Project 3: Assess Learners

Zhihua Jin
zjin80@gatech.edu

1. Does overfitting occur with respect to leaf_size? Use the dataset istanbul.csv with DTLearner. For which values of leaf_size does overfitting occur? Use RMSE as your metric for assessing overfitting. Support your assertion with graphs/charts. (Don't use bagging).

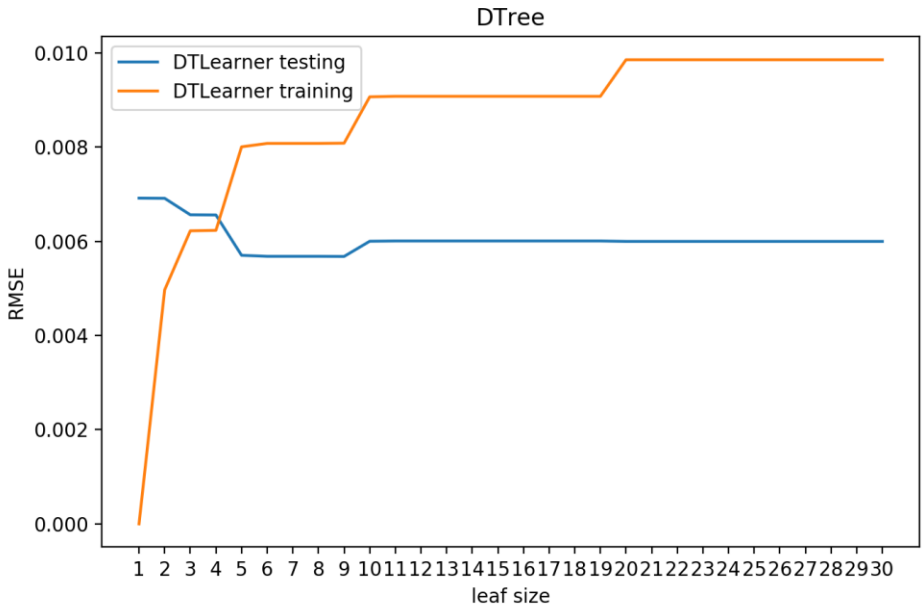

Figure 1. DTLearner: Leaf size vs RMSE

In decision tree, overfitting occurs when training data (in-sample) error is smaller than testing error (out-of-sample). In this case, the learner tightly fits the given training data and gives accurate output on training data, but lacks accuracy in predicting new test samples. One of the methods used to avoid overfitting in decision tree is pruning ("What is over fitting in decision tree?", 2019). Certain accuracy on the provided data shall be sacrificed so that the learner would achieve more generalized accuracy

As you could see in Figure 1, I used different leaf_size from 0 to 30 for training and prediction in my DTLearner and visualizing the results with RMSE as parameter. According to the definition I mentioned above, it could be found that when value of the leaf_size is under 3, the RMSE of training is smaller than testing data, so overfitting occurs. As the leaf size increases, it is more unlikely to over fit. When the leaf_size is 3 or 4, both RMSE are small and similar to each other. This is an ideal stage. Subsequently, when leaf_size is bigger than 5, the training error continue to increase while the testing error stabilizes. We should also prevent the case when the training error is much larger than the test data.

2. Can bagging reduce or eliminate overfitting with respect to leaf_size? Again use the dataset istanbul.csv with DTLearner. To investigate this choose a fixed number of bags to use and vary leaf_size to evaluate. Provide charts to validate your conclusions. Use RMSE as your metric.

The fixed bag numbers I chose varies from 10, 20 to 40. Their overall tendency is similar.

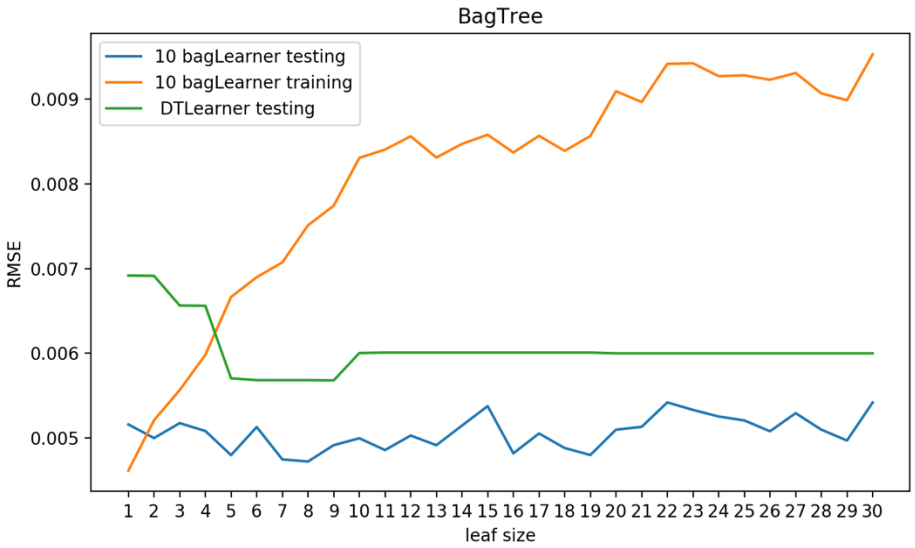

Figure 2. Bagging DTLearner - 10 Bags: Leaf size vs RMSE

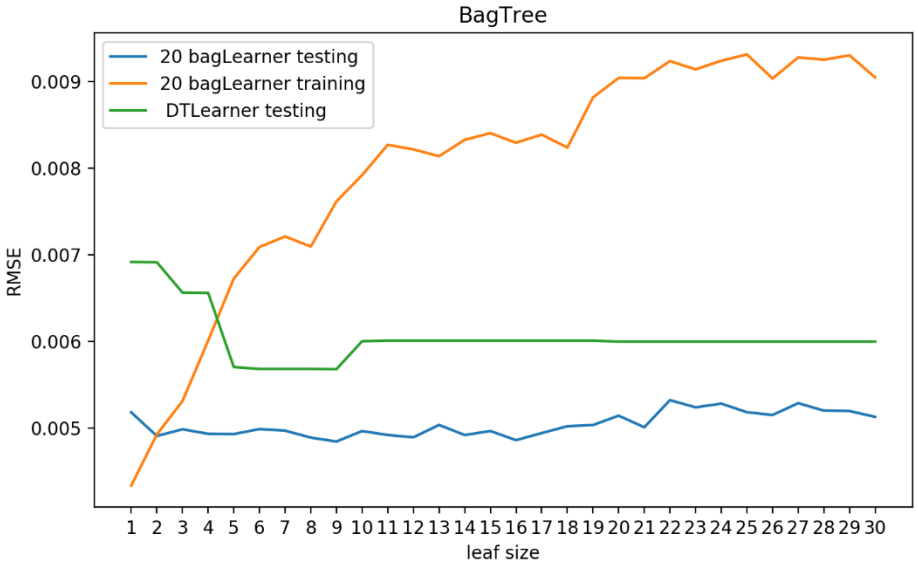

Figure 3. Bagging DTLearner - 20 Bags: Leaf size vs RMSE

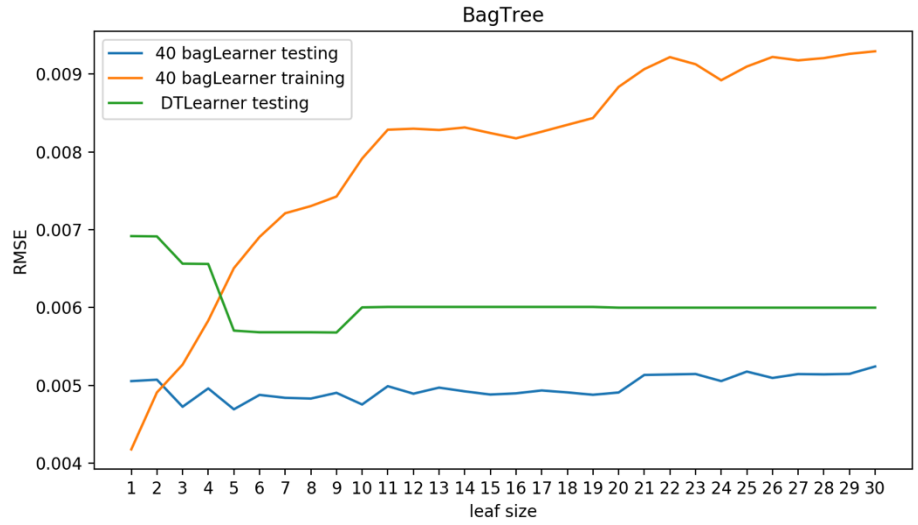

Figure 4. Bagging DTLearner - 40 Bags: Leaf size vs RMSE

By using different leaf_size (0 to 30) for the BagLearner model, the training and prediction are performed and compared with DTLearner testing. In the visualized result shown in Figure 2 to 4, it could be found that when to leaf size reaches 2, the model achieves the same accuracy on both training data and testing data. So

it is only when the leaf size is 1 that the overfitting would occur. Similar to the normal DTLearner, RMSE of training results increases as leaf size increases from 3 while the RMSE testing result fluctuates and then stabilizes at a certain level. After the leaf size reaches 10, the RMSE of bagging learner testing is relatively lower (around 0.05) compared to DTLearner testing (0.06). Another point worth mentioning is that more bags mean better stability in accuracy of test data. If we compare Figure 2 to 4, we could see that the RMSE of 40-bag BagLearner testing data does not fluctuate as drastically as the 10-bag BagLearner.

To sum up, we could assume that bagging reduces overfitting phenomenon to some extent.

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

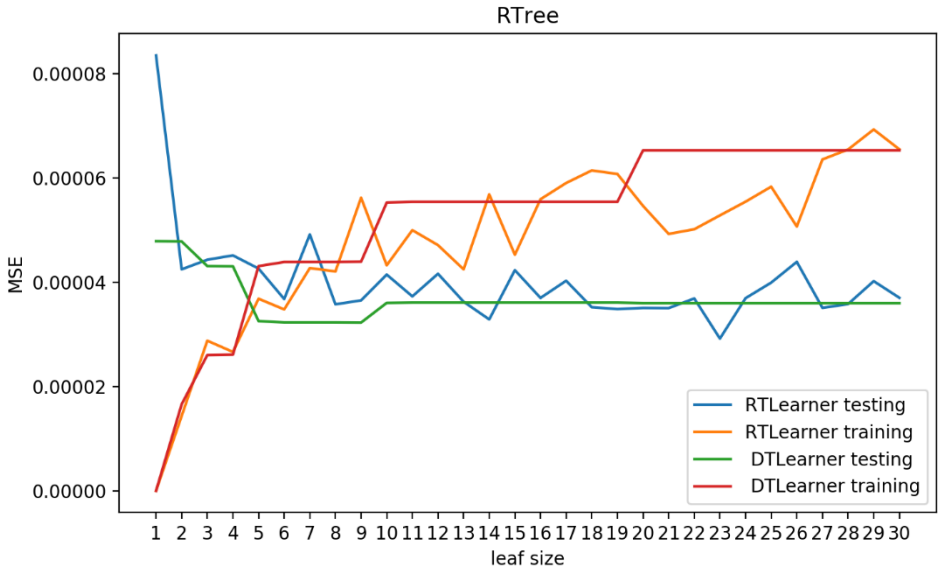

Figure 5. MSE of DTLearner and RTLearner

The first quantitative measure I chose is MSE (Mean Squared error). It measures the average of the squares of the errors. In this project with Istanbul data, it means the average squared difference between the estimated value and the actual value of index returns. This is a measure for accuracy. When MSE is smaller, the model is more accurate. In Figure 5, the MSE of both DTLearner and RTLearner with leaf_sizes from 0 to 30 are shown. The overall tendency of both training and test data is similar. For leaf_size from 9 to 30, the MSE of DTLearner and RTLearner training ranges from 0.4 to 0.7, while the MSE of testing data ranges from 0.3 to

0.4. However, when leaf_size is smaller than 3, the MSE of RTLearner testing is larger than DTLearner testing, which means DTLearner is more accurate when leaf_size is small. Besides, the volatility of DTLearner is lower than RTLearner. For leaf_size larger than 9, DTLearner training reaches two plateaus while testing reaches a plateau and stabilizes. On the contrary, RTLearner continues to fluctuate.

So from this metric, DTLearner is better than RTLearner.

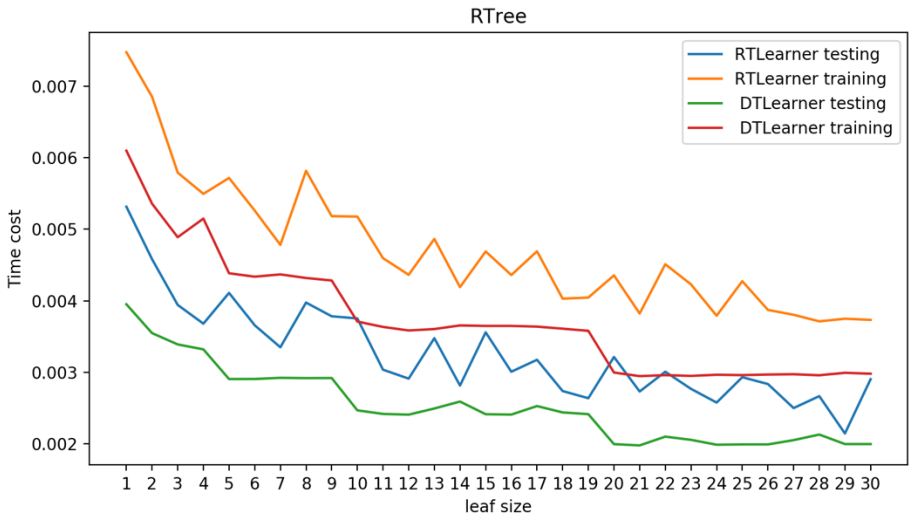

Figure 6. Running time of DTLearner and RTLearner

The second metric is efficiency ("Machine Learning for Trading | Udacity", 2019). By recording the running time of DTLearner and RTLearner with different leaf_size, Figure 6 was plotted. It is quite apparent that both RTLearning testing and training cost more time than DTLearner testing and training. I thought removing lines that compute the correlation should make the learner faster. But probably the random number generator function I implemented slowed it down. Hence, from this perspective, my DTLearner is still better than RTLearner.

Reference

What is over fitting in decision tree?. (2019). Retrieved from https://www.researchgate.net/post/What_is_over_fitting_in_decision_tree

Machine Learning for Trading | Udacity. (2019). Retrieved from https://www.udacity.com/course/machine-learning-for-trading--ud501