CS7646 Project 3: Assess Learners

Thomas Kim
tkim338@gatech.edu

Abstract—This project explores the development and usage of different decision tree learning models. Included here is a deterministic decision tree (feature splits are determined by correlation coefficient and median feature value), random decision tree (feature splits are chosen randomly while split value is set to median feature value), ensemble learning using multiple decision trees and random sampling of training data, and another ensemble learning model made up of smaller ensemble learning models.

1 INTRODUCTION

Decision trees are tree-based models usually used to classify data. Starting from an initial node, a training dataset is separate at each subsequent node based on a specific feature and value. Once the subset of data at a particular node is reduced to a specific size, the node becomes a leaf node, which is terminal, leading to no further nodes and no further splits in data. Once all of the training data is processed and the decision tree is completed, the decision tree can be queried with test data to classify each data point in the test dataset.

The experiments in this paper evaluate the performance of a deterministic decision tree that chooses split features based on correlation coefficient, a random decision tree that chooses split features randomly, an ensemble learner that implements multiple trees, and an additional ensemble learner that implements multiple smaller ensemble learners that each implement a number of decision trees.

It is hypothesized that a deterministic decision tree may provide good performance for some specific data with low noise, however ensemble learning will provide good performance that is more robust against noisy or more varied data.

2 METHODS

The deterministic decision tree developed here is a binary tree that splits features based on the highest absolute value of correlation coefficient between the feature set and the labels. The feature value on which to split is determined by finding the median of the feature value of the subset of data at the node. Nodes are created until the subset of data is reduced to a maximum of a given leaf size. Leaf values are set to the median label of the associated data subset. In these experiments, labels are all numerical, which makes calculating a median pragmatic.

The random decision tree is similar to the deterministic decision tree in all aspects except that split feature selection is purely random instead of determined by correlation coefficient.

An ensemble learning model is implemented as a collection of decision trees, where the type of decision tree is variable (deterministic decision tree, random decision tree, etc.). Each decision tree is trained on a unique set of data that is randomly sampled from the overall training dataset. In this implementation, each decision tree's training dataset is equal in size to the overall training dataset. When querying the model, each decision tree is queried with the test data and the final output is computed as the mean of the outputs from the trees.

An additional ensemble learning model (called "InsaneLearning" here) is implemented as a collection of the previously implemented ensemble learning models except that only linear regression decision trees are used instead of leaving this open as a parameter.

3 DISCUSSION

3.1 Experiment 1

In evaluating the performance of deterministic and random trees, overfitting is observed for small leaf sizes for these trees using the Istanbul.csv dataset. This can be seen most clearly in the RSME of the DTLearner with varying leaf size (Figure 1). RMSE reaches a minimum with a leaf size of approximately 5 to 10. Leaf sizes outside of this range results in higher RSME. Higher leaf sizes result in reduced model fidelity while lower leaf sizes result in overfitting, an effect of

the model too closely matching the training data and losing accuracy in data outside of this set.

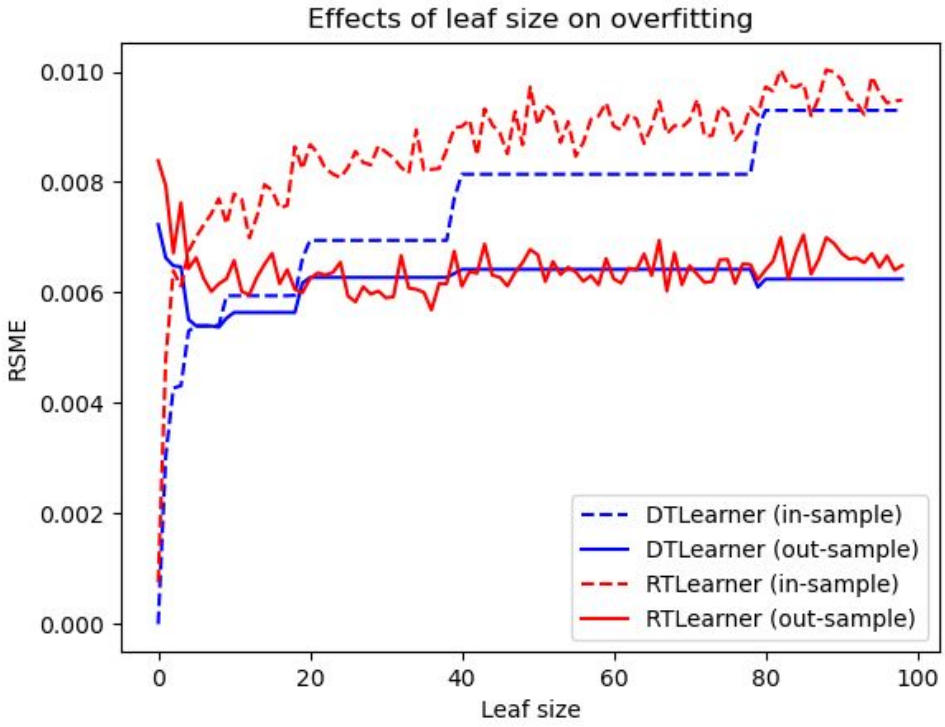

Figure 1—Effects of leaf size on overfitting.

3.2 Experiment 2

Bagging is a method of ensemble learning that produces multiple models by sampling from a training dataset and computing an output from the average of the internal models. Bagging is evaluated here for its capability in reducing overfitting. Due to random sampling, moderate noise is present in the results. However, some effects of overfitting can be seen in Figure 2 in the charts of models containing 1 and 2 bags. Overfitting appears to occur when leaf size is about 35. Increasing the number of bags appears to reduce overall RSME, but also reduce the effect of overfitting, which can be seen in the figure as a flattened curve for the lines for models containing 5 and 10 bags.

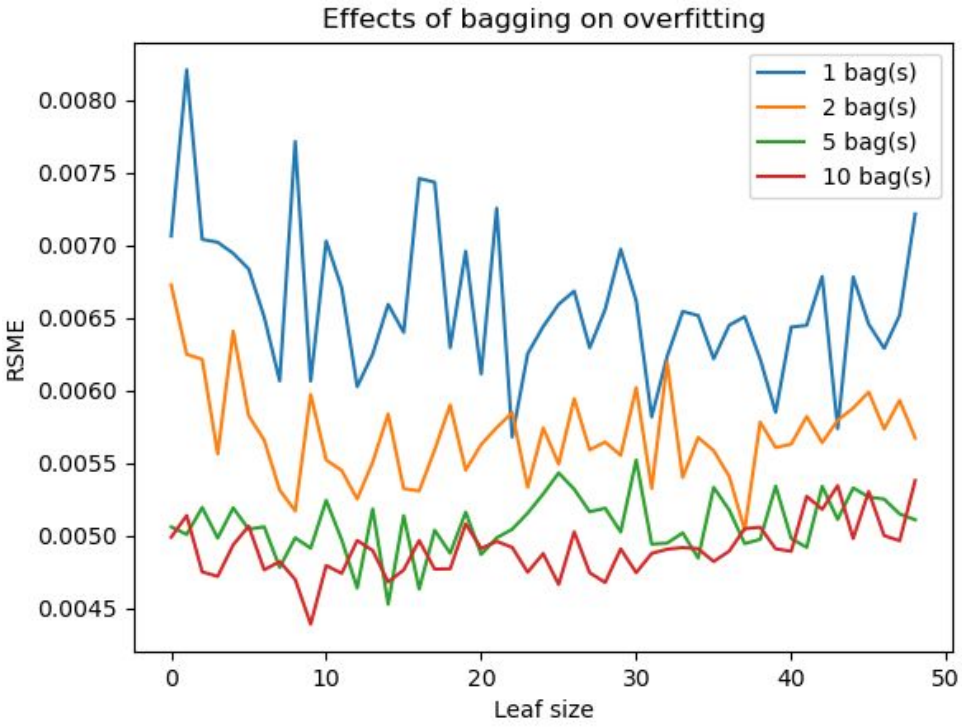

Figure 2—Effects of bagging on overfitting.

3.3 Experiment 3

Based on previous experiments (Figure 1), deterministic "classic" decision trees appear to usually outperform random decision trees. There are cases in which a particular random tree produces less error with a test dataset than a deterministic tree, but the average performance of random trees is generally worse.

Random trees have the advantage of quicker runtimes. Figure 3 below shows the time (in seconds) required to generate the trees using the winequality-white.csv dataset with varying leaf size. As the only difference between the DTLearner and RTLearner is the method in which the split feature is chosen and the method in DTLearner is more computationally expensive, this difference in processing time is expected.

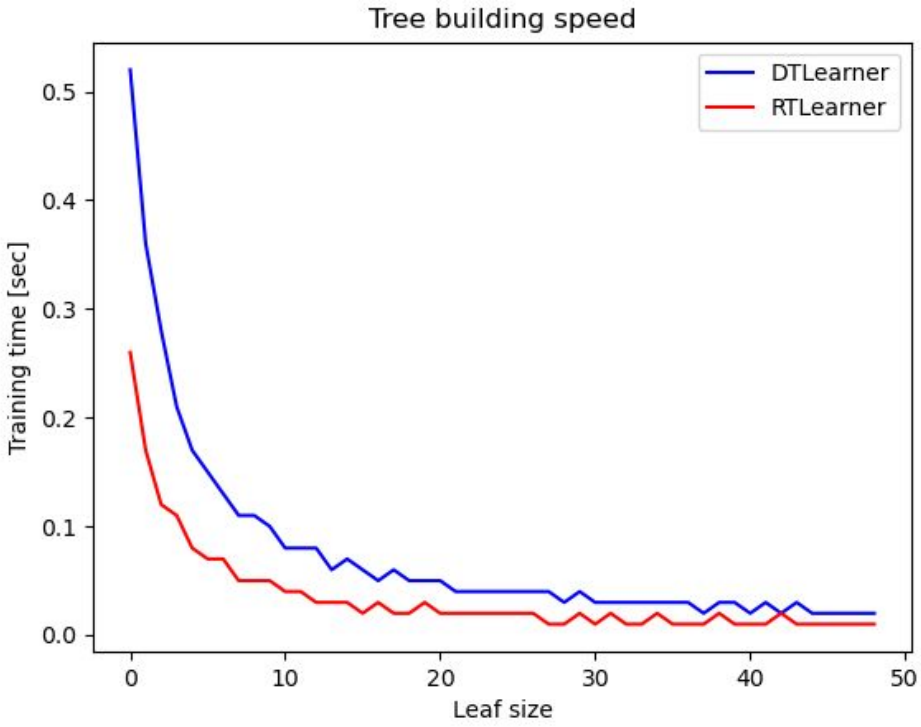

Figure 3—Speed comparison between DT and RT.

Deterministic trees and random trees show comparable performance in the depth of the tree produced. Figure 4 below shows the maximum depth of each tree produced by each learner for varying leaf size. This metric was explored as with large trees, average query time is proportional to the depth of the tree. Surprisingly, almost no difference was seen in the difference in depth of the trees generated by DTLearner and RTLearner. Because the methodology of these models uses the median feature value to split datasets, this is reasonable, although a greater difference was originally expected.

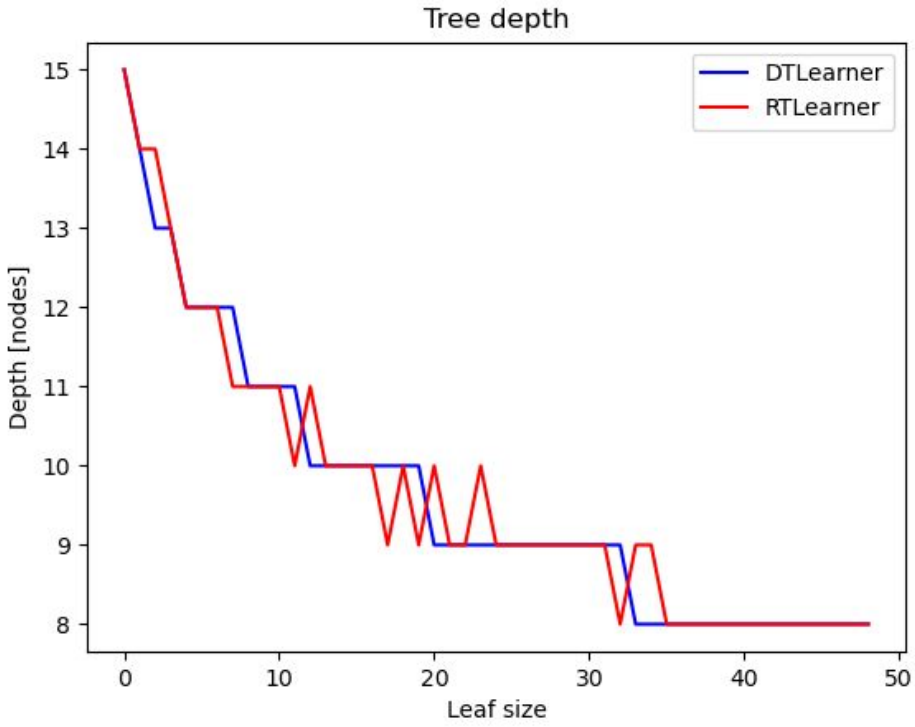

Figure 4—Tree depth comparison between DT and RT.

4 SUMMARY

In this examination of decision tree performance, it was found that there is only a slight difference in performance (for this particular dataset) between a random tree and a deterministic "classic" decision tree. Differences in accuracy and maximum tree depth between the DTLearner and RTLearner were comparable, with a slight accuracy advantage going to DTLearner. RTLearner does have a performance advantage in computing time, as it doesn't need to compute correlation coefficients before selecting a split feature. The RTLearner also exhibits less sensitivity to overfitting from small leaf sizes in these experiments, although this may be a result of increased noise masking the effects.

Bagging does improve overall accuracy at the cost of increased computing time, but also works to reduce the effect of overfitting due to small leaf sizes. Random sampling in bagging also introduces noise into the results, which could be obscuring some effects of overfitting. In future investigation, more tests can be run to reduce the noise produced by bagging and RTLearner to more closely

examine the effects of leaf size on overfitting. The methodology of generating these trees can also be explored further, such as using information gain or entropy to determine the value of a feature on which to split instead of using the median.