CSE 440

Assignment 5 – Linear Regression

Submitted to -

M Ehsanul Karim

Submitted by -
Mahbuba Tasmin
1610064042

Data plot

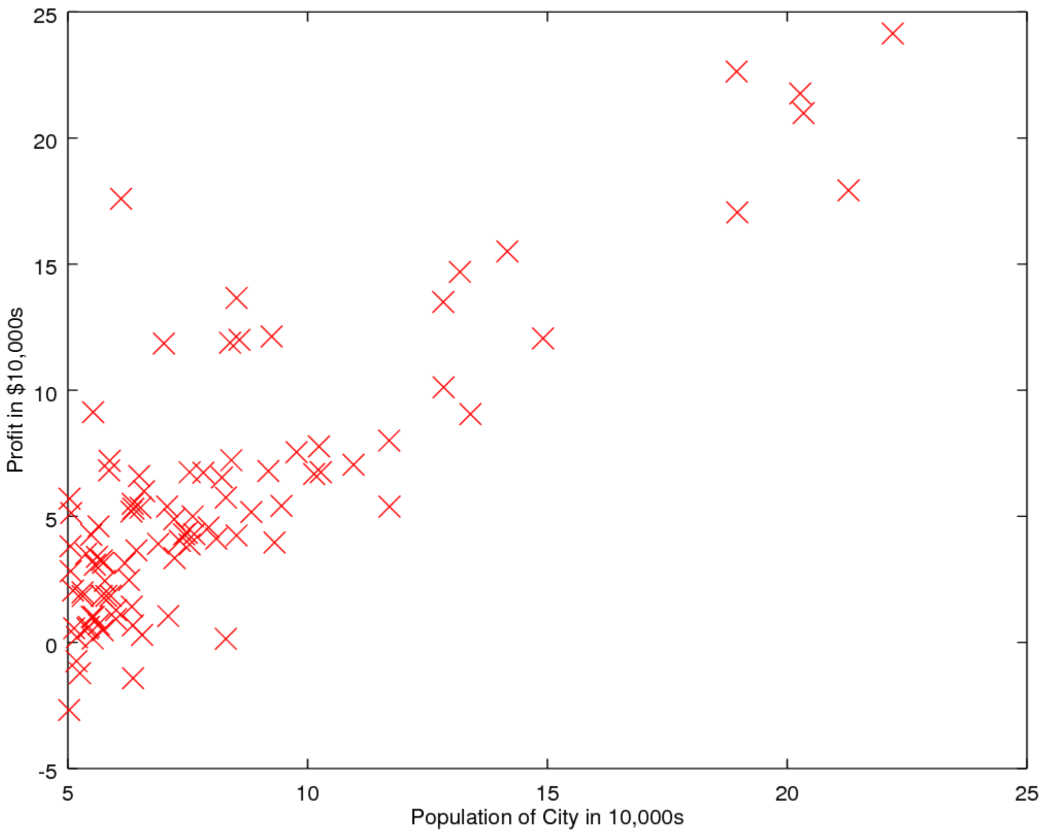


Figure – Scatter plot of training data

The file ex1data1.txt contains the dataset for our linear regression problem. The first column is the population of a city and the second column is the profit of a food truck in that city.

For 2-dimensional data, a scatter plot is a necessary first step in understanding the data.A scatter plot reveals relationships or association between two variables. Such relationships manifest themselves by any non-random structure in the plot.

A scatter plot is a plot of the values of *Y* versus the corresponding values of *X*, in this plotting :

•Vertical axis: variable *Y*--usually the response variable - profit in $10,000s is placed

•Horizontal axis: variable *X*--usually some variable we suspect may be related to the response - population of City in 10,000s

This sample plot reveals a linear relationship between the two variables indicating that a linear regression model might be appropriate

Training data with linear regression fit:

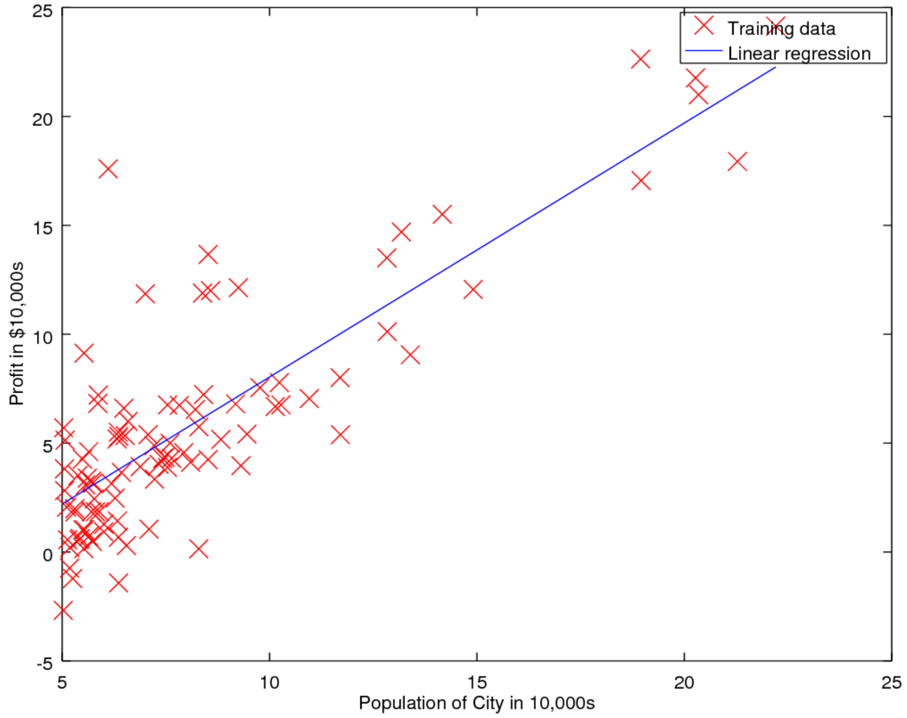


Figure 2: Training data with linear regression fit

Regression: Predict continuous real valued output (price)

hypothesis of linear regression is $h_\theta(x) = \theta_0 + \theta_1 * x$

And the idea is to choose θ_0 and θ_1 such that $h_\theta(x)$ is close to y for our training examples(x,y).

After minimizing values of θ_0 , θ_1 through running cost functions on $j(\theta_0 , \theta_1)$, we obtain a hypothesis $h_\theta(x)$. That hypothesis is represented through the blue straight line on the graph. As it is visible, most of the red marks lie nearby to the blue line , which justifies the correct choice of θ_0 and θ_1 in hypothesis.

Cost function J :

Through this graphs, linear regression implementation(cost function) is presented through $J(\theta_0 ,\theta_1)$. A cost function lets us figure out how to fit the best straight line to our data.

Hypothesis of linear regression is $\mathbf{h_\theta(x) = \theta_0 + \theta_1 * x}$ is like a prediction machine, throw in an *x* value, get a *y* value. **Cost -** is a way to, using your training data, determine values for your θ values which make the hypothesis as accurate as possible.This cost function is also called the squared error cost function.

$$h_\theta(x) = \theta_1 x$$

$$\theta_0 = 0$$

$$\theta_1$$

$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\underset{\theta_1}{\text{minimize}}\ J(\theta_1)$$

From the data set provided, for any training example pair, we can compute the value of minimized value of J(θ_0, θ_1) from the equations mentioned above.

Contour plot :

A contour plot is a graphical technique for representing a 3-dimensional surface by plotting constant z slices, called contours, on a 2-dimensional format. That is, given a value for z lines are drawn for connecting the (x,y) coordinates where that z value occurs.
The contour plot is an alternative to a 3-D surface plot.

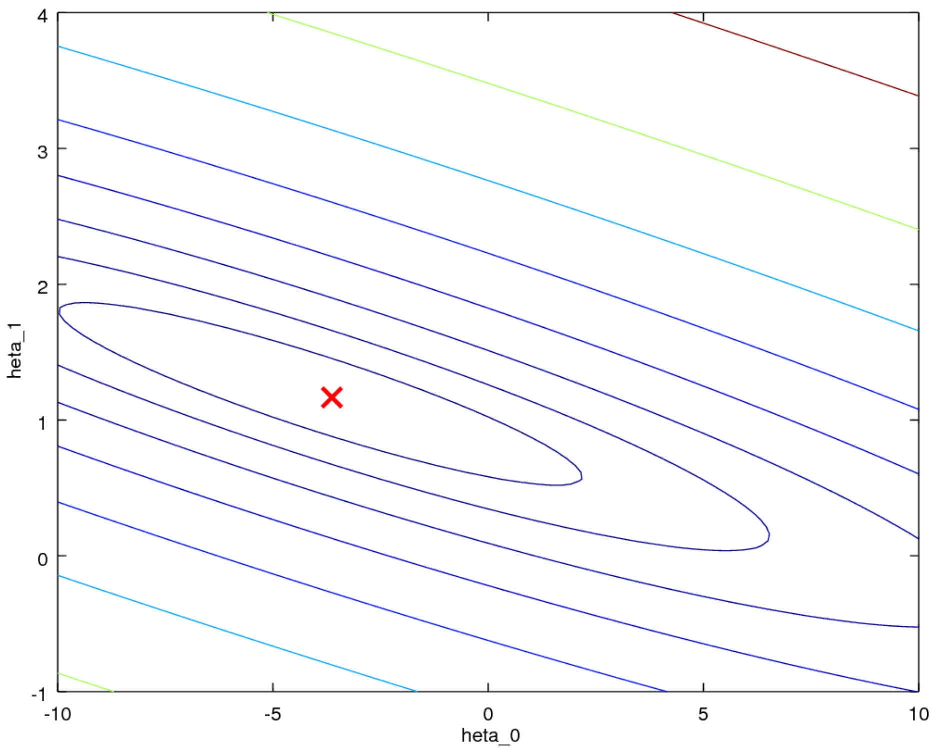


Figure :Contour plot ,showing minimum

The contour plot is formed by:

- •Vertical axis: Independent variable - θ_1
- •Horizontal axis: Independent variable - θ_0
- •Lines: iso-response values

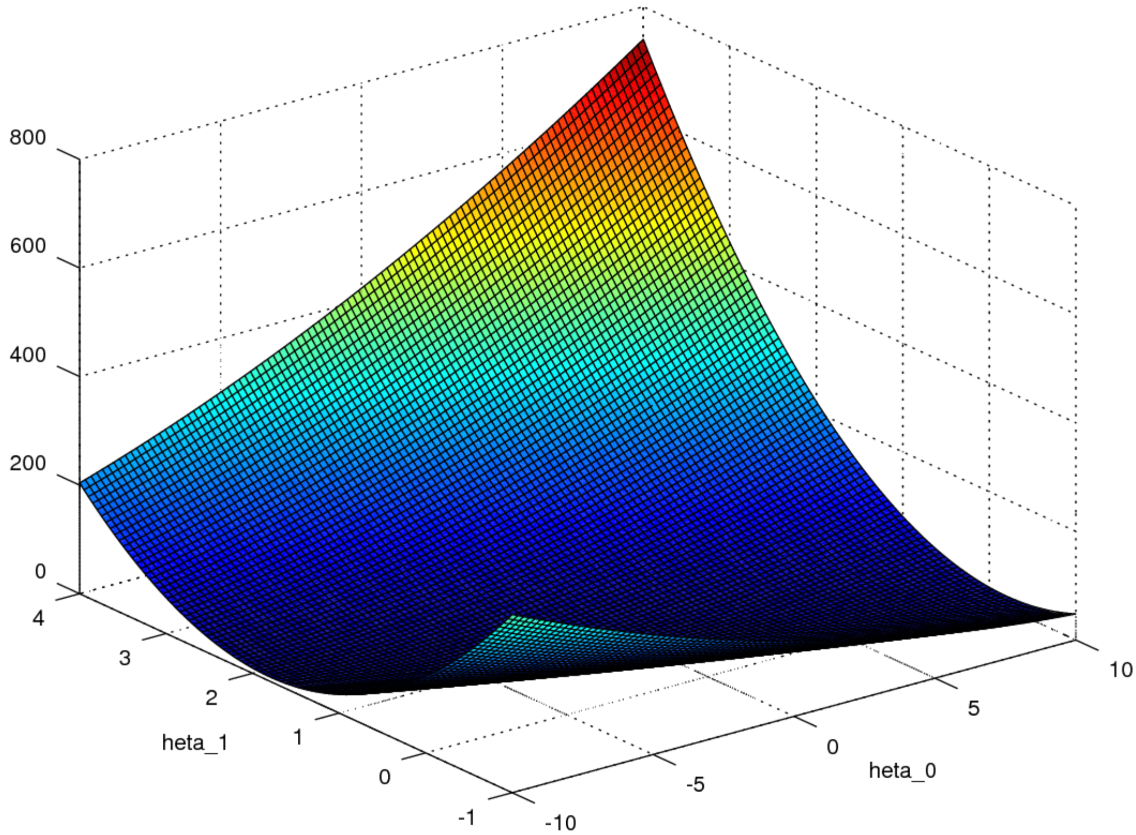
Using our original hypothesis with two variables, cost function stands as $J(\theta_0,\theta_1)$.

For example, for $\theta_0 = 5$, $\theta_1 = 1$, we have two parameters and generates a contour plot where ellipses in different colors are used to present values of $J(\theta_0,\theta_1)$. Each different color stands for the same value of $J(\theta_0,\theta_1)$, but obviously plot to different co-ordinates since θ_0,θ_1 will vary.

Each point (like the red one above in the middle) represents a pair of parameter values for Ө0 and Ө1.

Our example here put the values at $\theta_0 = -4$, $\theta_1 = 1.8$ and it lies in the center of all the concentric circles. It actually indicates the minimum $J(\theta_0,\theta_1)$.

Surface plot :



Cost function stands as $J(\theta_0,\theta_1)$ and since we have two parameters now,we are generating a 3D surface plot where axis are $X = \theta_1$, $Z = \theta_0$, $Y = J(\theta_0,\theta_1)$.

We can see that the height (y) indicates the value of the cost function, so for corresponding $\theta_0,\theta_1,$ minimum $J(\theta_0,\theta_1)$ can be computed and visualized in the surface plot.

The purpose of these graphs is to show that how $J(\theta)$ varies with changes in θ_0 and θ_1 . The cost function $J(\theta)$ is bowl-shaped and has a global mininum. (This is easier to see in the contour plot than in the 3D surface plot). This minimum is the optimal point for θ_0, θ_1 , and each step of gradient descent moves closer to this point.

Gradient Descent for multiple variables : Selecting learning rates

We can also observe how the error changes as we move toward the minimum. A good way to ensure that gradient descent is working correctly is to make sure that the error decreases for each iteration. Below is a plot of error values for the first 400 iterations of the above gradient search.

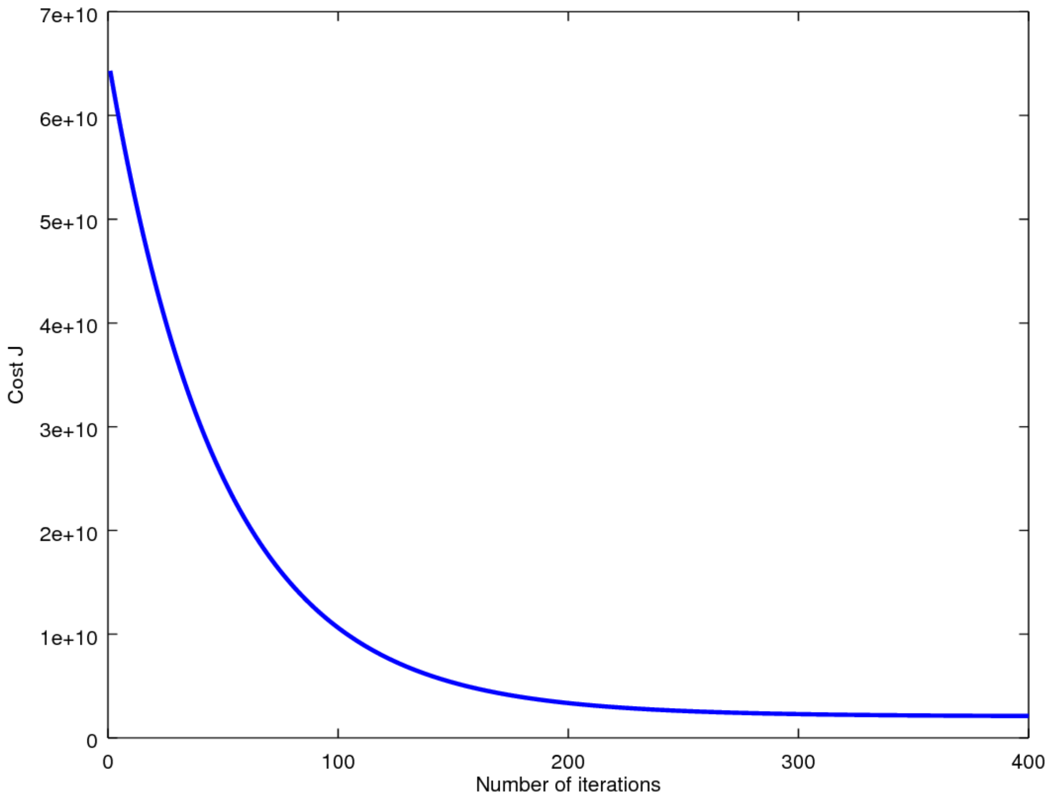
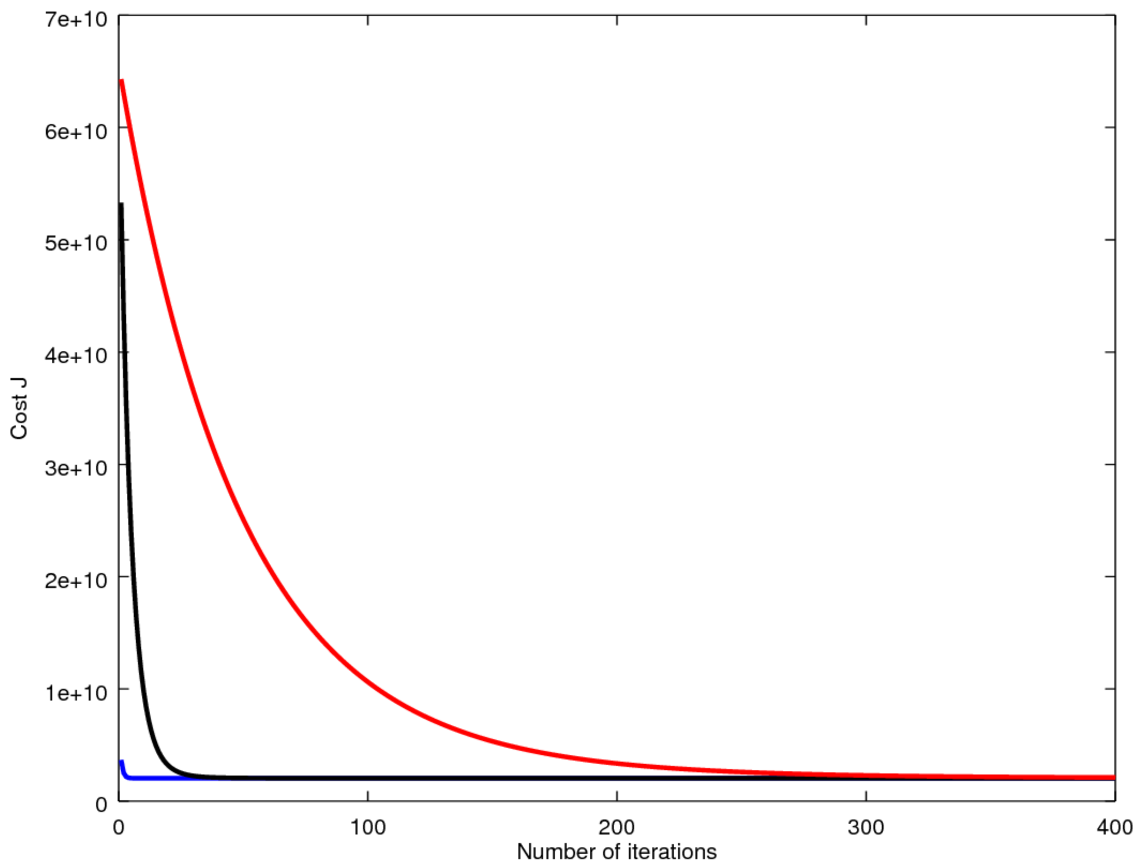


Figure : Convergence of gradient descent with an appropriate learning rate

On the x-axis, number of iterations for calculating cost J for different values of alpha is placed and on the y-axis, corresponding J values are placed.

This plot only shows the effect of one learning rate throughout 400 iterations .To compare how different learning learning rates affect convergence, it's helpful to plot J for several learning rates on the same figure.



On this plot, there are three learning rates working to show different efficiency to reach minimum J.

For the blue curve, alpha is set to to 1
for the black curve , alpha is set to 0.1
for the red curve, alpha is set to 0.01.

We can see from the trends of the curves that, for alpha = 0.01 , the curve took a lot of time steps to reach the minimum J (I.e at 200 iterations) . Whereas for alpha = 0.1, the curve took relatively less time to reach the convergence point (less than 100 iteration) . And for alpha = 1 , the curve almost instantly reached the minimum J (less than 50 iteration).
This proves that alpha = 1 is the best learning rate , which caused to reach the minimum J at least amount of time.