Project 3 - Assess Learners

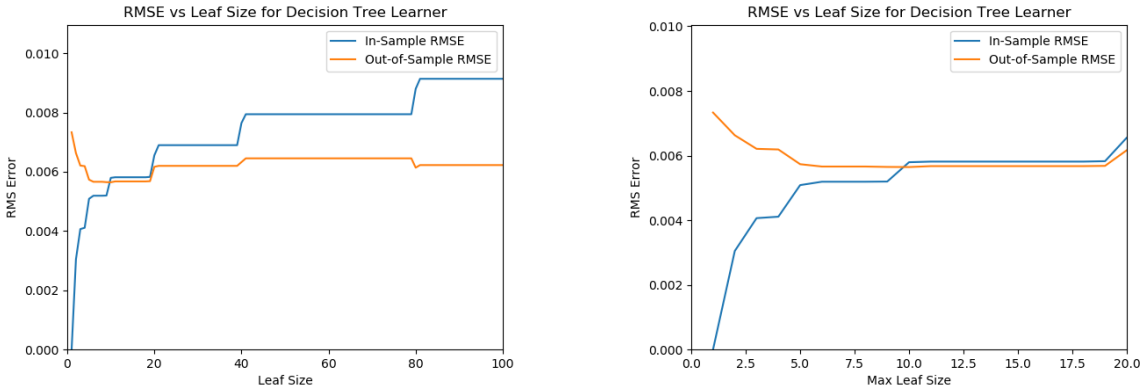
Shichao Liang
GTID: 903540912

September 21, 2019

A classic decision tree learner (DTLearner), a random tree learner (RTLearner), and a bootstrap aggregating learner (BagLearner) with variable ensemble size were implemented for simple, continuous features. An additional "insane" learner (InsaneLearner) was implemented as a 20-ensemble bootstrap aggregating linear regression learner, but that learner is excluded from this report. Since the data sets provided assume a regression problem, the learners were not implemented with classification problem features. Time order of the data is also ignored. For each learner, root-mean-squared-error (RMSE) and correlation averages can be calculated for a range of leaf sizes.

1 Overfitting with Respect to Leaf Size

To determine overfitting with respect to leaf size, an experiment was conducted in which classic DTLearners were trained on a portion of an initial data set, and tested on the remaining data. Each learner was trained with differing leaf size, ranging from 1 to 100.



(a) Max Leaf Size = 100

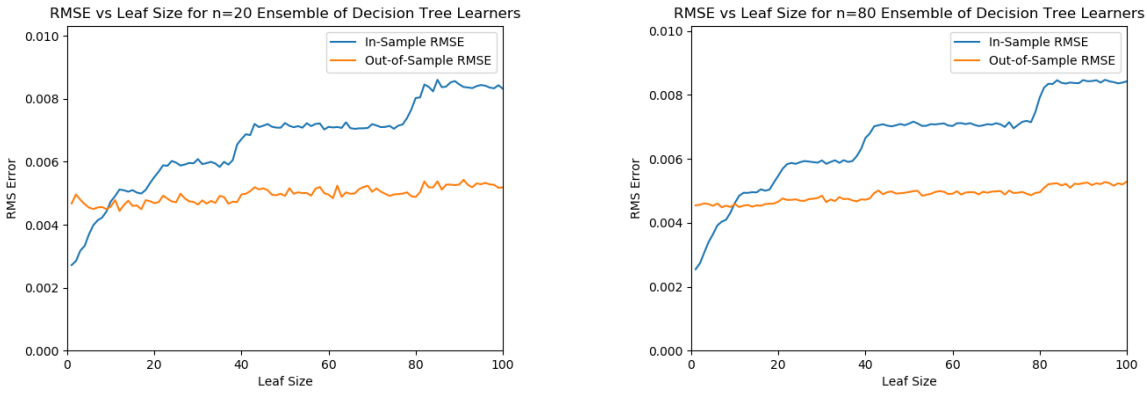
(b) Local around Leaf Size = 10

Figure 1: RMSE vs Leaf Size for DTLearner

From empirical results (Figure 1a), we can see that *overfitting*, the inability of the model to fit additional data due to too closely fitting training data, occurs when the leaf size is too small. Conversely, as leaf size approaches the size of the data set, deviation in RMSE increases. This could suggest *underfitting*, but more likely suggests that the feature set is *sparse*. From a local view of the intersection of the RMSE curves (Figure 1b), we can see that this inflection point occurs at leaf size 10, with overfitting increasing as leaf size decreases.

2 Bootstrap Aggregation Effects on Overfitting

Bootstrap aggregating, or *bagging*, involves random sampling of the training data with replacement (*bootstrapping*) and averaging the results of an ensemble of such learners (*aggregating*). To determine the effect of bagging on overfitting, the same experiment as in Section 1 was conducted, but this time with varying numbers (*bags*) of DTLearners trained on bootstrapped training data. RMSE and correlation values were calculated from the mean of queries for each of the learners in the ensemble. A subset of collected data is shown below ($n_{bags} = 20$ and $n_{bags} = 80$):



(a) Bag Size = 20

(b) Bag Size = 80

Figure 2: RMSE vs Leaf Size for BagLearner

From the empirical results[1], bagging does seem to reduce overfitting for the leaf sizes lower than the point determined in Experiment 1, but does not entirely eliminate overfitting. While the region of overfitting doesn't change, the deviation of in-sample RMSE from out-of-sample RMSE is much lower (Figure 2a). In addition, increasing the size of the ensemble of learners does not decrease the overfitting, but produces less noisy RMSE data. This makes intuitive sense, as variance is reduced or smoothed out from averaging a higher number of results from different BagLearners.

3 Classic Decision Trees vs. Random Trees

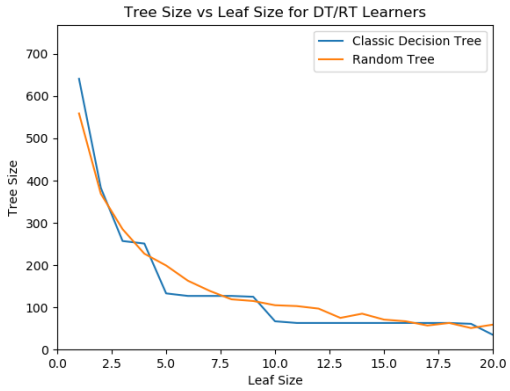
In terms of the performance of classic decision trees and random trees, we choose to analyze:

1. *Overall tree size*
2. *Tree construction time*

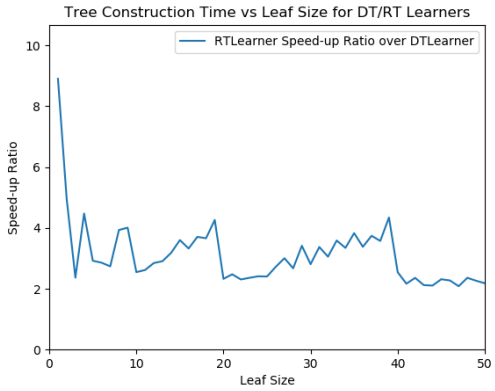
We construct a similar experiment as in Experiment 1. However, we collect data regarding the overall size of the trees for each leaf size, as well as the run-time for tree construction for each leaf size. In Figure 3a, we see that tree sizes are very similar for both DTLearner and RTLearner upon the Istanbul data set. This suggests that both learners will encounter the same memory constraints, and thus one is not more efficient than the other in that regard.

However, with regards to performance in tree construction, there are significant differences in run-time. In Figure 3b, we plot the result of calculating the ratio of run-times for DTLearner and

[1]Please note that the run-time of some of these experiments can take a significantly long time due to high bag count. In particular, the 80-bag experiment was edited out in *testlearners.py* to speed up runtime for submission.



(a) Overall Tree Size



(b) Tree Construction Time

Figure 3: Classic Decision vs. Random Tree Performance Metrics

RTLearner, respectively, in order to calculate the run-time advantage of RTLearner over DTLearner. Empirically, for low leaf sizes, speed-up ratio can be as high as 8, but seems to converge to 2 as leaf size increases. Since classic decision trees will calculate correlation to split at the best feature, random trees will save significant time in tree construction, but not in memory allocation. It is also likely that DTLearners generate models with better fit, but that is outside the scope of this experiment.[2]

[2] Ni