ML Assignment1

Saswat Kumar Pujari
2019MCS2571

10 February 2020

Introduction

The Objective of the assignment was to implement different basic algorithms of Machine Learning. The following four algorithms were implemented:

1. Linear Regression with Batch Gradient Descent
2. Stochastic Gradient Descent
3. Logistic Regression
4. Gaussian Discriminant Analysis

The above algorithms were implemented as per the problem statement and the corresponding observations are illustrated in the following sections.

Question1: Linear Regression

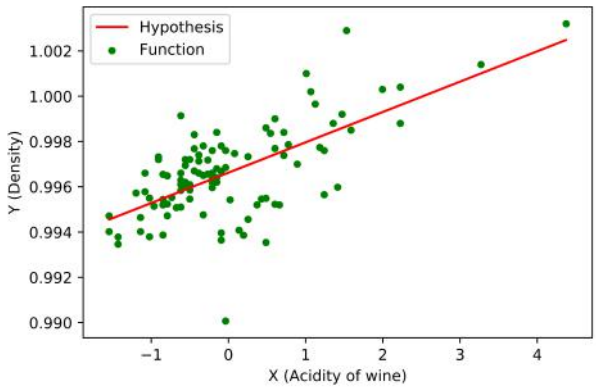
The first task was to implement least squares linear regression using batch gradient descent. For the same, the following parameters were taken:

- **learning rate**: $\boldsymbol{\eta = 0.1}$
- **stopping criteria**: $\boldsymbol{J(\theta^{t-1}) - J(\theta^t)} < 10^{-14}$

i.e. the difference between consecutive cost function value must be less than 10^{-14}.

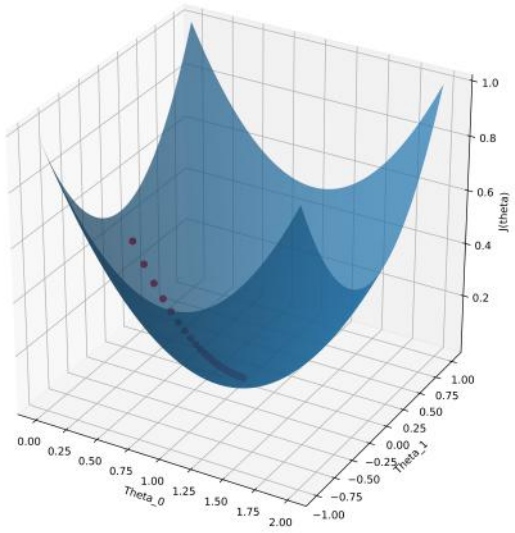
The final set of parameters obtained by the algorithm, i.e.
$\boldsymbol{\theta}$ = **[0.99661981 0.0013402]**

The next task was to **plot the hypothesis function along with the data points** on a two-dimensional graph. The following figure shows the graph:



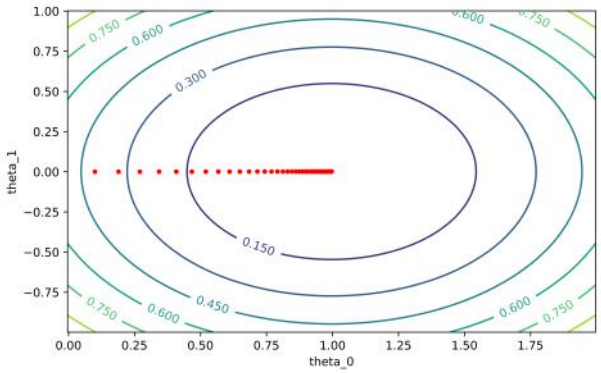
Plot of hypothesis with data

The next task was to draw a 3D mesh showing the error function J(θ) as a function of the parameter vector θ. The same is shown in the figure below:



3D Mesh of J(θ) i.e cost function

The next task was to draw a contour plot of the same cost function i.e. J(θ). The following figure shows the plot:



Contour plot of J(θ) i.e cost function

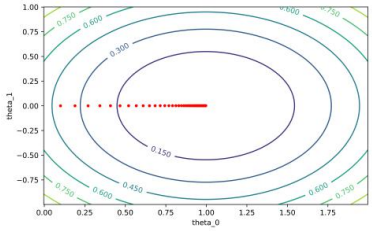
The above task, i.e. drawing of contours was repeated for different values of learning rate, $\boldsymbol{\eta}$ = **0.001, 0.025, 0.1**. The following observations were made on the basis of this experiment:

- As the learning rate increases, the number of epochs taken to minimize the cost function decreases.
- The number of epochs taken to converge is depicted for different learning rates in the table below:

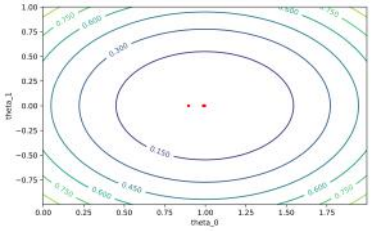
eta(η)	**No. of epochs**
0.001	12656
0.025	565
0.1	143
0.8	11
1	2
1.1	8
1.9	143
2.1	∞

- Until $\eta = 1$, the number of epochs keeps on decreasing, and hence the algorithm seems to converge gradually. After that, for eta > 1, the number of epochs keeps on increasing indicating the cost function oscillates around the minimum value until convergence and after $eta = 2$, it diverges.

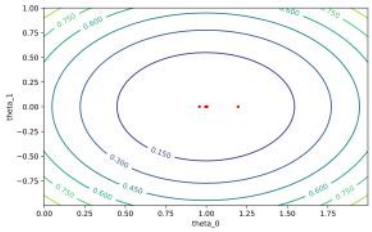
$\eta = 0.1$



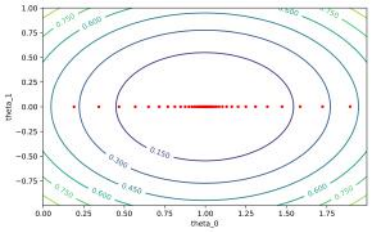
$\eta = 0.9$



$\eta = 1.2$



$\eta = 1.9$



The above figures show the mentioned observations about convergence in case of different learning rated.

Question2: Stochastic Gradient Descent

In the first task, 1 million data points were sampled using the parameters of the given distributions along with error.

After that Stochastic Gradient Descent was implemented on the previously sampled data to calculate new parameters for varying batch sizes with learning rate, $\eta = 0.001$.

Convergence Criteria:

For SGD convergence with different batch sizes, the convergence was decided using the following steps:

1. For every batch, calculate the loss function value of the current batch.
2. The average loss value of a fixed number of batches is taken and compared with the previous average loss value; if the difference is less a certain threshold ϵ, then the algorithm terminates.

   For ex: For **batch size 1**, convergence criteria was set as:

   Average cost of previous 1000 batches - Average cost of current 1000 batches $< \epsilon$

   The number of batches considered for calculating average loss function value and the ϵ value for different batch sizes is depicted in the table below:

Batch Size	**No. of Batches**	ϵ
1	1000	10^{-3}
100	1000	10^{-3}
10000	100	10^{-5}
1000000	1	10^{-6}

3. The θ values obtained in the previous step are the required parameters.

The $\boldsymbol{\theta}$ value learned each time **for varying batch sizes** is depicted in the table below:

Batch Size(r)	**Theta0**($\boldsymbol{\theta}_0$)	**Theta1**($\boldsymbol{\theta}_1$)	**Theta0**($\boldsymbol{\theta}$)$_2$
1	3.02832309	1.0036189	2.01222021
100	2.9531843	1.01192223	1.99126867
10000	2.96439926	1.00731504	1.99797805
1000000	2.88577473	1.02558348	1.9915187

Observations from the above experiment

- We see that different models converge to different parameter(θ) values. The reason for this is that stochastic gradient oscillates around the local minimum instead of converging to the local minima. Also as there are different criteria of convergence for different batch sizes, the parameter values are different.
- Also the parameters of all the different batches are different from the original parameters of the hypothesis for the sampled data set. This is because of the sampled error value ϵ induced into all the data points of the sampled data set.
- The number of iterations taken for the different batches is given below:

Batch Size(r)	**No of iterations**
1	30000
100	15000
10000	16100
1000000	11774

  We can see that the number of examples needed to be fit the parameters increases with increase in batch size, hence the convergence rate slows down with increasing batch size.

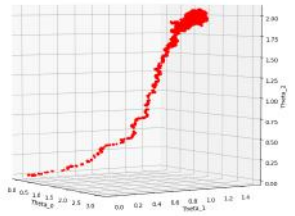
The next task was to test the above models obtained for different batch sizes on a given sampled test data by calculating the error for different models and also for the given parameters, i.e [3 1 2]. The following error values were calculated:

Error on original parameters	0.9829469215
Error on model for batch size 1	0.9892113282105423
Error on model for batch size 100	0.9861868621550188
Error on model for batch size 10000	0.9861868621550188
Error on model for batch size 1 million	1.0225003651318236

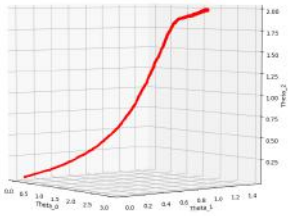
Here, we can see that the error values for different batches are relatively similar to the error values obtained from the parameters of the original hypothesis.

The next task was to plot the movement of $\boldsymbol{\theta}$ for different batch sizes on a 3D plot. The following figures show the same:-

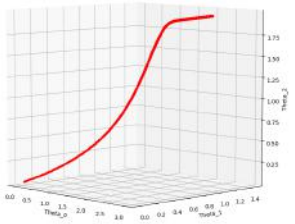
Batch size = 1



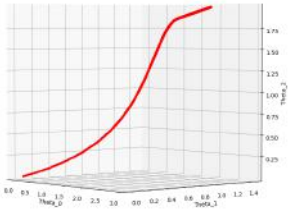
Batch size = 100



Batch size = 10000



Batch size = 1 million



From the above figures, we can observe that:

- The algorithm takes less no. of steps when the batch size is smaller. So, the points in the curve are far apart and the curve is less smoother. But as the batch size increases, large no. of examples are seen and theta is updated very slowly; hence the curve becomes smoother.
- The theta(θ) values oscillate more when batch size is 1, but there is less oscillation when batch size becomes larger. Hence, the curve is smoother and also the average cost difference between two consecutive batches also decreases as they converge towards the minimum as observed in ϵ values in the table above.

- Also the shape of the curve is relatively similar in all the curves showing that the algorithm converges faster initially and as it approaches the minimum, it becomes slower which is the property of gradient descent.

Question3: Logistic Regression

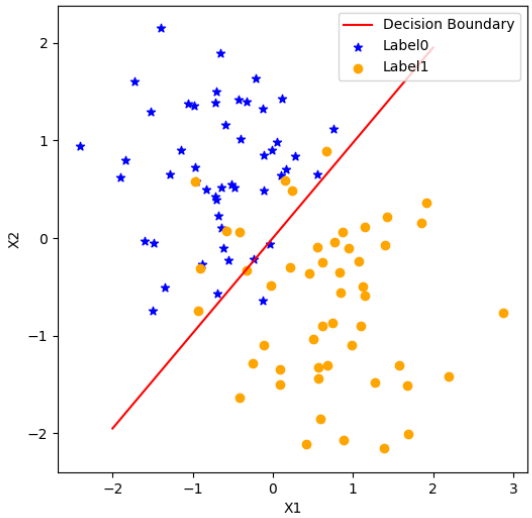
The first task was to implement Logistic Regression using Newton's method of convergence. For this, the equations of the gradient vector and Hessian matrix obtained are given below:

Gradient,$\nabla \mathbf{J}(\boldsymbol{\theta})$ $= X^T(Y - h_\theta(x))$
Hessian,H = -$\mathrm{X}^T(diag(h_\theta(x)(1 - h_\theta(x))))X$

The parameter vector,$\boldsymbol{\theta}$ derived from the above method is given as:

$$\boldsymbol{\theta} = \begin{bmatrix} 2.76358234e-16 \\ 8.97341408e-01 \\ -9.20112283e-01 \end{bmatrix}$$

The next task was to plot the training data along with the decision boundary on a 2D graph. The same is shown in the figure below:



Plot of Decision boundary and data

Question4: Gaussian Discriminant Analysis

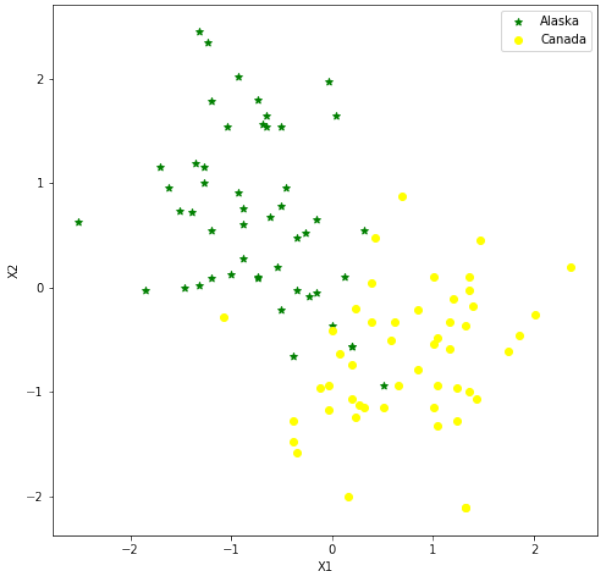
The first task was to implement GDA by finding out the values of $\boldsymbol{\mu}_0$(for Alaska), $\boldsymbol{\mu}_1$(for Canada) and $\boldsymbol{\Sigma}$. The following are the values:

$$\boldsymbol{\mu}_0 = \begin{bmatrix} -0.75529433 \\ 0.68509431 \end{bmatrix}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.75529433 \\ -0.68509431 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{bmatrix} 0.42953048 & -0.02247228 \\ -0.02247228 & 0.53064579 \end{bmatrix}$$

The next task was to plot the data points showing difference between the data for Alaska and Canada. The following shows the same:



Plot of data for GDA

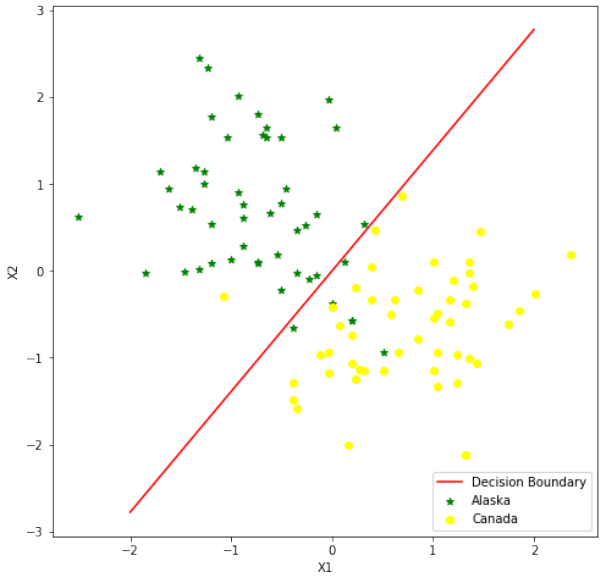
The next task is to define the equation for the decision boundary separating both classes. The **equation for the decision boundary** is given as:

$$\mathrm{P}(\mathrm{y}^{(i)} = 0|x;\mu_0,\Sigma) = \mathrm{P}(\mathrm{y}^{(i)} = 1|x;\mu_1,\Sigma)$$

or

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

The plot of the decision boundary is shown in the graph below:



Plot of Linear Decision Boundary with data

Next task is to implement GDA for unequal covariance matrix, i.e. $\Sigma_1 \neq \Sigma_0$.

The following are the values of the parameters:

$$\boldsymbol{\mu}_0 = \begin{bmatrix} -0.75529433 \\ 0.68509431 \end{bmatrix}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} 0.75529433 \\ -0.68509431 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_0 = \begin{bmatrix} 0.38158978 & -0.15486516 \\ -0.15486516 & 0.64773717 \end{bmatrix}$$

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.47747117 & 0.1099206 \\ 0.1099206 & 0.41355441 \end{bmatrix}$$

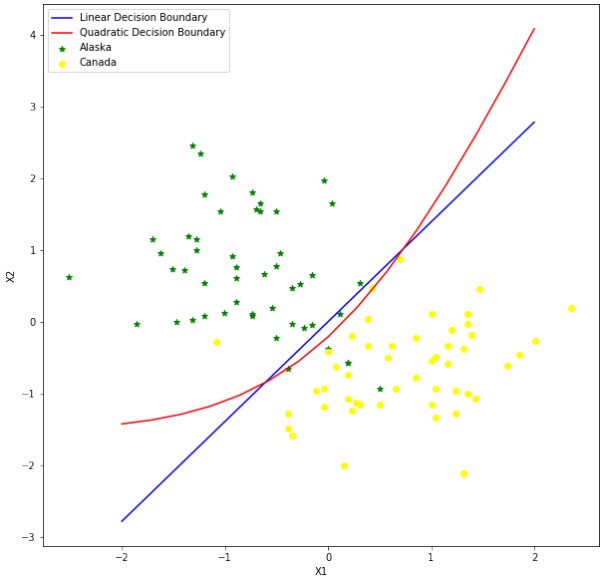
The **equation of decision boundary(quadratic in this case)** is given below:

$P(y^{(i)} = 0|x;\mu_0,\Sigma_0) = P(y^{(i)} = 1|x;\mu_1,\Sigma_1)$

or

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

The linear and quadratic decision boundaries is shown in the graph below:



Plot of Linear Decision Boundary with data

From the above figure, we can observe that the quadratic decision boundary seems to separate the data of different classes more aptly than linear decision boundary.