2:

Linear Regression with Multiple ~~fe~~ Variables.

(Features and Polynomial Regression)

Eg: House Price Prediction:

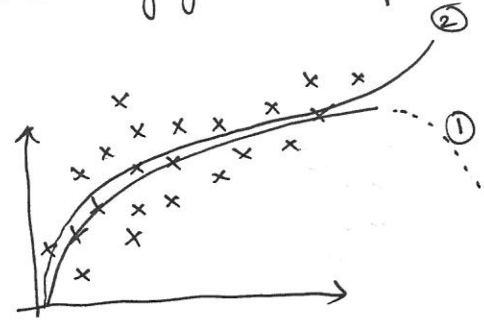
$$y(x) = b_0 + b_1 \underbrace{(\text{frontage})}_{x_1} + b_2 \underbrace{(\text{depth})}_{x_2}$$

Land Area $\Rightarrow$ $x = \text{frontage} \times \text{depth}$

$$y(x) = b_0 + b_1 \underset{(\text{land area})}{x}$$ might be better than

Defining New features may give better prediction result than using given feature

Polynomial Regression:



(quadratic model)

$b_0 + b_1 x + b_2 x^2$ — 1

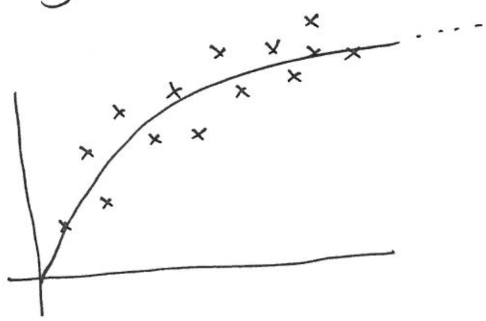
$b_0 + b_1 x + b_2 x^2 + b_3 x^3$ — 2

Example 1: $y(x) = b_0 + b_1 x + b_2 x_2 + b_3 x_3$ (cubic model)

$$= b_0 + b_1 (\text{size}) + b_2 (\text{size})^2 + b_3 (\text{size})^3$$

$$\left.\begin{aligned} x_1 &= \text{size} \\ x_2 &= (\text{size})^2 \\ x_3 &= (\text{size})^3 \end{aligned}\right\}$$ feature scaling (ie, Normalization / Standardization) is important here

$$\left.\begin{aligned} \text{size} &\Rightarrow 1 - 1000 \\ \text{size}^2 &\Rightarrow 1 - 1000{,}000 \\ \text{size}^3 &\Rightarrow 1 - 10^9 \end{aligned}\right\}$$ all make it to 0 – 1 range.

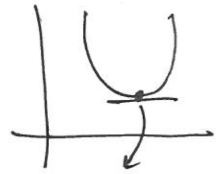


Example 2: $y(x) = b_0 + b_1 (\text{size}) + b_2 (\sqrt{\text{size}})$

Normal Equations:

(Matrix Notations) $\longrightarrow$ alternative to GD.

- solve for 'b_i' mathematically.
- No need for iterative approaches like GD.



$$E(b) = b_0 + b_1 x_1$$

$$\frac{d\,E(b)}{d b_i} \Rightarrow \text{and set } = 0.$$

} same as our SLR. $\frac{\partial E}{\partial b_j} = 0.$

Task: $E(b_0, b_1, \ldots, b_d) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$

$B \in \mathbb{R}^{d+1}$ $\rightarrow$ $(b_0, b_1, \ldots, b_d)$

$$\frac{\partial}{\partial b_j} E(b_0, b_1, \ldots b_d) = \ldots\ldots = 0 \quad [\text{for } j = 1, 2, \ldots d]$$

Solve for $b_0, b_1, \ldots, b_d$ parameters.

Example: N = # of training samples. d = # of features.

	x_0	x_1	x_2	x_3	x_4	y
1.	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	
2.						
⋮						
N	x_{N0}	x_{N1}	x_{N2}	x_{N3}	x_{N4}	

X $(N \times (d+1))$ $\qquad$ y $(N \times 1)$

$$\mathbf{x}_1 = \begin{bmatrix} x_{10} \\ x_{11} \\ x_{12} \\ \vdots \\ x_{1d} \end{bmatrix}$$

$$\boxed{B = (X^T \cdot X)^{-1} X^T \cdot y}$$

$$X = \begin{bmatrix} \text{---} & x_1^T & \text{---} \\ \text{---} & x_2^T & \text{---} \\ & \vdots & \\ \text{---} & x_N^T & \text{---} \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}$$

$N \times (d+1)$

Note: $(X^T \cdot X)^{-1} \Rightarrow$ inverse of $(X^T \cdot X)$ matrix.

set $A = X^T X$

$(X^T X)^{-1} = A^{-1}$

3.

GD	Normal Equations
Dis. Adv. { → Need to choose α	→ No need to choose α } Adv.
→ Needs many iterations	→ No need to iterate
Adv. { → Works well even if the d is large	→ Need to compute $(X^T \cdot X)^{-1}$ } dis adv. $O(d^3)$
	→ slow if 'd' is very large.
if 'd' is large use GD. (10^6's)	if 'd' is smaller, use NE.
	$d = 100$ or 1000