CS 7646 ML4T
Project 3: Assess Learners

Chen Peng
c.peng@gatech.edu

Abstract—This project considers a regression problem. Given a set of data, we try to find the connection between features and result for each sample. Using the connection we found, we can predict the result for another set of data based on their features. In this project, we construct and tested decision tree learner, random tree learner, and bag learner to model the connection. After we build our models, we test the on a validation data set. The result of prediction accuracy and computational cost for each algorithm is analysed and discussed.

1 INTRODUCTION:

In this project, we consider a regression problem. In this problem, we are given a set of data that are consist of N samples, each sample has M features X and a result Y. We need to use this $N \times (M+1)$ data set to build and validate our prediction algorithm. In this project, we ignore the time order between each samples. We assume it is a static data and treat the samples as independent.

In order to solve the problem, we implemented and validated a classic decision tree learner (DT learner), a random tree learner (RT learner), and a bag learner. A required insane learner is also build for validate the bagging algorithm.

The classic decision tree model we build for this project is a binary tree. A hyper parameter 'leaf size' is set for limit the maximum number of samples that are assigned to a leaf. For each node, we use the feature X_i which has the highest absolute correlation with result Y as the best feature to split the data. For each best feature, we use the median of the remaining samples as the split value to make the tree more balanced. On each leaf, it aggregate samples no more than the leaf size. We use the mean value of sample result Y as the leaf value and return as the regression result, since each sample has the same contribution to the leaf.

A random tree learner was build based on the classic decision tree learner. Since the feature selection step in the decision tree learner occupant the main cost in the training process, the random tree learner randomly choose a feature as the split feature at each node in order to avoid the large calculation of correlation.

The bag leaner is a simple ensemble learning algorithm which contains a bag of single learners. The bag learner can reduce the over fitting of a single leaner and avoid the bias from a single learner. In this project, the bag learner gather the prediction result for a bag of single learners and return the mean value as the single learners are equally weighted.

2 METHODS:

In this project we set up three experiments to validate the effectiveness of our algorithm and test the efficiency.

In Experiment 1, we validate the classic decision tree learner (DT learner) by calculate the required root mean square error (RMSE) under different leaf size condition. The data set we used in this experiment is Istanbul.csv. For pretreatment of the data, We shuffle the data order first to erase the time dependent. Then we choose the first 60% of samples as the training data set (in sample), and the last 40% of samples as the validation data set (out sample).

In Experiment 2, we validate and test the bagging with bag learner. The bag learner use 20 single DT learners and the same data set as in Experiment 1. For data pretreatment, the shuffle the raw data, erase the date, set the 60% of samples as training set, the 40% samples as validation set. For each single learner in the bag, we randomly select samples with replacement from the training set, and the number of samples for each single learner is same as the number of samples in training data set. We calculate the RMSE under different leaf size to study the overfitting effect and compared with the single DT learner in Experiment 1.

In Experiment 3, we quantitatively compared the classic decision tree learner and random tree learner. We use the same dataset and pretreatment as in Experiment 1. Since RMSE and correlation are not allowed in this experiment as required, we calculate mean absolute error (MAE) and R square of the prediction of the validation data set under different leaf size to evaluate and compare the efficiency of the DT learner and RT learner. We also compute and compare the training time and query time of the DT learner and RT learner under different leaf size with same dataset to evaluate the cost and efficiency of the two learners.

3 DISCUSSION:

3.1 Experiment 1:

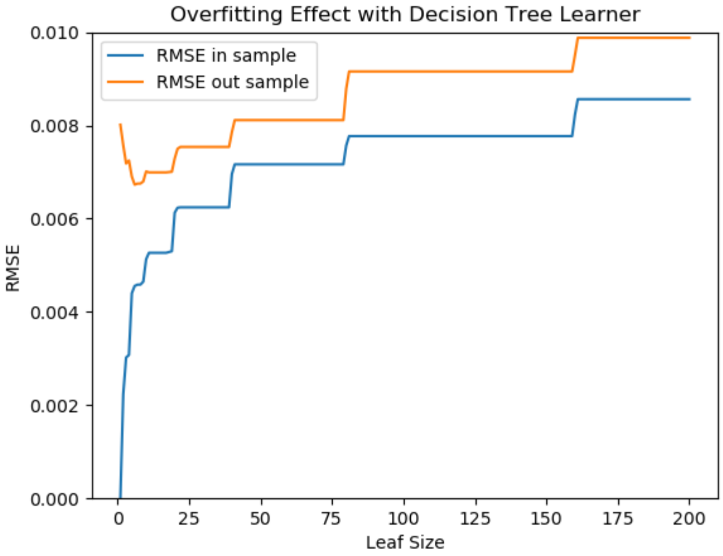

Figure 1—Observation of overfitting effect with decision tree learner using Istanbul.csv dataset. Here the blue curve is the root mean square error of the training data set, the orange curve is the root mean square error of the validation data. The horizontal axis leaf size is the maximum number of samples to be aggregated at a leaf.

In this experiment, we study the overfitting effect of a classic decision tree learner (DT learner). We use he root mean square error (RMSE) as a metric:

$$\mathrm{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1}$$

The overfitting effect appears when the out sample RMSE increases with the in sample RMSE decrease. As the Figure 1 shows, the RMSE vary with the leaf size. When the leaf size reduces, the RMSE for training data (in sample) decreases and eventually reach 0 with leaf size is 1. In contrast, the RMSE for validation data (out sample) first decreases until leaf size reach 6, than the RMSE significantly increase as the leaf size reduce from 6 to 1. Here, the overfitting effect occur at leaf size equal 6, and become more significant with reduce of the leaf size.

3.2 Experiment 2:

In this experiment, we compare the RMSE of training and validation data with the use of bagging algorithm and a single DT learning algorithm. The bagging contains 20 single DT learners. Figure 2 shows the results of in sample and out sample RMSE with different leaf size limit. Comparing with a single DT learner, the bag learner generally has a smaller RMSE with both in sample and out sample data. This is because the bagging learner considers a bunch of single DT learners and avoid the bias produce by each single learner. The predicted result $\hat{y}$ with large deviation in a single learner are averaged. Meanwhile, the metric RMSE itself put more weigh on the value with large deviation, compare with other metrics such as root mean square log error (RMSLE).

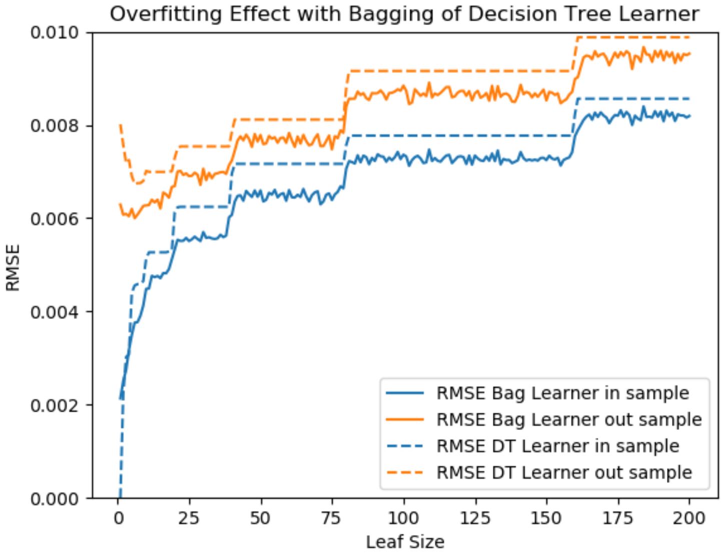

***Figure* 2**—Comparison of bag learner and decision tree (DT) learner using Istanbul.csv dataset. The simple bagging contains 20 classic DT leaner. Here the blue lines are the root mean square error (RMSE) of the training data set (in sample), the orange lines are the RMSE of the validation data (out sample). The solid lines are the results of the bag learner, the dashed lines are the results of the single DT learner as a comparison. The horizontal axis leaf size is the maximum number of samples to be aggregated at a leaf.

Compare the RMSE of in sample and out sample for the bag learner, we found that the overfitting effect is significantly reduced from a single DT learner. The increase of RMSE on the out sample with reduce of leaf size is not quite obvious when leaf size is small. However, we still observed an increase of RMSE on out sample when leaf size reduce from 6 to 1. In this experiment, the bagging can reduce overfitting with respect to leaf size, but not totally eliminate it. The overfitting start at 6 with the reduce of leaf size. This is same as the DT learner in Experiment 1.

3.3 Experiment 3:

Answer:

In this experiment, we compare the classic decision tree versus random tree learner (RT learner). Here we use metrics mean absolute error (MAE) and coefficient of determination (R square) to measure the effectiveness of the two algorithms. We also use the training time and query time to compare the efficiency of the two algorithms.

The mean absolute error, also known as L1 norm, measure the mean deviation of the predicted values $\hat{y}$, and defined as:

$$MAE(y, \hat{y}) = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{2}$$

The coefficient of determination (R square) reflects the proportion of all the variation of the dependent variable that can be explained by the independent variable through the regression relationship. The R square value is from 0 to 1. The larger value represents that the independent variable can explain larger percentage of the change in the dependent variable. It's defined as:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{3}$$

In Figure 3, the left figure shows the comparison of out sample MAE of DT learner and RT learner. Using MAE as metric, we notice the DT learner generally has a smaller error and more accurate result than the RT learner. Also, the result of DT leaner is more stable as the leaf size changes, so that we can find the best training leaf size easily. The R square metric shows that The predicted $\hat{y}$

can be explained better on the DT learner both in general and the best-leaf-size situation than the RT learner. Compare these two metrics, the DT learner performs better on the effectiveness in the experiment.

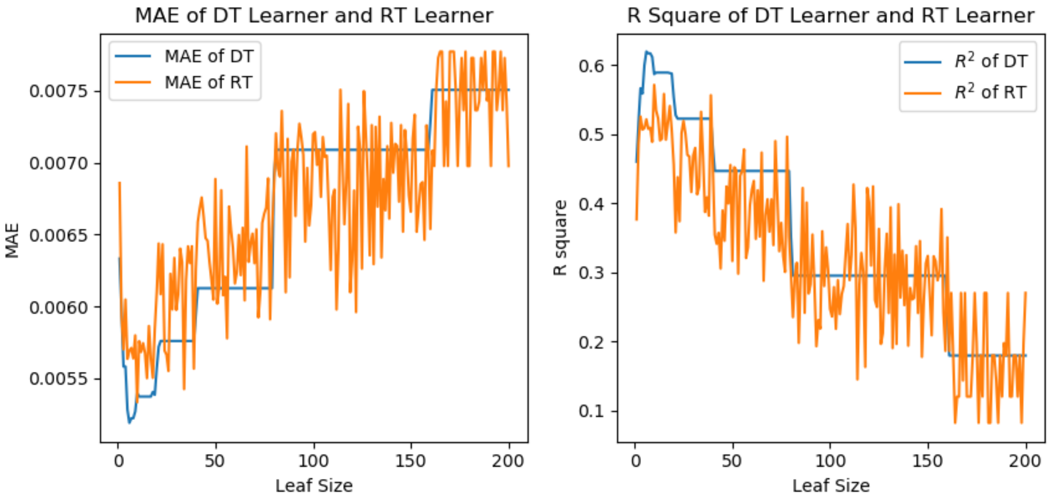

Figure 3—Comparison of statistical results of classic decision tree learner (DT learner) and random tree learner (RT learner) under different maximum leaf size. The left figure shows the mean absolute error of DT learner and RT learner predict on the validation data (out sample). The right figure shows the R square value of the validation data. In both figures, the blue curve is the result of DT learner, and the orange curve is the result of RT learner

Beside the effectiveness, we also study the efficiency of the two algorithms by comparing the computational cost of training and query. In Figure 4, the left figure shows the training time of DT learner and RT learner with changing the leaf size, right figure shows the query time of the two learners.

In the figure, we notice the training time of the RT learner is about a half of the DT learner in each leaf size condition. This is because the random tree replaces the calculation and selection of the best feature in each node when training with randomly select a feature. In contrast, the query time has no obvious difference on each leaf size in general. This is due to the same scale and structure of the tree. Thus, the RT learner is more efficiency on training than the DT learner. For the query efficiency, two learners have same performance.

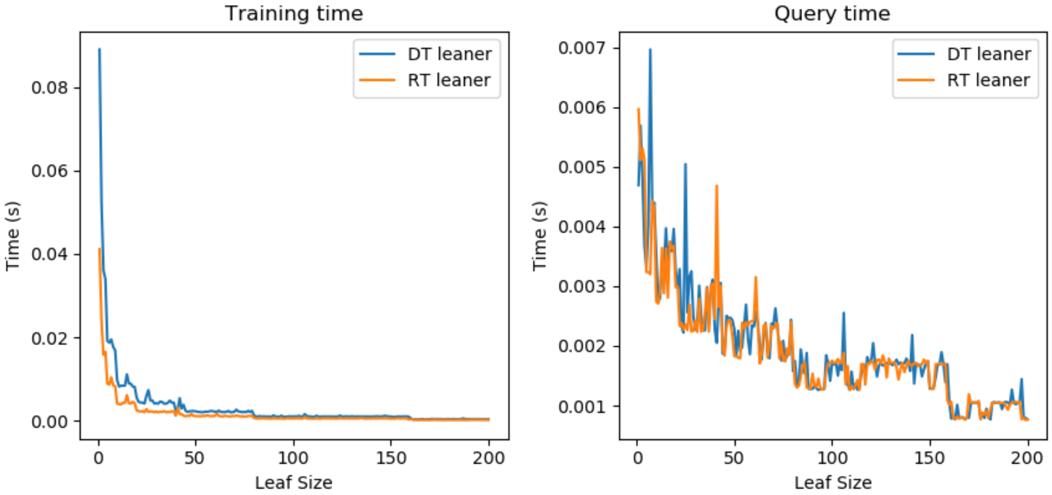

Figure 4—Computational cost comparison of the classic decision tree learner (DT learner) and the random tree learner (RT learner) under different maximum leaf size. The left figure shows the training time of DT learner and RT learner based on the same training data set. Right figure shows the total query time of each DT learner models and RT learner models predict on the same validation data set. In both figures, the blue curve is the result of DT leaner, and the orange curve is the result of RT learner.

4 SUMMARY:

In this project, we study the effectiveness and efficiency of the decision tree learner, the random tree learner, and the bag learner. In experiment 1 and 2, both of the DT learner and RT learner shows the overfitting effect when the leaf size reduce less than 6. The bagging can significantly reduce the overfitting, but can not eliminate it. In experiment 3, we found the DT learner is more accuracy on prediction than the RT learner. In contract, the RT learner has about a half traning time than the DT learner. The query time of the two learners has no obvious difference in this experiment.