

De La Salle University

Gokongwei College of Engineering

Electronics and Communications Engineering Department

**LBYCP29 Experiment 1**

Submitted By:

**Sidney Carlo P. Lopez**

**Carl Michael O. Sy**

**Jeremy Christian B. Reccion**

**Wellington S. Villasin**

Submitted To:

**Engr. Melvin Cabatuan**

Date:

**September 9, 2015**

Linear Regression

Wellington S. Villasin

11209534

Carl Michael O. Sy

11208929

Sidney Carlo P. Lopez

11222522

Jeremy B. Reccion

11210192

# Introduction

Linear Regression is a methodology for demonstrating the relationship between a scalar dependent variable y and one or more logical variables (or autonomous variables) denoted X. The instance of one informative variable is called simple linear regression. With the use of the scatterplot, it would help demonstrate the strength of two of the given variables. This was also one of the first types of regression analysis where it was greatly used in the field of practical analysis [1].

# Procedures

## Implement gradient descent using a learning rate of α = 0.07. Since Matlab/Octave and Octave index vectors starting from 1 rather than 0, you’ll probably use theta(1) and theta(2) in Matlab/Octave to represent θ0 and θ1. Initialize the parameters to θ= 0 (i.e., θ0 = θ1 =0), and run one iteration of gradient descent from this initial starting point. Record the value of θ0 and θ1 that you get after this first iteration. (To verify that your implementation is correct, later we’ll ask you to check your values of θ0 and θ1 against ours.)

## Continue running gradient descent for more iterations until θconverges. (this will take a total of about 1500 iterations). After convergence, record the final values of θ0 ­and θ1 that you get. When you have found θ, plot the straight line fit from your algorithm on the same graph as your training data. The plotting commands will look something like this:

## 

## Note that for most machine learning problems, χ is very high dimensional, so we do not be able to plot h θ(χ). But since in this example we have only one feature, being able to plot this gives a nice sanity-check on our results.

## Finally, we’d like to make some predictions using the learned hypothesis. Use your model to predict the height for a two boys of age 3.5 and 7.

**Debugging:** If you are using Matlab/Octave and seeing many errors at runtime, try inspecting your matrix operations to check that you are multiplying and adding matrices in ways that their dimensions would allow. Remember that Matlab/Octave by default interprets an operation as a matrix operation. In cases where you do not intend to use the matrix definition of an operator but your expression is ambiguous to Matlab/Octave, you will have to use the ‘dot’ operator specify your command. Additionally, you can try printing x, y, and theta to make sure their dimensions are correct.

# Data and Results

## 1st Procedure

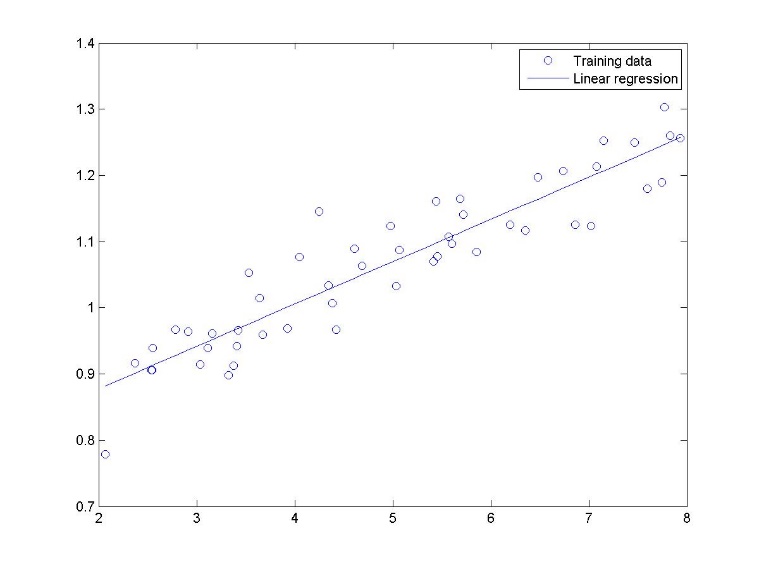
## First Iteration:

Θ1 = 0.0745

Θ2 = 0.3800

## 1500th Iteration:

Θ1 = 0.7502

Θ2 = 0.0639

## 2nd Procedure

## Plot:

## 3rd Procedure

Age: 3.5 Height: 0.9739

Age: 7 Height: 1.1973

## 3D Plot

## Problem Answers

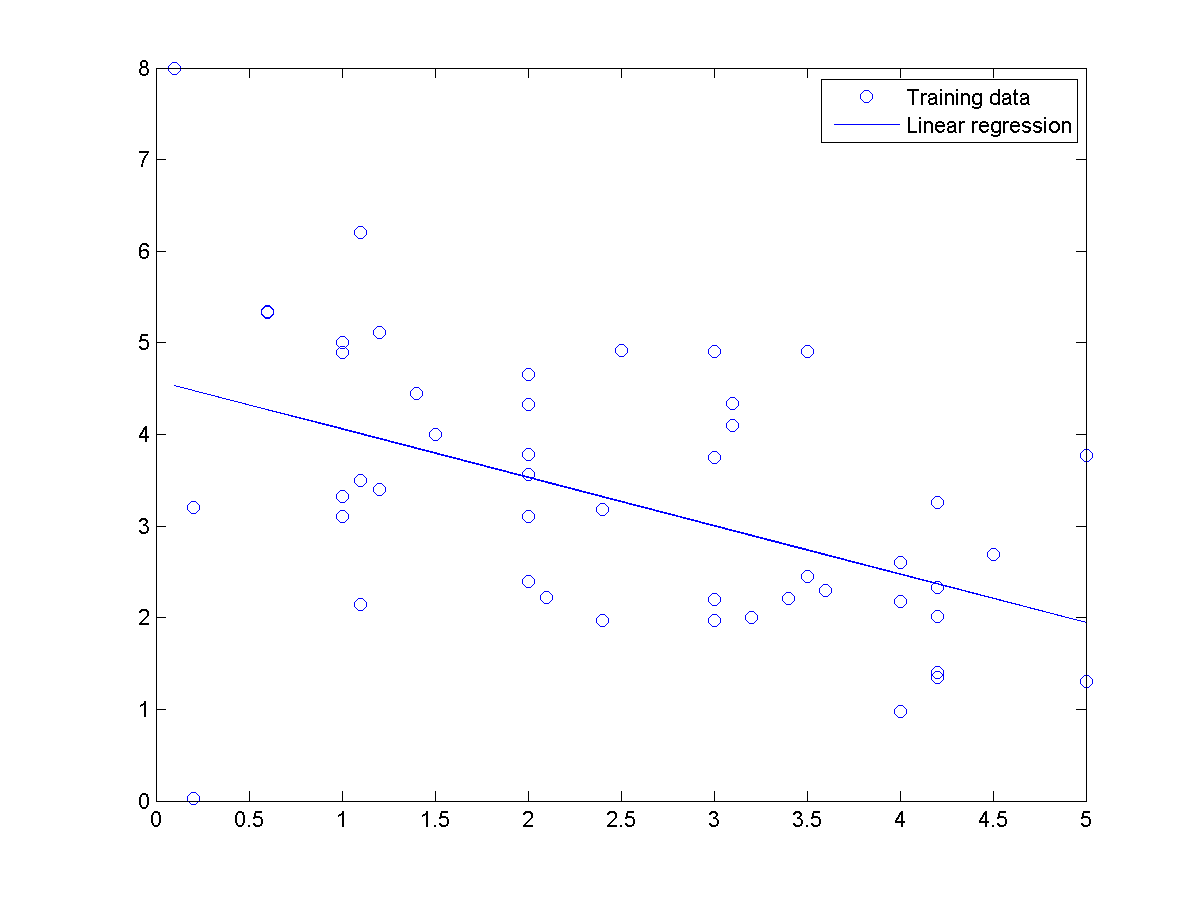
First Iteration:

Θ1 = 0.2299

Θ2 = 0.4987

1500th Iteration:

Θ1 = 4.5911

Θ2 = -0.5287

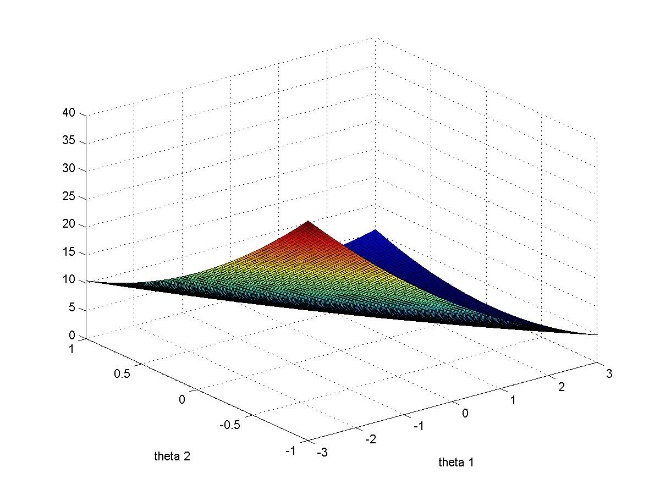
Plot:

## Time/Speed

Time: 2.5 Speed: 3.2694

Time: 4.15 Speed: 2.3971

Plot:



# CONCLUSION

In this activity, Linear Regression has proven to be a great exercise in using MATLAB/OCTAVE. The process in acquiring the linear regression utilizes the different functions of MATLAB/OCTAVE such as zeros, ones, load, and length. These functions were used to set the necessary elements in finding the gradient descent which is the solution in determining the most probable value in a set of input data. After learning the usage of these softwares to solve linear regression problems, it has expanded the capability of students in solving problems in relation to linear regression.

# Codes

x = load('ex2x.dat');

y = load('ex2y.dat');

plot(x,y,'o');

x = [ones(50,1), x];

%procedure 1

>> theta = zeros(2,1);

>> theta = theta - (0.07/50)\*x'\*(x\*theta -y)

theta =

0.0745

0.3800

%procedure 2

theta = zeros(2,1);

>> for i = 1:1500

theta = theta - (0.07/50)\*x'\*(x\*theta -y);

end

>> theta

theta =

0.7502

0.0639

hold on

plot(x(:,2), x\*theta, '-');

legend('Training data', 'Linear regression');

%procedure 3

z = 3.5\*theta(2) - theta(1);

zz = 7\*theta(2) - theta(1);

%3d plot

jvals = zeros(100, 100);

theta1 = linspace(-3,3,100);

theta2 = linspace(-1,1,100);

for i = 1:length(theta1)

for j = 1:length(theta2)

t = [theta1, theta2];

jvals(i,j) = (1/100) \* sum((x \*t' -y).^2);

end

end

jvals = jvals';

figure;

surf(theta1, theta2, jvals)

# Questions

1. What is the relationship between this 3D surface and the value of θ0 and θ1 that your implementation of gradient descent had found?

Both have similar results yet it showed that the there is a difference within the max values due to the difference of values in the data. Although the max values are different, it still portrayed more or less the same output in the 3D plot surface.

1. Provide your own data with at least 50 elements and apply the procedures performed in this experiment.

Answers stated in III. Data and Results

# REFERENCE

[1] Y. Personnel, "Yale Education," Yale University,[Online].Available: http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm. [Accessed 8 September 2015].