Logistic Regression

LBYCP29 – Laboratory 3

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 logistic regression in a classification problem. This type of regression is useful in data sets that are binary in nature (where y can only have two values, 0 or 1).

Keywords—Logistic Regression, binary classification

# Introduction

Logistic regression is similar to other types of regression, only that the values or data to be predicted consists of binary values only.

For this experiment, the linear regression model can still be used. However, doing this proves to be a poor solution to the problem. This is because values does not exceed values larger than 1 and smaller than 0, given that data set *y* is only binary (1 and 0).

(1)

Where

(2)

is called the logistic function or sigmoid function. The goal of the logistic regression, as for any type of regression model, is to minimize the cost function The cost function is defined as

(3)

By plotting the cost function, the minimum number of iterations can be deduced from the plot. This number is proven to be the minimum number of iterations, when the graph reaches a stagnant state; or when the graph does not change anymore, remaining constant.

# Objectives

The experiment aims to achieve the following objectives

* To implement logistic regression using Newton’s Method.
* To build a binary classification model that would estimate the results given the data sets.

# Data and Results

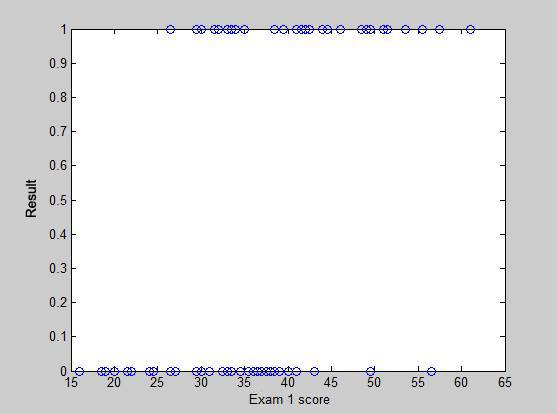


Figure 1. Exam 1 scores vs the results.

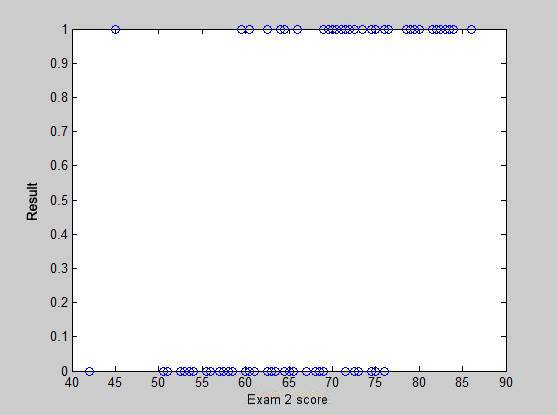


Figure 2. Exam 2 scores vs the results

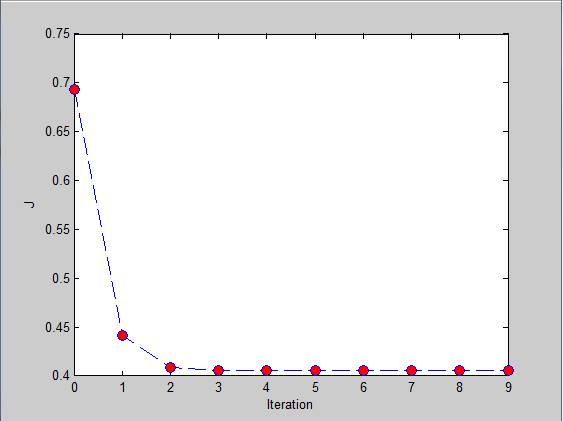


Figure 3. Cost function

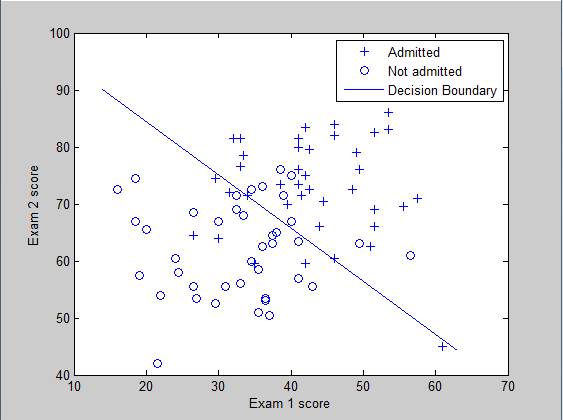


Figure 4. Decision boundary for the exam scores.

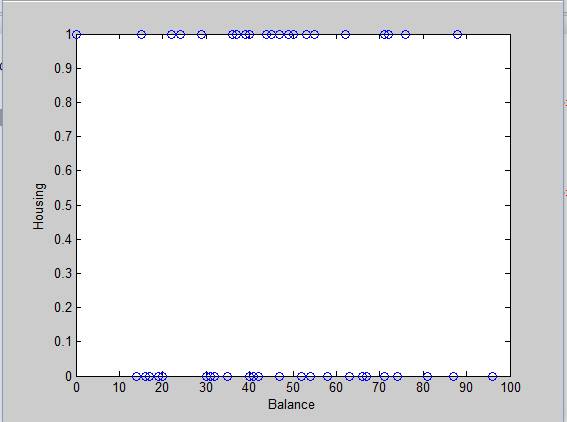


Figure 5. Housing vs Balance

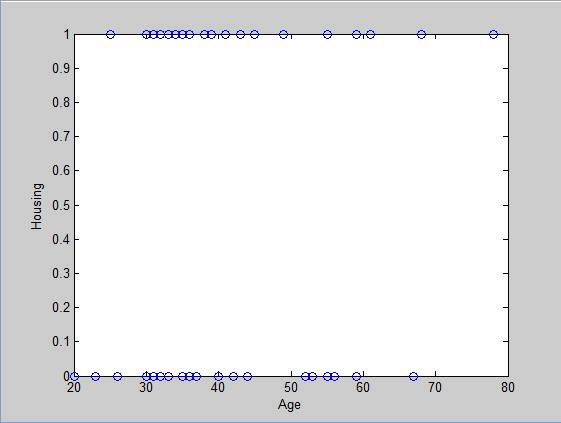


Figure 6. Housing vs Age

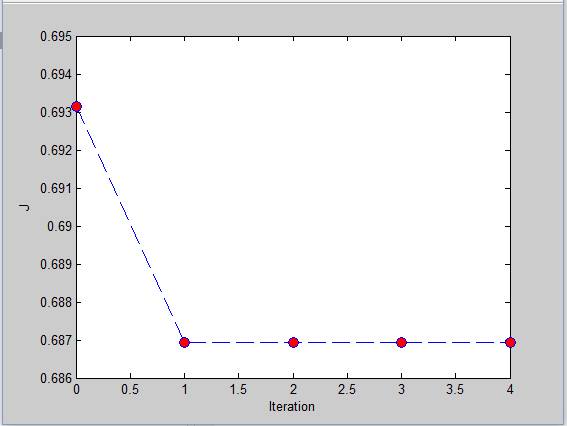


Figure 7. Cost function (for the custom data)

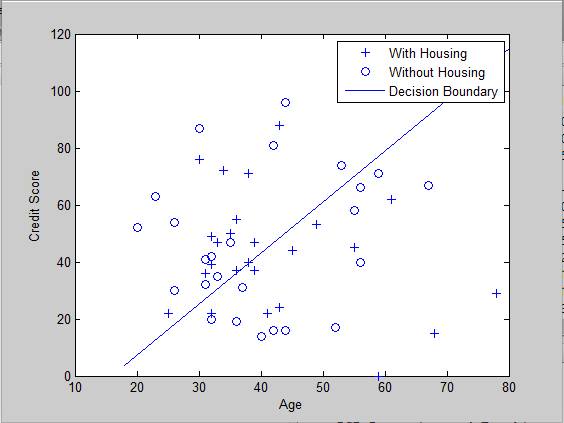


Figure 8. Decision boundary for the custom data.

# Analysis and Conclusion

As observed in figures 1, 2, 5 and 6, we can see that the values in the y axis produces values that are binary in nature therefore a logistic regression should be used. By using logistic regression, we can determine the predicted outcome of values that uses data that is binary in nature. Compared to linear regression, using logistic regression would yield a better solution rather than a poor solution by using the Newton’s Method. Newton’s Method is similar to Gradient Descent, it is a way to search for the minimum of the derivative of the cost function. From figure 3 and 7 shown above, we can see that the Newton's Method has converged by around 7 iterations and 4 iterations respectively. Compared to using Gradient Descent, Using Newton’s Method is faster.

# 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

function [theta]=Logistic\_Regression(x,y)

m= length(y);

theta = zeros(3, 1);

g = inline('1.0 ./ (1.0 + exp(-z))');

J = zeros(5, 1);

for i = 1:5

z = x \* theta;

h = g(z);

grad = (1/m).\*x' \* (h-y);

H = (1/m).\*x' \* diag(h) \* diag(1-h) \* x;

J(i) =(1/m)\*sum(-y.\*log(h) - (1-y).\*log(1-h));

theta = theta - H\grad;

end

plot\_x = [min(x(:,2))-2, max(x(:,2))+2];

plot\_y = (-1./theta(3)).\*(theta(2).\*plot\_x +theta(1));

plot(plot\_x, plot\_y)

hold on

plot(x(pos,2),x(pos,3),'+');

plot(x(neg,2),x(neg,3),'o');

legend('Admitted', 'Not admitted', 'Decision Boundary')

hold on

figure

plot(0:MAX\_ITR-1, J, 'o--', 'MarkerFaceColor', 'r', 'MarkerSize', 8)

xlabel('Iteration'); ylabel('J')

end