_{\max}-p_i}{p_{\max}-p_{\min}}\right), & p_{\max}\neq p_{\min},\\ 1, & p_{\max}=p_{\min}.\end{cases}\tag{7}$$

The dynamic adjustment of the weighting function in the above formula is based on the different prediction accuracy values of decision tree clusters in the random forest model. The higher the prediction accuracy, the larger the voting weight possessed by the decision tree cluster. Assuming that m decision tree clusters in random forest are constructed, the prediction output function is expressed as

$$T=\arg\max_{\boldsymbol{Y}}\sum_{i=1}^{m}W_iF(t_i(\boldsymbol{X},\boldsymbol{\Theta}_k)=y_j).\tag{8}$$

3 Experiments

3.1 Experimental data

In order to verify the validity of the proposed CM-RF algorithm, an empirical analysis was performed using desensitized credit data of a commercial bank (data time ranges from February 2016 to July 2017). Table 1 gives the specific content of A-Dataset and B-Dataset. According to the customers' loan information that whether the customers have overdue loans (in monthly), the customers are divided into two categories, namely normal customers and default customers. The CM-RF algorithm will validate the model performance by predicting whether the customers will default in the next three months (a default occurs in any of the next three months, which is considered a default). At the same time, in order to better illustrate the generalization ability of the model, the experiments are also carried out using the German credit dataset (G-Dataset) and sonar dataset (S-Dataset) in UCI's public dataset.

Table 1 Experimental data set

Dataset	Total number	Positive/negtive	Number of attributes	Lable
A-Dataset	30 000	21 924/8 076	22	normal/default
B-Dataset	20 000	16 364/3 636	22	normal/default
G-Dataset	1 000	371/629	20	normal/default
S-Dataset	208	111/97	60	metal/rock

Firstly, credit data are a common type of unbalanced data that need to be dealt with for balancing prior to modeling analysis. When the Bagging method is used to randomly extract the training sample subset to construct the decision tree, the ratio of the normal sample to the default sample is 1.5∶1, so as to achieve the purpose of balancing the data samples. Secondly, the data set needs to be normalized using Eq.(9) before modeling, and then all attribute values are converted into the values between [0,1].

$$\boldsymbol{s}_n=\frac{\boldsymbol{x}(i,j)-\min(\boldsymbol{x}(k,j))}{\max(\boldsymbol{x}(k,j))-\min(\boldsymbol{x}(k,j))}.\tag{9}$$

The sample number of the data set is n, where $x(i,j)$ represents the jth attribute value in the ith sample; $x(k,j)$, $k\in[1,n]$ represents the jth attribute of all the samples.

3.2 Evaluation criteria

Three metrics are selected as criteria for evaluating the model, including the model prediction average

accuracy(A_{ave}), the first type error rate(r_{err_1}) and the second type error rate(r_{err_2}), where r_{err_1} indicates that the proportion of the positive sample being misjudged as negative samples, and r_{err_2} indicates that the proportion of the negative sample being misjudged as positive samples[15]. According to the definition of confusion matrix in Eq.(2), the three metrics are calculated by

$$A_{ave} = \frac{a_{tp} + a_{tn}}{a_{tp} + a_{fn} + a_{fp} + a_{tn}}, \tag{10}$$

$$r_{err_1} = \frac{a_{fn}}{a_{tp} + a_{fn}}, \tag{11}$$

$$r_{err_2} = \frac{a_{fp}}{a_{tn} + a_{fp}}. \tag{12}$$

3.3 Experimental results and analysis

In the course of the experiment, the CM-RF model was implemented based on the PyCharm development platform using the Python language. In order to compare experiment results, the gradient boosted decision tree (GBDT)[16] and random forest model based on integrated non-return random sampling (SWR-RF)[7] were also implemented. In order to better evaluate the performance of the CM-RF model, the experimental model also includes random forest (RF)[1], logistic regression (LR)[17], convolutional neural network (CNN)[18] and the random subset SVM model (RS-SVM)[19].

In the experiment, the number of initial decision trees to construct the random forest model is 1 000. The functions in the hybrid model based on SVM algorithm are Lin, Poly and radial basis function (RBF), and the number of member classifiers is 15[20-21]. The correct rate of output on the four data sets are shown in Tables 2—5.

Table 2 Comparison of prediction results based on A-dataset

Model	Number of base classifiers	kernel	A_{ave}(%)	r_{err_1}(%)	r_{err_2}(%)
LR			83.71	15.82	17.26
CNN			86.12	12.44	14.02
GBDT			85.10	13.80	16.21
RF	1 000		84.76	14.78	15.71
SWR-RF	1 000		86.79	12.87	13.65
CM-RF	97		387.94	13.54	10.79
RS-SVM	15	Lin	85.13	15.05	14.44
		Poly	86.95	11.40	14.96
		RBF	86.77	11.31	13.07

Table 3 Comparison of prediction results based on B-dataset

Model	Number of base classifiers	kernel	A_{ave}(%)	r_{err_1}(%)	r_{err_2}(%)
LR			84.39	13.89	16.22
CNN			88.32	10.08	12.13
GBDT			86.88	12.24	14.77
RF	1 000		85.55	14.03	14.97
SWR-RF	1 000		88.93	10.47	12.25
CM-RF	988		89.16	9.84	11.90
RS-SVM	15	Lin	88.10	11.34	12.46
		Poly	87.05	11.67	13.41
		RBF	88.39	10.61	12.29

Table 4 Comparison of prediction results based on G-dataset

Model	Number of base classifiers	kernel	A_{ave}(%)	r_{err_1}(%)	r_{err_2}(%)
LR			85.31	13.22	15.61
CNN			89.32	10.11	11.04
GBDT			88.40	10.85	12.81
RF	1 000		86.91	12.78	13.49
SWR-RF	1 000		88.97	10.12	11.95
CM-RF	962		90.41	8.81	10.02
RS-SVM	15	Lin	89.70	9.71	10.97
		Poly	88.96	10.52	11.70
		RBF	91.07	8.72	10.65

Table 5 Comparison of classification results based on S-dataset

Model	Number of base classifiers	kernel	A_{ave}(%)	r_{err_1}(%)	r_{err_2}(%)
LR			85.89	13.58	14.90
CNN			89.04	9.75	11.97
GBDT			88.51	10.71	12.08
RF	1 000		87.40	12.11	13.07
SWR-RF	1 000		89.42	9.83	10.92
CM-RF	981		91.03	9.93	8.40
RS-SVM	15	Lin	89.10	10.37	11.73
		Poly	89.56	9.96	11.03
		RBF	88.07	11.20	12.57

The above tables give the experimental results of different models on the prediction and two-classification problem. According to the experimental results, the following conclusions can be drawn.

(1) The accuracy of the hybrid model is generally better than that of the single classifier model;

(2) Compared with the original random forest model, the CM-RF algorithm has improved the accuracy of result;

(3) The CM-RF model proposed in this paper

achieves the best results on A-Dataset, B-Dataset and S-Dataset. The optimal sub results are obtained on G-Dataset, and the result is slightly lower than the RS-SVM hybrid model with kernel function RBF, which shows the effectiveness of the improved random forest algorithm based on confusion matrix.

4 Conclusion

In this paper, we achieves good results that non-selective integration decision trees of the random forest model are optimized and improved according to the voting strategy that is subject to the majority rule. It can be seen from the experimental comparison that the CM-RF model is optimized, the scale of the original random forest model is reduced, and the influence of the noise tree on the model is eliminated, so that the output accuracy of themodel is improved. Based on different types of data, the optimization model CM-RF proposed in this paper can achieve higher accuracy, which is an effective model for data mining and analysis. Subsequent research will focus on the optimization and enhancement of the CM-RF model based on extremely unbalanced data.

The CM-RF model is optimized on the basis of the original random forest model, which increases the time complexity of the algorithm to a certain extent. The original random forest time complexity is $O(M(mn\log,n))$, where M is the number of decision trees in the random forest, m is the number of sample attributes and n is the number of samples. CM-RF increases time complexity using the optimal addition principle. The time complexity of the CM-RF model is $O(M'(mn\log,n))$.

References

[1] Breiman l. Random forests. Machine Learning, 2001, 45(1): 5-32.

[2] Ho t k. The random subspace method for constructing decision forests. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1998, 20(8): 832-844.

[3] Kulkarni V Y, Sinha P K. Efficient learning of random forest classifier using disjoint partitioning approach. Lecture Notes in Engineering & Computer Science, 2013, 2205(1): 1-5.

[4] Kulkarni V Y, Sinha P K. Pruning of random forest classifiers: a survey and future directions. In: Proceedings of International Conference on Data Science & Engineering, IEEE, Cochin, Kerala, India, 2012: 64-68.

[5] Oshiro T M, Perez P S, Baranauskas J A. How many trees in a random forest? In: Proceedings of International Workshop on Machine Learning and Data Mining in Pattern Recognition. Springer, Berlin, Heidelberg, 2012: 154-168.

[6] Jian C F, Chen J C, Zhang M Y. Improved random forest with S_Dbw based variable feature extraction operators. Journal of Chinese Mini-Micro Computer Systems, 2018, 39(2): 393-395.

[7] Li H, Li Z, She K. An improvement of random forests algorithm based on comprehensive sampling without replacement. Computer Engineering and Science, 2015, 37(7): 1233-1238.

[8] Guo J X, Chen W. Face recognition based on HOG multi-feature fusion and random forest. Computer Science, 2013, 40(10): 279-282.

[9] Breiman L. Bagging Preditors. Machine Learning, 1996, 24(2): 123-140.

[10] Breiman l. Out-of-bag estimation. 1996.

[11] Mu Y S, Liu X D, Yang Z H, et al. A parallel C4.5 decision tree algorithm based on MapReduce . Concurrency and Computation: Practice and Experience, 2017, 29(8): .

[12] Rutkowski L, Jaworski W, Pietruczuk L, et al. The CART decision tree for mining data streams. Information Sciences, 2014, 266: 1-15.

[13] Fawcett T . An introduction to ROC analysis. Pattern Recognition Letters, 2006, 27(8): 861-874.

[14] Bi K, Wang X D, Yao X, et al. Adaptively selective ensemble algorithm based on bagging and confusion matrix. Acta Electronica Sinica, 2014, 42(4): 711-716.

[15] Fawcett T. An introduction to ROC analysis. Pattern Recognition Letters, 2006, 27(8): 861-874.

[16] Duan D G, Gai X X, Han Z M, et al. Micro-blog misinformation detection based on gradient boost decision tree. Journal of Computer Application, 2018, 38(2): 410-414.

[17] Menard S. Applied logistic regression analysis. Technometrics , 2002, 38(2): 192-192.

[18] Girshick R. Fast R-CNN. Computer Science, 2015.

[19] Wang G, Ma J. A hybrid ensemble approach for enterprise credit risk assessment based on support vector machine. Expert Systems with Applications, 2012, 39(5): 5325-5331.

[20] Chen Y, Shi S, Pan Y, et al. Hybrid ensemble approach for credit risk assessment based on SVM. Computer Engineering and Application, 2016, 52(4): 115-120.

[21] Wang S J, Mathew A, Chen Y, et al. Empirical analysis of support vector machine ensemble classifiers. Expert Systems with Applications, 2009, 36(3): 6466-6476.

一种基于相似性度量和动态加权投票的随机森林算法

赵庶旭，马秦靖，刘李姣

（兰州交通大学 电子与信息工程学院，甘肃 兰州 730070）

摘　要： 随机森林模型易于理解，普适性强，常用于分类、预测等问题，但其采用无选择性集成和简单的少数服从多数投票原则进行最终结果判定，忽略了模型中各决策树之间的强弱差异，从而降低了模型的预测精度。针对该缺陷，提出了一种基于混淆矩阵的改进随机森林模型（Ramdom forest model based on confusion matrix，CM-RF）。在构建模型过程中通过相似性度量有选择性地构成决策树簇，并在最终投票环节使用动态加权投票融合方法进行结果输出。实验结果表明，该方法能减少低性能决策树对输出结果的影响，从而提升随机森林模型的正确率与泛化能力。

关键词： 随机森林；混淆矩阵；相似性度量；动态加权投票

引用格式： ZHAO Shu-xu，MA Qin-jing，LIU Li-jiao. A random forest algorithm based on similarity measure and dynamic weighted voting. Journal of Measurement Science and Instrumentation，2019，10(3)：277-284.［doi：10.3969/j.issn.1674-8042.2019.03.011］