# Random Forests

Mateusz Markiewicz

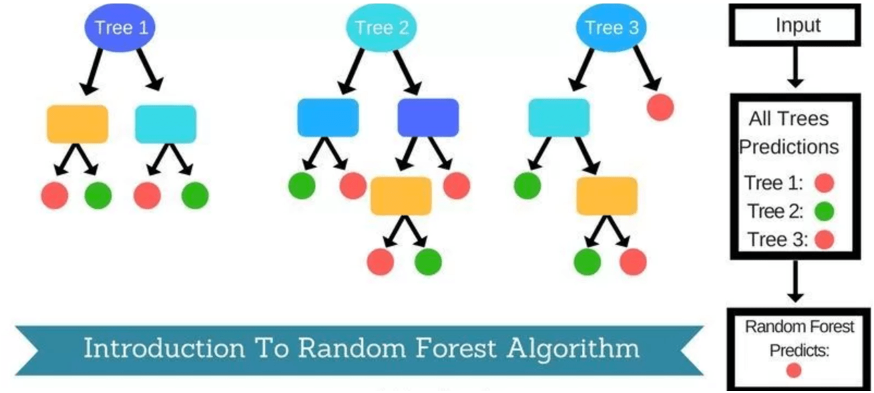
November 2020

## Introduction

### Random forests

Random forests (Breiman, 2001) is a substantial modification of bagging that builds a large collection of de-correlated trees, and then averages them.
On many problems the performance of random forests is very similar to boosting, and they are simpler to train and tune. As a consequence, random forests are popular, and are implemented in a variety of packages. [1]

## Introduction



## Definition of Random Forests

**Algorithm**

1. For $b = 1$ to $B$:
    1. Draw a bootstrap sample $Z$ of size $N$ from the training data
    2. Grow a random-forest tree $T_b$ to the bootstrapped data, by recursively repeating the following steps for each terminal node of the tree, until the minimum node size $n_min$ is reached
        1. Select $m$ variables at random from the $p$ variables
        2. Pick the best variable/split-point among the $m$
        3. Split the node into two daughter nodes
2. Output the ensemble of trees $\{Tb\}_1^B$

# Accuracy

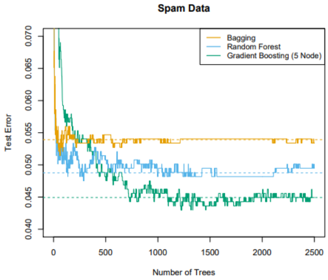


**FIGURE 15.1.** *Bagging, random forest, and gradient boosting, applied to the spam data. For boosting, 5-node trees were used, and the number of trees were chosen by 10-fold cross-validation (2500 trees). Each "step" in the figure corresponds to a change in a single misclassification (in a test set of 1536).*

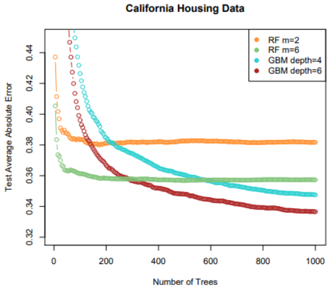


**FIGURE 15.3.** *Random forests compared to gradient boosting on the California housing data. The curves represent mean absolute error on the test data as a function of the number of trees in the models. Two random forests are shown, with $m = 2$ and $m = 6$. The two gradient boosted models use a shrinkage parameter $\nu = 0.05$ in (10.41), and have interaction depths of 4 and 6. The boosted models outperform random forests.*

## The $m$ parameter

### Idea

The idea in random forests is to improve the variance reduction of bagging by reducing the correlation between the trees, without increasing the bias too much. This is achieved in the tree-growing process through random selection of the input variables [1]

### $M$ value

Typically values for $m$ are $\sqrt{p}$ or even as low as 1

## Classification and regression

### Classification vs regression

Random forest can be used both for classification and regression

- when used for classification, a random forest obtains a class vote from each tree, and then classifies using majority vote
- when used for regression, the predictions from each tree at a target point $x$ are simply averaged

### M parameter

In addition, the inventors make the following recommendations:

- for classification, the default value for $m$ is $\lfloor \sqrt{p} \rfloor$ and the minimum node size is one
- for regression, the default value for $m$ is $\lfloor \frac{p}{3} \rfloor$ and the minimum node size is five [1]

# Out of Bag Samples

## Definition

For each observation $z_i = (x_i, y_i)$, construct its random forest predictor by averaging only those trees corresponding to bootstrap samples in which $z_i$ did not appear

## Usage

An OOB error estimate is almost identical to that obtained by N-fold crossvalidation. Once the OOB error stabilizes, the training can be terminated.[1]

## Out of Bag Samples

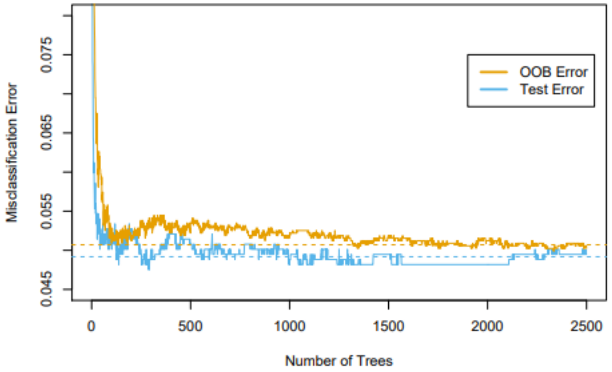


**FIGURE 15.4.** OOB *error computed on the* spam *training data, compared to the test error computed on the test set.*

## Variable Importance

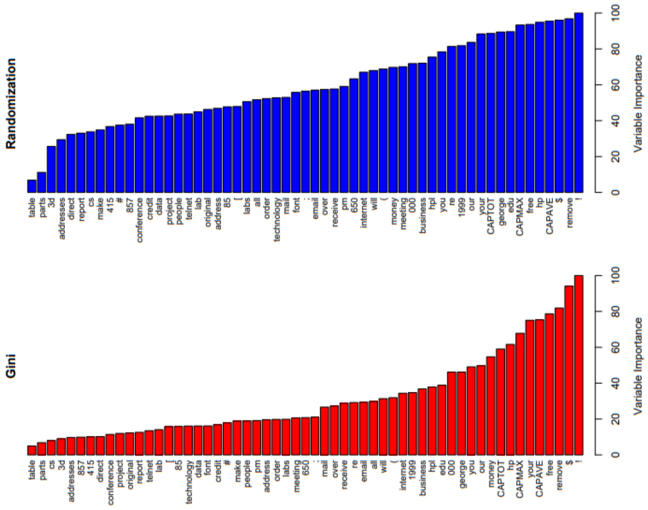
### Improvement

At each split in each tree, the improvement in the split-criterion is the importance measure attributed to the splitting variable, and is accumulated over all the trees in the forest separately for each variable.

### OOB

When the $b_{th}$ tree is grown, the OOB samples are passed down the tree, and the prediction accuracy is recorded. Then the values for the $j_{th}$ variable are randomly permuted in the OOB samples, and the accuracy is again computed. The decrease in accuracy as a result of this permuting is averaged over all trees, and is used as a measure of the importance of variable $j$ in the random forest.

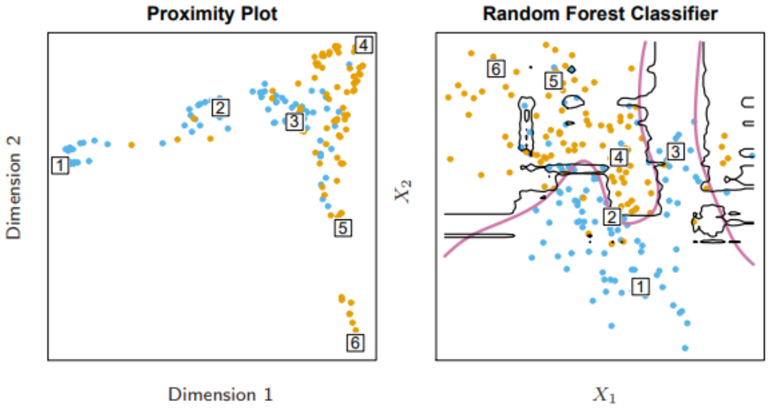
## Variable Importance



## Proximity Plots

**Proximity matrix**

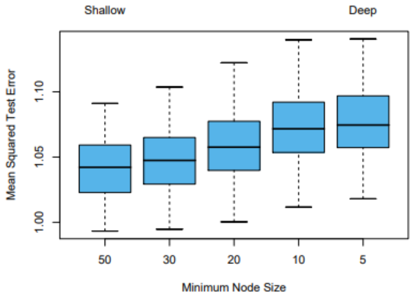
For every tree, any pair of OOB observations sharing a terminal node has their proximity increased by one. This proximity matrix is then represented in two dimensions using multidimensional scaling.



## Overfitting

### Overfitting

Another claim is that random forests “cannot overfit” the data. It is certainly true that increasing B does not cause the random forest sequence to overfit. However, the average of fully grown trees can result in too rich a model, and incur unnecessary variance. Segal (2004) demonstrates small gains in performance by controlling the depths of the individual trees grown in random forests.



## Bibliography

[1] Tibshirani R. Friedman J. Hastie, T. *The Elements of Statistical Learning*. 2001.