COL773 Machine Learning Assignment 1

Manoj Kumar (cs5180411@cse.iitd.ac.in)

November 6, 2020

1. Linear Regression

(a) Learning rate (α): 0.001
stopping criteria for convergence:

$$|J(\theta^{t+1}) - J(\theta^{t})| < 0.0000001$$

$$\forall j, |{\theta_j}^{t+1} - {\theta_j}^{t}| < 0.0000001$$

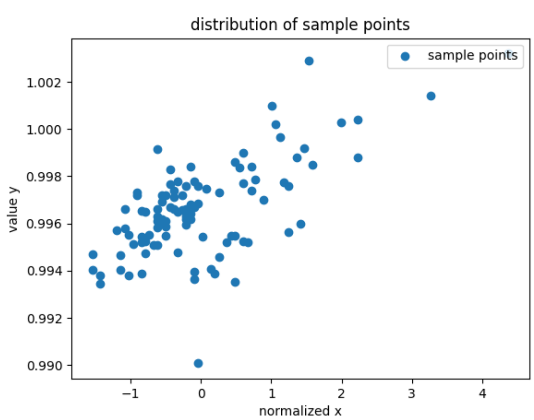
Total epoch : 9206
Final parameters obtained:

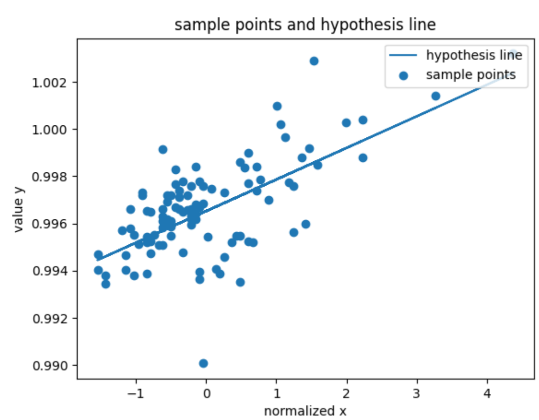
$$\theta_0 = 0.9965203647313094$$

$$\theta_1 = 0.0013400619003951573$$

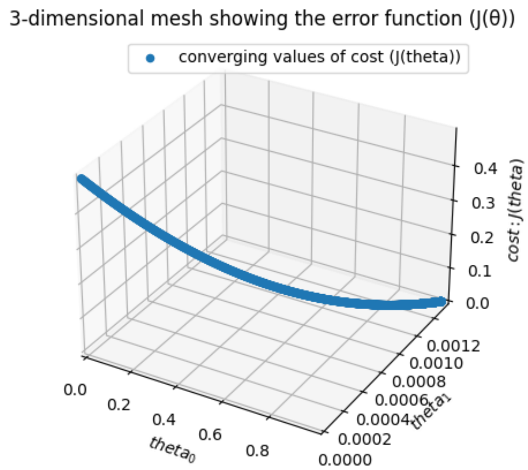
$$FinalCost = 1.1997633818878828 * 10^{-6}$$

(b) Plot of initial data and hypothesis :

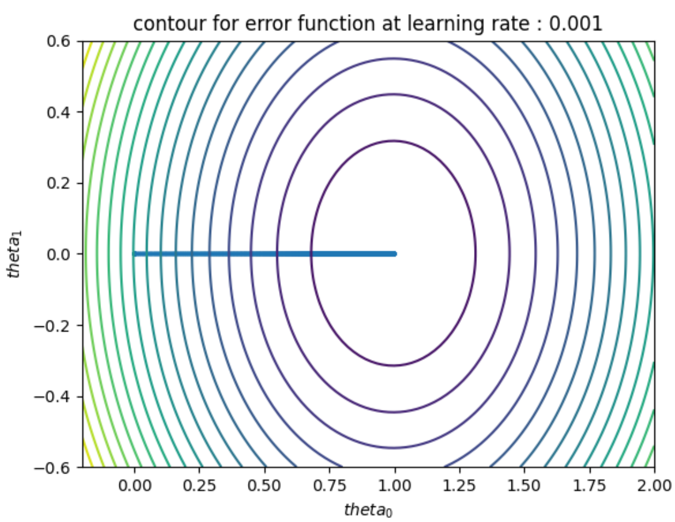




(c) 3D mesh:

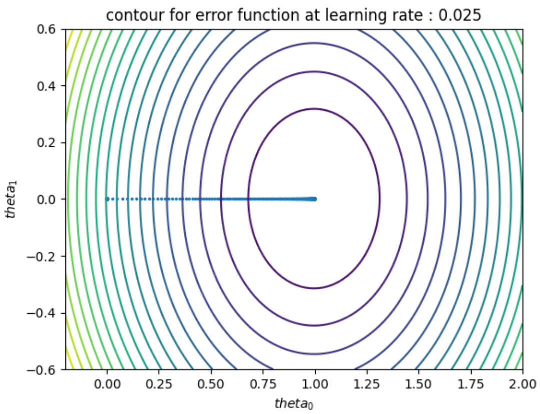


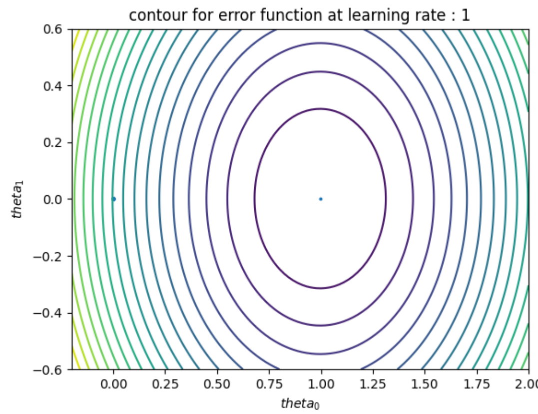
(d) Contour of error function:



(e) Contour for learning rate = 0.001 : given above

Contour for learning rate 0.025 and 0.1 :





Observations : The function is conversing to the center of contour in all cases but it is taking more iterations when learning rate is slow.
For learning rate 0.001, 0.025 and 1, epochs are : 9206, 493, 3
If we learning rate is too fast (3), then it diverges.

2. Sampling and Stochastic Gradient Descent

(a) output stored in the given output directory as **q2aSampleX.csv** and **q2aSampleY.csv**.

(b) Convergence Criteria:

i Batch Size : 1
Learning Rate : 0.0001

$$|J(\theta^{t+1}) - J(\theta^{t})| < 0.1$$

$$\forall j, |{\theta_j}^{t+1} - {\theta_j}^{t}| < 0.1$$

upto 1 updates (epochs)

ii Batch Size : 100
Learning Rate : 0.01

$$|J(\theta^{t+1}) - J(\theta^{t})| < 0.1$$

$$\forall j, |{\theta_j}^{t+1} - {\theta_j}^{t}| < 0.1$$

upto 5 updates (epochs)

iii Batch Size : 10000
Learning Rate : 0.1

$$|J(\theta^{t+1}) - J(\theta^t)| < 0.1$$

$$\forall j, |{\theta_j}^{t+1} - {\theta_j}^t| < 0.1$$

upto 6 updates (epochs)

iv Batch Size : 1000000
Learning Rate : 0.1

$$|J(\theta^{t+1}) - J(\theta^t)| < 0.01$$

$$\forall j, |{\theta_j}^{t+1} - {\theta_j}^t| < 0.01$$

upto 50 updates (epochs)

(c) Yes, for different values of r, converges to the almost same parameter values of theta.
When batch size is very high, it approaches general gradient descent and takes more time.
If we take learning rate too high and batch size too low, it does not converge easily in this case.

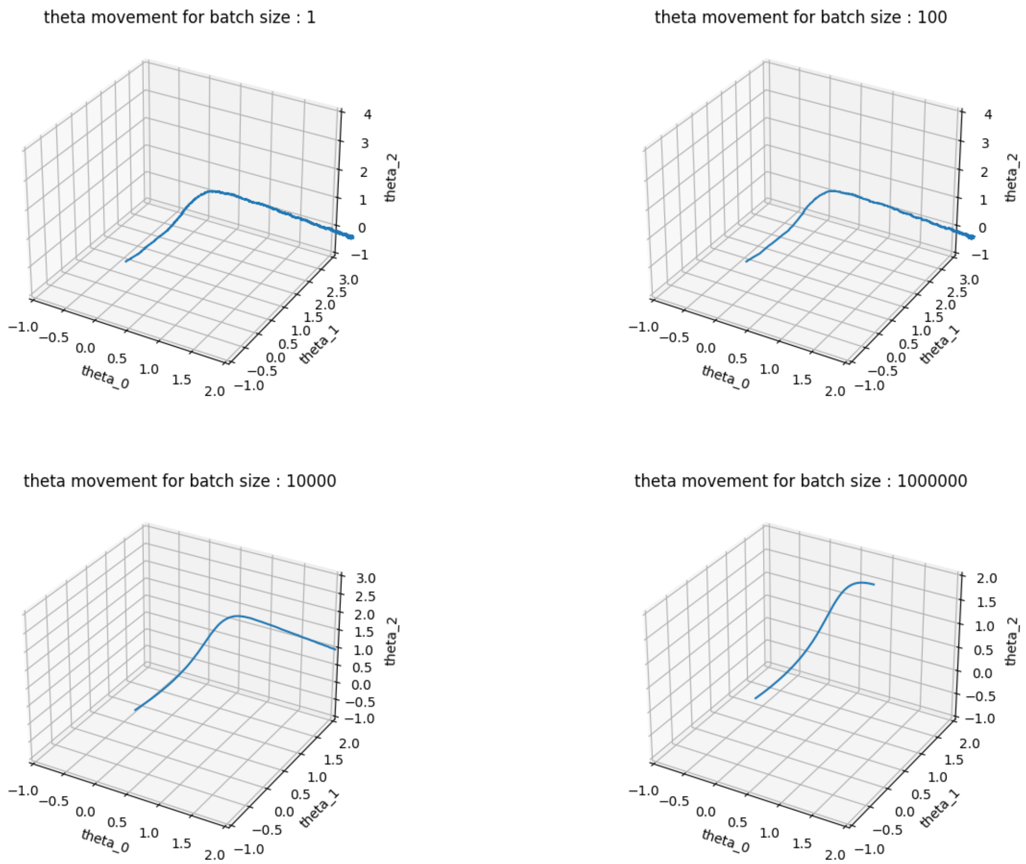
```
For batch size 1, theta is [2.9980863  1.00314631 2.02866509]
For batch size 100, theta is [2.99805865 1.00234634 2.02908017]
For batch size 10000, theta is [3.0047243  1.00597788 2.00343087]
For batch size 1000000, theta is [2.93159107 1.01467623 1.99454614]
```

Total epochs in these cases are : 1, 5, 6, 133
Total num of times, theta parameters updated : 1000000, 50000, 600, 133

Error for q2test.csv :
cost for sample when batchSize is 1 : 1.021094309067669
cost for sample when batchSize is 100 : 1.0221273177075823
cost for sample when batchSize is 10000 : 0.9844782123390015
cost for sample when batchSize is 1000000 : 0.9965394186192018

cost for sample with original theta : 0.9829469215000003

(d) 3d θ movement :



Observations: theta parameters are fluctuating when batch size is too less.
We need to take learning rate low if batch size is low.
Time : When batch size is very high, it is taking more time, it behaves like general gradient descent at very high batch size.

3. Logistic Regression

(a) Applied Newton's method for optimizing $L(\theta)$:
Hessian Matrix :

$$H = X^T D X$$

Matrix D will be : $\begin{bmatrix} \alpha_1(1-\alpha_1) & 0 & 0... & 0 & \\ 0 & \alpha_2(1-\alpha_2) & 0 & ... & 0 \\ . & . & . & . & . \\ 0 & 0 & 0 & ... & \alpha_n(1-\alpha_n) \end{bmatrix}$

where $\alpha = sigmoid(\theta^T X)$

$$\theta^{t+1} = \theta^t - H^{-1}\nabla_\theta J(\theta)|_{\theta^t}$$

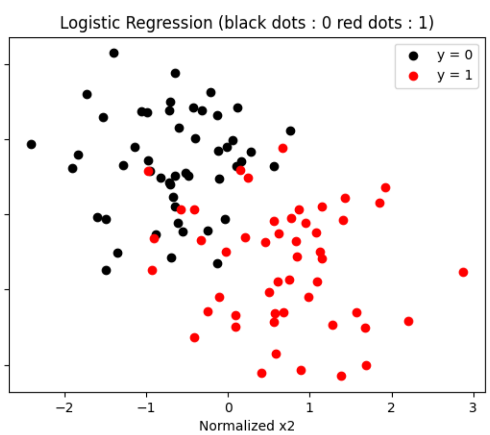
Resulting θ coefficients : [0.40125316 2.5885477, -2.72558849]

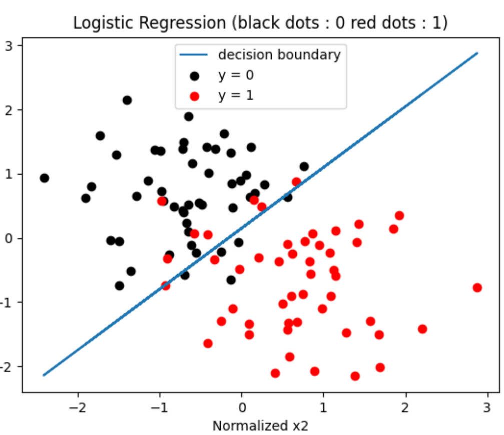
Total epochs : 7

Converging criteria :

$$\forall j, |{\theta_j}^{t+1} - {\theta_j}^{t}| < 0.001$$

(b) Plot with and without prediction boundary:





4. Gaussian Discriminant Analysis

(a) After normalizing the data, we got the following :

$$\phi = 0.5$$

$$\mu_0 = [-0.755294330.68509431]$$

$$\mu_1 = [0.75529433 - 0.68509431]$$

```
Co-variance Matrix :
[[42.95304781 -2.24722795]
[-2.24722795 53.06457925]]
```

We assume both classes share common co-variance matrix, thus, we obtain a linear decision boundary. Formulae used to calculate the following :

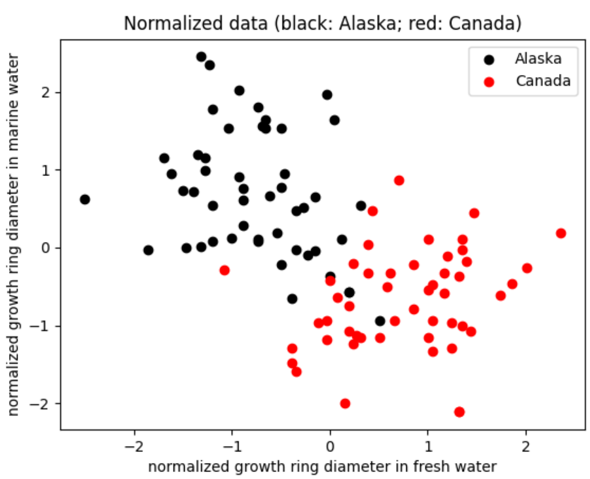
$$\phi = \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)} = 1\}$$

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

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

$$\sum = \frac{1}{m}\sum_{i=1}^{m} (x^{(i)} - \mu_k)(x^{(i)} - \mu_k)^T \text{ where } k = 1\{y^{(i)} = 1\}$$

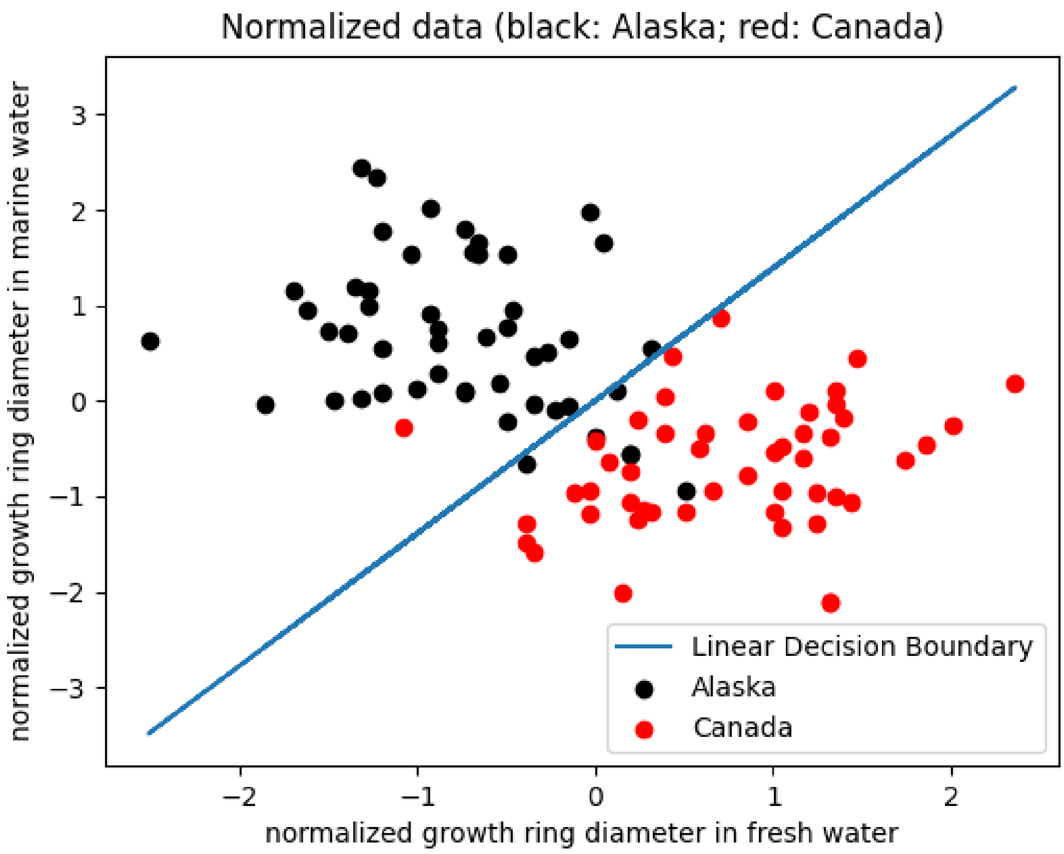
(b) Plot :



(c) We assume that sigma0 and sigma1 to be equal. So GDA becomes logistic regression.
The decision boundary equation:

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

Plot :



(d) μ_0 and μ_1 will be same as above.

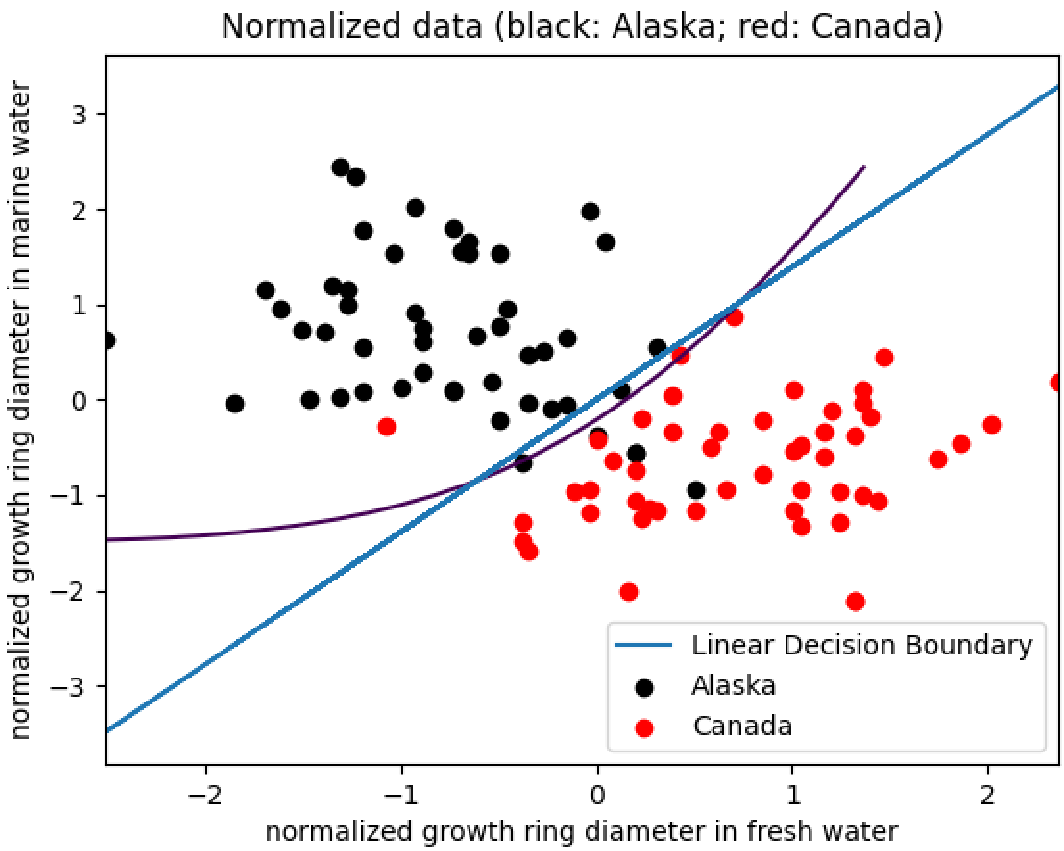
```
sigma0 :
    [[ 0.38158978 -0.15486516]
    [-0.15486516  0.64773717]]

sigma1 :
    [[0.47747117 0.1099206 ]
    [0.1099206  0.41355441]]
```

(e) Decision Boundary will be quadratic in this case, because both the classes have different co-variance matrix. so the 2nd degree term will remain in the equation.

Quadratic decision boundary equation:

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

Plot of sample points with both linear and quadratic hypothesis decistion boundaries:



(f) If we analyze both decision boundaries, we find that quadratic decision boundary is much accurate. Quadratic decision boundary will approach to linear decision boundary if and only if difference between co-variance matrices of both classes approach to each other.

Thanks
Manoj Kumar
2018CS50411