Project 3: Evaluation of supervised regression learning models

Ted Pham

Tpham328@gatech.edu

***Abstract*—This report evaluate the performance of different supervised regression learning models including decision tree, random tree, and bagging. Different quantitative measures are used to evaluate the performance of learning model include Root Mean Squared Error, R-Squared, running time, etc. There is no learner that is clearly superior than one another and the use of other improvement method like bagging, pruning and boosting is necessary to achieve better fit and lower overfitting.**

INTRODUCTION

In this assignment, we implement and evaluate four Classification and Regression Trees (CART) algorithm. Using the Istanbul data, which contains the returns of 8 worldwide indexes for several days in history. Our target variable is the return for the MSCI Emerging Market (EM) index. We are using 4 algorithms: decision tree, random tree, bootstrap aggregating and insane leaner to build predictive models for the EM index. We evaluate the performance of the four algorithm learners in term of accuracy, and overfitting to see if certain leaners are more superior than another.

METHOD:

For the purpose of these experiments, we did not consider time-series and therefore remove the timestamp of each record in our data. The eight different indexes returns are consider independent variables or features, and our target variable is the EM index return. We will create four predictive models from the CART algorithms to see if those features can be used to predict the EM index return.

From our original data, we randomly select 60% of the data to train on and uses the other 40% for testing purpose.

The decision tree learner is built from the training data. At each node of the tree, the best feature that is used as a criterion to split the data is based on its correlation with the target variable. Feature with higher correlation will have higher priority to split. The leaf node is concluded when the numbers of sample left on current node is lower than the leaf size, or when all sample in the node point to the same target variable value or when there is no correlation. The predicted value y on a leaf is the median of all actual value y on that leaf.

The random tree learner is built in a similar way as the decision tree learner but instead of selecting best feature based on correlation, it is selected randomly among the eight features. In both implementation of the decision tree leaner and random tree leaner, we select different leaf size from 1 to 50 to evaluate the overfitting result.

The bootstrap aggregation or bagging learner is built by repeating three different class learners: The decision tree and random tree learner above and the linear regression learner. For our experiment, we use bag size of 20, which mean we will built 20 bags, and run each bag using the decision tree learner (our implementation do allow the use of random tree and linear regression leaner in bagging as well). Each bag data is selected randomly and independently from our original training data (60% of total data) with replacement allowed. The number of samples in each bag is equal the number of sample in our training data. We estimated that 60% of sample in a bag are uniquely picked from training data and the rest is duplicate sample because of allowed replacement in our bag selection process.

Lastly, the Insane leaner is a repeated process of bagging learner. The insane leaner contains 20 bagging learner instances where each instance is composed of 20 learners from decision tree, random, and linear regression.

We run our learners through the testing data and evaluate accuracy using root mean square error or (RMSE). We also exploring some other quantitative measures to evaluate our learners. These metrics are Mean Absolute Error (MAE), R-squared, and time it takes to train the leaners.

In experiment 1, we run decision tree leaners with different leaf size from 1 to 50 in the data and evaluate the overfitting. In experiment 2, we run the bagging learner using the decision tree learner in each bag, 20 bags in total. For the last

experiment, we want to compare the decision tree and the random tree leaners using different quantitative measure, so we use MAE, R-squared and time to train. We conduct MAE and R-squared experiment in respect to leaf size and running time in respect to number of samples.

DISCUSSION

Experiment 1:

With leaf size equal 1, in most case, it means we run the algorithms until each leaf end up with only one sample. In uncommon case, a leaf might have more than one sample if all samples in that leaves have the same target variable value. But overall, the resulting tree is complicated with over 600 nodes for leaf size of one. It is obviously overfitting, RMSE is close to 0. As leaf size increase, overfitting decrease. Look at chart in figure 1, with leaf size increase from 1, RMSE of testing data set actually decreases or get better. It means increasing leaf size from 1 make the learner less overfitting and more accurate, a good thing. With leaf size decrease from 50, we can see that orange line start going flat while blue line going down further away from orange line, this is when overfitting starts. One may say around 5, it is more obvious overfittings starts, but around leaf size equal 10, we see that decreasing leaf size, the RMSE out of sample not decrease much, which does not make the learner better but blue line already start going down. Over fitting start below leaf size of 10 and as leaf size decrease, overfitting appears more often.

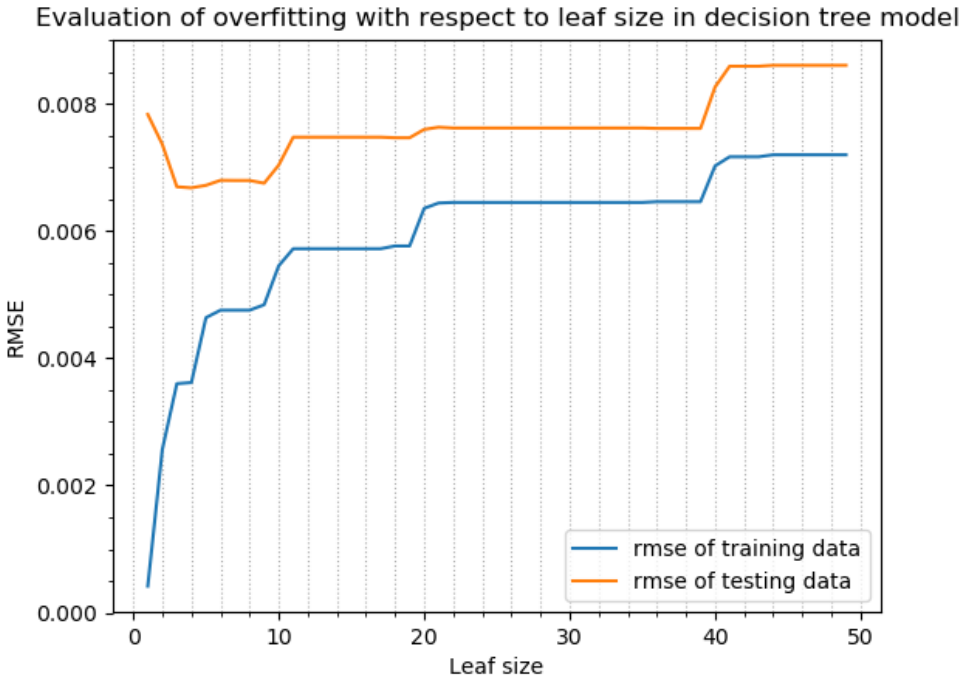

Figure 1: RMSE vs Leaf size of decision tree learner

Experiment 2:

In this experiment, we run bagging learner with bag size of 20, using the decision tree learner in each bag. We run the decision tree using different leaf size from 1 to 20. It is difficult to find overfitting in the chart as both orange and blue line show a steady increase as leaf size increase, there is no divergence of two lines compared to experiment 1. So, bagging reduced and eliminated overfitting with respect to leaf size in this case.

However, in reality, bagging does not always eliminate overfitting completely. It is possible for overfitting to occur in bagging if the base leaners in this case, the decision tree is overfitting to the training data. With leaf size decreasing starting from 10, we can clearly see that the gap or difference between RMSE of training and testing data is getting bigger and bigger, so there might be possible for overfitting here and further analysis is required as using RMSE alone as measure of fit is not enough.

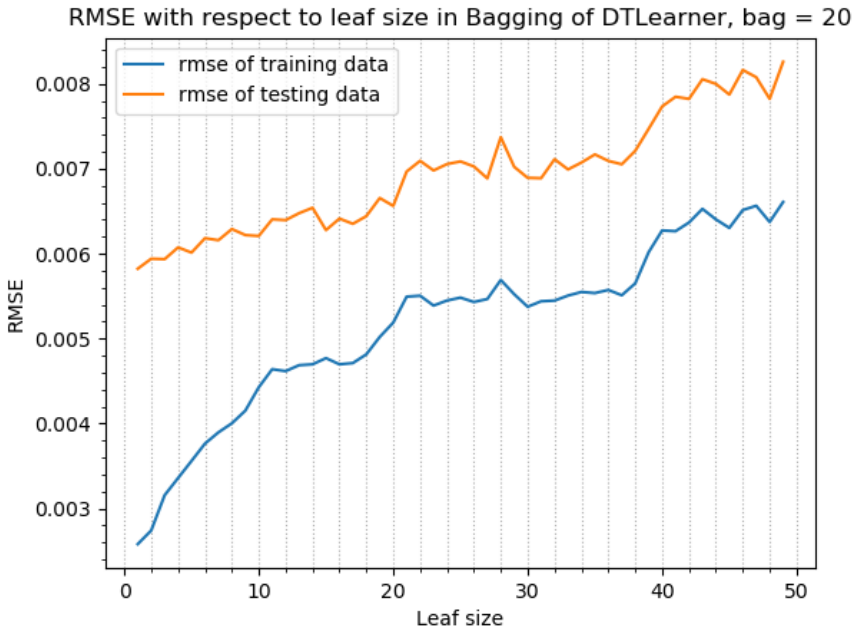

Figure 2: RMSE vs leaf size of bagging learner

Experiment 3:

To quantitatively compare decision trees versus random trees learners, we used the Coefficient of Determination (R-Squared), the Mean Absolute Error (MAE) and running time to evaluate learners' performance.

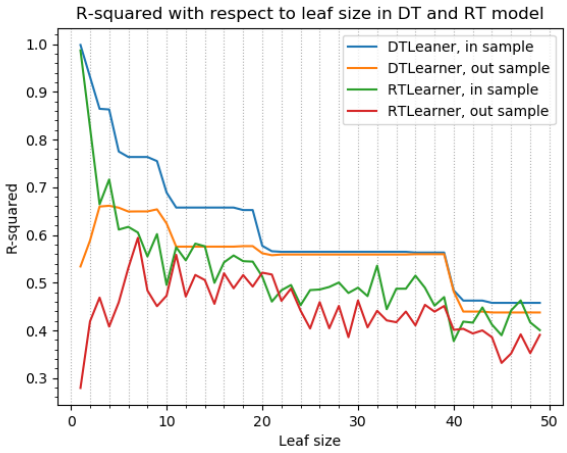

Figure 3A: R-squared

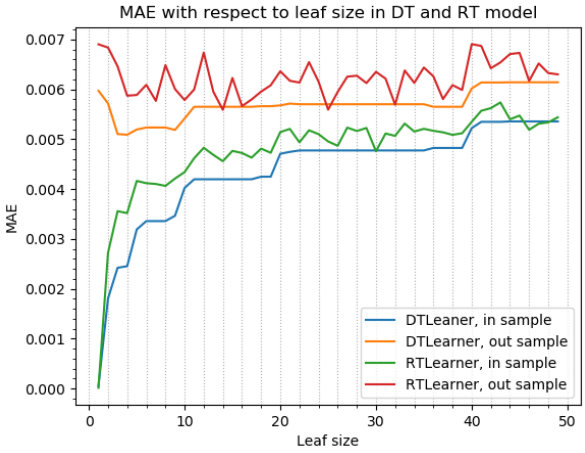

Figure 3B: MAE

R-squared measure the proportion of the variance in the features that is explained by the target variables. A higher R-squared that is closer to 1 (1 is maximum) indicates that the model is a good fit. From figure 3A, we can see that in both in and out of sample, the DT learner results in a higher R-squared. Therefore, Decision Tree (DT) learner is better than random trees (RT) in term of goodness of fit. It is also noted that as leaf size increase the difference in R-squared of DT and RT learner gets smaller, indicating that less differences between two learners. So as leaf size increase, performance of both leaners are closer to be the same, but it is not high indicating that both models might not be a good fit if leaf size increase.

Mean Absolute Error, is calculated as average of the absolute differences between predicted and actual values of target variable.

$MAE = (1/n) * \Sigma \, |y\_actual - \hat{y}\_predict|$

Lower MAE means better fit. Based on figure 3B, DT learner have lower MAE than RT learner in both in and out of sample cases. So DT learner is a better performance in term of fit if we use MAE as measurement. This is very similar to the use of RMSE in experiment 1 and 2.

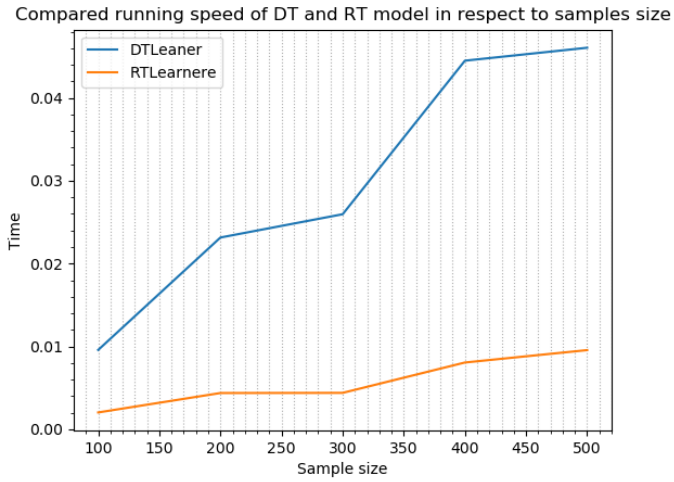

Figure 3C: Running time vs sample size

Lastly, we conduct running time to measure speed performance of both leaners. For this experiment, we used the total data available because we want to compare the running time with respect to sample size. Larger samples required longer

running time. Based on figure 3C, it seems RT learners is obvious winner here. As sample size gets bigger, the DT leaner take significantly more time to run. Calculating correlation to decide best feature (DT) is more complicated than taking random feature as best feature (RT).

It seems that no learner is superior to another. In term of speed, RT is better. In term of fit, DT is better.

Based on R-squared, as leaf size increase, the performance of DT and RT is closer to one another, and both are not a good fit when leaf size is more than 10. With smaller leaf size, both model is likely to overfit. Therefore, the use of other improvement method is necessary for both DT and RT learner. They include bagging, pruning, and boosting.

SUMMARY:

In conclusion, the decision tree learner with lower leaf size might lead to overfitting. In comparison with random tree learner, the decision tree learner is a better fit model but required more computing power. As leaf size increase, both learners perform significantly worse in term of fit. But as leaf size decrease, both are subjected to overfitting. Therefore, the use of other improvement method like bagging, pruning and boosting is necessary to achieve optimal performance (better fit, but reduce overfitting)

REFERENCES:

https://realpython.com/python-recursion/

https://en.wikipedia.org/wiki/Coefficient_of_determination

https://en.wikipedia.org/wiki/Overfitting

https://en.wikipedia.org/wiki/Bootstrap_aggregating