Ensembles: Bagging and Random Forests

Objectives

Afternoon

- What is the **"Out-of-Bag" (OOB) Error**?
  - How it is calculated? What it is an estimate of?
- Explain how to get **Feature Importances** from a random forest and what those importances mean
- General comments on random forests
- Paired Assignment
  - Random forests with *sklearn*

Random Forests in sklearn

The following is an example of how one can implement a Random Forest using sklearn:

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error
from sklearn.cross_validation import train_test_split

y = df.pop('target').values
X = df.values

X_train, X_test, y_train, y_test = \
    train_test_split(X, y, test_size=0.33, random_state=42)
```

Random Forests in sklearn

```
rf = RandomForestRegressor(n_estimators=100)
rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)
print(mean_squared_error(y_test, y_pred))
```

Out-of-Bag Error

- One nice side effect from using bootstrapped models is we can easily estimate the test error of a bagged model without the need to perform cross-validation!
- Each tree in our forest has only seen 2/3 of the observations of our training data
- We can thus measure the test error of a particular tree by running the remaining out-of-bag (OOB) observations through the random forest

```
rf = RandomForestRegressor(n_estimators=20, oob_score=True)
rf.fit(X, y)

print rf.oob_score_
```

Out-of-Bag Error

- OOB error is a valid estimate of the test error since the observation is predicted using models that did not use that observation
- OOB approach is convenient when bagging on large data sets for which cross-validation would be computationally intensive
  - Cross-validation is still needed for methodology comparisons. . .

Randon Forest Error

- Overall forest error rate depends on two things:
  - The *correlation* between any two trees in the forest
  - The *strength* of each individual tree in the forest
- Reducing m reduces both the correlation and strength, while increasing it increases both
- Another trade-off, use OOB error to find the optimal value of m

Feature Importance

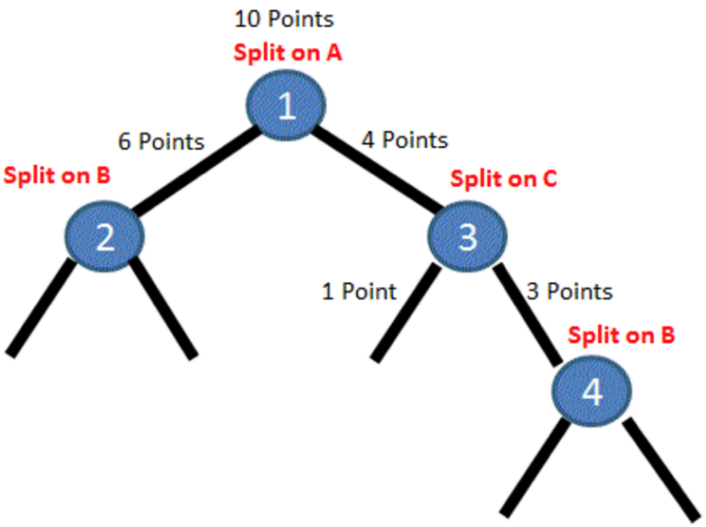
- When you have many features, you want to know which are the most important
  - For recommendations
  - For model building
- However, when we bag a large number of trees, it is no longer possible to represent the resulting statistical learning procedure using a single tree and it is difficultcult to ascertain which features are most important
  - We can, however, obtain an overall summary of the importance of each predictor
  - Classification Trees: record total amount Gini index decreases due to splits over given predictor, averaged over all B trees
    - ★ Larger value indicates 'higher importance'
  - Regression Trees: record total amount RSS decreases due to splits over a given predictor, averaged over all B trees
    - ★ Larger value indicates 'higher importance'

Feature Importance - Method #1

The following is the method used in `sklearn`:

- The higher in the tree the feature is, the more important it is in determining the final prediction of a data point
- The fraction of data points that reach a node is used as an estimate of that feature's importance for that particular tree
- Average those values across all trees to get the feature's importance
- The higher the average fraction of observations that were sent through the feature, the more important

Feature Importance - Method #1 Example



Feature Importance

Split #	# of Data Points	Split on Which Feature	Information Gain	# of Nodes * Information Gain
1	10	A	0.26	2.6
2	6	B	0.4	2.4
3	4	C	0.3	1.2
4	3	B	0.1	0.3

Importance

Feature	Importance	Normalized
A	2.60	0.40
B	2.70	0.42
C	1.20	0.18

Figure 1:Feature Importance

Feature Importance - Method #2

Random permutation method (Breiman)

- To evaluate the importance of the j^{th} variable
  - When the b^{th} tree is grown, the OOB samples are passed through the tree
    - ⋆ Compute accuracy
  - Values of the j^{th} variable in the OOB samples are randomly permuted
    - ⋆ Compute new (lower) accuracy
- Average the decrease in accuracy over all the trees

The larger the average decrease in accuracy, the more important the feature

Intuition

- The features that have the greatest increase in OOB error when permuted are the most important
  - If you can set a feature's value to basically anything and not change the OOB error, that feature is not important

Feature Importance

- Feature importances are almost always put forth as normalized values. What is important is that we can compare features to other features.
- Typically, the more features you have in a random forest, the less important any individual feature will be
- The original authors of randon forests state that you should be interested in rank, not magnitude
- Highly correlated features tend to split importance

How to Handle Categorical Data

- String values need to be converted into numeric
- If possible, convert categorical variables into continuous variables
  - E.g., S, M, L to size in actual weight or height
- **DO NOT DROP ONE OF THE CATEGORIES!**
- `sklearn` doesn't support splitting on multiple features

Some Advantages of Bagging/Random Forests

- All trees are trained independently
  - possibly in parallel reducing computation time
- Can handle thousands of input variables without variable deletion
- Gives estimates of which features are important in the classification
- If the number of variables is very large, forests can be run will all variables, then run again using only the most important from the first run