Assess Learners Report

Varun Behl

February 12, 2018

Preface:

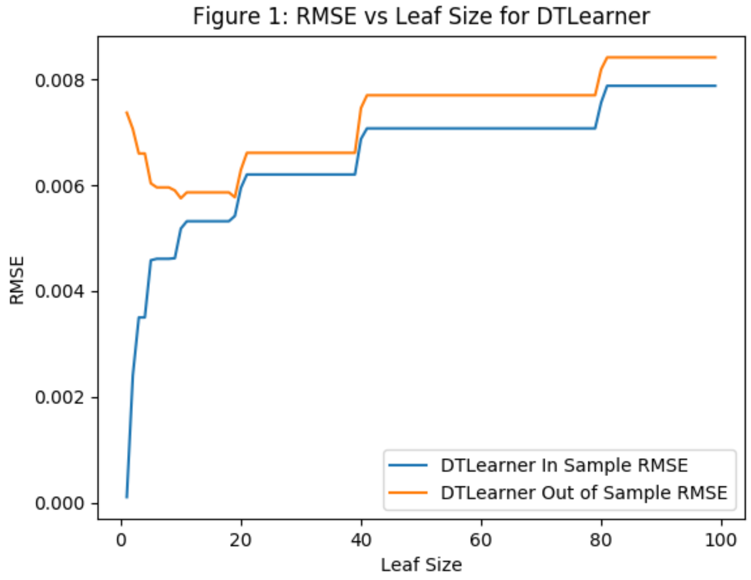
The assess learners assignment involves using random tree and decision tree learners to create trees and then using those trees to query for given values. The outputs include correlation and root mean square error (RMSE) for both the in sample and out of sample data that the learner and query functions receive. Next, we created a bag learner using both the decision tree learner and the random tree learner with a bag size of 20. We then created an insane learner using 20 bag learner instances, where each instance is composed of 20 linear regression learners.

After creating the learners, the test learner was edited in a way that it would output the correlation and RMSE values for leaf sizes ranging from 1 to 100. While leaf sizes greater than 100 were tested, for the purposes of this report I limited the size to a max of 100 so the data is easily readable. Furthermore, the data after 100 leaves did not vary as much as cases with less than 20 leaves, since with about 600 points of data, more than 100 leaves would only include two splits in the tree.

This report includes figures that demonstrate the differences in correlation and RMSE. The first section relates to if overfitting occurs with respect to leaf size, which was tested for a decision tree learner. The second section related to the same question but with bag tree learners including decision tree learners. The third section is a comparison between random trees and decision trees. The code trains on a random 60% of the data and then tests on the other 40% of the data.

1. Does overfitting occur with respect to leaf size?

Overfitting occurs on smaller leaf sizes. Between 0 and around 10, overfitting occurs because the in sample RMSE is a very low value. Around 10-12 leaves, normal expected fitting occurs wherein the in sample follows the out of sample RMSE. Therefore, overfitting occurs when there is ten or less leaves, but tends to dissipate with more leaves. This is shown in Figure 1, where the in sample and out of sample RMSEs become more closely correlated as the number of leaves passes 10.



The other interesting thing to note is that although Figure 1 shows the in sample RMSE following below the out of sample RMSE, some samples that were not trained on good data could cause the in sample error to actually get a worse RMSE than the out of sample. This occurs if there is a high standard deviation on the training data due to, for example, data that isn't well correlated.

2. Can bagging reduce or eliminate overfitting with respect to leaf size?

Bagging does reduce overfitting within a dataset as the number of leaves increases, although it does not seem to be able to eliminate overfitting. Although, with more bags overfitting seems to be reduced. Additionally, the root square mean error reduces with more bags. This is shown in Figures 2-5 below. Figure 2 has 1 bag, Figure 3 has 20 bags, Figure 4 has 50 bags, and Figure 5 has 100 bags.

Figure 2: RMSE vs Leaf Size for DTLearner, Bags=1

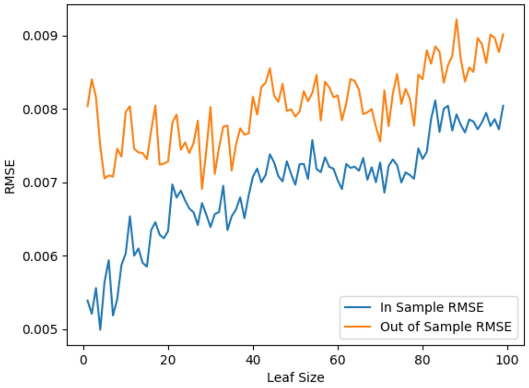


Figure 3: RMSE vs Leaf Size for DTLearner, Bags=20

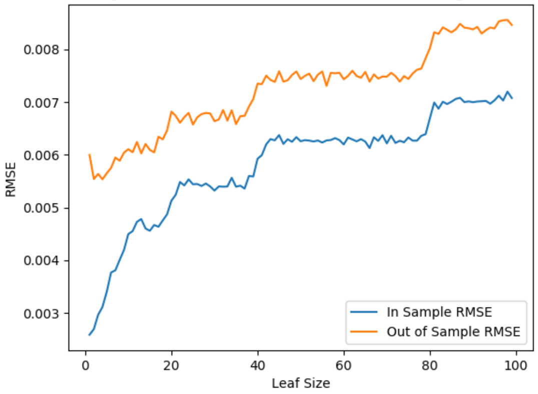


Figure 4: RMSE vs Leaf Size for DTLearner, Bags=50

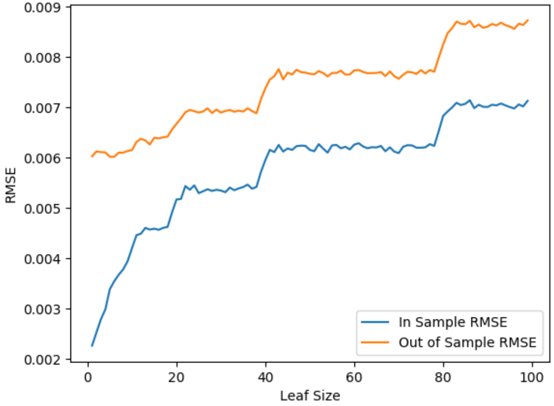
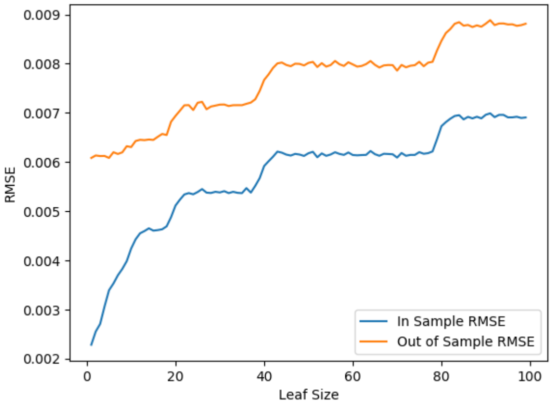


Figure 5: RMSE vs Leaf Size for DTLearner, Bags=100



In addition, the variance in RMSE for nearby leaf sizes is much more similar with more bags, causing the graphs to be smoother. With 1 bag, the RMSE vs leaf size values vary hugely, compared to 100 bags wherein. However, comparing the out of sample RMSE curve between Figure 1 and Figure 5, it is clearly noticeable that there is much more overfitting without bags than with. In fact, with 100 bags there is almost no overfitting, because as shown, the orange out of sample RMSE line doesn't have a negative slope early on.

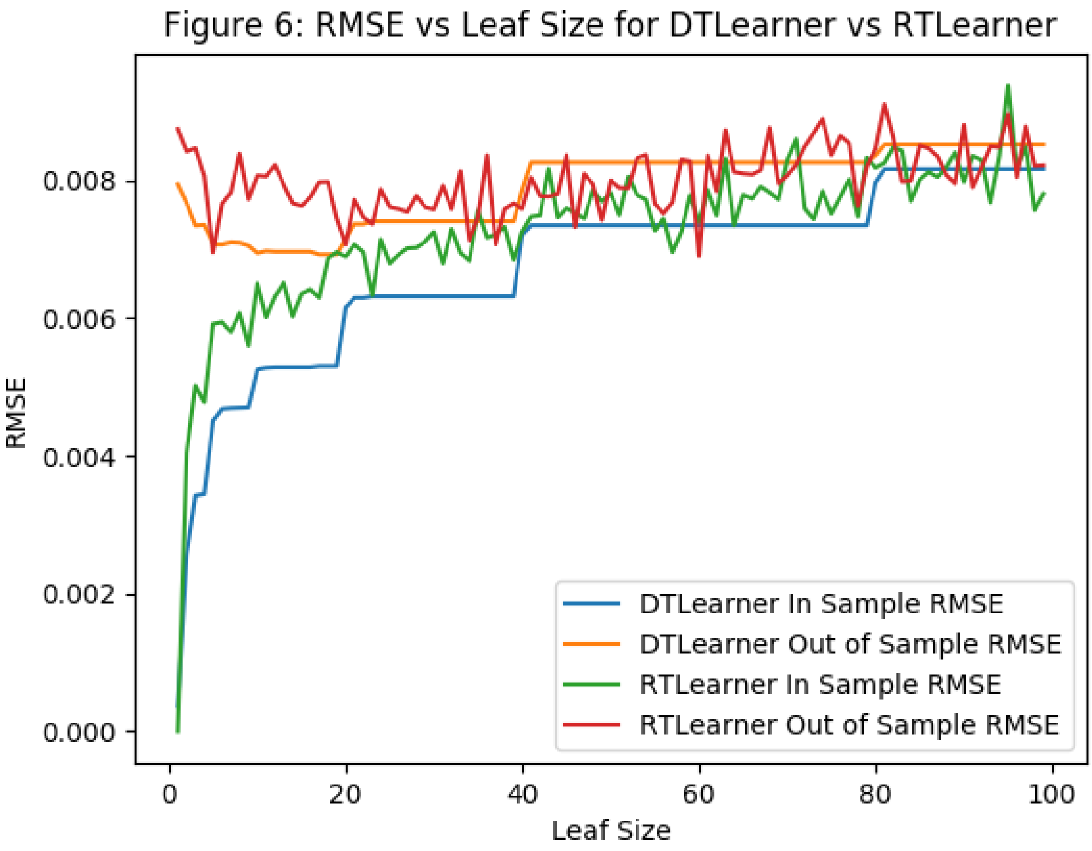
3. Quantitatively compare decision trees versus random trees. In which ways is one method better than the other?

Both decision trees and random trees are built by selecting a feature index and recursively adding nodes to the tree based on whether the node has a feature value greater than or less than the feature index. Querying in a decision tree and random trees also follows the same routine in both types of trees.

The differences come in during the creation of each tree. In a decision tree, a feature is selected by choosing the highest correlation in the set and getting the median of the values of that correlation. Then the tree is split based on that median value based on whether each node in the dataset has a feature that is less than or equal to that value or greater than that value. This is done recursively by choosing the next greatest correlation and so on until the tree is complete. This guarantees that the tree will be split relatively evenly.

In contrast, a random tree chooses features randomly by any random algorithm. The features may not be the best, but the efficiency of the algorithm compared to decision trees makes it so random trees run much faster. In addition, if multiple decision trees are used in, for example, a bag or a forest, the randomly chosen features matter less because many trees can be generated and used at a faster speed than one decision tree.

As shown in Figure 6, a random tree has a much more random RMSE curve than a decision tree's RMSE with respect to leaf size. Both have overfitting, but while the decision tree has a clear downwards overfitting slope, the random tree has both a negative and a positive slope early on. This is potentially caused by badly chosen random features to split on, badly chosen training data compared to testing data, or a combination of the two. Therefore, it seems that random trees have less overfitting than decision trees. The main benefit for decision trees are that they can cause a perfect tree for querying for the training data. If the training data is good, the testing data would have a very low RMSE.



Additionally, in a test using 100 loops of decision tree creation with 1 leaf each versus 100 loops of random tree creation with 1 leaf each, it took ~4.4678 seconds to create 100 decision trees while it took only ~1.156985 seconds to create 100 random trees with 1 leaf each. This ratio of about 3:1 for random trees to decision trees makes it so random trees are much more efficient as far as creation.