Introduction to Data Science

RANDOM FORESTS

BRIAN D'ALESSANDRO

BOOTSTRAP AGGREGATING

We can also use bootstrapping to improve the underlying predictions.

This is done via a procedure called Bootstrap Aggregating, or BAGGING.

The Bagging Procedure:

1. Create a bootstrap sample of the data
2. Fit the model on the bootstrap sample from step 1
3. Make prediction on train/test data using model from step 2
4. Repeat this N times, storing each prediction of step 3
5. Make a final prediction by averaging the bootstrapped predictions

BAGGING (FORMALIZED)

The Bagging Procedure:

We have data $D = [(X_1, Y_1), (X_2, Y_2), \ldots, (X_N, Y_N)]$ and we want to learn: $E[Y|X] = \hat{f}(X)$

Define a bootstrap sample D^b as N samples from D, sampled with replacement.

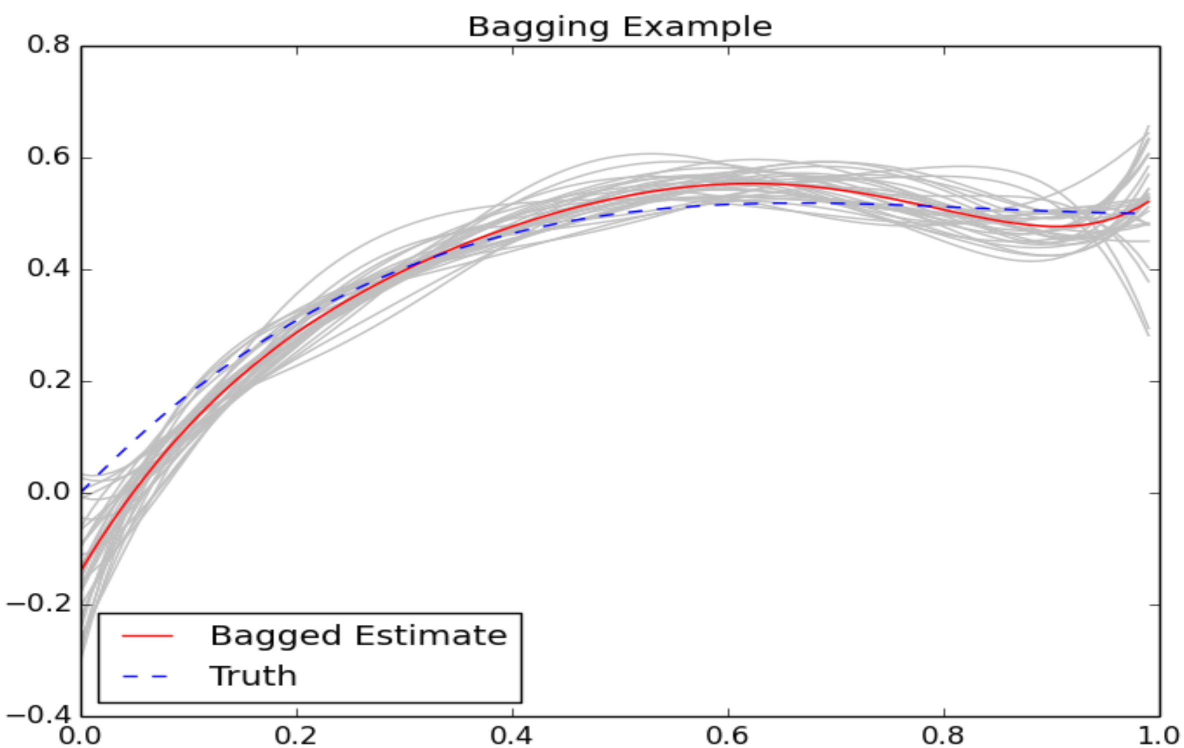
Let $E^b[Y|X] = \hat{f}^b(X)$ be the function learned from training set D^b.

Our bagged prediction is then the mean of all estimates of $f^b(X)$. I.e.,

$$\boxed{\hat{f}_{bag}(X) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}^b(X)}$$

BAGGING

Random Forests use a technique called Bagging (Bootstrap Aggregating). The idea of Bagging is learn N models off of N bootstrap samples and average the N models together.



WHEN TO USE BAGGING

According to the seminal paper on bagging by Leo Breiman, Bagging:

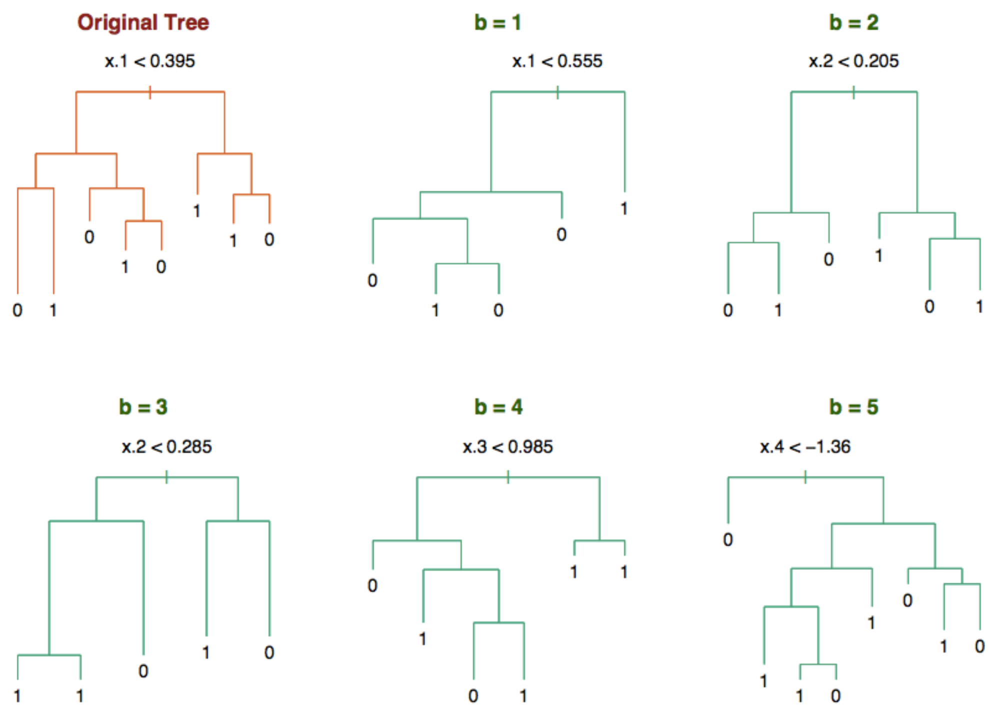
- ***"can push a good but unstable procedure a significant step towards optimality"***
- ***"can slightly degrade the performance of stable procedures"***

Q: What is an unstable procedure?

A: One in which small permutations of the training data result in dramatically different models

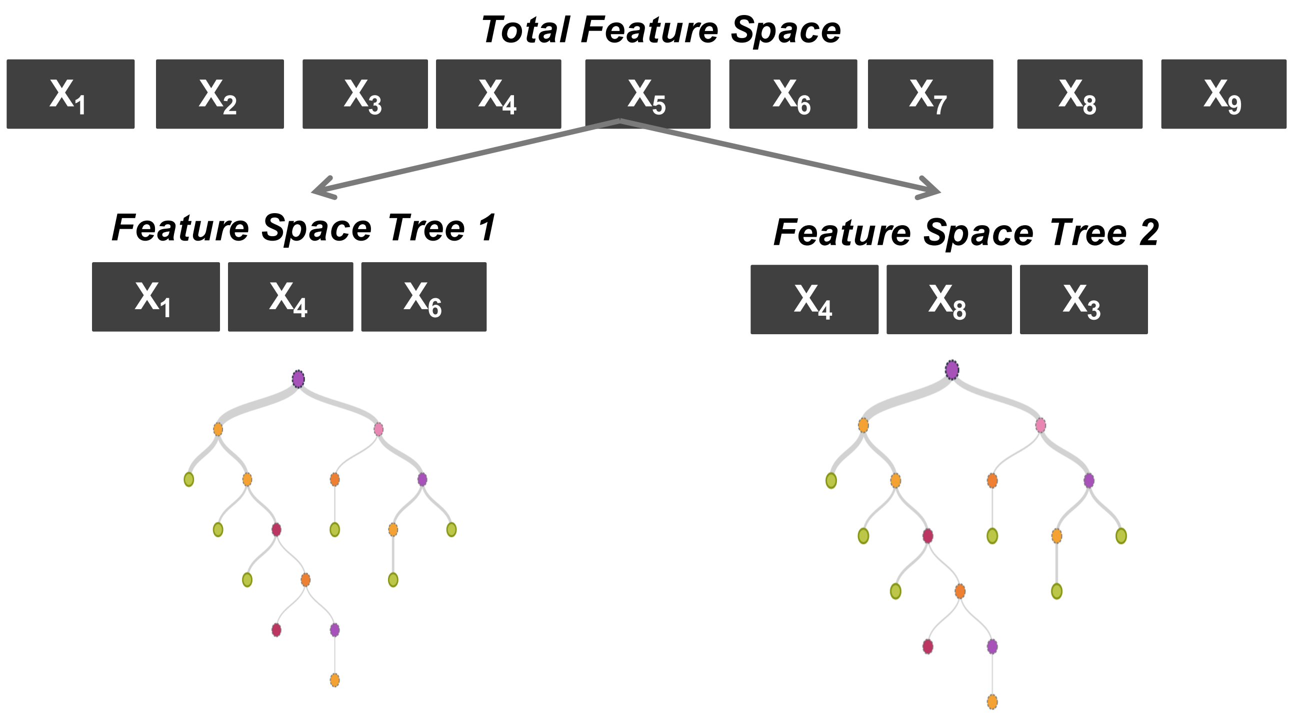
EXAMPLE UNSTABLE ALGORITHM

From ESL2: “…a decision tree on simulated data. Small variations of the data (generated by bootstrap sampling) produce wide variations in the learned tree.”



DE-CORRELATING INDIVIDUAL TREES

For Bagging to work best, individual estimators should be as uncorrelated as possible. The RF algorithm accomplishes this by using a subset of randomly chosen features for each tree iteration.



THE RF ALGORITHM

The RF procedure is straightforward and easily parallelized.

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 bth random-forest tree. Then $\hat{C}_{\text{rf}}^B(x) = \textit{majority vote}\ \{\hat{C}_b(x)\}_1^B$.

Source: ESL2

WHY IT WORKS

Bias

A decision tree is unstable, and has high variance, but can also have extremely low bias

- Can detect all manner of interaction effects
- Especially when allowed to grow very deep

The bias of the average of identically distributed trees is equal to the bias of the individual trees (in this case, very low).

Variance

If the variance of an individual tree is σ^2 and the pairwise correlation of any two trees is ρ, then the variance of the forest is:

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

Randomly sampling features reduces the pairwise correlations ρ and reduces the 1st term above, while bootstrapping reduces the 2nd term above. Note though, that reducing the features decreases total variance but also increases the bias.

Source: ESL2

TUNING

RF's are quick to set up and might do fairly well straight out of the box.
But nonetheless, tuning is always recommended.

Forest Level Parameters

- # trees (n_estimators) – increasing this decreases variance, but increases training time.
- # of features to sample (max_features) – the number of features sampled in each tree. Reducing this # increases the bias but decreases the RF variance.

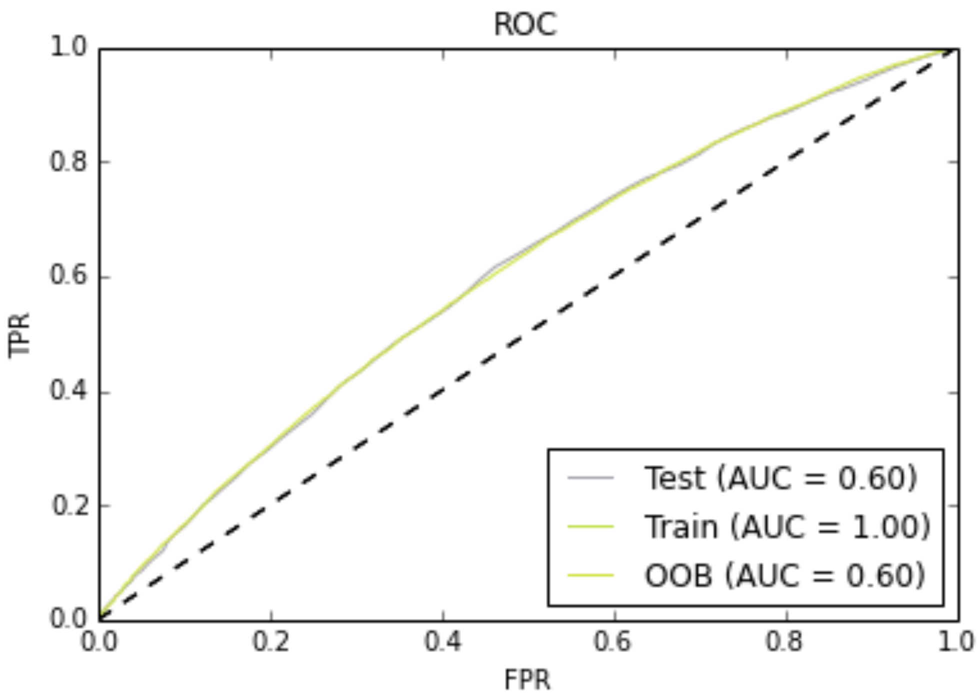
Tree Level Parameters

- # intermediate nodes (max_depth) – the size of the tree. Usually you don't want to limit this (i.e., set max_depth=None)
- Size of intermediate notes (min_sample_split) – the number of instances in an intermediate node, before splitting (usually good to set to 1).

We usually want to over-fit the individual trees, and as always, use some hold-out method to optimize Forest level parameters.

OUT-OF-BAG ERROR

Tuning via cross-validation can be very slow due to the number of trees that have to be built. Random Forests have a built in validation property called “out-of-bag” error estimation that can help us avoid x-validation. For each record, average $f(x_i, y_i)$ on all bootstrap iterations in which record i was not sampled. This out-of-bag prediction can then be used for out-of-sample error estimation.



Observations:

- Training AUC is 1!
- OOB AUC is almost = Holdout AUC