Building and Improving Tree-based Models

Kun Gao

***Abstract*—**We identify the overfitting issue associated with the Decision Tree model using a financial dataset and show that using the bagging ensemble method can reduce the overfitting problem. We also compare the accuracy and efficiency of the "classic" Decision Tree with its variant, i.e., the Random Tree model.

1 INTRODUCTION

Decision Tree is a widely used machine learning algorithm for its simplicity and reasonable accuracy. However, an intrinsic issue related to Decision Tree model is overfitting, which refers to the fact that the trained model has high accuracy in predicting the training data but does poorly in new (previously unseen) data. In this project, we train a Decision Tree model using a financial dataset and explore how the overfitting problem manifest itself with varying leaf size. We then further examine if using an ensemble method, namely the bagging method, could reduce the overfitting problem. Finally, we compare the "classic" Decision Tree with its variant, the Random Tree model. Our initial hypothesis is that the Decision Tree is prone to overfitting when the leaf size is small and using the bagging method could reduce the overfitting problem.

2 METHOD

2.1 Training and Test Dataset

We use the Istanbul dataset for all the experiments in this project. The objective is to predict the return for the MSCI Emerging Markets index (Y) based on the other index returns (X_1, X_2, …, X_8). We split the entire dataset into two subsets, the training (60%) and test (40%) subsets. We found that if the total dataset is unshuffled, the out-sample model error (evaluated using the test data) can often be smaller than the in-sample error (evaluated using the training data). This indicates that the test subset is easier to predict than the training subset. Therefore,

to ensure the training and test data have similar quality, we first **shuffle** the total dataset randomly and then split the total into the training and test subsets.

2.2 Experiment 1

Experiment 1 is designed to identify the overfitting problem associated with the Decision Tree model. We vary the leaf size from 1 to 50 (with an increment of 1), and then train the model accordingly. The upper bound of leaf size is set to 50 because we do not think a larger leaf size would be a reasonable choice considering there are only 321 records in the training dataset.

2.3 Experiment 2

Experiment 2 is designed to examine if using bagging method can control the overfitting problem. Bagging refers to the method in which we ensemble multiple instances of the same model, each trained on a slightly different subset (or bag) of the training data. We conducted four groups of experiments. In each group the number of bags is fixed and the leaf size is varied from 1 to 20 (with an increment of 1). The number of bags in the four groups is set to 2, 5, 10 and 20, respectively.

2.4 Experiment 3

Experiment 3 is designed for comparing the Decision Tree and Random Tree models. We conduct two groups of experiments using each model, one group does not apply the bagging method, and the other uses the bagging method with the bag size set to 10. Similar to Experiment 1, the leaf size is varied from 10 to 30 (with an increment of 2) for both Decision Tree and Random Tree models. Note we do not investigate the overfitting problem here, and therefore we no longer consider small leaf size in Experiment 3.

3 DISCUSSION

3.1 Experiment 1

Here we discussion the overfitting problem associated with the Decision Tree, with a focus on how this problem manifest itself with decreasing leaf size. Figure

1 shows the change of Root Mean Square Error (RMSE) with the leaf size evaluated using both the training and test datasets.

3.1.1 Does overfitting occur with respect to leaf size?

Overfitting does occur with respect to leaf size. As can be seen from Figure 1, with leaf size decreasing roughly from 15 to 1, the model error in predicting the training dataset decreases. Not surprisingly, when leaf size is 1, the model even preforms perfectly on the training dataset. However, when evaluated using the test dataset, the model error grows quickly with decreasing leaf size (when the leaf size is smaller than 6). This is a clear indication that overfitting occurs when leaf size is small.

3.1.2 For which values of leaf size does overfitting occur?

Overfitting occurs when the in-sample error decreases while the out-sample error increases. Based on this criterion, it is clear that overfitting occurs when the leaf size is equal or less than 5. With leaf size smaller than 5, the model RMSE gets smaller in the training dataset but larger in the test dataset.

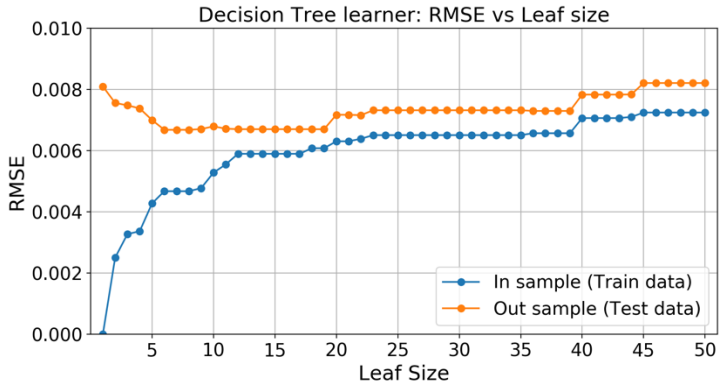

Figure 1—RMSE as a function of leaf size in a Decision Tree model.

3.2 Experiment 2

The above experiment shows that the decision tree model is prone to overfitting when the leaf size is small. Here we discuss if using bagging can effectively control the overfitting problem.

3.2.1 Can bagging reduce overfitting with respect to leaf size?

The bagging method does reduce the overfitting problem. As shown in Figure 2, the in-sample error still decreases quickly when the leaf size becomes roughly smaller than 10, but the out-sample error is relatively stable (not increasing rapidly) at the same time. Even using a minimal bag number of 2 (Figure 2a), the overfitting problem is not as significant as in Experiment 1.

3.2.2 Can bagging eliminate overfitting with respect to leaf size?

Bagging does not eliminate overfitting. As shown in Figure 2, the out-sample error does not show the similar decreasing trend as in the in-sample error with decreasing leaf size (leaf size smaller than 10). When the bag number is small (2 or 5), the out-sample error still trends upward slightly with decreasing leaf size. This indicates that the overfitting issue still exists even using the bagging method.

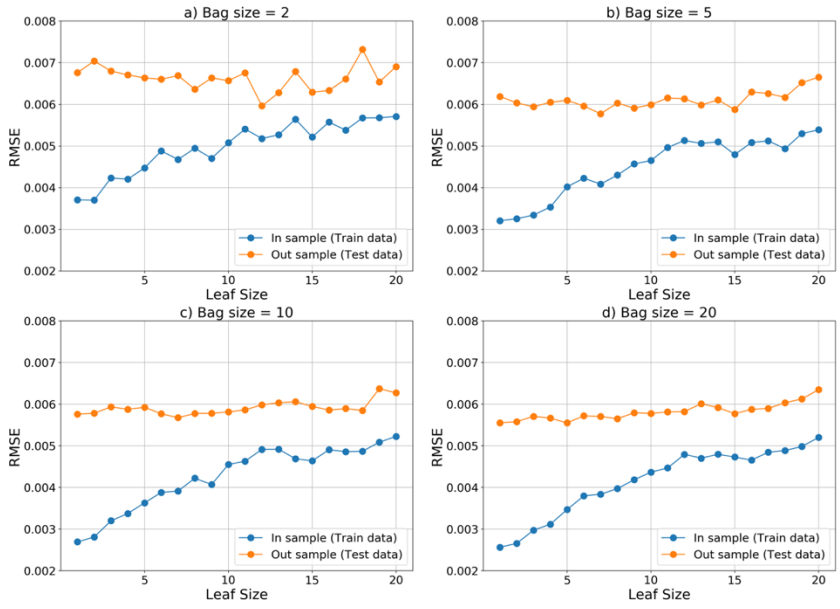

Figure 2—RMSE as a function of leaf size when bagging method is used. The bag number is indicated in the figure titles.

3.3 Experiment 3

Here we quantitatively compare the decision tree and random tree models from two perspectives: the model accuracy and the cost of training the model. We use

two metrics to measure the model performance and target to address the question in which way(s) is one model better than the other.

3.3.1 Accuracy: Mean Absolute Error

Here we use Mean Absolute Error (MAE) as a metric to measure the model accuracy, which is defined as follows.

$$MAE = \frac{1}{N}\sum_{i=1}^{i}|y_i - \hat{y}_i|$$

Figure 3 show the model MAE evaluated in the training and test datasets. A few important points can be inferred from Figure 3. First, for both Decision Tree and Random Tree models, using the bagging method can improve the model accuracy (dashed lines have smaller errors than the solid lines). Secondly, the Decision Tree model has **higher accuracy** than the Random Tree regardless of whether the bagging method is used or not (blue lines have smaller errors than the yellow lines). This is an expected result for the dataset considered. As shown in Figure 4, some features (Xi) have higher correlation with the labels (Y) in the Istanbul dataset. The Random Tree model just picks random features when splitting the data and it will likely miss the most relevant features in the training process. In contrast, the Decision Tree model detects the most relevant features recursively in the training process, which should result in a better performance.

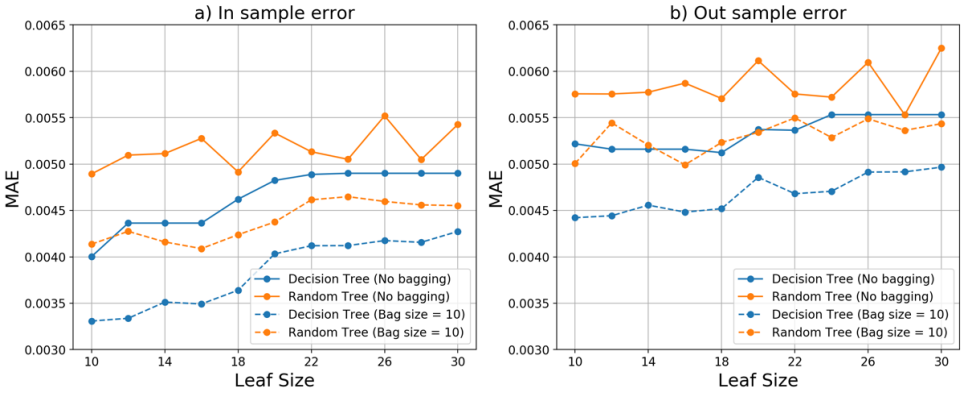

Figure 3—MAE as a function of leaf size in Decision Tree and Random Tree models with or without bagging. a) shows in-sample error and b) shows out-sample error. Solid lines represent the group not using bagging and dashed lines represent the group using the bagging with a bag size of 10.

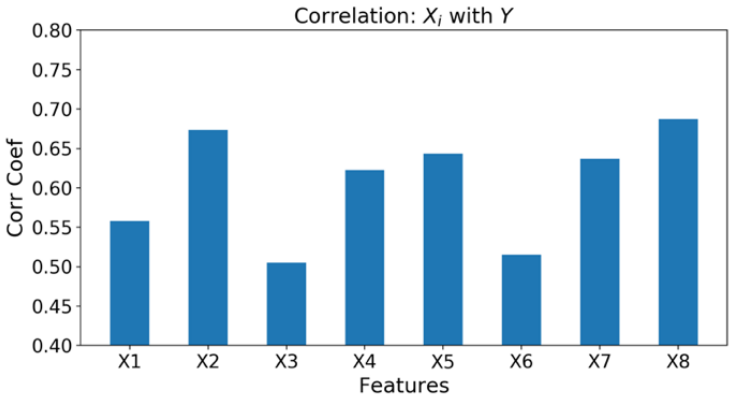

Figure 4—Correlation coefficients between features (X1 to X8) and labels (Y) in the Istanbul dataset.

3.3.2 Efficiency: Time spent on training the model

The cost of training a given model is an important factor to consider when we have a very large training dataset with a lot of features. Here we compare the time spent on training the Decision Tree and Random Tree models using the same training dataset on the same computer (Figure 5). Clearly, the Random Tree **takes less time** to train. This is because in the Decision Tree model we need to recursively scan through all the features to find out the one that has the highest correlation coefficient with the labels in the subset, which is a costly process. The Random Tree just pick a feature randomly. For this reason, the training time for the Random Tree model is only about 10-15% of that for the Decision Tree model. We can expect that in the case when there are a large number of features and all features have similar correlations with the labels (Y), the Random Tree could be a better choice considering its faster training speed.

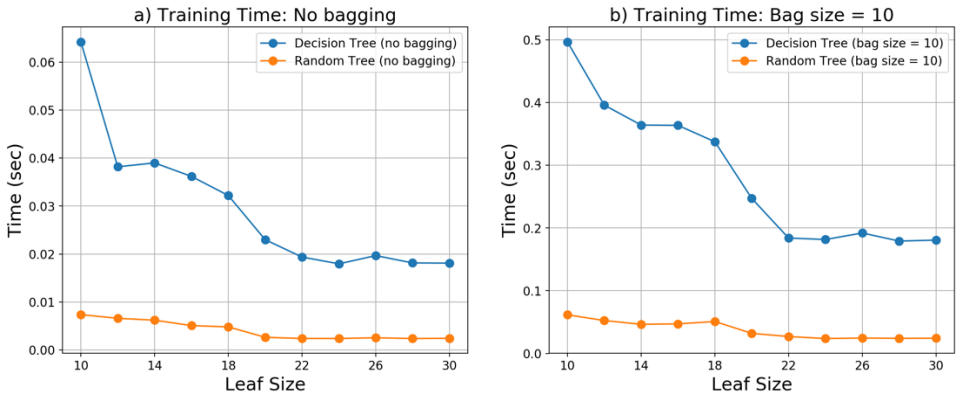

Figure 5— Time spent on training the Decision Tree and Random Tree models as a function of leaf size. a) shows the training time when bagging method is not applied, while b) shows the training time when bagging method is used and the bag size is set to 10.

4 SUMMARY

In this project, we explored various aspects of the Decision Tree model. Key results are summarized as follows. The Decision Tree model is prone to overfitting when the leaf size becomes smaller than a certain number. We also show that using the bagging method can reduce the overfitting problem. When bagging method is used, although the in-sample error still decreases quickly with decreasing leaf size, the out-sample error is relatively stable. Finally, we show that the Decision Tree model has a high accuracy than the Random Tree because its exhaustive feature selection method. Also, for the same reason, it takes much longer time to train the Decision Tree model. In the Istanbul dataset we used, some features have noticeably higher correlation coefficients with the labels. It would be interesting to compare the accuracy of the Decision Tree and Random Tree models using a dataset in which all features have similar correlation with the labels.