COL774: Assignment 1

Jai Javeria, 2018CS10340

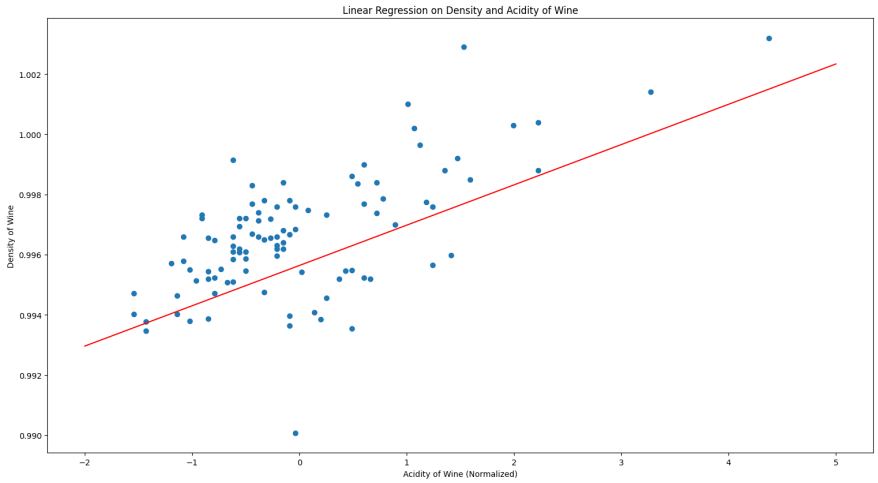
September 2021

1 Q1 Linear Regression

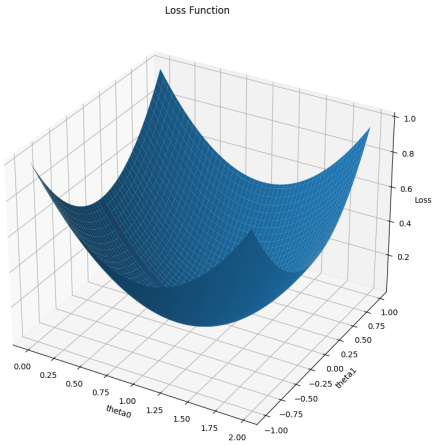
1.1 Part a

- Stopping Criteria: I calculate the loss function J(Theta) over epochs and stop only when the change in 2 subsequent values of loss function is less then epsilon.
- Learning Rate: 0.01 was a good enough learning rate
- epsilon: After experimenting and plotting results, I found epsilon=1e-8 to give a good results.
- Parameters: theta0:- 0.9956 theta1:- 0.001339 epochs taken:- 689

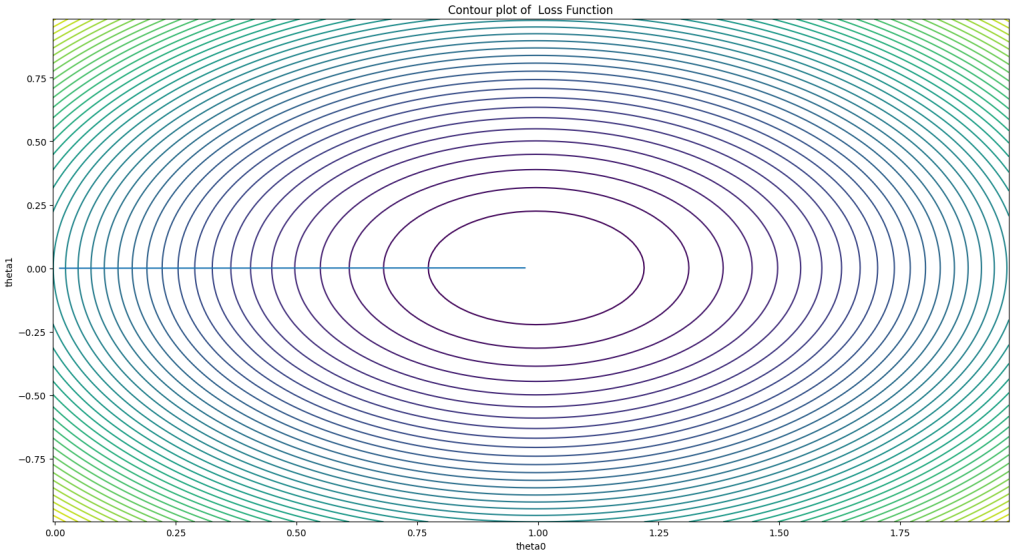
1.2 Part b



1.3 Part c



1.4 Part d



1.5 Part e

- for larger eta, the animation becomes faster since theta is taking larger steps towards the minima whereas for smaller, the updates are slower. The algorithm converges for all learning rates.

2 Q2

2.1 part b

- Stopping Criteria: I run the training for the whole epoch and compute the average JTheta that I got. and then compare it with the previous average and stop when that value is less than epsilon.
- I took epsilon= 10e-4

2.2 part c

- The values I got after experimentation

Batch Size	Learned Theta	epochs	Number of Updates
1	[2.99535658 0.99836632 1.9015134]	3	3*10e6
1e2	[2.99968856 0.99626985 1.99723052]	3	3*10e4
1e4	[2.65544195 1.07550397 1.97548108]	77	7.7*10e3
1e6	[0.64751335 1.45492499 1.65080709]	616	616

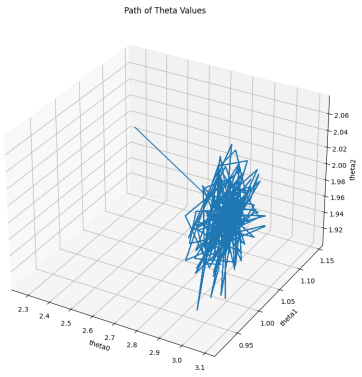
- The number of updates column above is the number of time theta values are updates in the SGD algorithm. It is calculated by epochs*(corpus size/batch size).
- As we can see, for cases with batch size 1e4 and 1e6, the number of updates to theta is low and the values of learned theta is also not that good with respect to the original hypothesis.
- At the same time we see that we get better theta values in batch size=1e2 whereas the number of updates is the biggest for batch size=. Thus there is a tradeoff between batch size and number of updates required for best theta values.
- What may be happening in the case of batch size =1 is that only one example isnt sufficient to model the whole dataset. Thus the learning would become very noisy and batch size should not be as small as 1 in this example.
- The error on original hypothesis(theta=[3,1,2]): 0.9829469215
- The following is the error computed on the predicted hypothesis:

Batch Size	J(Theta)
1	1.490294083726616
1e2	0.9846406592256947
1e4	1.3263278241432426
1e6	18.620590248057493

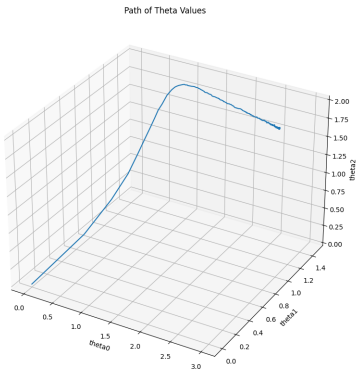
- As was expected, the error value is the least for batch size = 1e2

2.3 Part d

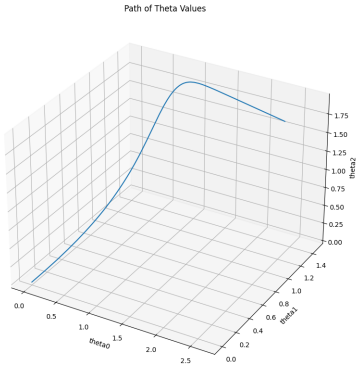
- Batch size = 1
  - For this, the algorithm ran for 3 epochs. For plotting I have taken 200 points in each and plotted 599 points. The starting point theta= is omitted for readability.



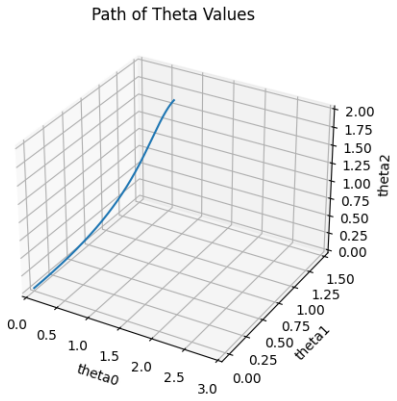
- Batch size = 100
  - Data points collected via same process in above point. theta=0 is also plotted



- Batch size = 1e4
  - For this, the algorithm ran for 77 epochs. For plotting I have taken 10 points in each and plotted 770 points.



- Batch size = 1e6
  - For this, the algorithm ran for 616 epochs. For plotting I have taken 11 points in each and plotted 616 points.



- So as we can see, batch size=1 is really noisy and as we increase the batch size, the graph becomes more smoother.

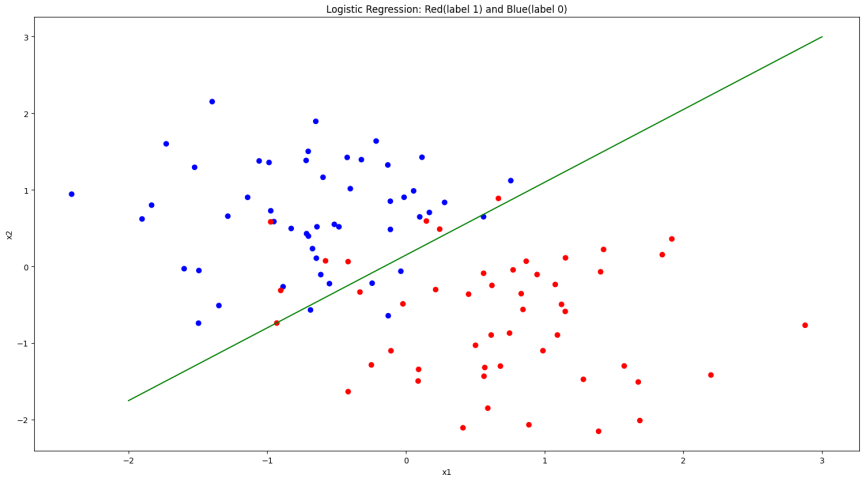
3 Q3

3.1 Part a

- The values of theta that I got are: [0.40125316, 2.5885477, -2.72558849]
- The algorithm converged in 7 epochs. I took epsilon=1e-6 and stopped when difference of 2 consecutive loss function is less than epsilon.

3.2 Part b

- The resultant plot:



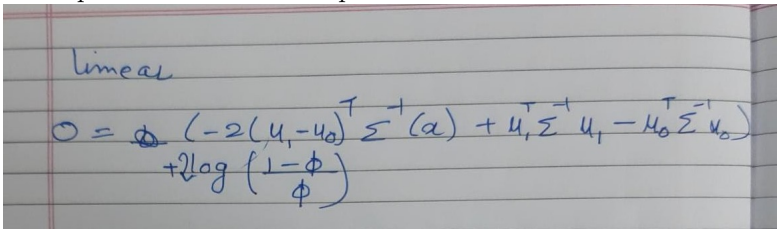
4 Q4

4.1 Part a

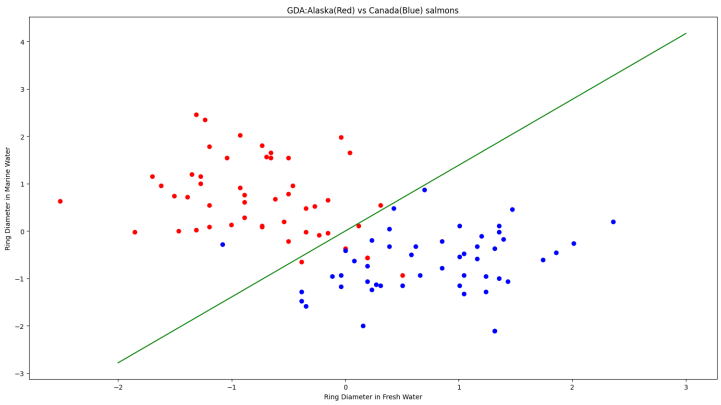
- mu0:[-0.75529433 0.68509431], mu1: [0.75529433 -0.68509431]
- sigma: [[0.42953048 -0.02247228] [-0.02247228 0.53064579]]

4.2 Part b,c

– The equation of the linear separator is:



– The resultant plot is:

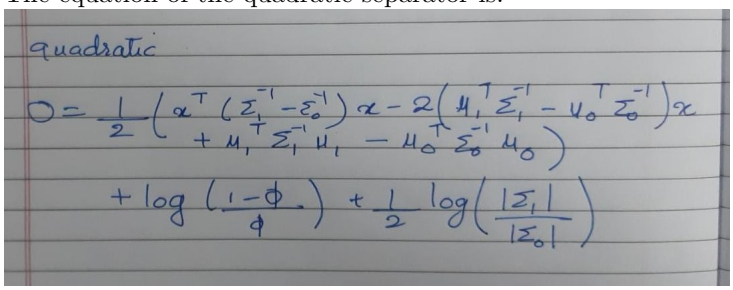


4.3 Part d

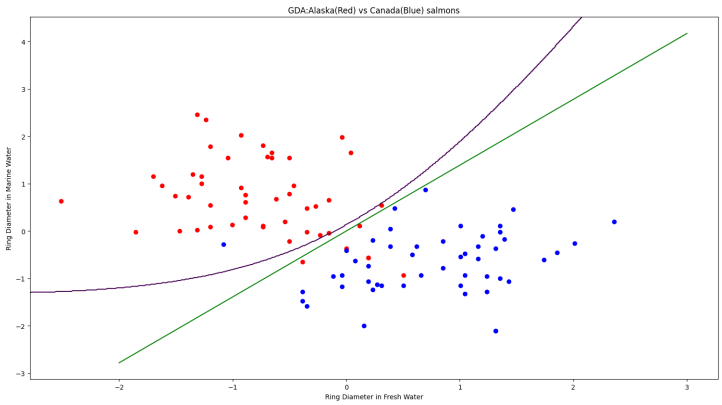
- Sigma1: [[0.38158978 -0.15486516] [-0.15486516 0.64773717]]
- Sigma2: [[0.47747117 0.1099206] [0.1099206 0.41355441]]
- mu0, mu1 are the same

4.4 Part e

- The equation of the quadratic separator is:



- The plot is:



4.5 Part f

- I see that in the given example, both the separators divide the points with roughly same error.