Machine Learning - Assignment 1

Meraj Ahmed

February 2020

1 Linear Regression

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

1.1 Implement batch gradient descent

1.1.1 Parameters

$\eta = 0.5$

Stopping criteria $= |newcost - oldcost| < 10^{-15}$

Number of iterations $= 26$

Final cost $= 1.1947898110939116 * 10^{-6}$

Final $\theta : [\theta_0, \theta_1] = [0.99662009, 0.0013402]$

1.1.2 Plot

Plot of learned hypothesis, cost function and contour is shown below

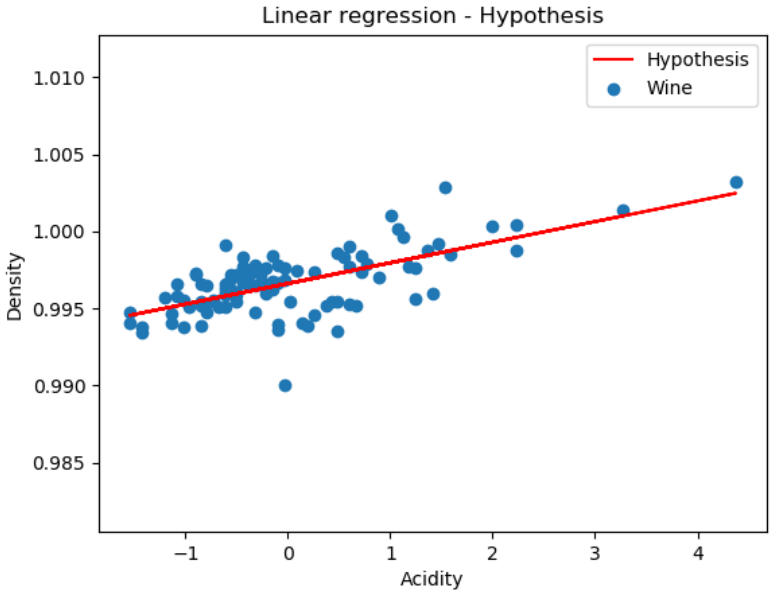


Figure 1: Linear Regression

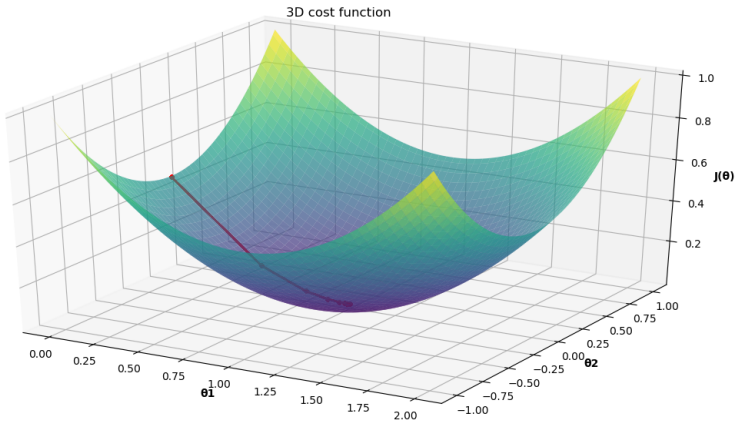


Figure 2: Cost function

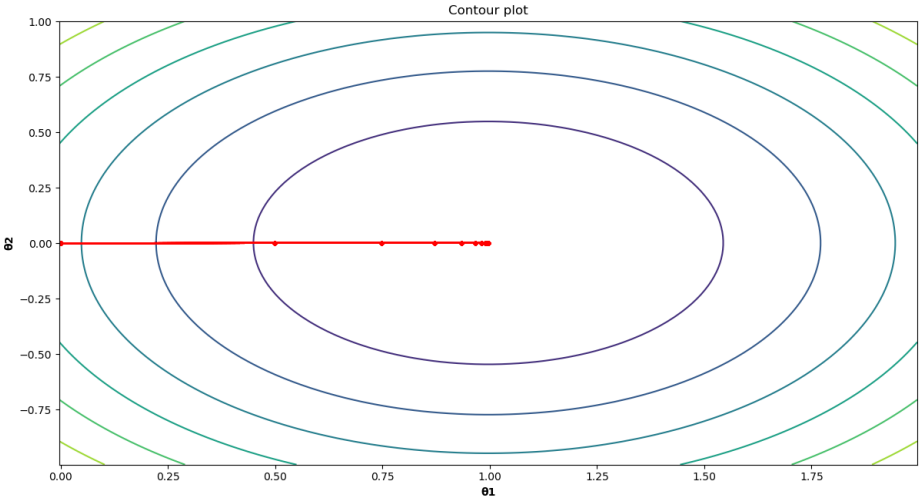


Figure 3: Contour

1.1.3 Parameters

$\eta = 0.001$

Stopping criteria $= |newcost - oldcost| < 10^{-15}$

Number of iterations $= 13806$

Final cost $= 1.1947903102024764 * 10^{-6}$

Final $\theta : [\theta_0, \theta_1] = [0.9966191, 0.00134019]$

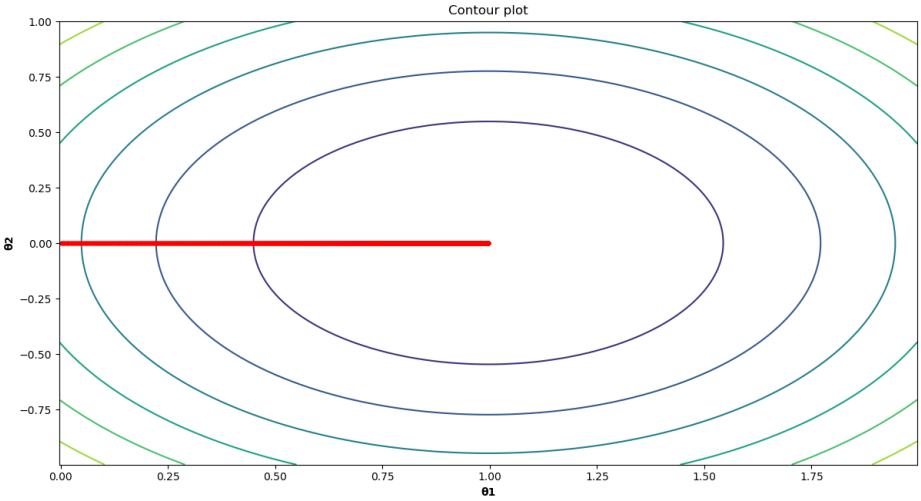


Figure 4: Contour

1.1.4 Parameters

$\eta = 0.025$

Stopping criteria $= |newcost - oldcost| < 10^{-15}$

Number of iterations $= 610$

Final cost $= 1.1947898301112708 * 10^{-6}$

Final $\theta : [\theta_0, \theta_1] = [0.9966199, 0.0013402]$

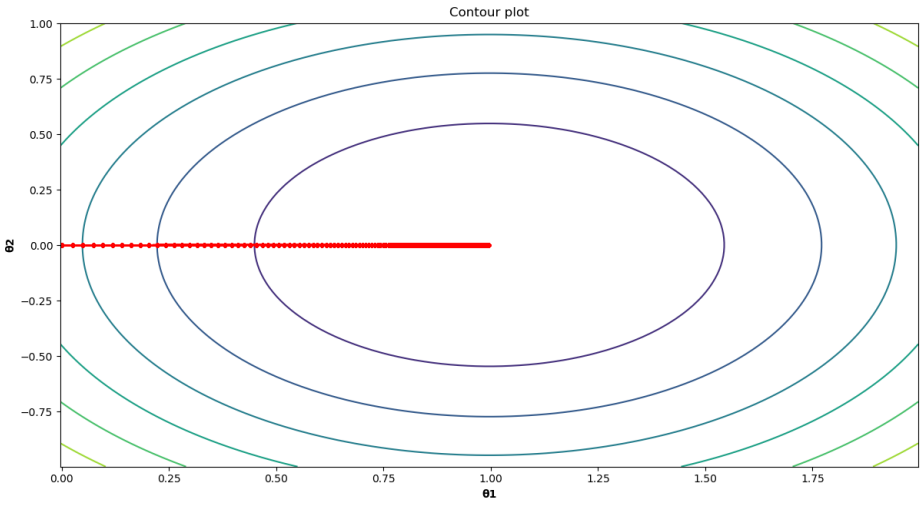


Figure 5: Contour

1.1.5 Parameters

$\eta = 0.1$

Stopping criteria $= |newcost - oldcost| < 10^{-15}$

Number of iterations $= 154$

Final cost $= 1.1947898149897456 * 10^{-6}$

Final $\theta : [\theta_0, \theta_1] = [0.9966199, 0.0013402]$

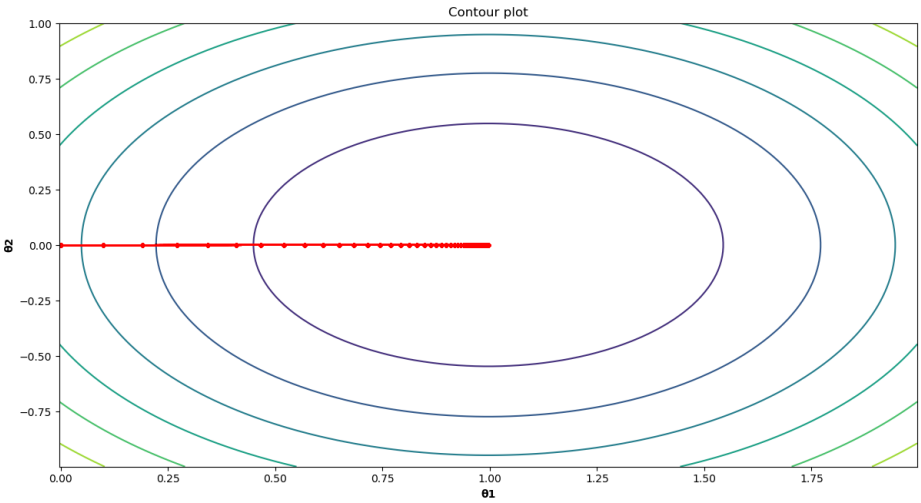


Figure 6: Contour

1.1.6 Comments

With very small learning rate like 0.001 cost converges very slowly

Moderate learning rate like 0.1 to 0.5 cost converges more faster

But if we take high learning rate cost may diverges

Number of iterations to converge is decreases as eta increase from 0.001 to 0.1

Updated theta parameter follow straight descent along the bowl shaped cost curve

2 Stochastic Gradient Descent

$$J(\theta) = \frac{1}{2k} \sum_{k=1}^{r} (y^{(i)} - h_\theta x^{(i)})^2$$

$$\Theta = \begin{bmatrix} 3 \\ 1 \\ 2 \end{bmatrix}$$

$$\eta = 0.001$$

2.1 Batch size = 1

Convergence condition If average of last two 1000 iteration differs by 10^{-5}

Epoch = 1

Iteration = 282000

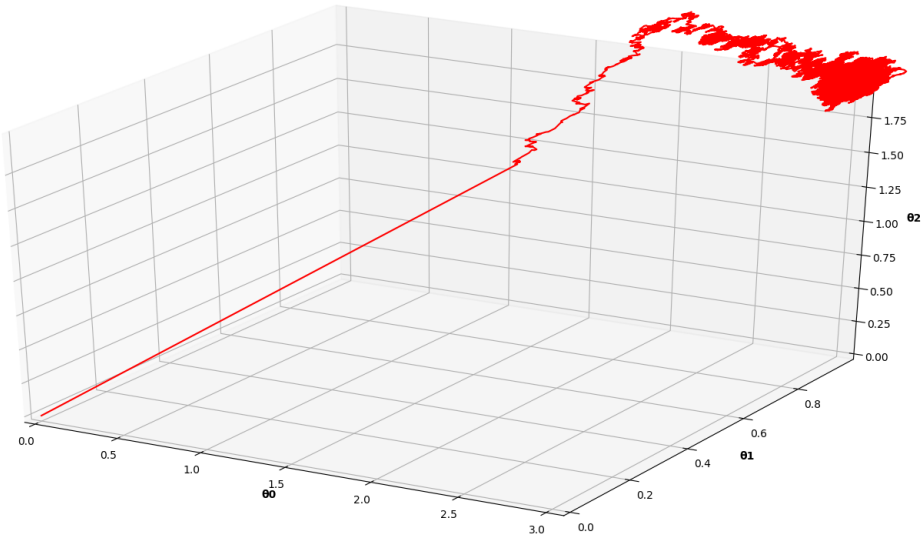
Final $\theta : [\theta_0, \theta_1, \theta_2]$ = [2.94100331, 0.96607746, 1.9814644]

Final cost = 0.10853914002

Time = 4 sec

Cost on test data using learned theta = 1.06566427

Cost on test data using original theta = 0.981464399


θ0
θ1
θ2
0.0
0.5
1.0
1.5
2.0
2.5
3.0
0.0
0.2
0.4
0.6
0.8
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75

2.2 Batch size = 100

Convergence condition If average of last two 1000 iteration differs by 10^{-3}

Epoch = 2

Iteration = 13000

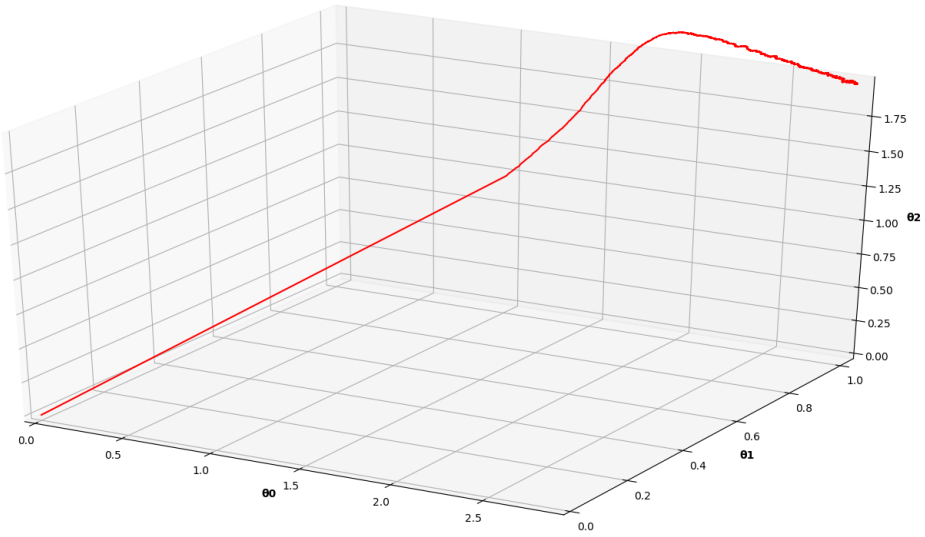
Final θ : $[\theta_0, \theta_1, \theta_2]$ = [2.94263662, 1.0061253, 1.99820773]

Final cost = 0.83679025717

Time = 3 sec

Cost on test data using learned theta = 0.9836719670

Cost on test data using original theta = 0.998207732



2.3 Batch size = 10000

Convergence condition If average of last two 1000 iteration differs by 10^{-6}

Epoch = 251

Iteration = 25000

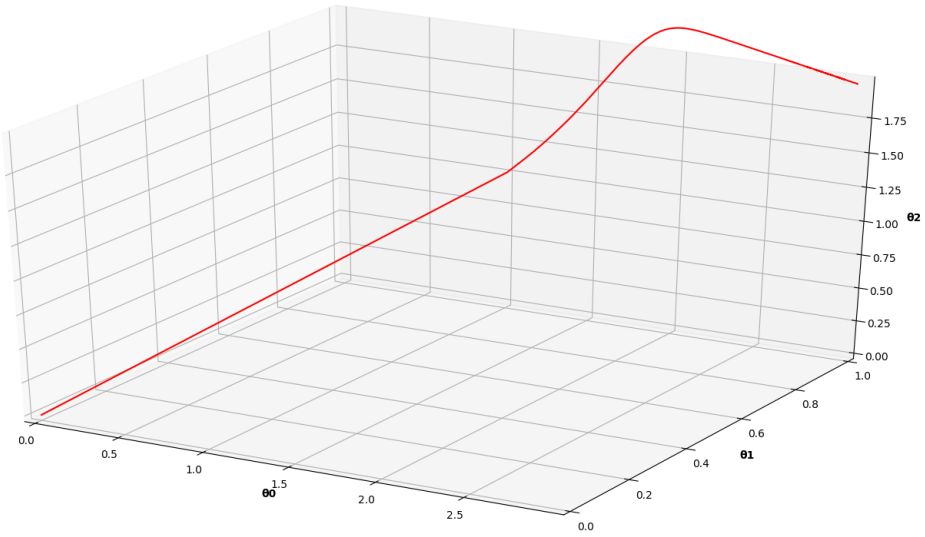
Final $\theta : [\theta_0, \theta_1, \theta_2]$ = [2.9967473, 1.00170161, 1.99974448]

Final cost = 1.0069734840

Time = 25 sec

Cost on test data using learned theta = 0.98304517

Cost on test data using original theta = 0.9829469215



2.4 Batch size = 1000000

Convergence condition If average of last two 1000 iteration differs by 10^{-6}

Epoch = 25001

Iteration = 25000

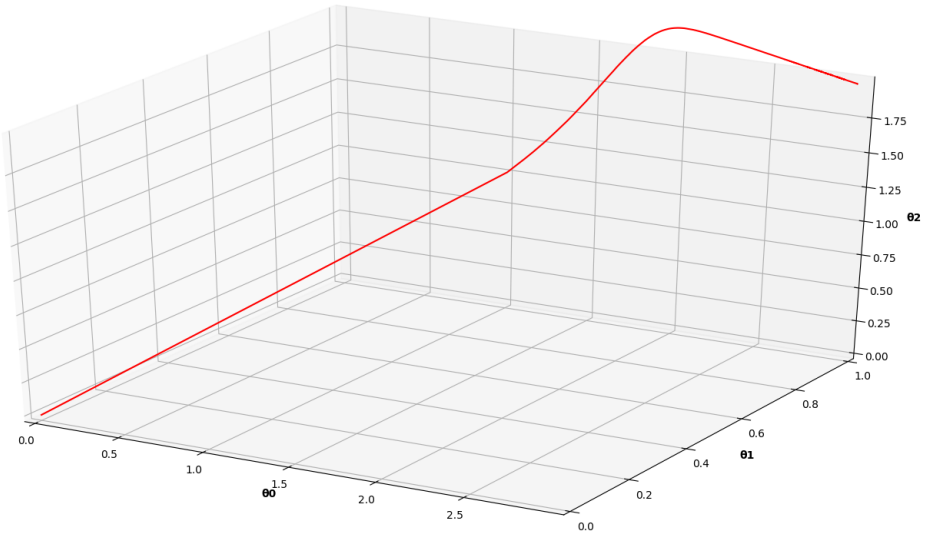
Final $\theta : [\theta_0, \theta_1, \theta_2]$ = [3.0000451231, 1.00037029, 1.9996493242]

Final cost = 1.0020444282030

Time = 399 sec

Cost on test data using learned theta = 0.98298534

Cost on test data using original theta = 0.982946921



2.5 Comments

As the batch size is increased the epoch required is also increased.
Number of iterations required is very large is batch size = 1, but it gradually decreases and then again increases.
SGD with batch size = 1 converges very fast and it gradually increases as the batch size is also increases.

Relative speed of convergence also depend on convergence criteria.
If we have tight convergence criteria, speed slows down at the end.
Convergence time also depend on the data.
If we have biased data convergence may be slow or it may diverges for some batches iterations.

Movement of theta: For smaller batch size, theta moves like zig-zag. The curve becomes more smooth as the batch size is also increased.
At last moment around convergence line has more zig-zag movement. It is intuitive that SGD wanders around optimum theta value.

3 Logistic Regression

$$L(\theta) = \sum_{i=1}^{m} (y^{(i)} log(h_\theta(x^{(i)})) + (1 - y^{(i)})(1 - log(h_\theta(x^{(i)})))$$

Newton's Method Formula:

$$\theta_{n+1} = \theta_n + H_{l(\theta)}^{-1} \nabla l(\theta)$$

where

$$\nabla l(\theta) = x^T (y - h_\theta(x))$$

$$H = -x^T diag[\sigma(1 - \sigma)]x$$

3.1 Parameters and plots

Number of iteration = 9

Stopping criteria = $|\theta_{t+1} - \theta_t| < 10^{-15}$

Final cost = $1.1947898149897456 * 10^{-6}$

Final $\theta : [\theta_0, \theta_1, \theta_1]$ = [0.40125316, 2.5885477, -2.72558849]

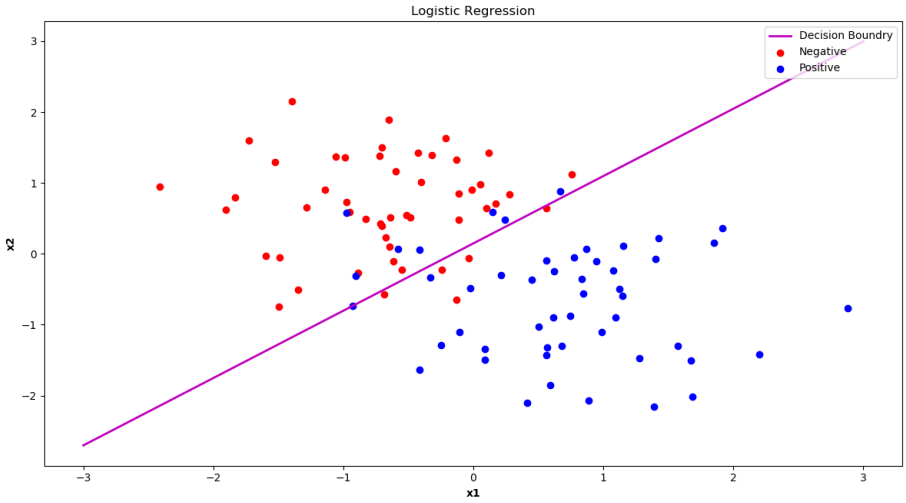


Figure 11: Logistic Regression

4 Gaussian Discriminant Analysis

4.1 Parameters (same co-variance matrix)

$$\phi = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

$$\sigma = \frac{\sum_{i=1}^{m} (x^{(i)} - \mu_{y^{(i)}})(x^{(i)} - \mu_{y^{(i)}})^T}{m}$$

We get the following parameters

$$\phi = 0.5$$

$$\mu_0 = \begin{bmatrix} -0.75529433 \\ 0.68509431 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 0.75529433 \\ -0.68509431 \end{bmatrix}$$

$$\sigma = \begin{bmatrix} 0.42953048 & -0.02247228 \\ -0.02247228 & 0.53064579 \end{bmatrix}$$

4.2 Scatter Plot

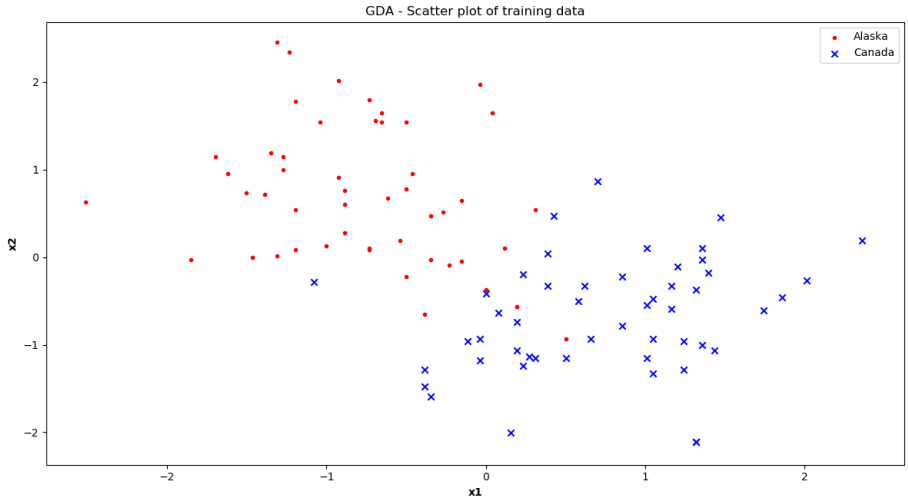


Figure 12: Training Data

4.3 Linear Decision Boundary

Linear decision boundary can be expressed as

$$ax_1 + bx_2 + c = 0$$

or

$$AX + B = 0$$

.

Where

$$A = 2(\mu_0 - \mu_1)^T \Sigma^{-1}$$

$$B = \mu_1^T \Sigma^{-1} \mu_1 - \mu_0^T \Sigma^{-1} \mu_0$$

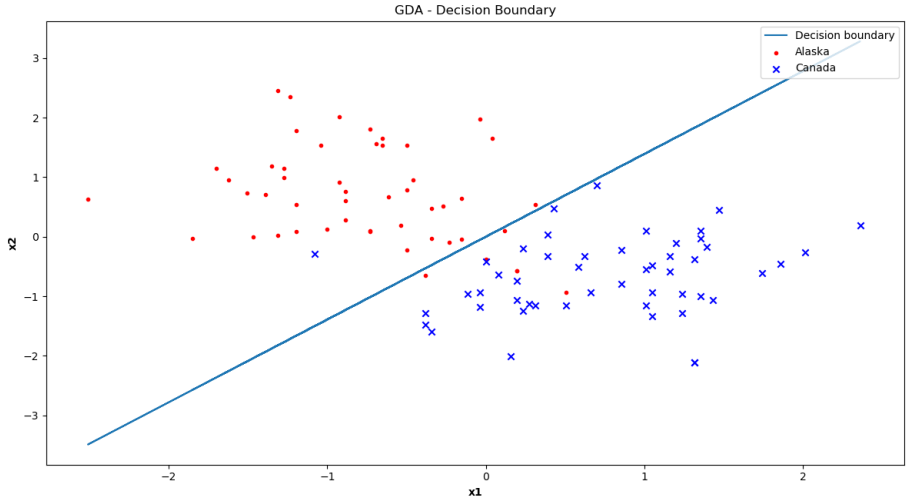


Figure 13: Linear Boundary

4.4 Quadratic Decision Boundary

$$\phi = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}$$

$$\mu_0 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}}$$

$$\mu_1 = \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}$$

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

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

We get the following parameters

$$\phi = 0.5$$

$$\mu_0 = \begin{bmatrix} -0.75529433 \\ 0.68509431 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 0.75529433 \\ -0.68509431 \end{bmatrix}$$

$$\Sigma_0 = \begin{bmatrix} 0.38158978 & -0.15486516 \\ -0.15486516 & 0.64773717 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 0.47747117 & 0.1099206 \\ 0.1099206 & 0.41355441 \end{bmatrix}$$

Quadratic decision boundary can be expressed as

$$X^T AX + BX + C = 0$$

.

where

$$A = \Sigma_0^{-1} - \Sigma_1^{-1}$$

$$B = -2(\mu_0^T \Sigma_0^{-1} - \mu_1^T \Sigma_1^{-1})$$

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

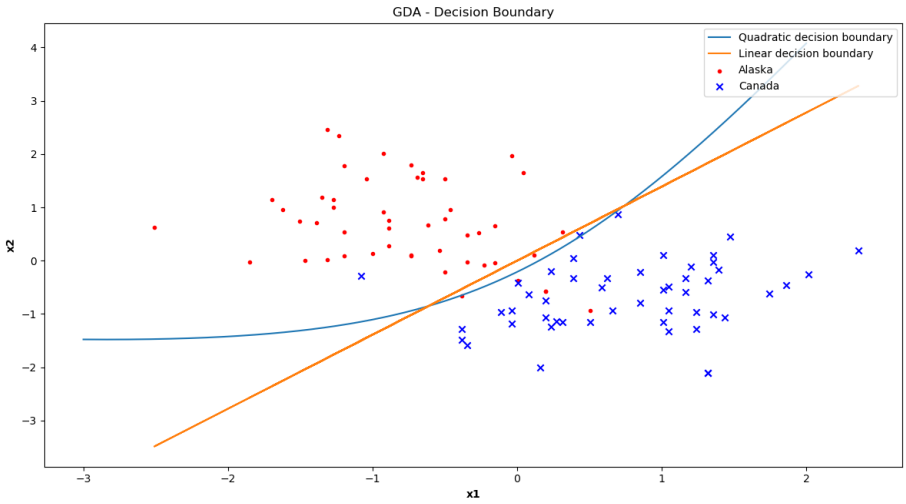


Figure 14: Quadratic Boundary

4.5 Comments

We can see that quadratic boundary separates some points very well in comparisons to linear boundary.

As we can see that there is only one blue point that quadratic boundary can't cover, so this boundary may over-fit.

Also we can see that linear also separates data very well and it didn't under-fit.

So both are good boundary for given data.