Trees and Random Forests

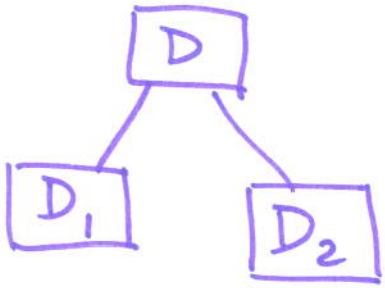
- We now examine some modern, computationally intensive methods for regression and classification.
- Regression trees are used when we have one response variable which we want to predict/explain using possibly several explanatory variables.
- The goals of the regression tree approach are the same as the goals of multiple regression:
  * Determine which explanatory variables have a significant effect on the response.
  * Predict a value of the response variable corresponding to specified values of the explanatory variables.

- Regression trees are a method that is more algorithm-based than model-based.
- We form a regression tree by considering possible divisions of the data into partitions based on the value of one of the predictors:

Example: For data D, let

D_1 = data for which $X_1 < 12$

D_2 = data for which $X_1 \geq 12$



- Calculate the mean of the responses in each partition set.
- Compute the residual sum of squares for this partitioning:

$$RSS_{partitioning} = RSS_{part1} + RSS_{part2}$$

$$= \sum_{i \in part1} (y_i - \bar{y}_{part1})^2 + \sum_{i \in part2} (y_i - \bar{y}_{part2})^2$$

- Of all possible ways to split the data, pick the partitioning that produces the smallest RSS.
- Continue the algorithm by making subpartitions from the most recent splitting.
- The result is a treelike structure subdividing the data.
- This also works well when a predictor is categorical – we can divide the data based on the categories of the predictor.
- Splitting on one variable separately within partitions of another variable is essentially finding an interaction between the two variables.
- The usual regression diagnostics can be used — if problems appear, we can try transforming the response (not the predictors).

- Eventually we will want to stop splitting and obtain our final tree.
- A criterion to select the "best" tree is the cost-complexity:

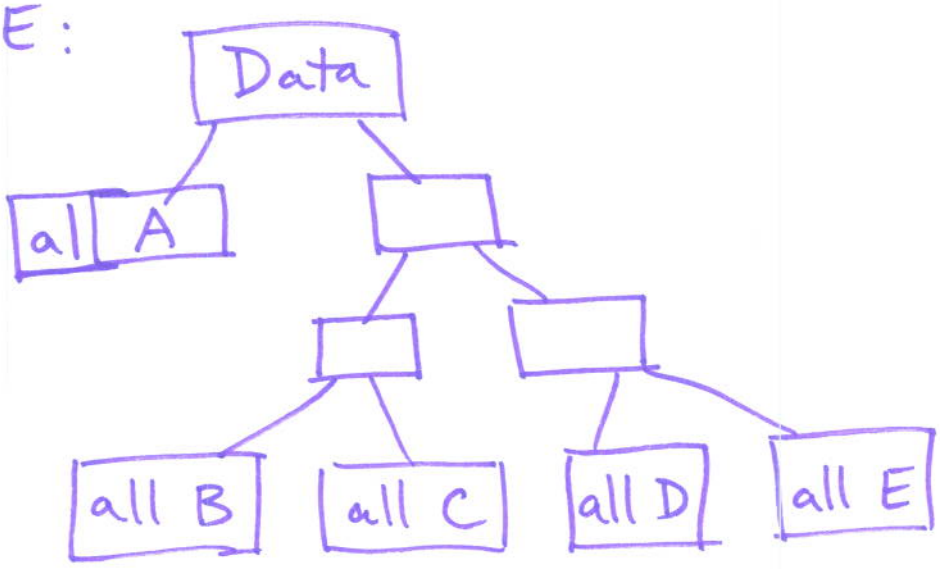
$$CC(\text{Tree}) = \sum_{i \in \{\text{terminal nodes}\}} RSS_i + \lambda(\#\text{ of terminal nodes})$$

- The first piece measures fit and the second piece penalizes an overly complex tree.
- Another approach to tree selection is <u>cross-validation</u>:
  - We select a random subset of the data, build a tree with that subset, and use the tree to predict the responses of the remaining data.
  - Then a cross-validation prediction error can be calculated: A tree with low CV error is preferred.

- The 'rpart' function in the 'rpart' package of R produces regression tree analyses.
- See example with Boston housing data: A plot of the graph of the tree reveals the important variables.
- More (or less) complex trees may be obtained by adjusting the 'cp' argument in the 'prune.rpart' function.
- The 'cp' value is directly proportional to λ, so a larger value of 'cp' encourages a simpler tree.
- The 'plotcp' function can guide tree selection by plotting CV error against 'cp': We look for the elbow in the plot.

- Classification Trees work similarly when the response is categorical (or discrete).
- Instead of using RSS as a criterion to guide the splits, a "node purity" criterion is used.
- With classification trees, the goal is that the terminal nodes should classify observations into the correct groups.
- Ideally, we set up the tree so the class types within a particular split are all of one kind:

Example: Data having 5 known groups, A, B, C, D, E:



- This ideal situation does not usually occur in reality.
- The node purity measures how close a node is to having observations of only one category.
- There are several possible measures of node purity. The 'rpart' function by default uses the Gini Index:
  - If p_{ik} = proportion of sample observations assigned to node i having category k, then the Gini Index is $\sum_i D_i$, where for node i, $D_i = 1 - \sum_k p_{ik}^2$
- In the "ideal" case when all nodes are perfectly pure, the Gini Index is minimized (at zero).
- See example with hsb data.

Random Forests

- The random forest approach is an ensemble method — it generates many individual classifiers/predictions and aggregates them to produce a better overall method.
- As the name suggests, a random forest consists of many trees.
- It relies on the principle of bagging (bootstrap aggregating) proposed by Leo Breiman.
- Different trees are constructed using n_{tree} bootstrap resamples of the data, and the nodes are split based on random subsets of predictors, each of size m_{try}.
- Classification of new observations is done by majority vote among the constructed trees.

- In regression, prediction is done by averaging predicted response values across the predicted trees.
- The error rate is typically assessed by predicting out-of-bag (OOB) data — the data not chosen for the bootstrap sample — using each constructed tree.
- The 'randomForest' function in the 'randomForest' package will obtain a random forest, for either regression (continuous response) or classification (categorical response).
- It also provides a measure of which explanatory variables are most important.
- See examples on course web page.