Random Forests

Jones And His High MPG Cars . . .

Jones is a car collector who has moved to a new planet and is looking for cars to collect. In choosing which car he wants, his only desire is on one thing: quickly buying new cars which have an excellent fuel efficiency (How many miles the car can run on one gallon of fuel). Unfortunately, this planet is without easily accesible data on cars fuel efficiency and without used car salesmen to easily give us all the details on the cars at the dealership. We'll have to develop an algorithm to quickly estimate the cars miles per gallon.

After spending some time looking at algorithms, Jones has found that the decision tree is a great way to estimate the cars fuel efficiency given a features readily available at the dealership, such as # of engine cylinders, model year, and weight.

For example, after going between a few dealerships, he calculates that first checking if the weight is under 3000 pounds, then checking if the model year is newer than 2005, then if it has less than 3 cylinders, and finally if the displacment is greater than 3000 cubic centimeters. Which works on a few cars, but he finds that trying to create his own decision tree from only his decisions leads to overcomplex sequences of decisions that don't generalize well between different car dealerships.

He realizes that if he enlists a few of his henchmen to develop their own decision trees for each car dealership and everyone average their results, that should work better than solely doing it alone.

This is the intuition behind the power of Random Forests over Decision Trees.

We've covered decision trees before in the previous note, but let's review a few of the mechanics of a decision tree and see how they compare to Random Forests.

Gini Impurity Index or Information Gain

Say we had a dataset of 30 green cars and 70 red cars. From our dataset, we would randomly classify a car as being green 3/10 of the time and red 7/10 of the time.

If we were to randomly pick an item in our dataset and then randomly classify it according the class distribution of our dataset. *The Gini Impurity Index* measures the proablity we would correctly classify this data point.

The Gini Impurity Index is defined as:

$$G = \sum_{i=0}^{c} p_i * (1 - p_i)$$

Given, C is the number of classes and p_i is the probability of picking a datapoint with class i.

Remember that these probilities for each class p_i all sum up to 1.

Random Forest

Contain an *ensemble* of *n* decision trees and *m* boostrap samples, which stop splitting after reaching a specified depth *d*.

Random Forests Algorithm

1. At the current node, select p features randomly from the available features D.

2. Using the specified splitting metric, compute the ideal split point for tree*i* of *n*. Split this node into daughter nodes, which have various subsets of features from their parent node.
3. Repeat the above steps until a the specified tree depth*d* has been reached.
4. For each tree *i* of *k* in the forest, repeat steps 1-3
5. Aggregate or vote on the outputs calculated by each of the trees.

The number of trees in the ensemble*k* is typically very large, in the magnitude of*hundreds to thousands.*

Advantages and Weakness of Random Forests

!! Images to Use In Slides and Notes !!
– Decision Tree Divying Up Information

– A Random Forest is multiple decision trees

–