COL774
Assignment 1

Sayam Sethi

12 September 2021

Contents

1 Linear Regression

1.a Batch Gradient Descent

On performing *batch gradient descent* with $\eta = 0.05$ and $\varepsilon = 10^{-15}$ (difference between consecutive cost functions), we learn the following model (rounded off to 5 decimal places):

$$\theta = \begin{pmatrix} 0.99662 \\ 0.00134 \end{pmatrix} \tag{1}$$

The learning takes 309 iterations.

1.b Regression Plot

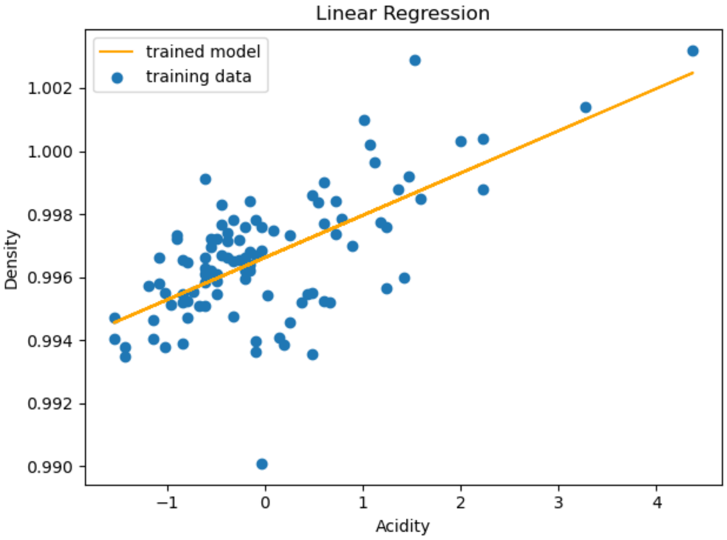


Figure 1: Data and Hypothesis plot for Q1

1.c 3D Mesh and Learning θ

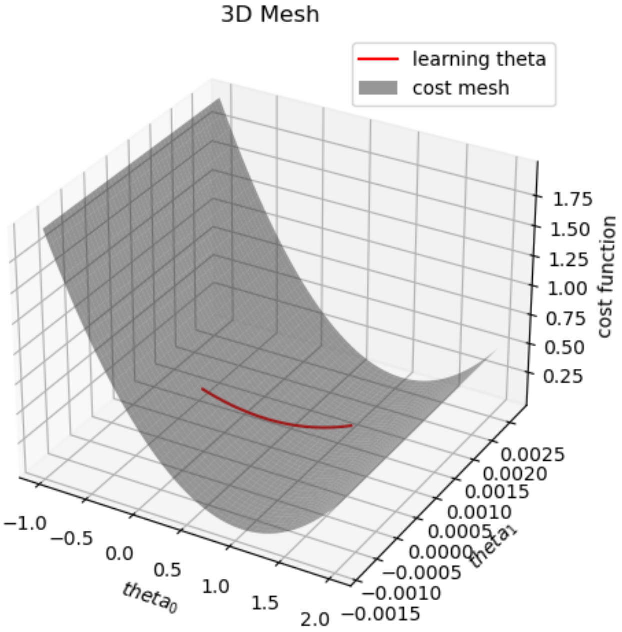


Figure 2: Mesh of Cost Function and Movement of θ

1.d Learning over the Contour

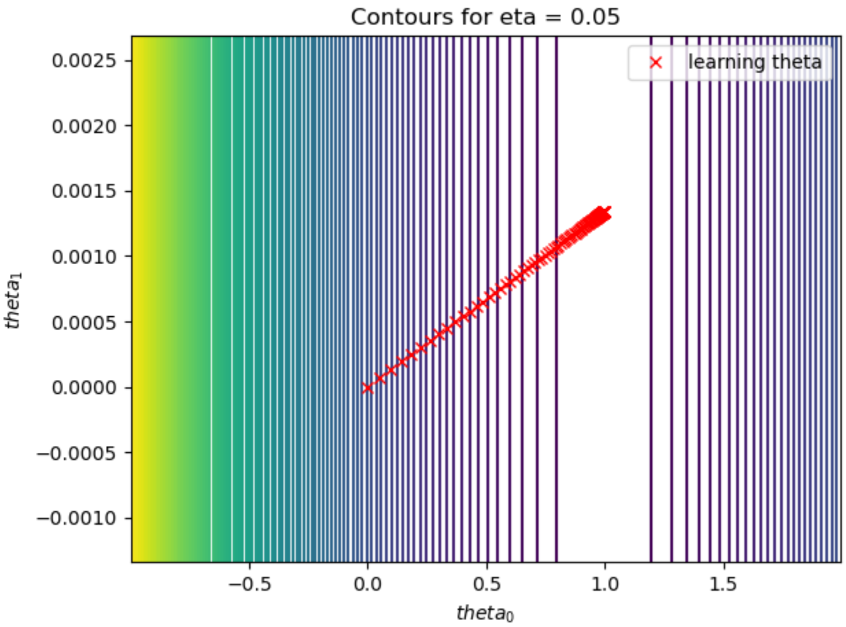


Figure 3: Movement of θ over the Contours

1.e Experimenting with Different η

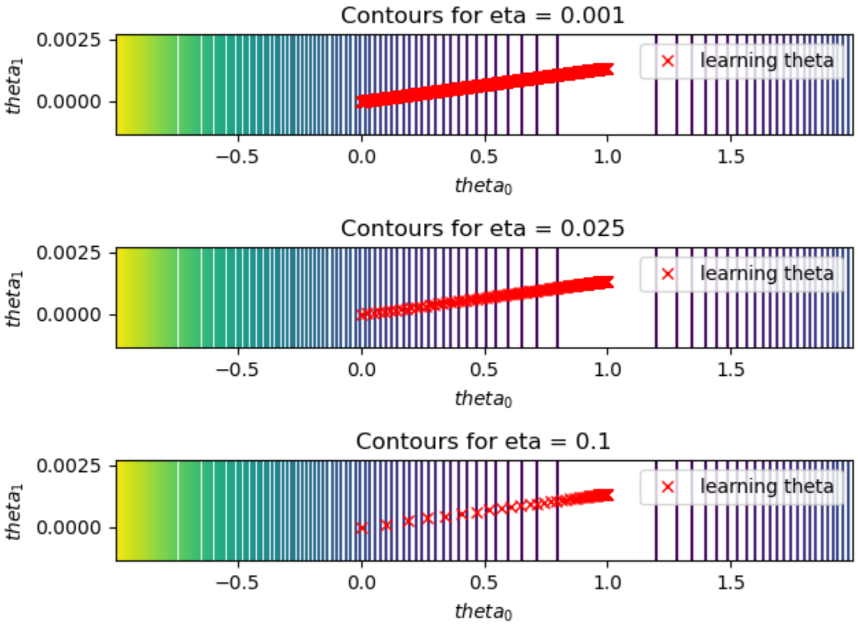


Figure 4: Movement of θ for different η

2 Sampling and Stochastic Gradient Descent

2.a Sampling

The data is sampled as follows:

$$\begin{aligned} x &\sim \begin{pmatrix} \mathcal{N}(3,4) \\ \mathcal{N}(-1,4) \end{pmatrix} \\ y &\sim \theta^T x + \mathcal{N}\left(0, \sqrt{2}\right) \end{aligned} \tag{2}$$

1 million samples are generated using the above sampling criteria.

2.b Stochastic Gradient Descent

The convergence criteria used is:

$$|\underset{\text{epoch } t+1}{average}(J(\theta^{t+1})) - \underset{\text{epoch } t}{average}(J(\theta^{t}))| \leq \varepsilon, \text{ where } \varepsilon = 10^{-5} \tag{3}$$

The following parameters are learnt for different batch sizes:

Batch size (r)	θ	**Iterations**	**Time taken**
1	$\begin{pmatrix} 2.99630 \\ 0.98101 \\ 1.99178 \end{pmatrix}$	3000000 (3 epochs)	43.42 seconds
100	$\begin{pmatrix} 2.99767 \\ 0.99998 \\ 1.99899 \end{pmatrix}$	40000 (4 epochs)	0.74 seconds
10000	$\begin{pmatrix} 2.96458 \\ 1.00759 \\ 1.99770 \end{pmatrix}$	16200 (162 epochs)	4.29 seconds
1000000	$\begin{pmatrix} 2.64004 \\ 1.07834 \\ 1.97405 \end{pmatrix}$	7535 (7535 epochs)	105.58 seconds

2.c Test Error

The errors for different batch sizes on the test set are:

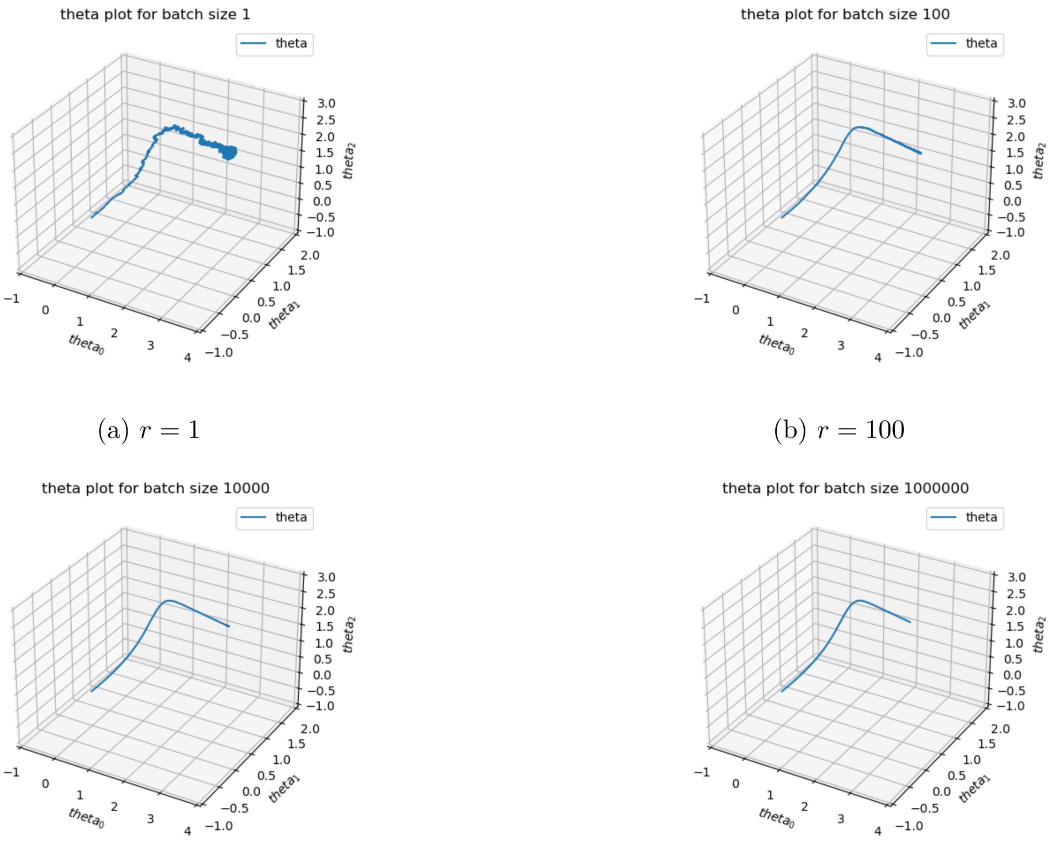
Batch size (r)	**Error**
1	1.00680
100	0.98315
10000	0.98645
1000000	1.35444
original θ	0.98295

The different algorithms (as a parameter of the batch size) have different converging values of θ and thus different values of test error too. The best predictions are done by the algorithms with batch sizes 100 and 10000. Both of them reach convergence in a very small time but taking different number of iterations.

Convergence is the fastest for smaller batch sizes which are not too small. This is because computation of the smaller batches is much quicker than larger batches. However, it needs to be noted that the drawback of having a very small batch size is that the data isn't representative enough for the entire sample and hence θ doesn't converge perpendicular to the contours but instead moves at an angle to it. This slows down the convergence and hence having a slightly larger batch size (which is still relatively small) is ideal.

For larger batch sizes, the jump in θ is relatively larger towards the best value, however, the computation for each jump is costlier. Additionally, this *direct* jump leads to early convergence since the change in θ reduces as we reach closer to the optimal value. This leads to smaller change in the cost function since the contours are farther spaced as we approach closer to the optimal value. Thus, we need a smaller ε for larger batch sizes to converge to the same value.

2.d Movement of θ in the θ Space



(a) $r = 1$

(b) $r = 100$

(c) $r = 10000$

(d) $r = 1000000$

Figure 5: Movement of θ for different batch sizes

The movement of θ in each of the 4 cases above is consistent with the explanation given in Section 2.c. When $r = 1$, θ fluctuates a lot closer to the minimal value and hence convergence is slowed down. Movement of θ for $r = 100, 10000$ is relatively smoother and thus convergence is reached in a small time. For the case of $\theta = 1000000$, convergence criteria is met much before θ reaches close to the optimal value. This is because the change in consecutive values of θ leads to a smaller change in the cost function since the contours are not as close when closer to the optimal value.

3 Logistic Regression

3.a Newton's Method

To compute the equation of the separator using *Newton's method*, we need to compute the double derivative of $LL(\theta)$, i.e., the *Hessian matrix*. We proceed as follows:

$$\begin{aligned}
\mathcal{H}(LL(\theta)) &= \frac{\partial\left(\nabla_\theta(LL(\theta))\right)}{\partial\theta} \\
&= \frac{\partial}{\partial\theta}\left(X^T\left(Y - \frac{1}{1+e^{-X\theta}}\right)\right) \\
&= \begin{pmatrix} \frac{\partial}{\partial\theta_0}\left(X^T\left(Y - \frac{1}{1+e^{-X\theta}}\right)\right) \\ \frac{\partial}{\partial\theta_1}\left(X^T\left(Y - \frac{1}{1+e^{-X\theta}}\right)\right) \\ \vdots \\ \frac{\partial}{\partial\theta_n}\left(X^T\left(Y - \frac{1}{1+e^{-X\theta}}\right)\right) \end{pmatrix}
\end{aligned} \tag{4}$$

Now, computing each row separately, we get:

$$\begin{aligned}
\mathcal{H}_i &= \frac{\partial}{\partial\theta_i}\left(X^T\left(Y - \frac{1}{1+e^{-X\theta}}\right)\right) \\
&= \frac{\partial}{\partial\theta_i}\left(\left(\frac{1}{1+e^{-X\theta}}\right)^T X\right) \\
&= \frac{\partial(X\theta)^T}{\partial\theta_i}\frac{\partial\left(\frac{1}{1+e^{-X\theta}}\right)^T}{\partial(X\theta)^T}X \\
&= X_i^T\left(\frac{e^{-(X\theta)^T}}{\left(1+e^{-(X\theta)^T}\right)^2}\right)X
\end{aligned} \tag{5}$$

The complete *Hessian matrix* can thus be written as:

$$\mathcal{H} = X^T I_m\left(\frac{e^{-(X\theta)^T}}{\left(1+e^{-(X\theta)^T}\right)^2}\right)X, \text{ where } I_m = \text{identity matrix} \tag{6}$$

We can now perform iterations as follows:

$$\theta^{t+1} \leftarrow \theta^t - \mathcal{H}^{-1}\nabla_\theta(LL(\theta)) \tag{7}$$

On learning the model using *Newton's method*, we get the following θ in 9 iterations when ε is 10^{-20}:

$$\theta = \begin{pmatrix} 0.40125 \\ 2.58855 \\ -2.72559 \end{pmatrix} \tag{8}$$

3.b Regression Plot

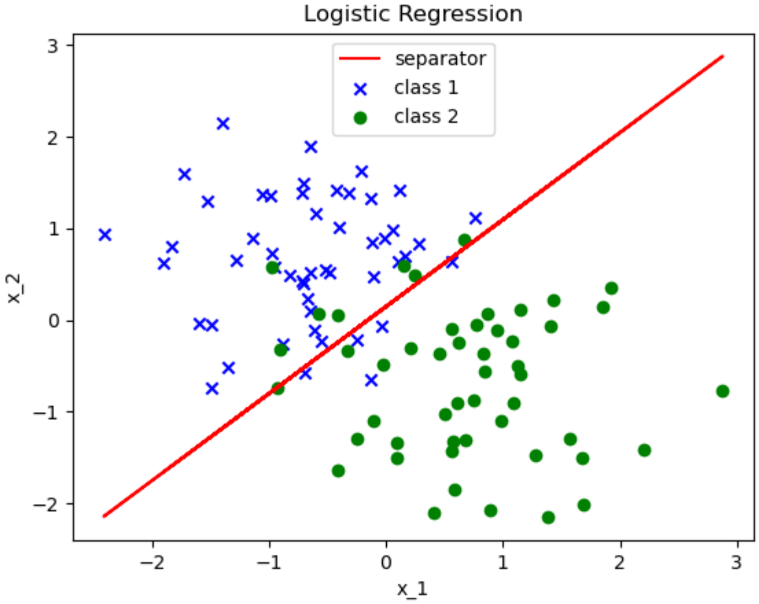


Figure 6: Plot of the data along with the separator

4 Gaussian Discriminant Analysis

4.a Linear GDA

Classifying Alaska as 1 and Canada as 0, the model (Θ) for *linear GDA* is obtained as follows:

$$\begin{aligned} \phi &= 0.5 \\ \mu_0 &= \begin{pmatrix} 0.75529 \\ -0.68509 \end{pmatrix} \\ \mu_1 &= \begin{pmatrix} -0.75529 \\ 0.68509 \end{pmatrix} \\ \Sigma &= \begin{pmatrix} 0.42953 & -0.02247 \\ -0.02247 & 0.53065 \end{pmatrix} \end{aligned} \tag{9}$$

4.b Training Data Plot

Plot is done in Section 4.e.

4.c Linear Separator Plot

The separator is given by:

$$\log\left(\frac{1-\phi}{\phi}\right) - (\mu_1 - \mu_0)^T \Sigma^{-1} x + \frac{1}{2}\left(\mu_1^T \Sigma^1 \mu_1 + \mu_0^T \Sigma^{-1} \mu_0\right) = 0 \tag{10}$$

Plot is done in Section 4.e.

4.d Generic (Quadratic) Separator

The following parameters (Θ) are obtained for *generic GDA*:

$$\begin{aligned}
\phi &= 0.5 \\
\mu_0 &= \begin{pmatrix} 0.75529 \\ -0.68509 \end{pmatrix} \\
\mu_1 &= \begin{pmatrix} -0.75529 \\ 0.68509 \end{pmatrix} \\
\Sigma_0 &= \begin{pmatrix} 0.47747 & 0.10992 \\ -0.10992 & 0.41355 \end{pmatrix} \\
\Sigma_1 &= \begin{pmatrix} 0.38159 & -0.15487 \\ -0.15487 & 0.64774 \end{pmatrix}
\end{aligned} \tag{11}$$

4.e Generic Separator Plot

The separator is given by:

$$\log\left(\frac{1-\phi}{\phi}\sqrt{\frac{|\Sigma_1|}{|\Sigma_0|}}\right) + \frac{1}{2}\left(x^T(\Sigma_1^{-1} - \Sigma_0^{-1})x - 2(\mu_1^T\Sigma_1^{-1} - \mu_0^T\Sigma_0^{-1})x + \mu_1^T\Sigma_1^{-1}\mu_1 - \mu_0^T\Sigma_0^{-1}\mu_0\right) = 0 \tag{12}$$

The plot of the data along with the linear and generic separator is:

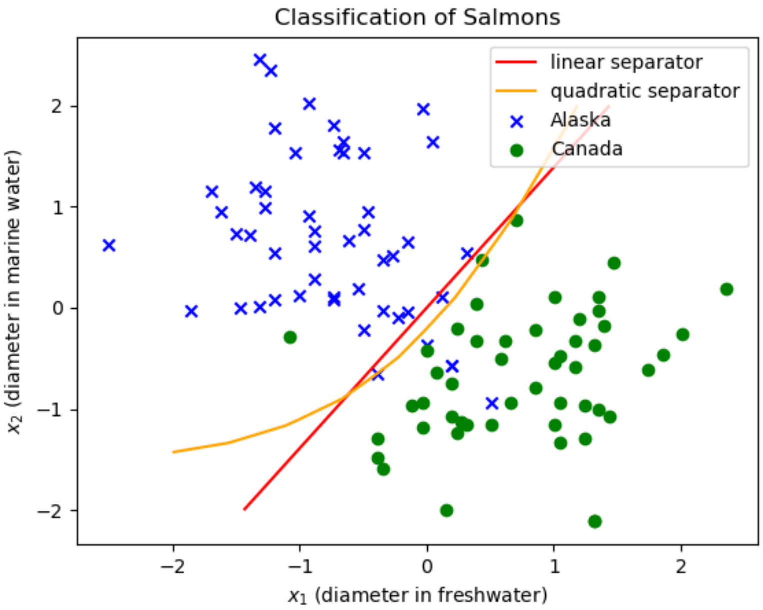


Figure 7: Plot of the data along with the separators

4.f Analysing the Separators

The quadratic separator barely fits about two more points correctly compared to the linear separator. Additionally, the quadratic separator gives the notion that we are more *likely* to have salmons from Canada as compared to Alaska since the separator bends towards Alaska. But looking at the training data, it is apparent that it is not the case. Therefore, the quadratic separator leads to overfitting on the test data without actually giving significantly better results even on the test data.