Nick Montana

Professor Yan Yan

Intro to Machine Learning

3/20/2018

Project 1 Report

Introduction: Project 1 consists of completing various functions. Significant code snippets and outputs can be found below:

**computeCost.m**:

```
% ====================== YOUR CODE HERE ======================
% Instructions: Compute the cost of a particular choice of theta
%               You should set J to the cost.


h = X*theta;     %This gives the h_theta values in a 97-length array.

sum = 0;

cost_sum = 0;
for i = 1:m
    a = h(i) - y (i);
    cost_sum = cost_sum + a^2;
end

J = cost_sum/(2*m);

% =========================================================================
```

**computeCostMulti.m**:

```
% ====================== YOUR CODE HERE ======================
% Instructions: Compute the cost of a particular choice of theta
%               You should set J to the cost.


h = X*theta;     %This gives the h_theta values in a 97-length array.

sum = 0;

cost_sum = 0;
for i = 1:m
    a = h(i) - y (i);
    cost_sum = cost_sum + a^2;
end

J = cost_sum/(2*m);

% ==========================================================================
```

**featureNormalize.m**:

```
% ====================== YOUR CODE HERE ======================
% Instructions: First, for each feature dimension, compute the mean
%               of the feature and subtract it from the dataset,
%               storing the mean value in mu. Next, compute the
%               standard deviation of each feature and divide
%               each feature by it's standard deviation, storing
%               the standard deviation in sigma.
%
%               Note that X is a matrix where each column is a
%               feature and each row is an example. You need
%               to perform the normalization separately for
%               each feature.
%
% Hint: You might find the 'mean' and 'std' functions useful.

m = size (X, 1);
n = size (X, 2);

mu = mean (X, 1);
sigma = std (X, 1);

X_norm = X;

for j = 1:n
    for i = 1:m
        X_norm (i, j) = X (i, j) - mu (j);
    end
end

for j = 1:n
    for i = 1:m
        X_norm (i, j) = X_norm (i, j) / sigma (j);
    end
end

% ============================================================
```

**gradientDescent.m**:

```
% ====================== YOUR CODE HERE ======================
    % Instructions: Perform a single gradient step on the parameter vector
    %               theta.
    %
    % Hint: While debugging, it can be useful to print out the values
    %       of the cost function (computeCost) and gradient here.
    %

    h = X*theta;      %This gives the h_theta values in a 97-length array.
                      %This is basically the prediction in our model.

    cost_sum = 0;     %Starting with sum being 0.

    %%%%%%%%%%%%This updates theta_0 part%%%%%%%%%%%%%%%
```

```
    for i = 1:m       %This calculates the summation part
        a = h(i) - y (i);
        cost_sum = cost_sum + a*(X(i, 1));
    end

    update = (alpha / m) * cost_sum;   %This is the - " " part

    temp0 = theta (1) - update;        %Updates theta and stores in temp
                                       % variable

    %%%%%%%%%%%This updates theta_1 part%%%%%%%%%%%%%
    %Follows the same process from theta_0
    cost_sum = 0;

    for i = 1:m
        a = h(i) - y (i);
        cost_sum = cost_sum + a*(X(i, 2));
    end

    update = (alpha / m) * cost_sum;

    temp1 = theta (2) - update;

    theta (1) = temp0;           %This actually updates theta
    theta (2) = temp1;

end
% ============================================================
```

**gradientDescentMulti.m**:

```
% ====================== YOUR CODE HERE ======================
    % Instructions: Perform a single gradient step on the parameter vector
    %               theta.
    %
    % Hint: While debugging, it can be useful to print out the values
    %       of the cost function (computeCostMulti) and gradient here.
    %



    h = X*theta;      %This gives the h_theta values in a 97-length array.
                      %This is basically the prediction in our model.

    temp = zeros (1, size (X, 2));

    for j = 1:size (X, 2)

        cost_sum = 0;     %Starting with sum being 0.

        for i = 1:m       %This calculates the summation part
            a = h(i) - y (i);
            cost_sum = cost_sum + a*(X(i, j));
        end
```

```
        update = (alpha / m) * cost_sum;    %This is the - " " part

        temp (j) = theta (j) - update;         %Updates theta and stores in
                                              % temp variable
    end

    for i = 1:size (X, 2)
        theta (i) = temp (i);
    end


    % ============================================================
```

**normalEqn.m**:

```
% ====================== YOUR CODE HERE ======================
% Instructions: Complete the code to compute the closed form solution
%               to linear regression and put the result in theta.
%
% ----------------------------------------------------------------

theta = inv(X'*X)*(X'*y);

% ============================================================
```

**plotData.m**:

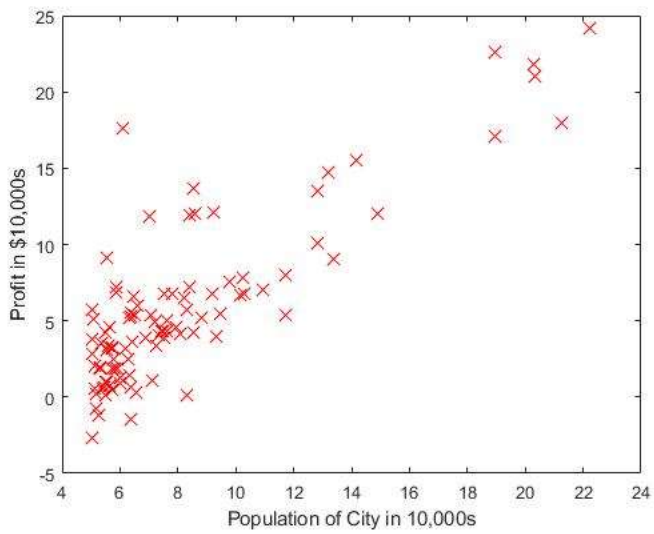
```
% ====================== YOUR CODE HERE ======================

plot (x,y, 'rx', 'MarkerSize', 10);         % Plot the data
ylabel('Profit in $10,000's');              % Set the y-axis label
xlabel('Population of City in 10,000s');    % Set the x-axis label

% Instructions: Plot the training data into a figure using the
%               "figure" and "plot" commands. Set the axes labels using
%               the "xlabel" and "ylabel" commands. Assume the
%               population and revenue data have been passed in
%               as the x and y arguments of this function.
%
% Hint: You can use the 'rx' option with plot to have the markers
%       appear as red crosses. Furthermore, you can make the
%       markers larger by using plot(..., 'rx', 'MarkerSize', 10)

% ============================================================
```

**warmUpExercise.m**:

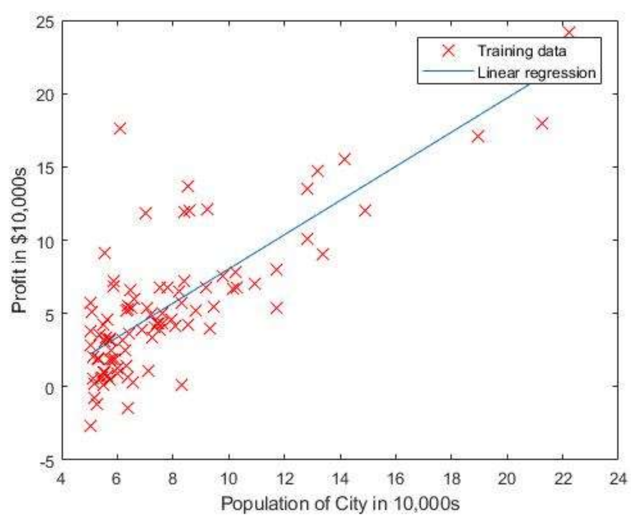
```
% ============= YOUR CODE HERE ==============
% Instructions: Return the 5x5 identity matrix
%               In octave, we return values by defining which variables
%               represent the return values (at the top of the file)
```

```
%                 and then set them accordingly.

A = eye (5);

% ===========================================
```

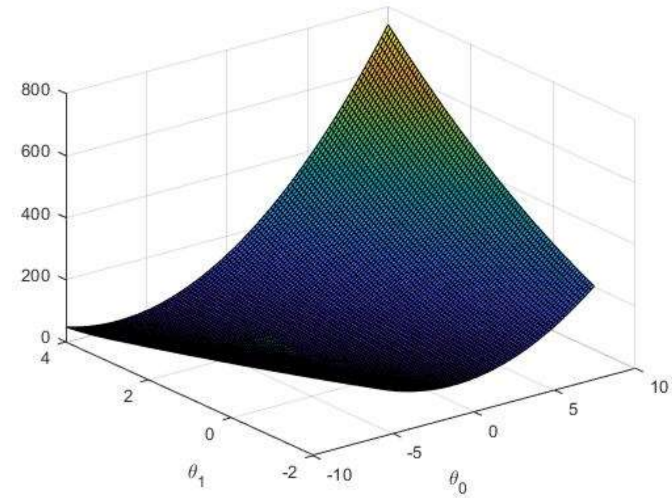
ex1 Figure1:



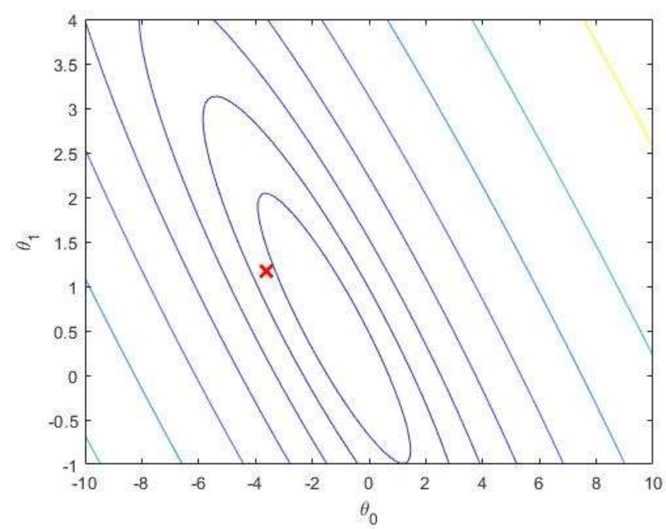

ex1 Figure 1 with Line:



ex1 Figure 2:




ex1 Figure 3:



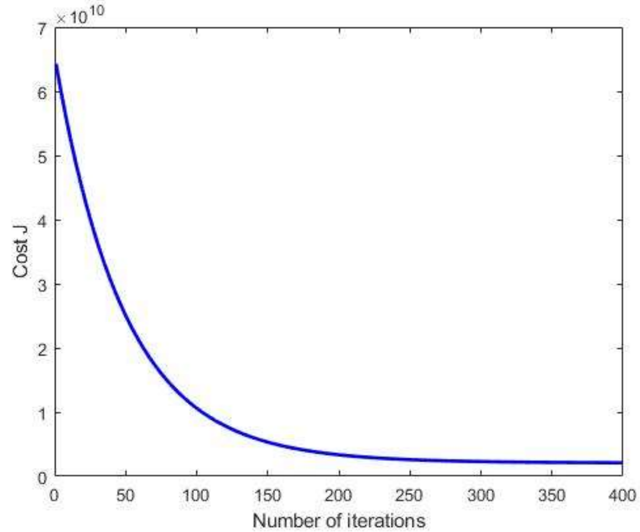
ex1 Output:

Command Window

New to MATLAB? See resources for Getting Started.

```
Running warmUpExercise ...
5x5 Identity Matrix:

ans =

     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
     0     0     0     1     0
     0     0     0     0     1

Program paused. Press enter to continue.
Plotting Data ...
Program paused. Press enter to continue.

Testing the cost function ...
With theta = [0 ; 0]
Cost computed = 32.072734
Expected cost value (approx) 32.07

With theta = [-1 ; 2]
Cost computed = 54.242455
Expected cost value (approx) 54.24
Program paused. Press enter to continue.

Running Gradient Descent ...
Theta found by gradient descent:
-3.630291
1.166362
Expected theta values (approx)
 -3.6303
  1.1664

For population = 35,000, we predict a profit of 4519.767868
For population = 70,000, we predict a profit of 45342.450129
Program paused. Press enter to continue.
Visualizing J(theta_0, theta_1) ...
>>
```

ex1_multi Figure 1:




ex1_multi Output:

Command Window

New to MATLAB? See resources for Getting Started.

```
Loading data ...
First 10 examples from the dataset:
 x = [2104 3], y = 399900
 x = [1600 3], y = 329900
 x = [2400 3], y = 369000
 x = [1416 2], y = 232000
 x = [3000 4], y = 539900
 x = [1985 4], y = 299900
 x = [1534 3], y = 314900
 x = [1427 3], y = 198999
 x = [1380 3], y = 212000
 x = [1494 3], y = 242500
Program paused. Press enter to continue.
Normalizing Features ...
Running gradient descent ...
Theta computed from gradient descent:
 334302.063993
 99411.449474
 3267.012854

Predicted price of a 1650 sq-ft, 3 br house (using gradient descent):
 $289221.547371
Program paused. Press enter to continue.
Solving with normal equations...
Theta computed from the normal equations:
 89597.909543
 139.210674
 -8738.019112

Predicted price of a 1650 sq-ft, 3 br house (using normal equations):
 $293081.464335
fx >>
```