Chapter 1

1.1 Trees and Random Forests

Trees

Decision trees may be familiar to many with a background in the social or medical sciences as convenient ways to represent data and can assist in decision making. Morgan and Sonquist (1963) derived a way for constructing trees motivated by the specific feature space of data collected from interviews and surveys. Unlike, say agricultural data which involves mostly numerical variables like rainfall, the data collected from interviews is mostly categorical. On top of this issue, the datasets Morgan and Sonquist dealt with had few participants (n) and much data collected on them (p). To continue with their list of difficulties, there was reason to believe that there were lurking errors in the variables that would be hard identify and quantify. Lastly, many of the predictors were correlated and Morgan and Sonquist doubted that the additive assumptions of many models would be appropriate for this data. Morgan and Sonquist noted that while many statistical methods would have a difficult time accurately parsing this data, a clever researcher with quite a lot of time could create a suitable model simply by grouping values in the feature space and predicting that the response corresponding to these values would be the mean of the observed responses given the grouped conditions. Their formalization of this procedure in terms of "decision rules" laid the ground work for future research on decision trees.

In 1984, Breiman et al introduces a revolutionary new algorithm for trees. **Need to acquire** *Classification and Regression Trees* **to make sure the method discused in MASS is the same that Breiman uses/is used in** `randomForest`

Tree Algorithm CART?

To get any further on this topic, we must develop the framework behind trees, namely the splitting algorithm. In the regression case, which is what is considered here the algorithm looks like this:

Begin by considering the entire feature space $X_1, ..., X_n$. Then:

1. Consider every possible pair of partitions of this feature space, P_1, P_2, so that if $X_1 = x_1, X_2 = x_2, ..., X_n = x_n$ where $x_1, ..., x_n \in P_1$ then our prediction is the mean value of y given $x_1, .., x_n \in P_1$.
2. Choose the partitions that minimize RSS
3. For each new partition, repeat steps 1 and 2 until some stopping condition is reached.

An alternative to this method is conditional inference trees. As noted by ______, CART often overfits and is biased towards variables with many possible splits. This mehtod implemented by Hothorn et al https://cran.r-project.org/web/packages/partykit/vignettes/ctree.pdf uses theory from the field of permutations. Their method is:

1. For case weights w test the global null hypothesis of independence between any of the m covariates and the response. Stop if this hypothesis cannot be rejected. Otherwise select the j_{th} covariate Xj with strongest association to Y.
2. Choose a set $A \subset X_j$ in order to split X_j into two disjoint sets A and $X_j \; A$. The case weights w_{left} and w_{right} determine the two subgroups with $w_{left,i} = w_i I(X_{j \cdot i} \in A)$ and $w_{right,i} = w_i I(X_{ji} \in A)$ for all $i = 1, ..., n$ (I(ů) denotes the indicator function).
3. Recursively repeat steps 1 and 2 with modified case weights w_left and w_right, respectively.

from https://eeecon.uibk.ac.at/~zeileis/papers/Hothorn+Hornik+Zeileis-2006.pdf

After step 1 is completed, any goodness of fit method can be used to generate the split and choose the set A. Note that in this method the splitting is done seperately from the variable selection.

Random Forests

There is a limit to the predictive capabilities of a single tree; they suffer from high variance. To alleviate this, random forests are often used instead. They function by enlisting the help of many trees, and then aggregating the responses over all of them.

- history
- algorithm
- uses

1.x What We Mean When We Talk About Inference

- Inferential vs Descriptive
- Frequentist vs Bayesian

1.x Permutations and Populations

As stated in the introduction of the *Chronical of Permutations Statistical Methods* by KJ Berry et al, 2014, there are two models of statistical inference. One is the population model, where we assume that the data was randomly sampled from one (or more) populations. Under this model, we assume that the data generated follows some known distribution. “Under the population model, the level of statistical significance that results from applying a statistical test to the results of an experiment or a survey corresponds to the frequency with which the null hypothesis would be rejected in repeated random samplings from the same specified population(s)”, (Berry et al, 2014).

The permuation family of methods, on the other hand, only assumes that the observed result was caused by experimental variablility.

1.x Inference on Random Forests

The Problem

Random forests create models with great predictive-, but poor inferential capabilities. A single tree is simple to i ### Proposed solutions to this problem

Statisticans Leo Breiman and ____ Cutler proposed a method of permuted variable importance that hoped to answer this problem. Their method compares the variable importance for each variable in a tree-wise manner. For each tree, the permuted variable importance of the variable X_j is:

$$PV^t(x_j) = \frac{\sum_{i\in|B|} y - \hat{y}^t}{|B|} - \frac{\sum_{i\in|*B|} y - *\hat{y}^t}{|*B|}$$

Where B is the matrix represetning the feature space, $|B|$ is the number of observations, $*B$ is the matrix of predictors but with X_j permuted, $\hat{y}$ is the predicted outcome, and $*\hat{y}^t$ is the predicted outcomes after variable X_j has been permuted. This value is averaged over all the trees. It’s important to note that if the variable X_j is not split on in the tree t, the tree-wise variable importance will be 0.

Creating a permutation-based method is certainly an attractive solution to our problem. One, it allows us to estimate the distribution of variable importance and gernerate a Z score under the null hypothesis that $PV = 0$.

$$PV(x_j) = \frac{\sum_1^n treePV^t(x_j)}{\frac{\hat{\sigma}}{\sqrt{ntree}}}$$

Strobl et al from the University of Munich criticize this method in their 2008 technical report, *Danger: High Power! – Exploring the Statistical Properties of a Test for Random Forest Variable Importance*. One, this method has the downside of increasing power with increasing numbers of trees in the forest. This is a more or less arbitrary parameter which we would hope would not affect our importance estimates. Secondly, the null hypothesis under Breiman and Cutler's strategy is that the variable importance V for any variable X_j is not equal to zero given Y, the response. Because random forests are most often used in situations with multicolinearity that would make other methods like the linear model difficult, Strobl argues that any variable importance measure worth its salt should not be mislead by correlation within the predictors.

The researchers at the University of Munich published a fully fleshed response to the Breiman and Cutler method in 2008, titled *Conditional Variable Importance for Random Forests* that address these issues. Strobl et al propose restructuring the Breiman and Cutler algorithm to account for conditional dependence among the predictors. Their algorithm looks like this:

1. Fit a random forest to the model, R_0, and calculate base variable importance for each variable V
2. For every predictor $X_j \in X_1, ..., X_n$:

- 2a. Conditionally permute X_j given the splits found in R_0
- 2b. Fit a new random forest R_j with the permuted data
- 2c. Calculate a new variable importance $\hat{V}_j$

3. For every variable $X_1, ..., X_n$,
$$CV(X_j) = \hat{V}_j - V_j$$

The null hypothesis is that $CV(X_j) = 0$ given the predictor Y *and all other predictors* $X_1, ..X_n$.This accounts for interactions between X_j and the other predictors. Using the simulated data from the previous example, here's an implementation of the algorithm outlined here as it is in the `party` package.

This paper aims to provide a response to this method. One the conditional permutation algorithm is notoriously slow with any dataset of a size that is appropriate for a random forest. Two, the partitions are made from the random forest corresponding to the formula of Y $X_1, ..., X_n$ instead of a model of X_j $X_1, ..., X_n$.