Lecture 9: Ensemble Learning and Random Forests

CS 167: Machine Learning

Ensemble Learning

Discuss: What do you do if different machine learning algorithms make different predictions on the same data?

Ensemble Learning: using multiple learners/hypotheses for coming up with predictions - often performs better than using one algorithm alone

Random Forests

Random Forests is an effective learning algorithm that uses an *ensemble* of decision trees.

Basic idea: build a bunch of trees and have them vote on the prediction

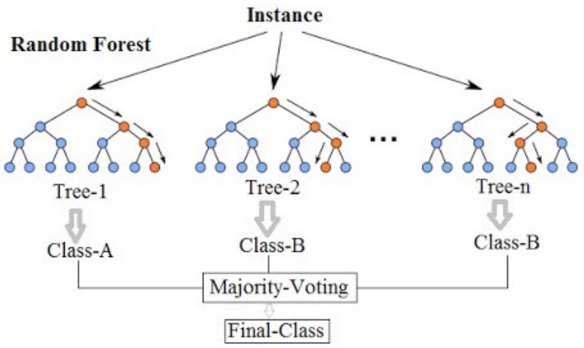

image credit: https://www.youtube.com/watch?v=ajTc5y3OqSQ

Good resource: https://en.wikipedia.org/wiki/Random_forest

Sampling

Before we get into the details, a brief diversion...

Discuss: we want to predict an election by calling a sample of the voters and asking who they're going to vote for. What should be true about your sample if you want to get good results?

Something you don't normally do when sampling: **sample with replacement** - allow the same thing to be picked twice.

image credit: http://www.troutbum2.com/catch-and-release-fly-fishing/

Bagging

Each tree is built using a different variation on the data set.

The technique it uses to get the variations on the data set is called **bagging**, which is short for **bootstrap aggregating**.

Training: Starting with n training examples
Do this B times to create B trees:

1. Create a random, size-n sample (with replacement) of the training set
2. Train a decision tree on that example
   - can still prune, stop early, etc.

Predicting:

1. Get prediction from each of the B trees
   - Classification: return most common prediction from the B trees
   - Regression: return the average (mean) prediction from the B trees

Why is this better?

- A single tree is sensitive to noise
- trees trained with different sets are less *correlated*
- sensitivity averages out
  - decreases variance

What would happen if you used the same training set for all B trees?

How many trees in the forest?

How big should B be, i.e., how many trees should I train?

- as always, depends on the data
- depends on how much time you have
- it's a parameter you can mess with until you find an optimal value

`scikit-learn`: default is 10

hundreds or thousands is common

Random Forests Algorithm

One more detail: **Random Forests** algorithm also uses a *random subset of the attributes* for each tree.

the size of these subsets should also be tweaked for optimal performance

Usually,

$$\text{classification: } \sqrt{\#attributes}$$

$$\text{regression: } \#attributes/3$$

side benefit: attributes that are used by more trees must be important - you can find out which things the learning algorithm things are important