Random Forests: Presentation Summary

Theodoro Koulis

April 1, 2003

1 Introduction

Random forests are a combination of tree predictors, where each tree in the forest depends on the value of some random vector $\underline{\theta}$. We have already seen an example of random forests when bagging was introduced in class. In bagging, one generates a sequence of trees, one from each bootstrapped sample. In this case, the random vector $\underline{\theta}$ represents a single bootstrapped sample. Please note that in this report, we shall discuss random forests in the context of classification.

Note: The ideas presented here can be found in the technical report by Breiman (1999).

In essence, random forests are constructed in the following manner:

- At step k, a $\underline{\theta}_k$ is generated.
- The $\underline{\theta}_k$'s are i.i.d.
- A tree predictor $h(\underline{\mathrm{x}}, \underline{\theta}_k)$ is then constructed using the data chosen by $\underline{\theta}_k$.
- A large number of trees are then generated, $k = 1, \cdots, K$ (usually $K >= 100$).
- After a large number of trees have been generated, they all vote for the most popular class.
- The random forest then classifies $\underline{\mathrm{x}}$ by taking the most popular voted class from all the tree predictors in the forest ($h(\underline{\mathrm{x}}, \underline{\theta}_k)$, $k = 1, \cdots, K$).

Example 1 (Bagging). *Let's look at bagging with a two class problem:*

- *The $\underline{\theta}_k$'s pick out the training set data to be used in the bootstrapped sample.*
- *$\underline{\theta}_k$'s consist of integers between 1 and M, where M is the training set size.*
- *We may have $\underline{\theta}_5 = (3, 4, 56, 3, \cdots, 2)$.*
- *Table 1 gives the relationship between tree classifiers and the random vectors $\underline{\theta}_k$.*

$h(\underline{\mathrm{x}}, \underline{\theta}_1)$	$h(\underline{\mathrm{x}}, \underline{\theta}_2)$	$\cdots$	$h(\underline{\mathrm{x}}, \underline{\theta}_N)$
classifier from 1'st sample	classifier from 2'nd sample	$\cdots$	classifier from N'th sample

Table 1: Random Forest in Bagging

- *Let $P = \sum I(h(\underline{\mathrm{x}}, \underline{\theta}_k) = 1)$.*
- *Let $Q = \sum I(h(\underline{\mathrm{x}}, \underline{\theta}_k) = 0)$.*
- *If $P > Q$ then the forest classifies $\underline{\mathrm{x}}$ as 1*
- *If $Q > P$ then the forest classifies $\underline{\mathrm{x}}$ as 0*

2 Theoretical Background

In this section, we will discuss some theoretical aspects of random forests. In particular, we shall see that random forests do not over-fit the data. This result will be associated with two notions of strength and correlation. We shall follow the same arguments found in Breiman (1999).

2.1 Random Forests Converge

Let $h_1(\underline{\mathrm{x}}), \cdots, h_K(\underline{\mathrm{x}})$ be a collection of classifiers. Also, let $\mathrm{Y}, \underline{\mathrm{X}}$ be a random vector sampled (from some distribution) from the training data. We now define the the margin function of a collection of classifiers.

Definition 1 (Margin Function). The margin function for a collection of classifiers is defined as

$$mg(\underline{\mathrm{X}}, \mathrm{Y}) = \frac{\sum_{k=1}^{K} I(h_k(\underline{\mathrm{X}}) = \mathrm{Y})}{K} - \max_{j \neq \mathrm{Y}} \left[\frac{\sum_{k=1}^{K} I(h_k(\underline{\mathrm{X}}) = j)}{K} \right] \tag{1}$$

where, $I(\cdot)$ is the indicator function.

Note the following:

If $mg(\underline{\mathrm{X}}, \mathrm{Y}) > 0$ then the set of classifiers votes for the correct classification.
If $mg(\underline{\mathrm{X}}, \mathrm{Y}) < 0$ then the set of classifiers votes for a classification that is incorrect.

Definition 2. The generalization error is

$$PE^* = P_{\underline{\mathrm{X}},Y}(mg(\underline{\mathrm{X}}, Y) < 0) \tag{2}$$

This probability is over the space $\underline{\mathrm{X}}, Y$.

The generalization error is simply the misclassification rate that we are familiar with. In the context of random forests, each classifier $h_k(\underline{\mathrm{x}})$ is just $h(\underline{\mathrm{x}}, \underline{\theta}_k)$. From now on, the margin function and the generalization error will be with respect to the random forest $h(\underline{\mathrm{x}}, \underline{\theta}_k)$, $k = 1, \cdots, K$. It was shown in Breiman (1999) that as the random forest gets larger $(K \to \infty)$

$$PE^* \to P_{\underline{\mathrm{X}},\mathrm{Y}} \left(P_{\underline{\theta}}(h(\underline{\mathrm{x}}, \underline{\theta}) = \mathrm{Y})) - \max_{j \neq \mathrm{Y}} P_{\underline{\theta}}(h(\underline{\mathrm{x}}, \underline{\theta}) = j)) < 0 \right) \tag{3}$$

This means that the generalization error has a limiting value and that random forests do not over-fit the data.

2.2 Correlation and Strength

Now we shall see what affects the generalization error for a random forest. The margin function for a forest is:

$$mr(\underline{\mathrm{X}},\mathrm{Y}) = P_{\theta}(h(\underline{\mathrm{x}},\underline{\theta}) = \mathrm{Y})) - \max_{j\neq\mathrm{Y}} P_{\Theta}(h(\underline{\mathrm{x}},\underline{\theta}) = j)) \tag{4}$$

Let

$$s = E_{\underline{\mathrm{X}},Y}(mr(\underline{\mathrm{X}},Y)) \ , \tag{5}$$

the expected strength of the classifiers in a random forest. The strength can be interpreted as a measure of accuracy for each tree in the forest. On top of this we can define the raw margin function

$$rmg(\underline{\theta},\underline{\mathrm{X}},\mathrm{Y}) = I(h(\underline{\mathrm{X}},\underline{\theta}) = Y) - I(h(\underline{\mathrm{X}},\underline{\theta}) = \hat{\jmath}(\underline{\mathrm{X}},Y))) \ , \tag{6}$$

where $\hat{\jmath}(\underline{\mathrm{X}},Y)$ is the most probable predicted class other than Y. One can calculate the correlation between any two raw margin functions, $rmg(\underline{\theta},\underline{\mathrm{X}},\mathrm{Y})$ and $rmg(\underline{\theta}',\underline{\mathrm{X}},\mathrm{Y})$ say. Denote this correlation by $\rho(\underline{\theta},\underline{\theta}')$. Also, let $\overline{\rho}$ be the mean correlation (averaged across all possible pairs $(\underline{\theta},\underline{\theta}')$). Then one can show that (Breiman (1999))

$$PE^{*} \leq \overline{\rho}\frac{(1-s^{2})}{s^{2}} \ . \tag{7}$$

The idea is that one would prefer to use random forests with small $\overline{\rho}$ and to get s close to 1.

Example 2 (Two Class Problem). *In a two class problem, the margin function of a random forest is*

$$mr(\underline{\mathrm{X}},\mathrm{Y}) = 2P_{\underline{\theta}}(h(\underline{\mathrm{x}},\underline{\theta}) = Y) - 1 \tag{8}$$

If $s > 0$ is required, then one would require a random forest such that

$$E_{\underline{\mathrm{X}},Y}(P_{\underline{\theta}}(h(\underline{\mathrm{x}},\underline{\theta}) = Y)) > 0.5 \ . \tag{9}$$

The last equation simply states that we would like to do better at classifying than random chance would allow.

3 Random Selection

In this section we will introduce a random forest which is similar to bagging. The usual tree construction such as CART, a search is performed over all variables to find the best split at a given node. Usually the tree is grown as much as possible and then it is pruned back (1-SE rule for instance). Breiman (1999) introduces the idea of random inputs in tree construction. The following describes random input selection to create random forests.

- Start with a bootstrap sample. Let M be fixed and $M << \#$ of variables.
- At each node, select M variables at random.
- Search for the best split over these M variables.
- The splitting criterion for random input selection is the GINI criterion.
- The largest possible tree is grown and not pruned.
- This is done for each bootstrap sample.
- One then bags the results to enhance accuracy.

Random input selection is faster than methods such as bagging which usually use a greedy search algorithm to find the best split points. The resulting set of tree classifiers is then a random forest.

4 Out-of-Bag Estimation

The use of bootstrapped samples to grow each tree brings us to the notion of out-of-bag estimation. Let $T =$ the training set and $T_k =$ the k'th bootstrapped sample associated with $h(\underline{\mathrm{X}}, \Theta_k)$. For each $\underline{\mathrm{x}}, y$ in the training set, aggregate the votes only for those classifiers for which T_k does not contain $\underline{\mathrm{x}}, y$. These are the out-of-bag data points. Bagging these points results in out-of-bag classification. Since these points are independent of the classifiers, we can use them to get an unbiased error rate. This practically eliminates the need for a test set or cross-validation. More details on out-of-bag estimation can be found in Breiman (2003).

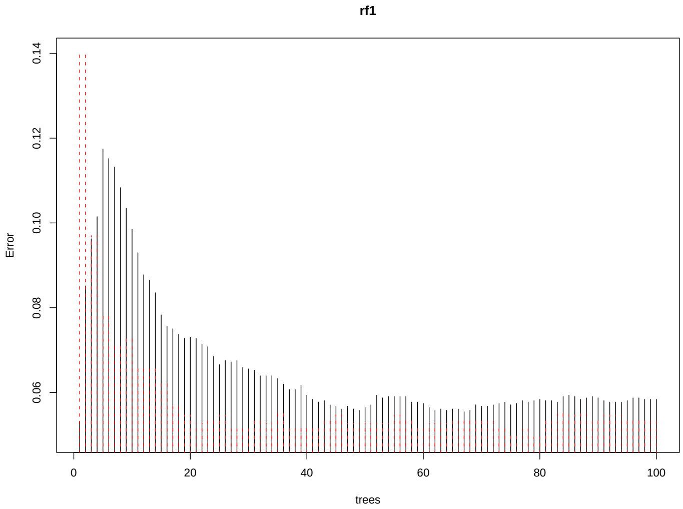


Figure 1: Out-of-bag Error as Forest Gets Larger, $M = 7$

5 SPAM Data

In this section we shall illustrate the methods of random input selection and out-of-bag estimation with the SPAM data set that we are familiar with from class (Hopkins et al. (1999)).

First we used SPAM data with a training set and test set. This is redundant since the out-of-bag estimates make test sets obsolete. For demonstration purposes, they were included in the analysis. We used $M = 7$ for the random input selection. Here is the result:

```
        OOB estimate of  error rate: 5.84%
Confusion matrix:
     0    1 class.error
0 1762   65  0.03557745
1  114 1124  0.09208401

                Test set error rate: 5.34%
Confusion matrix:
    0   1 class.error
0 928  33  0.03433923
1  49 526  0.08521739
```

As you can see, the out-of-bag error estimate is close to the test set estimate. Recall that this was not always the case with CART or RPART. This compares very well with boosting. The test set error rate obtained in class using boosting was around 5.2%. Figure 1 shows that the estimate for the error rate converges to a lower-bound as expected from the theoretical results of random forests.

We re-performed the analysis with $M = 2$. Here is the result:

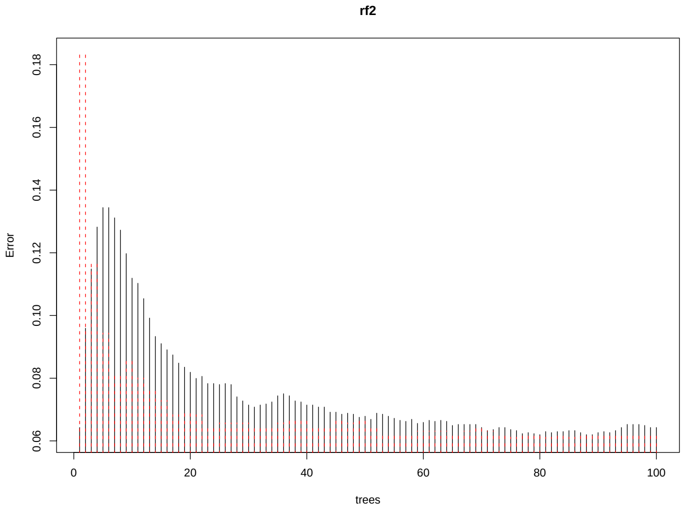


Figure 2: Out-of-bag Error as Forest Gets Larger, $M = 2$

```
        OOB estimate of  error rate: 6.43%
Confusion matrix:
     0    1 class.error
0 1770   57  0.03119869
1  140 1098  0.11308562
                Test set error rate: 6.18%
Confusion matrix:
    0   1 class.error
0 921  40  0.04162331
1  55 520  0.09565217
```

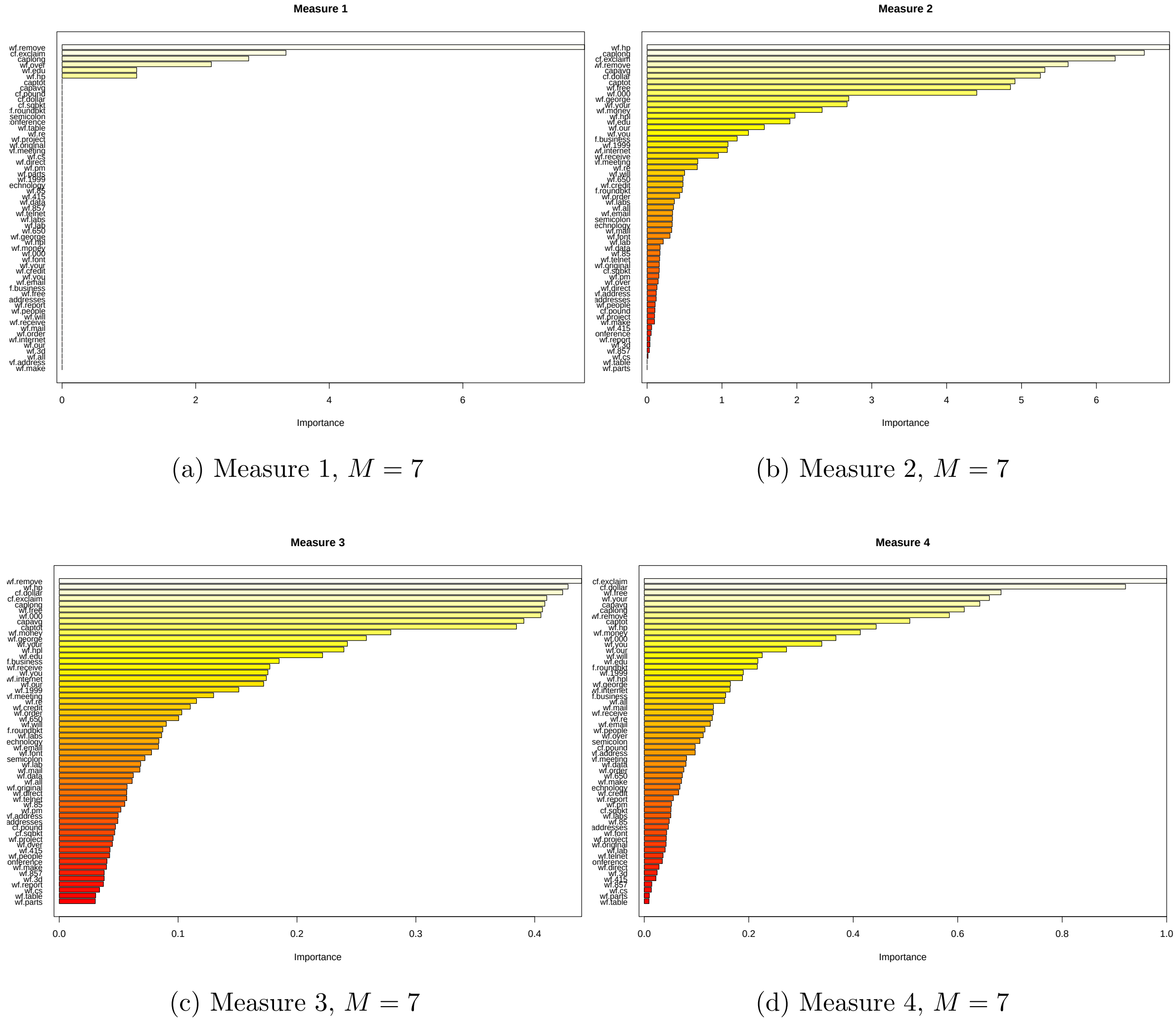
As you can see there is a sligh improvement when M is larger. Figure 2 illustrates the convergence of the generalization error as before.

6 Importance of Variables

There are four measures of variable importance in random forests

- Measure 1: Get the left out cases for the k'th tree. Randomly permute values of the m'th covariate. Compute the error rate. The amount by which this new error rate exceeds the original out-of-bag estimate is the importance of the m'th covariate
- Measure 2: Instead of error, use the empirical margin function for each permuted covariate. The 2'nd measure is the average lowering of the of the margin.
- Measure 3: Using the same permute values, count how many margins are lowered minus the number raised.
- Measure 4: The splitting criterion for random input selection is the GINI criterion. Measure 4 is the sum of all decreases in the GINI criterion due to a given a variable, normalized by the number of trees.

We can use the SPAM data to illustrate these measures. The figure on the next page shows the importance of the covariates using the 4 measures described above. The first measure is the harshest one of all, and this is usually the case. Measure 4 is similar to the importance measure used in boosting (as seen in class).



(a) Measure 1, $M = 7$

(b) Measure 2, $M = 7$

(c) Measure 3, $M = 7$

(d) Measure 4, $M = 7$

7 Conclusions

We have seen that random forests have a limiting generalization error which means that no over-fitting is possible. This is a good feature which allows one to grow a random forest as much as possible to obtain a generalization error that is close to the lower-bound (in principle). Also, there is a relationship between strength, correlation and error. It is not clear at present how one could improve strength and correlation or even how to construct a random forest with optimal correlation and strength. This would be a great topic for future research. As seen with the SPAM data, the precision in random forests is comparable to that of boosting methods. Another neat feature of random forests is the use of out-of-bag estimates which essentially eliminates the need for test data. Although not discussed here, random forests can be easily extended to regression type settings. With further research, random forests has the potential to become a powerful tool in tree based classification and regression methods.

References

Breiman, Leo (1999). Random Forests – Random Features. http://oz.berkeley.edu/users/breiman/randomforest2001.pdf.

Breiman, Leo (2003). Manual on setting up, using, and understanding random forests v3.1. http://oz.berkeley.edu/users/breiman/Using_random_forests_V3.1.pdf.

Hopkins, Mark, Reeber, Eric, Forman, George, and Suermondt, Jaap (1999). SPAM E-mail Database. Donor: George Forman (gforman at nospam hpl.hp.com), Hewlett-Packard Labs, 1501 Page Mill Rd., Palo Alto, CA 94304. URL `http://www-stat-class.stanford.edu/ tibs/ElemStatLearn/datasets/spam.data`