Shaun (Siyeon) Kim
EID: sk4973
Project Name: Random Forest

Random Forest is an ensemble learning method for classification and regression tasks that operates by creating multiple decision trees and taking the majority vote of those trees. This way, the algorithm reduces the variance and produces a well-performing prediction model. The time complexity for building a complete unpruned tree is O(v* n log(n)) where n is the number of input sizes and v is the number of predictors. Because the model constructs multiple decision trees by randomly sampling dataset with replacement (bagging) and randomly subsetting predictors, the algorithm will greatly benefit from parallelization. For this project, we will be focusing on the classification task of the algorithm.

Monad

Decision Tree

We will be following the ID3 algorithm for creating the decision tree. ID3 is a greedy algorithm with no backtracking capability.

1. Starts with the root node set S - dataset
2. We calculates Entropy of all the attributes. Then, we select the attribute with the smallest Entropy
3. The set S is splitted by the selected attribute to create subset of the data
4. We recurse back to (2.) until maximum depth is met
5. Return the Decision Tree

Functionality

Train

Input & Parameters

- Input data: v x n tabular data
- Ntree: Many existing packages of the algorithm take in the number of trees you want to build as a parameter. This parameter allows the users to expedite the training process. .
- Maximum depth: Since the decision tree is not guaranteed to converge, setting a maximum depth is a common practice in other random forest libraries.

Output: Collection of trees

Functionality

1. Read the tabular data in a similar way as the K-means example
2. Random sampling of the data and random subsetting of different predictors
3. Build Decision Tree
4. Aggregate all the trees and vote on the prediction

Predict

Parameters

- Input data: v x n tabular data

Output: Prediction (nx1)

Functionality

1. Read the tabular data and produces prediction by running through the trees that are made in the train function