Ayush Madhan Sohini

2019156

ML- Assignment 1

Q1 $Y = X\theta + \epsilon \rightarrow$ Loss $= \epsilon = Y - X\theta$

$Y \rightarrow$ Vector of predicted values
$X \rightarrow$ Vector of i/p features
$\epsilon \rightarrow$ error value

The loss function needed to be minimized $\Rightarrow$

$$L(\theta) = \frac{1}{2N} \sum_{i=1}^{N} \left(y(x^i) - x^i\theta \right)^2$$ $\rightarrow$ Mean Squared Loss

$x_i \in R^d \rightarrow i^{th}$ sample from data set of size N

In vector form $\rightarrow$

$$L(\theta) = \frac{1}{2N} (Y - X\theta)^2$$

Y ↓ predicted

$X\theta$ ↳ label vector

$$= \frac{1}{2N} (Y - X\theta)^T (Y - X\theta)$$

$$= \frac{1}{2N} \left(Y^T - (X\theta)^T \right) (Y - X\theta)$$

$$= \frac{1}{2N}\left(Y^T - \theta^T X^T\right)\left(Y - X\theta\right)$$

$$= \frac{1}{2N}\left(Y^T Y - Y^T X\theta - \theta^T X^T Y + \theta^T X^T X\theta\right)$$

$$\frac{\partial L(\theta)}{\partial \theta} = \frac{1}{2N}\left[-X^T Y - X^T Y + 2X^T X\theta\right]$$

$$= \frac{1}{2N}\left[2X^T X\theta - 2X^T Y\right]$$

To move in the direction of the optimal solution, we equate the derivative to zero

$$\frac{1}{2N}\left[2X^T X\theta - 2X^T Y\right] = 0$$

$$\therefore X^T X\theta = X^T Y$$

$$\theta = (X^T X)^{-1} X^T Y$$

Q2 Whenever the inverse of $X^T X$ exists, the closed form solution exists. If $X^T X$ is a singular matrix, the closed form solution won't exist.

Q3 The closed form solution is a better option ONLY when the size of the i/p matrix X is small or X is sparse. When X is a very large (suppose it has 10^5 entries), $X^T X$ would be a $10^5 \times 10^5$ matrix ie it has 10^{10} entries, which would be very difficult to store. Also performing $(X^T X)^{-1}$ is also computationally inefficient on such a large matrix. Also if $X^T X$ is singular, the inverse doesn't exist anyways. In such cases, the iterative methods like gradient descent are a better choice.

Q4
$$y(x^i) = \theta_0 + x_1^i \theta_1 + x_2^i \theta_2 \ldots x_n^i \theta_n + \epsilon^i$$

$$y(x^1) = \theta_0 + x_1^1 \theta_1 + x_2^1 \theta_2 \ldots x_n^1 \theta_n + \epsilon^1$$

$$\vdots$$

$$y(x^m) = \theta_0 + x_1^m \theta_1 \ldots\ldots x_n^m \theta_n + \epsilon^m$$

$$y(x^1) + \dots y(x^m) = m\theta_0 + \theta_1 \sum_{i=1} x_1^i$$

$$+ \theta_2 \sum_{i=1}^{m} x_2^i \quad \dots ,$$

$$+ \theta_n \sum_{i=1}^{m} x_n^i + \sum_{i=1}^{m} \epsilon^i$$

Dividing both sides by m

$$\frac{y(x^1) + \dots y(x^m)}{m} = \theta_0 + \frac{\theta_1 \sum_{i=1}^{m} x_1^i}{m}$$

$$\dots + \frac{\theta_n \sum_{i=1}^{m} x_n^i}{m} + \boxed{\frac{\sum_{i=1}^{m} \epsilon^i}{m}}$$

$= 0$ as ϵ follows a zero mean gaussian

$$\therefore \bar{Y} = \theta \bar{X} + \theta_0 \longrightarrow \text{eqn of a line}$$

where $\bar{X} = \begin{bmatrix} \sum_{i=1}^{m} x_1^i \\ \vdots \\ \sum_{i=1}^{m} x_n^i \end{bmatrix} \times \frac{1}{m}$

$$\Theta = [\Theta_1 \ \Theta_2 \ldots \Theta_n]$$

$$\bar{Y} = \begin{bmatrix} y(x^1) \\ \vdots \\ y(x^m) \end{bmatrix} \times \frac{1}{m}$$

Q5 Linear Regression model throws a continuous output. So if we can set a particular threshold ie if the o/p of the linear regressor is above a threshold then it belongs to class A else class B, it can work as a binary classifier. The threshold values can be determined using tools like the ROC-AUC curve. However the data is rarely distributed as a gaussian, so this is not a good classifier as in linear regression the error in the data is assumed to be a gaussian.