Random Forest

Decision Trees are good, but you know what would be better? a bunch of them.

Now, different decision trees wouldn't really learn anything different if trained in the exact same manner, they would also take quite some time if we are training a lot of them.

Enter Bagging

Now we would give each decision tree only a fraction of the data, not just that we would also limit the number of features at each node when choosing the best split.

Advantages

1. For a rather large dataset, which is too large to train a single classifier
2. Decision trees only have linear boundaries, but random forest can have arbitrary boundaries.

Algorithm

1. Select a random subset of data to train each tree
2. while training each tree select m random features at each node to find the best split, m can be calculated by Breiman's bagger which is typically one of $0.5\sqrt{m}, \sqrt{m}, 2\sqrt{m}$

The trees are not prunned or anything since they are already weak learners wouldn't really overfit.

Why Bagging works?

Bagging is just like a group of people trying to collectively come at an answer by individual voting, you would expect that more number of people would be right. Also, the classifiers are not very correlated since decision tree structure depends a lot of training set.

Features

1. One of the top performing models on Kaggle
2. Hard to Overfit
3. Runs efficiently on large data

Questions

1. Is Correlation between individual trees good?
2. Why does bagging work?
3. Why should you not prune individual trees?

4. What is OOB (out of bag error)?