**Abstract**

The goal of this project is to understand overfitting using Random Tree Learner (RTLearner). This report focuses on analyzing the influence of leaf-size and bagging on overfitting.

**Background**

In statistics and machine learning, a common task is to build a predictive model using training data, which can make reliable predictions on unseen data. Thus, its performance is determined not by the training data but by unseen data. The criterion used for training the model typically is to maximize its performance on seeable data (training set), whereas its efficacy is determined by its ability to perform well on unseen data. Overfitting occurs when a model fits too well to the training data so it is not generalized enough to perform well on unseen data, since it describes too much of the random error or noise in the training set, instead of focusing on the underlying relationship. It “overreacts” to minor fluctuations in the training data.

When overfit, the model is excessively complex. In parametric models, too many parameters relative to the number of observations typically lead to overfitting. As an extreme example, if the number of parameters equals to or greater than the number of observations, the model can perfectly predict the training data simply by memorizing it entirely. Yet such a model typically fail to make good predictions on unseen data.

Non parametric models can be even more susceptible to overfitting than parametric model, so many methods have been developed to overcome it. This report tests some of them.

**Experimental Methods**

Experiments was done using the Istanbul data. Random Tree Learner (RTLearner) and BagLearner were implemented according to the project requirement (Ref1).

At high level, RTLearner works like this: When building a tree1) randomly select a feature 2) randomly select two values in the feature 3) use the average of these two values as split point 3) split samples into branches based on the split point 4) repeat this process until all the branches became leafs that have uniform value within or until the leaf size is less or equal to the defined leaf size. When scoring a tree 1) locate the sample into a leaf based on the tree structure and split values 2) use the average value of the leaf as the value of the sample.

At high level BagLearner works like this: 1) randomly sample with replacement N records from training set which has N records 2) build a model with this randomly sampled set 3) repeat this sampling and model building process according to the number of bags.

The Istanbul data was randomly split into train and test datasets using 6:4 ratio. Models are trained with train dataset and Root-Mean-Square Error (RMSE) is evaluated with test dataset. Different number of leaf sizes and number of bags were experimented on, and RMSE(s) under difference scenarios were plotted to show the effects. More details of the experiments can be found in each specific session.

**Results**

As describe in the background session, overfitting is a common problem in predictive modeling. Decision tree model is particularly susceptible to overfitting. An extreme case is building a single tree with leaf-size one. In this situation, the training sample can fit 100%. RMSE can be reduced to 0 and correlation can be increased to 1. Unless the training sample does not have any noise, otherwise the model is overfitting the random error in the training set which cannot be generalized to test data.

Because of predictive model algorithm’s potency to overfitting, many mitigation methods were developed. The first category of methods is Pruning, which is frequently used in tree models. Common pruning methods include increasing the leaf size and/or restricting the depth of the tree. The second category is ensemble methods, which use (weighted) average of multiple models, such as RTLearners. Bagging falls into this category. A third category of methods is to vary the training sample to hide full information and/or add noises. Dropout and training data transformation (often used in modify image data for neural network) falls into this category. Other methods such as regularization are also used, which will not be described in detail here.

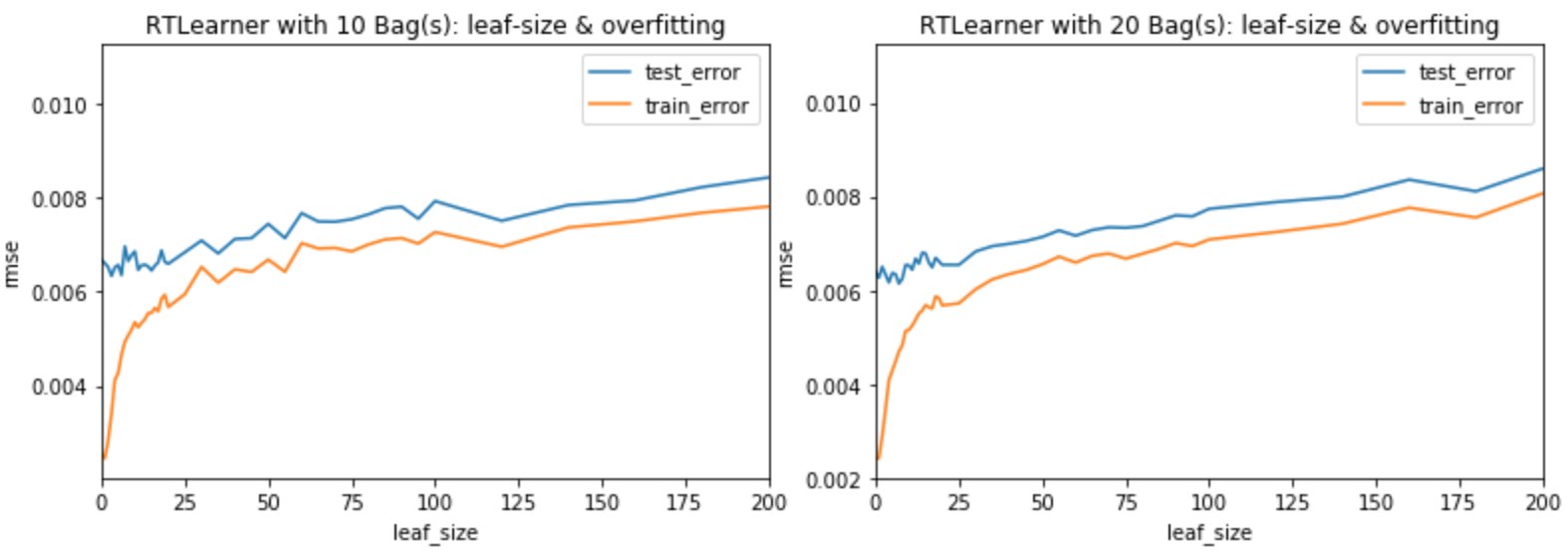
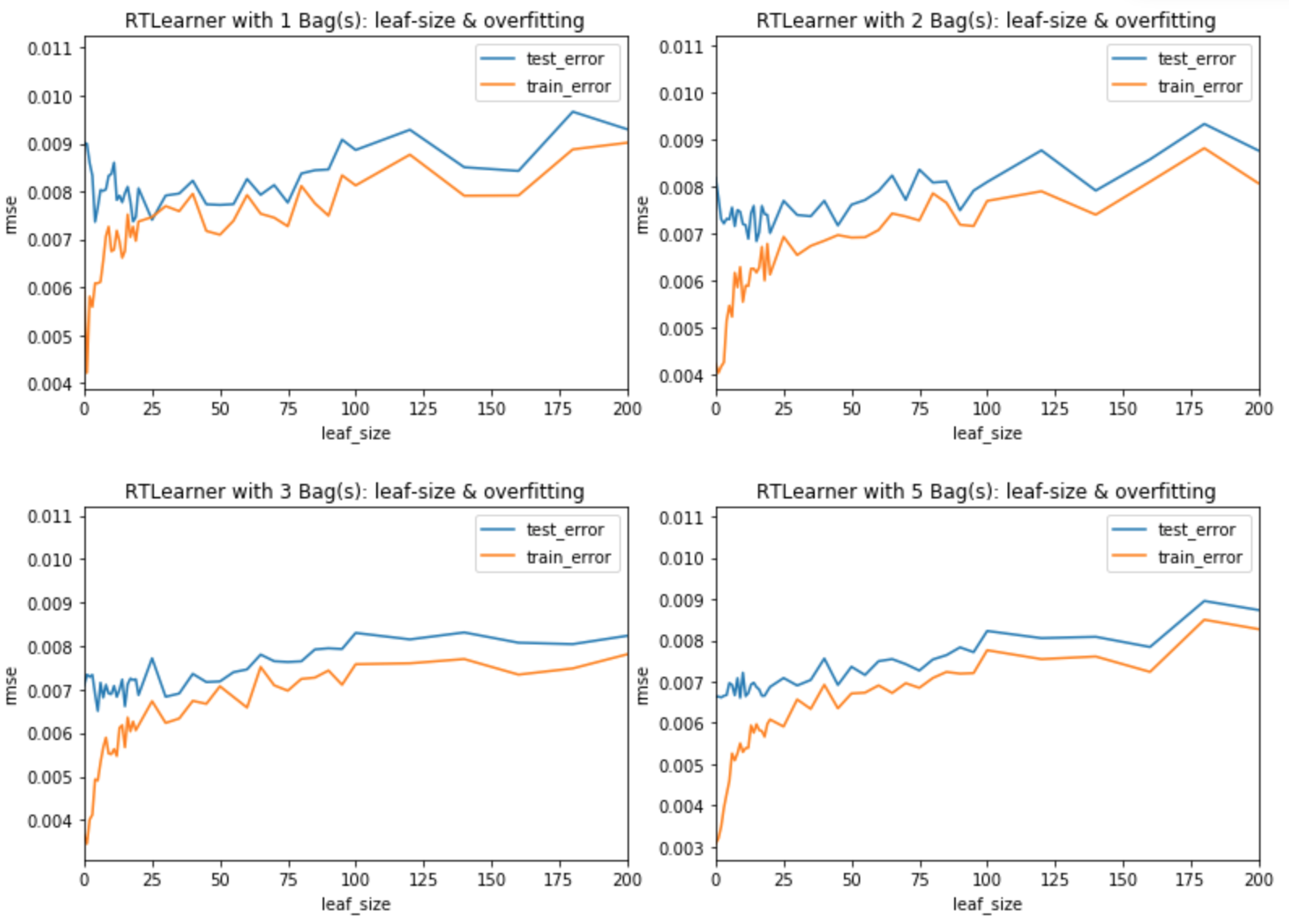
This report experimented with the first two categories of methods, and assess their effect on overfitting.

Pruning: prevent overfitting by increasing leaf size

In this experiment, leaf size was varied to test its effect on overfitting. A single tree is created and no bagging method is implemented. Training error and testing error are plotted to investigate the model fitness.



**Figure 1: Overfitting and underfitting with different leaf size.**



As shown in **Figure 1**, when decreasing the leaf size, the training error keep dropping and the test error started increasing. This is a typical phenomenon of overfitting. Due to the randomness of the RTLearner, it is hard to identify the exact number of leaf size when overfitting occurs. Based on the specific experiment in **Figure 1**, it started around leaf size 13, which is marked using a vertical red line. When the leaf size drop below 13, the models start to overfit.

However, the lowest leaf size without overfitting varies not only according to modeling methods used (such as bagging, gradient boosting, etc.), but also varies with the sample. Sample size and noise level are two major contributors.

Another interesting observation to point out is that increasing leaf size excessively lead to underfitting, which is marked by increase of both training and testing error.

Thus, choosing the right leaf size that neither overfit nor underfit is critical in pruning.

Ensemble: prevent overfitting by bagging

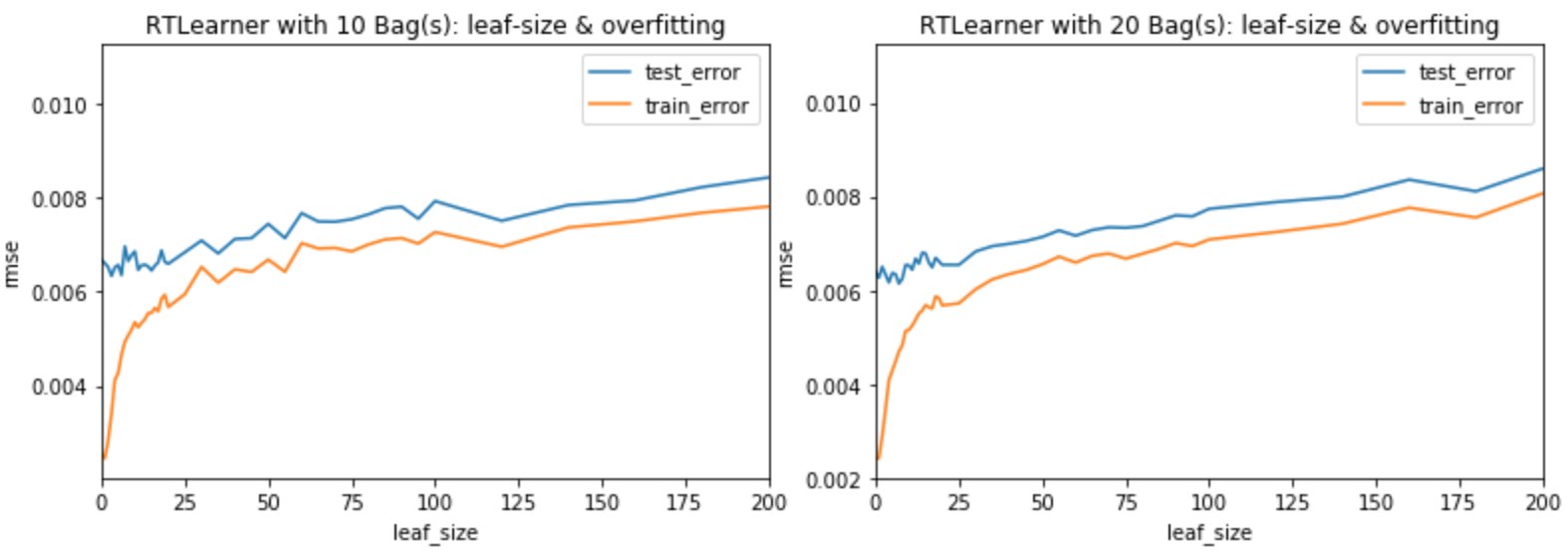
In this set of experiments, RTLearner is used in combination of bagging methods (BagLearner). Each panel in **Figure 2** used a fix number of bags and the leaf sized is varied. Training error and testing error are plotted to investigate overfitting.

**Figure 2:** **Overfitting and underfitting with different leaf size when using bagging**

As shown in **Figure 2**, when using 1 bag, overfitting still occurred. Although the starting point of the overfitting is hard to define due to the randomness of the tree, it sure overfit when the leaf size is 1. This is understandable because 1 bag essentially is a resampling of the training data. Although this resampling make the model does not overfit to the original training data, it overfit to the newly sampled data. Thus, the model cannot generalize well to the test set.

Even with just 2 bags, the overfitting started to diminish. With the bag number increased to 20, decreasing leaf size lead to similar or better performance.

The hallmarks of overfitting also have subtle differences. The most commonly acknowledged one is the increase of testing error when training error decreases. Another phenomenon some time also consider as “overfitting” is the increase of the spread between training and testing error, even when the testing error still decreases. This signals that the model is dramatically better tailored to the training set than the test set. By this definition, the overfitting still occurs when the leaf size drops below 15 despite the implementation of bagging.



Bagging: improve prediction when leaf size is fixed

Besides the set of experiments describe in the previous session, another set of experiments was performed to address the relationship among bagging, leaf size and fitness. The leaf size of the RTLearner was fixed at 1 and 20 and bag number was varied in this experiment.

**Figure 3: model fitness with different bag numbers**

As shown in Figure 1 and Figure 2, when the leaf size is 20, overfitting generally does not occur. Adding the number of bags improves the fitness (Figure 3). On the other hand, leaf size 1 leads to obvious overfitting without bagging (Figure 1) and with low number of bags (Figure 2). When increase the bag number, overfitting start to diminish and fitness also improves at the same time until plateau. This indicate bagging is a very good way to prevent overfitting at the same time achieve high fitness on testing sample.

**Summary**

Overfitting is a common pitfall in predictive model building. Many methods have been developed to overcome it. This report clearly showed two of the methods pruning (varying leaf size) and ensemble (bagging) methods are very effective in preventing overfitting.

**References:**

Ref1: <http://quantsoftware.gatech.edu/MC3-Project-1>

Does overfitting occur with respect to leaf\_size? Consider the dataset istanbul.csv with RTLearner. For which values of leaf\_size does overfitting occur? Use RMSE as your metric for assessing overfitting. Support your assertion with graphs/charts. (Don't use bagging).

Overfitting / leaf\_size question:

Is data (either a chart or table) provided to support the argument? (-5 points if not)

Does the student state where the region of overfitting occurs (or state that there is no overfitting)? (-5 points if not)

Are the starting point and direction of overfitting identified supported by the data (or if the student states that there is no overfitting, is that supported by the data)? (-5 points if not)

**Overfitting, leaf size and bagging**

Can bagging reduce or eliminate overfitting with respect to leaf\_size? Fix the number of bags and vary leaf\_size to investigate. Provide charts and or tables to validate your conclusion.

Overfitting / bagging fixed, change leaf\_size:

Is data (either a chart or table) provided to support the argument? (-5 points if not)

Does the student state where the region of overfitting occurs (or state that there is no overfitting)? (-5 points if not)

**Overfitting and number of bags**

Does overfitting occur with respect to number of bags? Choose some leaf\_size and keep it fixed. How does RMSE vary as you increase the number of bags? Does overfitting occur with respect to the number of bags? Support your assertion with graphs/charts.

Overfitting / leaf\_size fixed, change number of bags:

Is data (either a chart or table) provided to support the argument? (-5 points if not)

Does the student state where the region of overfitting occurs (or state that there is no overfitting)? (-5 points if not)

Are the starting point and direction of overfitting identified supported by the data (or if the student states that there is no overfitting, is that supported by the data)? (-5 points if not)

References:

Ref1: http://quantsoftware.gatech.edu/MC3-Project-1