PROJECT 3: ASSESS LEARNERS

Malcolm Nathaniel Ng Bao Kun
mkun6@gatech.edu

Abstract—Decision trees are non-parametric supervised learning algorithms used for regression and classification problems. The purpose of this report was to determine how hyperparameters like leaf size can cause overfitting or underfitting to occur and at what magnitude. Furthermore, we look at different types of decision trees and other techniques that can be used to improve model performance and accuracy.

Introduction

The report will study the behavior and performance of the four Classification and Regression Trees (CARTs) algorithms, mainly the Decision Tree (DT) and Random Tree (RT) Learners, Bootstrap aggregating - (Baglearner) of DT excluding the Insane Leaner. The primary dataset includes financial data of a range of indexes' returns over the years. Using the select algorithms experiments will be conducted to evaluate the findings of its accuracy to predict the return for the MSCI Emerging Markets (EM) index. Quantitative measures like Root-mean square error (RMSE), Mean Absolute Error (MAE) and training time will be used to evaluate overfitting in relation to leaf size.

Methods

Data Preparation

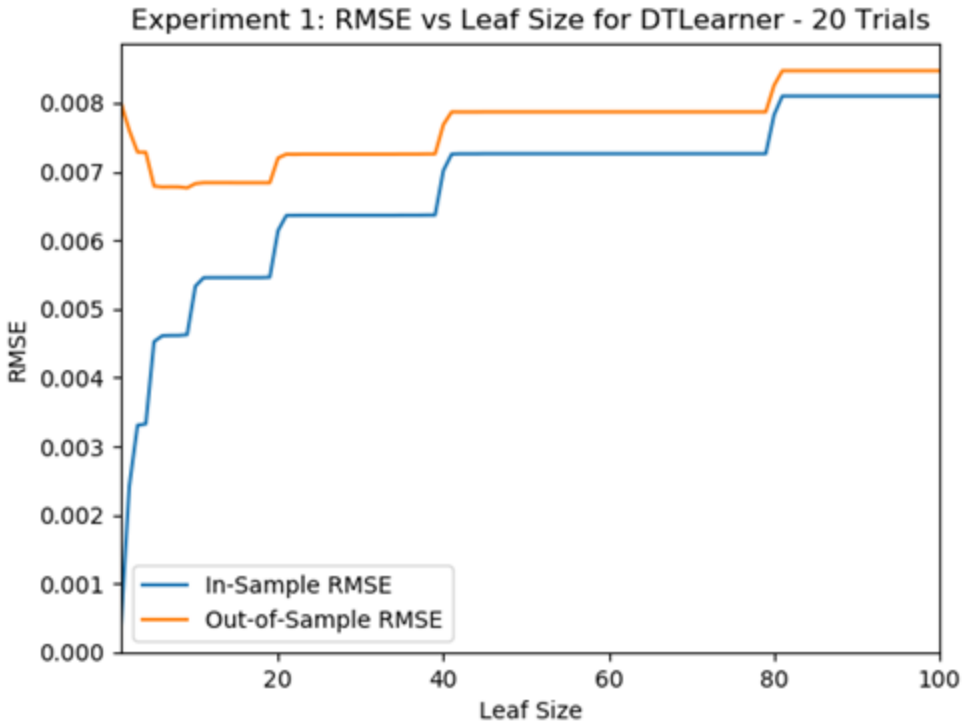
Data file is read in with NumPy's genfromtxt function. Then, treated as a non-time series dataset so we ignore the date column and only use numerical columns as our features. The training/testing split is 60% and 40% respectively. For each trial, we will select the training/testing dataset randomly with the NumPy's random permutation while retaining the split. We will use leaf size as the only hyperparameters to evaluate the model performance. Each trial will run the leaf size from 1 to 100. During the initial experiments RMSE for in-sample and out-of-sample will be recorded for every trial and later averaged to plot the findings.

Model Implementation

For the first experiment, the "classic" DT learner by *JR Quinlan* will be evaluated to determine an overfitting scenario. The second experiment will build an ensemble of DT learners to determine differences in performance from before. This method is known as bootstrap aggregation-bagging where each bag is trained on a random subset of the data with replacement and later also averaged. Lastly the DT and RT learners will be used to evaluate their performance with other quantitative metrics. We have used MAE and training time metrics to evaluate the performance.

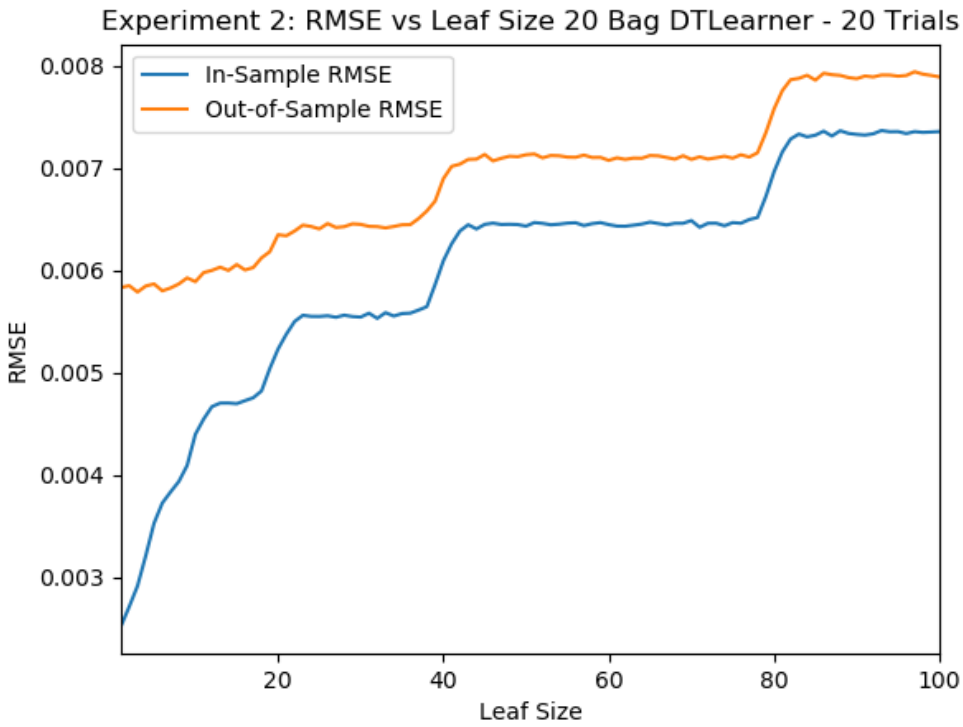
For the implementation of the DT "best feature to split on" will be the feature that has the highest absolute value correlation with Y. While the RT's feature will be picked randomly. The split value will be the median of the selected feature. The use of NumPy's random seed is called once for experiment and chart repeatability. Each learner has a function for training the data (i.e add_evidence) and testing the data (i.e querying the predictions). For best practice experiments are conducted for more than 3 trials minimum to give a more accurate picture. For the purpose of the report, all experiments are conducted for 20 trials.

Experiment 1



In this experiment, we will be using Root-mean square error (RMSE) as an evaluation metric to determine if the model is overfitting. From the empirical results, at leaf size ~8 there is an inflection point where out-of-sample RMSE increases and in-sample RMSE decreases; that is where overfitting occurs. From the starting point at leaf size 5 overfitting increases when leaf size decreases. The direction is from right to left. In short, when leaf size is small overfitting can occur, and when leaf size increases the model becomes more generalized and starts performing worse on both training and test data. Furthermore, we can explain that the model is underfitting but more likely due to the sparseness in the feature set during training.

Experiment 2



In Experiment 2, we attempt to use Bootstrap aggregating-bagging to build an ensemble of decision tree learners. Each learner will be using different set of data known as bags where it selects n-sample training data randomly with replacement. We are using 20 bags for this experiment. From the empirical results, we also notice a similar occurrence of overfitting at leaf size ~8 where the starting point is, noting the in-sample RMSE decreases while leaf size becomes small from right to left. While the out-of-sample RMSE is more stable at smaller leaf size before gradually increasing when leaf size becomes larger. We also note that the area of overfitting is reduced significantly (0.008 to 0.0058) from applying bagging but does not eliminate overfitting entirely. In addition, the deviation of both RMSE is much lower compared to a single learner.

Experiment 3

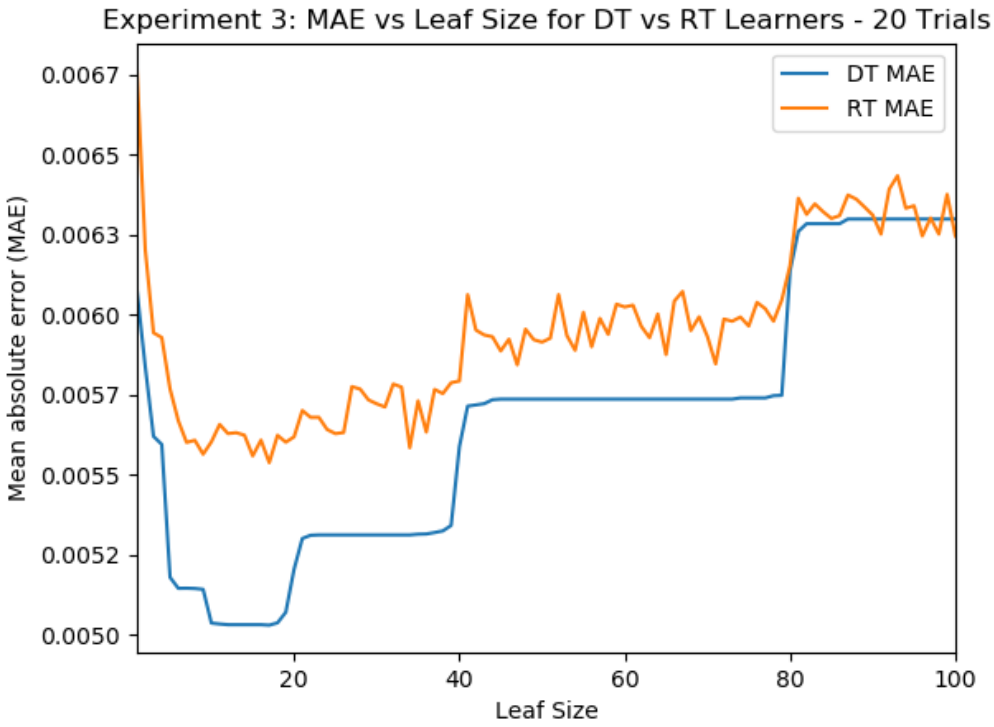


Figure 3a—MAE vs Leaf Size for Decision Tree vs Random Tree Learners over 20 trials

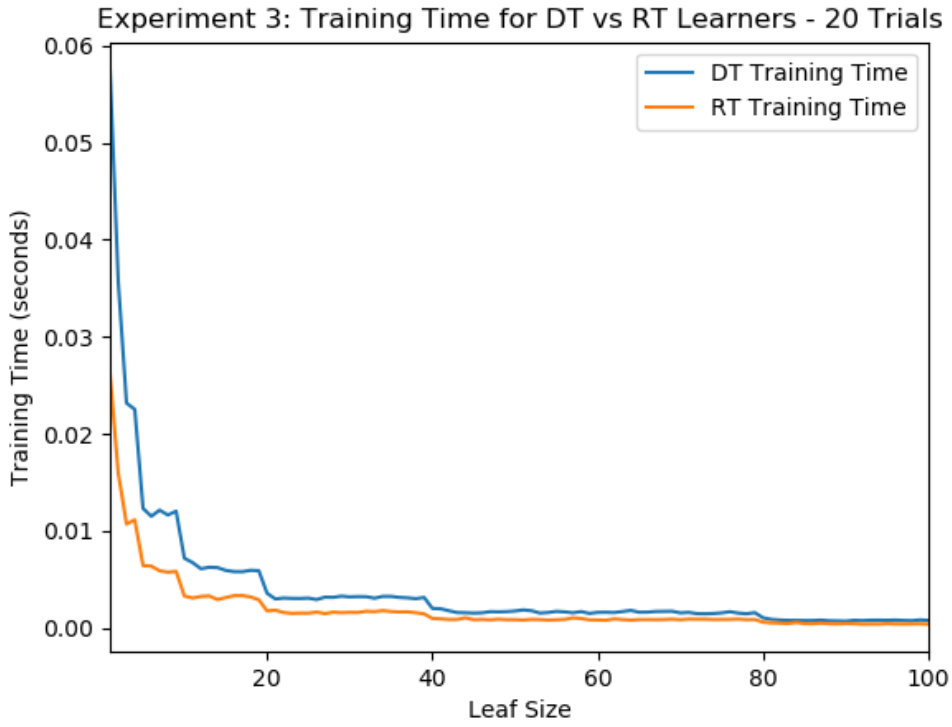


Figure 3b —Training Time vs Leaf Size for Decision Tree vs Random Tree Learners over 20 trials

In addition to RMSE metrics, additional metrics used to explain the performance will be Mean Absolute Error (MAE) and time to train. From the empirical results for figure 3a, it is clear that performance of DT is better than RT with significant lower MAE vs RT. Although, both models share similar overfitting occurrences at leaf size of 10 before increasing in errors in larger leaf size. Overall deviation for both model's MAE is lesser from leaf size 40 onwards and finally converges. In addition to the MAE result, the way DT was implemented where it uses correlation to determine the "best feature to split" may have influenced better performance vs the random selection of features of RT.

From the empirical results for figure 3b, we see that training time for RT is faster than DT difference in ~0.03 seconds at leaf size range 1 to 20. It would seem that the DT has better performance based on the factors that it has lower overall MAE deviation and that training time flattens while approaching larger leaf size. It is reasonable to say that the training time is minusculus for both learners as it converges overtime.

It could be argued that one learner is superior to another depending on the use case and situation. For cases when a DT is superior is when you want your model to be simple and easy to explain and have a clear idea of what feature is important for prediction as the tree is built on the entire dataset we can expect it to be more prone to overfitting and less rigorous predictions. On the other hand, RT is superior when you do not have a distinct idea of the kind of features to use for accurate prediction and want a more robust model that can be highly accurate and be less prone to overfitting and variance. But RT may be more complex to interpret than a DT and lastly the leaf size can affect the final result of both trees. As smaller trees may neglect some features resulting in poor prediction accuracy.

It would seem that the MAE method is a better way to explain the performance of the model as it is a more robust measure to data with outliers which is common. In contrast to RMSE, MAE is more desirable when large errors occur as it is not heavily penalised as errors are squared.

Summary

In conclusion, from experiment 1 and 2 we noted that by introducing bagging to a set of DT learners can outperform a single DT learner. By doing so, we also lowered RMSE and the model is less overfitting. Regarding model bias, by bagging it also reduces the individual biases that we see often in single learners. In experiment 3, we compared the differences in quantitative measures used in other experiments and new measures like MAE and training time. We found that MAE can be a good measure to use on a dataset with outliers that can cause large errors. While we also note that DT and RT is superior when applied correctly to specific problems. Overall, the new measures suggest that DT is a better model than RT with lower errors and reasonable training time. Further studies can be done to explore Random Forest performance as it relies on combining several DT which can significantly increase prediction accuracy than the 'classic' DT. Also to look into other information gain approaches to improve the best feature split on both for the DT and RT.