# Advanced Topics in Statistics

*~ Bagging and Random Forests ~*

6

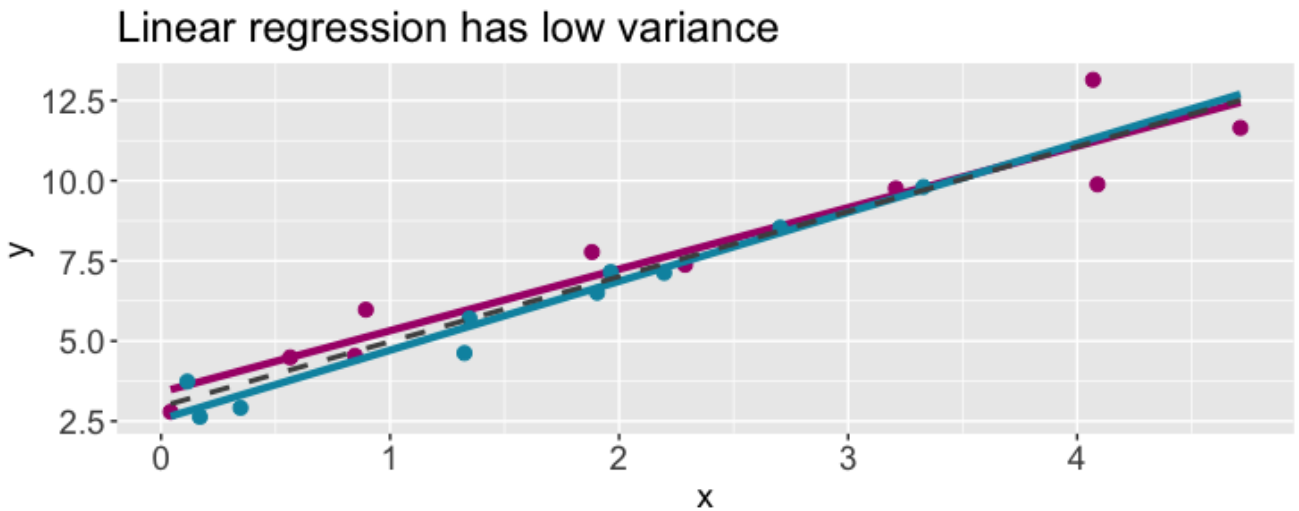
# Introduction.

Recall that despite all the advantages, decision trees suffer from high variance.

This essentially means that if we randomly split the training data into two parts, and fit decision trees on both parts, the results could be quite different.

In contrast, linear regression has low variance. If we split the data in half, fit a line over one half of the points and another through the other half of the points, the resulting lines are likely to be very similar.

In the figure below the two solid lines are the linear regression lines fitted to the two halves separately, while the dashed line is the regression line we would get if we used all the available data points.



# Reducing variance.

Ideally we would like to have models with low variance.

Recall that if we take a set of $n$ independent observations $Z_1, Z_2, \ldots, Z_n$ each with variance $\sigma^2$, then the variance of the mean $\bar{Z}$ is $\sigma^2/n$.

This suggests that a natural way to reduce variance and hence increase prediction accuracy is to take many training sets, build separate prediction models using each of these training sets and then take the average of the resulting predictions.

That is, if we had $B$ training sets, we could calculate $\hat{f}_1(x), \ldots, \hat{f}_B(x)$, then take the average

$$\hat{f}_{\text{avg}}(x) = \frac{1}{B} \sum_{i=1}^{B} \hat{f}_i(x)$$

to get a single learning model with lower variance.

The problem is that we don't have access to multiple training sets. To solve this problem, we can use bagging (**b**ootstrap **agg**regat**ing**).

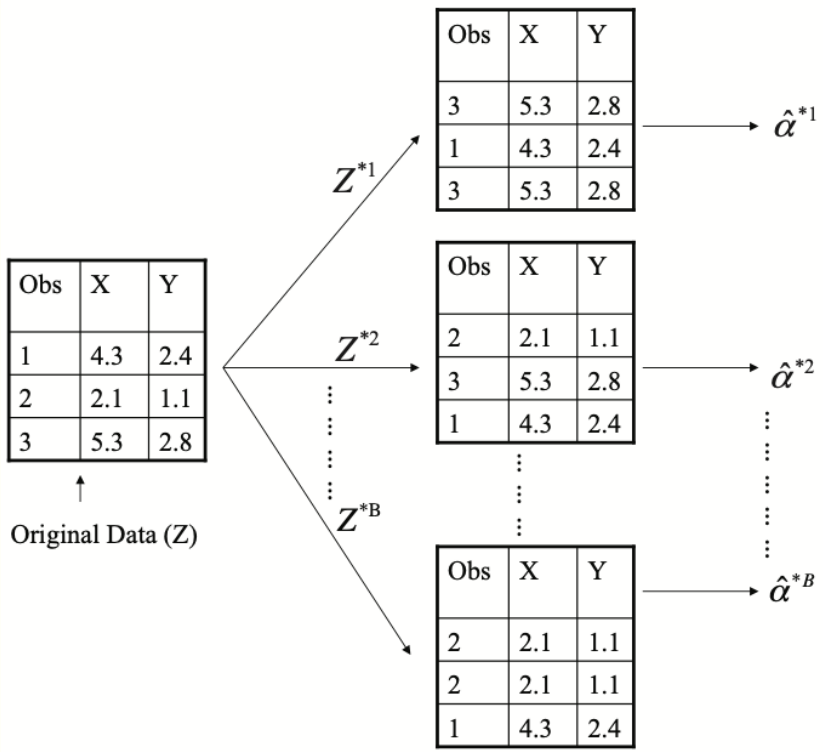
Recall that bootstrapping can be used to estimate quantities and obtain related uncertainty in the case when the sampling distribution is unknown. But here it is used in a different context.

# BOOTSTRAPPING.

Bootstrapping is a general purpose procedure to reduce variance.

The idea is to draw samples from the observed dataset with replacement, until this resampled dataset is of equal size to the observed dataset.

By using this method repeatedly we can obtain as many so called bootstrapped training datasets as we want.



# Bagging for regression trees.

Bagging is an extremely powerful idea based on two things:

1. Averaging: reduces variance!
2. Bootstrapping: gives plenty of training datasets!

How does averaging work?

Bootstrapping allows us to overcome the problem of not having access to multiple training datasets. Thus, the first step is to generate $B$ different bootstrapped training datasets.

Next, we train the statistical learning method on each of the $B$ training datasets, that is, for regression problems, we construct $B$ regression trees, and obtain a prediction using each tree.

Finally, when making prediction, we average all predictions from the $B$ trees.

Note that the trees are not pruned, but instead we grow them quite deep.

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.

When doing classification we construct $B$ classification trees using $B$ bootstrapped training datasets.

Computing an 'average prediction' is not as straightforward in the classification case.

There are two main approaches we can take when making predictions:

1. Record the class that each bootstrapped data set predicts and provide an overall prediction to the most commonly occurring one (majority vote).

   For example, if three trees predict label 2, one tree predicts label 1 and two trees predict label 3, then the majority vote would give label 2.

2. The classifier can also produce probability estimates, that is it can give the probability that a given observation has label 1, the probability that it belongs to class 2, and so on.

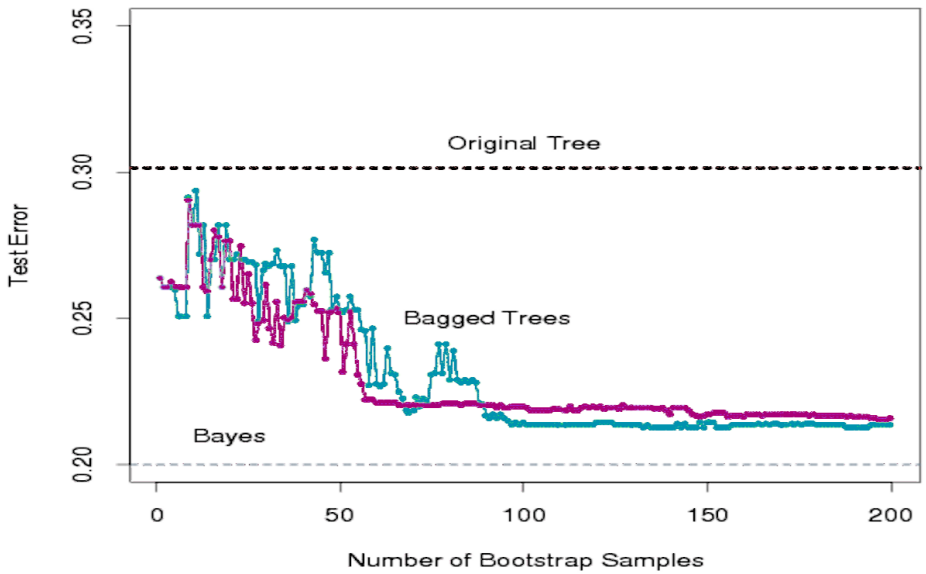
   Then we can just average the probabilities produced by the trees for each class, and when making prediction, assign the observation to the class with the highest probability.

As we will see both methods work well.

# BAGGING FOR CLASSIFICATION.

Here the **blue** line represents a simple majority vote approach, while the **pink** line corresponds to averaging the probability estimates.

We can see that both approaches do far better than a single tree (dashed black) and get close to the Bayes error rate (dashed grey).




We can generally do a better job when we are using more trees, but then after a certain point we are no longer gaining anything from using more and more trees.

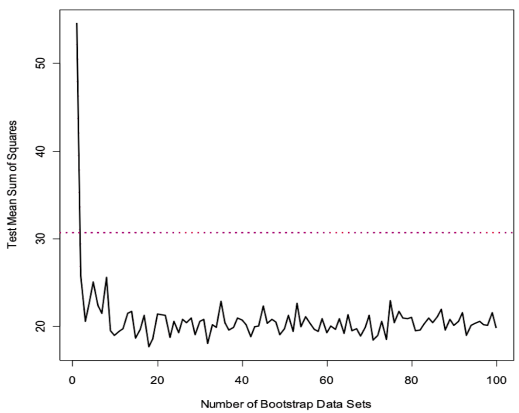
However we should note that we can't actually overfit with bagging. So the only disadvantage of using too many trees is that it might be computationally expensive.

# Bagging examples.

**Regression: Housing data**

The dotted line represents the test mean sum of squares using a single tree.

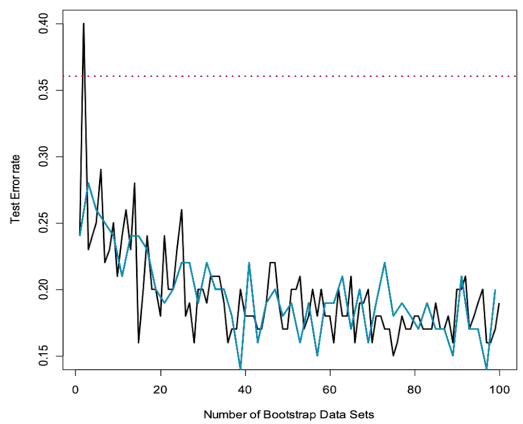
The solid line corresponds to the bagging error rate.




**Classification: Car seat data**

The dotted 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.




Notice that using one resampled tree gives a higher error than using the original dataset. This is because on average a bagged tree contains only 2/3 of the original dataset, so the prediction for a single bootstrapped tree is less accurate than the prediction made by a single tree that uses the whole dataset.

# Out-of-bag error estimation.

With bagging it is quite straightforward to estimate the test error rate.

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.

Recall that 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.

We call the remaining 1/3 of observations, out-of-bag observations.

- Thus, on average each observation is part of the training set for 2/3 of the bagged trees, and therefore the same observation is part of the test set for the remaining 1/3 of the bagged trees.
- So 1/3 of the time this observation is out-of-bag. In these cases we can predict for this out-of-bag sample, average these predictions to get a single prediction for this sample.
- Once we have a prediction for each observation, we can obtain an overall MSE or a classification error.

# VARIABLE IMPORTANCE MEASURE.

Bagging reduces variance and thus improves the accuracy of the prediction over using a single tree. It is also easy to obtain error rates.

But what are the disadvantages of bagging?

Intuitively we are averaging a large number of trees, where the resulting model can no longer be represented by a single tree. And as such, it is hard to interpret the model!

Also, since we have hundreds of trees, it is no longer clear which variables are most important to the procedure. Thus the improvement in prediction accuracy comes at the cost of interpretability.

It is not obvious at first which features are important in the model.

However, we can still get an overall summary of the importance of each predictor using Relative Influence Plots. These give a score for each variable, where this score represents the decrease in MSE when splitting on that particular variables.

- If this decrease is large then the variable is important, since using that results in a large improvement.
- On the other hand a number close to zero indicates that the variable is not important, therefore we can drop that form the model.

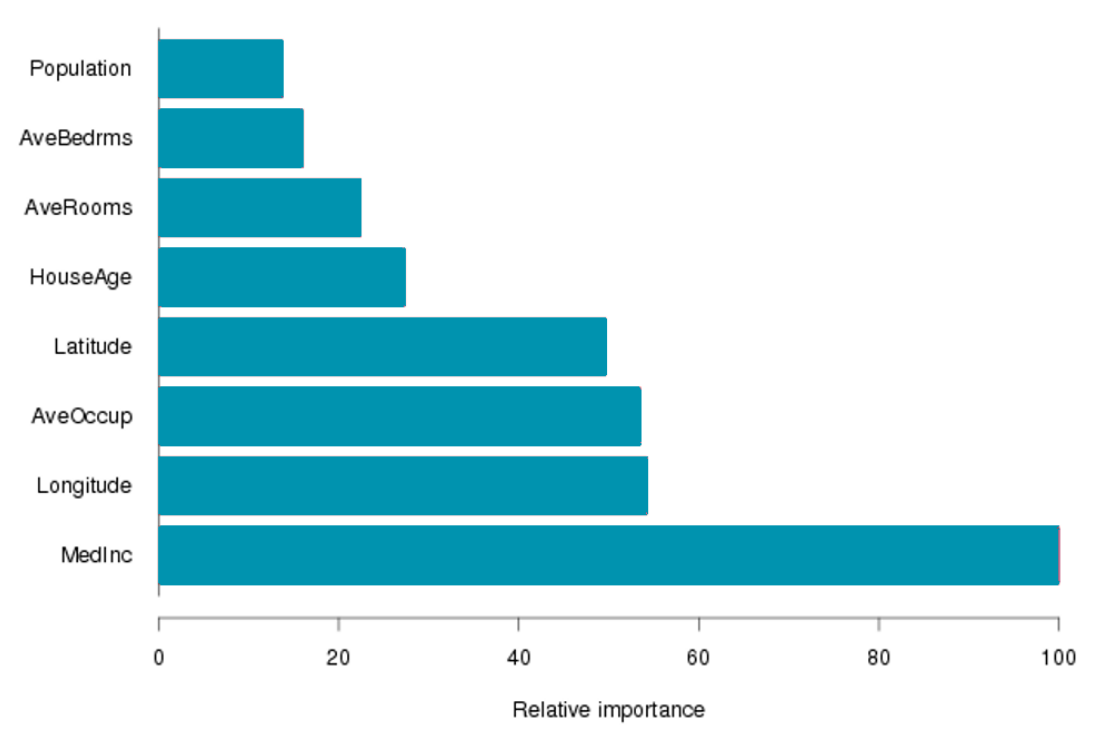
# Relative Influence Plot Example.

The example below shows that the score is always expressed relative to the maximum.

The median income is the most important variable, it gets the score 100.

The other scores show how important the other variables are relative to the median income variable.

Longitude, latitude and average occupancy seems quite important, while population and average number of bedrooms are less important predictors.



# Random Forests.

The problem with bagging is that the bootstrapped observations, and hence the generated trees are highly correlated.

Random forests build on the idea of bagging but they provide an improvement as they de-correlate the trees. How does it works?

- 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 does considering a random sample of $m$ predictors for splitting instead of all $p$ predictors help?

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!

Thus all bagged trees will look similar, and so all the predictions from the bagged trees will be highly correlated.

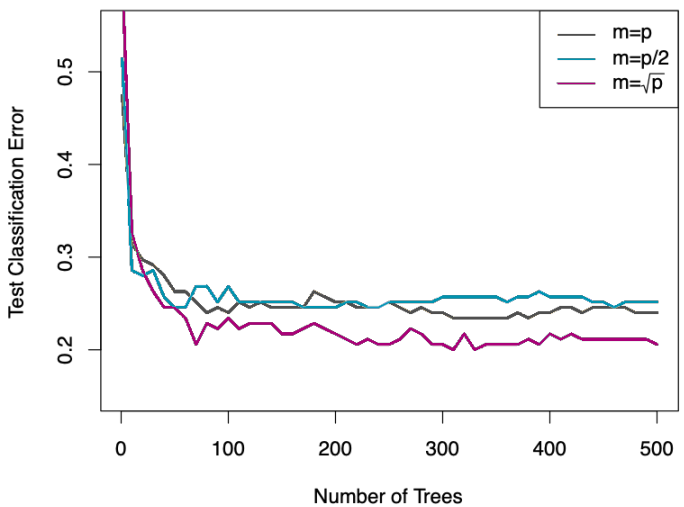
Averaging many highly correlated quantities does not lead to a large variance reduction, and thus random forests 'de-correlates' the bagged trees leading to more reduction in variance.

# Random Forests.

So a split is allowed to use only one of $m$ predictors, and it can't consider the other $p - m$ predictors.

The figure shows the test classification error of random forests using different values of $m$.

- When $m$ is $p$, so we are allowed to consider all the predictors, the result is simply what we would get by doing bagging.
- What we can see in the figure is that using $\sqrt{p}$ as $m$ results in reduction of the test error.
- Using $p/2$. however, doesn't really result in an improvement.




Usually using a small value of $m$ is desirable when we have a large number of correlated predictors.

We can't overfit by using a very large number of trees, so we usually choose a number that results in settled down error.

In this case around 200 trees should suffice.