MICROECONOMETRICS

The Performance of RandomForest on High-dimensional Data

WISE, XMU

Haihua XIE

27720181153991

Statistics

Dec. 22, 2019

1 Introduction

Random forest (RF) is a flexible machine learning algorithm, it has a wide range of application prospects. It can not only be used to model marketing simulation, statistics of customer sources, retention and loss, but also to predict the risk of disease and the susceptibility of patients.

Random forest is an algorithm that ensembles multiple decision trees through the idea of ensemble learning. Its basic unit is the decision tree, and its essence is a big branch of machine learning-ensemble learning.

Actually, we have known about the random forest and ensemble learning in the previous class, this project is to apply the random forest to analyze a real dataset ARCENE, which involves a binary classification problem. And this dataset is typically high-dimensional ($p >> n$), then I want to explore the performance of random forest in such a case.

2 Methodology

Random forest is a subclass of ensemble learning. It depends on the voting of decision tree to decide the final classification result (CART decision tree is used by default). For an input sample, N trees will produce N classification results. Random forest integrates all classification voting results, and specifies the category of the instance with the most voting times as the final output. It is a simple ensemble learning method, using the idea of bagging. Like the decision tree, random forest can be used for classification and regression. In this project we mainly focus on the classification.

Algorithm of the Random Forest
1. For a training sample with size N, we bootstrap N observations from the training sample (with replacement). And use this bootstrap sample to grow a decosion tree.
2. If the dimension of features of each sample is M, then we specify a constant m<<M, and randomly select the subset of size m from M features, and select the best one from the m features for each time the tree is split.
3. Each tree can grow as larger as possible without prunning.
4. Repeat step 1-3 for many times, and these decision trees form a random forest.
For each new instance, random forest is used to classify the new instance.
The result of classification depends on the number of votes of tree classifier.

Table 1: The algorithm of the Random Forest.

An advantage of random forest is that there is no need to do cross validation or use an independent test set to obtain an unbiased estimate of the error. It can be evaluated in the process of growing the tree, that is to say, an unbiased estimation of the error can be established during the growing process, which is called OOB error rate. For each tree, the prediction error on the out-of-bag portion of the data is recorded (error rate for classification, MSE for regression). Then the same is done after permuting each predictor variable. And the random forest can give the importance of the features. There are two measures of the importance. The first is the OOB error rate,The second measure is the total decrease in node impurities from splitting on the variable,averaged over all trees. For classification,the node impurity is measured by the Gini index. For regression,it is measured by residual sum of squares.

The disadvantage is that for some data sets with high-level noise, it is likely to be overfitting, Also, the tree-growing process is similar to a black box, with weak interpretability. And for very small sample or low-dimensional data, it may not produce good classification rsults.

3 Data

The dataset I choose is ARCENE, obtained from **UCI machine learning repository**[1]. The dataset ARCENE was obtained by merging three mass-spectrometry datasets to obtain enough training and test data for a benchmark. The original features indicate the abundance of proteins in human sera having a given mass value. Based on those features one must separate cancer patients from healthy patients. The order of the features and patterns were randomized.

ARCENE's task is to distinguish cancer versus normal patterns from mass-spectrometric data. This is a two-class classification problem with continuous input variables. This dataset is one of 5 datasets of the NIPS 2003 feature selection challenge.

The total number of features in the data set is 10000, and the numbers of samples in the training set and the validation set are both 100. Because the label of the test set is unknown, the test set is not used in this project. The validation set in the original data is mainly used to calculate the prediction error.

[1]https://archive.ics.uci.edu/ml/index.php

	positive	nagative	total number
training	44	56	100
validation	44	56	100

Table 2: The positive and negative examples in the training and validation set.

There is no missing value in this dataset, and the structure is complete. In data preprocessing, this project mainly deals with the data standardization, at the same time, I also remove the constant columns. For the features that have constant value over all instances, it can be considered that it has no effect on the training and prediction of the model, so it is reasonable to eliminate the constant columns, then the final number of total features is 9920.

4 Real Data Analysis

4.1 Random forest

Selecting the parameters

In a simple random forest model, the preprocessed training data are used as the training set to train the random forest model by using the default parameters. From figure 1, it can be found that when the number of decision trees is close to 500, the classification error rate tends to be stable, so in the following work, I set the number of decision trees to be 500.

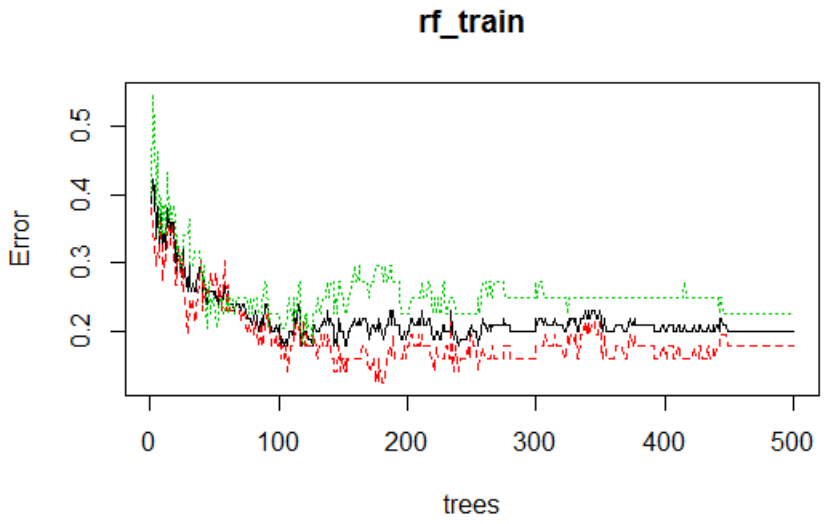


Figure 1: The number of decision trees and training error rate.

In addition to the number of decision trees, another important parameter in the random forest model is the number of candidate features that are randomly selected each time the decision tree is split. Empirically, the square root of the total number of features is selected. I search around the square root (100) of the total number of features (10000) with step size 5, that is from 50 to 150 with step size equals to 5. From figure 2, I have the result that the number of candidate features should be 75, it has the minimum classification error rate. So I choose 75 as the number of candidate features in each split.

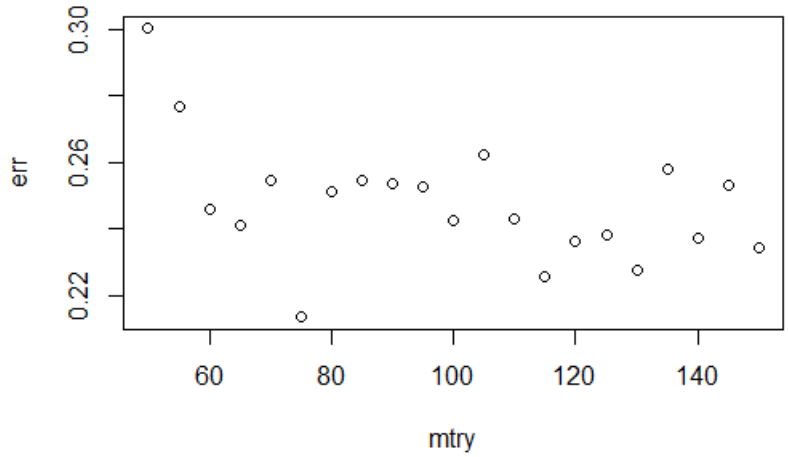


Figure 2: The number of candidate features for each split and training error rate.

Prediction on the validation dataset

In machine learning, confusion matrix is usually used to evaluate the performance of classifiers. It can easily show whether the predicted categories are confused. As shown in the following table, the classification error rate of the model is 22%, the precision rate is 78.3%, and the recall rate 83.9%.

		prediction		
		positive	negative	error
true	positive	47	9	0.1607143
	negative	13	31	0.2954545

Table 3: The confusion matrix of the random forest prediction on the validation dataset.

The importance of variables

After the parameters are determined, the training set can be put into the model to get the importance (mean confidence in Gini index) of features, as shown in the figure 3. Finally, we can remove the features whose importance is exactly equal to zero, which can do variable selection actually,. And I get new data sets (including new training sets and verification sets), which can be used for further model training and prediction, and finally The number of important features is 3046.

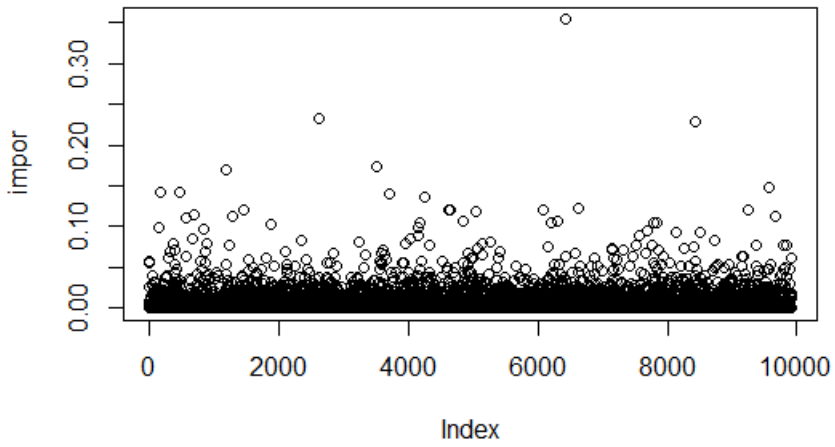


Figure 3: The importance of variables.

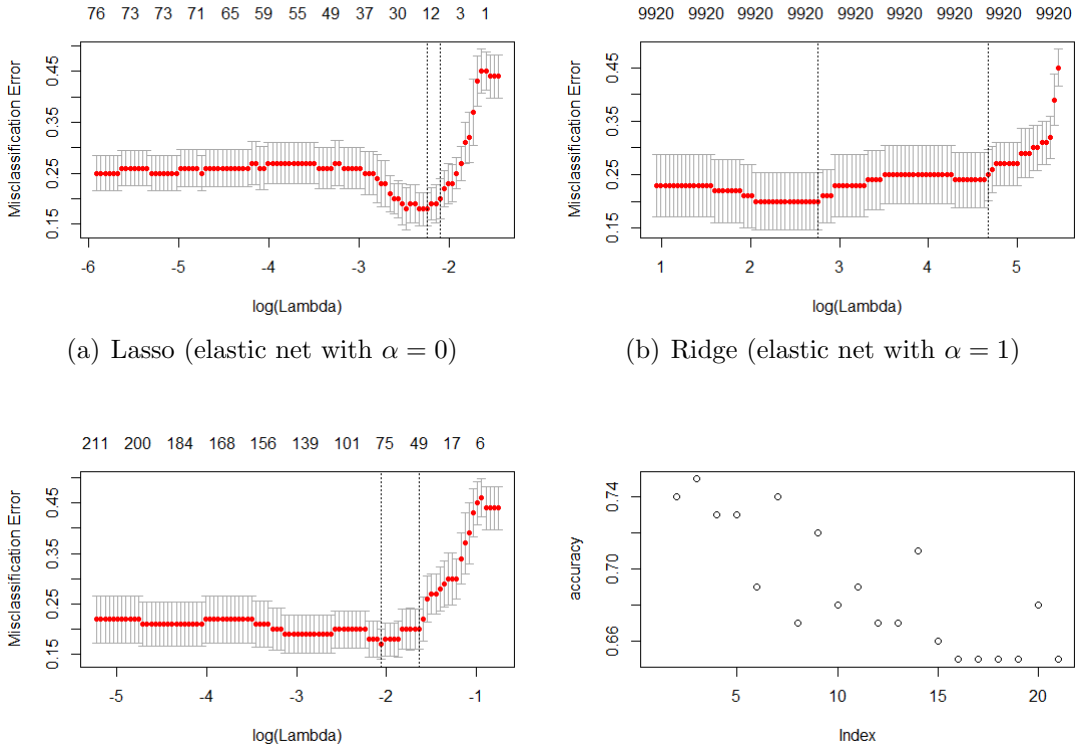
4.2 Comparison between random forest and penalized logistic regression

Lasso and elastic net are commonly used variable selection methods in high-dimensional case. The logistic regression with such penalties can do variable selection and classification simultaneously. The elastic network actually combines the characteristics of ridge regression and Lasso, and its form is as follows:

$$\underset{\beta}{\operatorname{argmin}} \left(\|y - X\beta\|^2 + (1-\alpha)\|\beta\|^2 + \alpha\|\beta\|_1 \right)$$

First, I try Lasso regression, ridge regression a elastic net regression with $\alpha = 0.5$, then the values of important parameters are selected by cross validation, which is represented in figure 4. The parameter lambda.1se is used to predict, the accuracy of lasso regression is 69%, and elastic net (α= 0.5) was 72%. In order to find a better valueof the parameter, this project takes a value between 0 and 1 with step size equal to 0.05. The corresponding prediction accuracy is shown in the following figure (lambda.1se is used for

prediction, that is, the simplest model is selected when the prediction error is as small as possible). Finally, the minimum prediction error is obtained when α= 0.1. And the best accuracy rate achieved by logistic regressing with elastic net is 75%, which is about 3% less than that of random forest.



(a) Lasso (elastic net with $\alpha = 0$)

(b) Ridge (elastic net with $\alpha = 1$)

(c) Elastic net with $\alpha = 1$

(d) Elastic net with different values of α

Figure 4: Time series plot and empirical Volatility of processes

5 Conclusion

As is shown in the following figure, when a single model is built directly, the performance of random forest is the best, the prediction accuracy is 78%, and followed by elastic network and lasso.

Model		Number of features	Accuracy of prediction
Random forest		9920	78%
Logistic reg.	Lasso	12	69%
	Elastic net	49	75%

Table 4: Comparison between random forest and penalized logistic regression

To sum up, for high-dimensional data, especially for this ARCENE dataset, the effect of random forest model is better than other logistic models with elastic net penalty. That is, random forest has better prediction accuracy and generalization ability in this case, and can effectively run on high-dimensional dataset. It is important to study the prediction and classification problems related to pathobiology (such data often have higher dimensions). However, when the variable is high, the interpretability of random forest is weak, and the combination of random forest and other models might be considered further.

Reference

[1] Breiman L. Random Forests[J]. Machine Learning, 2001, 45(1):5-32.

[2] Hui Z, Hastie T. Regularization and variable selection via the elastic net[J]. Journal of the Royal Statistical Society, 2005, 67(5):768-768.

[3] https://scikit-learn.org/stable/modules/ensemble.html

[4] https://www.stat.berkeley.edu/breiman/RandomForests/cc_home.htm#workings