Decision Forests

Andrei Karpau
Institute of Technology Blanchardstown
Dublin 16

November 20, 2016

Abstract

Decision forest is the efficient and popular ensemble model based on the decision trees. This paper takes a look into the structure, main features and the advantages of this classificator. Also, it makes a brief overview of some famous techniques for building the forest and describes in detail the random forest technique.

Keywords: decision forests, random forests, data mining, classification algorithms, supervised learning

1 Introduction

A decision tree is a simple and widely used predictive model that is often applied for solving real world tasks. Quinlan (Quinlan, 1987, p. 304) states that it is a recursive process in which a case, described by the attributes, is assigned to one of a disjoined set of classes, that are the tree leaves. So, the decision tree is tree-like model where each branch represents a possible decision and each leaf shows its outcome. This helps people to visually see and weight the possible actions, risks and profits. Decision trees have also some other advantages. Among them are intuitiveness of the idea, straight-forward testing algorithms and the quickness of the classification (Ho, 1995, p. 278). There are many different learning algorithms for building decision tree, probably most popular are CART and C4.5. They both work from top to down and use splitting criterion (Rokach, 2016, p.112). However, there is the high variance problem that often appears by decision tree, which means that the model too precisely represents the training set and that leads to overfitting. This problem occurs because decision trees create a graph of choices so that the separate decision rules can be created also for non-general cases, noise and outliers. Additionally, the testing of one variable on each phase creates the axis-parallel rectangular regions that can have high bias (Dietterich, 2002, p. 7). This means that some cases are not considered by the model, which leads to underfitting and occurs when some classes dominate in the training set.

There are some techniques such as pruning that help to struggle with the overfitting of the decisions trees. However, they do not overcome the fundamental limitation trees should not be grown too complex to overfit the training data (Ho, 1995, p. 278). Kam Ho (1998, p. 1) argues that building the combination of multiple trees constructed in randomly selected subspaces can achieve the increase in generalization accuracy while preserving the perfect accuracy on training data and hence to overcome the high variance problem. Moreover, it does not depend on the algorithms used for building the trees. Actually, this model is the ensemble of trees that is named decision forest. This paper aims to provide the overview of this concept and algorithms for building it as well as describe one of them in detail.

1.1 Contribution and Structure

This paper addresses the decision forest model and makes the overview of it. Then it considers the advantages of using the decision forest and the main methods that are used for building them. Finally, it describes in detail one of the popular algorithms. This paper is structured as follow: in Section 1 is the introduction part; Section 2 provides the overview of the model with the main principles of how it works; Section 3 takes a look into the algorithms for building decision forests; and the last section 4 makes the final conclusions.

2 Decision forests

A decision forest is a collection of decision trees together with a decision combination function (Ho, 2002, p.102). This function returns the final result of the forest based on the results of each decision tree. This makes the final prediction more accurate especially if the outputs of the trees are independent. There are several reasons why decision forests and in general ensemble methods tend to provide better performance (Polikar, 2006, pp. 22-23):

1. From the statistical point of view even if a decision tree is accurate on test data set it can perform worse on the real samples. Especially, this is the case if the test data is not fully representative. Thus, the use of average result provided by several base models does not necessarily beat every single tree but reduces the risk of making the poor selection.

2. In cases of very big training sets this is not effective to build single model such as decision trees on the whole data. Instead, it is more practical to divide the training set, create a separate tree for each part of the data and then build forest on top of them.

3. In cases of small training sets, the resampling methods can be used. So, many decision trees are created and combined into the forest. It helps to improve the overall model accuracy.

4. Decision trees can be not flexible enough if the data needs non-linear and not axis parallel choices. Ensembles improve the accuracy in such cases.

5. Ensemble methods such as decision forests help to deal with data fusion problem. It takes place when the data set is collected from different sources that provide different attributes. Thus, it is inefficient to build one decision tree for the whole data set. Instead, separate base model can be created for each source and the ensemble of these models can be used.

Even though the decision forest is not such an intuitive classifier as decision tree, it is still an easy solution in comparison with many other data mining techniques. Moreover, since it is an ensemble model, it can be often implemented in parallel way so that each base decision tree is built in the separate thread and the results are aggregated at the end. It is very valuable when working in the cloud with the vast amounts of data.

3 Algorithms for building decision forests

3.1 Overview of popular algorithms

Since the mid-1990s plenty of different algorithms for building decision forests were presented. Generally, they can be divided into two main categories (Rokach, 2016, p. 118): using the general ensemble methods such as adaptive boosting, which can work with any base learning method, including the decision trees and the algorithms that were constructed individually for decision forests as an example random forests. The last one is one of the oldest and the most popular methods for building decision forests. Its first version was introduced by Ho (1995, p.278) and then was modified by Breiman (2001, p. 5). The final version of random forests uses bagging together with the random selection of features. This algorithm is more precisely discussed in subsection 3.2.
There are some other techniques that are widely used for building decision forests. Among them are bagging and boosting. Bagging (bootstrap aggregation) (Breiman, 1996, p. 123, pp. 127-128) in contest of decision forests is a method for generating multiple data sets from the original data by sampling with replacement. Then the separate tree is created for each generated data set and finally they are aggregated into the ensemble. This is very effective and simple algorithm that is used not only for building decision forests but also for improving accuracy of other data mining techniques. Boosting method iteratively creates decision trees on data that is taken from different distributions and then adds them into the strong composite forest (Rokach, 2016, p. 120). After adding the classifiers, the data is reweighted in order to focus on misclassified samples.
There are many other different algorithms for building the decision trees. However, the random forests tend to be one of the most popular and the most efficient. The recent study made by Fernndez-Delgado et al. (2014, p. 3175) compared 179 classifiers from 17 families of data mining techniques over 121

data set. Among them 14 variants of decision trees, 20 boosting classifiers, 24 types of bagging techniques, 8 random forests implementations and 11 other ensemble methods. The best results were achieved by random forest and, particularly, the parallel random forest from the R library that overcame the results of all the other classifiers.

3.2 Random Forests

Random forest is the ensemble learning method, that provides the averaging of multiple deep and unpruned decision trees trained on the different parts of the data set. The main steps of the algorithm are the following (Friedman et al., 2001, p. 588):

1. Repeat from b = 1 to B, where B is the number of decision trees in the Random Forest being built.
   (a) Draw the bootstrap sample Z of size N from the training data. This means that an item is randomly picked from the training data N times with replacement. So, the sample Z is of size N and contains approximately 63% of rows from the original data set.
   (b) Grow a tree T to the bootstrapped data Z, by recursively repeating the following steps for each node of the tree until the tree is fully grown:
      i. Select m attributes at random from P, where P is the total number of attributes and m is less than P.
      ii. Pick the best variable/split-point among the m. The best means that the attribute has the highest information gain where the information gain refers to the decrease in the entropy/information after a dataset is split on the attribute.
      iii. Split the node among the two daughter nodes based on the split-point.
2. Output the ensemble of trees, that is the final decision forest.

The prediction of unseen samples in the random forest is made by taking the majority vote among all the trees in the ensemble.
One of the valuable properties of the random forest is that it behaves best of all when its trees are uncorrelated (independent). Thus, increasing the correlation among the trees increases the error rate of the whole forest. The random selection of the attributes on the tree building phase helps to make them less correlated. So, the small values of m lead to small overall correlation value, but from the other side decrease the strength of the forest because of the use of non-optimal attributes in the splits of the trees. Thus, it is important to find the m value somewhere in the middle so, that the trees are uncorrelated but the strength of the forest is not too low.
Another property of the random forests is that they do not need the separate

training set for calculating the accuracy. For doing it the whole data set is used. The prediction of each row is made by voting among the trees that do not have this row in the bootstrap sample used to train them.
There are some advantages the random forests provide (Berkley, 2016, p. 1): it runs well on large data sets and can be parallelized, it can deal with a big amount of input attributes without the need of deleting them, it shows the most important attributes in the dataset, during the building process it can provide the estimate of the generalization error and finally it provides the good overview of the interactions of the variables in the data set.

4 Conclusion

Decision forest is an efficient and relative simple method of classification. Since, it is an ensemble of decision trees, from one side it uses the advantages of the simple and coherent tree model. From the other side, the usage of average result provided by multiple decision trees helps to avoid different difficulties that often occur when using one simple classifier. Among them the biggest role plays the avoidance of overfitting while still keeping the high accuracy of the final model. There are a lot of different algorithms for building the decision forest. Bagging and boosting are the popular and efficient examples that are briefly discussed in this paper. Random forest is another one technique that has shown good and even the best results in comparative studies. This algorithm uses bagging methodology together with the randomization of attributes used on decision trees building phase. Random forests are discussed in details in this paper including the building steps and the adjustment of the parameters of the algorithm that helps to achieve the best results and to increase the accuracy of the final model.

References

Berkley (2016). Random forests leo breiman and adele cutler. Retrieved November 10 2016, from http://www.stat.berkeley.edu/ breiman/RandomForests/cc_home.htm.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Dietterich, T. G. (2002). Ensemble learning. *The handbook of brain theory and neural networks*, 2:110–125.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D. (2014). Do we need hundreds of classifiers to solve real world classification problems. *J. Mach. Learn. Res*, 15(1):3133–3181.

Friedman, J., Hastie, T., and Tibshirani, R. (2001). *The elements of statistical learning*, volume 1. Springer series in statistics Springer, Berlin.

Ho, T. K. (1995). Random decision forests. In *Document Analysis and Recognition, 1995., Proceedings of the Third International Conference on*, volume 1, pages 278–282. IEEE.

Ho, T. K. (1998). The random subspace method for constructing decision forests. *IEEE transactions on pattern analysis and machine intelligence*, 20(8):832–844.

Ho, T. K. (2002). A data complexity analysis of comparative advantages of decision forest constructors. *Pattern Analysis & Applications*, 5(2):102–112.

Polikar, R. (2006). Ensemble based systems in decision making. *IEEE Circuits and systems magazine*, 6(3):21–45.

Quinlan, J. R. (1987). Generating production rules from decision trees. In *IJCAI*, volume 87, pages 304–307. Citeseer.

Rokach, L. (2016). Decision forest: Twenty years of research. *Information Fusion*, 27:111–125.

5 Student Self-Evaluation Form

Criteria 1: Evidence of understanding
Is it evident from your paper that you understand what you have written?

I think that the paper generally provides the evidence that I understand the decision forests. Moreover, I hope that it is not so difficult for another person to understand how decision forests work after reading my paper. The possible weakness is that I have not provided any diagram or graph that shows how decision forests work. But I do not think that it is really needed.

Criteria 2: Depth, scope & bibliography
Is it evident from your paper that you have researched your topic in depth, from a number of sources, and included content not covered in lecture notes?

Generally, during the search of papers I could find much more materials that are relevant to this theme. However, I do not feel that all of them should be referenced in the paper. Many of them have too specific information that is not relevant to the general overview that I have made.

Criteria 3: Structure & flow of the paper
Does the paper flow well, and is it properly structured.

I find my paper well structured. Nevertheless, if I could spend more time on it, I would like to provide wider information, especially about the building algorithms and in this case I would like to make some restructuring. However, the

main flow of current paper seems to be ok.

Criteria 4: Authors comments & analysis
Have you included your own analysis on what you have read?

I do not think that the points, that I have included, can be classified as an analysis from the point of view given in the example. But I do not feel that I should provide my own analysis in such a paper. From the other side, my points are highly based on the literature that was read by me, so the actual analysis of the literature was really made.