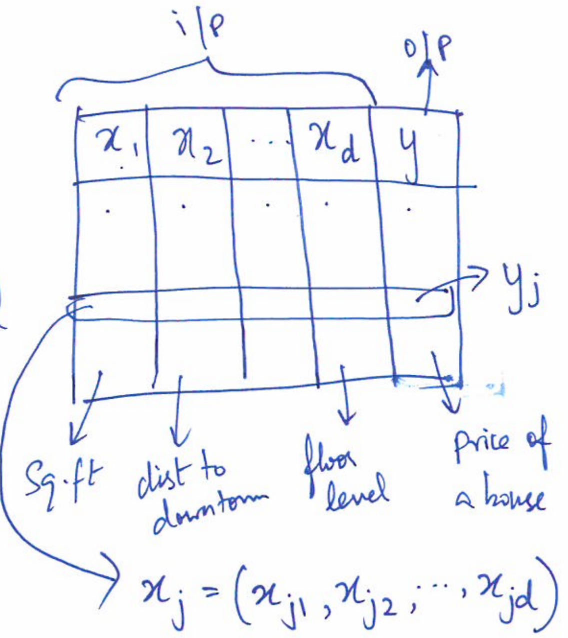
1

Linear Regression

(Multivariable)



Task: Given $(x_1, x_2, \dots x_d)$ we need to predict the value for y.

Training Data
↓
Learning Algo
↓
Model

$(x_1 = 100, x_2 = 5, \dots x_d = 10)$ → Model → price of house $(y) = 21.2$

New data

$(b_0, b_1, b_2, \dots, b_d)$

---

for 1-variable → $\hat{y}_i = b_0 + b_1 x_i$

Regression fn. / Hypothesis:

$$\boxed{\hat{y}_i(x) = b_0 + b_1 x_{i1} + b_2 x_{i2} + \dots + b_d x_{id}}$$

find the parameters $(b_0, b_1, b_2, \dots, b_d)$?
such that Error $(y_i - \hat{y}_i)$ is min.

cost fn. / Error fn.

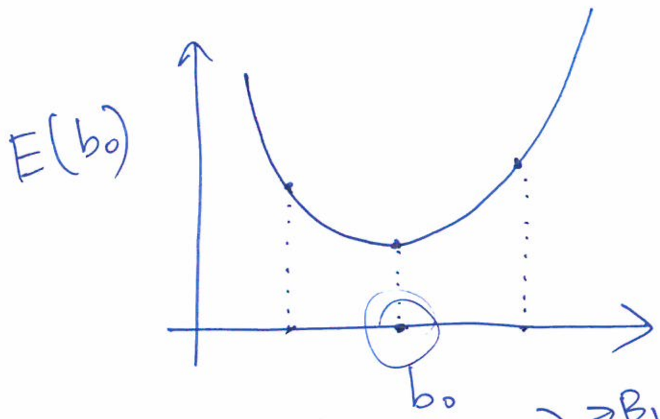
S.S.E

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

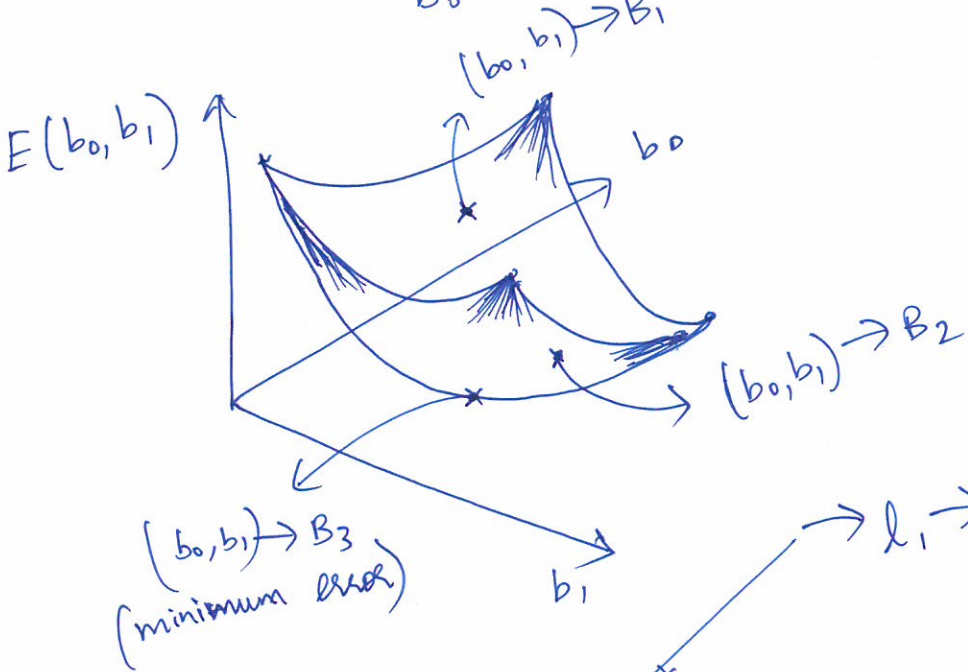
find $(b_0, b_1, \dots, b_d) \Rightarrow \min\limits_{(b_0, b_1, \dots, b_d)} E(b_0, b_1, \dots, b_d)$.

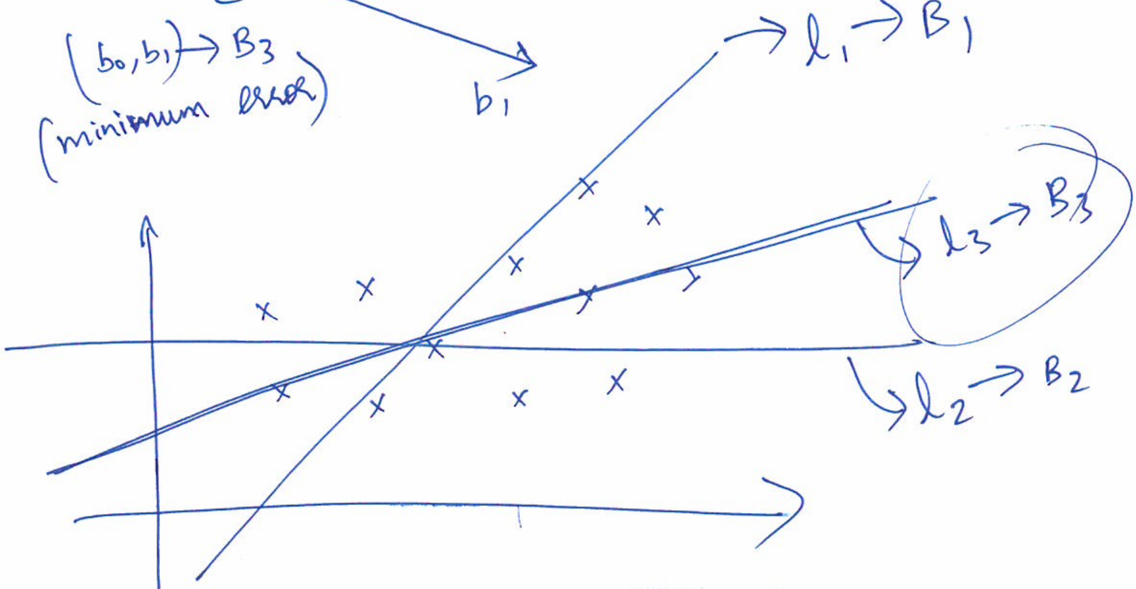
2

Cost fn / Error fn :



$b_0 \rightarrow E(b_0)$ is min.





$E(b_0, b_1, b_2, \ldots, b_d) \rightarrow (d+1)$ params.

Goal $\rightarrow$ $\underset{(b_0, b_1, \ldots, b_d)}{\text{minimize}}\; E(b_0, b_1, \ldots, b_d)$

$\boxed{\text{Learning Algo}}$ $\rightarrow$ Gradient Descent.

3

Gradiant Descent

Outline :

1. Start with some (b_0, b_1) (Random values)
2. Keep changing (b_0, b_1) value to reduce $E(b_0, b_1)$ until we end-up in min-value.

Math:

repeat until convergence {

$$b_j := b_j - \alpha \cdot \frac{\partial}{\partial b_j} E(b_0, b_1) \quad \text{for } j = 0, 1.$$

}

alfa (learning rate)

partial derivative of E w.r.t b_j

simultaneous update

{

$$temp_0 := b_0 - \alpha \frac{\partial}{\partial b_0} E(b_0, b_1)$$

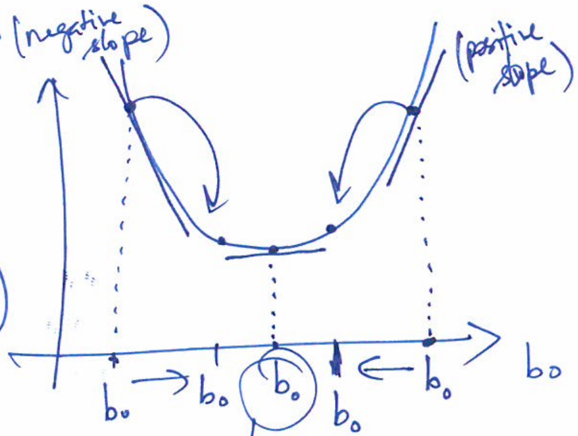
$$temp_1 := b_1 - \alpha \frac{\partial}{\partial b_1} E(b_0, b_1)$$

$$b_0 := temp0$$

$$b_1 := temp1$$

}

④ **Partial Derivative:** $\frac{\partial}{\partial b_j} E(b_0, b_1)$



$E(b_0)$

$b_0 := b_0 - \alpha$ (negative value)

$b_0 := b_0 - \alpha \frac{\partial E(b_0)}{\partial b_0}$

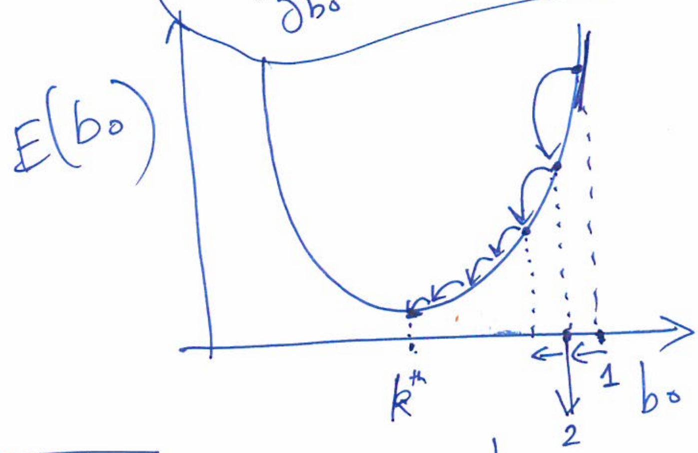
rate of change of E at b_0

$\downarrow$

slope of tangent to the fn at b_0

$\frac{\partial E(b_0)}{\partial b_0} = 0$

$b_0 := b_0 - \underset{0-1}{\alpha}$ (positive value)



$\boxed{GD}$	$\boxed{L.R}$
repeat until convergence $\{$ $b_j := b_j - \alpha \frac{\partial}{\partial b_j} E(b_0, b_1)$ (for $j = 0$ & 1) $\}$	$\hat{y}(x) = b_0 + b_1 x$ $E(b_0, b_1) = \frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2$ $\underset{(b_0, b_1)}{\text{minimize}}\ E(b_0, b_1)$

$$\frac{\partial}{\partial b_j} E(b_0, b_1) = \frac{\partial}{\partial b_j}\left[\frac{1}{2N} \sum_{i=1}^{N} (\hat{y}_i - y_i)^2\right]$$

$$= \frac{\partial}{\partial b_j}\left[\frac{1}{2N} \sum_{i=1}^{N} (b_0 + b_1 x - y_i)^2\right]$$

5

$$j=0 \quad \frac{\partial}{\partial b_0} E(b_0, b_1) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)$$

$$j=1 \quad \frac{\partial}{\partial b_1} E(b_0, b_1) = \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)\, x_i$$

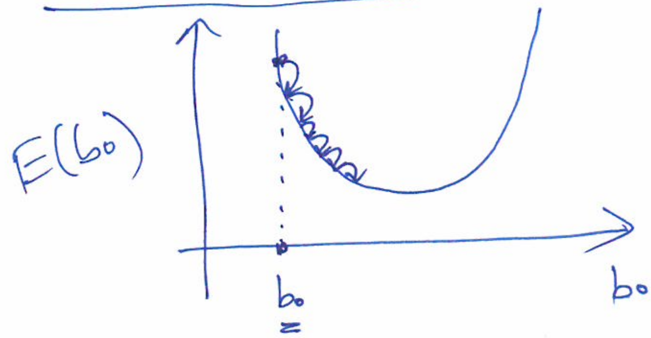
G.D. for LR:

Repeat until Convergence

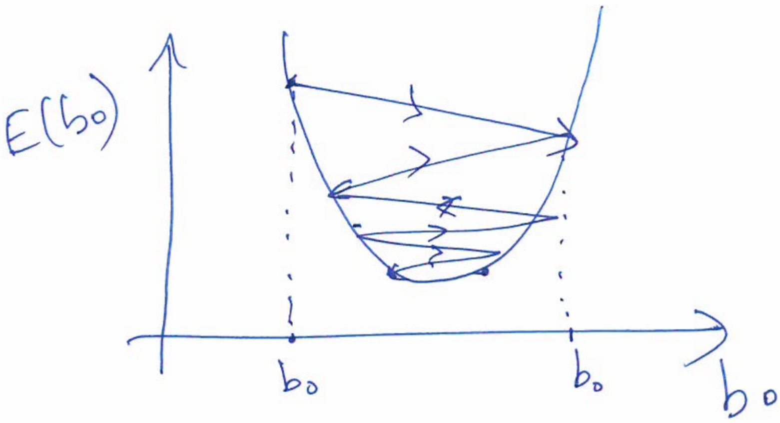
$$\left\{ \begin{aligned} b_0 &:= b_0 - \alpha \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i) \\ b_1 &:= b_1 - \underbrace{\alpha}_{\text{learning rate}} \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)\, x_1 \end{aligned} \right\} \text{ Simultaneous update.}$$

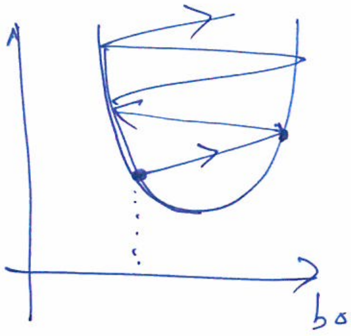
α-value: (0 to 1) Eg: 0.001 (trial-and-error)

if α is too small, G.D will converge very slowly.

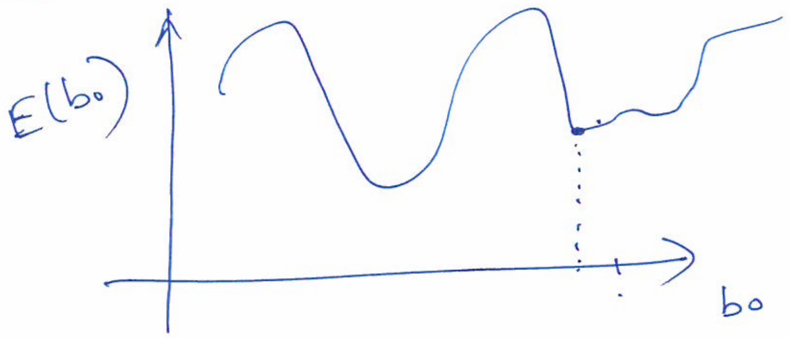


- if α is too large, GD can overshoot minimum.



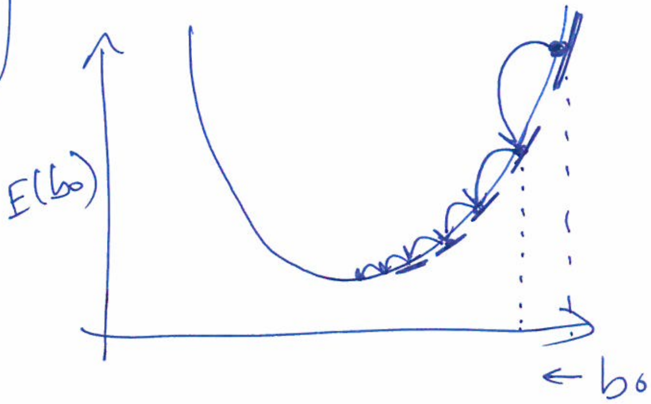


Local Minimum:

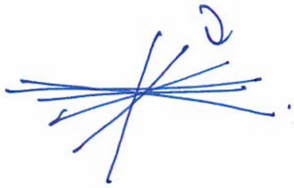


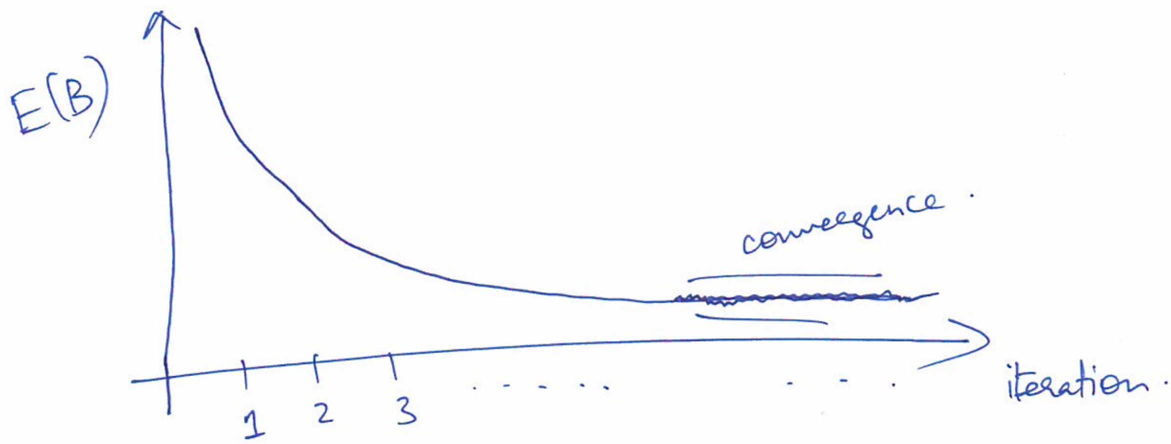
[if your error fn. is not convex]

Fix α-value



$\alpha = 0.001$



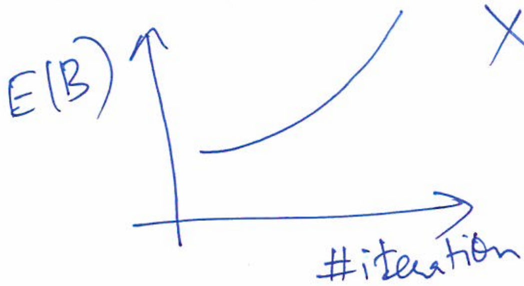


Notes on GD:

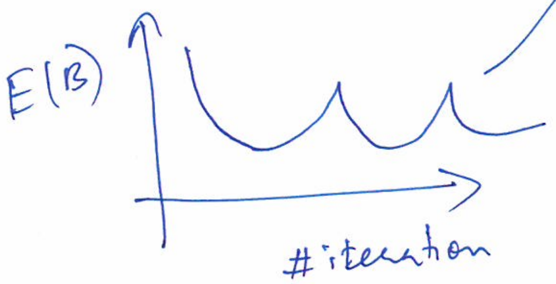
1. Feature scaling $\rightarrow$ Normalization / Standardization

2. $E(B)$ vs. Iteration

3. Not working?



X
- check your code
- choose smaller α-value



try $\alpha = 0.001, \underline{0.01}, \ldots$

0.003, 0.002

(8)

G.D for Multivariate Variable (L.R.)

for 1-var,

$y = b_0 + b_1 x$

$= b_0 x_0 + b_1 x_1$

$= b_0 (1) + b_1 (x_1)$

d-variables

	x_0	x_1	x_2	....	x_d	y
1	1	x_{11}	x_{12}	....	x_{1d}	y_1
2	⋮	⋮	⋮		⋮	⋮
⋮						
N	1	x_{N1}	x_{N2}	....	x_{Nd}	y_N

$x_1 \ldots x_d$: i/p y: o/p

N-samples.

$$\boxed{\hat{y} = b_0 x_0 + b_1 x_1 + \ldots + b_d x_d}$$

$$\hat{y}_i = b_0 x_{i0} + b_1 x_{i1} + \ldots + b_d x_{id}$$

$$X_i = \begin{bmatrix} x_{i0} \\ x_{i1} \\ \vdots \\ x_{id} \end{bmatrix} \rightarrow X_i \in R^{d+1}$$

$$B = \begin{bmatrix} b_0 \\ b_1 \\ \vdots \\ b_d \end{bmatrix} \rightarrow B \in R^{d+1}$$

$$\boxed{y = B^T \cdot X}$$

$$= \underbrace{[b_0\, b_1 \ldots b_d]}_{(d+1)\times 1} \cdot \underbrace{\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_d \end{bmatrix}}_{1 \times (d+1)}$$

Reg. Eq. for Multiple Variables.

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

G.D ⇒

Repeat until convergence {

$$b_j := b_j - \alpha \cdot \frac{1}{N} \sum_{i=1}^{N} (\hat{y}_i - y_i)\, x_{ij}$$

} (simultaneously update b_j for $j = 0, 1, \ldots, d$)