SCHOOL OF COMPUTING, UNIVERSITY OF UTAH

Machine Learning CS-6350, Assignment - 3
Due: 08^{th} October 2013

Chandramouli, Shridharan
sdharan@cs.utah.edu
(00873255)

Singla, Sumedha
sumedha.singla@utah.edu
(00877456)

October 8, 2013

INTRODUCTION

In this assignment, we are asked to work on and implement binary classification using the logistic regression, and the gaussian disriminant analysis, which is a specialized form of the binary classification. The first part of the assignment dealas with the logistic regression, whereas the second and the third parts of this assignment deals with the gaussian discriminant analysis and its variations interms of decision boundaries.

QUESTION 1 : LOGISTIC REGRESSION

In this section, we use the logistic regression to determine the decision boundary between the two classes of data points. We initially implement a classifier with a linear decision boundary, and then extend it to a conic section. We also present the efficiency of the different types of boundaries in correctly classifying the data points.

Question: Write a MATLAB function that reads in and plots the data points in a 2-D image (a= 2 in this example) with different markers for data points for which $y = 0$ and for data points for which $y = 1$. Based on the plot, what shape should the decision boundary have between data labeled 0 and data labeled 1?

Answer: We ploted the data points in matlab using the plot function. The positive data points (with $y_i = 1$) are marked as + where as the negative data points (with $y_i = 0$) are marked as o. The plot is shown below in Figure 0.1

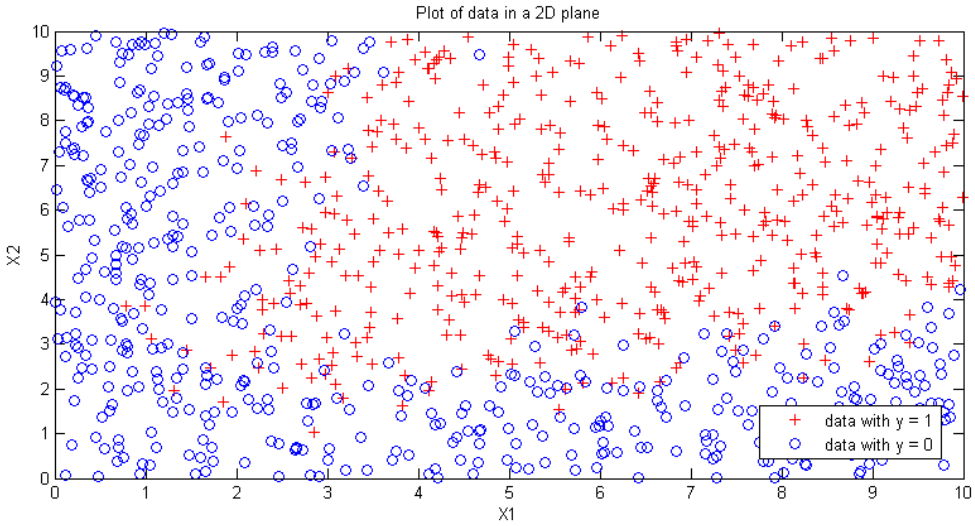

Figure 0.1: Plot of the data on a 2-D plane

Question: Based on the plot of the previous question and your conclusion regarding the shape of the decision boundary, what should the feature vector $\hat{x}$ look like?

Answer: From the figure 0.1, it is clear that a good classification should have a decision boundary which looks like a parabola encompassing the data points with $y = 1$.

The feature vector $\hat{x}$ therefore looked like below.

$$\hat{x} = \begin{pmatrix} x_1^2 \\ x_2^2 \\ x_1 x_2 \\ x_1 \\ x_2 \\ 1 \end{pmatrix}$$

We conducted experiments with different decision boundary shapes (line, ellipse and parabola) to confirm the effectiveness of our hypothesis that the decision boundary needs to be a parabola.

Question: Choose an initial guess for the values of θ and a value for the learning parameter α, and use gradient ascent on the entire corpus of data to find a (local) maximum-likelihood estimate for θ. Upon convergence, plot the decision boundary together with the data. Create a second plot showing the likelihood as a function of the number of iterations. Play around with the initial values of θ and the value of α and discuss its effect on the results in your report.

We wrote a Matlab script which updates the value of θ using the entire corpus of data until convergence. We choose chose an initial θ value as

$$\theta = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Upon convergence, we got a value of

$$\theta = \begin{pmatrix} -0.1550 \\ -0.1541 \\ 0.5552 \\ -0.1930 \\ -0.2097 \\ -0.1827 \end{pmatrix}$$

.

We plotted the parabola with the above coefficients to get the decision boundary as shown in figure 0.2

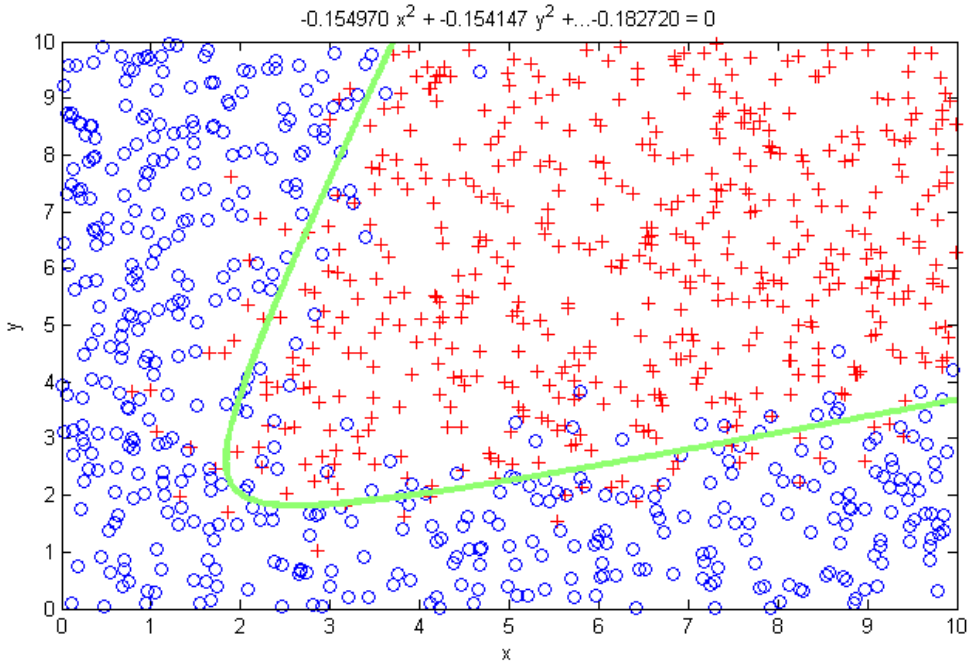

Figure 0.2: Plot of the original data on a 2-D plane with decision boundary.

The plot of the log likelihood against the number of iterations is shown below in figure 0.10

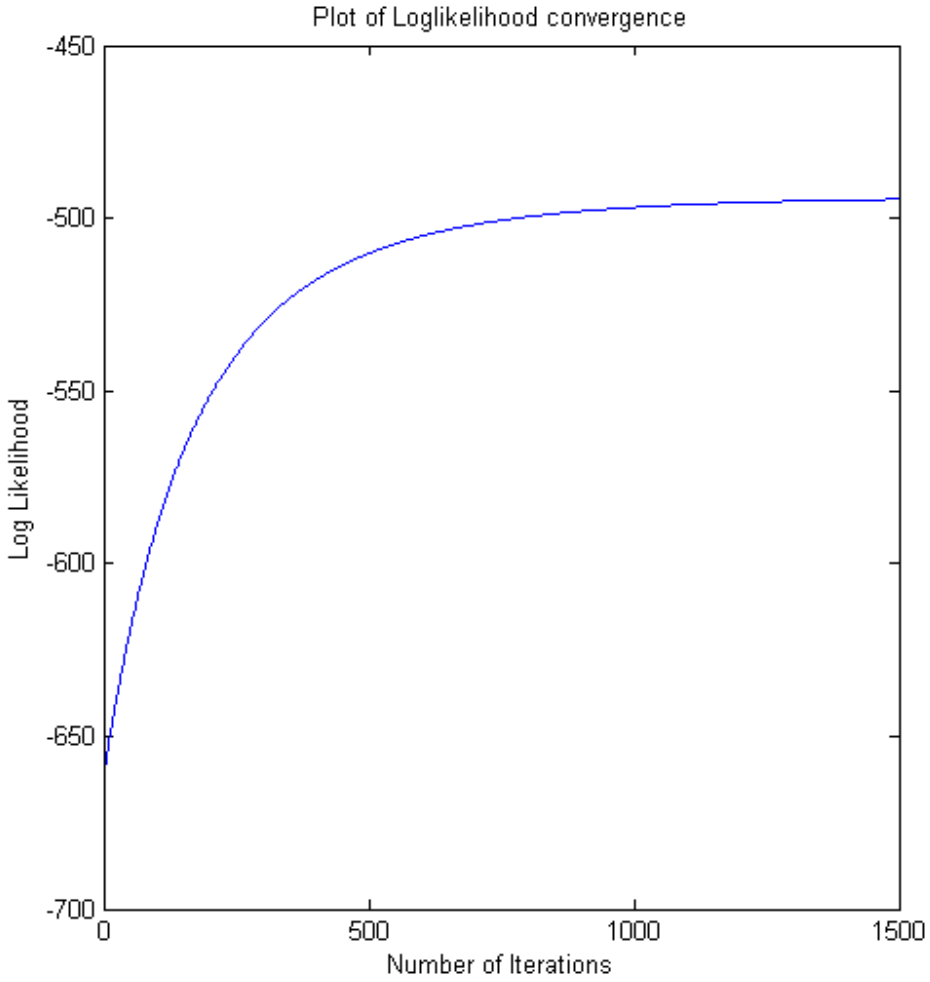

Figure 0.3: Plot of log likelihood against number of iterations

For the above curve, we used an alpha value of 0.0001. On increasing the value of alpha, we found that the log-likelihood function does not converge, and often oscillates for increasing number of iterations. A plot of this is shown below in figure 0.4

In order to experiment with the θ values, we set the initial value of θ to be a random number generated as $theta2 = rand(a,1)$. For different executions, we were able to see subtle to significant changes in the shape of the decision boundary. For example, figure 0.5 and 0.6 show the decision boundary for different random initial values.

Question : Given the learned values of θ, create an isocontour plot of the probability function $f(x)$ over the 2-D plane. What does the "distance" between the iso-contours signify?

We wrote a matlab function called plot_contour.m which automatically plots the contour of

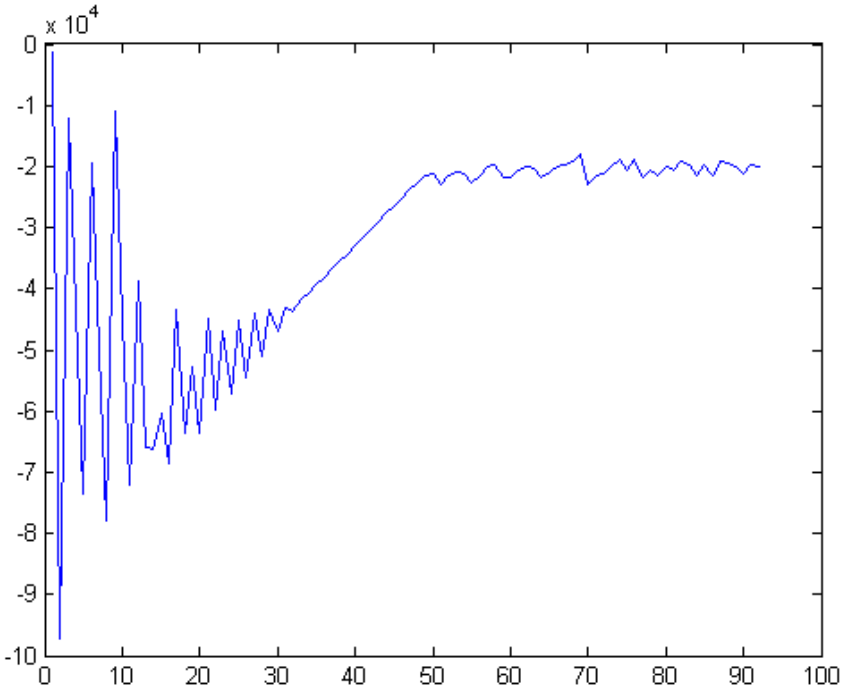

Figure 0.4: Plot of oscillating log likelihood against number of iterations for a big alpha = 2

the 2D plane for $f(x)$ given the value of θ. The results of the function on the above mentioned value of theta is shown below.

The distance between the iso-contours is in a way a measure of the certainty of the decision boundary. When the contour lines are close together, it signifies that there is a well defined boundary separating the two classes of data, as it signifies that there is a rapid increase to the maximum of the sigmoid curve from the center. On the other hand, a spread out contours is indicative of the larger areas of curve where the probability does not swing either way, which could be seen as uncertainty near the decision boundary.

Question :Choose an initial guess for the values of θ, and use online gradient ascent, where θ is updated with each data point that is processed. The value of α should decrease with each data point added, for instance proportional to the reciprocal of the number of data points already processed. After all data has been processed, plot the decision boundary together with the data. Play around with the initial values of θ and the definition of α, and describe your results.

We wrote a matlab script (ques1.m) which does the online gradient ascent. We started with

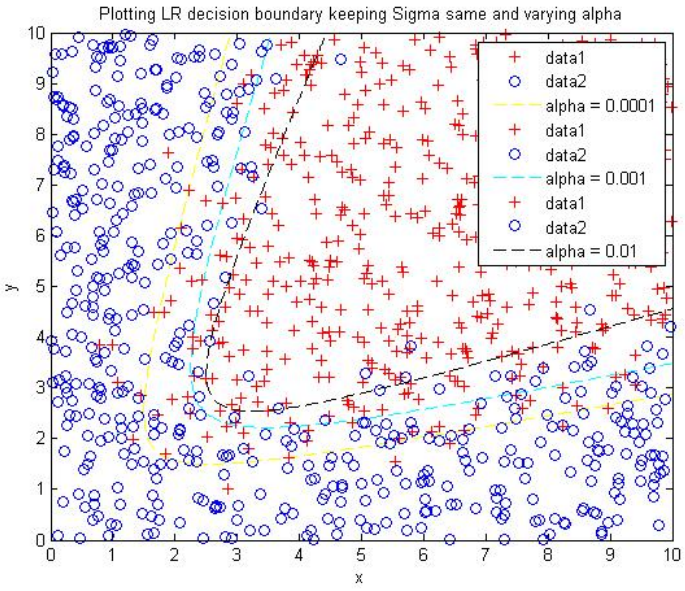

Figure 0.5: Plot of decision boundary varying alpha for same θ

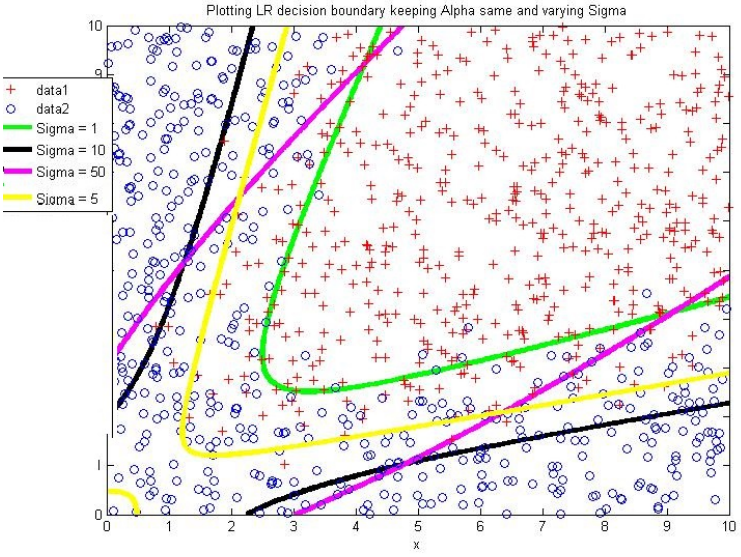

Figure 0.6: Plot of decision boundary varying θ for same alpha

a initial θ as a $\vec{0}$ and initial $alpha = 0.001$ and updated the value of theta for every data point in the input. The resulting decision boundary is shown below in figure 0.8. At each iteration, we incremented the value of alpha proportional to the inverse of the number of data points processed as, $\alpha = \alpha + \frac{K}{numberofdatapoints}$. The value of K was initially chosen to be $\frac{1}{1000}$.

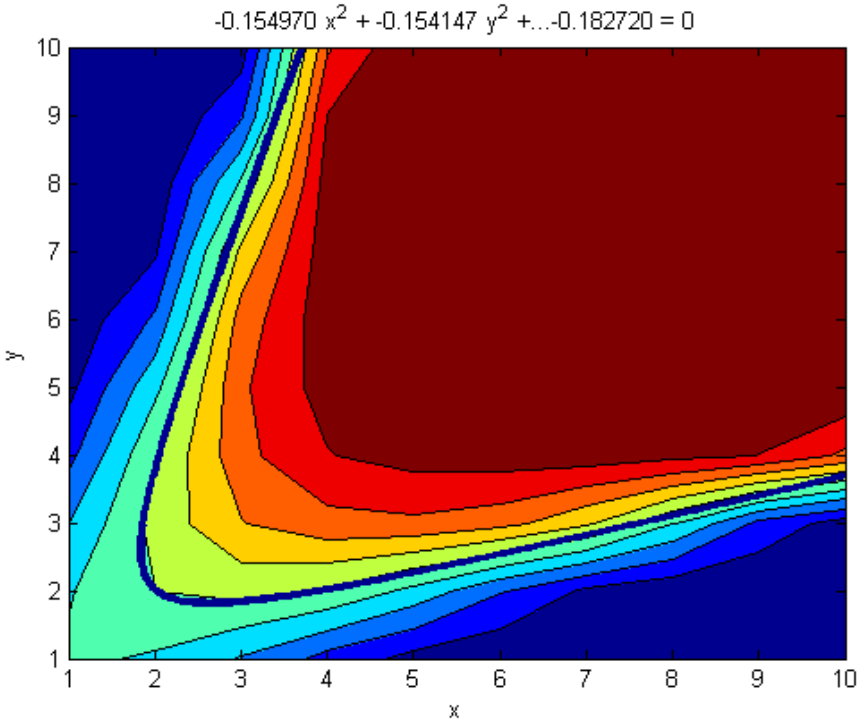

Figure 0.7: Plot of log likelihood against number of iterations

We repeated the same, with a K = 0.1 and got the curve shown below in figure 0.9.

As it could be seen from the figures, having a K value of 0.1 provides a better classification. We repeated the experiments with higher values of K=1,2,3..etc and found the $K = 0.1$ provided optimal results for the initial chosen α and θ

QUESTION 2: GAUSSIAN DISCRIMINANT ANALYSIS

In Gaussian Discriminant Analysis we try to learn the probability distributions $p(y)$ and $p(X|y) = p(\hat{X}|y)$ assuming they follow the model:

$$\begin{aligned} y &\approx Bernoulli(\phi), \\ \hat{X}|y &\sim N(\mu_y, \Sigma). \end{aligned} \tag{0.1}$$

Here, $\phi \in \Re, \mu_0, \mu_1 \in \Re^{dim(\hat{X})}, and \Sigma \in \Re^{dim(\hat{X})*dim(\hat{X})}$ are the parameters of the model. Here, $\hat{X}$ is a feature vector whose elements are functions of the raw data elements of x.

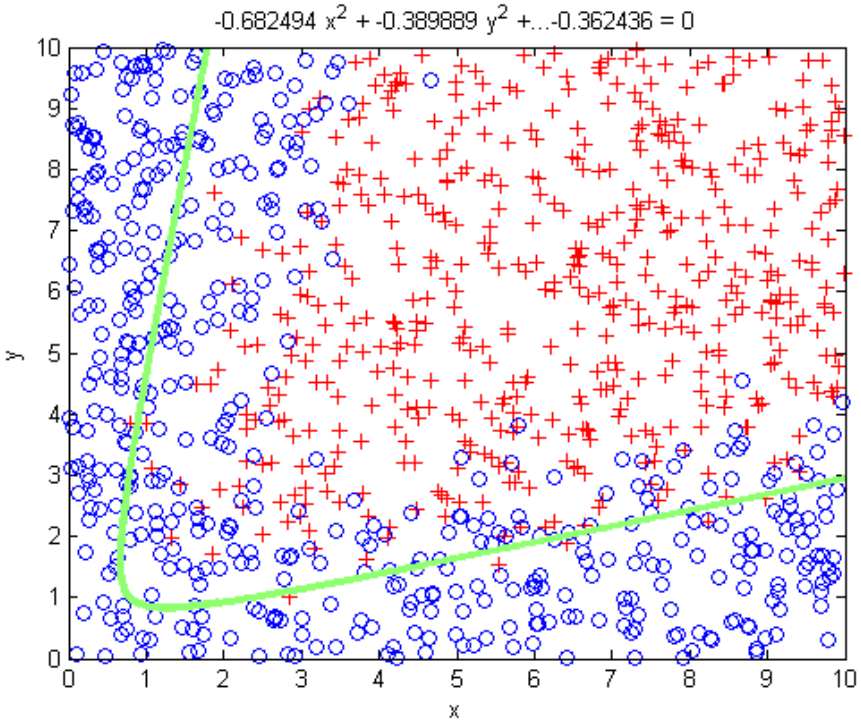

Figure 0.8: Plot of decision boundary $K = 0.001$, $\theta = \vec{0}$ along with the original data.

Question :(a) In the design of the feature vector in the context of logistic regression you probably chose to have the number 1 as one of your features. For Gaussian discriminant analysis, you can remove this feature from the feature vector. Why is that?

Answer: In the case of the logistic regression, the constant 1 was required as a part of the feature vector to provide some flexibility to the decision boundary. Since we are trying to define a decision boundary in terms of the given data using MLE, without the 1 in the feature vector, we would be forcing the decision boundary to be passing through the origin. On the other hand, in the case of the Gaussian Discriminant Analysis, the decision boundary is defined as the plane where the probabilities of the two normal distributions are equal. The probability is in turn defined by the parameters $\phi, \mu_0 \mu_1 and \Sigma$. In this scenario, the terms $log(1-\phi) - log(\phi) - \frac{1}{2}\mu_0^T \Sigma^{-1} \mu_0 + \frac{1}{2}\mu_1^T \Sigma^{-1} \mu_1$ only shifts the decision boundary parallel to the original boundary.

Question :(b) Derivation for the maximum-likelihood estimates of $\Sigma, \mu_0, \mu_1 and \phi$

By using the Gaussion Discriminant Analysis method we are trying to learn $P(y)$ and $p(X|y) = p(\hat{X}|y)$ where $\hat{X} = f(x)$

Assume

$$y \sim Bernoulli(\phi)$$

$$\implies p(y=1) = \phi and p(y=0) = 1-\phi$$

$$p(y) = \phi^y * (1-\phi)^{1-y}$$

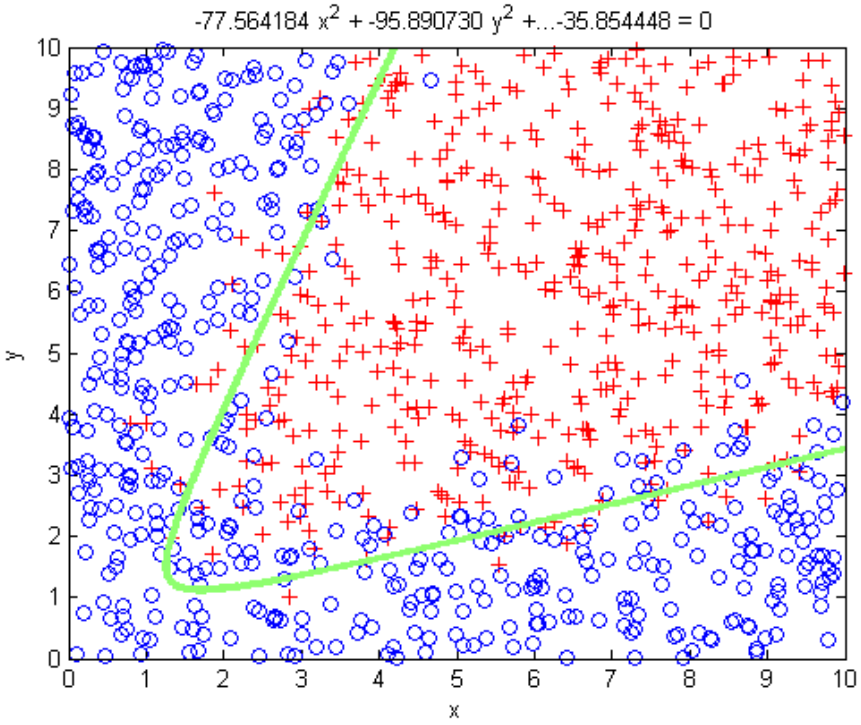

Figure 0.9: Plot of decision boundary with $K = 0.1$, $\theta = \vec{0}$ along with the original data.

$$\hat{X}|y \sim N(\mu_y, \Sigma)$$

$$p(\hat{X}|y=1) \sim N(\mu_1, \Sigma)$$

$$p(\hat{X}|y=0) \sim N(\mu_0, \Sigma)$$

$$p(\hat{X}|y=1) = \frac{1}{\sqrt{(2\pi)^a|\Sigma|}} e^{\frac{-1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)}$$

$$p(\hat{X}|y=0) = \frac{1}{\sqrt{(2\pi)^a|\Sigma|}} e^{\frac{-1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)}$$

$$\implies p(X|y) = [p(X|y=1)]^y [p(X|y=0)]^{1-y}$$

where

$$dim(\hat{X}) = a\mu_0, \mu_1 \in \Re^{a*a}\Sigma \in \Re^{a*a}$$

$$l(\phi, \mu_0, \mu_1, \Sigma) = \prod_{i=1}^{n} p(x_i|y_i)p(y_i)$$

Taking log on both sides

$$ll(\phi, \mu_0, \mu_1, \Sigma) = \sum_{i=1}^{n} [log(p(x_i|y_i))] + [log(p(y_i)]$$

$$= \sum_{i=1}^{n} log(p(x_i|y_i=1))^{y_i} + log(p(x_i|y_i=0)^{1-y_i}) + log(\phi^{y_i}) + log(1-\phi)^{1-y_i}$$

$$= \sum_{i=1}^{n} y_i \left(log \frac{1}{\sqrt{(2\pi)^a |\Sigma|}} - \frac{1}{2}(x_i-\mu_1)^T \Sigma^{-1}(x_i-\mu_1) \right) + (1-y_i) \left(log \frac{1}{\sqrt{(2\pi)^a |\Sigma|}} - \frac{1}{2}(x_i-\mu_0)^T \Sigma^{-1}(x_i-\mu_0) \right)$$

$$+ y_i log(\phi) + (1-y_i) log(1-\phi)$$

$$= \frac{-y_i}{2}(x_i-\mu_1)^T \Sigma^{-1}(x_i-\mu_1) - \frac{1-y_i}{2}(x_i-\mu_0)^T \Sigma^{-1}(x_i-\mu_0) + log \frac{1}{\sqrt{(2\pi)^a |\Sigma|}}$$

$$+ y_i log(\phi) + (1-y_i) log(1-\phi)$$

Taking derivative w.r.t ϕ to find the maximum-likelihood estimates of ϕ

$$\frac{\partial ll(\phi, \mu_0, \mu_1, \Sigma)}{\partial \phi} = \sum_{i=1}^{n} \frac{y_i}{\phi} + \frac{(1-y_i)(-1)}{1-\phi}$$

Equating with zero

$$\sum_{i=1}^{n} \frac{y_i}{\phi} = \sum_{i=1}^{n} \frac{1-y_i}{1-\phi}$$

$$\phi = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Taking derivative w.r.t μ_0 to find the maximum-likelihood estimates of μ_0

$$\frac{\partial ll(\phi, \mu_0, \mu_1, \Sigma)}{\partial \mu_0} = \frac{\partial}{\partial \mu_0} \left(\sum_{i=1}^{n} -\frac{1-y_i}{2}(x_i^T \Sigma^{-1} x_i - 2\mu_0^T \Sigma^{-1} x_i + \mu_0^T \Sigma^{-1} \mu_0) \right)$$

$$= \sum_{i=1}^{n} -\frac{1-y_i}{2}(-2\Sigma^{-1} x_i + 2\Sigma^{-1} \mu_0)$$

Equating with zero

$$\sum_{i=1}^{n} (1-y_i)(\Sigma^{-1} x_i) = \sum_{i=1}^{n} (1-y_i)(\Sigma^{-1} \mu_0)$$

$$\mu_0 = \sum_{i=1}^{n} \frac{(1-y_i)x_i}{1-y_i}$$

Rewriting

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

Taking derivative w.r.t μ_1 to find the maximum-likelihood estimates of μ_1

$$\frac{\partial ll(\phi, \mu_0, \mu_1, \Sigma)}{\partial \mu_1} = \frac{\partial}{\partial \mu_1} \left(\sum_{i=1}^{n} -\frac{y_i}{2}(x_i^T \Sigma^{-1} x_i - 2\mu_1^T \Sigma^{-1} x_i + \mu_1^T \Sigma^{-1} \mu_1) \right)$$

$$= \sum_{i=1}^{n} -\frac{y_i}{2}(-2\Sigma^{-1} x_i + 2\Sigma^{-1} \mu_1)$$

Equating with zero

$$\sum_{i=1}^{n}(y_i)(\Sigma^{-1}x_i) = \sum_{i=1}^{n}(y_i)(\Sigma^{-1}\mu_1)$$

$$\mu_1 = \sum_{i=1}^{n}\frac{y_i x_i}{y_i}$$

Rewriting

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

Taking derivative w.r.t Σ to find the maximum-likelihood estimates of Σ

$$\frac{\partial ll(\phi,\mu_0,\mu_1,\Sigma)}{\partial\Sigma} = \frac{\partial}{\partial\Sigma}\sum_{i=1}^{n}\left(\frac{-y_i}{2}(x_i-\mu_1)^T\Sigma^{-1}(x_i-\mu_1) - \frac{1-y_i}{2}(x_i-\mu_0)^T\Sigma^{-1}(x_i-\mu_0) + log\frac{1}{\sqrt{(2\pi)^a|\Sigma|}}\right)$$

using identity

$$\frac{\partial(A^TXA)}{\partial X} = AA^T$$

$$\frac{\partial|Y|}{\partial x} = |Y|Tr\left[Y^{-1}\frac{\partial Y}{\partial X}\right]$$

$$\frac{\partial ll(\phi,\mu_0,\mu_1,\Sigma)}{\partial\Sigma} = \sum_{i=1}^{n}\left[\frac{-y_i}{2}(x_i-\mu_1)(x_i-\mu_1)^T - \frac{1-y_i}{2}(x_i-\mu_0)(x_i-\mu_0)^T + \frac{\Sigma}{2}\right]$$

Equating with zero

$$\Sigma = \frac{1}{n}\sum_{i=1}^{n}(1-y_i)(x_i-\mu_0)(x_i-\mu_0)^T + y_i(x_i-\mu_1)(x_i-\mu_i)^T$$

Rewriting

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

Answere :(c) Values of ϕ,μ_0,μ_1,Σ after solving for given data are:

$$\phi = 0.5170$$

$$\mu_0 = \begin{bmatrix} 22.5149 \\ 24.2539 \\ 9.1487 \\ 3.5942 \\ 3.8624 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 43.4830 \\ 40.5752 \\ 38.2702 \\ 6.1987 \\ 5.9750 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 826.7667 & -112.8146 & 283.5746 & 75.0158 & -12.8738 \\ -112.8146 & 808.8131 & 257.9116 & -14.1155 & 73.2031 \\ 283.5746 & 257.9116 & 265.4717 & 24.9236 & 22.7347 \\ 75.0158 & -14.1155 & 24.9236 & 7.2507 & -1.6489 \\ -12.8738 & 73.2031 & 22.7347 & -1.6489 & 7.0293 \end{bmatrix}$$

Decision boundary for the Gaussian discriminant model is given implicitly by the equation:

$$p(x|y=0)p(y=0) = p(x|y=1)p()y = 1$$

Solving for this decision boundary we get $\theta^T X = 0$ where

$$\theta = \begin{bmatrix} -0.2519 \\ -0.2533 \\ 0.2142 \\ 2.2624 \\ 2.3150 \\ -10.9519 \end{bmatrix}$$

Plot for the decision boundary in 2-D plane is shown in figure 0.10

The difference between the decision boundary resulting from Gaussian discriminant analysis and logistic regression is shown in figure 0.11

Which one is better in terms of the number of data points correctly classified

To find which classification is better we have counted the number of points for which $\theta^T X > 0$ and $\theta^T X < 0$ and compare them with the original data. Here $\theta^T X > 0 \sim y_{(i)} = 1$ and $\theta^T X < 0 \sim y_{(i)} = 0$

For given data number of data points with

$$\sum_{i=1}^{n} \left[y_{(i)} = 1 \right] = 517$$

$$\sum_{i=1}^{n} \left[y_{(i)} = 0 \right] = 483$$

For Logistic Regression

$$\sum_{i=1}^{n} \left[\theta^T X >= 0 \right] = 523$$

$$\sum_{i=1}^{n} \left[\theta^T X < 0 \right] = 477 \sim 98.75\% sucess$$

For Gaussian Discriminant Analysis

$$\sum_{i=1}^{n} \left[\theta^T X >= 0 \right] = 533$$

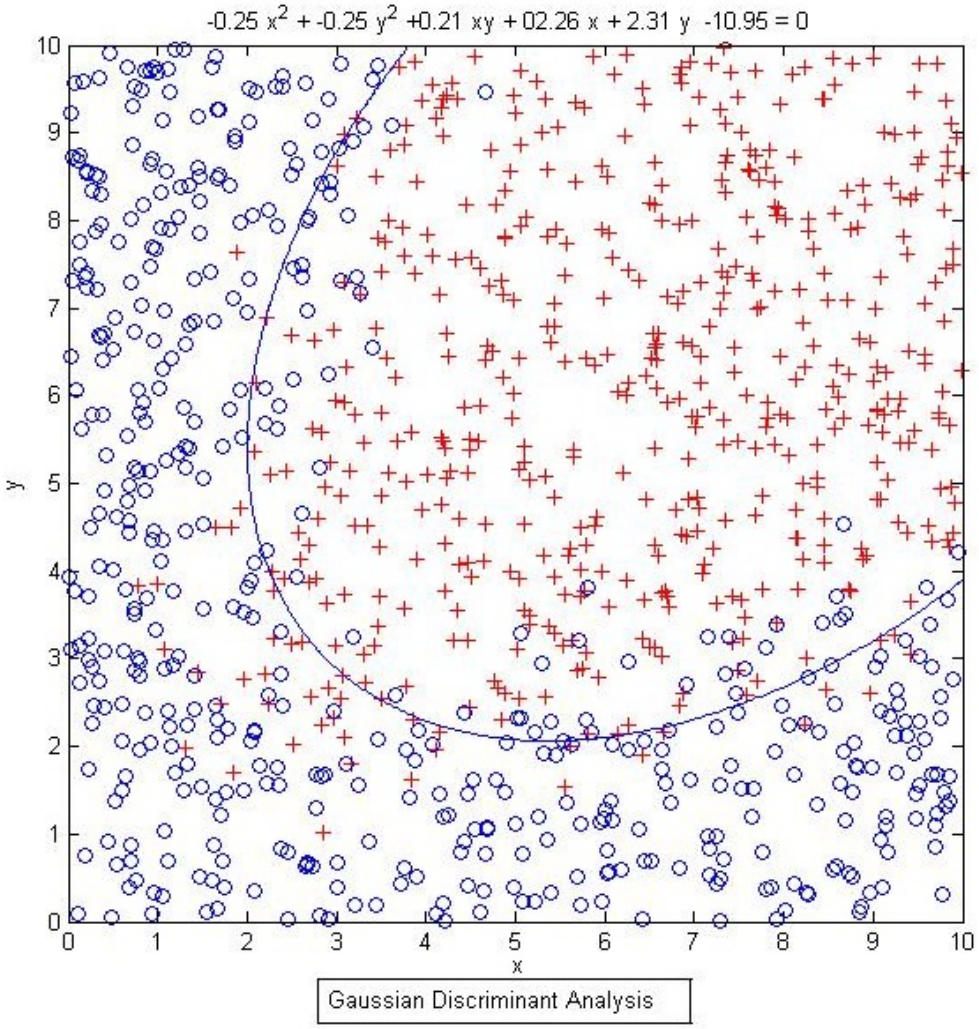

Figure 0.10: Plot of decision boundary from GDA

$$\sum_{i=1}^{n} [\theta^T X < 0] = 467 \sim 96.68\% success$$

As per this analysis, Logistic Regression is more accurate.

QUESTION 3: GAUSSIAN DISCRIMINANT ANALYSIS

a) Let the feature vector be the same as the raw data, i.e. $\hat{x} = x$. Find maximum-likelihood values for ϕ, μ_0, mu_1 and Σ in this case and plot the decision boundary in the 2-D plane. Why is the resulting classifier crappy?

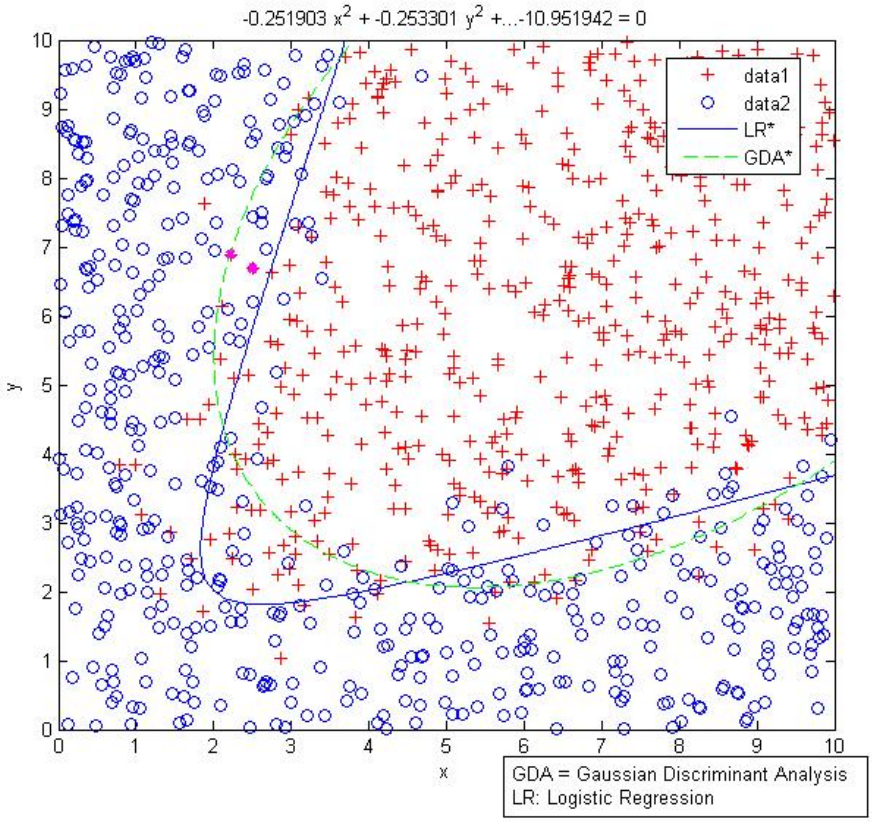

Figure 0.11: Difference in decision boundary: comparing Logistic Regression versus GDA

For this question, we determined the values of the model parameters by using $\hat{x} = (x_1 x_2)^T$ and by using $\hat{x} = (x_1 x_2 1)^T$. The resulting parameters for first part (without the constant term in the feature vector is shown below)

$$\mu_0 = \begin{bmatrix} 3.5942 \\ 3.8624 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 6.1987 \\ 5.9750 \end{bmatrix}$$

$$\Sigma = \begin{bmatrix} 7.2507 & -1.6489 \\ -1.6489 & 7.0293 \end{bmatrix}$$

The resulting plot of the decision boundary is shown in figure 0.12

As it could be seen the resulting classifier fails to provide even a decent classifier. This is because, without the constant 1 in the feature vector, there decision boundary is not shifted to the position which divides the two classes ($y = 1 \& y = 0$) equally.

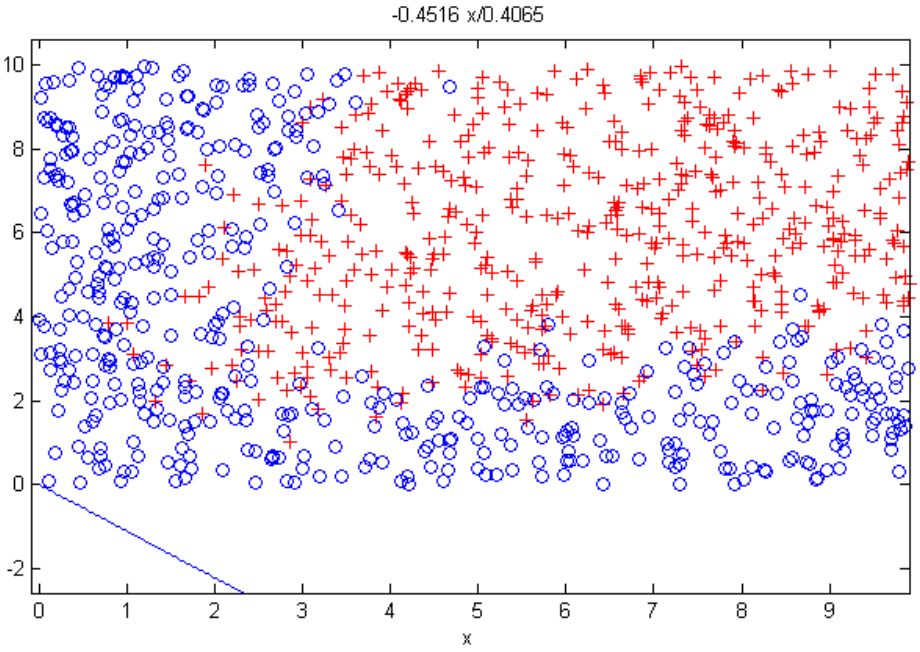

Figure 0.12: Plot of the decision boundary when using X as feature vector

On introducing the constant 1 term in the feature vector, we find that the classifier provides a decent classifying boundary for the two data sets as shown in figure 0.13. This result is still not as efficient in correctly defining the decision boundary as using a quadratic boundary.

In this case the update probability equations are:

$$p(\hat{X}|y=1) = \frac{1}{\sqrt{(2\pi)^a|\Sigma_1|}} e^{\frac{-1}{2}(x-\mu_1)^T\Sigma_1^{-1}(x-\mu_1)}$$

$$p(\hat{X}|y=0) = \frac{1}{\sqrt{(2\pi)^a|\Sigma_0|}} e^{\frac{-1}{2}(x-\mu_0)^T\Sigma_0^{-1}(x-\mu_0)}$$

Calculating for maximum-likelihood values for Σ_0 and Σ_1

$$\Sigma_1 = \frac{\sum_{i=1}^{n}(y_i)(x_i-\mu_1)(x_i-\mu_1)^T}{\sum_{i=1}^{n}(y_i)}$$

rewriting

$$\Sigma_1 = \frac{\sum_{i=1}^{n}\left(1\{y^{(i)}=1\}\right)(x_i-\mu_1)(x_i-\mu_1)^T}{\sum_{i=1}^{n}1\{y^{(i)}=1\}}$$

$$\Sigma_0 = \frac{\sum_{i=1}^{n}(1-y_i)(x_i-\mu_0)(x_i-\mu_0)^T}{\sum_{i=1}^{n}(1-y_i)}$$

rewriting

$$\Sigma_0 = \frac{\sum_{i=1}^{n}\left(1\{y^{(i)}=0\}\right)(x_i-\mu_0)(x_i-\mu_0)^T}{\sum_{i=1}^{n}1\{y^{(i)}=0\}}$$

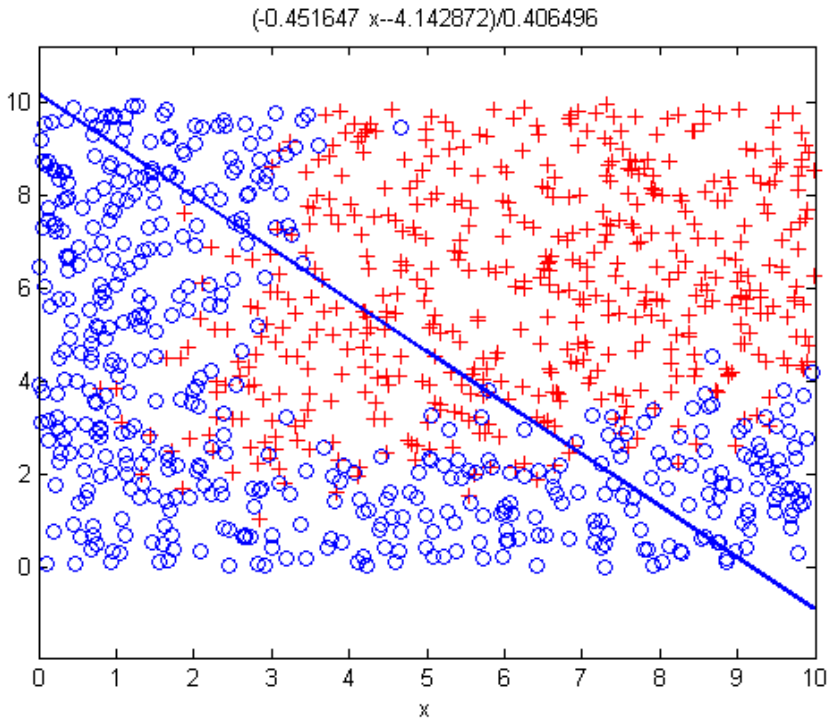

Figure 0.13: Plot of the decision boundary when using $[X1]^T$ as feature vector

Compute maximum-likelihood values for ϕ,μ_0, μ_1,Σ_1,Σ_0.Plot the decision boundary. Does this give a better classifier ?

For the QDA, I used a feature vector as $(x1, x2)^T$ and calculated the values of ϕ,μ_0, μ_1,Σ_1,Σ_0 using the above equations.
The values for the parameters are listed below.

$$\mu_0 = \begin{bmatrix} 3.5942 \\ 3.8624 \end{bmatrix}$$

$$\mu_1 = \begin{bmatrix} 6.1987 \\ 5.9750 \end{bmatrix}$$

$$\Sigma_1 = \begin{bmatrix} 5.4152 & 1.3195 \\ 1.3195 & 5.2174 \end{bmatrix}$$

$$\Sigma_0 = \begin{bmatrix} 8.9655 & -4.4221 \\ -4.4221 & 8.7220 \end{bmatrix}$$

The plot of the decision boundary with in the input space is shown below in figure 0.14

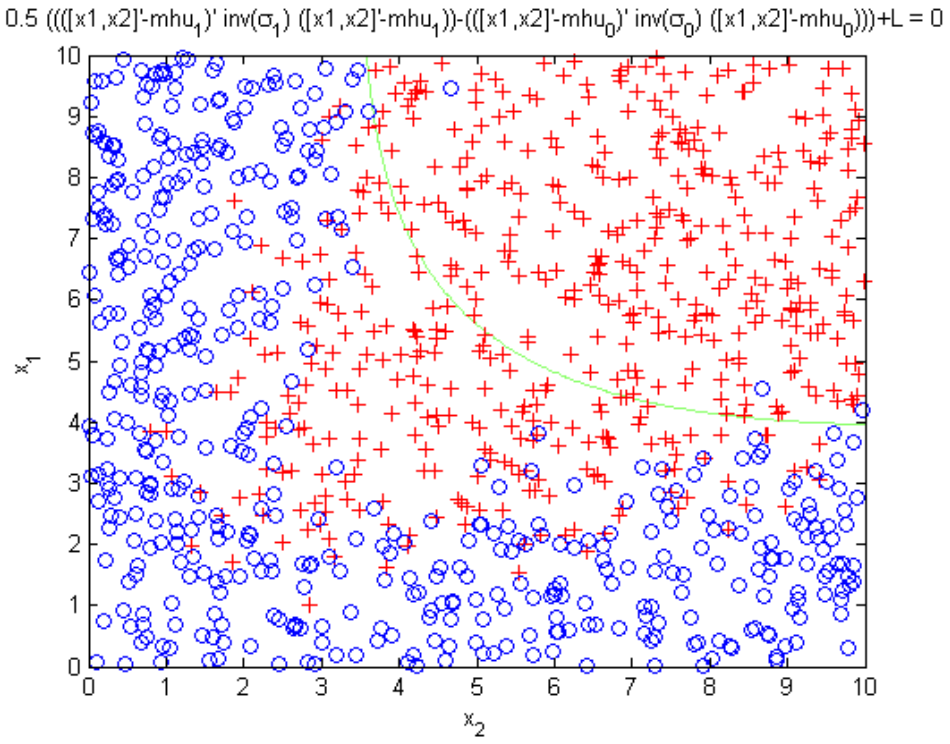

Figure 0.14: Plot of the decision boundary when using $[X1]^T$ as feature vector for Quadratic Discriminant Analysis

As it could clearly be seen from the above image, QDA provides a much better classification when compared to the Linear Discriminant Analysis.

1 Work Split-up

The work was split evenly between the authors. The matlab programs were coded was coded together and the equations for the second andn third question was solved individually and verified.

2 Conclusion

In this assignment, we worked on using class of linear classifiers and applied it to the problem of classifying a binary data set. From the results in the assignment, we can conclude that both the linear logistic regression and the Gaussian Discriminant Analysis methods provide a easy and computationally feasibe way of classifying a set of input vectors and assign each of the data sets a label defining which class it belongs to. The Quadratic Gaussian Discriminant

Analysis offers a non linear way of classifying data, and provides a better estimate of the decision boundaries for the same set of input features at the cost of higher omputational requirements in estimating the additional sigma parameter.