Machine Learning Nano Degree Project 1

Kai Xiong

December 2015

1 Introduction

In this report, we analyse the boston house price dataset. We use 13 features to fit a decision tree model to predict a house price. We first calculate the statistics of the dataset, then we evaluate the model performance by varying the performance metric. After that, we fine tune the model and use it to make a prediction about a house's price.

2 Statistical Analysis and Data Exploration

The general statistics of the Boston house price dataset is summarized as follows:

Statistics	Value
Number of data points (houses)	506
Number of features	13
Minimum housing prices	5
Maximum housing prices	50
Mean price	22.53
Median price	21.2
Standard deviation	9.19

Table 1: General statistics

We observe that the standard deviation is relatively high, which is about half of the mean.

3 Evaluating Model Performance

As we point out in previous section, the standard deviation is fairly high, indicating that the dataset is noisy. So the Median Absolute Error function is naturally the best measure because it has the best performance in terms of filtering out outliers. Other measures might fail to work because it considers all cases, which may be biased because of the outliers.

We split the dataset into a training set and a testing set. And we did multiple iterations with different training set and testing set size in each iteration. The training set is to help fine tune the model parameter and the test set is used to asses the predictive power of the tuned model. If we don't split the dataset, we may risk overfitting the dataset, getting very small in-sample error while very large out-sample error.

However, there is one more degree of freedom in the model, which is the depth of the decision tree. We need the depth that gives the best result. Thus, we use grid search to find the optimal depth of the decision tree that gives the lowest testing error. This is a cross validation process, where the training and test set must cross-over in successive rounds so that each data point has a chance of being validated against. It helps to reduce the chance of overfitting. If we just split the full dataset into training set and test set once, we might risk overfitting the model if the training and test set are not balanced. In addition, cross validation enables the grid search to take into account the entire dataset. This prevents any random anomalies due to random dataset splitting. Also, when data size is small or the number of parameters to estimate is large, cross validation allows an extensive use of the limited data point that helps better realize the potential of our model.

Looking at Figure 1 and Figure 2, regardless of the max depth, the test error is very big and training error is small when we have a small training size. This is because we are probably underfitting the model. So the model would have little predictive power. The test error reduces and training error increases as training size until it reaches an optimal size that the model is fully trained. After that, the model is overfitted. The test error does not decrease anymore with increasing training size.

When we use a decision tree model with max depth equal to 1, both the training error and testing error are higher than the model with max depth equal to 10. This is because the model is underfitting. In this case, the model is not well tuned to recognize the data patterns so that the in sample fitting and prediction are poor. The model is highly biased. With a max depth equal to 10, the model is overfitted so that the training error is very small and the test error is still large. In this case, the model has poor performance for out of sample data and has high variance.

Figure 1: With max depth = 1

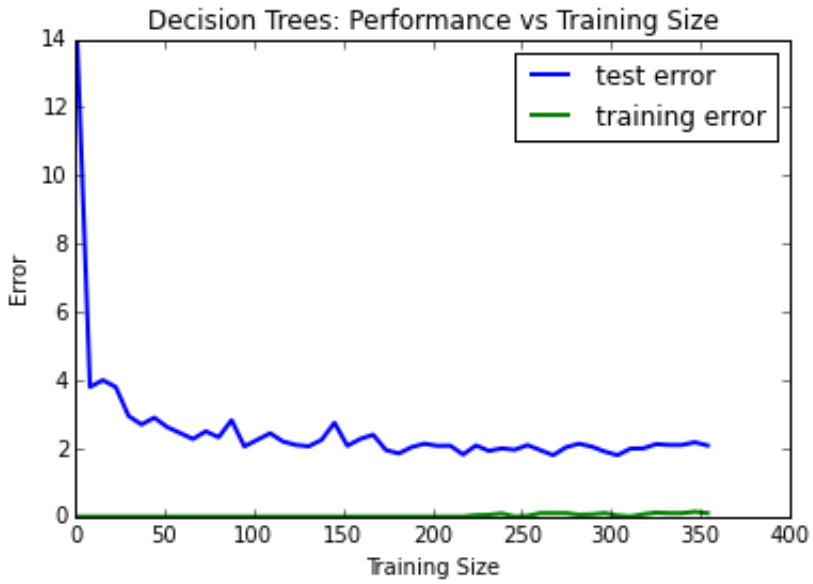

Figure 2: With max depth = 10

4 Analyzing Model Performance

We fit the model by varying the training size. The training error will increase with the training size while the testing error decrease. This is because we have a better fitted model using a larger training size if the training size is within reasonable range.
The complexity graph is presented below:

Figure 3: The complexity graph

We observe that when the max depth is 11, the training error is 0 while the testing error remains at the level

when the depth is 5 or higher. When the model is trained to have high complexity, we have very low training error while high testing error, thus the model suffers from high variance/overfitting. Both the training and test error have a decreasing trend with the model complexity. Based on the grid search, the max depth, 7, gives the lowest test error.

5 Model Prediction

We have a feature vector of size three. They are

Name	Description	Mean
CRIM	per capita crime rate by town	3.59
ZN	proportion of residential land zoned for lots over 25,000 sq.ft.	11.36
Indus	proportion of non-retail business acres per town	11.36
CHAS	Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)	0.0691
NOX	nitric oxides concentration (parts per 10 million)	0.554
RM	average number of rooms per dwelling	6.28
AGE	proportion of owner-occupied units built prior to 1940	68.5
DIS	weighted distances to five Boston employment centres	3.79
RAD	index of accessibility to radial highways	9.54
TAX	full-value property-tax rate per $10,000	408.2
PTRATIO	pupil-teacher ratio by town	18.45
B	$1000(Bk - 0.63)^2$ where Bk is the proportion of blacks by town	356.6
LSTAT	% lower status of the population	12.65
MEDV (Target)	Median value of owner-occupied homes in $1000's	

Table 2: Attributes

With the optimal max depth 7 and the feature vector [11.95, 0.0, 18.1, 0, 0.659, 5.609, 90.0, 1.385, 24, 680.0, 20.2, 332.09, 12.13], the model predicts the price to be 21.0157. This prediction is less than both the mean and the median, because some of its attributes are worse than the mean. For example, this area has more crime it has less space, heavier nitric oxides concentration, it has less access to the highways, thus the price of this house should be cheaper.

6 Conclusion

In this report, we study basic statistics of the boston housing dataset. We use grid search and cross validation to tune the decision tree model. And finally, we use the model to predict a house's price, which makes sense according to our analysis on its attributes.