Homework 4 Boosting

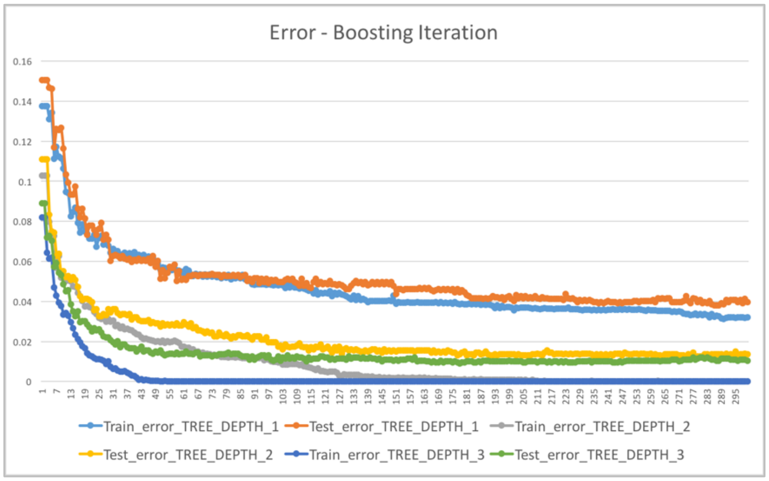
Yawen Zhang

1. My AdaBoost algorithm (**with n_learners = 50, tree_depth = 1**). I got the results:

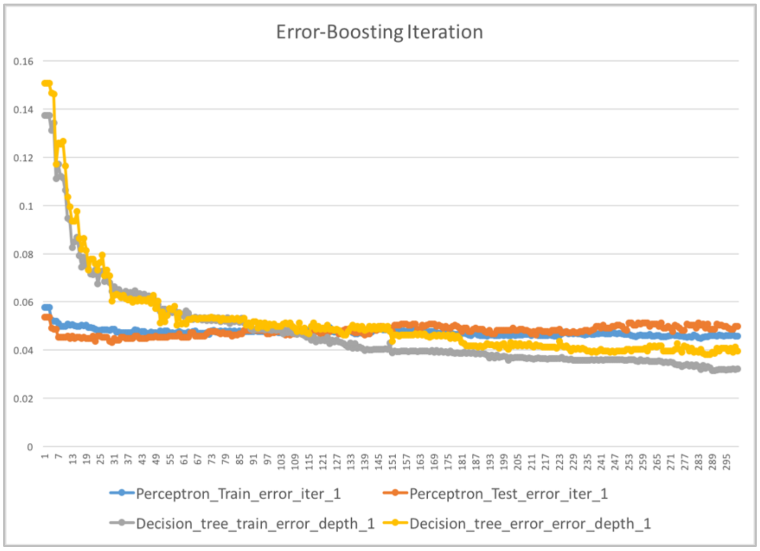
Accuracy_score_training_data = 0.940997258048

Accuracy_score_test_data = 0.939728779508

2. As shown in the picture below, tree depth = 1 works best, the overfitting starts much slower than the other two depths, and the error difference between training data and testing data is not that much compared to the other two depths. It is kind of a good bias and viariance trade-off. For all these three depths 1, 2, 3, I have seen signs of overfitting when the training error is less than testing error, but they happen at different iteration round. For depth 1, it happens at around iteration round 100, for depth 2, it happens at around 30, and for depth 3, it happens at the very beginning. When depth is equal to 3, the training data can get a nearly 100% accuracy. On further iterations, the accuracy on the test data increases slowly, tend to be stable.



3. Here, I selected the linear model Perceptron, and used three n_iter of it: 1, 2, 3, similar with Decision Tree depth, also, n_learners = 300. It has the similar character with Decision Tree, larger n_iter will resulting in smaller error and also faster overfitting. As shown in the picture below, it is obvious that Perceptron can get a smaller training and test error than Decision Tree at the beginning. However, when overfitting happens, Perceptron test error tends to increase while Decision Tree test error is stable. Boosting is more effective with Decision Tree.


Error-Boosting Iteration
Perceptron_Train_error_iter_1
Perceptron_Test_error_iter_1
Decision_tree_train_error_depth_1
Decision_tree_error_error_depth_1