Logistic Regression

LBYCP29 – Laboratory 5

Gervin Guevarra, Allen Koizumi, Nicholas Moreno, Charleston Uy

Department of Electronics and Communications Engineering

Gokongwei College of Engineering, De La Salle University

Manila, Philippines

*Abstract*—This laboratory report presents the implementation of an optimized linear regression using the minFunc package. As opposed to the previous linear regression implemented, wherein a line represents the data (as well as the resulting predictions), the optimized linear regression plots multiple points that follows the trend of the original data, thus the prediction is not limited to the values true for the linear equation and the accuracy is improved.

Keywords—Linear regression, optimization

# Introduction

In the Lab1 exercise, the linear regression is implemented to develop a line that will represent the trend of the original data. This line will be used to predict values, with respect to a certain input. The Ɵ0 and Ɵ1 , the values representing the y-intercept and slope of the line, respectively, are chosen while minimizing the cost function J(Ɵ). The result is a line that is more or less following the trend of the scatter data.

However, the cost function J(Ɵ) does not always guarantee that the hypothesis, as well as the resulting predictions, is accurate. As we can remember from the previous laboratory exercise, a line can be fitted in many ways – underfit, “just right”, overfit. Underfit and overfit may still occur even when the graph of cost function is generally acceptable (that is the J(Ɵ) approaches 0 as the number of iterations increases). With that being said, J(Ɵ) cannot guarantee the accuracy of the prediction.

# Objectives

The experiment aims to achieve the following objectives

* To reestablish the steps in using linear regression
* To apply the minFunc package in order to minimize the cost function

# Data and Results

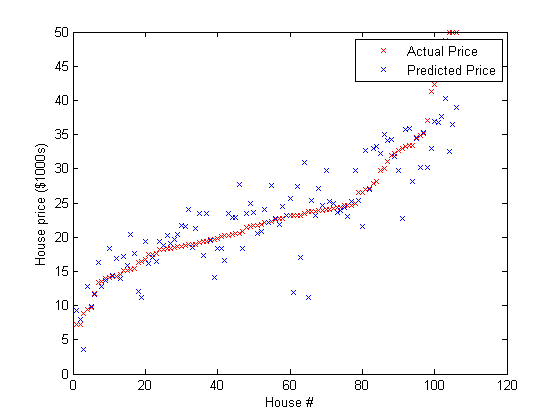


Figure 1. Predicted price using optimized linear regression plotted against the actual price.

# Analysis and Conclusion

the linear regression is optimized using the minFunc package. Instead of only relying on the J(Ɵ), we also take a look at the deviation and attempts to minimize it while also minimizing J(Ɵ). In this way, predicted values will be more closer to the actual values.

# Bibliography

|  |  |
| --- | --- |
| [1] | A. Ng, "Spervised Learning," n.d. [Online].  Available: http://cs229.stanford.edu/notes/cs229-notes1.pdf [Accessed 22 September 2015]. |
| [2] | A. Ng, "Machine Learning,"  n.d. [Online]. Available: https://class.coursera.org/ml-003/lecture.  [Accessed 22 September 2015]. |
| [3] | Institute for Digital Research and Edcation, "Stata Data Analysis and  Exampes", 2015 [Online]. Available: http://www.ats.ucla.edu/stat/stata/dae/logit.htm.  [Accessed 22 September 2015]. |

# Appendix

Logistic Regression

% Run using ex1a\_linreg.m

function [f,g] = linear\_regression(theta, X,y)

%

% Arguments:

% theta - A vector containing the parameter values to optimize.

% X - The examples stored in a matrix.

% X(i,j) is the i'th coordinate of the j'th example.

% y - The target value for each example. y(j) is the target for example j.

%

m=size(X,2);

n=size(X,1);

f=0;

g=zeros(size(theta));

%

% TODO: Compute the linear regression objective by looping over the examples in X.

% Store the objective function value in 'f'.

%

% TODO: Compute the gradient of the objective with respect to theta by looping over

% the examples in X and adding up the gradient for each example. Store the

% computed gradient in 'g'.

%%% YOUR CODE HERE %%%

f= .5 \* (y-theta' \* X)\*(y-theta' \* X)';

g= X \* X' \*theta-X \* y';

end