Decision Trees

The core idea of decision trees is to find a partition of the space of predictors and then predict a constant in each set of the partition. It could be used in both regression and classification problem.

Regression Tree

1. Algorithm

- Start with a single region R_1, and iterate:
  1. Select a region R_k, a predictor X_j, and a splitting point s, such that splitting R_k with the criterion $X_j < s$ produces the largest decrease in RSS:

$$\sum_{m=1}^{|T|} \sum_{x_i \in R_m} (y_i - \bar{y}_{R_m})^2$$

  2. Redefine the regions with this additional split.
- Terminate when there are 5 observations or fewer in each region.
- This grows the tree from the root towards the leaves.

2. Additionals

- How to predict the value?

  For regression problem, once we get the whole tree, we could use the mean value of all predictors in that class to predict the response.
- How to choose the split point for quantative variables?

  Take each value of the quantative variable as the splitting point and then find the one which could produce the largest decrease in RSS.

Classification Tree

1. Algorithm

For classification tree, it works much like regression tree. There are two differences between them: prediction and loss function.

Problem	Prediction	Loss
Regression	Mean Value	RSS
Classification	Majority Vote	Classification Loss

2. Classification Losses

In classification, we hope the tree could make samples classified in a group as pure as possible. Therefore, classification losses actually measure impurity of samples within classes.

- The 0-1 loss or misclassification rate:

$$\Sigma_{m=1}^{T}\Sigma_{x_i\in R_m}\mathbf{1}(y_i\neq\hat{y}_{R_m})$$

- The cross entropy:

$$-\Sigma_{m=1}^{T}q_m\Sigma_{k=1}^{K}\hat{p}_{mk}log(\hat{p}_{mk})$$

  where $\hat{p}_{mk}$ is the proportion of class k within R_m and q_m is the proportion of samples in R_m .

- The Gini index:

$$\Sigma_{m=1}^{T}q_m\Sigma_{k=1}^{K}\hat{p}_{mk}(1-\hat{p}_{mk})$$

In practise, cross entropy and gini index are better measures of the purity of a region. And it's typical to use the gini index or cross entropy for growing the tree, while using the misclassification rate when pruning the tree.

There are some specific algorithms in classification trees, such as ID3, C4.5 and CART. Most of them differs in the measure of purity.

- ID3 and C4.5 are entropy-based classification trees.
  - ID3 uses the information gain to decide the best splitting point while C4.5 uses the gain ratio. The reason is that information gain would classify each samples into one group if the dataset has the index variable. Gain ratio is introduced to solve this problem.
  - Note: when we use gain ratio to choose the splitting point, we cannot directly choose the point which maximizes the gain ratio because this policy might prefer to choose features with less values/categories. A better way is to choose the features whose gain information are higher than the average at first and then choose the feature whose gain ratio is the largest.
- CART uses the gini index to split samples. **CART could be used in both regression and classification problem.**

Overfitting and Missing data

- How can we control overfitting?
  - Stop growing the tree when the loss doesn't drop by more than a threshold with any new cut. (*But it's possible to find good cuts after bad ones.*)
  - Take the number of nodes as the penality term in the loss function. (control the depth and width of trees)
  - Pre-prune and post-prune.

- How can we handle missing data?
    - Idea 1: Each time we only consider the samples which have the variable and then in addition to choosing the best split, we also choose a second best split using a different variable, and a third best,...If it is missing a variable to make a decision, try the seconde best decision, or the thir best,...
    - Idea 2: Calculate loss based on non-missing values on the variable and then for the sample with missing features, we could assign it to every group according to weights.

Additionals

- Advantages:
    - Easy to interpret and closer to human decision-making
    - Easy to visualize graphically
- Could features repeat to be splitting points?

  For categorical variables, we cannot use it repeatedly. While for quantative variables, we could use it in later split.
- Sometimes, we don't have to choose only one feature as splitting point each time. We could try to make a linear combination of many features and then use it as the splitting point. This trick could help us find a linear classifier (*Note: classical classification trees have non-linear decision boundary.*) and make the cost smaller.