Evaluating Random Forests

Summary

An option that we didn't cover while evaluating regression models in our last study were Decision Trees. Decision Trees are very powerful, yet simple and easy to understand algorithms. An added benefit being they are versatile as they can perform both classification and regression tasks. One downfall to Decision Trees is that they are very sensitive to small changes in the training data. To factor this in, we decided to evaluate a Random Forests algorithm, which is basically averaging the predictions of a number of trees, and then compare it to the evaluations from the previous case study.

Research Design

Once again we will be using the data from the Boston Housing study to determine which regression model would best help us determine the median value of homes. The four models from the previous study were Linear, Ridge, Lasso and Elastic. The mechanisms of a tree or forest are very different to the rest, in that instead of finding a correlation to different variables along some sort of line equation, the data within the variables are dropped into different buckets/nodes depending on the weight of their influence and separation from other sets.

Data Analysis

The data used was the same as the previous Boston Study evaluation. The neighborhood field was dropped, integers converted to floats and the data was scaled as regularized data can be sensitive to the scale of input features.

Model Evaluation

Similar to the previous study we once again used a cross validation design with 10 folds and the root mean squared error (RSME) as an index or prediction error to identify the best model. To evaluate a Random Forest, Scikit Learn's RandomForestRegressor was used. Appendix A, Figure 1 shows the RSME

results obtained for each model using default features for the Random Forest. You will notice that the Random Forest has the lowest value in the test set making it the most suitable model. The reason I pulled the results for the training set was so that I can get an idea of the over-fitting occurring. For example looking at the training and test results for Random Forests, the difference in value is around 0.34, which is much greater than the difference between the rest of the models and hence determines there is a reasonable amount of over-fitting in training.

Trying to obtain a better RSME value for Random Forests we decided to alter the max_features meta parameter. The maximum value you can have depends on the number of explanatory variables, in this case being 12. A max feature of 1 eliminates the possibility to study interactions, which is when you have two variables in the same branch. Appendix A, Figure 2 shows the results obtained for various values of max_feature; in this case a max_feature of 12 had the lowest RSME.

In order to further refine the Random Forest model we decided to use bootstrap and set the number of estimators equal to 100. The purpose of Bootstrap Sampling is to get a diverse set of classifiers, and this is achieved by sampling subsets of rows and replacing them. The results can be seen in Figure 3 of Appendix A. The lowest value of 0.444 was obtained, having bootstrap equal to True, number of estimators equal to 100 and max features = to log2.

Recommendation

Another key feature in using Random Forests is its ability to easily determine which explanatory variable is most important in predicting the target variable. Using the feature_importance class, we were able to determine that “lstat” with a percentage of 26.4% was the variable with the most insight into predicting housing prices.

Appendix A

```
Method              Root mean-squared error training set
Linear_Regression        0.504557
Ridge_Regression         0.504573
Lasso_Regression         0.570767
ElasticNet_Regression    0.550480
Random_Forest            0.145293
dtype: float64

Method                Root mean-squared error test set
Linear_Regression        0.561940
Ridge_Regression         0.560511
Lasso_Regression         0.587381
ElasticNet_Regression    0.568084
Random_Forest            0.486018
```

Figure 1 – RSME scores for Training and Test Set

max_features	RMSE
1	0.589828
6	0.490656
12	0.486018
log2	0.500543

Figure 2 – RSME Scores for Random Forest with varying max_features

max_features	RMSE
1	0.532731
6	0.428159
12	0.454378
log2	0.444

Figure 3 – RSME Scores for Random Forest with varying max_features, Bootstrap = Ture and n_estimators = 100