Analysis of Classification Random Forests Model Based on C4.5 Decision Tree Learning Algorithm

Machine Learning

Zhaokai Huang

Abstract

Random forests are a set of decision trees, which combine the results predicted by each tree. Due to the practical use and values of classification people face today, I am strongly motivated to implement classification random forests that each decision tree of the model is created by adjusted C4.5 algorithm rather ID3 algorithm and test the model with UCI machine learning repository (UCI, 2013) to get the insight that RF (Random Forest) performs well in classifying data sets. In this report, I identify C4.5 as a extension of ID3 and specify the methods and techniques to build an accurate decision tree with regards to splitting criteria, dealing with continuous attributes and pruning trees. Then, I highlight how bagging is applied in building random forest - a randomized forest of decision trees and compare the model with other classifiers. In addition, I present experimental evidence to prove the claimed results and analyse the results with deep understanding and explanation. At last part of the report, I make a conclusion of important points about random forests and expect future potential work meanwhile.

1. Introduction

1.1 Data Classification

Classification of data sets plays a indispensable role in the fields of data mining and machine learning, which is a popular research topic. As people employ classification to many practical uses in daily life such as diagnosing patients' symptoms and judging the income of specific range of persons, the classifier models undoubtedly become the most important part of classification although the quality of data cannot be overlooked (Cherkassky, 2007). The classification consists of two basic learning tasks: training and testing. Specifically speaking, the first task is to build a classification model by using training data (usually historical records) with given classes or labels and the second one is to exploit derived rules to predict the labels for unseen testing data while sometimes both training data and testing data have missing values (Zhang, et al., 2015). Different classifiers make different performance in assigning a label to unseen samples.

1.2 Random Forests

Random forests (Breiman, 2001), a relatively new machine learning approach, seemed to conduct better performance than most classification models in terms of its low error rate in data classification and strong theory basis. It is a collection of decision trees (Ho, 1998), applying bagging tips to each creating procedure of decision tree, which means randomly selecting features and samples as the input parameters. Also, the number of trees really makes a big difference to the result. For a test sample without given label, it gets a response from each tree and take a majority vote for the final result.

1.3 ID3 Algorithm

In addition, decision tree is the model consists of branch nodes and leaf nodes, which represents choices between several intervals and decisions respectively. It is commonly used for the purpose of making precise decisions and there are three types of decision trees: classification tree, regression tree and CART (classification and regression tree, i.e., CART is an aggregation of classification tree and regression tree). ID3 (Iterative Dichotomiser 3) is the original and typical decision tree learning algorithm proposed by Ross Quinlan in 1975, which is frequently uses in areas of machine learning and natural language processing. The principle of this classification algorithm is entropy theory, a mathematical term belongs to information theorem, referring to the uncertainty of information. Decision tree models implemented in ID3 split attributes by using information gain which is negatively relevant with information entropy.

1.4 C4.5 Algorithm

Since there are some shortcomings of ID3, C4.5 (Quinlan, 1993) is presented as the improved version of ID3, extending all advantages of it and adopting some improvement and complement. The big difference is C4.5

algorithm uses information gain ratio as the criteria to split attributes and it can sole with continuous attributes. Moreover, C4.5 algorithm can take different strategies to deal with samples have missing values, which occurs often in real world. The major shortcoming of C4.5 is its low efficiency of algorithm in scanning and sorting data repeatedly. Although the performance of C4.5 has been exceeded by C5.0, it is still the commonly used decision tree algorithm because it is free and available.

2. Background

In this section, I identify the original decision tree learning algorithm ID3 and go into more technical details of C4.5 algorithm that overcome the shortcomings ID3 has. Moreover, I specify how bagging is applied in the creation of random forests and principles of it.

2.1 Optimisation of C4.5 compared with ID3

The decision tree implemented in ID3 algorithm is basically exploiting a top-down and greedy search through the given training data sets to generate the model. Then it predicts the label of unseen testing data sets by matching attributes with decision tree nodes. For the purpose of classifying precisely, ID3 takes the strategy - information gain to split the attribute that is most related and useful. It is the recursive algorithm to construct the decision tree automatically that continue to conduct on each subset of training data and this only stops in defined cases. The pseudo-code of ID3 algorithm are as below:

ID3 Decision Tree Learning Algorithm

```
1: function BuildTree(subsample, depth)
2:
3:  //Base Case
4:  if depth==0 OR labels are all same
5:      return most common labels in the subsample
6:  end if
7:
8:  //Recursive Case
9:  for each attribute
10:     Calculate information gain
11: end for
12: Pick attribute has maximum information gain
13:
14: Find left/right subset of subsample
15: BuildTree(leftSubsample, depth-1)
16: Add it to left branch
17: BuildTree(rightSubsample, depth-1)
18: Add it to right branch
19: return tree
20:
21: end function
```

C4.5 primarily makes improvement of four aspects: handling both continuous and discrete attributes, replacing information gain with information gain ratio as the splitting criteria, handling data with missing attribute values and pruning decision trees after creation. As for continuous attributes, C4.5 algorithm at first sorts the values and calculates every information gain of split point, whose value is average between two point values. And there is optimisation to calculate information gain, subtracting each information gain with $\log_2(N-1)/|D|$ (Quinlan, 1996), i.e., N-1 is the number of possible thresholds and |D| is the size of data set. Then select the split point that has maximum information gain as the threshold and calculate corresponding information gain ratio. With respect to information gain ratio, it is a extension to information gain, which overcomes the cons of information gain and employs a kind of normalisation to information gain by dividing it by split information value, whose value relies on the number of values that the attribute has and their distribution. Gain ratio can get rid of becoming more likely to select attributes have more number of values. While handling with attributes have missing values, C4.5 assign the most common values of the attribute to it. The better strategy is to assign possibilities of each value to the attributes without values. Decision trees are easy to fit training data very well, which fail to predict on testing samples. C4.5 usually prunes trees after its creation by replacing sub trees with leaf node if the accuracy is close although it can also stop growing the trees earlier by setting depth of trees.

2.2 Bagging and Principles of Random Forests

The bagging algorithm makes the models a bit difference by training them with slightly different data sets (Dietterich, 1998). For a training set T of size N, it randomly select N examples with replacement as new training data sets, which means picking one instance and putting it back to the original instances. This data sampling technique is regarded as a bootstrap and the new training data set is usually called bootstrap example T'.

Random forest is a forest of randomized decision trees (i.e., here it is implemented in C4.5), applying main ideas of bagging and a random selection of attributes at each split point to guarantee its randomisation (Amit, et al., 1997). Building a decision tree is using a bootstrap sample T' from T and a random partition of attributes at each split point as new subset. Due to the characteristics of random forests, pruning each decision tree is not really

necessary. Therefore, it returns a combination of trees. For a test point, it is classified by taking a majority vote of the response from each tree.

3. Experiments

Basically, I implement C4.5 decision tree learning algorithm in Java, which is specialised in classifying sample. And then I construct random forests model via two mechanisms of ensuring randomisation, which is a set of decision trees based on C4.5. The training data sets and testing data sets are downloaded from UCI machine learning repository, which are used for generating the model and evaluating the performance of model respectively. The experiment is about verifying whether C4.5 algorithm does better in classifying data sets than ID3 algorithm and the optimisation of C4.5 really makes a big difference to classification accuracy. Besides, it also involves estimating the efficiency and performance of random forests primarily in terms of error rate and time used to train and test.

At first, I justify whether the decision tree models implemented by adjusted can deal with both continuous and discrete features very well by training the model with play tennis data set which has temperature and humidity as the continuous attributes and test the model with testing sample. The structure of decision tree is presented in figure 1.

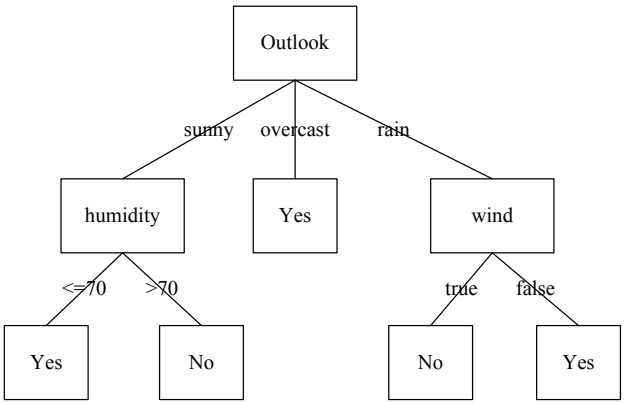

Figure 1. Structure of decision tree.

I notice information gain ratio is a better choice to split features in terms of information relevance based on information theorem than information gain ratio. I use voteall.txt downloaded from UCI machine learning repository as the source of data sets and split it into 5 folds for performing cross validation. Therefore, I train decision tree model (i.e., basically implemented in C4.5) that applies information gain ratio with vote train data set and test it with vote test data set five times and then get the final cross-validation error rate. I repeat these procedures for decision tree model that applies information gain. At last, I get the comparative results and show them in figure 2 and figure 3. The numbers in figure 2 and figure 3 have similar tendency and error rate of C4.5 decision tree model with information gain ratio to split features does not implicitly defeat that of C4.5 decision tree model with information gain

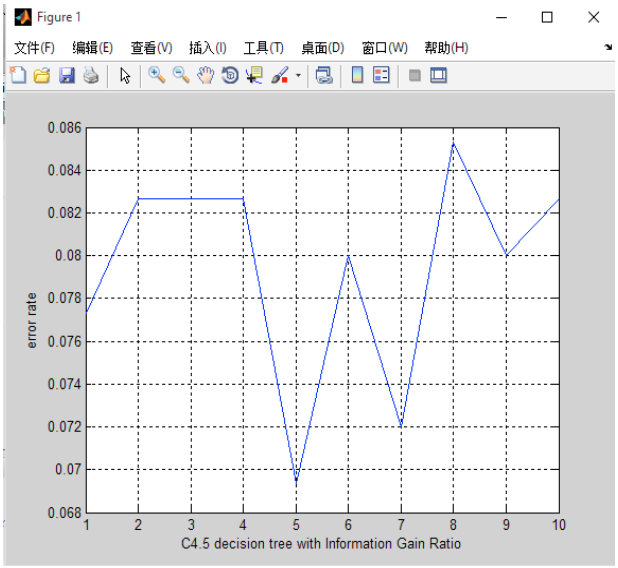

Figure 2. Error rate of C4.5 decision tree model with information gain ratio.

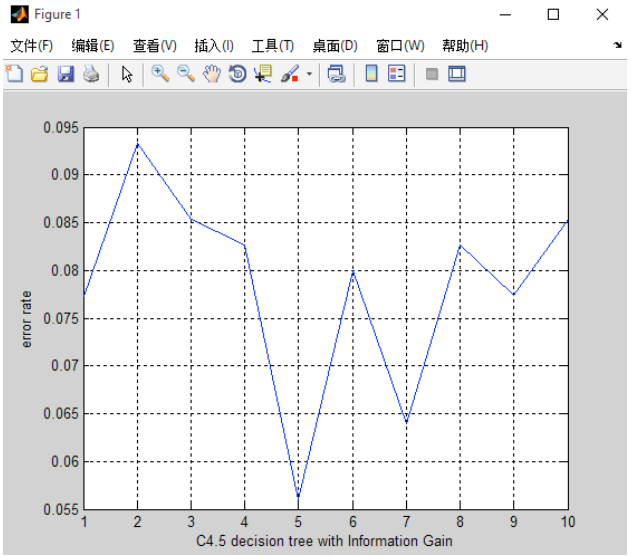

Figure 3. Error rate of C4.5 decision tree model with information gain.

I compare the performance of C4.5 decision tree model with that of random forests in terms of classifying instances correctly. I conduct the training and testing tasks on vote data sets, shuffling the data each time. The major criteria of how well each model perform is error rate and their standard deviation. The results are showed in figure 4 and figure 5.

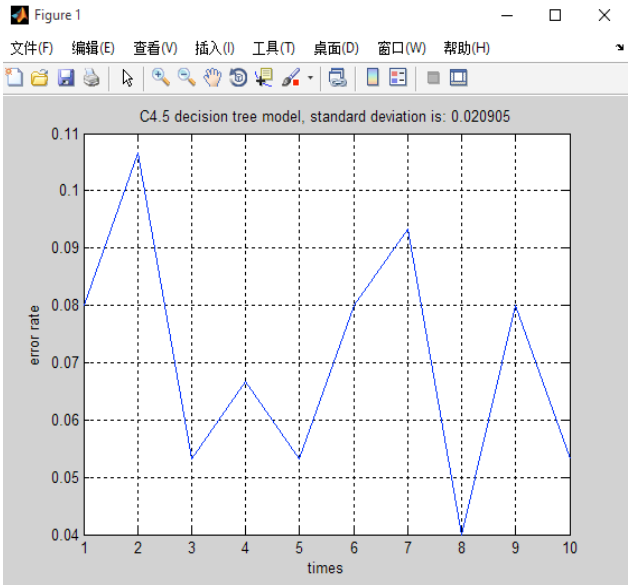

Figure 4. Error rate and standard deviation of C4.5 decision tree model.

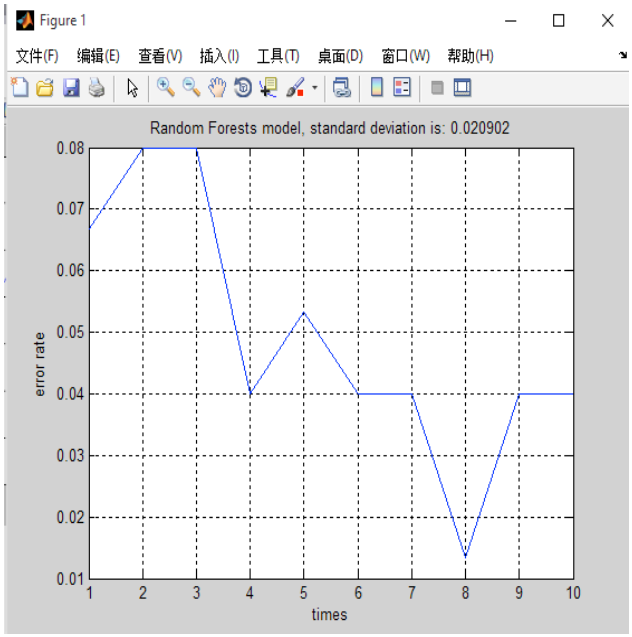

Figure 5. Error rate and standard deviation of random forests model.

Random forests and C4.5 decision tree implemented in this report are the classifiers that output only a class label. I obtain the values of confusion matrix by respectively training each mode with adult.data and testing it with adult.test. Two confusion matrix and common performance matrix are as what table1 and table 2 show. ROC graphs are two-dimensional graphs which consist of fp rate as the parameter on the X axis and tp rate as the parameter on the Y axis. The tp rate is defined as: True Positive/(True Positive+False Negative); and the fp rate is defined as: False Positive/(False Positive+True Negative). I present these two classifiers in the form of ROC graph, the results are showed as figure 6. As for random forests, the tp rate is 96/(96+3)=0.97 and the fp rate is 17/(17+9)=0.654; and for C4.5 decision tree model, the tp rate is 94/(94+5)=0.949 and the fp rate is 19/(19+7)=0.731. The data points that represent specific classifier are important to identify.

Table 1. Confusion matrix and common performance matrix of Random Forests

Hypothesized/True class	P	N
Y	True Positive96	False Positive17
N	False Negative3	True Negative9

Table 2. Confusion matrix and common performance matrix of C4.5 decision tree model

Hypothesized/True class	P	N
Y	True Positive94	False Positive19
N	False Negative5	True Negative7

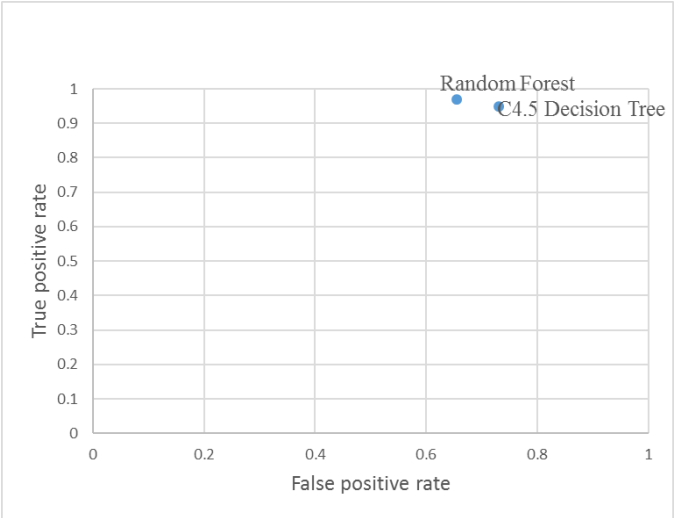

Figure 6. A basic ROC graph showing two discrete classifies: classification random forests and C4.5 decision tree.

The number of trees that a random forest has generally influence the final voting outcome, which means affecting classification accuracy to some extent. Accompanied with increasing number of trees, the tasks of training and testing the model become such time-consuming correspondingly. While increasing number of decision trees in the forest, the error rate of each specific forest is

showed in figure 7, and the cost time is presented in figure 8. It is clearly seen from figure 7 that although number of trees rise, the error does not make any significant difference.

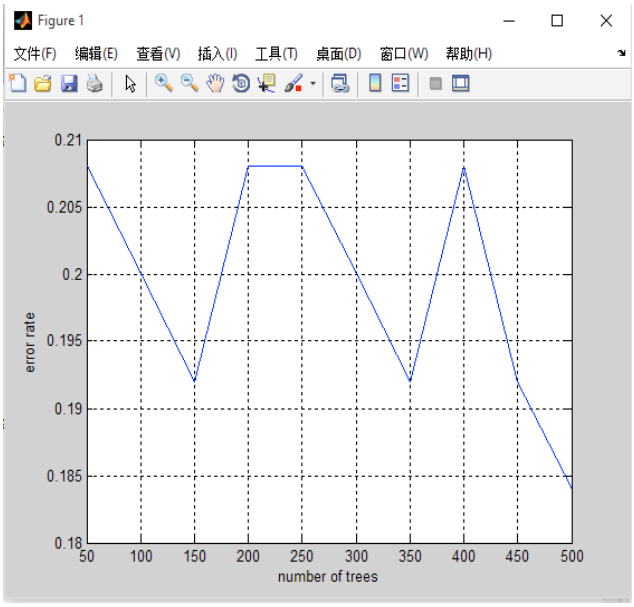

Figure 7. The error rate of each random forest according to its different numbers of trees.

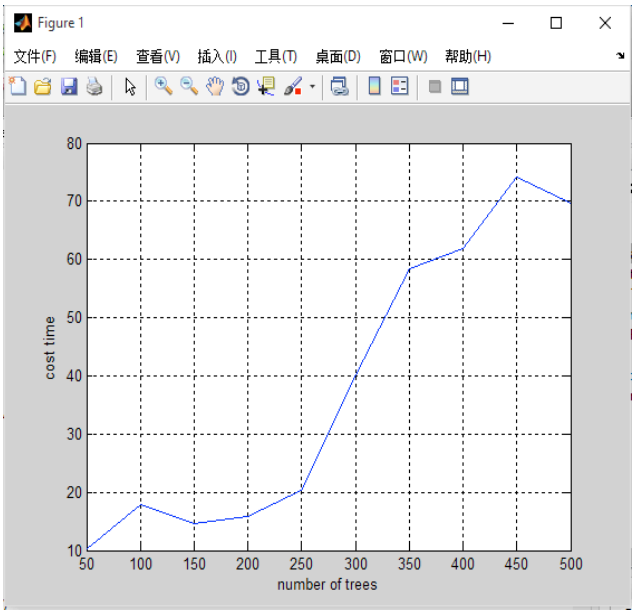

Figure 8. The time (in seconds) used to train and test each specific random forest according to its different numbers of trees.

4. Analysis

Random forest is an aggregation of various types of techniques such as bagging, C4.5 decision tree learning algorithm, information theory and so on.

Seeing from the results in figure 1, classification random forests model based on C4.5 learning algorithm can deal with continuous features very well although the play tennis data set is very simple. Because in this report, while handling with values of attributes, I use regular expression to find whether it is numeric and nominal and then select best threshold that has maximum information gain.

In addition, the tendencies and error rates in figure 2 and figure 3 are quite similar, even the C4.5 decision tree model that use information gain ratio to split attributes does not perform better than the one uses information gain sometimes. To some extent, it contradicts with the research achievements that information gain ratio is the better choice to split attributes. However, never forget information gain ratio is specifically aiming at continuous attributes have numeric values. With regards to that attribute, I employ the ideas of charging increased cost in terms of testing on a continuous attribute by subtracting information gain with $\log_2(N-1)/|D|$. It also the factor leads to the final situation that a vast majority of vote data sets is discrete or nominal.

The results in figure 4 and figure 5 are produced by performing 5 folds cross-validation several times. Random forests nearly perform better than C4.5 decision trees all the time according to the average error rate. Random forests grow an ensemble of decision trees and let them vote for the most common class label, which result in more precise classification (Breiman, 2001). Furthermore, the numbers of error rates are small and very close, leading to the values of standard deviation in the figures not significant.

One point in ROC space does better in classifying data sets than another point if it is to the northwest, which means having higher tp rate and lower fp rate. Seeing from the figure 6, these two classifiers all appear on the left-hand side of the ROC graph and random forests are regarded as better classifier, having higher tp rate and lower fp rate. As C4.5 decision tree is slightly nearer to the X axis, it is more conservative than random forests because it requires strong evidence to make positive classifications.

In figure 7, classification accuracy still remain at around 80%. The increasing number of trees does not help to make classifications more precisely. And the error rate is higher then the normal level, which is caused by that the training data set and testing data set have such number of missing values. C4.5 learning algorithm comes up the solution with assigning them common values of the attribute they belong to. Additionally, in figure 8, the cost time generally rises very fast with the increase of number of decision trees that random forest possess, since the creation procedures of trees propose questions of low efficiency.

5. Conclusion

In conclusion, random forest is a better classifier and performs more effectively in predicting class labels. Injecting two strategies: a bootstrap and a random selection of attributes at each split point to ensure randomness makes it an accurate model in classification. Needless to talk, random forest regards the majority vote of class label by each decision trees as the predictive outcome. And the C4.5 learning algorithm is an extension of ID3 algorithm, used for implementing decision trees, which mainly makes some adjustments to four aspects: C4.5 algorithm adopts information gain ratio to split attributes; it can sole with continuous attributes; moreover, C4.5 algorithm can deal with examples have missing values while during the period of training the model. In the part of experiments, I evaluate the performance and accuracy of random forest compared with C4.5 decision tree with statistical tests. Concluding from all the theoretical foundation and experimental evidence, the random forest is a good classifier.

However, there are some weaknesses of this model. The procedure of building random forest is computationally expensive, costs a lot of time and is easy to over-fit training data. The next potential research focus is to improve efficiency of random forest and the post-pruning sub trees.

Reference

Amit, Y. & Geman, D. (1997). Shape quantization and recognition with randomized trees. Neural Computation, 9, 1545-1588.

Breiman, L. (1996a). Bagging predictors. *Machine Learning,* 26(2): 123-140.

Breiman, L. (2001). Random Forests. *Machine Learning,* 45, 5-32.

Cherkassky, V., & Mulier, F. M. (2007). Learning from Data: Concepts, Theory, and Methods. John Wiley and Sons, IEEE Press.

Dietterich, T. (1998). An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting and randomization, *Machine Learning,* 1-22.

Ho, T. K. (1998). The random subspace method for constructing decision forests. IEEE Trans. on Pattern Analysis and Machine Intelligence, 20(8), 832-844.

Quinlan, J. R. (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers Inc.

Quinlan, J. R. (1996). Improved Use of Continuous Attributes in C4.5. Journal of Artificial Intelligence Research, 4, 77-90.

UCI (2013). Centre for machine learning and intelligent systems. Retrieved from http://archive.ics.uci.edu/ml

Zhang, S., Sadaoui, S., and Mouhoub, M. (2015) An Empirical Analysis of Imbalanced Data Classification. *Computer and Information Science,* 8(1): 151-162.