Machine Learning - Assignment 1 (COL774)

Ankit Solanki
2016CS50401

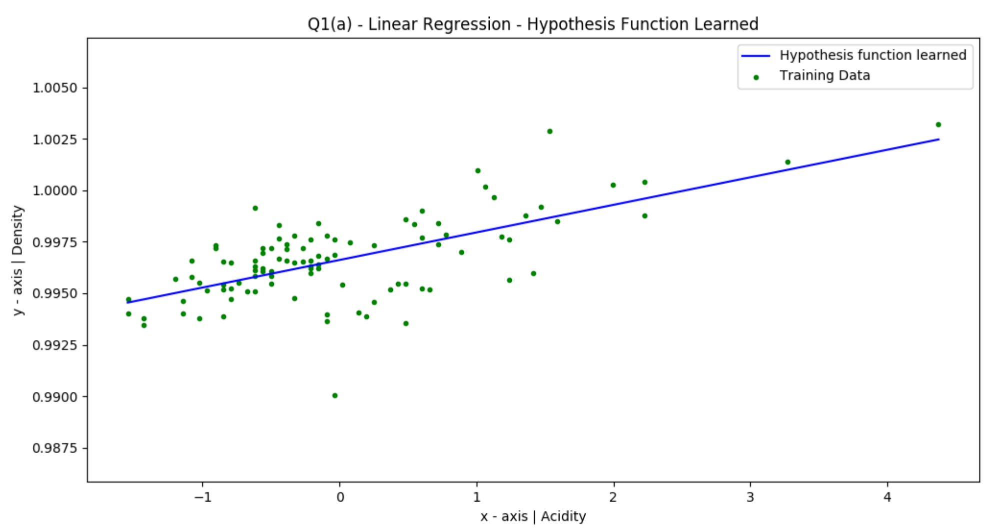
February 12, 2019

1 Problem-1 | Linear Regression

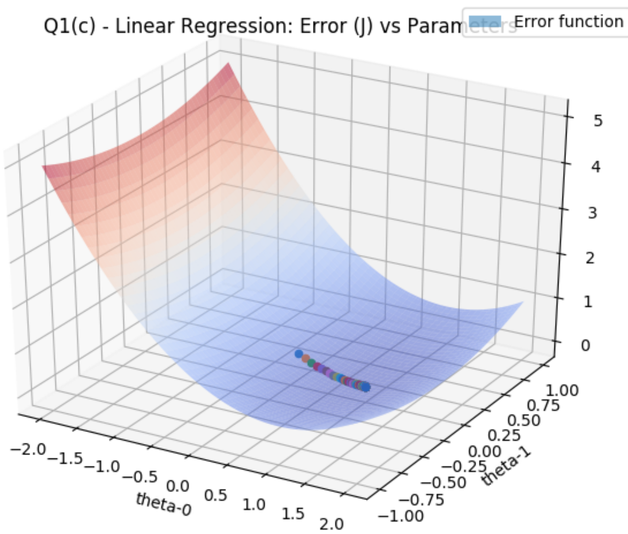
1.1 Part-a

- Learning rate: 0.1
- Stopping Criteria: If change in cost function between two consecutive theta values is less than a predefined threshold (1e-12)
- Parameters: [0.9966172 0.00134019]

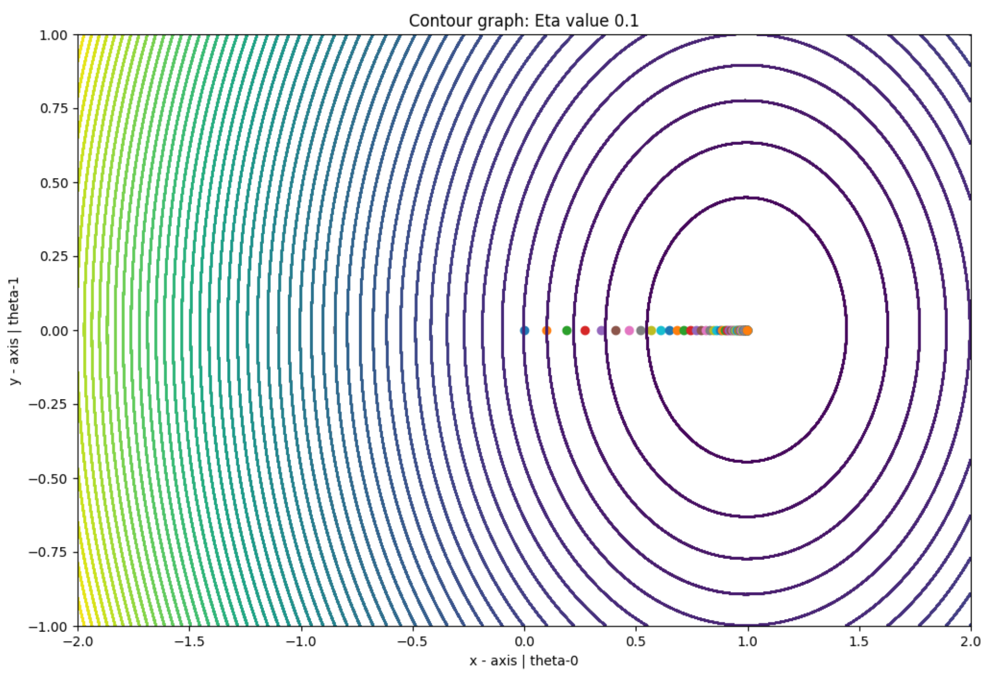
1.2 Part-b



1.3 Part-c



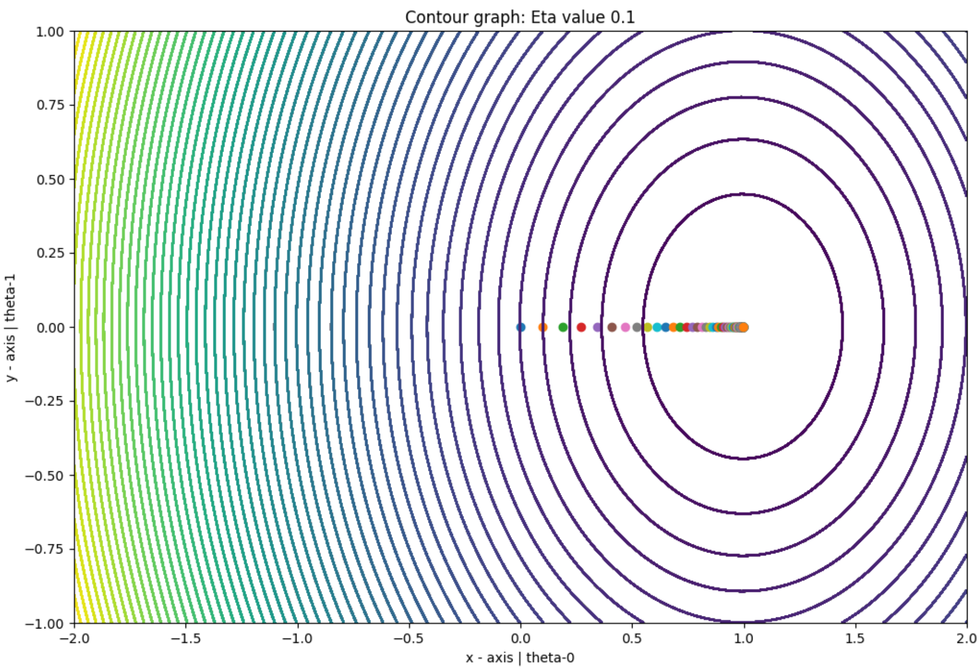
1.4 Part-d



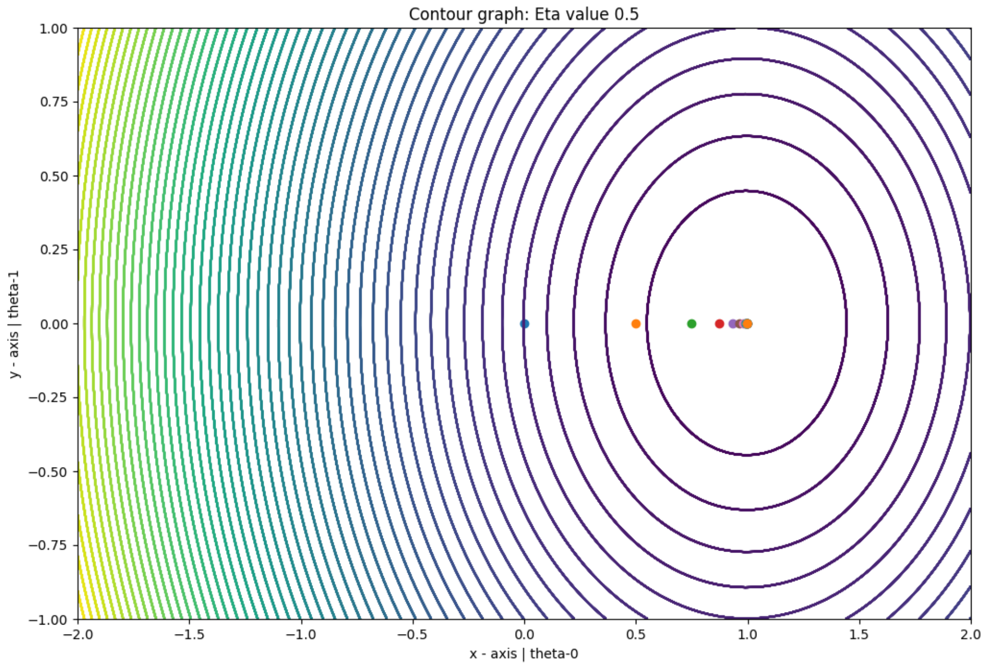
1.5 Part-e

For learning rate (η), mentioned in the subsection names:

1.5.1 0.1

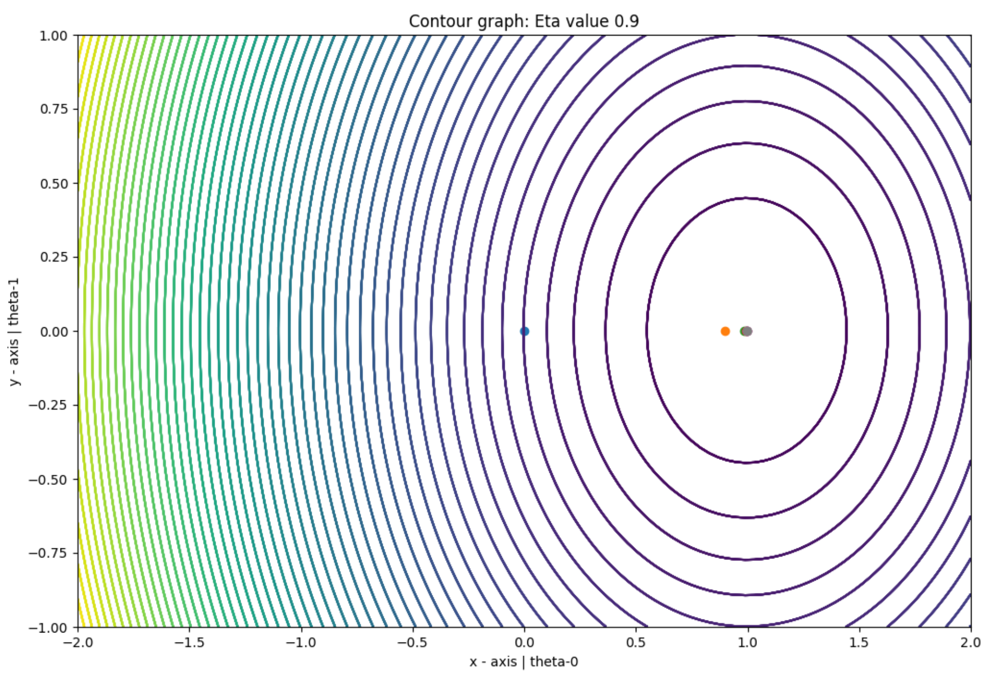


1.5.2 0.5



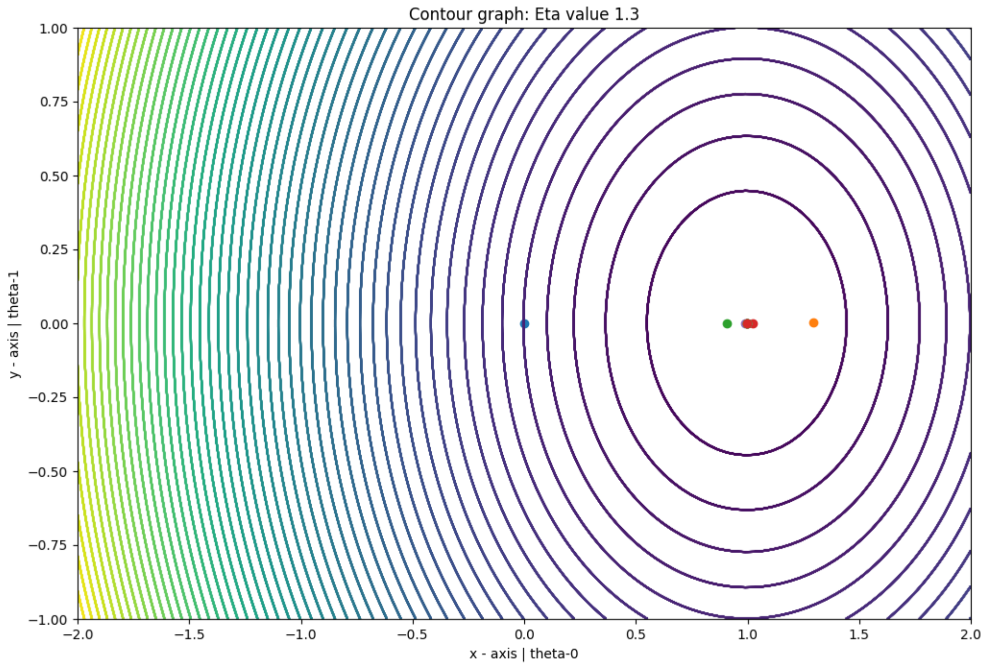
Contour graph: Eta value 0.5
y - axis | theta-1
x - axis | theta-0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

1.5.3 0.9



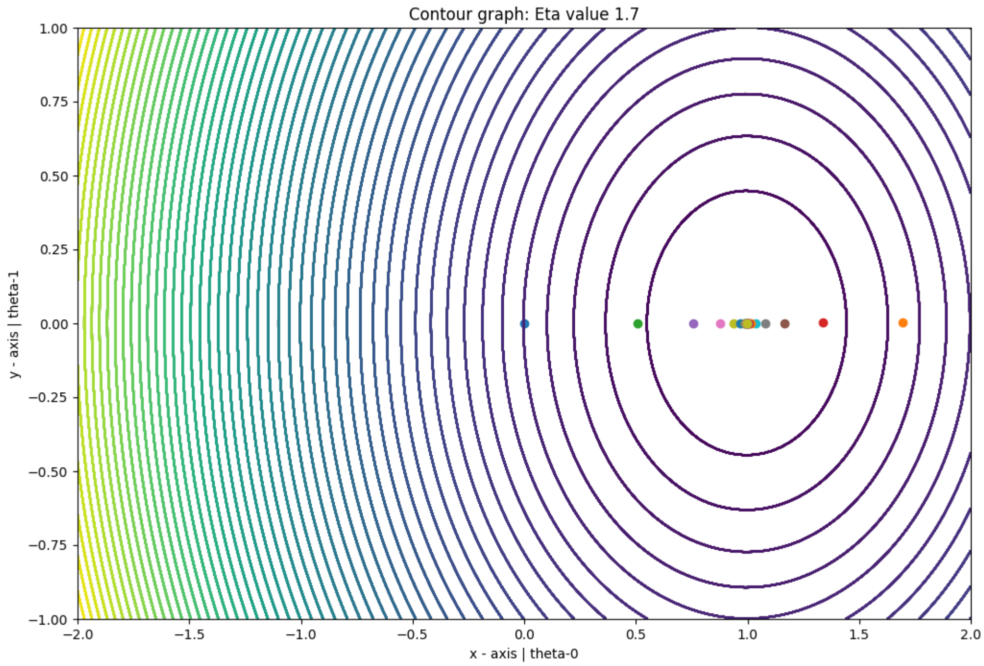
Contour graph: Eta value 0.9
y - axis | theta-1
x - axis | theta-0
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0

1.5.4 1.3



As the learning rate increases the values starts to jump around the minima, here we see for the first time it jumps through the local minia, but still manages to converge somehow.

1.5.5 1.7



Here we see the jumping has increased, but still it converges.

1.5.6 2.1

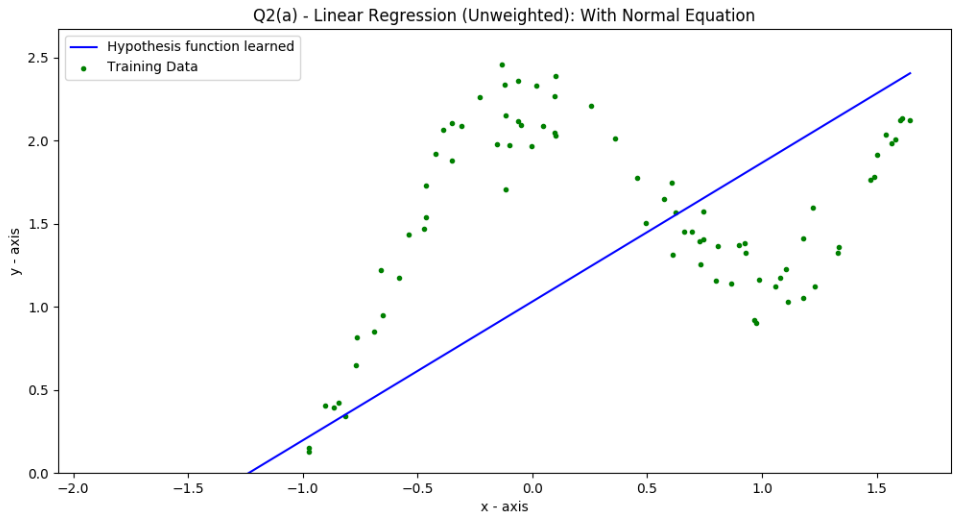
Learning rate has crossed the threshold after which it jumps too much and never converge at the local minima. Till 3,00,000 iteration it doesn't converge, thus it probably will never.

1.5.7 2.5

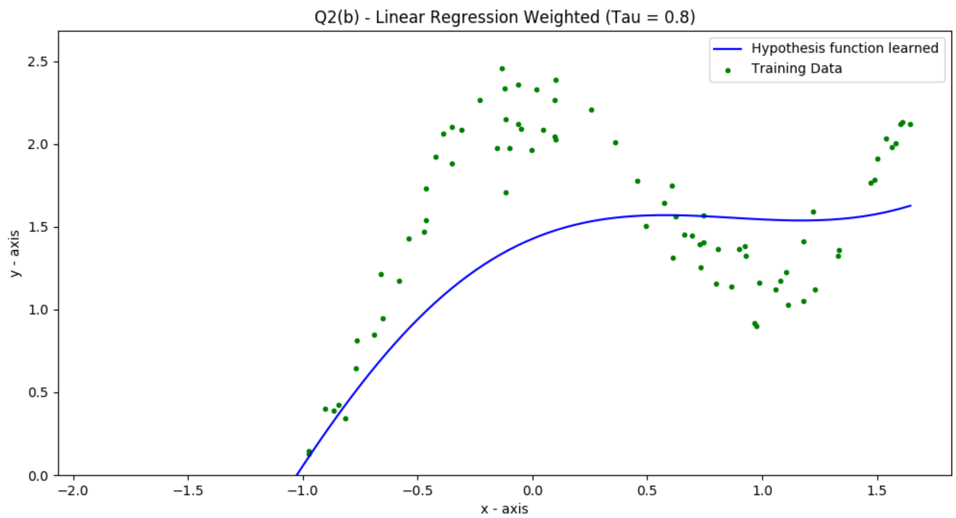
Learning rate > last time. And it is also jumping too much and didn't converge at the local minima. Till 5,00,000 iteration observed, it doesn't converge.

2 Problem-2 | Locally Weighted Linear Regression

2.1 Part-a

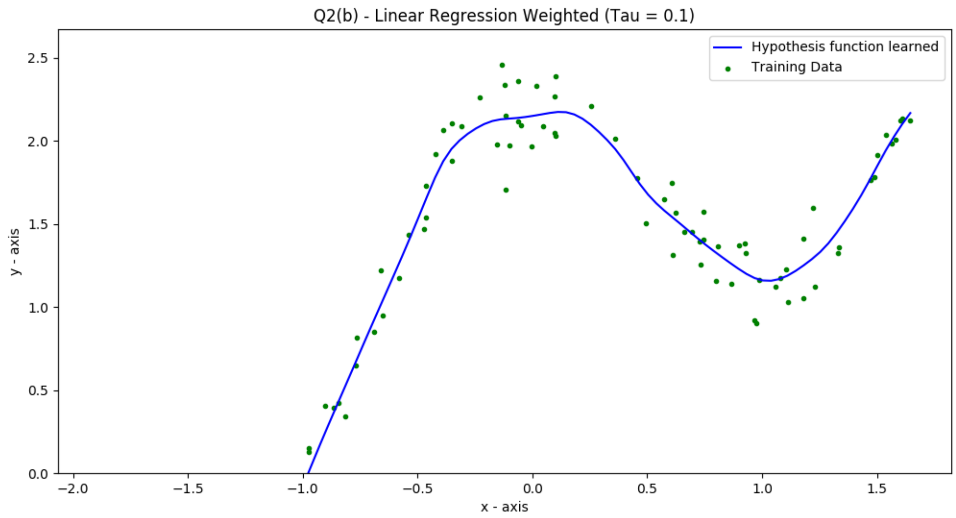


2.2 Part-b

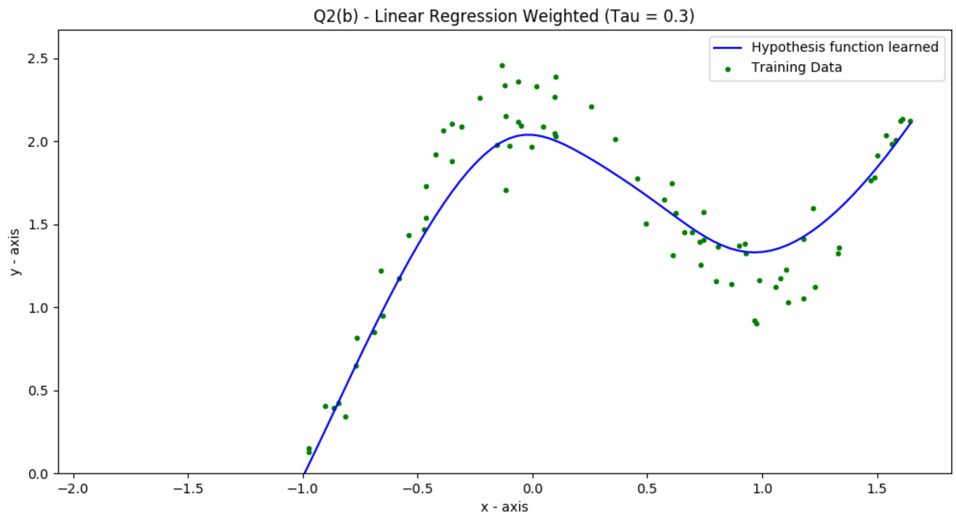


2.3 Part-c

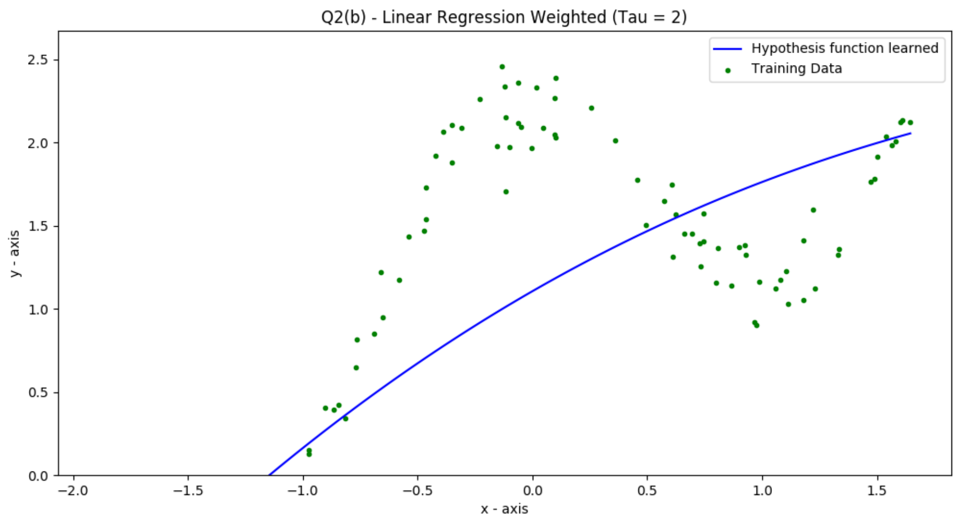
For value τ= 0.1



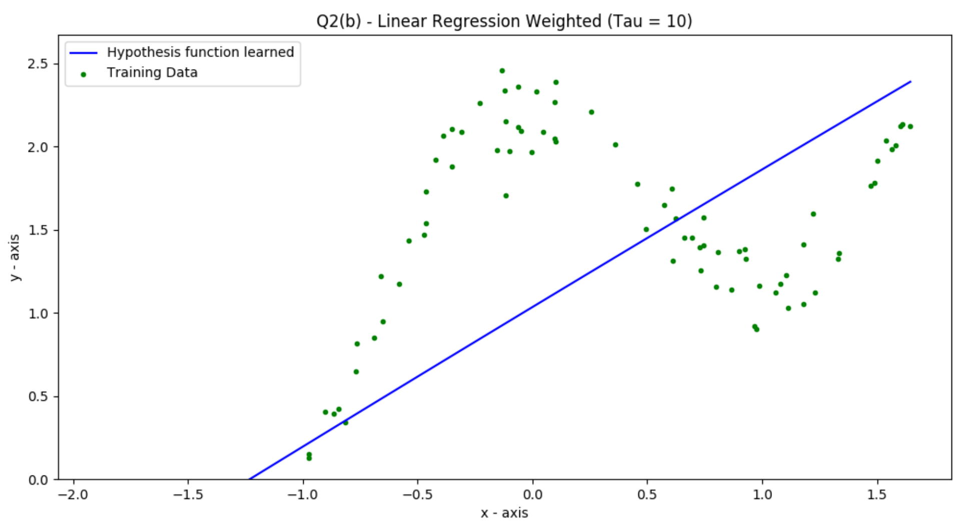
For value τ= 0.3



For value τ= 2



For value $\tau = 10$



I think the best accuracy could be provided by testing on a test set. Nevertheless, just by viewing the graphs, I think $\tau = 0.1$ works the best, as it seems to have approximated the function accurately. Maybe there is a chance of over-fitting because of very close approximation to training function.

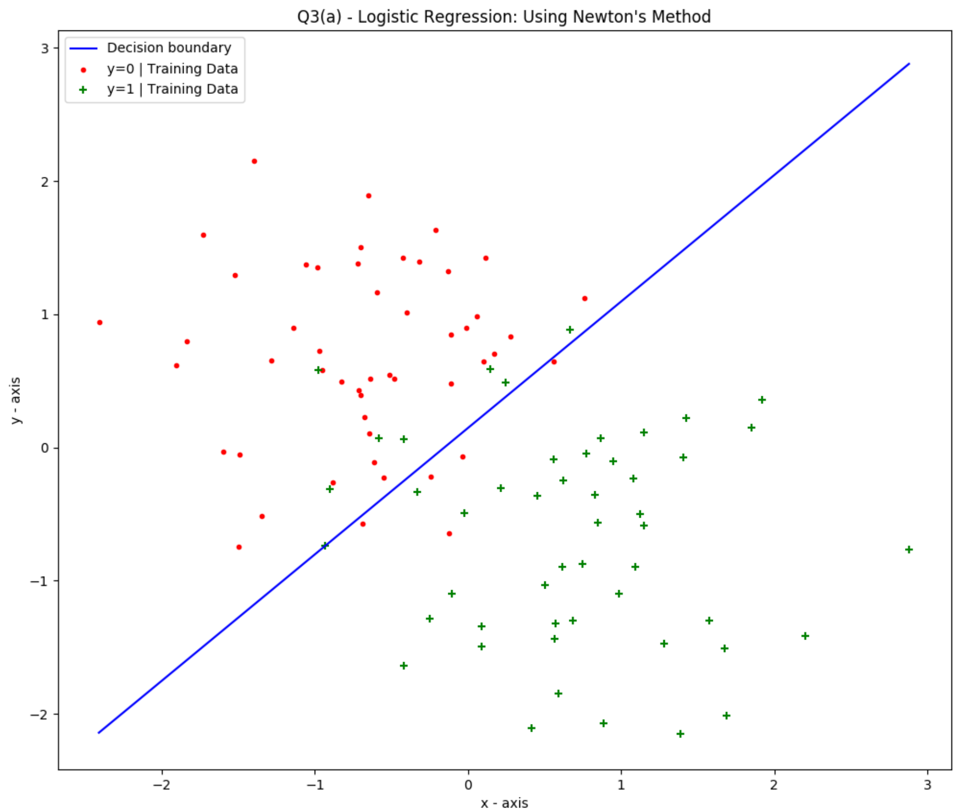
If τ (dispersion parameter) is very large, all features become equally likely to be contributing to the hypothetical function, and their weights can't identify the "dispersion", whereas with a small value of τ, it accurately learns which weights are suitable for which features. Thus, learns well. Too small value could always create over-fitting, but let's not consider that here.

3 Problem-3 | Logistic Regression With Newton's Method

3.1 Part-a

The coefficients resulting from the fit are: [0.40125316 2.5885477 -2.72558849]

3.2 Part-b



4 Problem-4 | Gaussian Discriminant Analysis

4.1 Part-a

Consider the mapping to be: "Alaska": 0, "Canada": 1

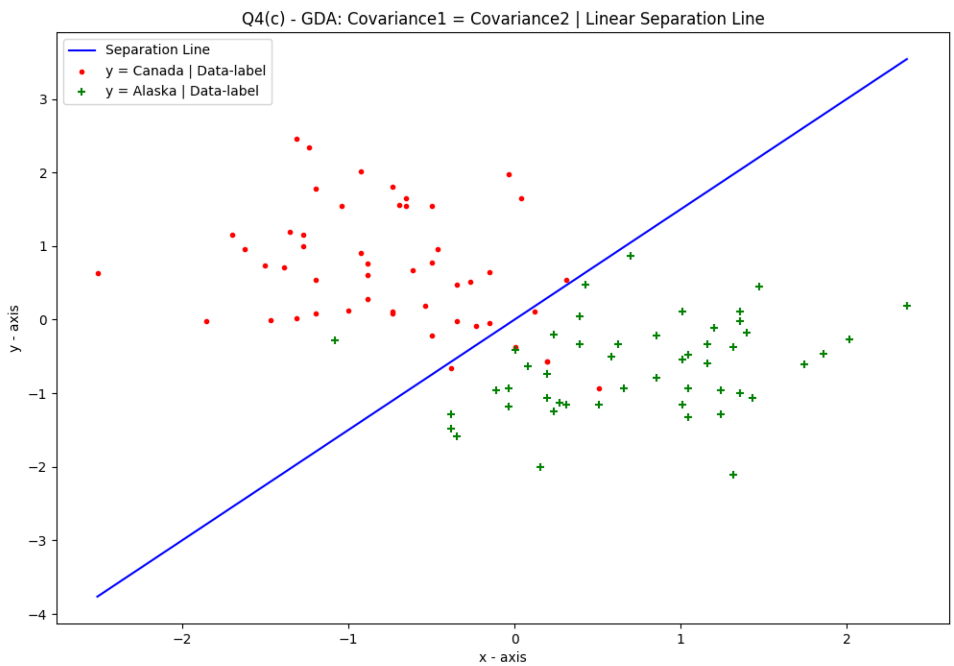
Mean x | y=0: [-0.75529433 0.68509431]

Mean x | y=1: [0.75529433 -0.68509431]

Covariance Matrix: [[0.42953048 -0.02247228] [-0.02247228 0.53064579]]

4.2 Part-b and c

$$X^T\Sigma^{-1}(\mu_0-\mu_1)+(\mu_0-\mu_1)^T\Sigma^{-1}X+(\mu_1^T\Sigma^{-1}\mu_1-\mu_0^T\Sigma^{-1}\mu_0)=2log_e(\frac{\phi}{1-\phi})$$



4.3 Part-d

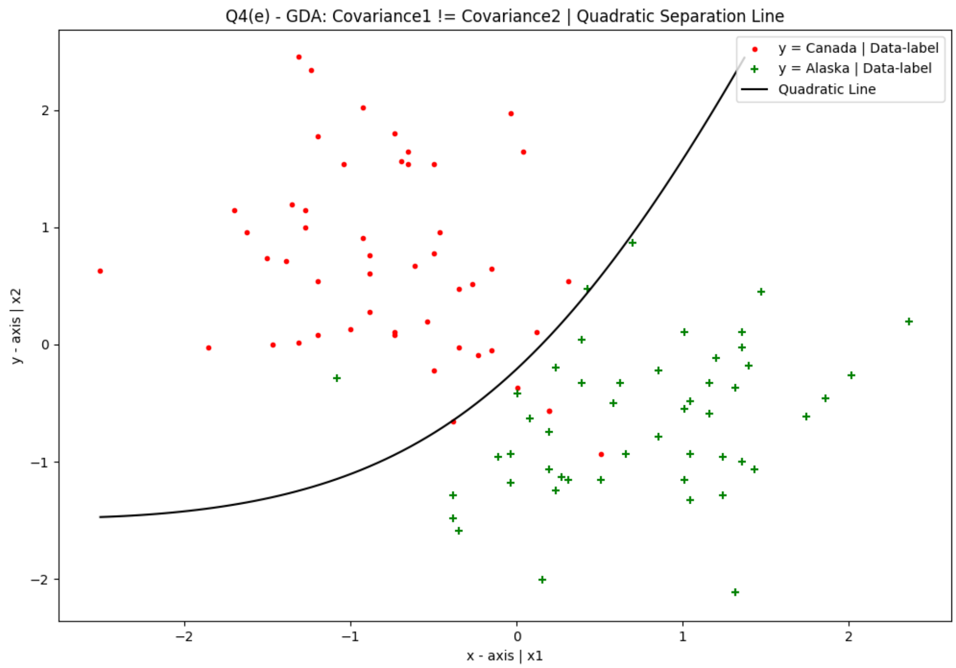
Mean x | y=0: [-0.75529433 0.68509431]
Mean x | y=1: [0.75529433 -0.68509431]

Covariance Matrix for x | y=0: [[0.38158978 -0.15486516] [-0.15486516 0.64773717]]

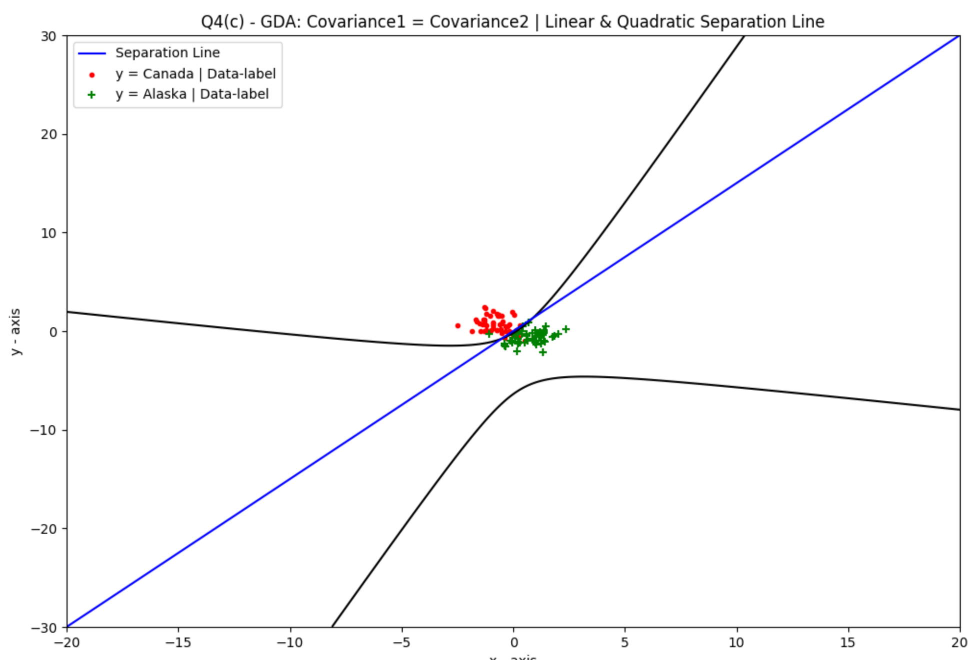
Covariance Matrix for y=1: [[0.47747117 0.1099206] [0.1099206 0.41355441]]

4.4 Part-e and f

$$(X-\mu_1)^T\Sigma_1^{-1}(X-\mu_1)-(X-\mu_0)^T\Sigma_0^{-1}(X-\mu_0)-2log_e(\frac{\phi}{1-\phi})+log_e(\frac{|\Sigma_1|}{|\Sigma_0|})=0$$



A broad diagram with linear and quadratic boundaries would be:



The quadratic curve is a hyperbola as we can see in the zoom out diagram.

The quadratic performs better than linear, it moves around to get points which linear can't get.