Sahil Jindal

2013CS10253

COL774 Assignment1

Q1.

(a) Update equation for θ :

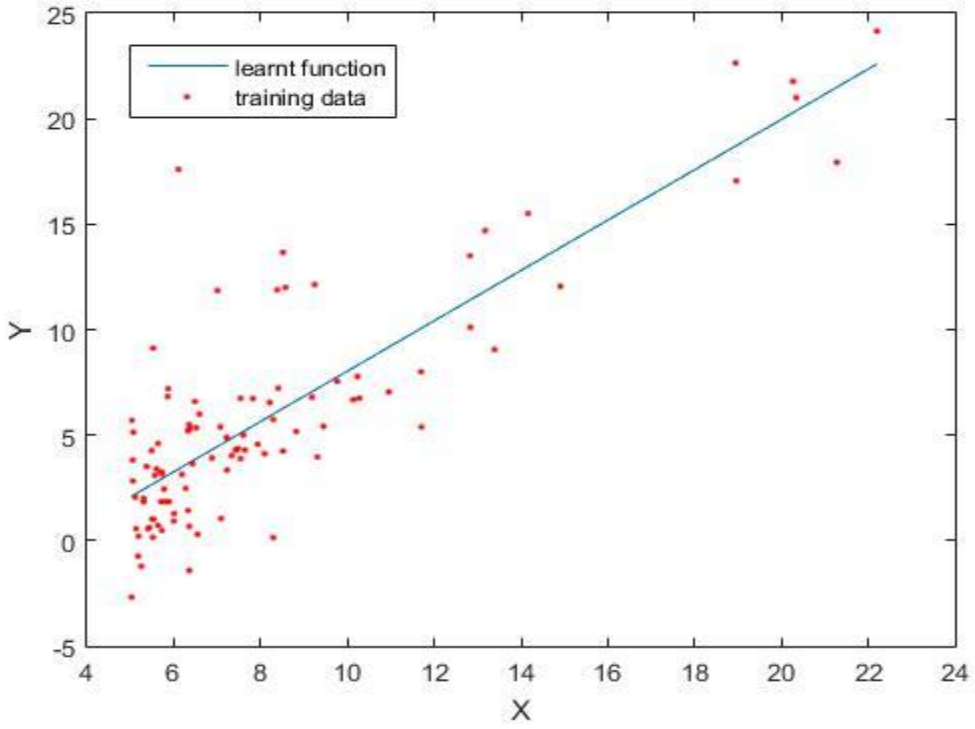
$$\theta_{(t+1)} = \theta_t + \frac{\eta}{m} * \sum_{i=1}^{m} \left[Y^{(i)} - \theta_t^T * X^{(i)} \right] * X^{(i)}$$

Learning rate: 0.1

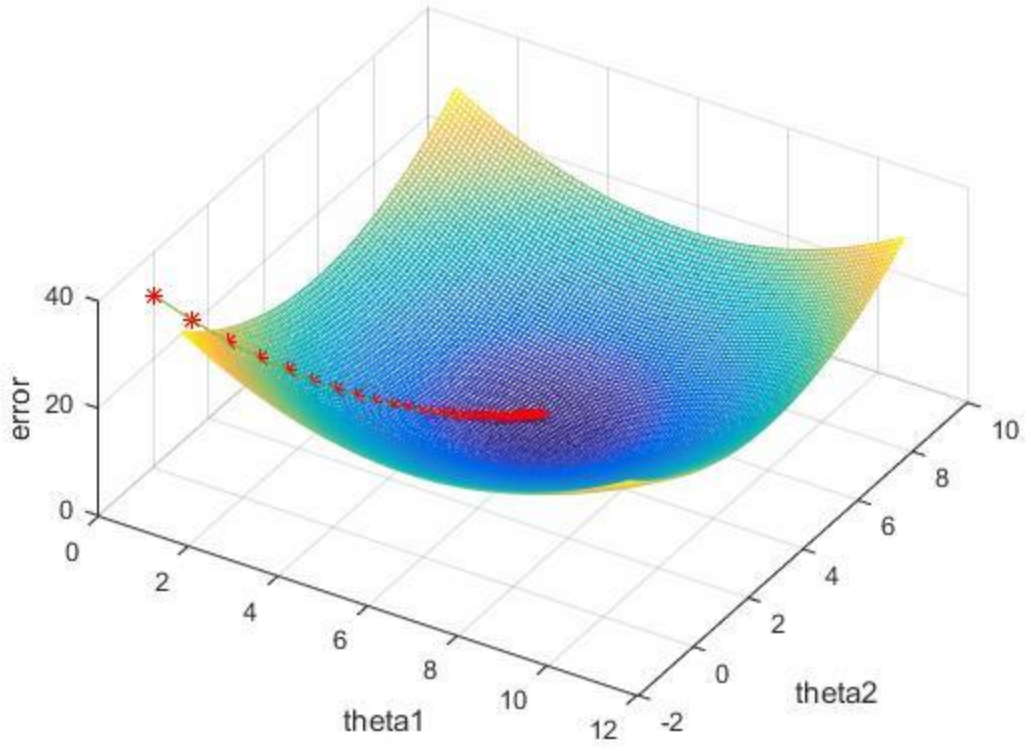
Stopping Criteria: $\left| J(\theta_{(t+1)}) - J(\theta_t) \right| \leq 10^{-9}$

Final Parameters: $\theta = \begin{bmatrix} 5.8391 \\ 4.6168 \end{bmatrix}$

(b) Plot of data and the hypothesis function learned by algorithm:



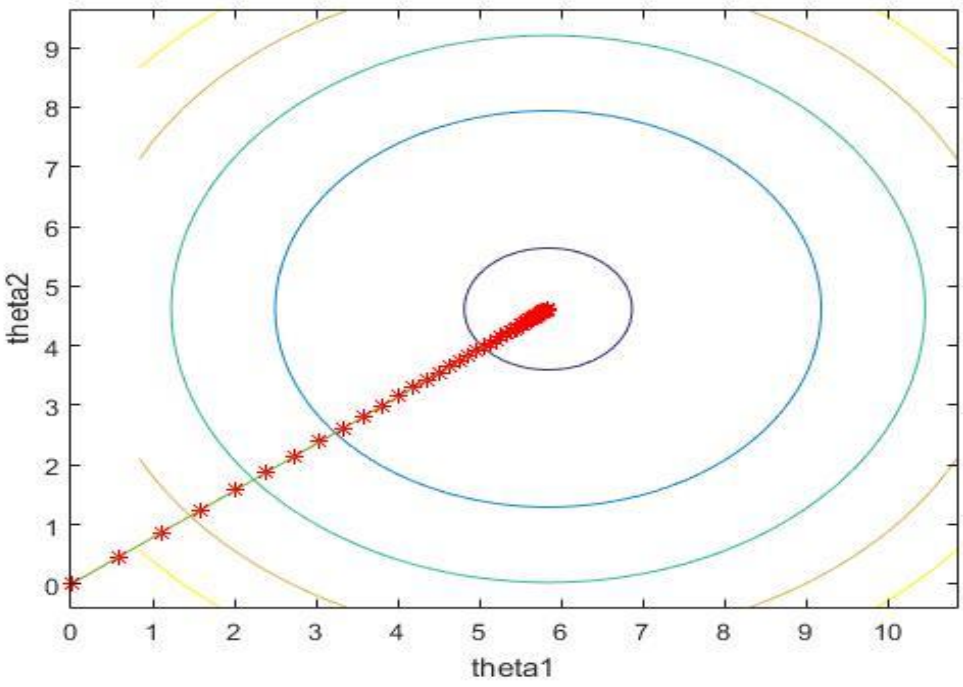
(c) 3-Dimensional mesh plot showing error function($J(\theta)$) on z-axis and parameters on x-y plane, and the line showing the path taken by the algorithm:



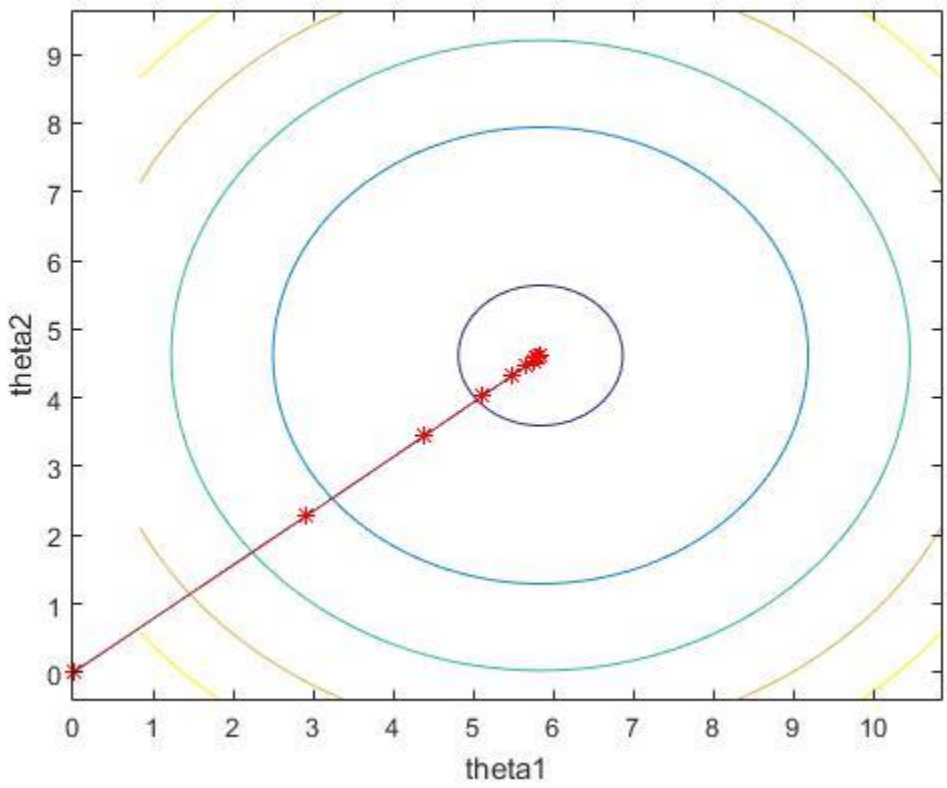
* represents state at end of each iteration

(d) Contour plots:

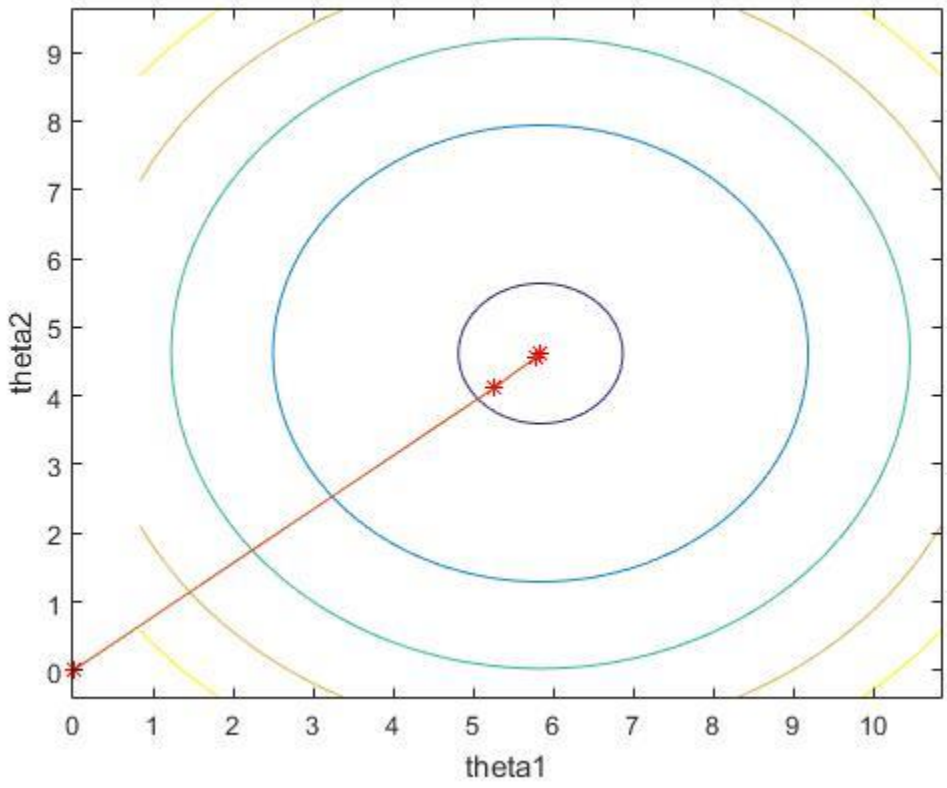
- $\eta = 0.1$ => No. of iterations = 110



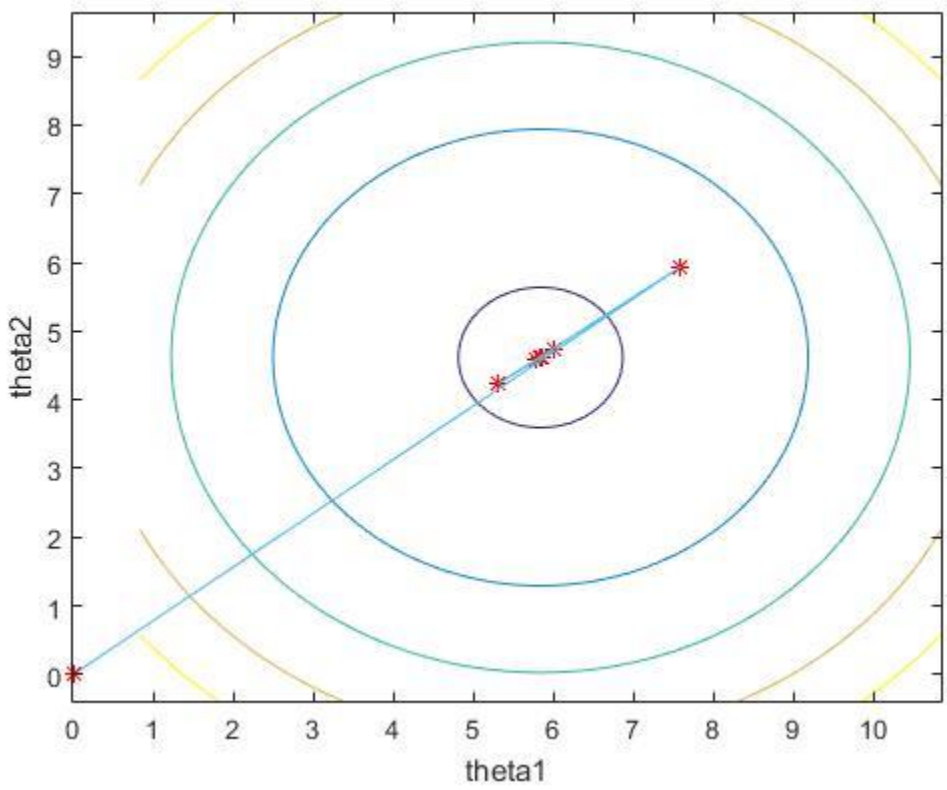
- $\eta = 0.5$ => No. of iterations = 21



- $\eta = 0.9$ => No. of iterations = 9



- $\eta = 1.3$ => No. of iterations = 13



From the above plots, we can see that $\eta = 0.9$ takes minimum number of iterations to reach minima. For $\eta < 0.9$ the learning rate is slow thereby travelling only a small distance towards the minima, hence taking more number of iterations. For $\eta > 0.9$, as in $\eta = 1.3$, θ overshoots the optimal value at which minima is obtained, and then comes back to the minima. Due to the overshooting, it takes more number of iterations. Now as η is increased, number of iterations taken increase due to overshooting of the parameters and the coming back. Hence, from above values of η, $\eta = 0.9$ is the best learning rate in terms of number of iterations taken to converge.

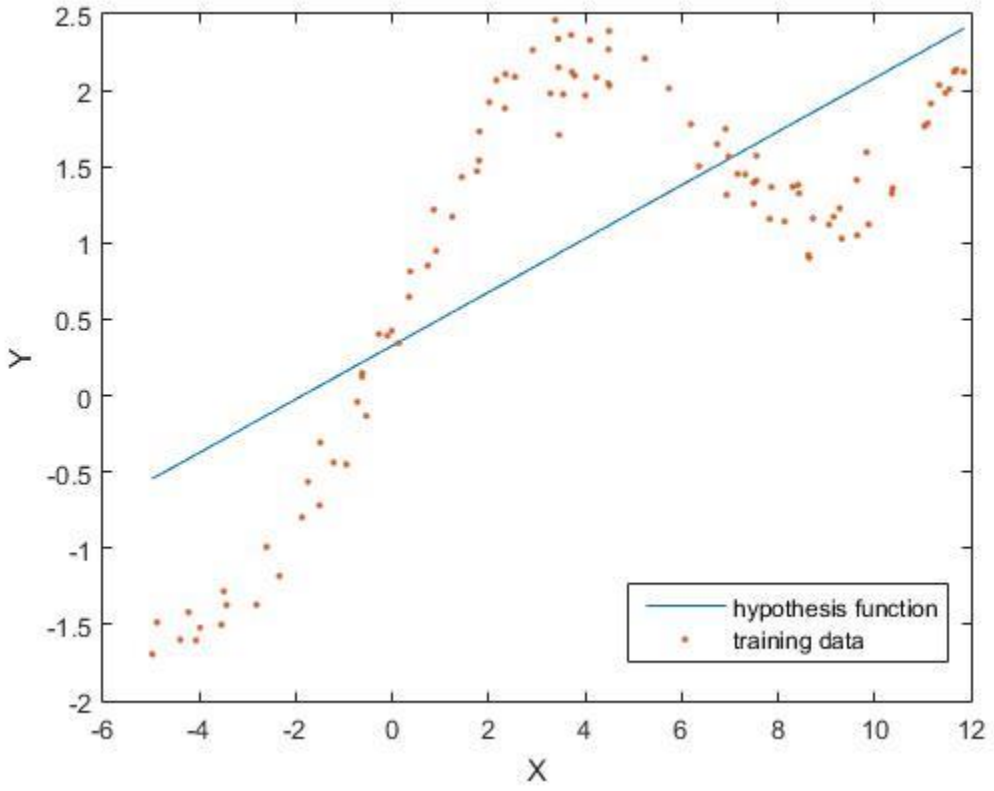
- For $\eta = 2.1$ and $\eta = 2.5$, the learning rate is so high that in each iteration, the parameters jump to the other side of optimal parameters (at which error is minimum), in effort of coming closer to the minima, but the magnitude of distance between the new point and optimal point becomes greater than that between previous point and the optimal point. Hence, in these cases, minima is never achieved, and error keeps on increasing in each iteration.

Q2.

(a) In case of unweighted linear regression, analytical solution for the parameters θ :

$$\theta = (X^T X)^{-1} X^T Y$$

Plot of data and the hypothesis function learnt by above θ:



(b) In case of weighted linear regression:

$$J(\theta) = \frac{1}{2}(X\theta - Y)^T W (X\theta - Y)$$

To minimize $J(\theta)$:

$$\frac{\partial(J(\theta))}{\partial\theta} = 0$$

$$\Rightarrow \quad \frac{\partial\left(\frac{1}{2}(X\theta - Y)^T W (X\theta - Y)\right)}{\partial\theta} = 0$$

$$\Rightarrow \quad \frac{\partial(\theta^T X^T W X\theta - \theta^T X^T W Y - Y^T W X\theta + Y^T W Y)}{\partial\theta} = 0$$

$$\Rightarrow \quad 2X^T W X\theta - X^T W Y - (Y^T W X)^T = 0$$

(because, $\frac{\partial(X^T AX)}{\partial X} = 2AX;\ \frac{\partial(A^T X)}{\partial X} = A;$

$A^T X = X^T A, when\ A^T X\ is\ symmetric;$)

=> $2X^T WX\theta - X^T W^T Y - X^T WY = 0$

As W is a diagonal matrix,

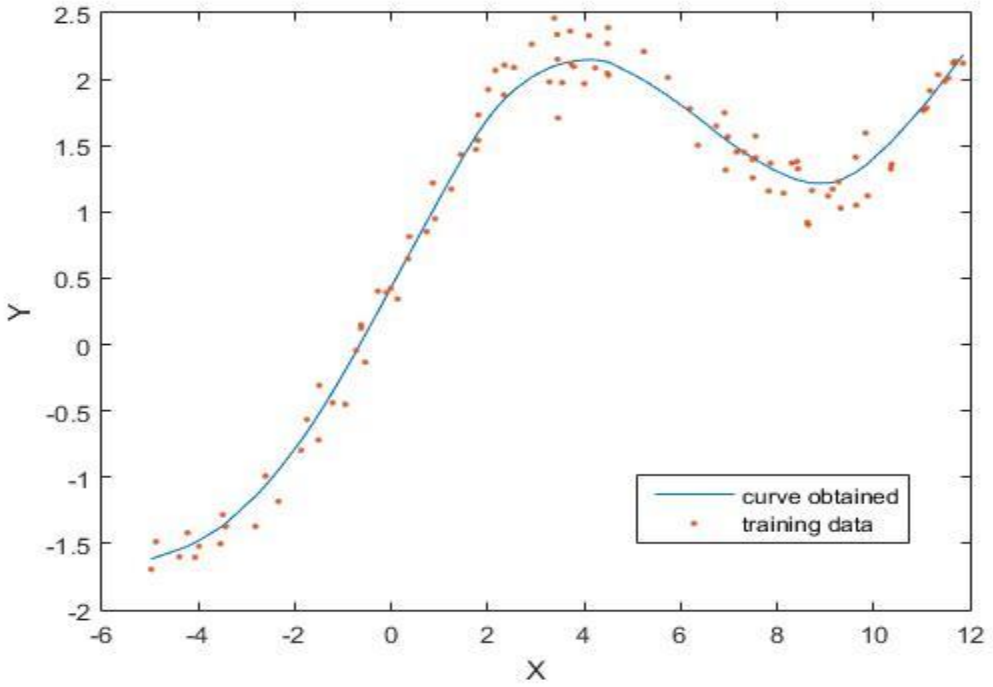
$$W^T = W$$

=> $2X^T WX\theta - 2X^T WY = 0$

=> $X^T WX\theta = X^T WY$

=> $\theta = (X^T WX)^{-1} X^T WY$

Plot of data and the curve resulting from algorithm ($\tau = 0.8$):

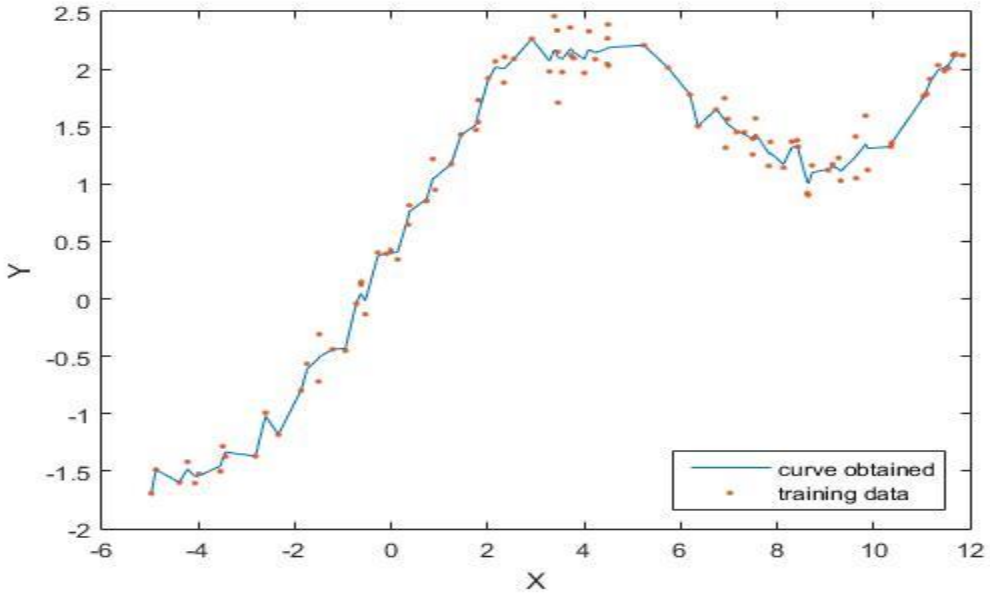


W is a diagonal matrix with $W_{ii} = w^{(i)}$,

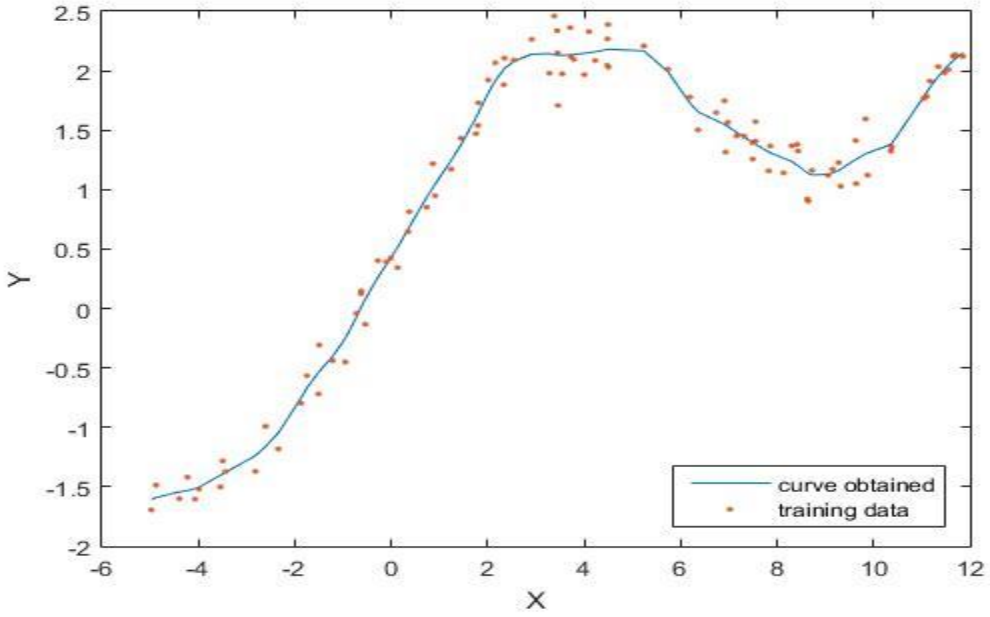
$$w^{(i)} = e^{-\frac{\left(x - x^{(i)}\right)^2}{2\tau^2}}$$

Plots of data and curves for different values of η:

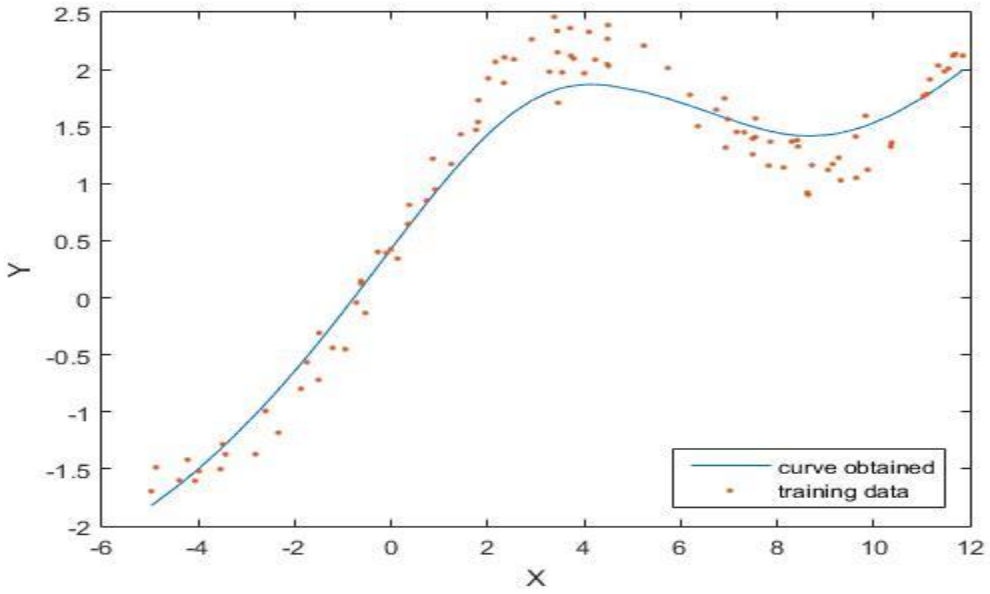
- $\tau = 0.1$


curve obtained
training data
X
Y

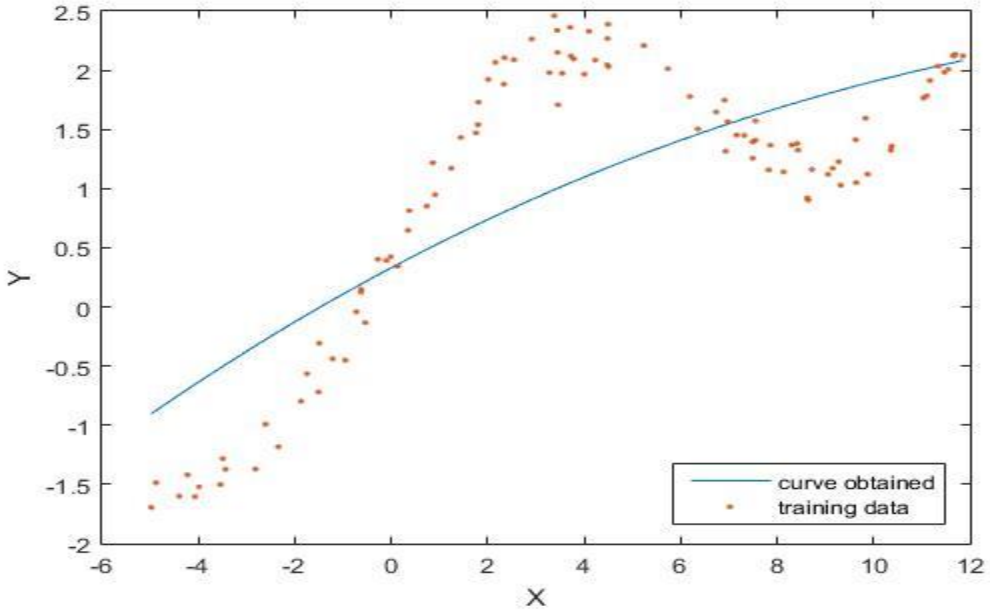
- $\tau = 0.3$


curve obtained
training data
X
Y

- $\tau = 2$



- $\tau = 10$



When τ is too small (as seen in case of 0.1), the curve tries to fit every data point, and hence is very rough. The weights of $x^{(i)}$ very close to x are high and there is a steep decay in the weights as we move far from x. This results in high change in weights on moving x slightly and hence the curve over-fits the data. So, as τ->0, the curve tends to go through every data point.
As τ is increased, the curve gets smoother, and we get a better optimized curve of the data which is acceptable.

On further increasing τ (τ = 10), the curve goes more towards linear regression, as the weights of training points as we move far from query, decrease slowly. As $\tau \rightarrow \infty$, $w(i) \rightarrow 1$, that is W tends to become an Identity matrix. Thus the analytical solution for θ becomes same as the one for unweighted linear regression.
As it can be seen from above plots $\tau = 0.8$ works best to fit the data.

Q3.

(a) Log-likelihood function for logistic regression:

$$LL(\theta) = \sum_{i=1}^{m} \left[y^{(i)} \log\left(h_\theta\left(x^{(i)}\right)\right) + \left(1 - y^{(i)}\right) \log\left(1 - h_\theta\left(x^{(i)}\right)\right) \right]$$

$$h_\theta\left(x^{(i)}\right) = \frac{1}{1 + e^{-\theta^T X}}$$

$$\Rightarrow \frac{\partial\left(h_\theta\left(x^{(i)}\right)\right)}{\partial \theta_j} = h_\theta\left(x^{(i)}\right) * \left(1 - h_\theta\left(x^{(i)}\right)\right) * x_j^{(i)}$$

Entry in Hessian matrix:

$$H = \nabla_\theta^2 LL(\theta)$$

$$H_{ij} = \frac{\partial^2\left(LL(\theta)\right)}{\partial \theta_i \theta_j}$$

$$\Rightarrow H_{jk} = -\sum_{i=1}^{m} h_\theta\left(x^{(i)}\right)\left(1 - h_\theta\left(x^{(i)}\right)\right) x_j^{(i)} x_k^{(i)}$$

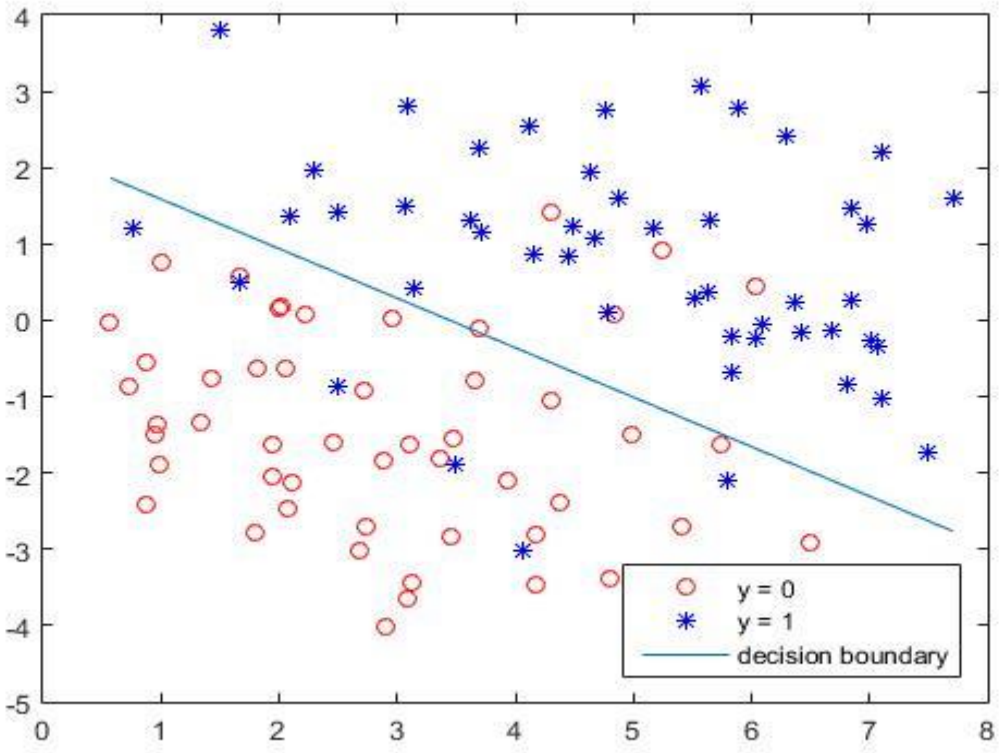
Newton's update method:

$$\theta^{(t+1)} = \theta^t - (H^{-1}) \nabla_\theta LL(\theta)$$

Coefficients θ resulting from the Newton's update algorithm:

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{-2.6205} \\ \mathbf{0.7604} \\ \mathbf{1.1719} \end{bmatrix}$$

(b) Plot of training data and decision boundary obtained by logistic regression:



Q4. Gaussian Discriminant Analysis

(a) Values obtained:

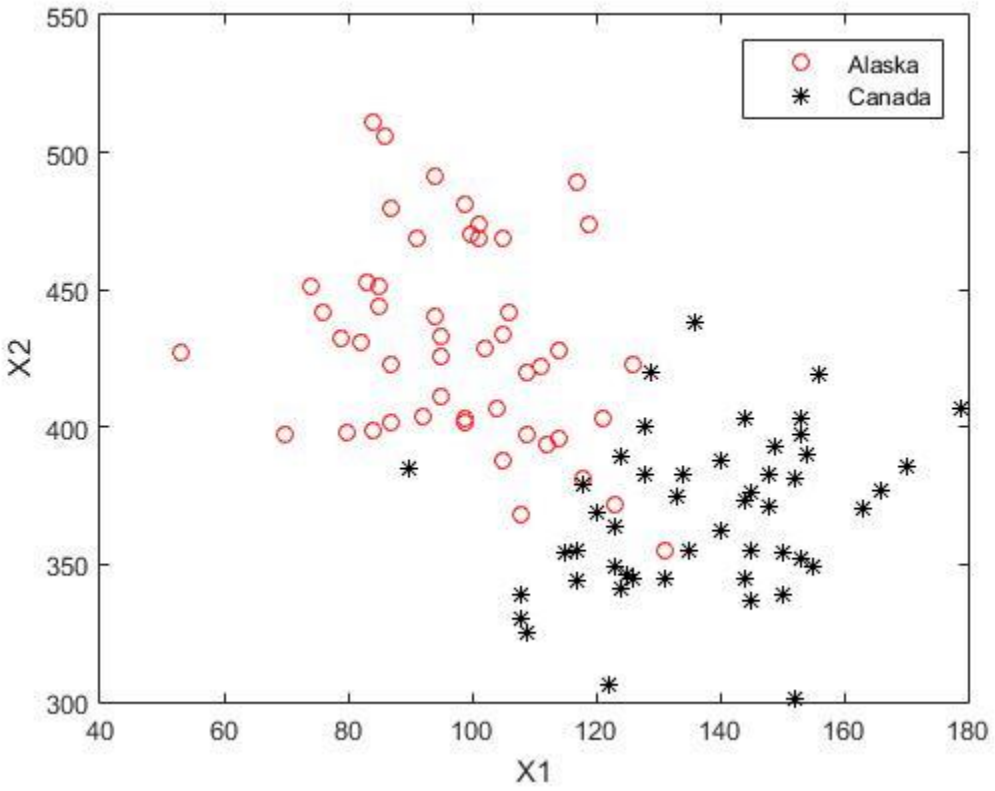
$$\phi = 0.5$$

$$\mu_0 = [98.38 \quad 429.66]$$

$$\mu_1 = [137.46 \quad 366.62]$$

$$\Sigma_0 = \Sigma_1 = \Sigma = \begin{bmatrix} 287.5 & -26.7 \\ -26.7 & 1123.3 \end{bmatrix}$$

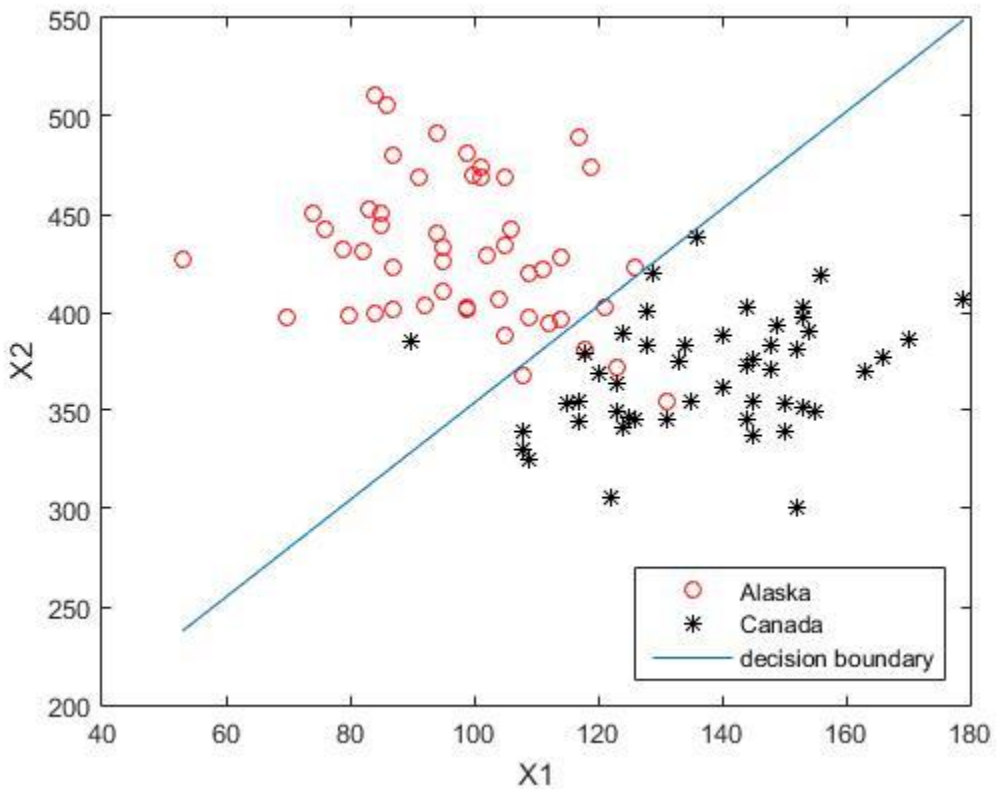
(b) Plot of training data:



(c) Equation for decision boundary in case the two classes have identical covariance matrix:

$$\log\left(\frac{1-\phi}{\phi}\right) - \frac{1}{2}[(x-\mu_0)^T\Sigma^{-1}(x-\mu_0) - (x-\mu_1)^T\Sigma^{-1}(x-\mu_1)] = 0$$

Plot for GDA in case of $\Sigma_0 = \Sigma_1$:



(d) Target classes have their own covariance matrix that might be different.

$$\Sigma_0 = \frac{\left(\sum_{i=0}^{m} 1\{y^{(i)} = 0\}\left(x^{(i)} - \mu_{y^{(i)}}\right)\left(x^{(i)} - \mu_{y^{(i)}}\right)^T\right)}{\sum_{i=0}^{m} 1\{y^{(i)} = 0\}}$$

$$\Sigma_1 = \frac{\left(\sum_{i=0}^{m} 1\{y^{(i)} = 1\}\left(x^{(i)} - \mu_{y^{(i)}}\right)\left(x^{(i)} - \mu_{y^{(i)}}\right)^T\right)}{\sum_{i=0}^{m} 1\{y^{(i)} = 1\}}$$

Values obtained:

$$\mu_0 = [98.38 \quad 429.66]$$

$$\mu_1 = [137.46 \quad 366.62]$$

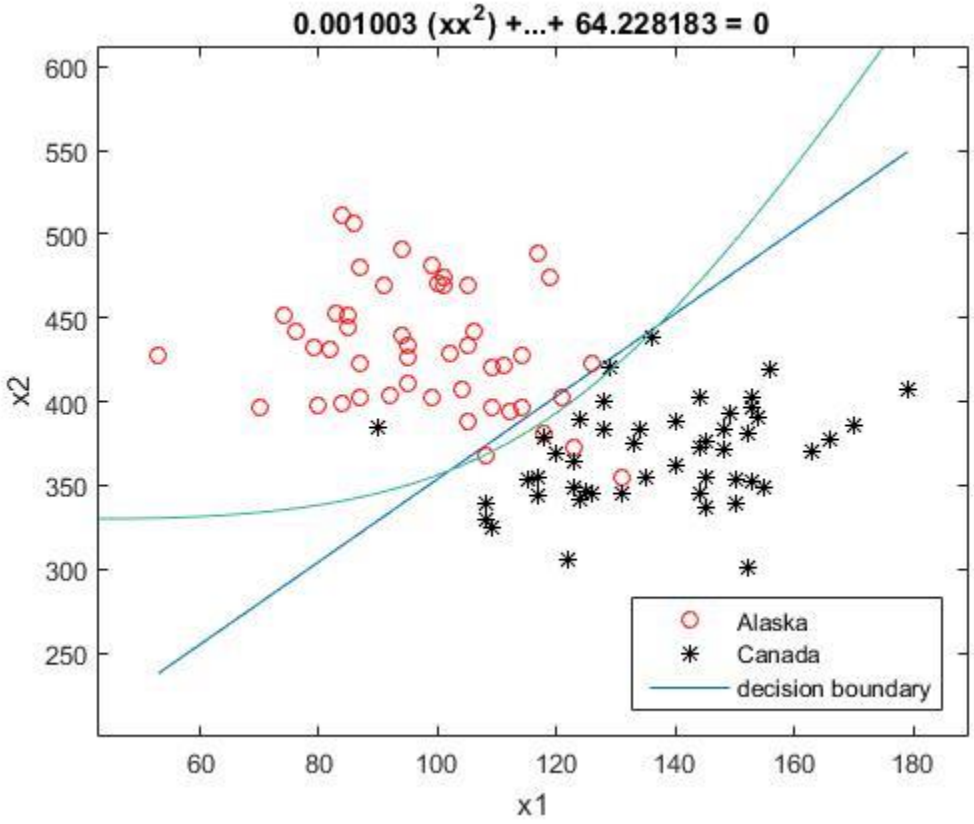
$$\Sigma_0 = \begin{bmatrix} 255.4 & -184.3 \\ -184.3 & 1371.1 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 319.5684 & 130.8348 \\ 130.8348 & 875.3956 \end{bmatrix}$$

(e) Equation for quadratic boundary separating the two regions in case of $\Sigma_0 \neq \Sigma_1$:

$$\log\left(\frac{(1-\phi)|\Sigma_1|^{\frac{1}{2}}}{\phi|\Sigma_0|^{\frac{1}{2}}}\right) - \frac{1}{2}[(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0) - (x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)] = 0$$

Plot of training data and quadratic boundary as well as linear boundary:



As seen in the graph above, quadratic boundary better classifies the data as compared to the linear boundary. The quadratic boundary takes into account the difference in covariance matrix of the two classes, and hence providing a better estimate, and less mistakes. Quadratic boundary provides a better estimate of the classification of data at the boundary points (crossing from one class to other), and takes into account the data points which are on the wrong side (exceptions) (e.g. the circle at (123,372) in Alaska but is classified as Canada). More points that are in Alaska are classified in Canada than the points in Canada being classified in Alaska, hence giving the boundary the shape above tending to include more points that belong to Alaska in Alaska, rather than having no effect of them. The point (121,403) is in Alaska. But, the linear boundary classifies it

as Canada whereas quadratic boundary correctly categorizes it as Alaska. Hence, a better view of the classification of data is given by the quadratic boundary.