Linear Regression (2D) [Linear Relationship]

Equation of line : $y = mx + c \longrightarrow y = w_1 x + w_0$ (helps in generalization)

Let there be 'n' data points in our data.

So 'n' (x, y) pairs.

Our goal is to compute w_0 & w_1 in order to obtain the line.

Writing in vector form,

$$\begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}_{n\times 1} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}_{n\times 2} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}_{2\times 1} \Rightarrow Y = XW$$

~~We~~ Multiplying both sides with X^T to make square matrix.

$$X^T Y = X^T X W$$

$$\therefore \boxed{W = (X^T X)^{-1} X^T Y} \quad -(1)$$

Closed form solution with MSE loss optimization.

Closed form solution is not ideal when 'n' tends to be very large. In which case we will be using gradient descent based approach.

We need to optimize for, Mean square error loss (MSE),

ie minimize $\frac{1}{2} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

Truth Predicted

$$\Rightarrow \text{minimize } \frac{1}{2} \sum_{i=1}^{n} (y_i - (w_1 x_i + w_0))^2$$

$$\Rightarrow E = \min \frac{1}{2}\left[(y_1 - (w_1 x_1 + w_0))^2 + \dots + (y_n - (w_1 x_n + w_0))^2 \right]$$

In order to minimize the loss, we need to obtain partial derivatives w.r.t w_0 & w_1. ~~and make them~~

$$\therefore \frac{\partial E}{\partial w_0} = \left[\left(y_1 - (w_1 x_1 + w_0)\right)(-1) + \dots + \left(y_n - (w_1 x_n + w_0)\right)(-1) \right]$$

$$\frac{\partial E}{\partial w_1} = \left[\left(y_1 - (w_1 x_1 + w_0)\right)(-x_1) + \dots + \left(y_n - (w_1 x_n + w_0)\right)(-x_n) \right]$$

To obtain the updated weights,

$$\begin{bmatrix} w_0(new) \\ w_1(new) \end{bmatrix} = \begin{bmatrix} w_0(old) \\ w_0(old) \end{bmatrix} - \eta \begin{bmatrix} \frac{\partial E}{\partial w_0} \\ \frac{\partial E}{\partial w_1} \end{bmatrix}$$

$\eta \rightarrow$ Learning Rate.

It can be proved that, $\partial E = \begin{bmatrix} \frac{\partial E}{\partial w_0} \\ \frac{\partial E}{\partial w_1} \end{bmatrix} = X^T(X\theta - Y)$

So $\boxed{W_{new} = W_{old} - \eta\, \partial E}$ — (2)

Iterate for 'k' epochs, closing in the real w_0 & w_1 with every epoch.

Both (1) & (2) are generalizable to Multiple Linear Regression (having more than 1 features) by just changing X.

$$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}$$ for 'p' features

Ex, Age

Ex, height

Data Normalization

If the features in your data have large Variance difference among them then it is possible that the weight corresponding to the less Variance never gets ~~updatter~~ updated at all (or updated very small) during gradient descent. This is because we have same learning rates for all the feature weights.

One way to prevent this is by normalizing the features so that all of them are on a similar scale.

One of the most popular noramalization technique is Zscore.

$$Zscore = \frac{x - \bar{x}}{\sigma_x}$$

It distributes the data such that it has zero mean and unity deviation.

So before applying gradient descent method to obtain line, always normalize the features. Normalization of ~~targeg~~ target(y) is not required.

Train Test Split

You verify how well your line has fit the data by using a test dataset, which is not used while training the Regression Model. Use SKlearn.

Statistical Linear Regression

$$y - \bar{y} = b_{yx}(x - \bar{x})$$

Where $b_{yx} = r\left(\frac{\sigma_y}{\sigma_x}\right)$

$r \Rightarrow$ Coefficient of Correlation. Measure of how well one variable is related to other. Value lies in interval $[-1, 1]$

$r = -1 \Rightarrow$ ~~Strong~~ perfect −ve correlation

$r = 0 \Rightarrow$ No correlation

$r = 1 \Rightarrow$ perfect +ve correlation.

for $|r| > 0.85$, Linear Regression predictions works well.

$$r = \frac{\text{Covar}(x, y)}{\sigma_x \sigma_y}$$

$$\text{Covar}(x, y) = \text{Covariance}(x, y) = \Sigma (x_i - \bar{x})(y_i - \bar{y})$$

$$\sigma_x = \text{Standard deviation} = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{N}}$$

$$\bar{x} = \text{Mean} = \frac{\Sigma x_i}{N}$$