Random Forest Cheat Sheet

Yan Zeng, v1.0, last updated 2017-05-25

Abstract

Cheat sheet based on (Hartshorn, 2016).

Algorithm of Constructing a Random Forest

1. Determine how many trees to grow. Using 100 trees in the RF is a good place to start. To set the number of trees to n:

   ```
   model = RandomForestClassifier(n_estimators=n);
   ```

2. For each tree, generate it as a decision tree on a data set bootstrapped from the original data set, with the same size as the original data set.
   1) At each branch, use a randomized selection of features to decide on how to split.
      a) By default, a RF will use the square root of the number of features as the maximum features that it will look at on any given branch.
      b) The decision tree evaluates the chosen multiple features and picks the best location in all the features that it looks at when deciding where to make the split.
      c) The criterions for "the best location" include the Gini criteria and the Entropy criteria. They make very little difference and in Python, the default is the Gini criteria.
   2) Keep on mind the following stopping parameters while continuing the splits:
      a) `max_depth`: the maximum number of depths allowed to build a tree.
      b) `min_samples_split`: the minimum number of data points required in a branch to continue the split.
      c) `min_sample_leaf`: the minimum number of data points required in each leaf to continue the splits.
      d) `max_leaf_nodes`: the maximum number of (supposedly best) leaf nodes.

Calculate the Out of Bag (OOB) Error

1. For every data point in the original data set, use each of the trees in which the data point is an OOB to make prediction and we average the prediction errors.
2. Average the OOB errors of all data points in the original data set to get the OOB error of the RF.

Calculation of Feature Importance

1. Information gain.
2. OOB.

Bibliography

Hartshorn, S. (2016). *Machine Learning with Random Forests and Decision Trees.* Amazon Kindle.