Regression Tree

- RSS(split) = $\sum_{i\ in\ R_1}(Y_i - \bar{Y}_1)^2 + \sum_{i\ in\ R_2}(Y_i - \bar{Y}_2)^2$, RSS(full data) = $\sum_{i=1}^{n}(Y_i - \bar{Y})^2$
- If a categorical explanatory variable is present, optimal search is across levels ordered according to mean Y at each level (1 split for each level)
- Can keep splitting data (can get single observation in each node) but may have overfitting, use pruning to improve tree

Classification Tree

- Choosing "best" split not using misclassification rate, but rather **Gini Index/deviance**
- n_t = number of observations in node t
- p_{t_k} = proportion of observations in node t from class k = $\frac{n_{t_k}}{n_t}$
- Gini Index: $\sum_{k=1}^{K}\widehat{p_{t_k}}(1 - \widehat{p_{t_k}})$, across the proposed split
    - Pure node Gini index is 0, and maximum value is $\frac{K-1}{K}$
- Cross-Entropy/Deviance: $-\sum_{k=1}^{K}\widehat{p_{t_k}} ln\ \widehat{p_{t_k}}$ (related to log-likelihood in multinomial)

Logistic Regression

- K = 2, Y = 0 or 1, want $P(\hat{Y} = 1|X)$ with probability [0,1]
- $\ln\left(\frac{p(x)}{1-p(x)}\right) = \hat{\beta}_o + \hat{\beta}_1 X$ for x in (-∞,∞) where $\hat{p}(x) = \frac{e^{\hat{\beta}_o + \hat{\beta}_1 x}}{1+e^{\hat{\beta}_o + \hat{\beta}_1 X}}$
- $\ln\left(\frac{p_k(x)}{1-p_k(x)}\right) = \hat{\beta}_{k_o} + \hat{\beta}_{k_1} X$ where $p_k(x) = P\left(\hat{Y} = k\middle|X\right)$, k = 1,…,K-1 and K is the baseline category
- *multinom* from *nnet*:
    - skips the hidden layer and combines the inputs directly into a linear combination
    - uses a sigmoidal output function
    - explanatory variables scaled to be between 0 and 1
- *glmnet*:
    - does a logistic regression on each binary indicator
    - provides a set of coefficients for all K classes instead of for K-1 comparisons with the baseline class
- Regular linear regression does not constrain the estimated probability to lie between 0 and 1, and does not account for E(Y) being a probability

Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA)

- In logistic regression, parameter estimates and boundaries become unstable when classes have little overlap
- LDA models the distribution of the explanatory, given the response (Bayes' rule)
- Assumes that X has a multivariate normal distribution (MVN):
    - In LDA, the variances and correlations are not changing across k, but the means may change, whereas in QDA the variances and correlations are different across k
- Produces K-1 linear discriminants, $Z_1,\ldots,Z_{K-1}$ (similar to Z in PCA, PLS)
- QDA may reduce bias, but can increase variance for linear surfaces

Unsupervised Learning

- No clear outcome what to optimize for, and often no true value to compare with
- Qualitative rather than quantitative, but has an important role in a statistical toolkit
- **K-means clustering**:
    1. Randomly assign each observation to one of K clusters
    2. While cluster labels are changing after each iteration:
        - For all observations in each of the current clusters, compute the cluster centroid, the mean of each of the p predictors
        - Compute the distance from each observation to these K centroids, reassign that observation to the cluster it's nearest to
- Since we started with the points in random clusters, we may not get to the **best overall** solution (only finds a *local optimum* – best possible solution near that starting point)
- For best overall solution, try different starting points and compare results (pick smallest value)

Bootstrap Aggregation (Bagging)

- Works for any type of model, no pruning, reduces variance
- Final model is an average of the B separate trees, where each tree is a "base learner"

Random Forest

- **For each parent node (potential split):**

1. Randomly choose $m \leq p$ variables
2. Pick best split only from among candidate variables for that parent node

- **Reduces correlation between trees** because they won't always be based on as many similar splits
- "weak learners", each tree has some bias, but decrease variance significantly
- Variable importance:
    1. In every parent node we have a set of candidate variables for splitting.
    2. One gets chosen, and the split reduces the RSS by some amount.
    3. Having different candidate subsets means that more variables get used at one time or another
        - "Poor" variables' splits will not change RSS much when they are chosen
        - "Good" variables will make relatively larger changes when they are chosen.
        - "Good" variables will be chosen more frequently, including near the top when they are candidates there
            - Greater potential for reduction
    4. In each tree we can measure how much each variable contributed to that tree's reduction in RSS
        - Average this across trees: Mean Decrease in RSS

Boosting

- Add new trees depending on the previous tree
- Each tree explains only a little bit of the true structure
- Trees are small and we use many trees to construct a potentially very complex structure
- Parameters:
    - B, number of trees
    - D, size of trees to be fit at each step
    - λ, shrinkage parameter:
        - reduces the influence of each individual tree (smaller λ means more trees)
        - prevents overfitting and reduces variance

Step Function

- Takes different value at each region, with its own mean
- Drop 1 of the indicators and use it as baseline

Splines

- Cubic regression:
    - Fit a polynomial within each region, and add constraints on the model that force the pieces to join together smoothly
    - Likely can fit a much simpler function in that region than what would fit the entire range of X
    - By keeping functions simple within regions, they are more stable than high-order polynomials
    - Need to choose K (possibly by tuning), number of cut points, and degree of freedom is K+3
- Natural cubic:
    - Replaces cubic with lines at ends, because often end segments are very variable since there are no data on the other side to help place the curve
    - Degree of freedom reduced to K+1 (allows 2 additional knots for the same DF)

Neural Network

- Input variables X "fed" into a *hidden layer of nodes*
- Need massive data, and predicts many parameters
- Not interpretable, tend to overfit
- **Regularization** of the parameters can prevent overfitting, and **backpropagation** is used to get these estimates

Ridge, LASSO, and PLS

- Choice of shrinkage parameter λ is a bias-variance tradeoff: increasing λ from 0 adds to bias and decreases variance
- **Ridge**:
    - Shrinkage parameter aims to minimize L2-norm
    - Keeps all variables
- **LASSO**:
    - Some parameters may be shrunk to 0, minimizes the L1-norm
    - λ_{1SE} can be used instead of λ_{min} if sample is large, to increase a little bit of bias and decrease variance
- **PLS**:
    - Uses a linear combination of explanatory variables, based on both explanatory and response for each dimension