LAB ACTIVITY 2:
MULTIVARIATE LINEAR REGRESSION

CARANDO, Paolo Creas
GOMENDOZA, Frank
LIM, Theo Paolo
LORENA, Neil John
ZABAT, Juan Paolo

I.INTRODUCTION

In this machine learning lab activity, the group familiarize themselves with Multivariate Linear Regression. For data analysis regression methods are commonly used. Multivariate regression is used as a statistical tool and several response variables are studied simultaneously. The multivariate regression model must concern two things. First, practical limitations must be implemented because the accurate estimation of all the regression coefficients may require a relatively large number of observations. Second the interpretation of a large number of regression coefficients can become unwieldy (G. C. Reinsel, 1998). Multivariate linear regression is a feature for multivariate regression model. Multivariate linear regression has linear restrictions on the coefficient matrix (G.C. Reinsel, 1998). This experiment focuses in applying gradient descent into using multivariate regression. The methods and output of this experiment must reflect a reliable plot for analysing data. Like the previous experiment each iteration must show the most accurate and reliable estimation based from the data given in the experiment.

II.PROCEDURE

Load Data and Initialization

1. Load the training data and assign it to x & y variable. X will pertain to the living area and number of bedrooms while Y will set the prices. Assign also a variable for the length of x. For theta assign 3X1 zero values. Please see code 1.

```
x = load('ex2x.dat');
y = load('ex2y.dat');
m=length(x);
theta=[0 0 0];
```

Code 1: Loading of Data and Initialization

2. Plot the raw and preprocessed data. First plot the housing prices with respect of the living area of the raw data (a). Then, plot the housing prices with respect of the number of bedrooms (b). See Code 2.

```
a. plot(x(:,1),y,'o');
b. plot(x(:,2),y,'o');
```

Code 2: Plotting the Raw Data

3. Pre-process the data first before plotting. Pre-processing the data will make sure that the values are in the same magnitudes. Both x-axis of Figure 3 and 4 have the same magnitude; that will converge the data well. See Code 3. After plotting the raw data, plot the preprocessed data similar with the plotting of the raw data.

```
x = [ones(m, 1), x];
sigma = std(x);
mu = mean(x);
x(:,2) = (x(:,2) - mu(2))./ sigma(2);
x(:,3) = (x(:,3) - mu(3))./ sigma(3);
```

Code 3: Pre-processing

Gradient Descent

4. The formula for the gradient descent will be similar from the previous experiment. However, the matrix of x will contain another value.
5. First identify a suitable learning rate, this will be alpha. Trial and error will be used to identify a suitable learning rate for the data. Use the vectorized cost function (J) to see which plot has the fastest drop with respect to the number of iterations. Plot J with respect to number of iterations. See Code 4.

```
for num_iterations = 1:100
J(num_iterations) = (1 / 2 * m) .* (x *
theta - y)' * (x * theta - y);
h = (1 / m) .* x' * ((x * theta) - y);
theta = theta - alpha .* h;
endfor
plot(0:99, J(1:100), '-')
```

Code 4: Using Vectorized Cost Function to determine learning rate

6. After getting the good learning rate, find the values of theta. Predict the price of the house using the values of theta.

Normal Equations

7. In order to minimize the error between the plot vs the data, normal equations are used. Instead of getting the gradient descent, assign zero to the derivative set and find the theta, then find the price of the house. Compare the result from the previous result using gradient descent. See Code 5.

```
theta = inv (xx' * xx) * xx' * y;
price = theta(1) + (theta(2)*1650) +
(theta(3)*3);
```

Code 5. Using Normal Equations

III.RESULTS AND DISCUSSION

PROCEDURE 2.0

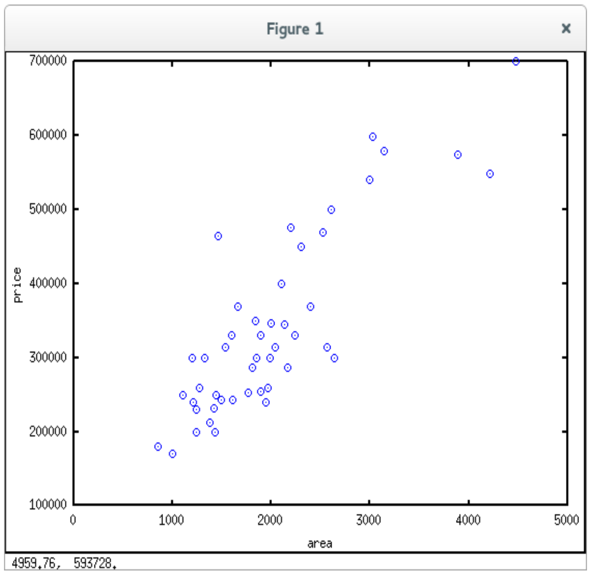


Fig. 1. Raw Data Plot of Housing Prices vs Living Area

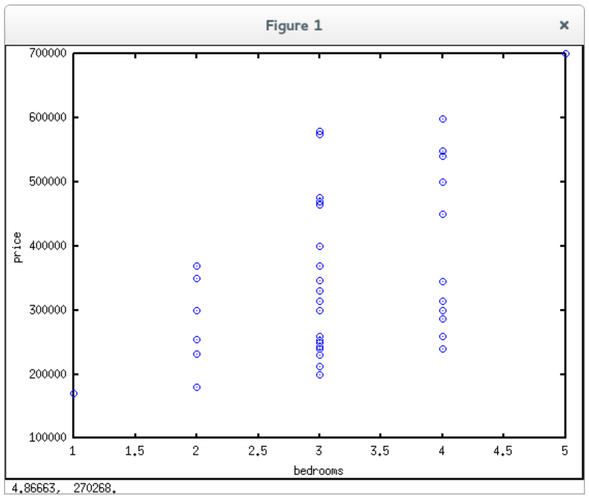


Fig. 2. Raw Data Plot of Housing Prices vs Number of Bedrooms

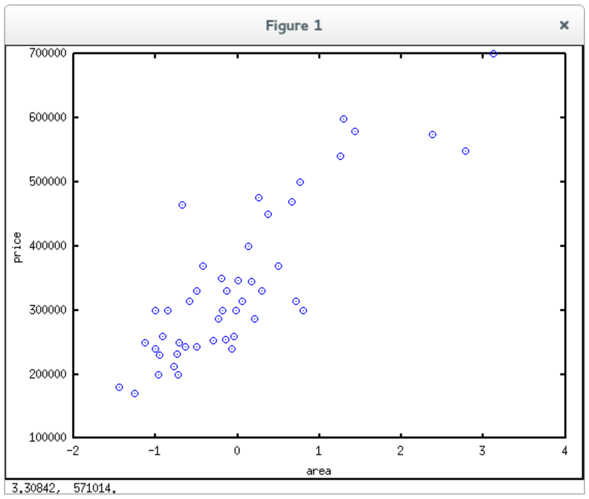


Fig. 3. Preprocessed Data Plot of Housing Prices vs Living Area

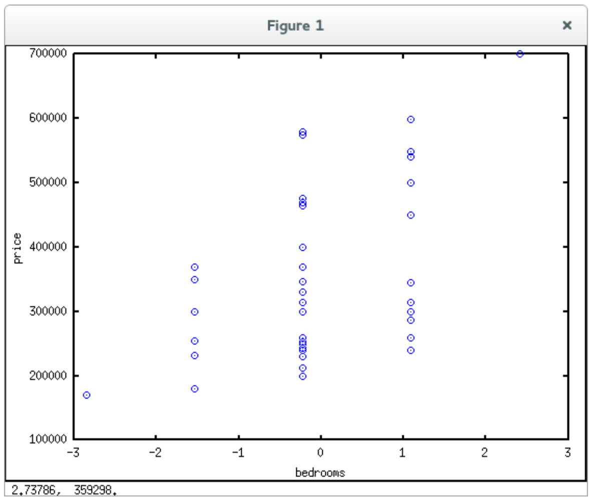


Fig. 4. Preprocessed Data Plot of Housing Prices vs Number of Bedrooms

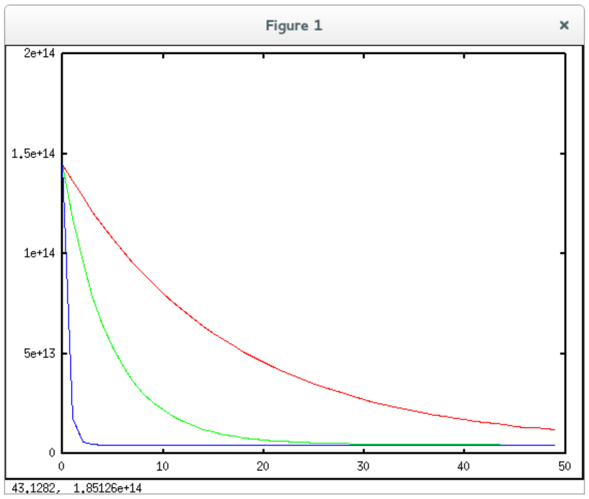


Fig. 5. Output 1: Number of Iterations vs Cost

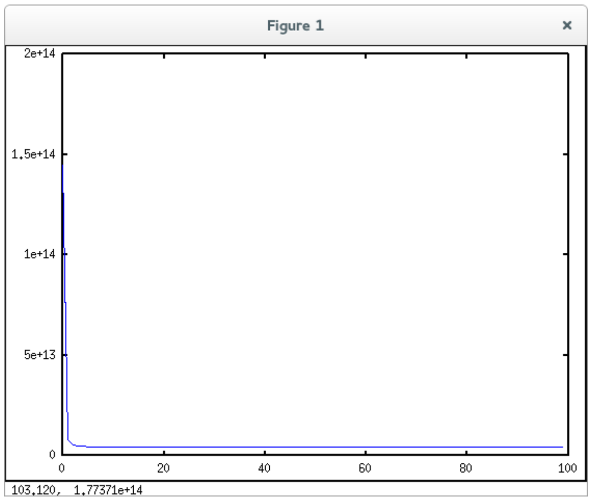
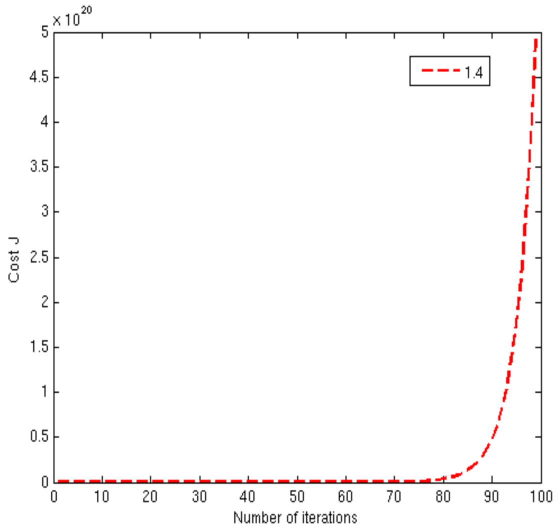


Fig. 6. Learning Rate Plot

PROCEDURE 2.1

What happens when the learning rate is too small? Too large?

If the learning rate of the system is too small, convergence will take place slowly during gradient descent. And if learning rate is too large, gradient descent might not converge at all or might not find values close to the optimum value and might result to increasing J(θ) after some iterations. Refer to Plot 1 to envision the behaviour when the learning rate is too large. The J(θ) overshoots the optimum value in the plot and diverge from it.



Plot 1. Plot with Too Large Learning Rate [2]

What is the best learning rate that you have found?

α = __1__

PROCEDURE 2.2

Using the best learning rate that the group found, run gradient descent convergence to find:

1. Final values of θ1 = ___89,598_____
   θ2 = ___139.21______
   θ3 = ___-8738_______
2. Predicted price of a house with 1650 square feet and 3 bedrooms
   House Price = __$ 293,080__

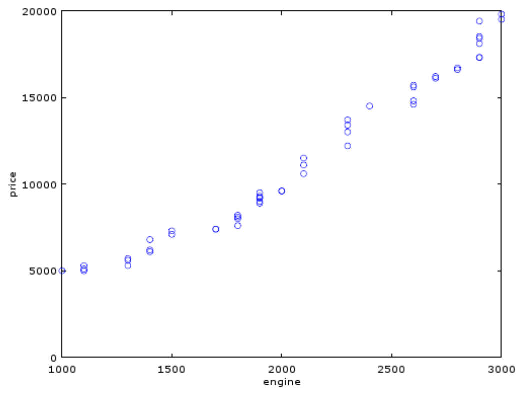
PROCEDURE 2.3

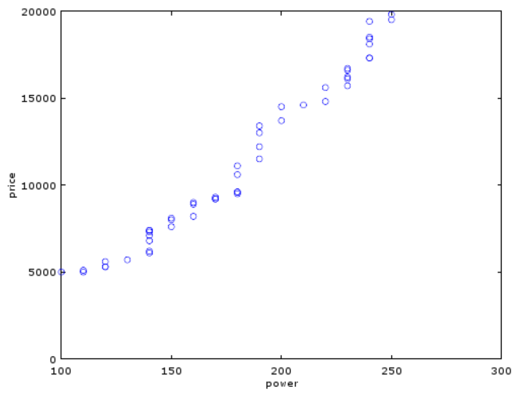
Did you get the same price that you found through gradient descent?

Yes the group were able to obtain the same price that was generated in the gradient descent which is $293,080.

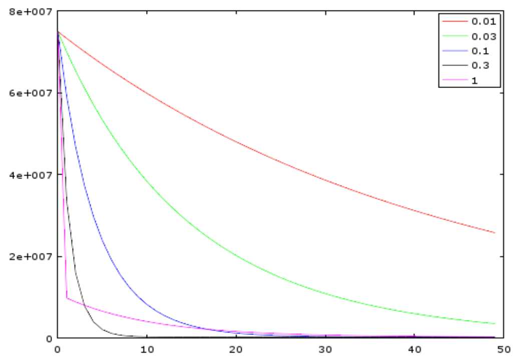
PROCEDURE 2.4

Provided the group's own data with at least 50 elements and at least 2 input features, repeat steps 2.1 - 2.3.





What is the best learning rate that you have found?



Best learning rate, alpha = 0.3

Final values of θ after gradient descent

θ1 = ___11322____

θ2 = ___2313.2____

θ3 = ___2334.8____

Final values of θ after normal equation

θ1 = ___-6344.5710

θ2 = ___3.8027____

θ3 = ___53.9828____

Estimation using gradient descent (X2 = 1600, X3 = 200)
= <u>10 536</u>
Estimation using normal equation (X2 = 1600, X3 = 200)
= <u>10 498</u>

IV. CONCLUSION

In this lab experiment, we learned how to implement multivariate linear regression by using gradient descent formula which could be used to optimize the learning rate of plots. By using multivariate regression, we obtained a very accurate and reliable plot which is evidently shown when the output theta of the gradient formula was compared to the theta obtained using the normal equation. We also learned that alpha or the learning rate of the data should be balanced or in the middle value because it being too small or too large may result to diverging values. By analysing the plot for the cost function (J), we learned that a faster drop means a better learning curve.

V. APPENDICES

Octave/Matlab Code:

```
x = load('ex2x.dat');
y = load('ex2y.dat');
m=length(x);
x = [ones(m,1), x];
theta=[0 0 0];
sigma = std(x);
mu = mean(x);
x(:,2) = (x(:,2) - mu(2))./ sigma(2);
x(:,3) = (x(:,3) - mu(3))./ sigma(3);

theta1 = [0 0 0];
alpha1 = 0.03;
for num_iterations = 1:50
J1(num_iterations) = (1 / 2 * m) * ((x * theta1') - y)' * ((x * theta1') - y);
theta1 = theta1 - (alpha1 / m .* (x' * ((x * theta1') - y)))';
endfor
theta2 = [0 0 0];
alpha2 = 0.1;
for num_iterations = 1:50
J2(num_iterations) = (1 / 2 * m) * ((x * theta2') - y)' * ((x * theta2') - y);
theta2 = theta2 - (alpha2 / m .* (x' * ((x * theta2') - y)))';
endfor
theta3 = [0 0 0];
alpha3 = 0.3;
for num_iterations = 1:50
J3(num_iterations) = (1 / 2 * m) * ((x * theta3') - y)' * ((x * theta3') - y);
theta3 = theta - (alpha3 / m .* (x' * ((x * theta3') - y)))';
endfor

theta = [0; 0; 0];
alpha = 1;
for num_iterations = 1:100
J(num_iterations) = (1 / 2 * m) .* (x * theta - y)' * (x * theta - y);
h = (1 / m) .* x' * ((x * theta) - y);
theta = theta - alpha .* h;
endfor

plot(0:99, J(1:100), '-');
price = (theta(1) * 1) + (theta(2) * (1650 - mu(2)) / sigma(2)) + (theta(3) * (3 - mu(3)) / sigma(3));

theta = [0; 0; 0];
theta = inv (x' * x) * x' * y;
price = theta(1) + (theta(2) * 1650) + (theta(3) * 3);
```

VI. BIBLIOGRAPHY

[1] G.C. Reinsel, 1998, *pg. 1 Multivariate Linear Regression.* Multivariate Reduced Rank Regression.
[2] Ng, A. 2012. *Multivariate Linear Regression.* Machine Learning.