COL774 Assignment 1

Siddhant Mago, 2017CS50419

February 2020

1 Linear Regression

I implemented the least squared error metric and applied gradient descent on the given data set. In this part I have normalised the acidity values before applying gradient descent

1.a Implementation

I used the stopping criteria to be:

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

with $\epsilon = 1e - 10$ and learning rate of 0.01.

The parameters learnt by the algorithm were: $\Theta_0 = 0.9965$, $\Theta_1 = 0.00134$ and the LSE on these parameters was observed to be: 1.199698629377484e-06

1.b Hypothesis function plot

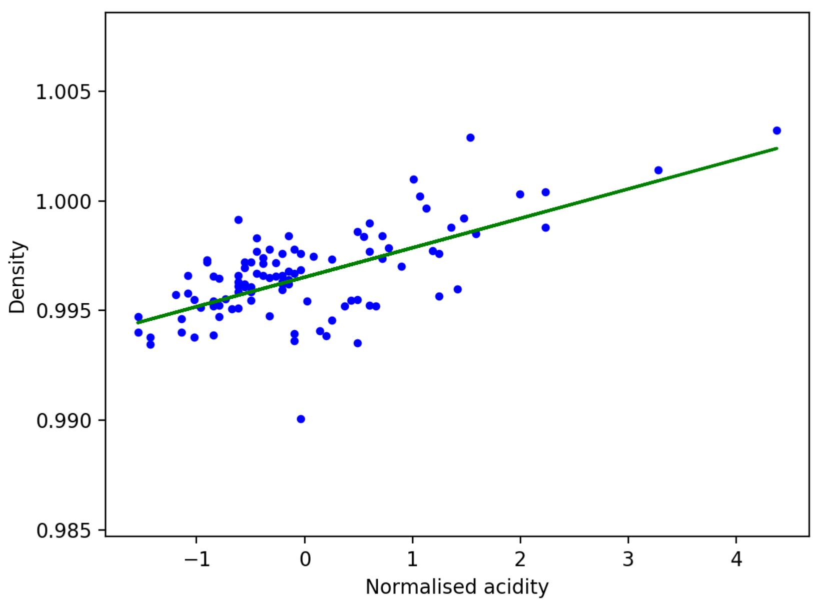


Figure 1: *HypotheisFunction*

The equation of line obtained was 0.00134x + 0.9965

1.c 3-D Mesh of $J(\theta)$

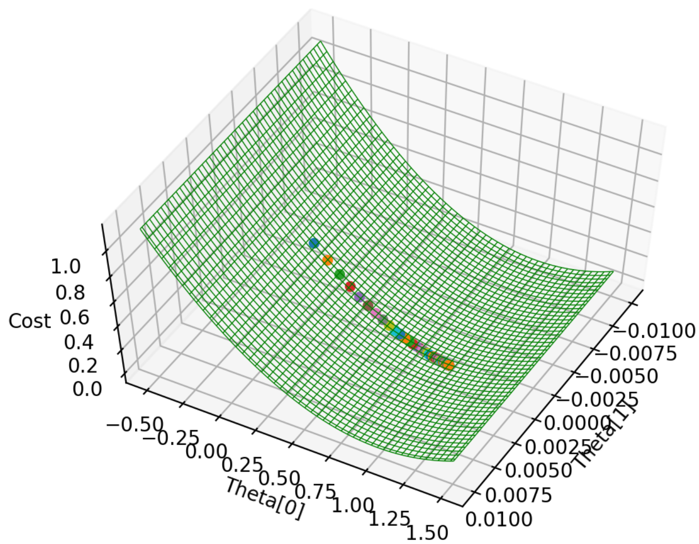


Figure 2: The path taken by gradient descent

This is the final plot obtained, though, the plotting takes place in real time

1.d Contour plot of $J(\theta)$

In this part the learning rate is 0.01

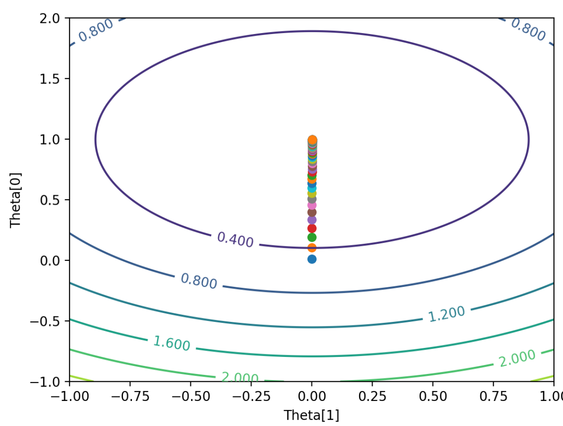
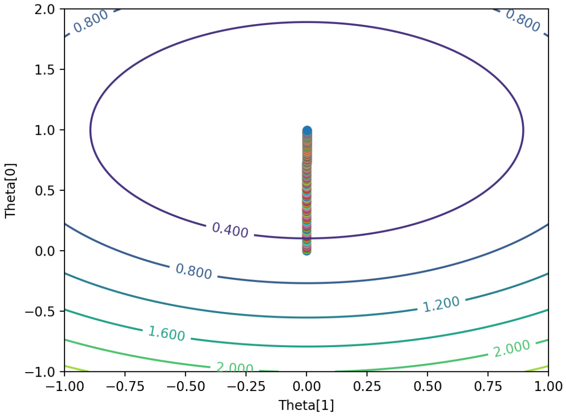


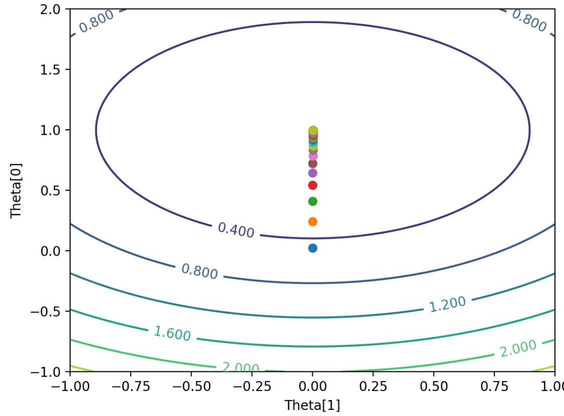
Figure 3: The path taken by gradient descent

1.e Contour plot of $J(\theta)$ at different learning rates

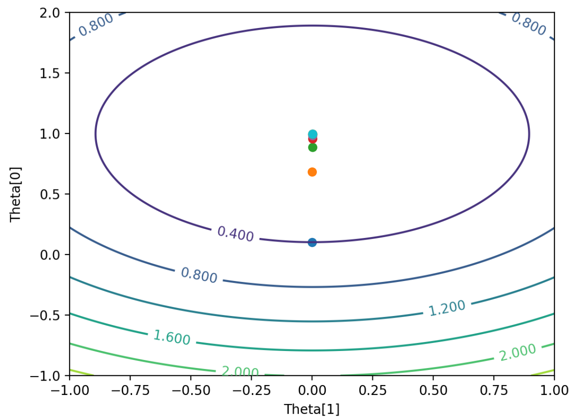
: The Different plots are as shown below



(a) Learning rate = 0.001



(b) Learning rate = 0.025



(c) Learning rate = 0.1

Figure 4: Contours at different learning rates

It is evident from the contour plots that as we increase the learning rate, the convergence becomes faster. It is fastest in the case of learning rate = 0.1 and slowest when learning rate = 0.001. The parameters learnt and the error observed was the same in all the cases.

2 Sampling and Stochastic Gradient Descent

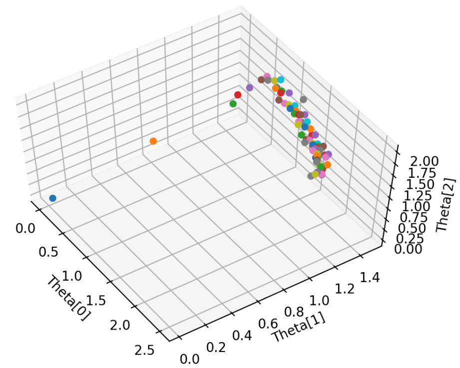
2.a Timings

Batch Size	θ *learnt*	Number of iterations	Time	Average loss(Training	Average loss(Test)
1	$\begin{pmatrix} 3.004 \\ 1.004 \\ 1.991 \end{pmatrix}$	112000	6s	1.001	0.988
100	$\begin{pmatrix} 3.002 \\ 0.994 \\ 1.995 \end{pmatrix}$	36000	6s	1.0007	0.986
10000	$\begin{pmatrix} 2.998 \\ 1.000 \\ 1.999 \end{pmatrix}$	30000	8s	1.000	0.983
1000000	$\begin{pmatrix} 2.998 \\ 1.000 \\ 1.999 \end{pmatrix}$	30000	4min	1.000	0.983

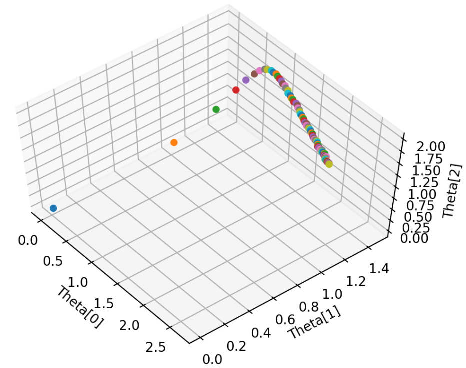
Figure 5: Timings

Error of new data on original hypothesis: **0.9829**.
Different batch sizes converge to different values but the parameters learnt are very close to the original values, however, batch size = 10K and 1M do converge to the same value. This maybe because we have generated the data using normal distribution and 10K is also a big batch size. The stochastic gradient descent with batch size = 1M or 10K converge within the same number of iterations. Batch size 1 takes the most updates followed by batch size 100. In terms of time taken batch size = 1 and 100 take almost the same time but batch size 100 gives slightly better accuracy. Batch size = 1M is slowest.
The observations are as expected because having a smaller batch size means the steps are not that well directed which takes more updates but time per update is very less. As the batch size increases the updates are more well directed but each update takes more time.

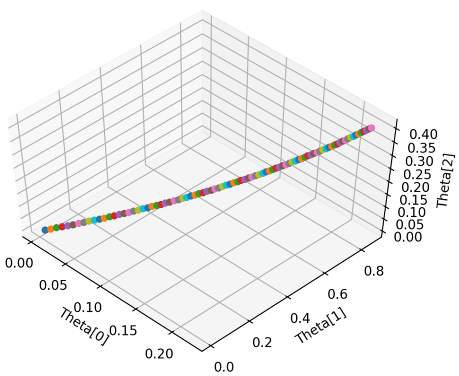
2.b Plots



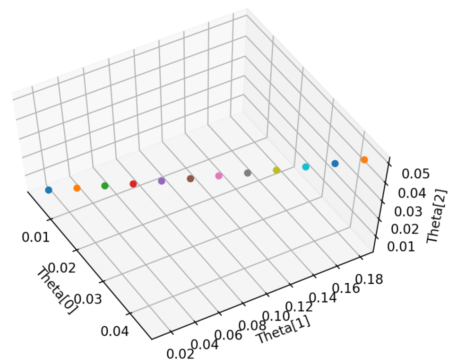
(a) Batch size = 1



(b) Batch size = 100



(c) Batch size = 10000



(d) Batch size = 1000000

Figure 6: Theta values at different batch sizes

3 Logistic Regression:

3.a Implementation

The parameter values obtained, after implementing newton's method, were:

$$\Theta = \begin{pmatrix} 0.4012 \\ 2.588 \\ -2.7255 \end{pmatrix}$$

3.b Plot

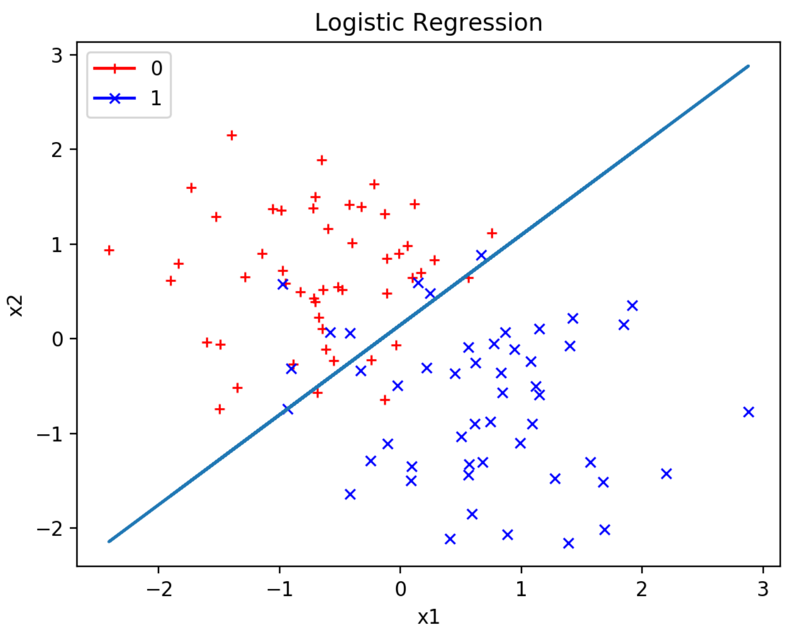


Figure 7: Decision boundary

The equation of the linear boundary is: $0.9497x + 0.147$. The log likelihood was observed to be: -22.83
Newton's method converges quickly and gives a good decision boundary.

4 Gaussian Descriminant Analysis:

Linear boundary equation

$$x^T\Sigma^{-1}(\mu_1 - \mu_0) = \frac{\mu_1^T\Sigma^{-1}\mu_1}{2} - \frac{\mu_0^T\Sigma^{-1}\mu_0}{2} + \log(\frac{1-\phi}{\phi})$$
$$\text{where } x = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}$$

Quadratic boundary equation

$$\log(\frac{\phi}{1-\phi}) + \frac{1}{2}\log(\frac{|\Sigma_0|}{|\Sigma_1|}) + x^T\Sigma_1^{-1}\mu_1 - x^T\Sigma_0^{-1}\mu_0 - \frac{1}{2}x^T\Sigma_1^{-1}x + \frac{1}{2}x^T\Sigma_0^{-1}x - \frac{1}{2}\mu_1^T\Sigma_1^{-1}\mu_1 + \frac{1}{2}\mu_0^T\Sigma_0^{-1}\mu_0 = 0$$
$$\text{where } x = \begin{pmatrix} x_0 \\ x_1 \end{pmatrix}$$

After implementing the values of different constants obtained were:

$$\mu_0 = \begin{pmatrix} -0.755 \\ 0.685 \end{pmatrix}$$

$$\mu_1 = \begin{pmatrix} 0.755 \\ -0.685 \end{pmatrix}$$

$$\textbf{If } \Sigma_1 = \Sigma_0 = \Sigma$$

$$\Sigma = \begin{pmatrix} 0.429 & -0.022 \\ -0.022 & 0.530 \end{pmatrix}$$

$$\textbf{If } \Sigma_1 \neq \Sigma_0$$

$$\Sigma_0 = \begin{pmatrix} 0.381 & -0.154 \\ -0.154 & 0.647 \end{pmatrix}$$

$$\Sigma_1 = \begin{pmatrix} 0.477 & 0.109 \\ 0.109 & 0.413 \end{pmatrix}$$

The plot of the linear and qudratic boundary obtained was:

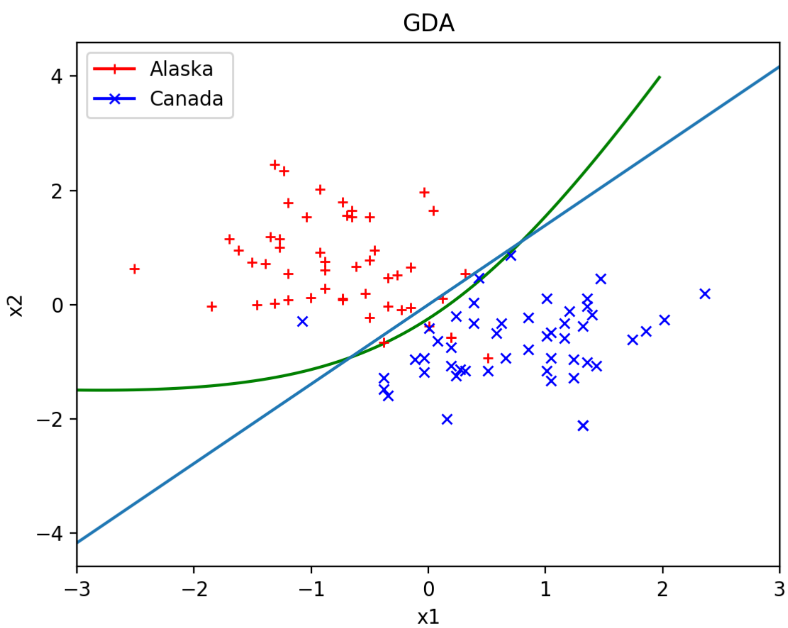


Figure 8: Decision boundaries

The quadratic boundary appears to be a better fit than the linear boundary. As the quadratic case is more general and has more parameters, it can also be said the linear case is under fitting to some extent.