CS385 Machine Learning (Summer 2018)
Assignment 3

Goh Kee Chin

16/6/2018

Abstract

In statistics, linear regression is a linear approach for modeling the relationship between a scalar dependent variable y and one or more explanatory variables (or independent variables) denoted X. For more than one explanatory variable, the process is called multiple linear regressions. In this report, gradient descent algorithm is applied to predict prices given various house-pricing predictors obtained from a Boston Housing Dataset collected in 1993. This report displays the relationship between various predictors and the median value of owner-occupied homes, usage of error functions, determination of a suitable alpha learning rate and observation using various factors for k-fold cross validation.

1 Linear Regression Equation

In this project, various equations are used to fulfill the requirements. Firstly, considering that the dataset contains 13 predictors attributes and a outcome attribute (MEDV), a polynomial function can be constructed as such:

$$\hat{y} = b_0 x_0 + b_1 x_1 + ... + b_{13} x_{13}$$

Where b_i represents theta coefficients that we are trying to optimize to provide a predicted $\hat{y}$ nearest to the actual MEDV and x_i represents the predictor values. This allows for a set of polynomial coefficient (theta) values to be selected to attempt to find a best fit. Obtaining of various 'weights' via coefficient adjustment can be done via Gradient Descent. Essentially we obtain a predicted price by applying the estimated "weights" b_n to the respective predictors x_n.

2 Feature-Price Plot

There are 13 predictors or input variables in the Boston dataset. Amongst those are a few critical ones (per capita crime rate by town, average number of rooms per dwelling, proportion of owner-occupied units built prior to 1940, weighted distances to five Boston employment centers, percent lower status of the population) that are highlighted in the plot below.

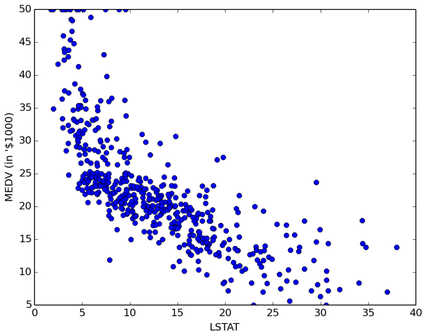


Figure 1: Plot of Percentage Lower Status to MEDV

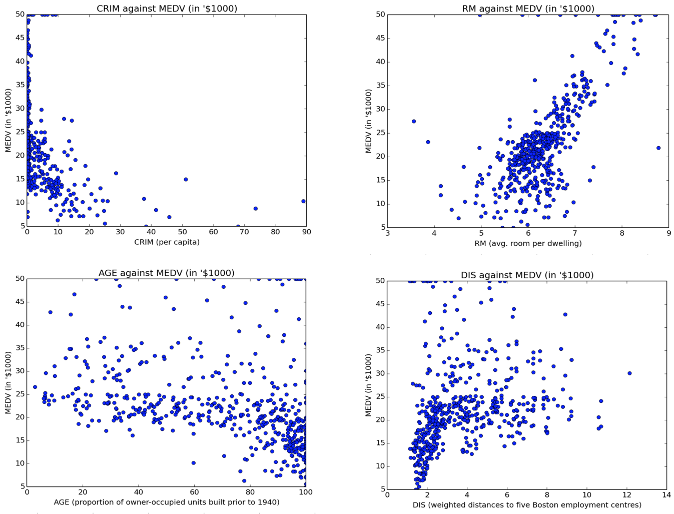


Figure 2: Four other predictors against MEDV

3 Error Function

The error function used to measure the disparity between predicted and actual housing prices is described as such:

$$Error_{(\theta)} = \frac{2}{N}\sum_{i=1}^{n}(y_i - (\theta_0 x_0 + \theta_1 x_1 + ... + \theta_{13} x_{13}))^2$$

The above function represents a mean-squared error function to measure how close the predicted values are to their corresponding real values. Essentially, the equation finds the difference between the predicted value and the actual output representative of the input values. This value is subsequently squared in order to ignore the sign to obtain a positive value and divided by the number of results in order to obtain the mean value of the error. In this case, N = 506 (size of the dataset)

4 Learning Rate

Within the gradient descent equation or algorithm, lies a coefficient, α, that determines the learning rate. Learning rate is a hyper-parameter that controls how much we are adjusting the weights of our network with respect to the loss gradient. If the α is too small, gradient descent can be slow. On the other hand, if α is too high, gradient descent can overshoot the minimum.

To facilitate the discovery of an optimal α value, considering that the dataset is relatively small, a brute force approach is taken. The following documents the pseudo-code for the aforementioned method:

```
SUBROUTINE FindOptimalAlpha
    SET alpha = 0
    SET optimal_alpha = 0
    SET lowest_avg_rmse = INFINITY

    FOR alpha <= ALPHA_MAX
        SET avg_rmse = RunTest(alpha)
        IF avg_rmse < lowest_avg_rmse
            lowest_avg_rmse = avg_rmse
            optimal_alpha = alpha
        alpha = alpha + ALPHA_INCREMENT

    RETURN optimal_alpha
```

In this experiment, we have selected an increment value of 0.001 and an *alpha_max* value of 0.5 (error goes up beyond this point) and plot it on a graph. We obtained a value of 0.417 (lowest AVGRMSE)

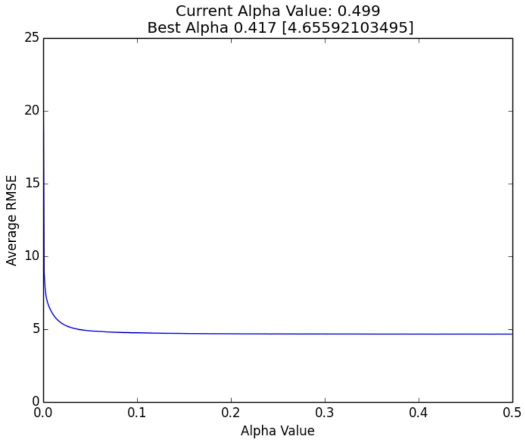


Figure 3: Alpha Value of 0.417 Suggested

5 Experimental Result

Using various k-folds parameters, we yield similar results. As we plot the predicted result against the expected results, we can see that the values converge close to a linear line of $y = x$ as such:

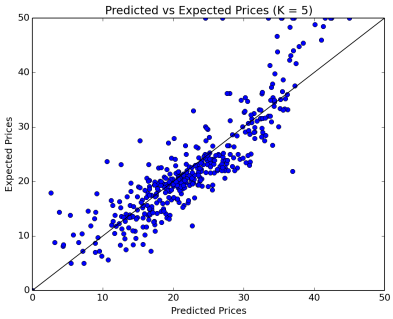


Figure 4: Predicted against Expected when K = 5

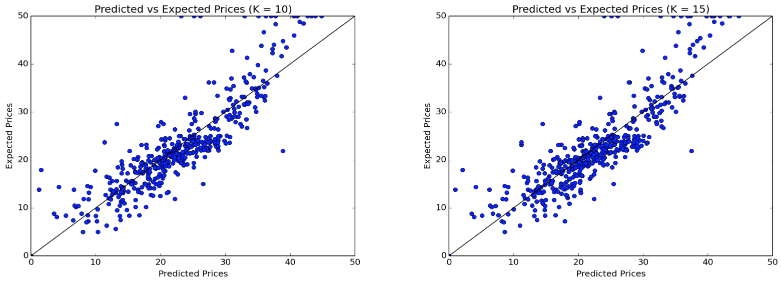
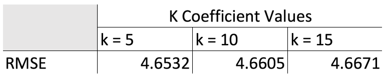


Figure 5: Predicted against Expected when K = 10, K = 15

The central line $y = x$ acts as a guiding line to show how close the predicted values are to the expected values. Any values that lies above the line shows an undervaluation of the actual price and vice-versa for any points below. We can see that the points are nucleated around the $y = x$ lines showing that the predicted prices are somewhat close to the actual values. However, visually comparing prediction accuracy is inaccurate and misleading, thus we are using RMSE to quantify the actual error.

5.1 Root Mean Squared Error (RMSE)

RMSE is a frequently used measure of the differences between values (sample or population values) predicted by a model or an estimator and the values observed. The RMSE represents the sample standard deviation of the differences between predicted values and observed values. The square root and dot essentially removes the sign (+/-) of the data and allows for more standardized manner of assessing data deviation.



	K Coefficient Values		
	k = 5	k = 10	k = 15
RMSE	4.6532	4.6605	4.6671

Figure 6: Average RMSE of Various K Values [alpha = 0.417]

In our experiment, we obtained the aforementioned RMSE and it is clear that the average RMSE is stabilized around the 4.6xxx region. Since an RMSE of 0 means that the data is completely accurate, a 4.6 RMSE in our case shows that our predicted data, on average, is off by 4.6 thousand of the MEDV value to the actual data.