COL774 Assignment - 1

Vidushi Maheshwari (2021CS10083)

February 16, 2025

Contents

1 Linear Regression **2**
- 1.1 Parameters . . . 2
- 1.2 Regression Plot . . . 3
- 1.3 Learning θ over 3D Mesh Plot . . . 4
- 1.4 Learning θ over 2D Contour Plot . . . 5
- 1.5 2d Contour Plot for different η . . . 6

2 Sampling, Closed Form and Stochastic Gradient Descent **8**
- 2.1 Sampling . . . 8
- 2.2 SGD on different Batch sizes($\eta = 0.001$) . . . 8
- 2.3 Comparision between obtained parameters . . . 8
- 2.4 Mean Square Error . . . 9
- 2.5 Parameters update trajectory . . . 10

3 Logistic Regression **11**
- 3.1 Newton's Method . . . 11
- 3.2 Training Data and Decision Boundary . . . 11

4 Gaussian Discriminant Analysis **12**
- 4.1 Learned Parameters for shared Covariance Matrix . . . 12
- 4.2 Training Data Plot (with Decision Boundary) . . . 12
- 4.3 Decision Boundary and Equation for shared Covariance Matrix . . . 13
- 4.4 Learned Parameters for separate Covariance Matrix . . . 13
- 4.5 Decision Boundary and Equation for separate Covariance Matrix . . . 14
- 4.6 Observations . . . 14

§1 Linear Regression

§1.1 Parameters

- Performed Batch Gradient Descent with learning rate = 0.025 and stopping criteria as difference between loss functions of consecutive iterations $< \epsilon$ where $\epsilon = 10^{-5}$
- Final Parameters learned by algorithm

$$\theta = \begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \begin{bmatrix} 6.21465047 \\ 29.04579095 \end{bmatrix}$$

  where hypothesis function $h(\theta) = \theta_0 + \theta_1 x$
- Final Loss $J(\theta) = 0.0050626965$
- Number of iterations to converge for given learning rate and stopping criteria: 290

§1.2 Regression Plot

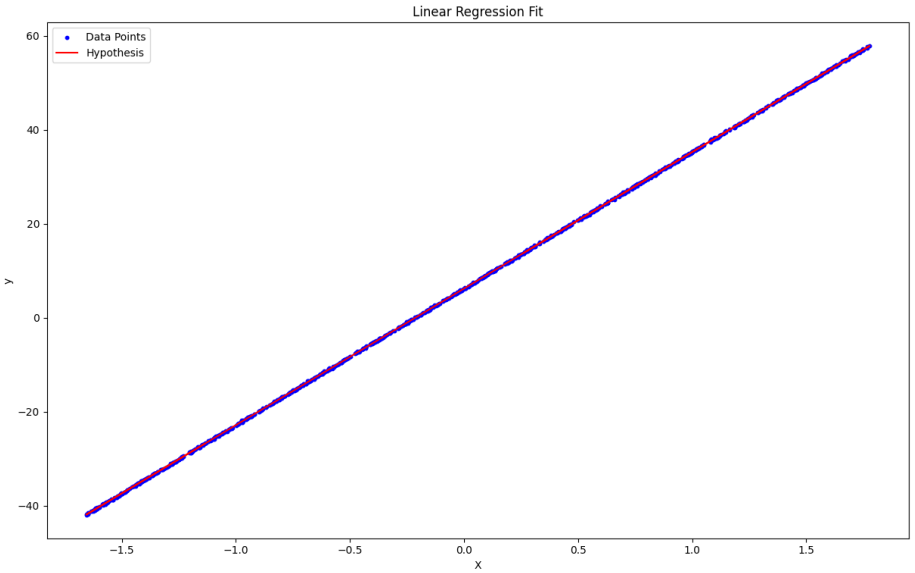


Figure 1: Data and Hypothesis Function plot

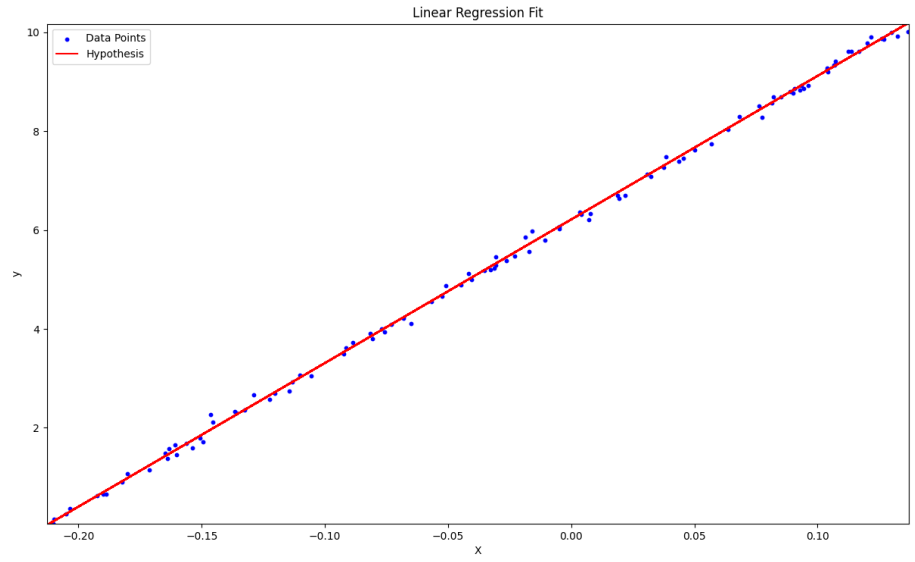


Figure 2: Zoomed-in view of plot

§1.3 Learning θ over 3D Mesh Plot

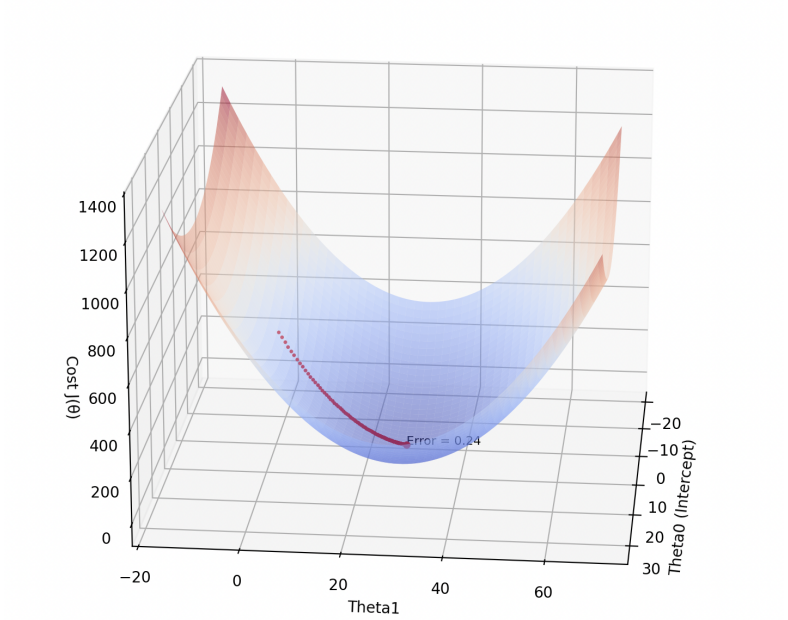


Figure 3: 3D Mesh Plot of Error $J(\theta)$ along with Gradient Descent path

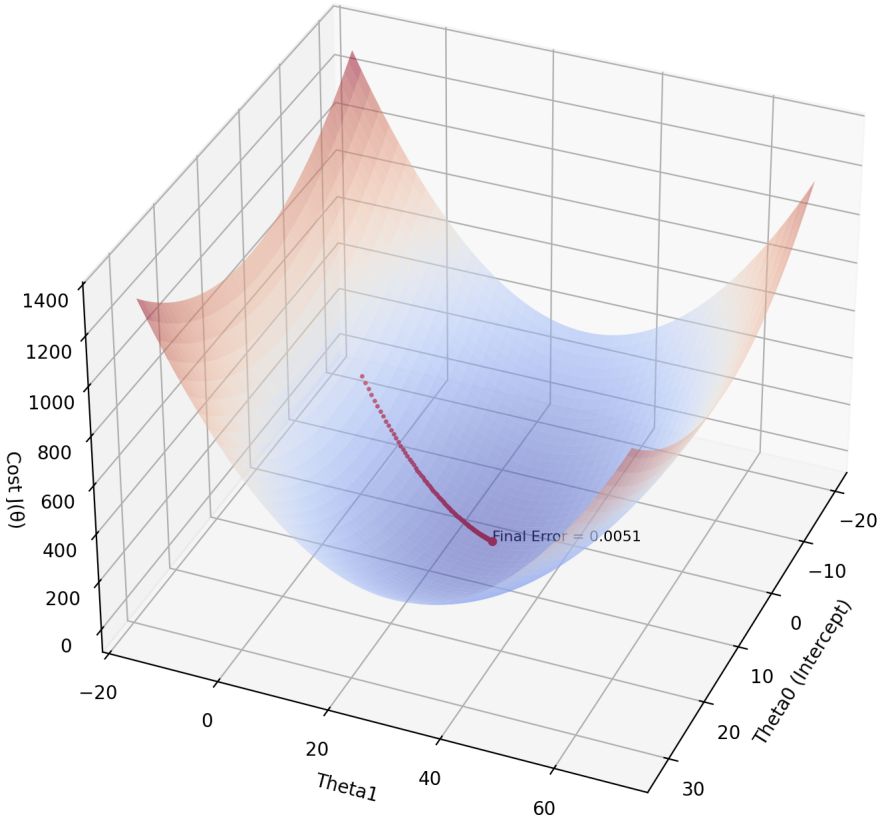


Figure 4: Plot after final iteration

§1.4 Learning θ over 2D Contour Plot

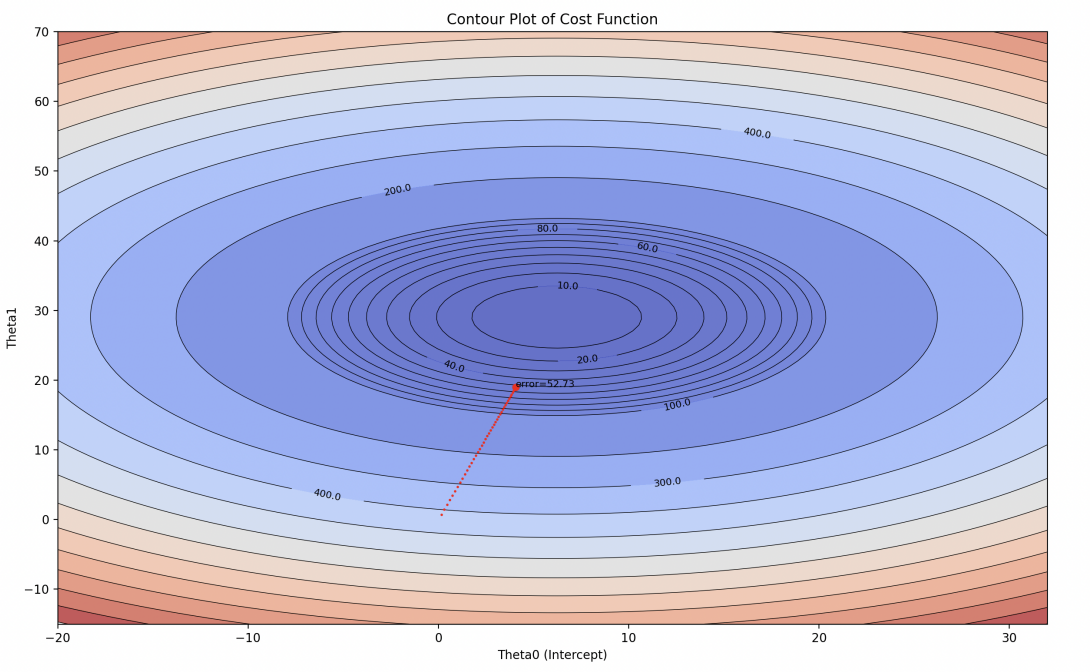


Figure 5: 2D Contour Plot of Error $J(\theta)$ along with Gradient Descent path

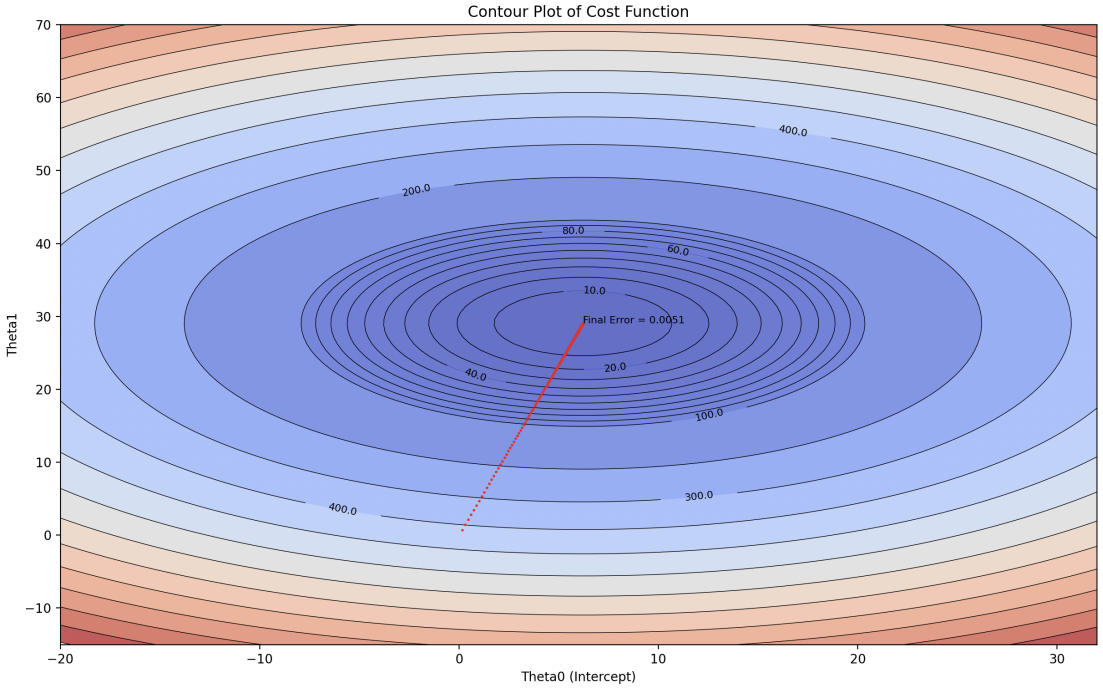


Figure 6: Plot after final iteration

§1.5 2d Contour Plot for different η

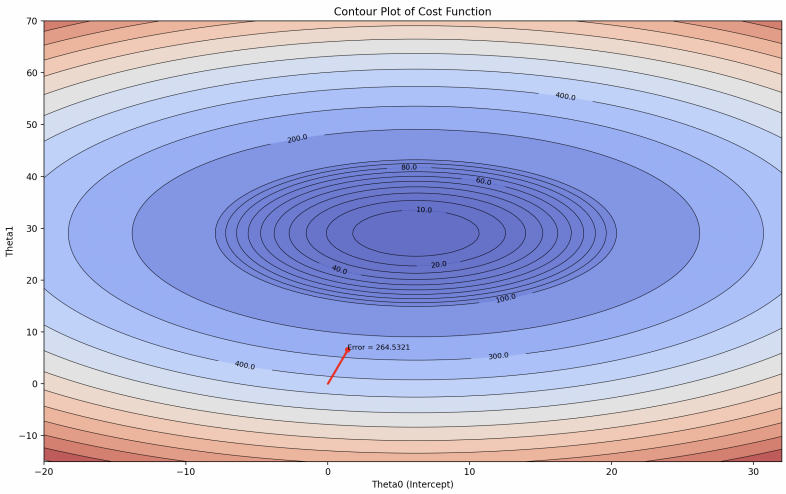


Figure 7: Learning Rate = 0.001

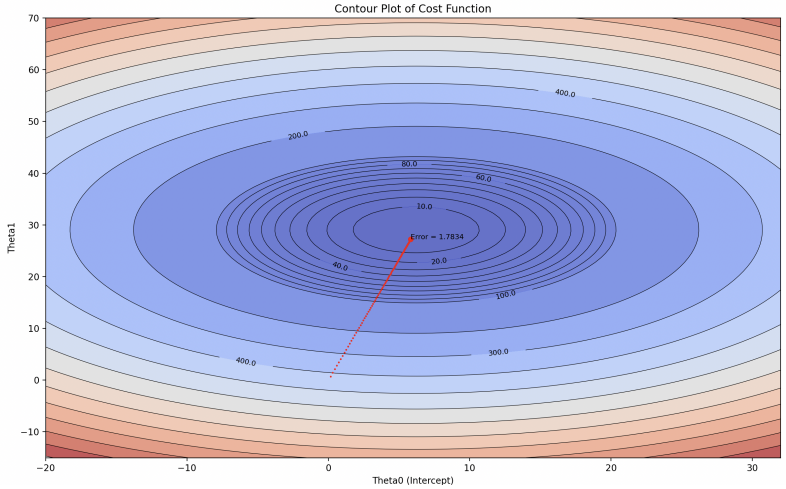
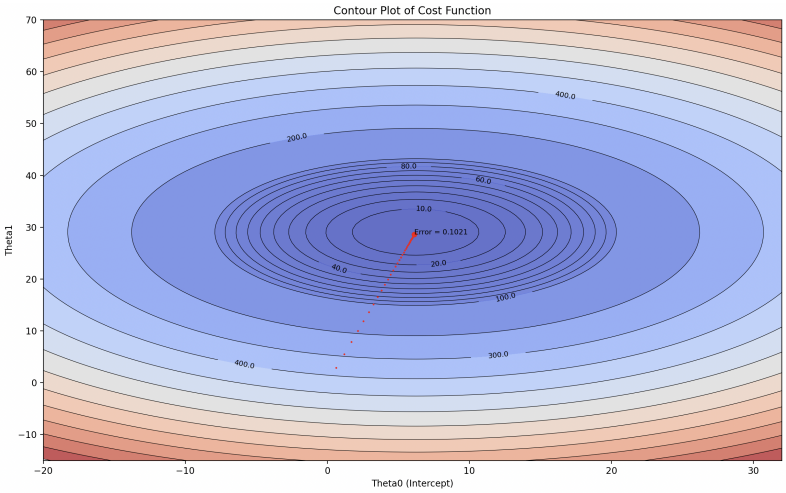


Figure 8: Learning Rate = 0.025



Learning Rate = 0.1

§1.5.1 Observations

- $\eta = 0.001$
  The gradient descent updates are very small. We can see on the contour plot, the red marker moves very tiny stepstowards the minimum. This indicates extremely slow convergence, meaning the algorithm with take many iterations to make significant process.

- $\eta = 0.025$
  The marker's path is smooth and faster. This rate provides a balance between stability and convergence speed. The updates are noticeable but not too large and the algorithm steadily approaches the minimum.

- $\eta = 0.1$
  The jumps in the red marker are very large which might result in overshooting the minimum. While the algorithm moves very quickly, it risks instability/ divergence and might not converge as reliably.

§2 Sampling, Closed Form and Stochastic Gradient Descent

§2.1 Sampling

Generated 1 million data points based on following distribution:

$$\theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}$$

$$x = \begin{pmatrix} x_0 \\ x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ \mathcal{N}(3,4) \\ \mathcal{N}(-1,4) \end{pmatrix}, \quad \epsilon = \mathcal{N}(0,2),$$

$$y = \theta^T x + \epsilon$$

§2.2 SGD on different Batch sizes($\eta = 0.001$)

Batch Size	Epochs	ϵ	θ_0	θ_1	θ_2
1	4	10^{-3}	3.02488	0.98214	1.99402
80	6	10^{-4}	2.99861	0.99728	2.00178
8000	162	10^{-5}	2.96991	1.00639	1.99829
800000	2000	10^{-5}	1.38727	1.35205	1.88188

Smaller batch sizes require loose tolerance because the inherent noise makes the loss function fluctuate. This prevents premature stopping and allows the algorithm to average out the noise over multiple iterations.
Larger batch sizes produce smoother loss curves that allow for tighter tolerance thresholds. The updates are more reliable, so the algorithm can be stopped once very small changes are observed.

§2.3 Comparision between obtained parameters

Convergence for various Batch Sizes

- While the parameters for batch sizes 1, 80 and 8000 converged to true parameter (3,1,2), batch size 800000 ended up with noticeably different parameter values. This indicates that it failed to converge under the given learning rate and stopping criteria, even after taking 2000 epochs.
- The smaller batch sizes (1 and 80) reached near-optimal parameters in just handful of epochs. Although each gradient step was noisier, the updates move through the parameter space quickly.
- Intermediate Batch (8000) required more epochs (162) but still converged to the true parameters, suggesting a reasonable balance between stability and speed.
- The updates for 800000 batch size were very stable but required large number of iterations (2000) and still failed to converge to the correct solution. With such a large batch size, and very low learning rate ($\eta = 0.01$), the algorithm was moving very sluggishly leading to suboptimal results.

Closed Form Parameters

Parameters learned using closed form equation $\theta = (X^T X)^{-1} X^T Y$

$$\theta = \begin{pmatrix} 3.00400433 \\ 0.99895449 \\ 2.00076897 \end{pmatrix}$$

We were able to recover close to true parameters using SGD for batch size (1, 80, 8000) and closed form equation but the parameters for batch size 800000 diverged a lot from correct value.

§2.4 Mean Square Error

Batch Size / Closed Form	Training MSE	Test MSE
Closed Form	1.999150	1.998768
1	2.000984	2.000990
80	1.999296	1.998942
8000	1.999483	1.999046
800000	2.747116	2.746968

For the majority of the cases (closed-form, batch sizes 1, 80, and 8000), the training error is almost equal to the test error. This consistency shows that the model is neither overfitting nor underfitting—what it learns from the training set generalizes well to unseen data. However, the very large batch size (800000) resulted in higher errors even after many iterations, indicating that the learning process did not converge to the optimal solution for that configuration.

§2.5 Parameters update trajectory

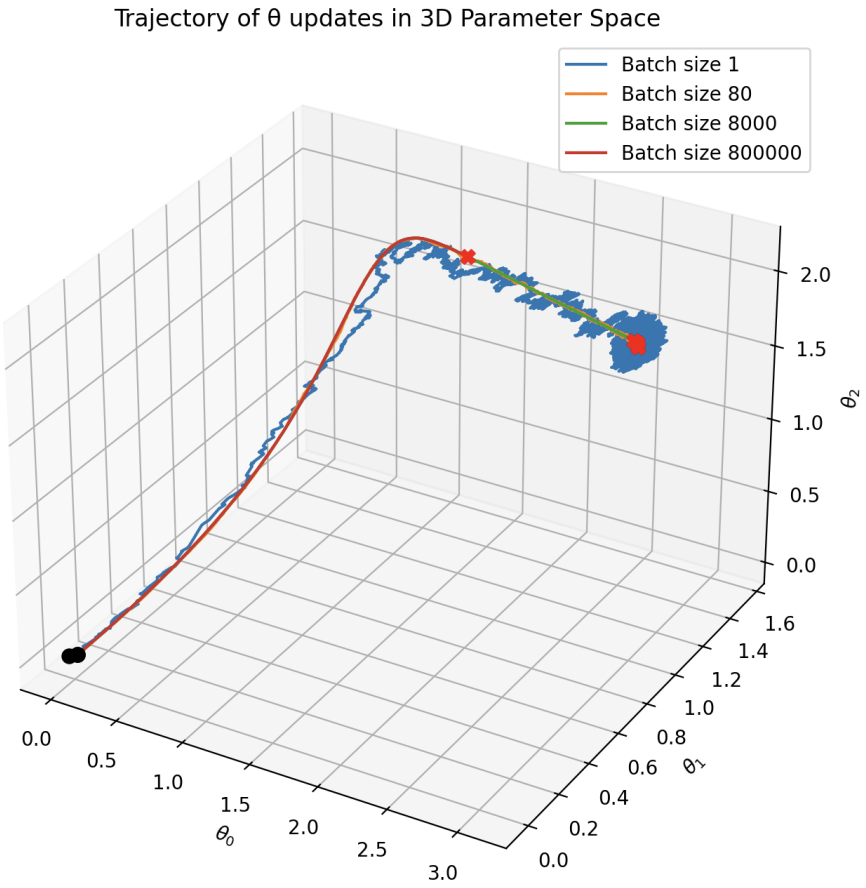


Figure 9: For batch sizes [1, 80, 8000, 800000] with $\eta = 0.01$

We can see the path for smaller batch sizes resembles zigzag pattern due to noise arising from very small amount of training samples in a batch (which increases the influence of each example during the update). For large batch size, the updates are very smooth and consistently directed towards the optimum without any detours. All the batch sizes except 800000 (which moved very slowly towards true parameters) converged to similar point.

§3 Logistic Regression

§3.1 Newton's Method

$$\ell(\theta) \;=\; \sum_{i=1}^{m}\Big[y^{(i)}\log\big(h_\theta(x^{(i)})\big) \;+\; \big(1-y^{(i)}\big)\log\big(1-h_\theta(x^{(i)})\big)\Big],$$

where

$$h_\theta(x) \;=\; \frac{1}{1+e^{-\theta^T x}}.$$

$$\nabla\ell(\theta) \;=\; X^T\big(h_\theta(X)-y\big), \quad \nabla^2\ell(\theta) \;=\; X^T\,B\,X, \quad B=\mathrm{diag}\big(h_\theta(x^{(i)})(1-h_\theta(x^{(i)}))\big).$$

$$\theta \;\leftarrow\; \theta \;-\; \big(\nabla^2\ell(\theta)\big)^{-1}\nabla\ell(\theta).$$

For $\epsilon = 10^{-7}$, the parameters converged to following in 7 iterations

$$\theta = \begin{bmatrix} 0.40105349 \\ 2.58789379 \\ -2.72487941 \end{bmatrix}$$

§3.2 Training Data and Decision Boundary

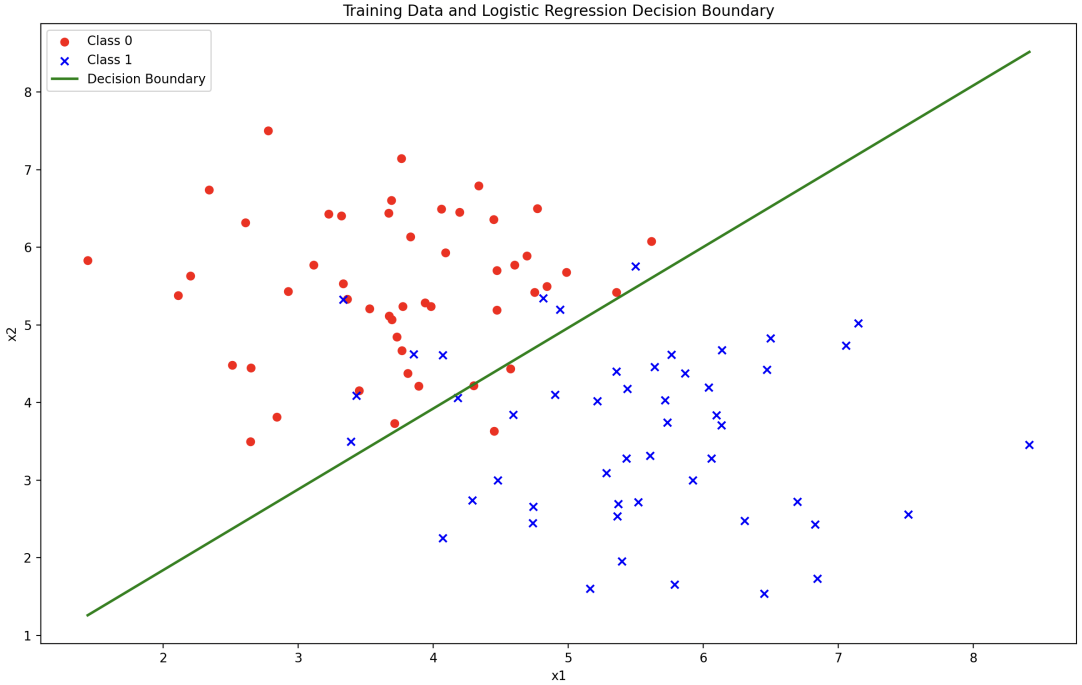


Figure 10: Data points with decision boundary

§4 Gaussian Discriminant Analysis

§4.1 Learned Parameters for shared Covariance Matrix

Normalized Scale

$$\mu_0(\text{Alaska}) = [-0.755, 0.685], \quad \mu_1(\text{Canada}) = [0.755, -0.685]$$

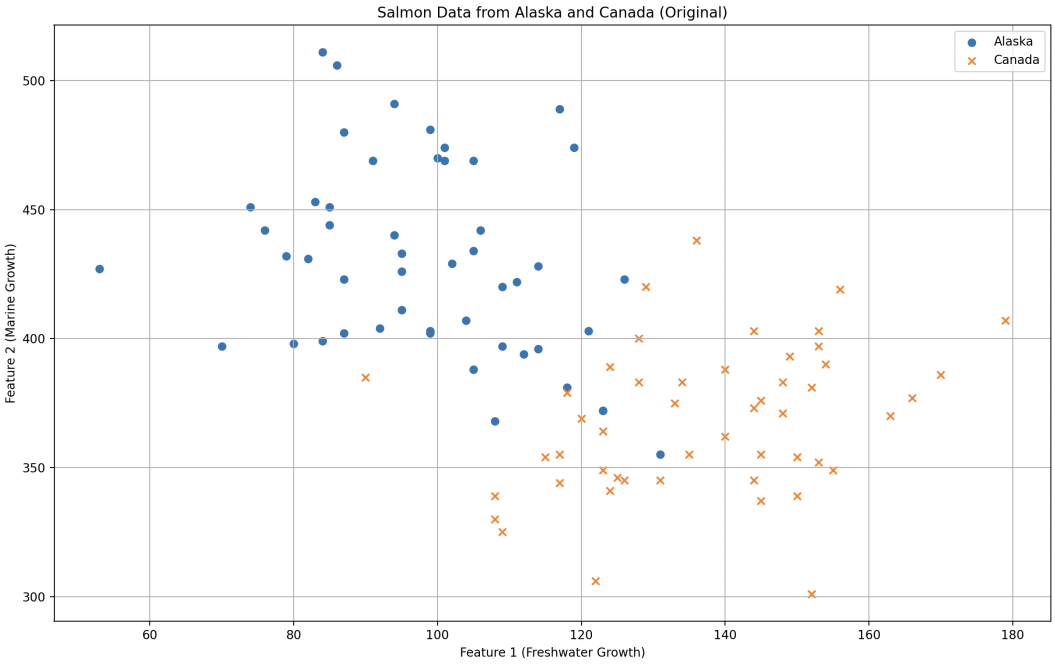
$$\Sigma = \begin{bmatrix} 0.430 & -0.022 \\ -0.022 & 0.531 \end{bmatrix}$$

Original Scale

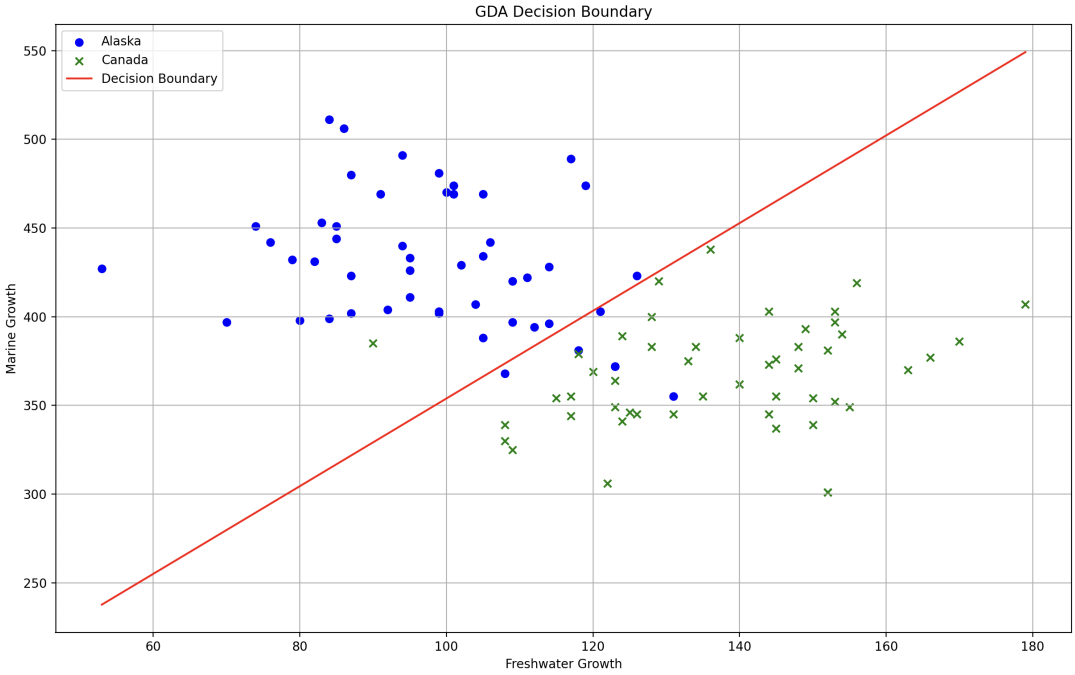
$$\mu_0(\text{Alaska}) = [98.380, 429.660], \quad \mu_1(\text{Canada}) = [137.460, 366.620]$$

$$\Sigma = \begin{bmatrix} 287.482 & -26.748 \\ -26.748 & 1123.250 \end{bmatrix}$$

§4.2 Training Data Plot (with Decision Boundary)



§4.3 Decision Boundary and Equation for shared Covariance Matrix



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

§4.4 Learned Parameters for separate Covariance Matrix

Normalized Scale

$$\mu_0(\text{Alaska}) = [-0.755, 0.685], \quad \mu_1(\text{Canada}) = [0.755, -0.685]$$

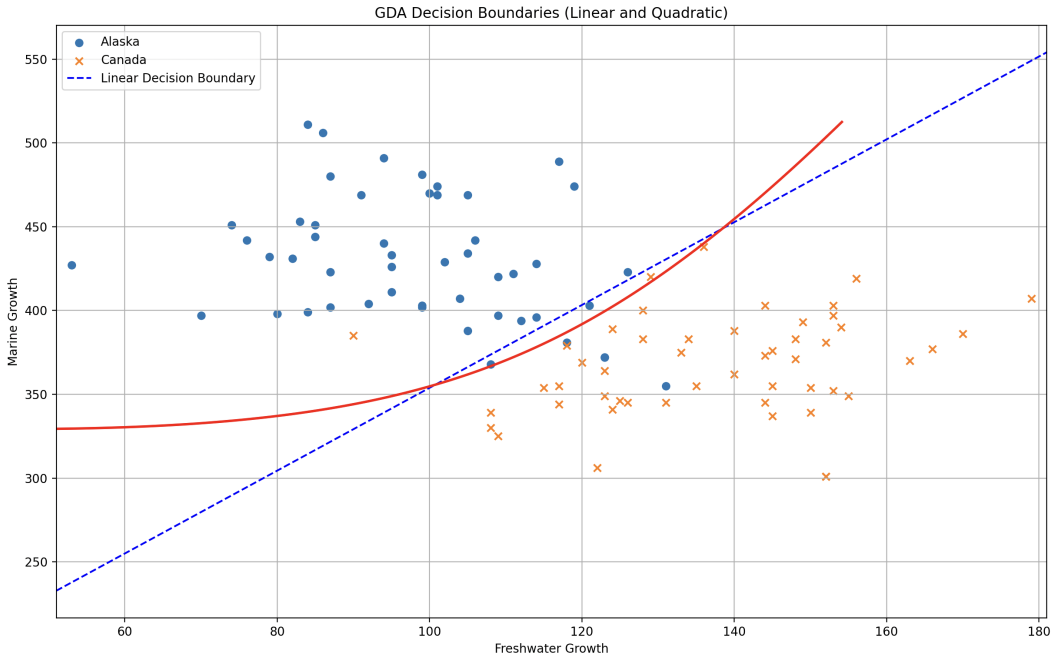
$$\Sigma_0 = \begin{bmatrix} 0.382 & -0.155 \\ -0.155 & 0.648 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 0.477 & 0.110 \\ 0.110 & 0.414 \end{bmatrix}$$

Original Scale

$$\mu_0(\text{Alaska}) = [98.380, 429.660], \quad \mu_1(\text{Canada}) = [137.460, 366.620]$$

$$\Sigma_0 = \begin{bmatrix} 255.396 & -184.331 \\ -184.331 & 1371.104 \end{bmatrix}, \quad \Sigma_1 = \begin{bmatrix} 319.568 & 130.835 \\ 130.835 & 875.396 \end{bmatrix}$$

§4.5 Decision Boundary and Equation for separate Covariance Matrix



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

§4.6 Observations

Linear Boundary

: The linear boundary offers a clean separation between the two classes, Alaska (blue dots) and Canada (orange crosses). While it misclassifies some points, especially in regions where the classes overlap, it provides a relatively balanced and consistent division, particularly for mid to high values of freshwater growth.

Quadratic Boundary

: The quadratic boundary appears to overfit the data. While it captures some curvature in the distribution, it introduces unnecessary complexity, especially at the lower and higher ends of the freshwater growth axis.