Assignment 3: ASSESS LEARNERS
CS7646

Jungeun (Judy) Cha
jcha64@gatech.edu

In this report, we like to explore the three tree-based learning algorithms: a "classic" Decision Tree learner, a Random Tree learner, and a Bootstrap Aggregating learner. Also, we will investigate impact of hyperparameter tuning and performance of the model on a training data vs. testing data by quantitative measure and bagging effect to reduce overfitting for tree-based learners.

INTRODUCTION

we will be using the data provided in the *Istanbul.csv* file for following experiments. The metric for assessing overfitting will be RMSE (root mean squared error) between the prediction and target variable. RMSE is a frequently used measure of the differences between values (sample or population values) predicted by a model and the values observed. RMSD is always non-negative, and a value of 0 would indicate a perfect fit to the data.

- Decision Tree Learner (DTLearner)

  For the purposes of this experiment, we have used JR Quinlan's implementation of the decision tree algorithm. Quinlan's paper is focused on creating classification trees, while we are creating regression trees here. We build a single tree (not a forest) and define "best feature to split on" as the feature (Xi) that has the highest absolute value correlation with Y.
- Random Tree Learner (RTLearner)

  This learner behaves exactly like DTLearner, except that the choice of feature to split on is made randomly. Random tree learner is an alteration of JR Quinlan's that chooses the feature to split on as a random feature instead of the feature with the highest correlation with the target variable.
- Bag learner (BagLearner)

Bag learner implements bootstrap aggregating method for training multiple learners on different subsets of the data. Where learner is the learning class to use with bagging, it is designed so that BagLearner can accept any learner as input and use it to generate a learner ensemble. The "bags" argument is the number of learners to train using Bootstrap Aggregation. Each bag is trained on a different subset of the data and the mean of the predictions is returned as the final prediction.

1. Experiment 1: Overfitting / leaf_size

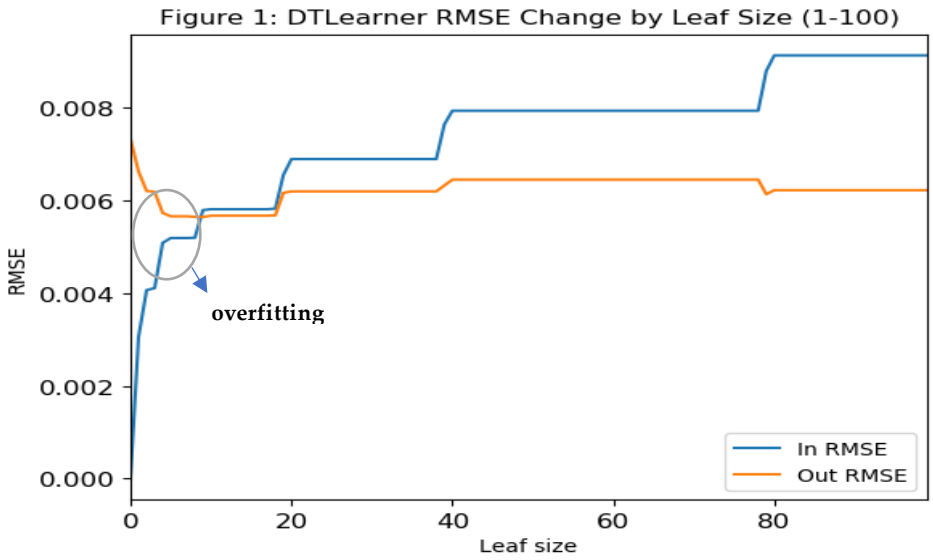

Figure 1: This figure shows overfitting (low in-sample RMSE and high out-of-sample RMSE) below 9 leaf_size. The area to the left of the point of intersection is the area of overfitting. (In sample RMSE vs. Out of sample RMSE)

Does overfitting occur with respect to leaf size?

In experiment 1, the leaf_size is hyperparameter which is a parameter whose value is used to control the learning process. "leaf_size" defines the maximum number of samples to be aggregated at a leaf. It turned from 1 to 100 and in-sample (train) RSME and the out-sample (test) RSME are used for each version of the decision tree.

DTLearner is least generalized when leaf_size = 1. In other words, when leaf_size =1, the model predicts each training point in the data set perfectly but fails to predict testing points accurately.

For which values of leaf_size does overfitting occur?

An overfit model matches the training set very well but fails to generalize to new examples. As a result, we see that increases in leaf_size decrease the probability of overfitting. Figure 1 shows for leaf sizes little less than 10 (around 9), the decision tree learner tends to overfit on our data (Istanbul.csv).

When leaf-size is decreasing below a little less than 10, in sample RMSE decreases and out of sample RMSE increase where overfitting happens. In other words, where the accuracy on the training set improves but the accuracy on unseen data degrades when overfitting.

2. Experiment 2: Does bagging reduce or eliminate overfitting?

Can bagging reduce overfitting with respect to leaf_size? Can bagging eliminate overfitting with respect to leaf_size?

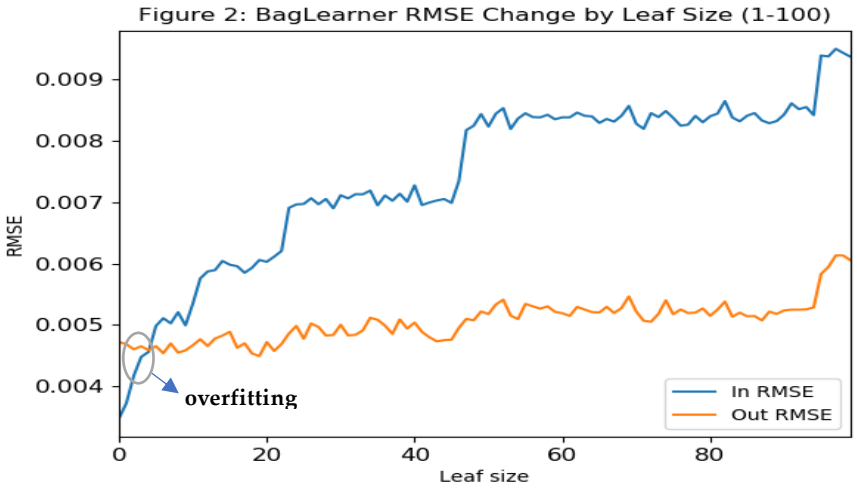

Figure 2: This figure shows the point of intersection moved further to the left, reducing the overfitting area. (In sample RMSE vs. Out of sample RMSE)

We noticed that figure 2 shows that the point of intersection for BagLearner moved further to the left which is reducing the overfitting area compared to

experiment 1(DTLearner). For experiment, we used 20 bags and varied leaf_size to evaluate. We take each bag and use it to train an instance of our learner. Just like when we have an ensemble of different learning algorithms, we query our bagged ensemble. We pass the same X to each trained model and collect and average the results to generate a Y.

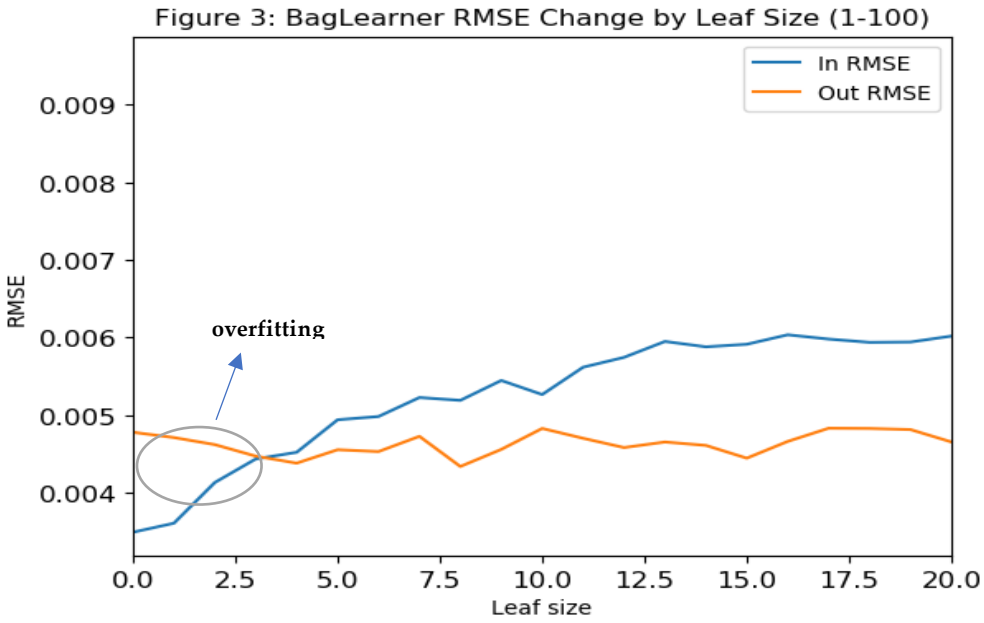

Figure 3: zooming graph of Figure 2

In above zooming graph, it clearly shows that the point of divergence of in sample and out of sample reduced to around 3.5 as compared to 9 from the experiment 1, DTLearner. We observe the area to the left of the point of intersection (area of overfitting) is smaller than experiment 1. This means bagging tries to avoid overfitting. The direction of overfitting is like experiment 1 as when leaf-size decreases from 3.5 to 0, in sample RMSE decreases and out of sample RMSE increase where overfitting occurs.

We can offer an intuitive explanation for these claims. Bagging use multiple instances of the same learner, training each on a slightly different subset of the data. Every instance that we use has its own type of bias. When we combine learners, these individual biases can cancel each other out, to an extent. Therefore, bagging ensemble learners are less likely to overfit than a single learner.

3. Experiment 3: Comparison of DT and RT learning

Quantitatively compare "classic" decision trees (DTLearner) versus random trees (RTLearner). In which ways is one method better than the other?

3.1 R squared

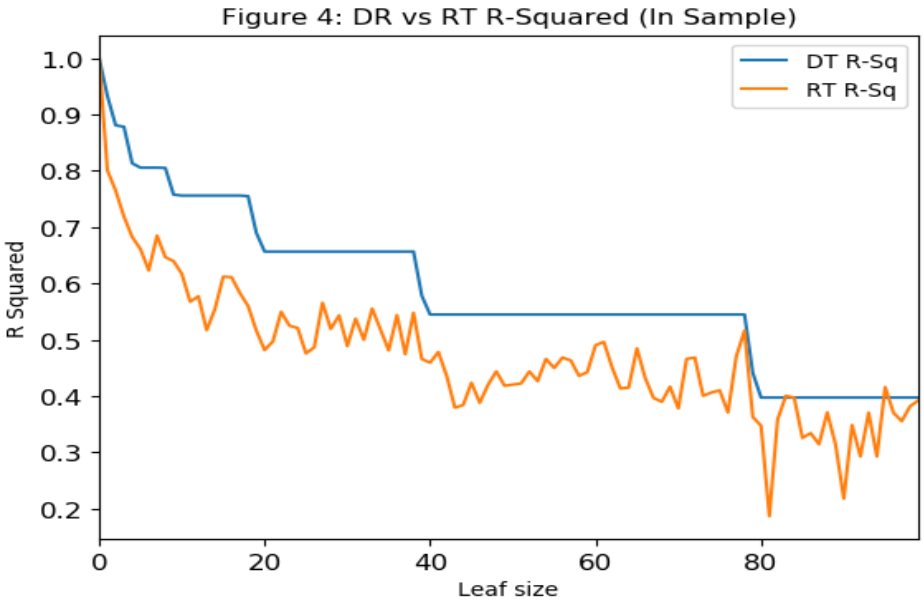

Figure 4: DTLearner vs. RTLearner performance of R^2 (In Sample)

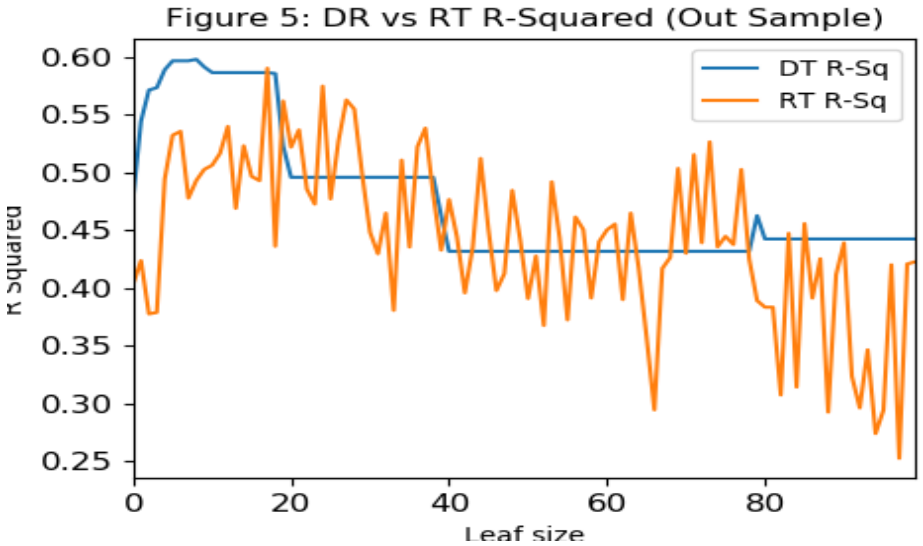

Figure 5: DTLearner vs. RTLearner performance of R^2 (Out of Sample)

The coefficient of determination, R-squared (R^2) is a statistical measure that represents the proportion of the variance for a dependent variable that is explained by an independent variable or variables in a regression model. R^2 test is used to determine the goodness of fit in regression analysis. Goodness of fit implies how better regression model is fitted to the data points. More is the value of r-square near to 1, better is the model. We plot R^2 associated with the predicted values for decision tree as well as random tree learners for in sample and out of sample datasets over leaf sizes varying from 1 to 100.

Figure 4 (in sample - training) shows that DTLearner is outperforming RTLearner since DTLearner uses the highest absolute value correlation as best feature to split on. In training test, DTLearner model is better fitted to the data points than RTLearner with the random choice of feature to split on.

However, Figure 5 (out of sample - testing) shows that DTLearner's outperforming only shows at the beginning (< 18 leap size) and then it is not clear afterwards. It seems the DTLearner's outperformance over RTLearner does not continue in out of sample data as leap size increases.

3.2 Training Time

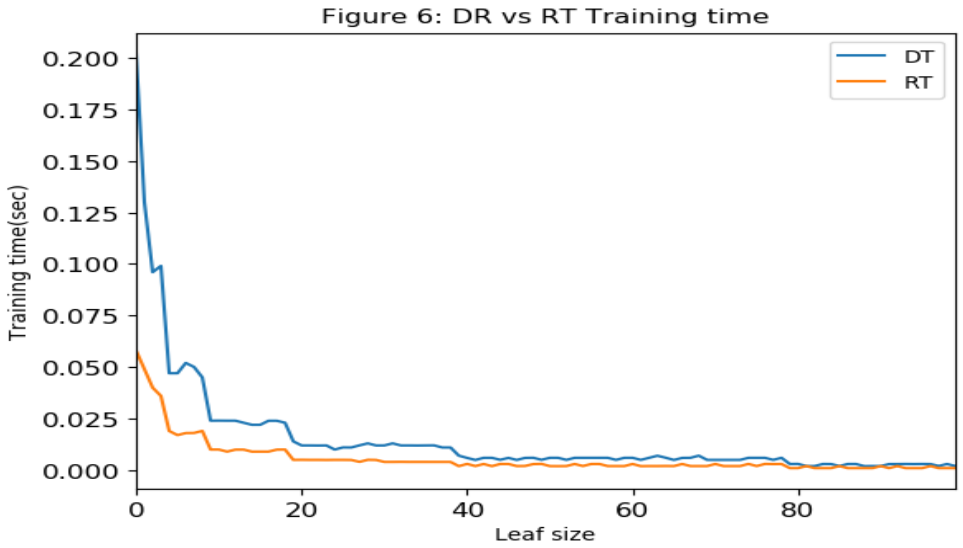

Figure 6: Training time comparison (DTLearner vs. RTLearner)

The time to train the model is measured for decision tree learners as well as random tree learners. Based on Figure 6, we have noticed RTLearner performs faster than DTLearner when training. Finding correlations for each split should be slower compared to randomly selection split factors. RTLearner would be significantly faster since they randomize the most time intensive tasks of best feature selection. The average training time for DTLearner is 0.016, while the average training time for RTLearner is 0.006. Thus, on an average a single random tree learner is 2.6 times faster than DTLearner.

4. REFERENCES

https://en.wikipedia.org/wiki/Root-mean-square_deviation

https://en.wikipedia.org/wiki/Coefficient_of_determination