Bagging, Random Forests, Extra Trees

Random Forests

Random Forests

From Bagging to Random Forests

Variance of an average of B i.i.d. random variables

$$\frac{1}{B}\sigma^2$$

$\rightarrow$ Bagging: Averaging over B trees decreases variance

Variance of an average of B i.d. random variables with $\rho > 0$

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

$\rightarrow$ **Random Forests**: Averaging over B trees with m out of p predictors per split decreases variance and decorrelates trees

Random Forests

From Bagging to Random Forests

Variance of an average of B i.i.d. random variables

$$\frac{1}{B}\sigma^2$$

$\rightarrow$ Bagging: Averaging over B trees decreases variance

Variance of an average of B i.d. random variables with $\rho > 0$

$$\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$$

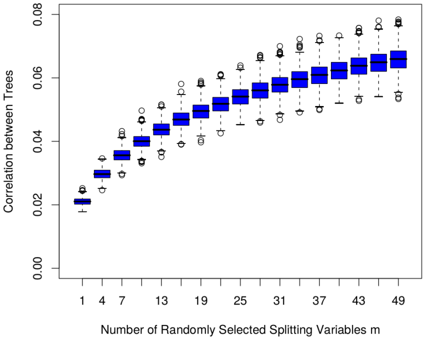
$\rightarrow$ **Random Forests**: Averaging over B trees with m out of p predictors per split decreases variance and decorrelates trees

Random Forests

The Random Forest trick (Breiman 2001)

- Randomization with respect to rows *and* columns
- Weaker predictors have more of a chance
- Results in diverse and *decorrelated* trees

Figure: Correlations between pairs of trees[1]



[1] Hastie et al. (2009)

Growing a Forest

Algorithm 1: Grow a Random Forest

```
Set number of trees B;
Set predictor subset size m;
Define stopping criteria;
for b = 1 to B do
    draw a bootstrap sample from the training data;
    assign sampled data to root node;
    if stopping criterion is reached then
        end splitting;
    else
        draw a random sample m from the p predictors;
        find the optimal split point among m;
        split node into two subnodes at this split point;
        for each node of the current tree do
            continue tree growing process;
        end
    end
end
```

Growing a Forest

A Random Forest

$$\{T_b\}_1^B$$

consists of a set of $b = 1, 2, \ldots, B$ trees which can be used for prediction by...

- Regression
  - Averaging predictions over all trees
  - $\hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$
- Classification
  - Using most commonly occurring class among all trees
  - $\hat{C}_{rf}^B(x) = \text{majority vote}\{\hat{C}_b(x)\}_1^B$
- Probability estimation
  - Using the proportion of class votes of all trees
  - Averaging predicted probabilities over all trees

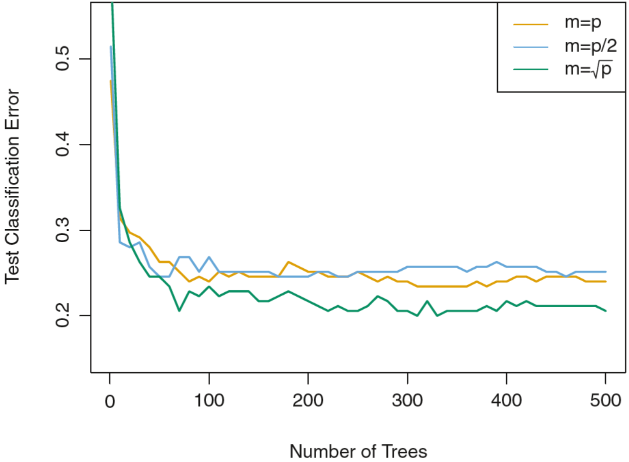
Tuning & Proximities

Tuning Random Forests

- Predictor subset size m out of p (`mtry`)
  - Most important tuning parameter in RF
  - Starting value; $m = \sqrt{p}$ (classification), $m = p/3$ (regression)
  - Can be chosen using OOB errors based on different m
- Number of trees
  - sufficiently high (e.g. 500)
- Node size (number of observations in terminal nodes)
  - sufficiently low (e.g. 5)

Tuning & Proximities

Figure: Test error curves by *m* out of *p* (example)[2]



[2] James et al. (2013)

Tuning & Proximities

$N \times N$ Proximity Matrix

- Represents distances between observations based on a random forest
- For each tree, pairs of OOB cases in the same terminal node get their proximity increased by one
- Can be used for missing value imputation
  1. Do a mean imputation of missings in x
  2. Update the imputed values by the average of x of the non-missing cases weighted by the proximities