Andrew Ng's

Supervised Learning - "Given right answer"

Regression problem

Training Set
↓
Learning Algo
↓
Size of house → h ⇒ Estimated price

hypothesis

↳ h maps from x's to y's

Notation

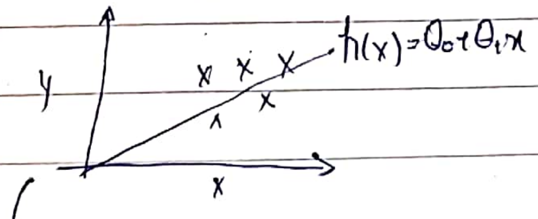
m - no. of examples (train)

x, y

$(x^i, y^i) \rightarrow i^{th}$ example

How to represent h?

$h_\theta(x) = \theta_0 + \theta_1 x$ — parameters

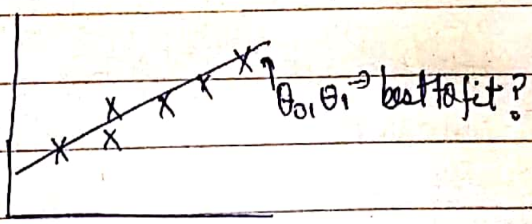


↳ Linear Regression with one variable
or
Univariate regression

fitting line to data

↳ Cost function:-

How to choose θ's?



Idea: Choose $\theta_0\ \theta_1$ so that $h_\theta(x)$ is close to y for our train. data (x, y)

$$\boxed{\underset{\theta_0\,\theta_1}{\text{minimize}}\ \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta(x^i) - y^i\right)^2}$$

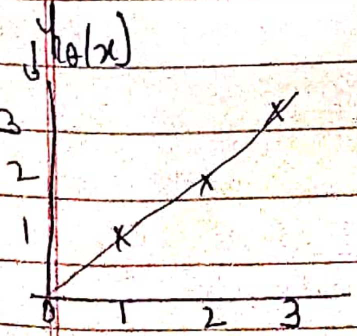
↳

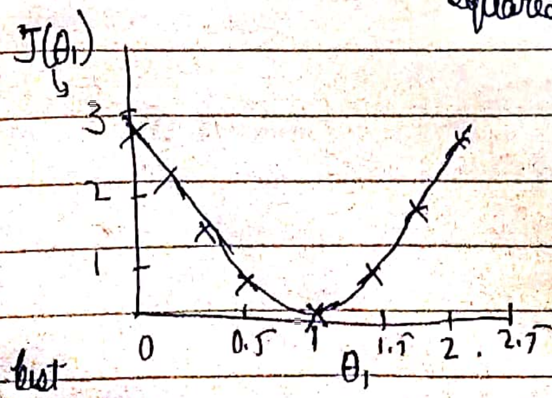
$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum^{m}\left(h_\theta(x^i) - y^i\right)^2$$

min. $\underbrace{J(\theta_0, \theta_1)}_{\text{Cost fn.}}$

or

Squared error function

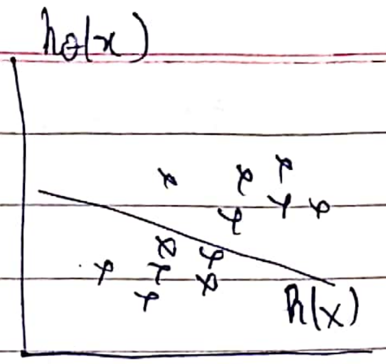




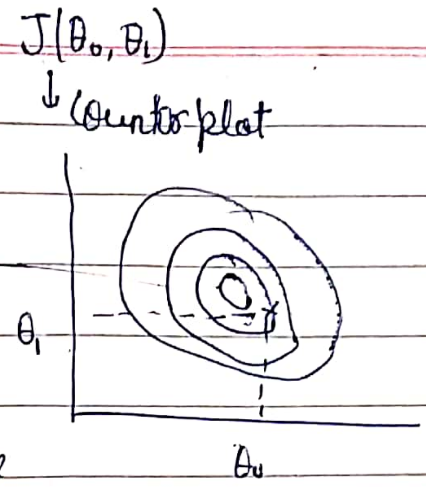
for $\theta_1 = 1$, $J(\theta) = 0$ but

$\theta_1 = 0.5$ $J(\theta_1) = 0.58$

$\boxed{\theta_1 = 1} \rightarrow$ min $J(\theta_1)$



not great



for close to centre close to min

Algorithm to automatically find values of θ_0, θ_1 to min $J(\theta_0, \theta_1)$

Gradient Descent (to minimize cost fn)

~~Have some function $J(\theta_0, \theta_1)$~~
~~What min $J(\theta_0, \theta_1)$~~

Outline:-

- ~~Start with some (θ_0, θ_1)~~
- Keep changing (θ_0, θ_1) to reduce $J(\theta_0, \theta_1)$ until end up at minimum.

algo:-

repeat until converge {

$$\theta_j := \theta_j - \alpha \boxed{\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)} \quad (\text{for } j=0 \text{ and } j=1)$$

}

learning rate (steps) → derivate term (later)

Correct: Simultaneous update

$temp0 := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1)$

$temp1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$

$\theta_0 := temp0$

$\theta_1 := temp1$

Incorrect

$temp0 :=$ ______

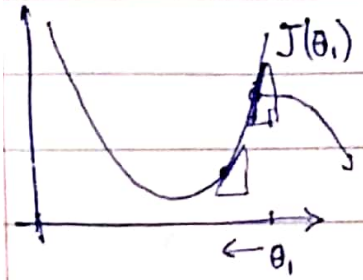
$\theta_0 := temp0$

$temp1 :=$ ______ $J(\theta_0, \theta_1)$

$\theta_1 := temp1$

New value

$$\theta_i = \theta_i - \alpha \frac{\partial}{\partial \theta_i} J(\theta_0, \theta_1) \qquad i = 0 \text{ and } 1$$



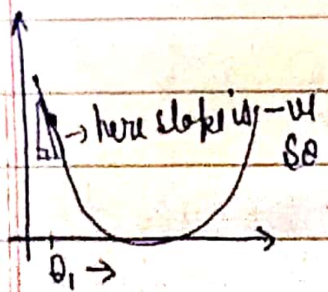
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta} J(\theta_1)$$

what is slope of line at this value of θ_1

In this case slope is +ve so

$$\theta_1 = \theta_1 - \alpha(\text{positive value})$$

so θ_1 decreases, close to min



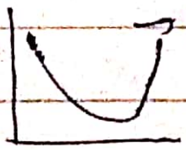
so

$$\theta_1 := \theta_1 - \alpha(\text{negative})$$

θ_1 increases

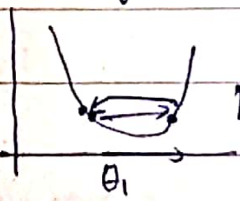
← close to min.

If α is too slow (small)

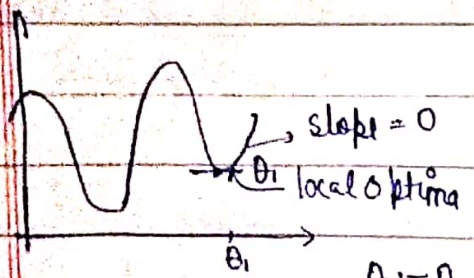


→ to many steps take time

If α is too large, overshoot

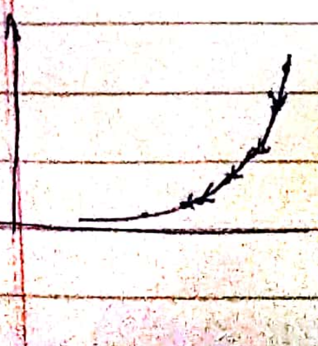


never converge



$$\theta_1 := \theta_1 - \alpha(0)$$

$\theta_1 := \theta_1$ unchanged.



As we approaches min slope ↓es so change in θ_1 ↓es
bcz smaller steps

Gradient descent with cost function:-

G.d.

$\theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1)$

$h_\theta(x) = \theta_0 + \theta_1 x$

$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta x^i - y^i)^2$

$\frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1) = \frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum (h_\theta x^i - y^i)^2$

or

$\frac{\partial}{\partial \theta_j} \frac{1}{2m} \sum^{m} (\theta_0 + \theta_1 x^i - y^i)^2$

for θ_0

$j=0: \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) = \frac{1}{m} \sum (h_\theta(x)^i - y^i)$

for θ_1

$j=1: \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) = \frac{1}{m} \sum (h_\theta x^i - y^i) . x^i$

Now,

$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)$

$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^i) - y^i) . x^i$

] update simultaneously

Cost function for linear regression is always bow shaped means no local optima, converges at global optima.

"Batch Gradient Descent", the one we just done.

↳ "Batch" → Each step of G.d. uses all training examples.

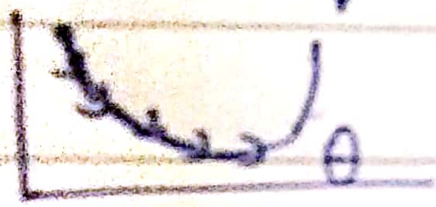
Scanned with CamScanner

$h_\theta(x) = \theta_0 + \theta_1 (size) + \theta_2 \sqrt{(size)} + \ldots$

Normal ~~Distribution~~ Eq^n :-

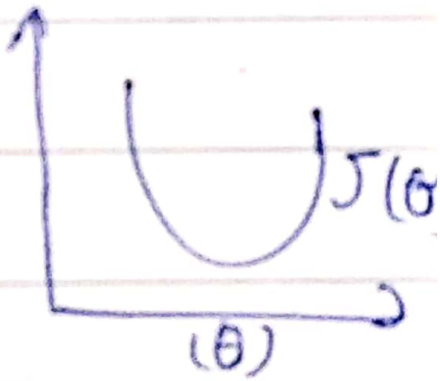
→ Method to solve for θ analytically

In 4D



Intuition ($\theta \in \mathbb{R}$)

$$J(\theta) = a\theta^2 + b\theta + c$$



To min. quad. eq^n

$$\frac{\partial}{\partial \theta} J(\theta) = 0$$

solve for (θ)

$$J(\theta_0 \ldots \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta x^i - y^i)^2$$

$$\frac{\partial}{\partial \theta_j} J(\theta) = 0 \text{ for every } j$$

Solve for $\theta_0\ \theta_1 \ldots \theta_n$

m-example n-features

G.d	Normal eq^n
• Need to choose α	• No need
• Need many iteration	• No need to iterate
• Works well even when n is large	• Need to compute $(X^TX)^{-1} \to n \times n$
	• Slow if n is very large $\underline{O(n^3)}$
	if $n > 1000$

Example

$m = 4$

x_0	Size	No. of bed	floor	Age	Price
1	2104	5	1	45	460
1	1416	3	2	40	232
1	1534	3	2	30	315
1	852	2	1	36	178

$$X = \begin{bmatrix} 1 & 2104 & 5 & 1 \\ 1 & 1416 & 3 & 2 \\ 1 & 1534 & 3 & 2 \\ 1 & 852 & 2 & 1 \end{bmatrix} \quad y = \begin{bmatrix} 460 \\ 232 \\ 315 \\ 178 \end{bmatrix}$$

$m \times (n+1)$ — rows, features

m-d vector

$$\boxed{\theta = (X^TX)^{-1} \cdot X^Ty}$$ → Optimal value of θ that min. $J(\theta)$

$$x^i = \begin{bmatrix} x_0^i \\ x_1^i \\ x_2^i \\ \vdots \\ x_n^i \end{bmatrix} \quad X = \begin{bmatrix} \text{---} (x^1)^T \text{---} \\ \text{---} (x^2)^T \text{---} \\ \vdots \\ \text{---} (x^n)^T \text{---} \end{bmatrix}$$

X = design matrix

$m \times n+1$

m examples
n features

Eg:- $x^{(i)} = \begin{bmatrix} 1 \\ x_1^{(i)} \end{bmatrix}$

$$X = \begin{bmatrix} 1 & x_1^{(1)} \\ 1 & x_2^{(1)} \\ \text{--} & \text{--} \end{bmatrix} \quad y = \begin{bmatrix} y^1 \\ y^2 \\ \vdots \\ y^n \end{bmatrix}$$

$(X^TX)^{-1}$ is inverse of X^TX

In normal equation method, no need to do feature scaling.

Normal Eq^n Noninvertibility :-

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

what if $X^T X$ is non invertible?
(Singular / degenerate)

In Octave

pinv(X' * X) * X' * y

- pinv
- inv

Cause of invertibility?

- Redundant features

Eg:- x_1 = size in feet^2

x_2 = size in m^2

$x_1 = (3.28)^2 \times x_2$

- Too many features (eg: $m \le n$)
  - Delete some features or use regularization.