# Bagging

Statistical Learning

Master in Big Data. University of Santiago de Compostela
Manuel Mucientes

# Bagging

- Bagging or bootstrap aggregation
- Technique to reduce the variance of an estimated prediction function
- Works especially well for high-variance, low-bias procedures
  - Example: trees
    - Split training in two parts at random
    - Fit a tree to both halves: results could be quite different
- Given a set of n independent observations $Z_1$, …, $Z_n$, each with variance $\sigma^2$: $var\left(\bar{Z}\right) = \sigma^2/n$
  - Averaging a set of observations reduces variance

# Bagging (ii)

- First approach:
  - Use many training sets
  - Build a separate prediction model for each training set
  - Average the resulting predictions

$$\hat{f}_{\mathrm{avg}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{b}(x)$$

  - Not practical: we do not have multiple training sets
- Bagging:
  - Use bootstrap to take samples from the training set
    - Generate *B* different bootstrapped training sets
  - Train a model on each *b*th training set
  - Average all the predictions

$$\hat{f}_{\mathrm{bag}}(x) = \frac{1}{B}\sum_{b=1}^{B}\hat{f}^{*b}(x)$$

  - For classification: majority vote

# Random Forest (RF)

- Boosting appears to dominate bagging in most problems
- RF:
  - Substantial modification of bagging
  - Builds a large collection of trees and averages them
    - Reduce the variance by averaging many noisy but approximately unbiased models
    - Trees are ideal candidates for bagging:
      - Capture complex information
      - If grown sufficiently deep, have relatively low bias
  - Performance similar to boosting, but simpler to train and tune

# Random Forest (ii)

**Algorithm 15.1** *Random Forest for Regression or Classification.*

1. For $b = 1$ to $B$:

   (a) Draw a bootstrap sample $\mathbf{Z}^*$ 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 $n_{min}$ 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:* $\hat{f}_{\text{rf}}^B(x) = \frac{1}{B}\sum_{b=1}^B T_b(x)$.

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

# Random Forest (iii)

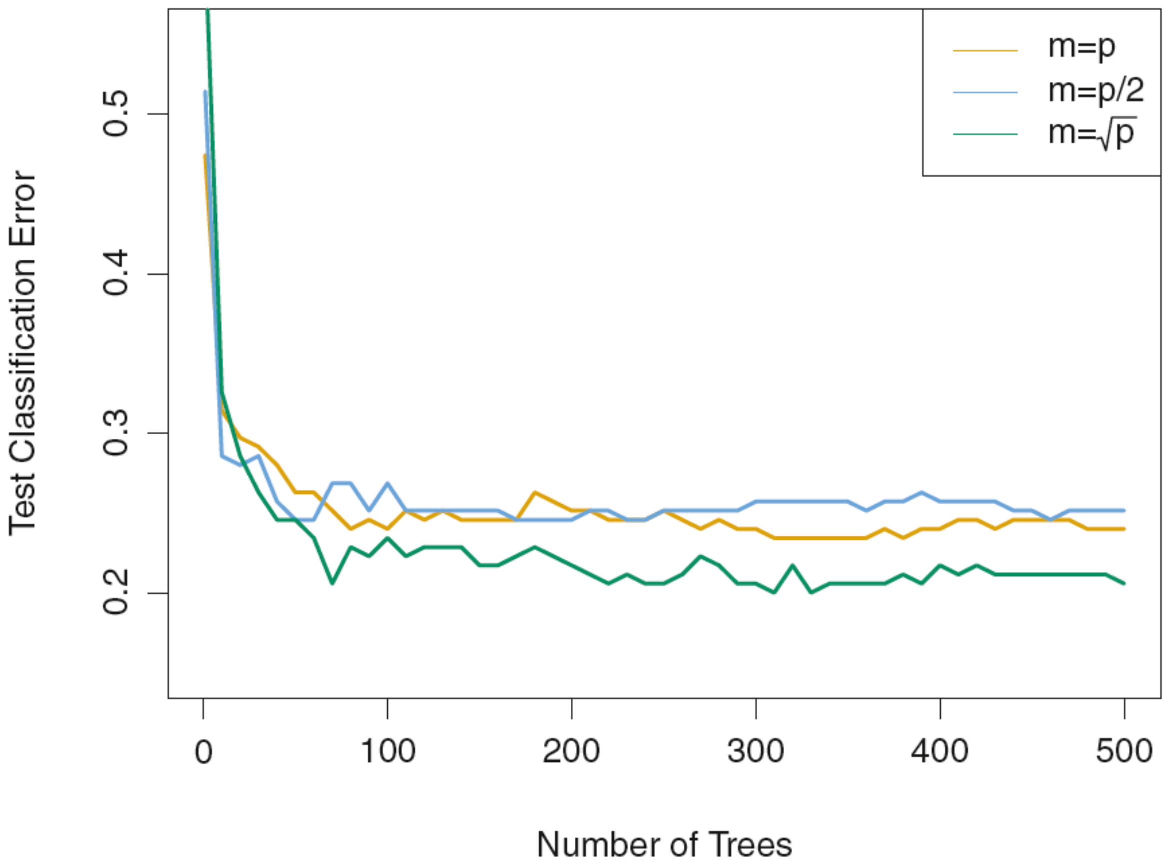
- Bagging with trees:
  - Bias of bagged trees is the same as that of individual trees
  - The only hope of improvement is through variance reduction
- Boosting of trees:
  - Trees are grown on an adaptive way to remove bias
- In bagging, as *B* increases the variance reduces, but up to a limit:
  - For *B* large, the correlation of pairs of bagged trees limits the benefits of averaging
  - i.i.d. random variables: $\frac{1}{B}\sigma^2$
  - i.d. (identically distributed) random variables: $\rho\sigma^2 + \frac{1-\rho}{B}\sigma^2$
    - $\sigma^2$: variance of a tree
    - $\rho$: positive pairwise correlation of two trees

# Random Forest (iv)

- Idea in RF: improve variance reduction of bagging
  - Decreasing the correlation between trees
  - Without increasing variance too much
- Radom selection of input variables as candidates for splitting
  - Typical value for *m*: $\sqrt{p}$
  - Reducing *m* will reduce the correlation between any pair of trees:
    - Reduces the variance of the average
    - This does not mean that the error improves

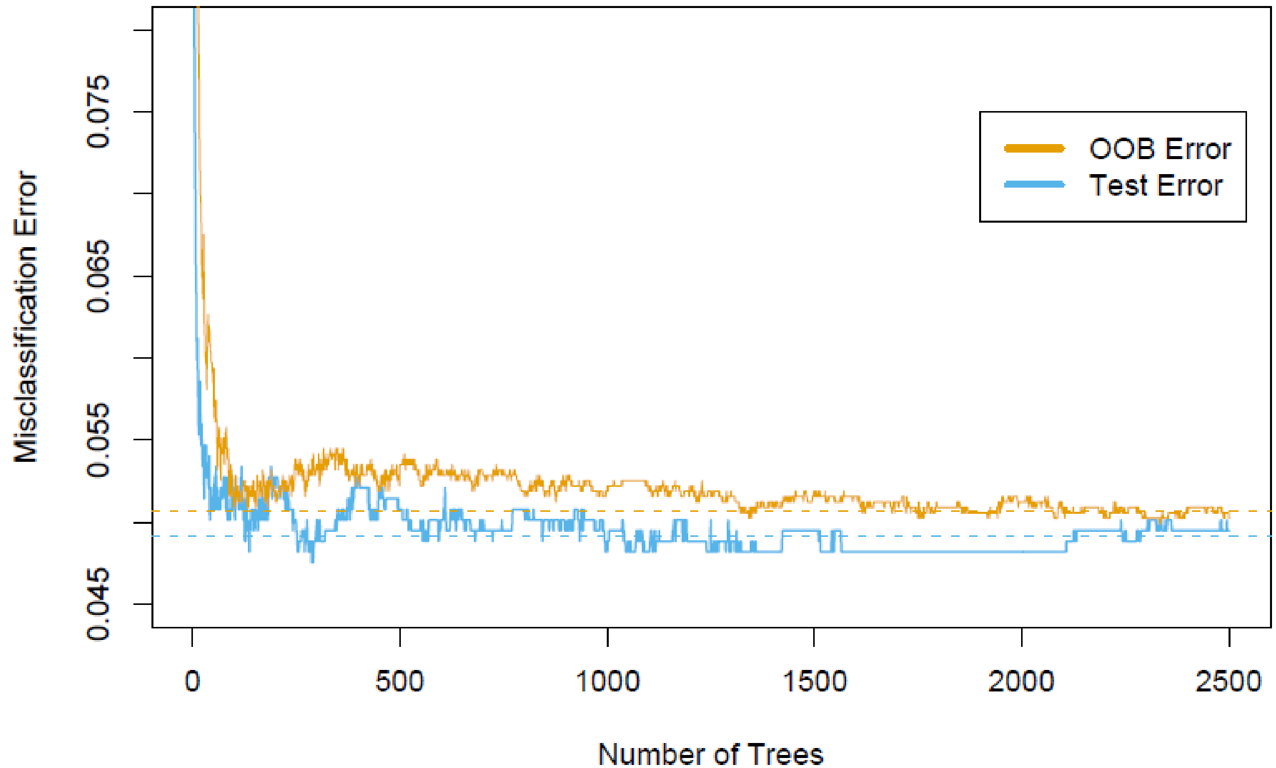
# Number of predictors for splitting

- The best value for *m* depends on the problem: tuning parameter
  - $\sqrt{p}$ is a reasonable choice
- Example: 15-class gene expression data



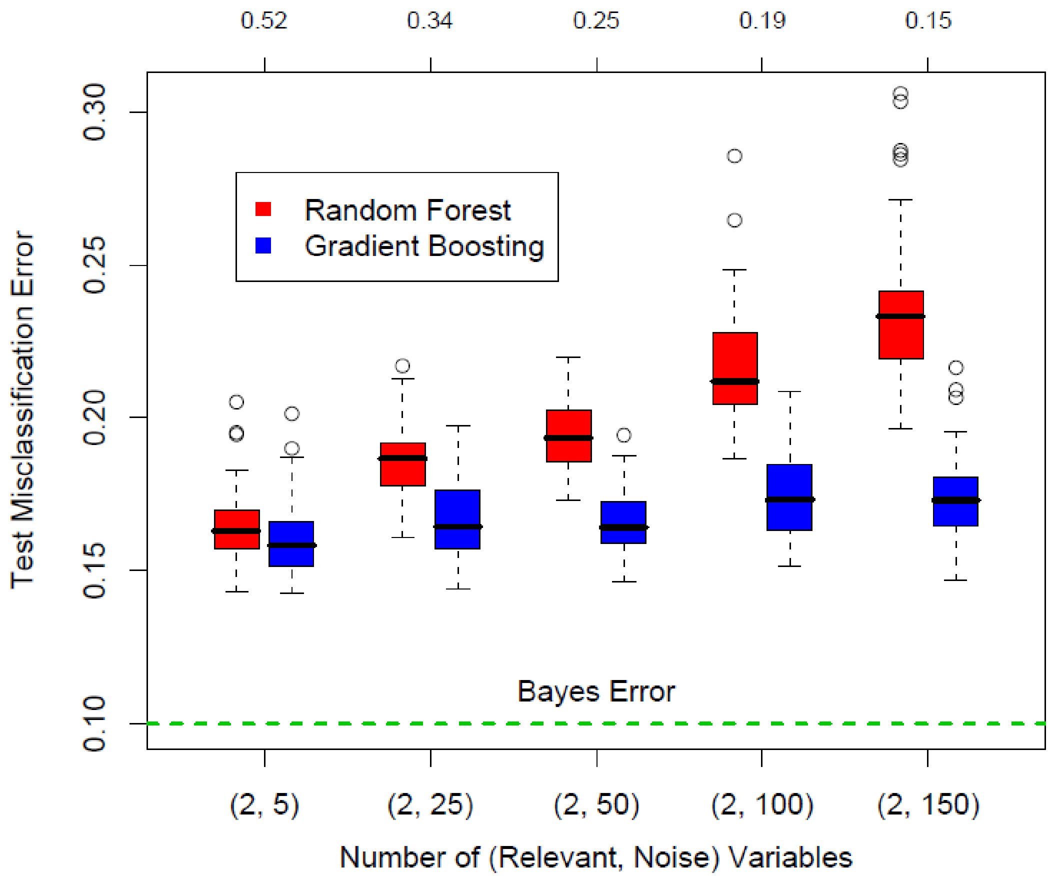
# Out of Bag (OOB) error

- OOB samples: for each observation $z_i = (x_i, y_i)$ construct the output by averaging those trees corresponding to bootstrap samples in which $z_i$ did not appear
- OOB error estimate is almost identical to that of N-fold cross-validation
- With OOB, RF can be fit in one sequence



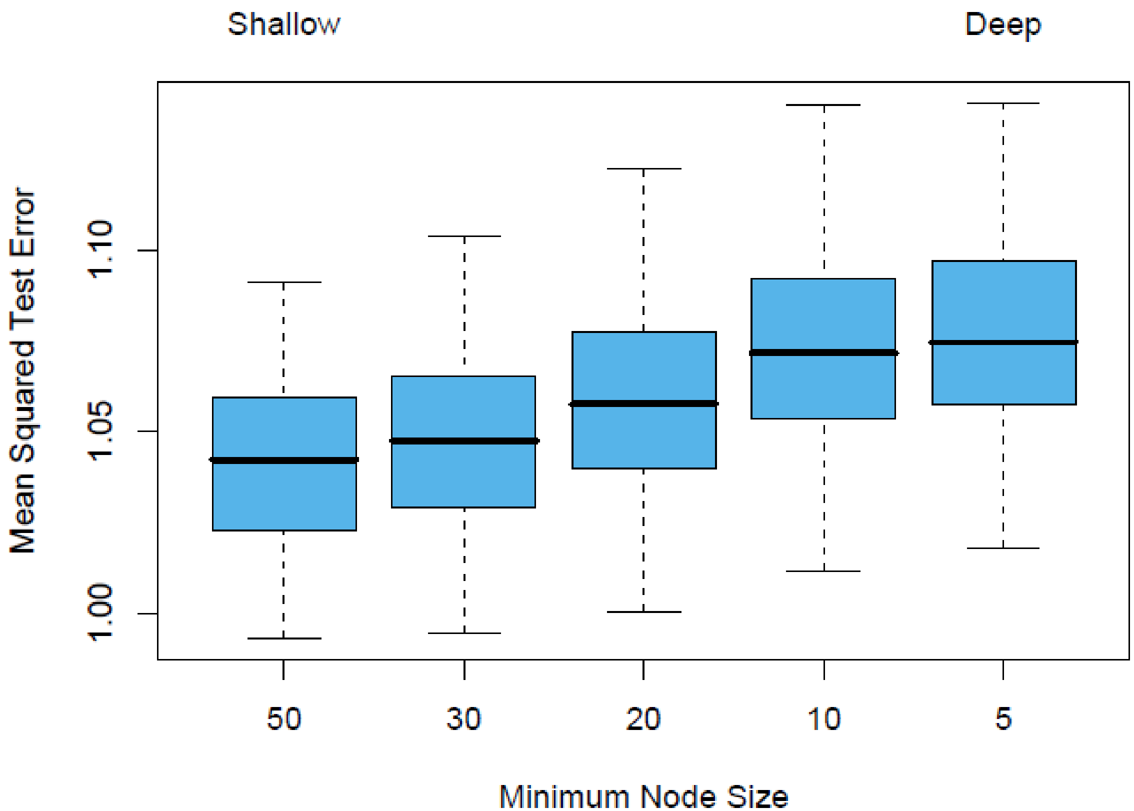
# RF and overfitting

- When the number of variables is high, but the fraction of relevant variables is small, RF is likely to perform poorly with small *m*
- If the number of relevant variables increases, RF is very robust to an increase in the number of noise variables
- Example: probability to select a relevant variable in any split for m=$\sqrt{p}$
  - (6, 100) gives 0.46 vs. (2, 100) gives 0.19



# RF and overfitting (ii)

- RF can overfit for large *B*
- Small gains in performance by controlling the depths of the individual trees in RF
  - Full-grown trees seldom cost much
  - One less tuning parameter
- Example:
  - Low increase in error for deeper trees



# Bibliography


- T. Hastie, R. Tibshirani, y J. Friedman, The elements of statistical learning. Springer, 2009.
  - Chapter 15

- G. James, D. Witten, T. Hastie, y R. Tibshirani, An Introduction to Statistical Learning with Applications in R. Springer, 2013.
  - Chapter 8, Sec. 8.2