Project 3: Assess Learners

Donald Ward – 6/8/2019

dward45@gatech.edu

Question 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?

Answer: By referring to Figure 1 we can see that by looking at in-sample vs out-sample data overfitting is a concern when leaf size is small. RMSE for in-sample data is near zero when leaf size is one, indicating an overfit but the RMSE gap is negligible as leaves numbering 20 are added and we can see RMSE rises to meet the more objective out-sample data. There is an optimal dip in the gap at around 8 leaves, after which the two samples move with each other.

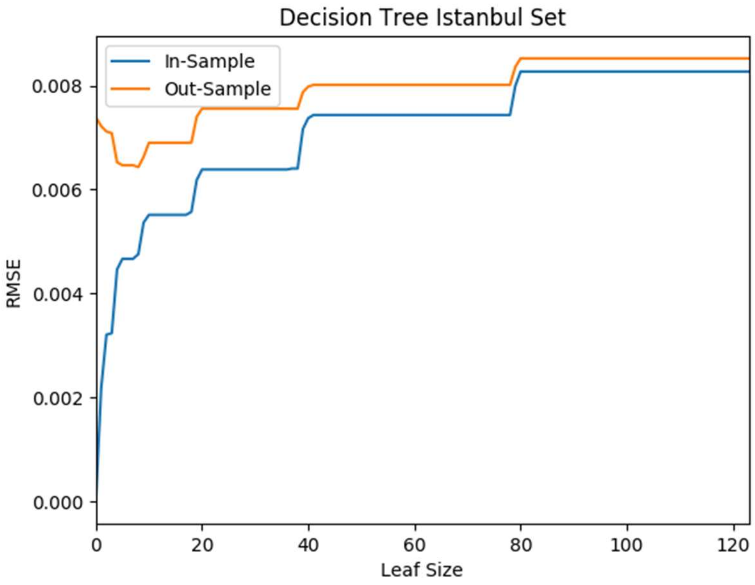

Figure 1. Decision tree showing RMSE in relation to leaf size

Question 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.

Answer: Bagging seems to reduce overfitting, I suspect because there are more permutations of the data in each bag we end up with a more averaged fit on the graph and we can see a much lower out-sample error as shown below in Figure 2. compared to Figure 1. RMSE is reduced by an estimated 25%. Bagging also seems to make the RMSE of in and out sample data track together much earlier (around 20 leaves) than in the non-bagged graph (somewhere around 40).

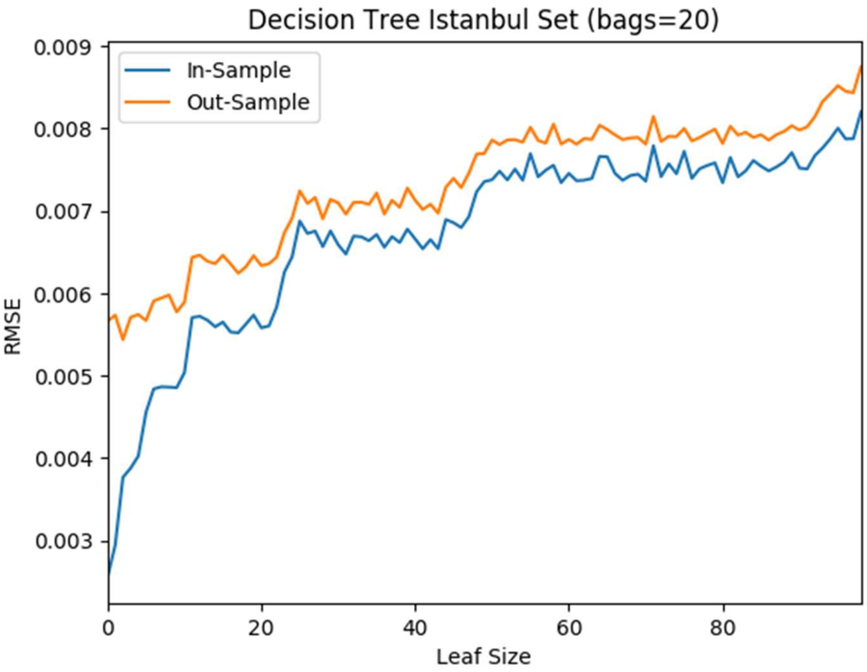

Figure 2. Decision tree using fixed bags and varying leaf sizes

Question 3

Quantitatively compare "classic" decision trees (DTLearner) versus random trees (RTLearner). In which ways is one method better than the other? Provide at least two quantitative measures. Important, using two similar measures that illustrate the same broader metric does not count as two. (For example, do not use two measures for accuracy.) Note for this part of the report you must conduct new experiments, don't use the results of the experiments above for this.

Answer: I assumed that DTLearner would build the tree slower than RTLearner because Prof. Balch mentioned in lecture that one of the most expensive operations was finding the best split and the RTLearner improves on this by performing this task randomly - the plot in Figure 3a. shows this to be true. I figured that the trade-off for this least-best split would be a tree that would perform worse when queried. However, much to my surprise after several runs, I didn't see a discernable difference in query times as indicated in Figure 3b. This may change with a larger data set but I couldn't find larger query times with the data sets that were given.

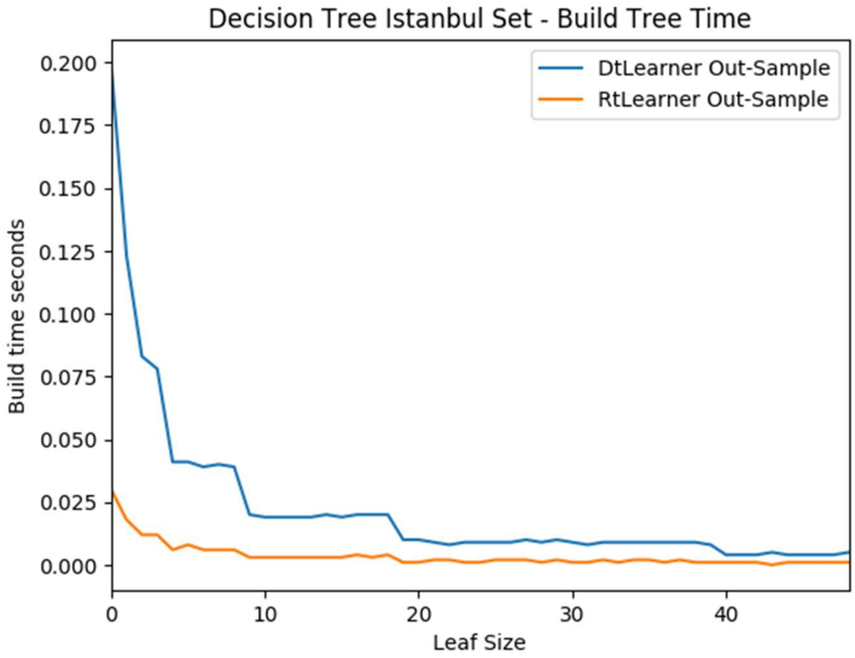

Figure 3a . Build tree times of DT and RT learners

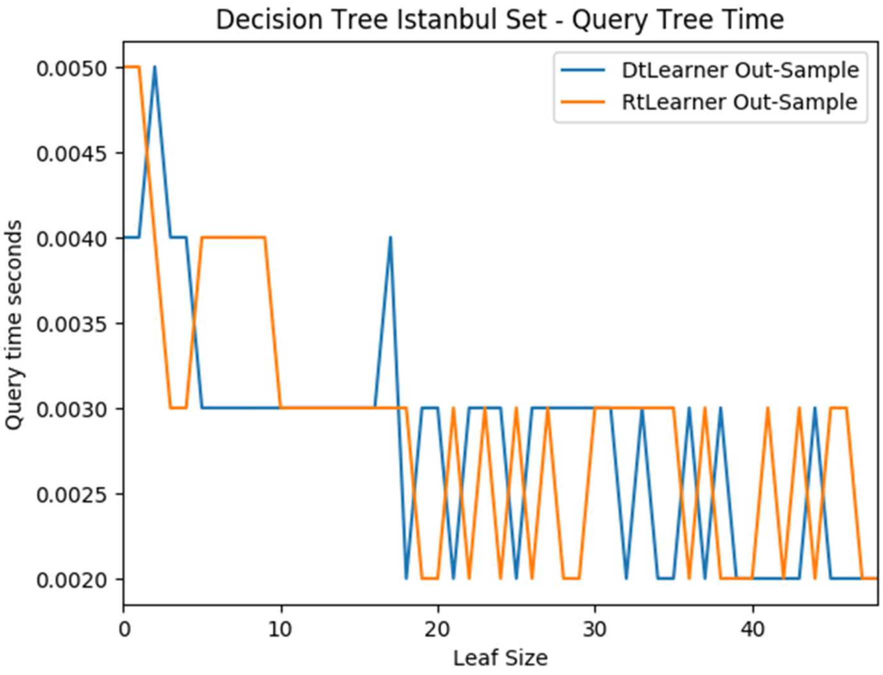

Figure 3b. Query times of DT and RT learners

I then performed a Mean Absolute Error (MAE) vs Leaf size comparing the absolute average of errors for DTLearner and RTLearner for out-sample data. Clearly we can see in Figure 4. below there is much more volatile error rate for the RTLearner.

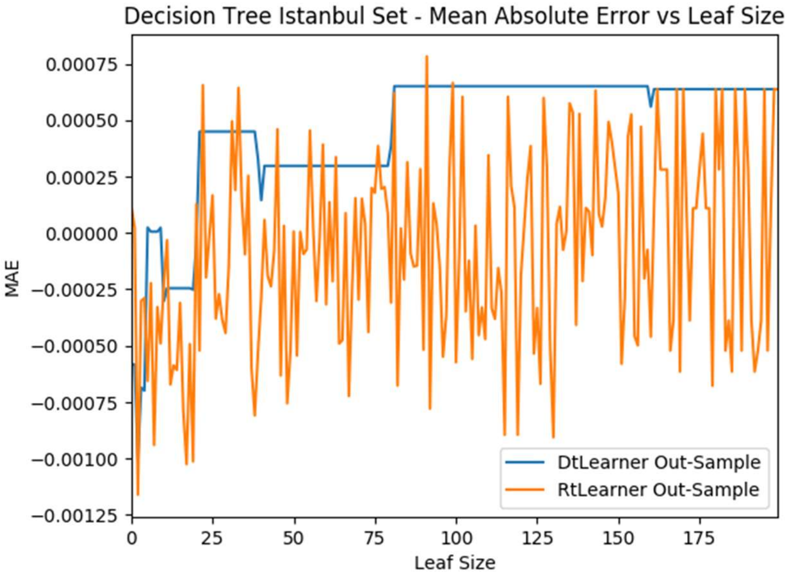

Figure 4. Out-sample RMSE vs Leaf size for DT and RT learners

The random tree shows high volatility as the error rates swings wildly up and down, sometimes finding great picks, and sometimes finding terrible picks. Where "good-enough" predictions are acceptable, the Random Tree may be the best implementation when performance is the most important consideration. The classical tree by comparison shows a much more stable MAE..