P3 AssessLearner Report

Yingdong Yang
yyang3052@gatech.edu

***Abstract*—This report discussed the performances of different supervised learning models by either changing the hyper-parameters of the same type or introducing the 'bagging' method adding randomness into the training process. In addition, this report discussed briefly the pros/cons of decision tree and random tree models.**

1 INTRODUCTION

Supervised machine learning models leverage the labeled datasets, identify the patterns between the inputs and the labels, and map with the outputs.
To train a supervised model, the original dataset would be split into the training set and the test set. The model will be based on the pattern or relationship found in the training set and then the test set will be used to evaluate its performance.
Tree model is one typical supervised learning model which will traverse the branches of the tree, ultimately assigning a class label (for classification) or a numerical value (for regression) to the input data. The decision-making process involves asking a series of questions based on the feature values, guiding the path through the tree until a prediction is reached. The split features and values can be determined in multiple ways. In this report, the decision tree would leverage root mean square error as the metrics for the split decision and the random tree would implement random split features and values instead.
To prevent overfitting, bagging was introduced which is able to pick a certain portion of the original dataset, train the model based on the new train and test sets, and repeat several times. After getting a set of outputs, the mean value would be calculated as the final output.
In the implementation part of this project, the decision tree, random tree, and bagging model were built. A few experiments were conducted to answer the following questions:
1. Does overfitting occur with respect to leaf size and what is the starting point and direction of overfitting? The decision tree model is used for analysis with the given dataset.
2. Can bagging reduce, or even eliminate overfitting with respect to leaf size? The decision tree model is used for analysis with the given dataset, with additional

bagging algorithms.
3. Between the decision tree and the random tree, which tree model is better quantitatively? Further discussion was also presented around the relative or absolute advantage of one model against another.

2 METHOD

The implementation and experiments of this report were conducted in Python (version 3.6). More details about the local environment setting can be found from local-environment.

3 DISCUSSION

3.1 Experiment 1

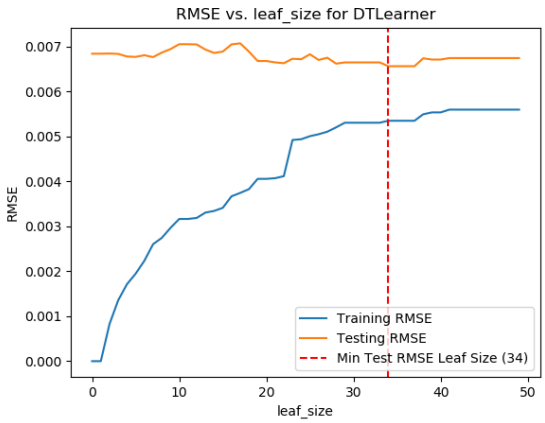

Figure 1—RMSE vs. leaf size for DTLearner

From Figure 1 above, the in-sample error decreased when the leaf size decreased, or, in other words, the number of leaves increased; while the out-sample error decreased in the beginning but increased when the leaf size was lower than 34. Therefore, The overfitting tended to occur when the leaf size became lower than a certain number.// From the experiment 1, the overfitting started from leaf size =10. When the leaf size increased (i.e. became larger and larger) the outfitting decreased and the model was underfitting. When the leaf size decreased the outfitting increased.

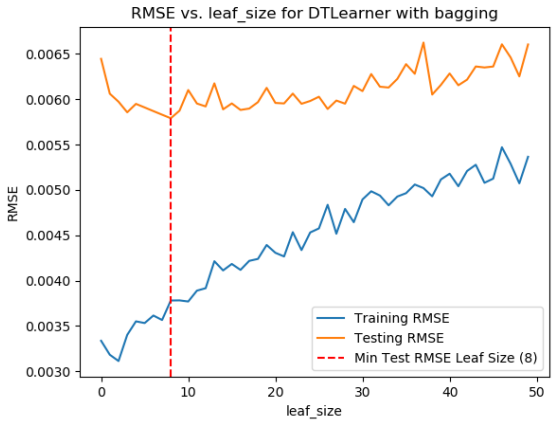

Figure 2—RMSE vs. leaf size for DTLearner with bagging

3.2 Experiment 2

According to Figure 2, the bagging could introduce more randomness when sampling the test and training sets. It could efficiently reduce the overfitting when the leaf size became smaller. However, it could not eliminate the overfitting. As shown in the chart, the error hit the global minimum when the leaf size was 8 and then increased when the leaf size increased. This indicated the outfitting still occurred even though the tree model was trained with bagging.

3.3 Experiment 3

In experiment 3, decision tree and random tree models were compared in terms of two quantitative dimensions: 1. training time. 2. maximum error.
The first metrics-training time was to evaluate the efficiency of training. It matters significantly if the training set is big or if there is a huge number of such models required.
The second metric- the maximum error was to evaluate the absolute largest error the models could reach. It could help the decision on the model choice when there is a more restrictive tolerance in the predictions.
Referring to Figure 3, the random tree had an outstanding performance in training time. No matter what the leaf size was, the random tree could train in almost a constant time; while the decision tree would use much more time and the time spent was growing linearly when the leaf size decreased. The reason for this was that the decision tree would choose the split feature and value by calculating the according metric (in this report mean square error was used). It costs time

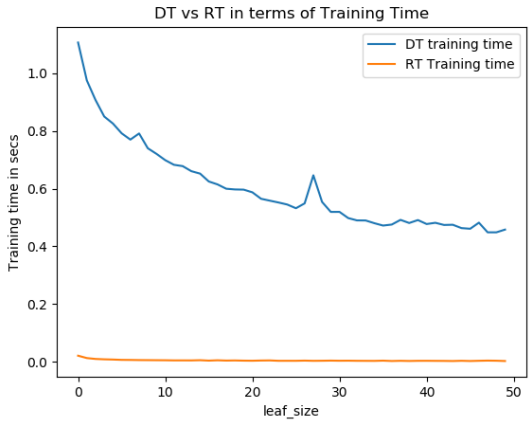

Figure 3—DT vs. RT in terms of training time

and resources to calculate. However, the random tree would pick random features and values for the split nodes and thus very limited time or resources are required.

Referring to Figure 4, the decision tree model had a much better performance in the test dataset for the maximum error. due to its ability to carefully select split features and values based on specific metrics (in this report, mean square error), optimizing the prediction process. On the other hand, the random tree model, which picks split features and values randomly, may not consistently minimize error, resulting in a comparatively higher maximum error in the test dataset.

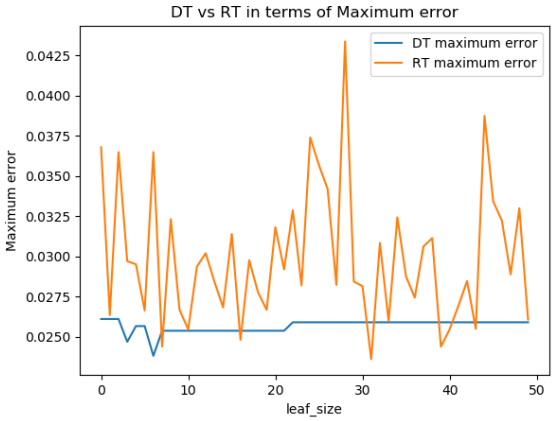

Figure 4—DT vs. RT in terms of Maximum error

Overall, between those two models, there was no superior advantage of one to

another. The random tree model sacrificed the accuracy of the resulting outputs for the little usage of training time and resources. It would be preferred if the dataset is large, the computational power is limited or the training needs to be repeated many times. On the other hand, the decision tree model would leverage the space and time for higher accuracy. The choice of model is highly dependent on the metrics, or more generally, the real-life question we want to answer.

4 SUMMARY

In summary, this report discussed the performance of supervised learning models, highlighting the effects of different parameters and the introduction of 'bagging' in training. It analyzed specific experiments on decision trees and random trees, providing insights into their performance and comparative advantages.