LAB ACTIVITY 3: LOGISTIC REGRESSION AND NEWTON’S METHOD

DANTING, Jules Raphel

DE VERA, Luis Paolo

LIM, Alyssa Raphaella

RAMOS, Brian

VALDEZ, Esteen Rae

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**1 INTRODUCTION**

The topic of experiment 3 is about using Logistic Regression and Newton’s Method in machine learning. The task is to apply the functions of newton’s method in logistic regression. To create a binary classification model that can will check the chance of college admission based on the students’ scores in two exams. Logistic regression is the process in determining the measurement of a variable of being one or zero according to the binary model [1]. There will be a nominal variable which is the dependent variable that will have two values for this experiment it will be admitted and not admitted. The measurement variable which is the independent variable will be exam 1 and exam 2. It will determine the probability based on the measurement value on whether the student is admitted or not admitted. Newton’s method is used in approximating the root of polynomial equations [2]. This will minimize the error within the cost function thus having a reliable value on determining the results of the student. In this experiment the group will have an experience in dealing with logistic regression to determine the binary result of any given nominal variable which is based on its measurement variables and to understand the codes used in applying the newton’s method in the cost function J (ϴ).

**2 PROCEDURE**

**Procedure 3.1**

● Plot the data.

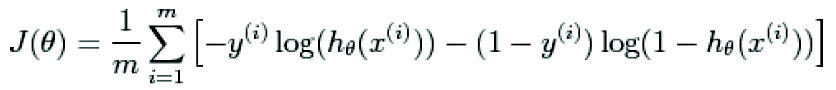
● Load the data for the training examples into your program and add the x\_0 = 1 intercept term into your x matrix.

● Before beginning Newton's Method, we will first plot the data using different symbols to represent the two classes. In Matlab/Octave, you can separate the positive class and the negative class using the find command.

**Procedure 3.2**

● Plot the Cost function J(θ).

● The cost function J(θ) is defined as:



**Procedure 3.3**

Answer the ff. questions:

1. θ = ?

2. How many iterations were required for convergence?

3. What is the probability that a student with a score of 20 on Exam 1 and a score of 80 on Exam 2 will not be admitted?

**3 RESULTS AND DISCUSSION**

**Procedure 3.1 Data Plot**

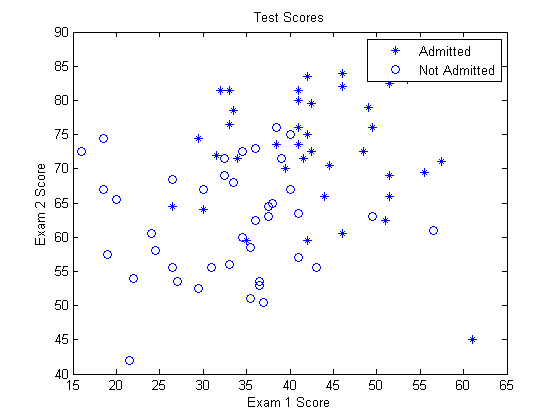


Fig. 1. Plot of Exam 1 and Exam 2 Scores

Figure 1 shows the training data of the high school students. There were two representation in the plot, one was students who were admitted and the other are the students who were not admitted. The x axis indicated the scores on the first exam and the y axis tells the scores on the second quiz. The figure tells that there are 40 students who were admitted and 40 students who were not admitted. Therefore, the total number of training data is 80 students.

**Procedure 3.2 Cost Function Plot**

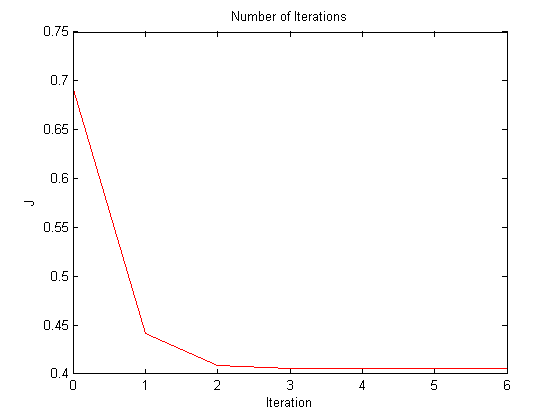


Fig. 2. Shows the plot of J vs. Iteration or the Cost

Function. In this graph, it shows that as the number of iterations increases, it becomes closer to being convergent but increasing it too much will make it divergent. The graph shows that on four iterations, the graph is becoming stable and as it reached the fifth iteration, it become more stable. It means that in the graph, five iterations is enough to make the plot convergent. Newton’s Method was used to plot and know how many iterations were needed to meet the convergence. The value of J was 0.4054 and 5 iterations were needed to be convergent.

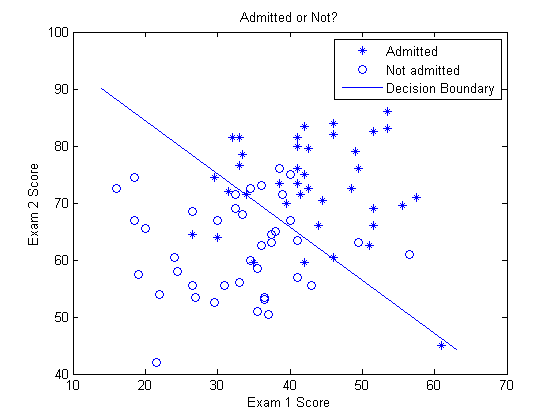


Fig. 3. Data Plot with Decision Boundary

This figure shows the decision boundary of the plot. As seen on the graph, most of the data above the line was admitted while most of the data below it was not admitted. It also tells that there are some students above the boundary line but was not admitted and some students below the boundary line were admitted.

**Procedure 3.3**

1. theta0 = -16.3787

theta1 = 0.1483

theta2 = 0.1589

2. J =

[1] 0.6931

[2] 0.4409

[3] 0.4089

[4] 0.4055

**[5] 0.4054**

[6] 0.4054

[7] 0.4054

Iterations required for convergence: 5

3. Probability of a student not being admitted with a score of

Exam 1 = 20

Exam 2 = 80

Probability: 0.6680

Number 1 tells the value of theta. The value of theta will be used to get the value of J.

Number 2 tells the number of iterations needed to meet the convergence. According to the data, 5 iterations will satisfy the convergence. 1 - 4 iterations will not meet the convergence since the value of J changes but as soon as it reached 5, the value of J didn’t change as the iteration was incremented. The value of J remained 0.4054 from 5 iterations to 7 iterations telling that five iterations are needed to make it converged.

Number 3 tells the probability of a student being admitted with a score of 20 and 80 on the first and second exam respectively. It show that the student has a 66.80% of being admitted.

**Procedure 3.4 Own Data**

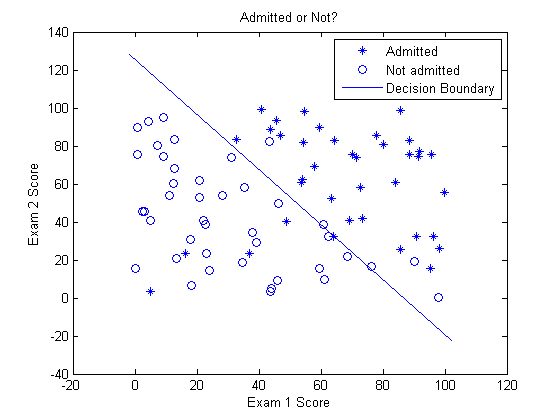


Fig. 4. Own Data Plot

Conclusion based data used: If the student’s average grade for exam 1 and 2 is greater than 55, he is more likely to be accepted. Else he is more likely to be rejected. Same procedure were made in this part, the only difference is the value of the training data. The plot is the same in Figure 3. Most of the student above the boundary line were admitted and some students below it were not admitted. But there are also some students who were not admitted but above the line and some students who were admitted but below the line. New set of training data was used here to be able to predict whether a student will be admitted or not.

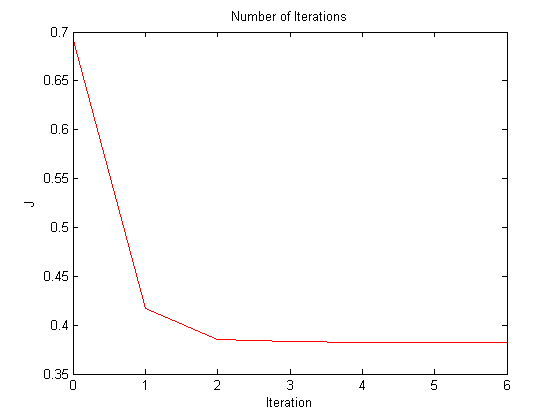


Fig. 5. Cost Function Plot of Own Data

Figure 5 shows the number of iterations needed for convergence. In this graph, the value of J is below 0.4 which is the value of J on figure 2. It tells that the value of J differs for every training data that is inputted. The number of iterations is 5 and it is the same on figure 2. It tells that for every training data, the number of iterations required will always be equal to 5. Below is the value of theta and the probability of a student being admitted with a score of 20 and 80 on the first and second examination respectively. The probability of a student being admitted is 68.23%. The result was almost near the probability result in the experiment. It means that even if the training data was changed, the probability of the student being admitted or not will be the same since the only thing that changed in the procedure was the data and not the method used.

theta =

-5.8034

0.0671

0.0462

Number of Iterations needed for convergence: 5

Probability of a student not being admitted with a score of

Exam 1 = 20 and Exam 2 = 80

Probability = 0.6823

**MATLAB Code**

%LAB03

clear all**;** close all**;** clc

x **=** load**(**'xNew.dat'**);**

y **=** load**(**'yNew.dat'**);**

theta **=** zeros**(**3**,**1**);**

m **=** length**(**x**);**

x **=** **[**ones**(**m**,** 1**),** x**];**

pos **=** find**(**y**==**1**);**

neg **=** find**(**y **==**0**);**

%Procedure 3.1

plot**(**x**(**pos**,** 2**),** x**(**pos**,** 3**),** **(**'\*'**));**

hold on

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

xlabel**(**'Exam 1 Score'**);**

ylabel**(**'Exam 2 Score'**);**

title**(**'Test Scores'**);**

legend**(**'Admitted'**,** 'Not Admitted'**);**

%z = theta\x;

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

%Procedure 3.2

iter **=** 7**;**

J **=** zeros**(**iter**,** 1**);**

**for** i **=** 1**:**iter

z = x \* theta;

h = g(z);

%Cost Function

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

%Newton's Law

%Logistic Regression

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

%Hessian

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

%Update Rule

theta = theta - H\grad;

end

figure;

plot(0:iter-1,J(1:iter),'r');

xlabel('Iteration');

ylabel('J');

title('Number of Iterations');

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

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

figure;

plot(x(pos, 2), x(pos, 3), ('\*'));

hold on

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

xlabel('Exam 1 Score');

ylabel('Exam 2 Score');

title('Admitted or Not?');

hold on;

plot(plot\_x, plot\_y)

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

hold off

% Plot J

figure

plot(0:iter-1, J, 'o', 'MarkerFaceColor', 'g', 'MarkerSize', 9);

xlabel('Iteration');

ylabel('J');

title('Number of Iteration Before Convergence');

%Procedure 3.3

%Theta

theta

Iterations = 5

J

probability\_of\_not\_passing = 1 - g([1, 20, 80]\*theta)

probability\_of\_passing = 1 - g([2, 20, 80]\*theta)

%not admitted = 1, admitted = 2

%exam 1 = 20, exam 2 = 80

**4 ANALYSIS & CONCLUSION**

In this experiment the group became familiar with the Logistic Regression. Using the 80 training data, the group produced a crossed function J(ϴ) which is the hypothesis. Using Newton’s method and the training data the group specifically the exam scores, it was predicted whether the student can be admitted or not. In the second part of the experiment, the group created new set of data, but performed same steps used in the first part to get hypothesis of the function. In both experiments, the system base the hypothesis in the average of exam 1 and exam 2. To sum things up using the two sets of training data, the group predicted whether the student will be admitted or not. By performing a number of iterations the system produced the J(ϴ).

**5 REFERENES**

[1] McDonald, J.H. (2014). Handbook of Biological Statistics (3rd ed.). [Online]. Availble: http://www.biostathandbook.com/simplelogistic.html.

[2] P. Garrett. Newton's Method. [Online]. Available: http://mathinsight.org/newtons\_method\_refresher