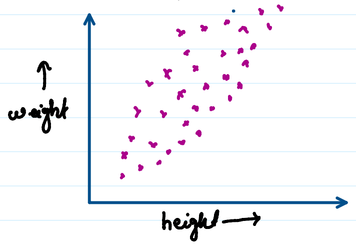
Linear Regression
12 April 2022 09:02 PM

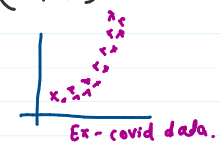
Assumption: x and y has a linear Relation



Linear Regression
- Easy to Learn → Intuition
- Application in other Algo

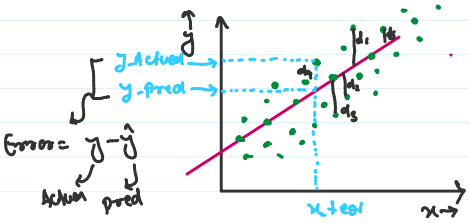
LR (Supervised) < Regression / classification

- Linear LR (when data is linear)
  - Simple: 1 - input, 1 - output
  - Multiple: 1 < input, 0 - output.
- Polynomial LR (when data is not linear)



Linear Regression
→ Aim is to find best fit line → Minimum Error.

Error = Actual Value − Prediction Value.



$$\text{Error} = d_1 + d_2 + d_3 + \cdots d_n$$
$$= (y_1 - \hat{y}_1) + (y_2 - \hat{y}_2) + (y_3 - \hat{y}_3) + \cdots$$
$$= \sum_{i=1} (y_i - \hat{y}_i)$$

But to manage +ve and -ve Error we have to square the Error.

$$\boxed{\therefore \underset{\text{(Loss function)}}{\text{Error}(J)} = \sum_{i=1} (y_i - \hat{y}_i)^2} \quad \text{— eq (1)}$$

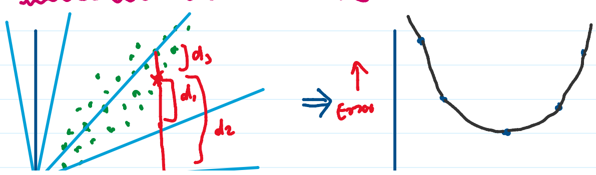
We know $\hat{y} = mx + c$
→ by puting in eq (1)

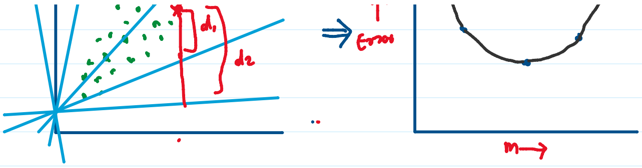
$$\boxed{\text{Loss function } (J) = \sum_{i=1}^{n} (y_i - mx_i - c)^2}$$

Here,
y_i, x_i → Const, we get in training time
m, c → variable, we have to calculate.

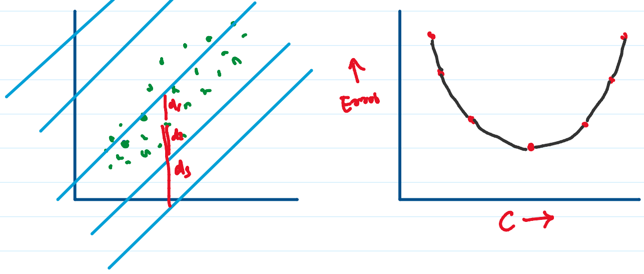
Let's Analyse the m and c w.r.t Loss-function

Relation bet M & loss-fun. (c = const)

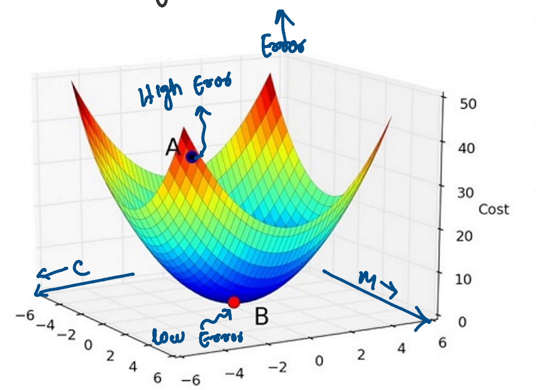




Relation bet c and loss-fun (m = const)



If we Analyse m and c both together we will get the figure like below.



we can find the point of low Error by differentiating the loss Function

Since for minima → $f'(x) = 0$

$$f(x) = \text{loss function}(J) = \sum_{i=1}^{n} (y_i - mx_i - c)^2$$

variable

Since we have 2 variable we have to do partial derivative → $\frac{dJ}{dc} = 0$ and $\frac{dJ}{dm} = 0$

$$\frac{dJ}{dc} = \frac{d}{dc}\left(\sum (y_i - mx_i - c)^2\right) = 0$$

$$\Rightarrow \sum \frac{d}{dc} (y_i - mx_i - c)^2 = 0$$

$$\Rightarrow \sum 2(y_i - mx_i - c) \times -1 = 0$$

$$\Rightarrow \sum (y_i - mx_i - c) = 0$$

$$\Rightarrow \sum y_i - \sum mx_i - \sum c = 0$$

$$\Rightarrow \sum y_i - \sum mx_i = \sum c$$

$$\Rightarrow \frac{\sum y_i}{n} - \frac{\sum mx_i}{n} = \frac{\sum c}{n}$$ (deviding both side)

$\Rightarrow \bar{y} - m\bar{x} = c$, $\bar{x}, \bar{y} \rightarrow$ mean

$\Rightarrow \boxed{c = \bar{y} - m\bar{x}}$ — eq2

2) $\frac{dJ}{dm} = \frac{d}{dm}\left(\sum (y_i - mx_i - c)^2\right) = 0$

$\Rightarrow \frac{d}{dm}\left(\sum (y_i - mx_i - \bar{y} + m\bar{x})^2\right) = 0$

$\Rightarrow \sum \frac{d}{dm} (y_i - mx_i - \bar{y} + m\bar{x})^2 = 0$

$\Rightarrow \sum 2(y_i - mx_i - \bar{y} + m\bar{x}) \times (-x_i + \bar{x}) = 0$

$\Rightarrow \sum (y_i - mx_i - \bar{y} + m\bar{x}) \times (x_i - \bar{x}) = 0$

$\Rightarrow \sum (y_i - \bar{y} - m(x_i - \bar{x})) \times (x_i - \bar{x}) = 0$

$\Rightarrow \sum \left[(y_i - \bar{y})(x_i - \bar{x}) - m(x_i - \bar{x})^2\right] = 0$

$\Rightarrow \sum m(x_i - \bar{x})^2 = \sum (y_i - \bar{y})(x_i - \bar{x})$

$$\Rightarrow \boxed{m = \frac{\sum (y_i - \bar{y})(x_i - \bar{x})}{\sum (x_i - \bar{x})^2}}$$ — eq3

Finally we got our m and c

now we can calculate y-pred by

✓ $\boxed{\hat{y} = mx + c}$ — we got from training data

y-pred   x-test

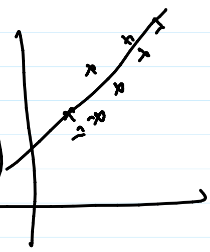
$\boxed{\hat{y} = mx + c}$

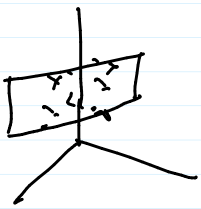
m = weight

c = offset

② Gradient Descent

Data (x-train)		$\hat{y}$
2D	$\longrightarrow$	1D line ($\hat{y} = mx + c$)
3D	$\longrightarrow$	2D plane ($\hat{y} = m_1x_1 + m_2x_2 + c$)
⋮		
(n+1)D	$\longrightarrow$	nD ($\hat{y} = m_1x_1 + m_2x_2 + \cdots m_nx_n + c$)





$(n+1)D \longrightarrow nD$ hyperplan $(\hat{y} = m_1x_1 + m_2x_2 + \dots m_nx_n + c)$

hyperplane

$$\hat{y} = w_1x_1 + w_2x_2 + w_3x_3 + \dots w_nx_n + w_0$$

$$= [w_1\ w_2\ w_3 \dots w_n] \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix} + w_0$$

$$\boxed{\hat{y} = W^T \cdot X + w_0}$$

$$W = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_n \end{bmatrix}$$

$$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

$$[1\ 2\ 3] \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix}$$

$\Rightarrow 1 \times a + 2 \times b + 3 \times c$

by default every vector is a col. vect.

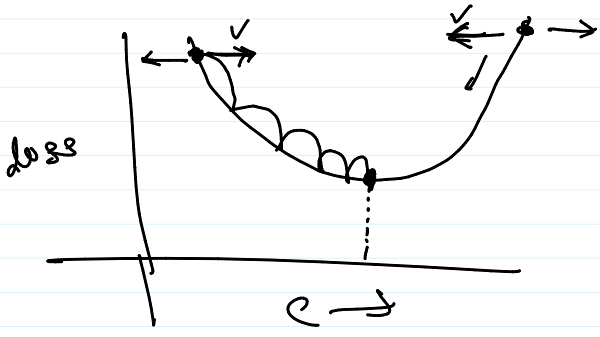
$$A = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix}$$

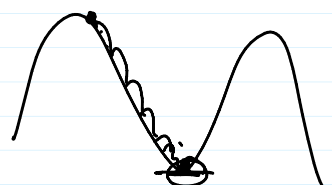
$$A^T = [a_1\ a_2\ a_3 \dots a_n]$$

$$\boxed{\hat{y} = W^T x + c}$$

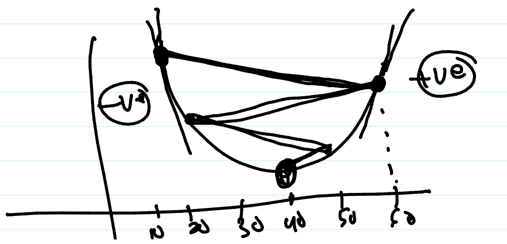
we have to find

loss vs c





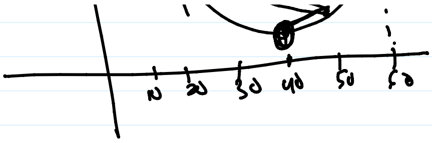
$$\boxed{C_{new} = C_{old} - \text{slope}} \quad \frac{dJ}{dc}$$



$c = 10 \rightarrow$ slope $= -50$

$C_{new} = 10 - (-50)$

$= 60 \rightarrow$ slop $= +40$



$= 60 \rightarrow$ slop $= +40$

$C_{new} = 60 - 40$

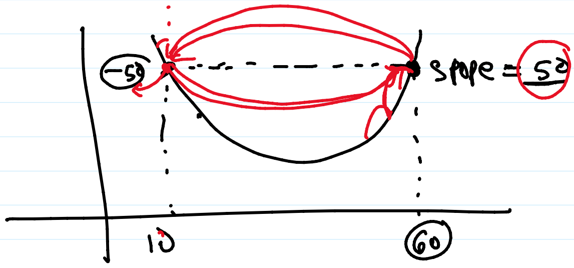
$= 20$

$C_{new} = 20 - (-30)$

$= 50$

→ step

→ epoch



$C_{new} - C_{old} = 0.00001$

$60 - 50 = 10$

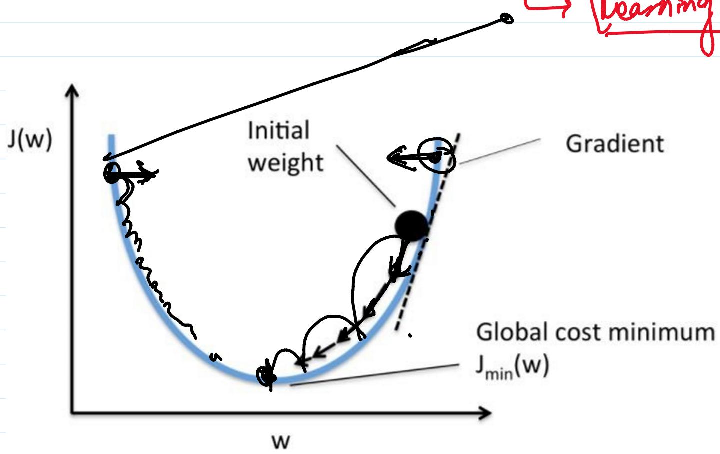
$10 - (-50) = 60 \times 0.01$

0.6

$C_{new} = C_{old} - \lambda \text{ (slope)}$

→ Learning ~~step~~ rate

$\boxed{\lambda = 0.01}$



$w_{new} = w_{old} - \lambda \text{ (slope)}$ ↓

$0.1 \times 100 = \boxed{10}$

$0.01 \times 100 = \boxed{1}$

$0.01 \times 90 = 0.9$

$0.01 \times 60 = 0.6$

$2 \times 100 \rightarrow 200$