Overview

- Intuition of Random Forest
- The Random Forest Algorithm
- De-correlation gives better accuracy
- Out-of-bag error (OOB-error)
- Variable importance
Intuition of Random Forest

New sample: old, retired, male, short

Tree predictions: diseased, healthy, diseased

Majority rule: diseased
The Random Forest Algorithm

1. For $b = 1$ to $B$:
   
   (a) Draw a bootstrap sample $Z^*$ of size $N$ from the training data.
   
   (b) 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 $n_{\text{min}}$ is reached.
      
      i. Select $m$ variables at random from the $p$ variables.
      
      ii. Pick the best variable/split-point among the $m$.
      
      iii. Split the node into two daughter nodes.

2. Output the ensemble of trees $\{T_b\}_1^B$.

To make a prediction at a new point $x$:

Regression: $\hat{f}^{B}_{\text{rf}}(x) = \frac{1}{B} \sum_{b=1}^{B} T_b(x)$.

Classification: Let $\hat{C}_b(x)$ be the class prediction of the $b$th random-forest tree. Then $\hat{C}^{B}_{\text{rf}}(x) = \text{majority vote } \{\hat{C}_b(x)\}_1^B$. 
Differences to standard tree

- Train each tree on bootstrap resample of data
  (Bootstrap resample of data set with N samples: Make new data set by drawing with replacement N samples; i.e., some samples will probably occur multiple times in new data set)

- For each split, consider only m randomly selected variables

- Don’t prune

- Fit B trees in such a way and use average or majority voting to aggregate results
Why Random Forest works 1/2

- Mean Squared Error = Variance + Bias^2
- If trees are sufficiently deep, they have very small bias

- How could we improve the variance over that of a single tree?
Why Random Forest works 2/2

\[
\text{Var} \left( \frac{1}{B} \sum_{i=1}^{B} T_i(c) \right) = \frac{1}{B^2} \sum_{i=1}^{B} \sum_{j=1}^{B} \text{Cov}(T_i(x), T_j(x))
\]

\[
= \frac{1}{B^2} \sum_{i=1}^{B} \left( \sum_{j \neq i}^{B} \text{Cov}(T_i(x), T_j(x)) + \text{Var}(T_i(x)) \right)
\]

\[
= \frac{1}{B^2} \sum_{i=1}^{B} \left( (B - 1)\sigma^2 \cdot \rho + \sigma^2 \right)
\]

\[
= \frac{B(B - 1)\rho \sigma^2 + B \sigma^2}{B^2}
\]

\[
= \frac{(B - 1)\rho \sigma^2}{B} + \frac{\sigma^2}{B}
\]

\[
= \rho \sigma^2 - \frac{\rho \sigma^2}{B} + \frac{\sigma^2}{B}
\]

Decreases, if number of trees $B$ increases (irrespective of $\rho$)

Decreases, if $\rho$ decreases, i.e., if $m$ decreases

De-correlation gives better accuracy
Estimating generalization error: Out-of bag (OOB) error

- Similar to leave-one-out cross-validation, but almost without any additional computational burden
- OOB error is a random number, since based on random resamples of the data

**Data:**
- old, tall – healthy
- old, short – diseased
- young, tall – healthy
- young, short – diseased
- young, short – healthy
- young, tall – healthy
- old, short – diseased

**Resampled Data:**
- old, tall – healthy
- old, short – diseased
- young, tall – healthy
- young, short – healthy
- young, tall – healthy

**Out of bag samples:**
- young, short – diseased
- young, short – healthy
- young, tall – healthy
- old, short – diseased

**Out of bag (OOB) error rate:**
\[ \frac{1}{4} = 0.25 \]
Variable Importance for variable \( i \) using Permutations

\[ \bar{d} = \frac{1}{m} \sum_{i=1}^{m} d_i \]
\[ s_d^2 = \frac{1}{m-1} \sum_{i=1}^{m} (d_i - \bar{d})^2 \]
\[ \mathcal{V}_i = \frac{\bar{d}}{s_d} \]
<table>
<thead>
<tr>
<th>Trees</th>
<th>vs.</th>
<th>Random Forest</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Trees yield insight into decision rules</td>
<td>+ RF as smaller prediction variance and therefore usually a better general performance</td>
<td></td>
</tr>
<tr>
<td>+ Rather fast</td>
<td>+ Easy to tune parameters</td>
<td></td>
</tr>
<tr>
<td>+ Easy to tune parameters</td>
<td>- Rather slow</td>
<td></td>
</tr>
<tr>
<td>- Prediction of trees tend to have a high variance</td>
<td>- “Black Box”: Rather hard to get insights into decision rules</td>
<td></td>
</tr>
</tbody>
</table>
Comparing runtime (just for illustration)

- Up to “thousands” of variables
- Problematic if there are categorical predictors with many levels (max: 32 levels)

**9 continuous predictors**

- RF: First predictor cut into 15 levels
RF vs. LDA

RF:
+ Can model nonlinear class boundaries
+ OOB error “for free” (no CV needed)
+ Works on continuous and categorical responses (regression / classification)
+ Gives variable importance
+ Very good performance
- “Black box”
- Slow

LDA:
+ Very fast
+ Discriminants for visualizing group separation
+ Can read off decision rule
- Can model only linear class boundaries
- Mediocre performance
- No variable selection
- Only on categorical response
- Needs CV for estimating prediction error
Concepts to know

- Idea of Random Forest and how it reduces the prediction variance of trees
- OOB error
- Variable Importance based on Permutation
R functions to know

- Function “randomForest” and “varImpPlot” from package “randomForest”