MLND: Problem Set #1

Due on December 22, 2015

Jonathan Li

Problem 1

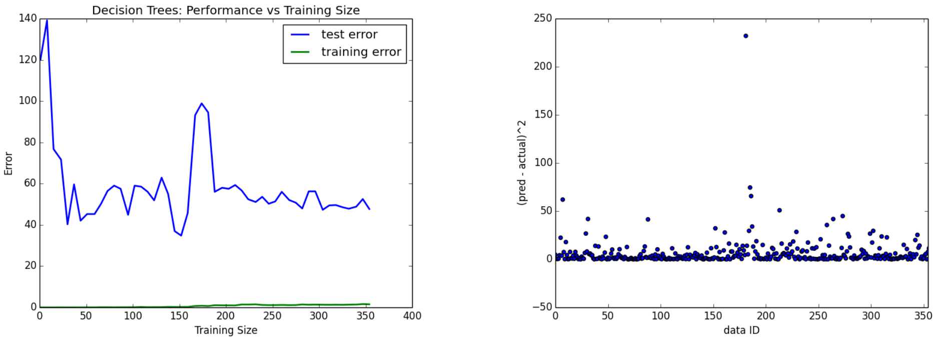
Statistics describing the housing prices and features are shown below.

Size of data	506
Number of features	13
Minimum price	5.0
Maximum price	50.0
Mean price	22.5
Median price	21.2
Standard dev.	9.2

Problem 2

- *Which measure of model performance is best to use for predicting Boston housing data and analyzing the errors? Why do you think this measurement most appropriate? Why might the other measurements not be appropriate here?*

  I chose the mean absolute error as my model performance metric. MAE is not as sensitive to outliers as a mean squared error metric. When using MSE, there appeared to be a peak in error when using a training size of 175 and a depth of 7 seen in Fig. 3(a). I suspected that this may have been the result of an outlier in the training set. Indeed, when I plotted squared errors between actual and predicted prices generated by the best-fit parameters, an outlier appears at 175 in Fig. 3(b). Thus, to account for outliers, I chose to use a MSE metric.



(a)

(b)

Figure 1: (a) A peak in test error is observed at a training size of 175. (b) An outlier is found at roughly the same location in the data.

- *Why is it important to split the Boston housing data into training and testing data? What happens if you do not do this?*

  The training data is clearly necessary to build a model. However, if we used all the Boston housing data to train the model, then we would not have any data left over to test how well the model performs. Furthermore, using some of the training data as testing data is not an option either, as this would not give us any indication about the predictive power of the model.

- *What does grid search do and why might you want to use it?*

The fit of a model can depend greatly on its input parameters. For instance, Fig. 2 shows that the depth of a decision tree can greatly affect training and test errors. The grid search allows us to test a space of one or more parameters to find the best fit model.

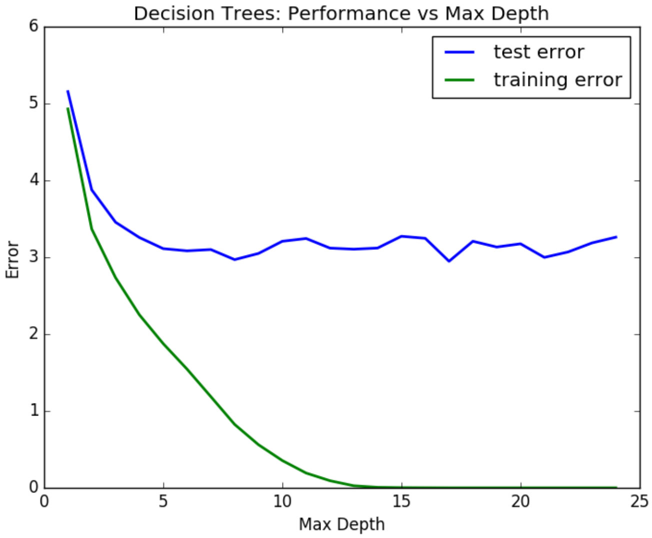


Figure 2: Training and test errors decrease with increasing depth, but hit a plateau at around 5.

- *Why is cross validation useful and why might we use it with grid search?*

  Cross validation is useful because it allows us to use all the data available for both training and testing. This allows us to maximize accuracy, since we do not have to worry as much about biased sampling of the data set. We would especially want to use it for grid search because we are testing different model parameters against each other, and need a more accurate score to determine their performances

Problem 3

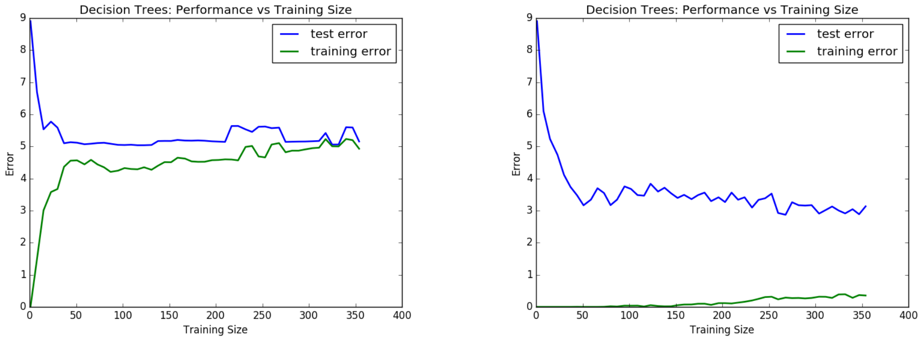
- *Look at all learning curve graphs provided. What is the general trend of training and testing error as training size increases?*

  The general trend is that training error increases gradually as training size increases, which is to be expected. However, the testing error initially decreases sharply, and plateaus quite quickly. This suggests that a good model could be created from a small sample size, and adding more data does not necessarily refine the model any further.

- *Look at the learning curves for the decision tree regressor with max depth 1 and 10 (first and last learning curve graphs). When the model is fully trained does it suffer from either high bias/underfitting or high variance/overfitting?*

  It appears that the models suffer from underfitting with the max depth is too small, as the error remains high. A model that is too simplistic may not be able to capture the important features of the data, and thus would not be able to accurately predict the test data. When we use a max depth of 10 (Figure 3(b)), we observe classic overfitting. In particular, the training error is much less than the test error, so the model does not generalize well to test data.

Problem 3 continued on next page. . .



(a)

(b)

Figure 3: Errors for test (blue) and training (green) errors for varying sample sizes for a maximum depth of (a) 1 and (b) 10.

- *Look at the model complexity graph. How do the training and test error relate to increasing model complexity? Based on this relationship, which model (max depth) best generalizes the dataset and why?*

  Fig. 2 shows that the test error reaches a stable minimum at a max depth of around 5. This suggests that the model generalizes well at a max depth of 5, since higher depths do not seem do decrease error. We note that training error also decreases with max depth, but at a slower rate. As expected, training error will eventually reach 0, as each leaf will eventually contain a singlet data point. However, we would prefer smaller trees due to space and time considerations.

Problem 4

- *Model makes predicted housing price with detailed model parameters (max depth) reported using grid search. Note due to the small randomization of the code it is recommended to run the program several times to identify the most common/reasonable price/model complexity.*

  The best fit decision tree model predicts an average price of 20.3 and a depth of 6.

- *Compare prediction to earlier statistics and make a case if you think it is a valid model.*

  The sample to be predicted is very similar to data point 470 which has the features

  `[12.05, 0.0,18.10,0,0.61,5.65, 87.6, 1.95,24,666,20.2, 291.55,14.10,20.8]`

  and has a price of 20.8. Furthermore, an analysis of the 10 nearest neighbors show the average price is 21.52, which is close to the prediction. In addition, the prediction is well within the range of one standard deviation from the mean price, and is also close to the median.