BAGGING AND RANDOM FORESTS

Chapter 08 (part 02)

Outline

- Bagging
  - Bootstrapping
  - Bagging for Regression Trees
  - Bagging for Classification Trees
  - Out-of-Bag Error Estimation
  - Variable Importance: Relative Influence Plots
- Random Forests

BAGGING

Problem!

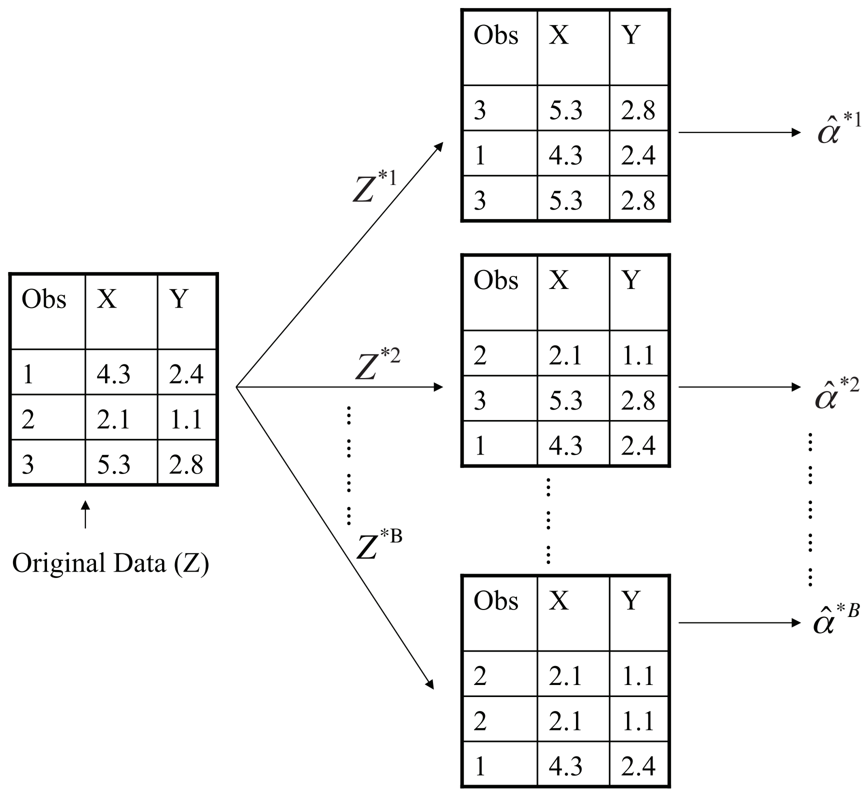
- Decision trees discussed earlier suffer from high variance!
  - If we randomly split the training data into 2 parts, and fit decision trees on both parts, the results could be quite different

- We would like to have models with low variance

- To solve this problem, we can use bagging (***b***ootstrap ***agg***regat***ing***).

Bootstrapping is simple!

- Resampling of the observed dataset (and of equal size to the observed dataset), each of which is obtained by random sampling with replacement from the original dataset.



What is bagging?

- Bagging is an extremely powerful idea based on two things:
  - Averaging: reduces variance!
  - Bootstrapping: plenty of training datasets!
- Why does averaging reduces variance?
  - Averaging a set of observations reduces variance. Recall that given a set of n independent observations $Z_1, \ldots, Z_n$, each with variance σ^2, the variance of the mean $\bar{Z}$ of the observations is given by σ^2/n

How does bagging work?

- Generate B different bootstrapped training datasets
- Train the statistical learning method on each of the B training datasets, and obtain the prediction
- For prediction:
  - Regression: average all predictions from all B trees
  - Classification: majority vote among all B trees

Bagging for Regression Trees

- Construct B regression trees using B bootstrapped training datasets
- Average the resulting predictions

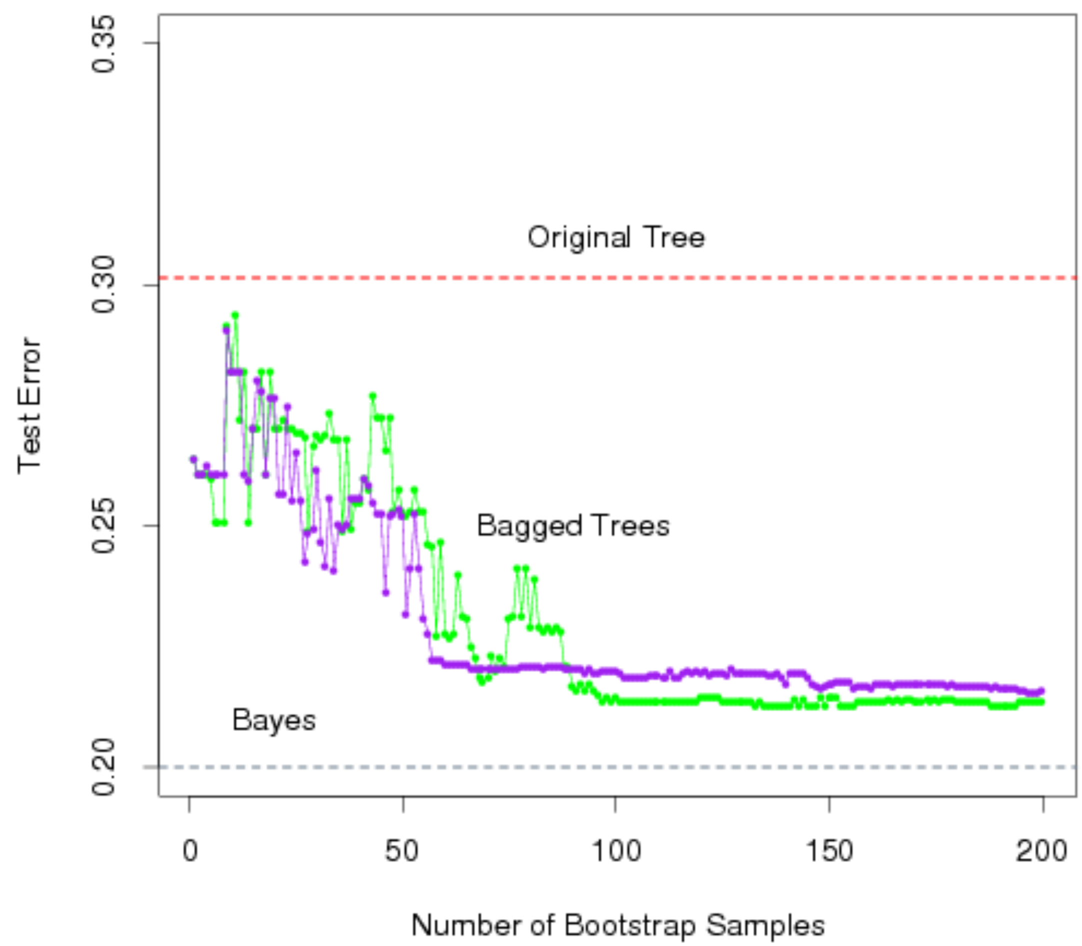
- Note: These trees are not pruned, so each individual tree has high variance but low bias. Averaging these trees reduces variance, and thus we end up lowering both variance and bias ☺

Bagging for Classification Trees

- Construct B regression trees using B bootstrapped training datasets
- For prediction, there are two approaches:
    1. Record the class that each bootstrapped data set predicts and provide an overall prediction to the most commonly occurring one (majority vote).
    2. If our classifier produces probability estimates we can just average the probabilities and then predict to the class with the highest probability.
- Both methods work well.

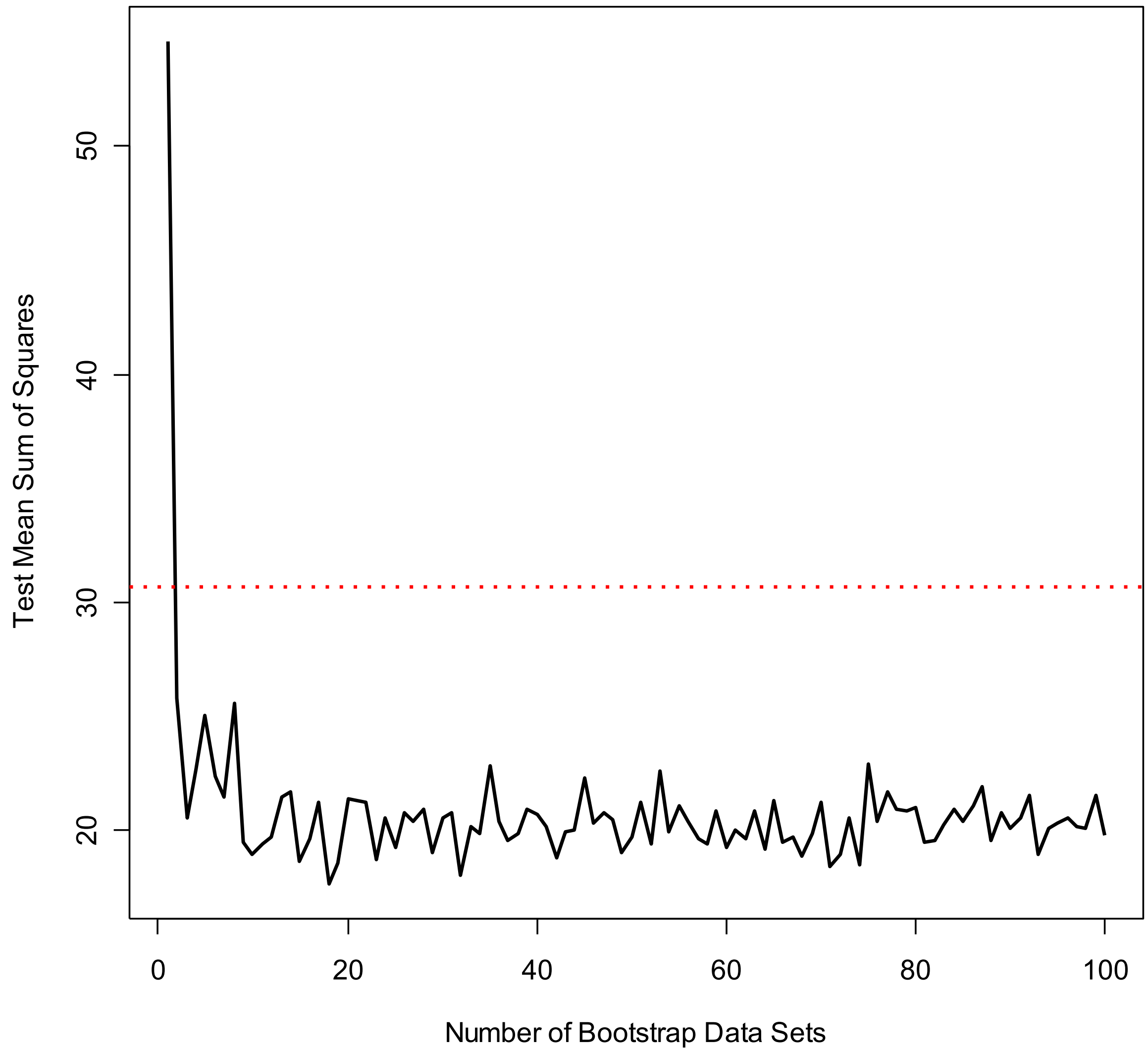
A Comparison of Error Rates

- Here the green line represents a simple majority vote approach
- The purple line corresponds to averaging the probability estimates.
- Both do far better than a single tree (dashed red) and get close to the Bayes error rate (dashed grey).



Example 1: Housing Data

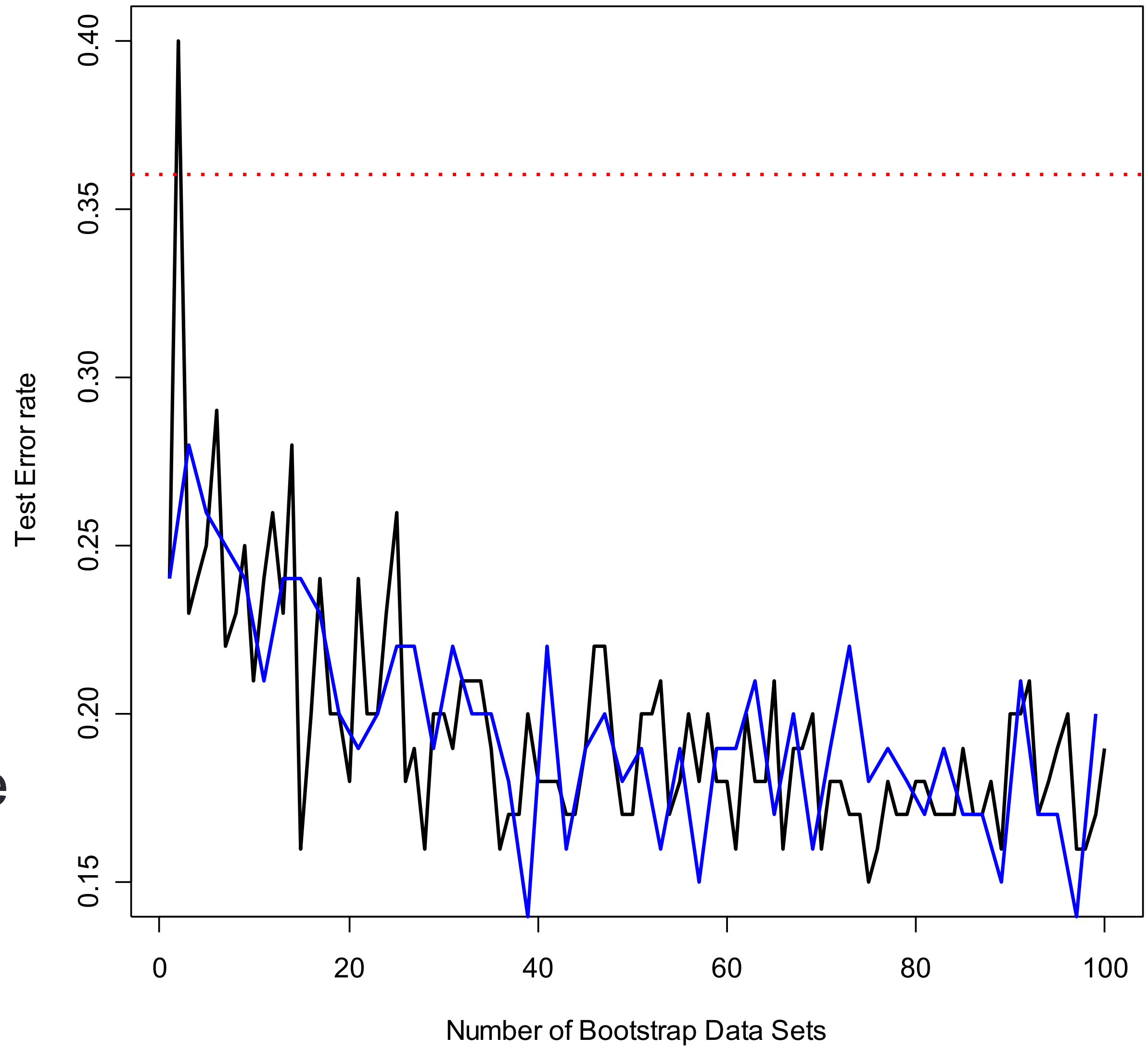
- The red line represents the test mean sum of squares using a single tree.
- The black line corresponds to the bagging error rate



Example 2: Car Seat Data

- The red line represents the test error rate using a single tree.

- The black line corresponds to the bagging error rate using majority vote while the blue line averages the probabilities.



Out-of-Bag Error Estimation

- Since bootstrapping involves random selection of subsets of observations to build a training data set, then the remaining non-selected part could be the testing data.
- On average, each bagged tree makes use of around 2/3 of the observations, so we end up having 1/3 of the observations used for testing

Variable Importance Measure

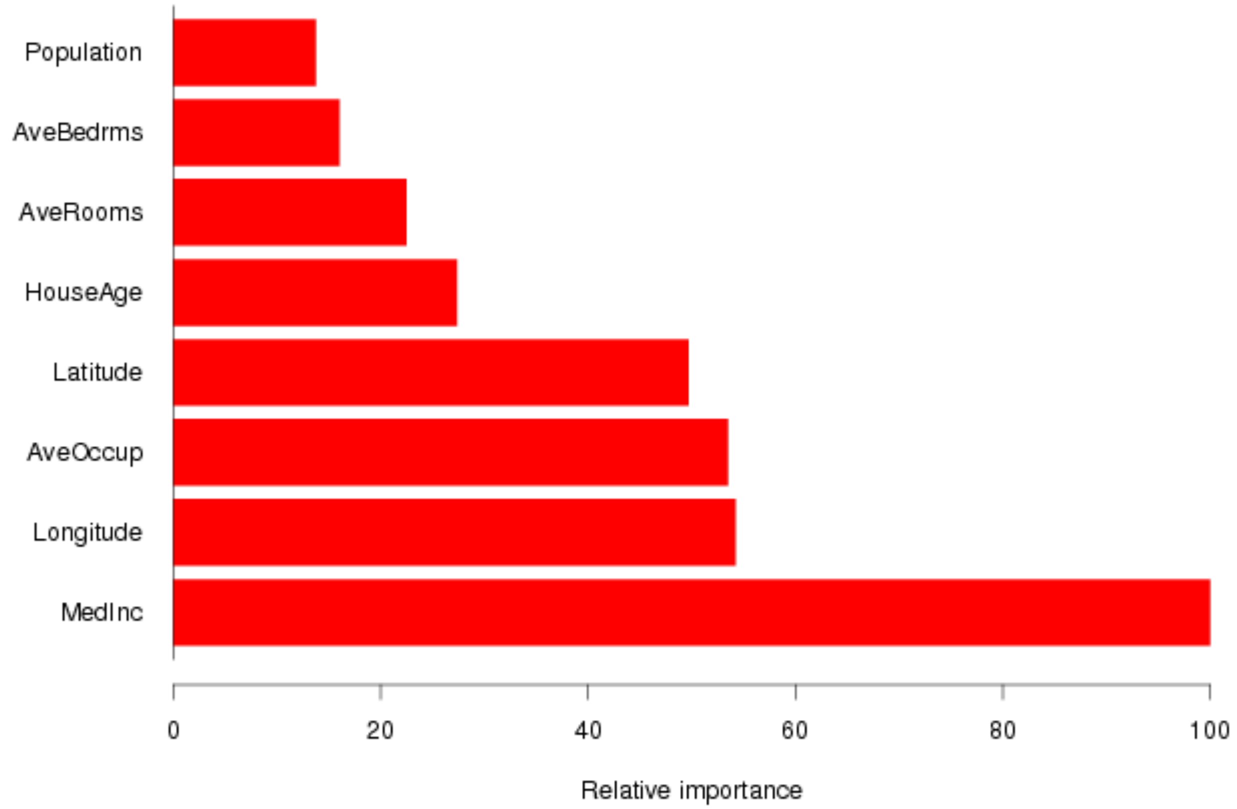
- Bagging typically improves the accuracy over prediction using a single tree, but it is now hard to interpret the model!
- We have hundreds of trees, and it is no longer clear which variables are most important to the procedure
- Thus bagging improves prediction accuracy at the expense of interpretability
- But, we can still get an overall summary of the importance of each predictor using Relative Influence Plots

Relative Influence Plots

- How do we decide which variables are most useful in predicting the response?
  - We can compute something called relative influence plots.
  - These plots give a score for each variable.
  - These scores represents the decrease in MSE when splitting on a particular variable
  - A number close to zero indicates the variable is not important and could be dropped.
  - The larger the score the more influence the variable has.

Example: Housing Data

- Median Income is by far the most important variable.
- Longitude, Latitude and Average occupancy are the next most important.



RANDOM FORESTS

Random Forests

- It is a very efficient statistical learning method
- It builds on the idea of bagging, but it provides an improvement because it de-correlates the trees
- How does it work?
  - Build a number of decision trees on bootstrapped training sample, but when building these trees, each time a split in a tree is considered, a random sample of *m* predictors is chosen as split candidates from the full set of *p* predictors (Usually $m \approx \sqrt{p}$)

Why are we considering a random sample of m predictors instead of all p predictors for splitting?

- Suppose that we have a very strong predictor in the data set along with a number of other moderately strong predictor, then in the collection of bagged trees, most or all of them will use the very strong predictor for the first split!
- All bagged trees will look similar. Hence all the predictions from the bagged trees will be highly correlated
- Averaging many highly correlated quantities does not lead to a large variance reduction, and thus random forests "de-correlates" the bagged trees leading to more reduction in variance

Random Forest with different values of "m"

- Notice when random forests are built using m = p, then this amounts simply to bagging.