COL774 - Machine Learning: Assignment - 1

Param Khakhar - 2018CS10362

27 October 2020

Introduction

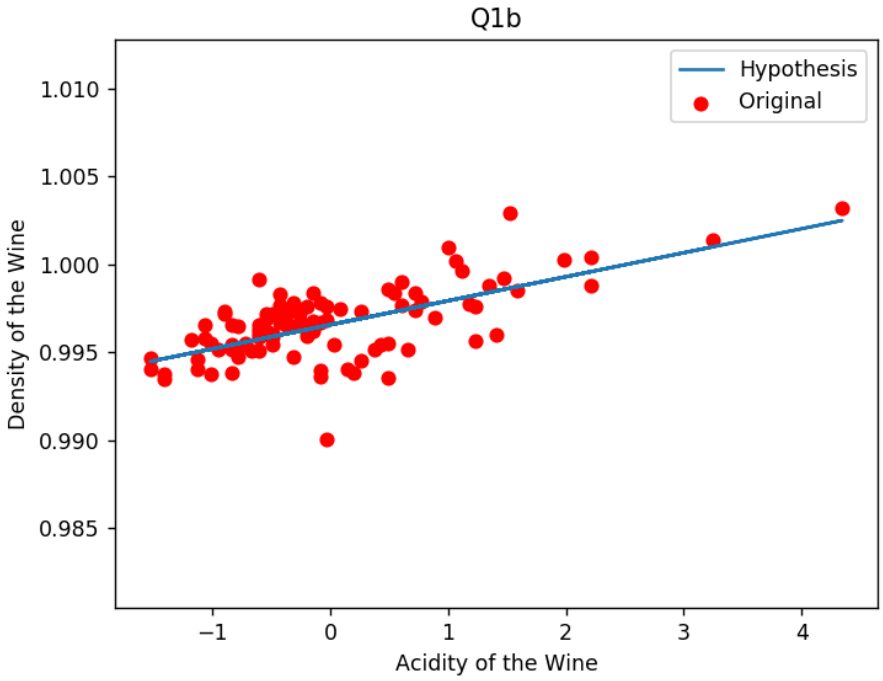
The following is a writeup for the Assignment-1, Machine Learning (COL774). There are sections corresponding to each of the four questions and within each section, there are sub-sections for each sub-part.

Task -1 : Linear Regression

Task - 1.a: Batch Gradient Descent

The theta_parameters are initialized as all zeros. The learning rate of 0.025 is able to converge after 382 iterations. The final value of the parameters is found to be $\theta_0 = 0.99657$ and $\theta_1 = 0.00136$. The stopping criteria used is when $|J(\theta^{t+1}) - J(\theta^{t})| \leq 10^{-10}$.

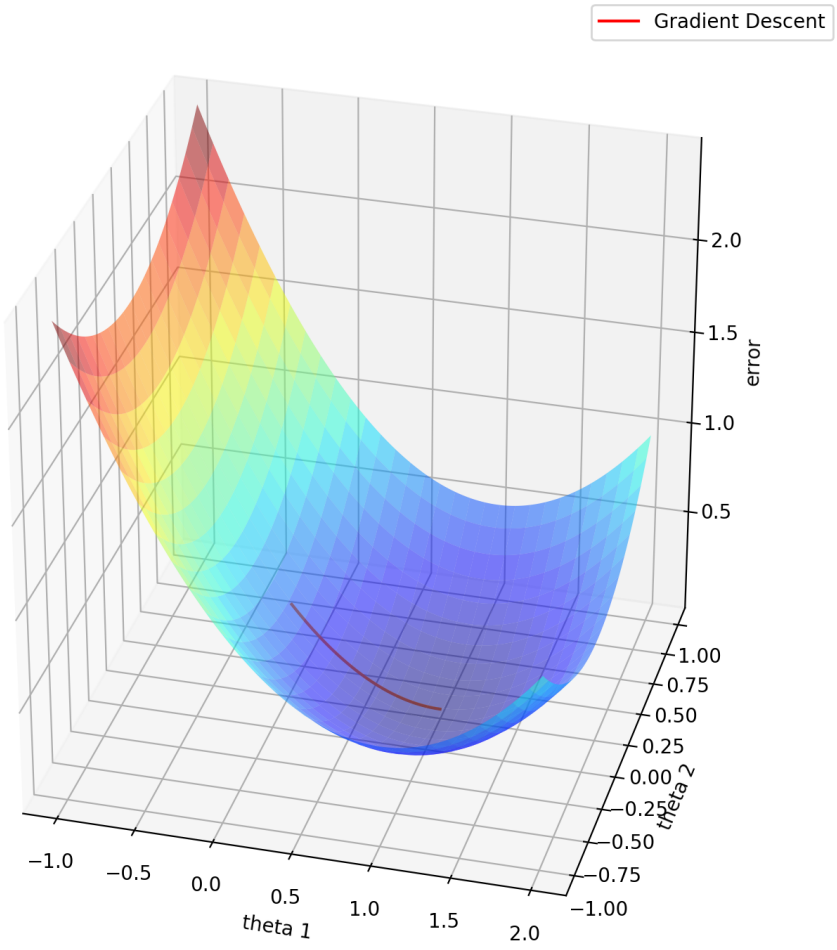
Task -1.b: Plotting the Data Points



The graph is plotted on normalized data as the values of the target variables are very close to each other. The acidity values have been normalized.

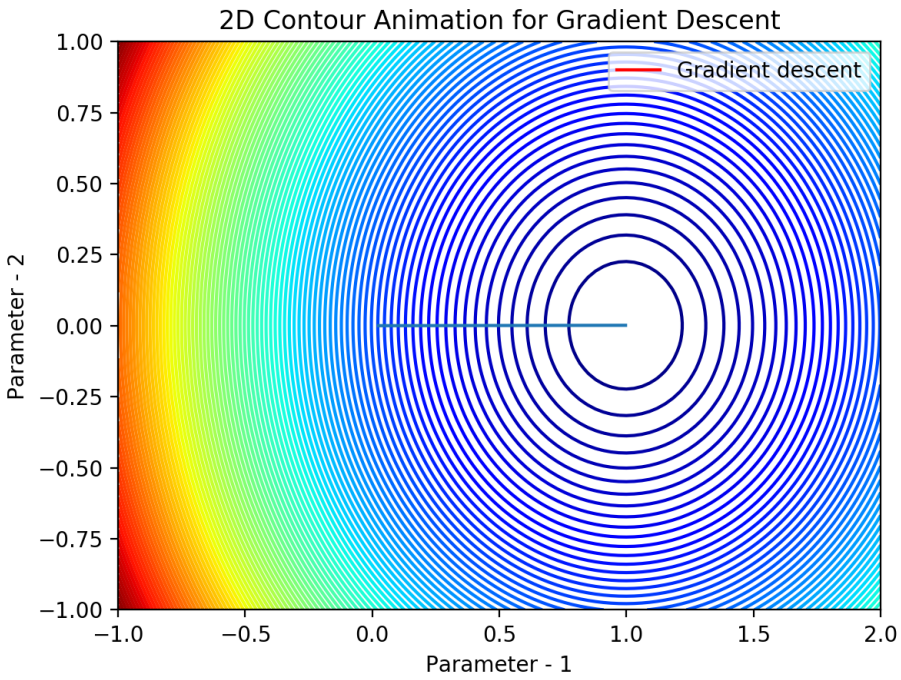
Task - 1.c: Meshgrid animation

The file q1c.mp4 is available in the output directory, which shows the progress. The snapshot is as follows:



Task - 1.d: Contour animation

The file q1d.mp4 is available in the output directory, which shows the progress. The snapshot is as follows:



Task - 1.e: Contour Animation Comparison

As evident in the video, the marker with learning rate = 0.001 is making very slow progress, whereas the markers with learning rate 0.025 and 0.1 have already converged. The video isn't completely save due to lack of efficient computational resources.

The following code was referred while plotting the visualization of contours and 3d mesh, https://xavierbourretsicotte.github.io/animation_ridge.html

Task - 2: Stochastic Gradient Descent

Task - 2.a: Sampling

Numpy's inbuilt functions were used to sample values from a normal distribution.

Task - 2.b: Training

Given the batch size, the data is shuffled randomly, and then divided into batches of size b. During each iteration, cost is computed for every batch, followed by computation of gradients, and updating the theta parameters. The convergence criteria is when for all the batches, the value of the cost in an iteration, and a consecutive iteration, becomes lesser than a particular threshold which is set as 10^{-8}.

Task - 2.c: Training Statistics

The following table indicates the statistics while training the model for different batch sizes, each with a threshold equal to 10^{-8} and a learning rate of 0.001:

Batch Size	**Epochs**	**Time in s**	**Final Parameters**
10^6	20284	523s	[2.989,1.002,1.999]
10^4	277	5.51s	[2.997,1.000,2.000]
10^2	7	1.23s	[3.000,0.9954,2.0013]
1	2	38.4s	[3.007,0.9896,1.9933]

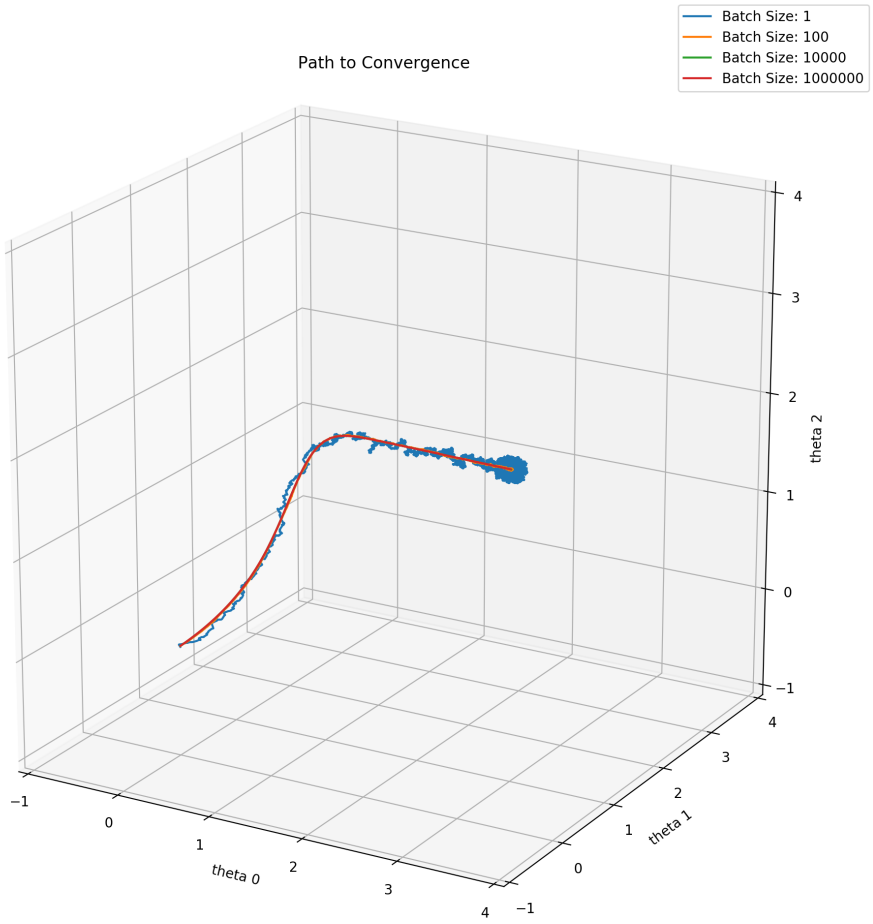
The differences in different batch sizes is clearly visible, in terms of the number of epochs, and the time required in s for convergence. Increasing the batch size, increasing the batch size decreases the time in which the solution converges, upto a certain extent, however for batch size = 1, the time required for convergence, is higher as compared to batch size = 100, and batch size = 10000.

The final value of the parameters in each and every case is very close to the original parameters, which are [3,1,2]. The following table represents the **Mean Squared Error** on the Test Set for different batch sizes and the original hypothesis.

Batch Size	**MSE**
Original Hypothesis	0.9826
10^6	0.9837
10^4	0.9832
10^2	0.9833
1	1.1138

The error for the original hypothesis is the least among all the different learnt parameters for different batches. however, the difference is less, and thereby implying that the learning algorithm's performance is fairly good in learning the parameters.

Task - 2.d

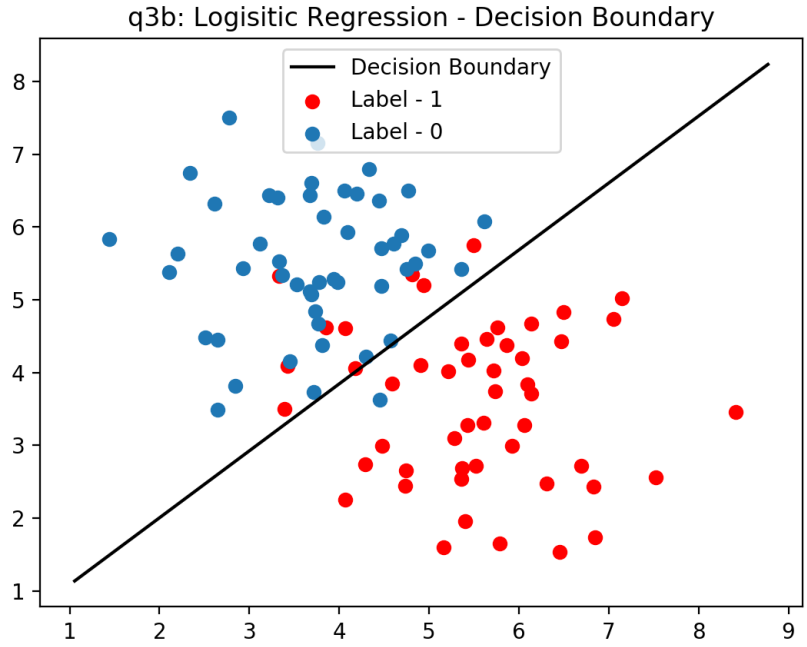


The path taken when the batch size is 1 is very oscillatory. Also, this oscillatory nature of the path decreases as the batch size increases. It makes intuitive sense as increasing the number of examples for calculating the gradient would result in more appropriate direction of the gradient which won't be the case when considering lesser number of samples.

Task - 3: Logistic Regression

The value of optimal parameters obtained using Newton's Method are: $\theta_0 = 0.46722676$, $\theta_1 = 2.57071759$, $\theta_2 = -2.7955926$. The process got completed

in 7 steps, and the threshold for convergence is 10^{-6}. The decision boundary corresponding to the parameters is:



Task - 4: Gaussian Discriminant Analysis

Task - 4.a : Linear Discriminant Analysis

The analytically computed values of the parameters are as follows:

- $\Phi = 0.5$
- $\mu_0 = \begin{bmatrix} 0.75529433 & -0.68509431 \end{bmatrix}$
- $\mu_1 = \begin{bmatrix} -0.75529433 & 0.68509431 \end{bmatrix}$
- $\Sigma = \begin{bmatrix} 0.42953048 & -0.02247228 \\ -0.02247228 & 0.53064579 \end{bmatrix}$

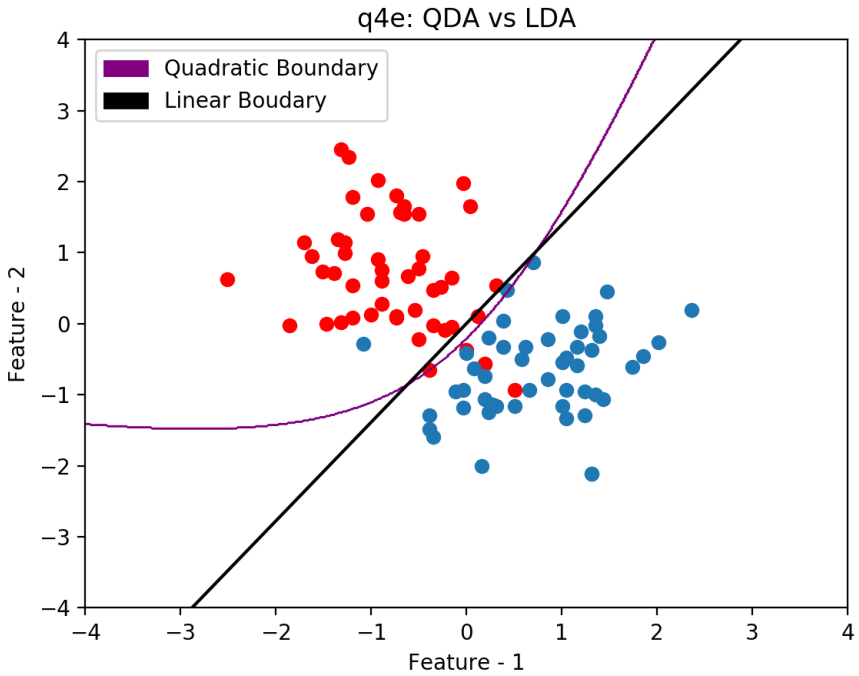
The training data given along with the linear decision boundary is as follows:

Task - 4.d : Quadratic Discriminant Analysis

The analytically computed values of the parameters are as follows:

- $\Phi = 0.5$
- $\mu_0 = \begin{bmatrix} 0.75529433 & -0.68509431 \end{bmatrix}$
- $\mu_1 = \begin{bmatrix} -0.75529433 & 0.68509431 \end{bmatrix}$
- $\Sigma_0 = \begin{bmatrix} 0.47747117 & 0.1099206 \\ 0.1099206 & 0.68509431 \end{bmatrix}$
- $\Sigma_1 = \begin{bmatrix} 0.38158978 & -0.15486516 \\ -0.15486516 & 0.64773717 \end{bmatrix}$

The decision boundary for both linear and quadratic discriminant analysis is as follows:



The quadratic decision boundary is better as compared to the linear decision boundary, as certain points would have been misclassified by the linear boundary, but they are rightly classified by the quadratic decision boundary.