11/06/2020

Simple Regression with mathematics [University of Washington]

* Dataset which is going to use in this is on "Predicting house prices"

- So starting with the Question:
  → How much is my house worth?
  ↳ Look at recent sales in my neighborhood. How much did they sell for?

- Regression fundamentals
  1) Data
  2) Model
  3) Task

1) Data

Input → x_i = sq. ft. , Output → y_i = $

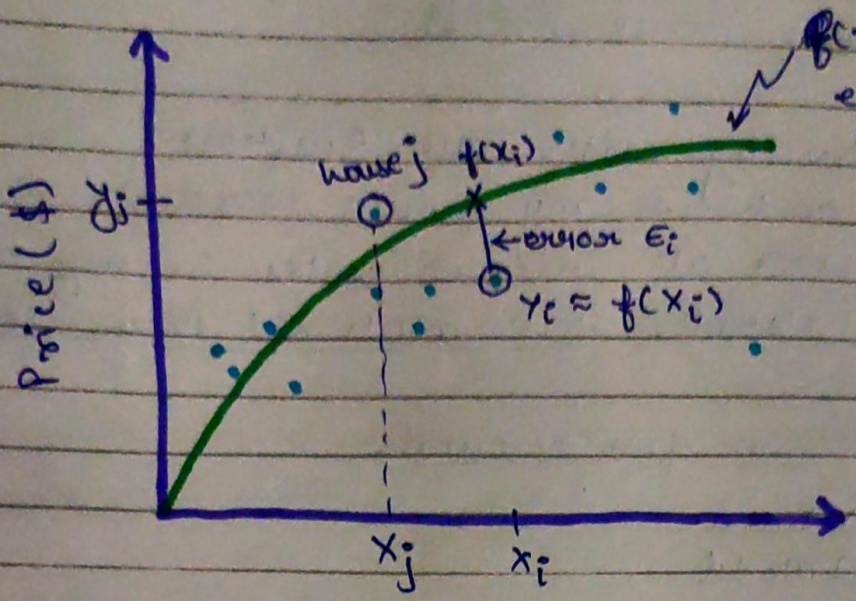
🏠 (x_1 = sq. ft. , y_1 = $)
🏠 (x_2 = sq. ft. , y_2 = $)
🏠 (x_3 = sq. ft. , y_3 = $)
🏠 (x_4 = sq. ft. , y_4 = $)
⋮

Input vs Output:

- y is the quantity of interest
- assume y can be predicted from x.

2) Model:

How we assume the model works?



Regression Model:

$$Y_i = f(X_i) + \epsilon_i$$

$E[\epsilon_i] = 0$ ← equally likely that error is +ve or -ve.

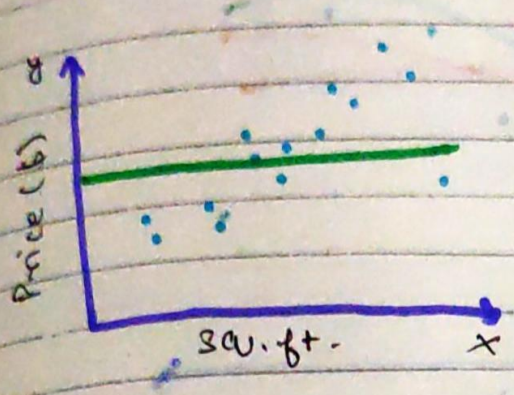
↑ expected value

⇓

Y_i is equally likely to be above or below $f(x_i)$

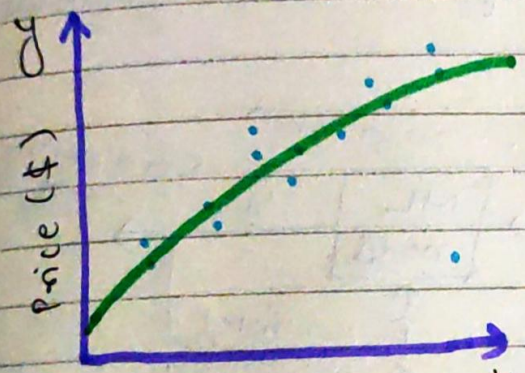
• "Essentially, all models are wrong, but some are useful."
— George Box, 1987.

TASK 1:
We have to choose which model $f(x)$?
Suppose here are 4 examples:

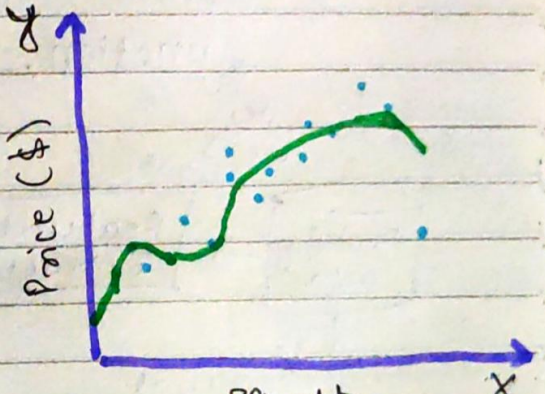


(a)

(b)



(c)

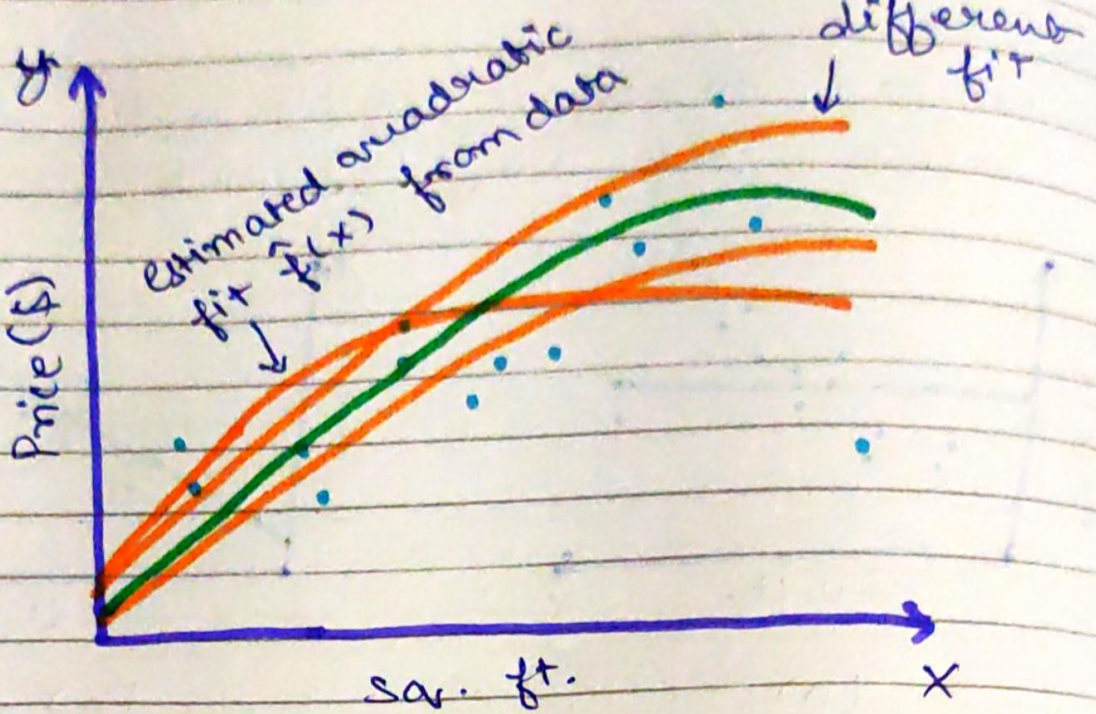


(d)

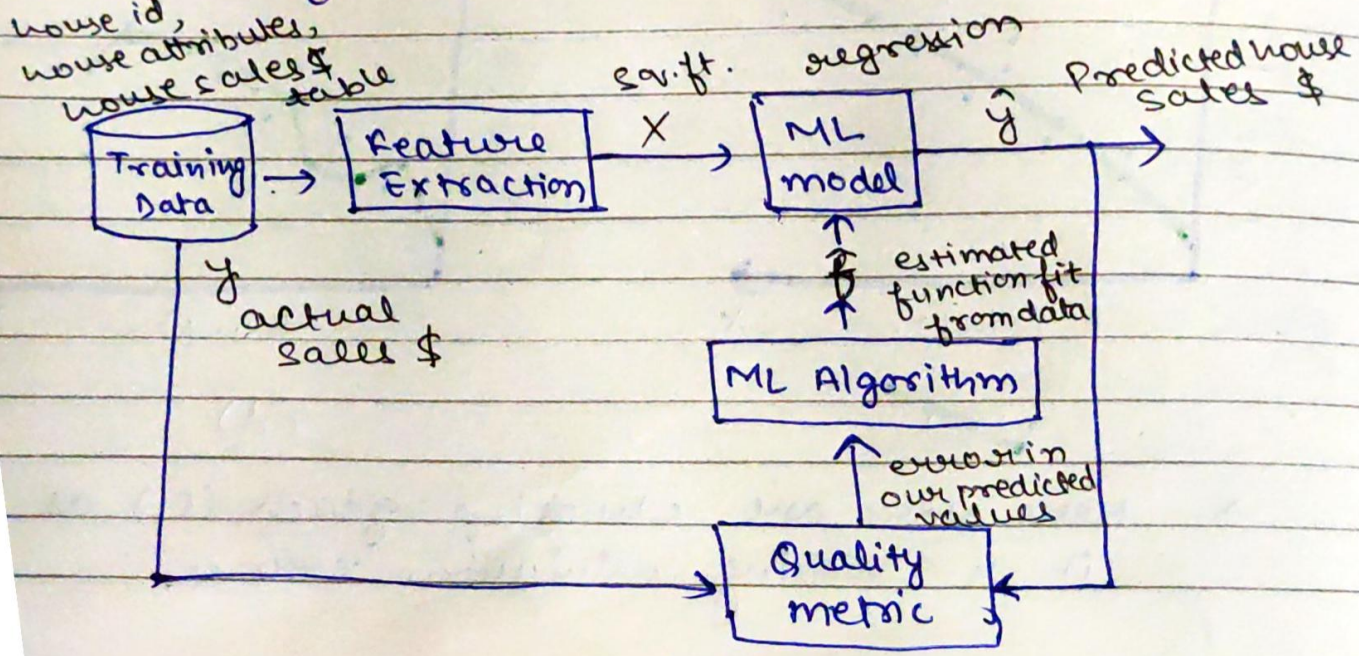
→ Here we are choosing graph (c) as it is having minimum error.

Task 2 :

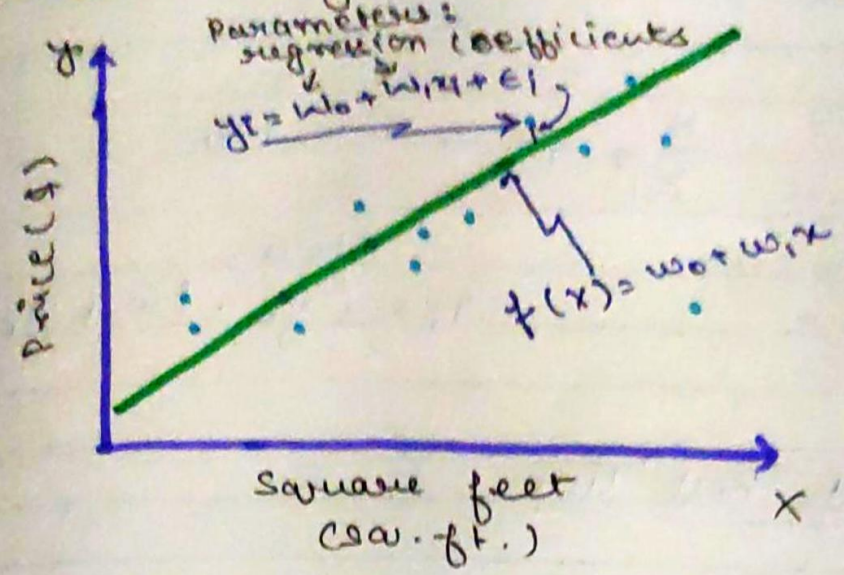
for a given model $f(x)$, estimate function $\hat{f}(x)$ from data



→ Assume mode $f(x)$ is a quadratic function.



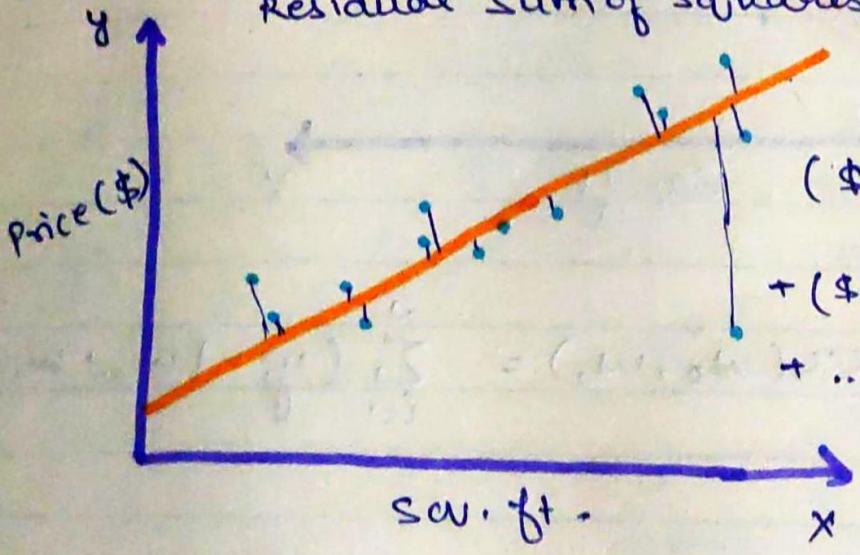
Simple Linear Regression Model



Fitting a line to data

"Cost" of using a given line

Residual Sum of Squares (RSS)



$RSS(w_0, w_1) =$
$(\$_{house\,1} - [w_0 + w_1 \text{sq.ft.}_{house\,1}])^2$
$+ (\$_{house\,2} - [w_0 + w_1 \text{sq.ft.}_{house\,2}])^2$
$+ \ldots$ [including all training houses]

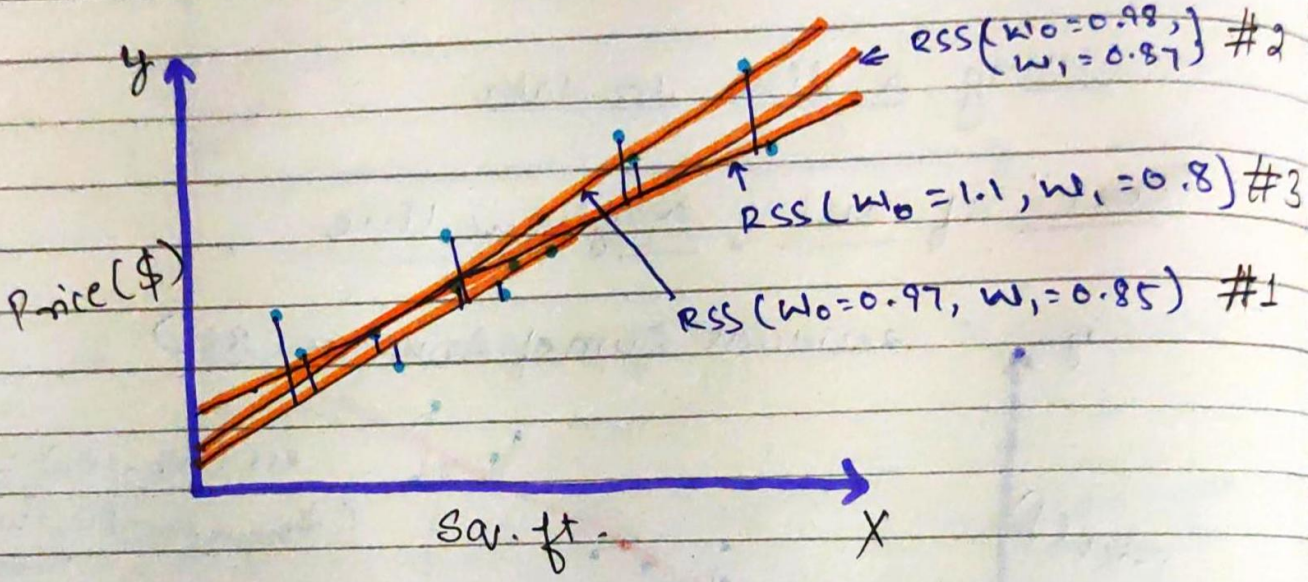
→ $RSS(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$

$\sum_{i=1}^{N} a_i = a_1 + a_2 + \cdots + a_N$

Here,

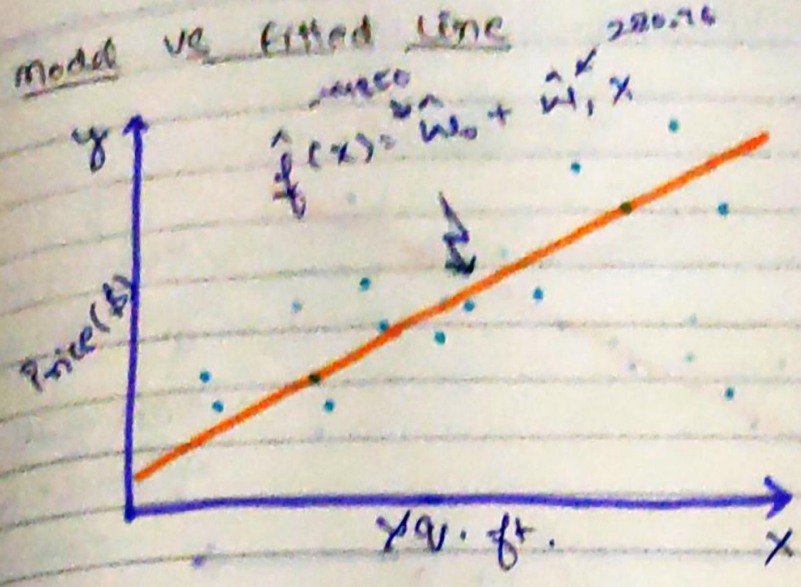
$a_i = (y_i - [w_0 + w_1 x_i])^2$

Find "Best" line



$RSS(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$

* Model vs fitted line



Regression model:

$y_i = w_0 + w_1 x_i + \epsilon_i$

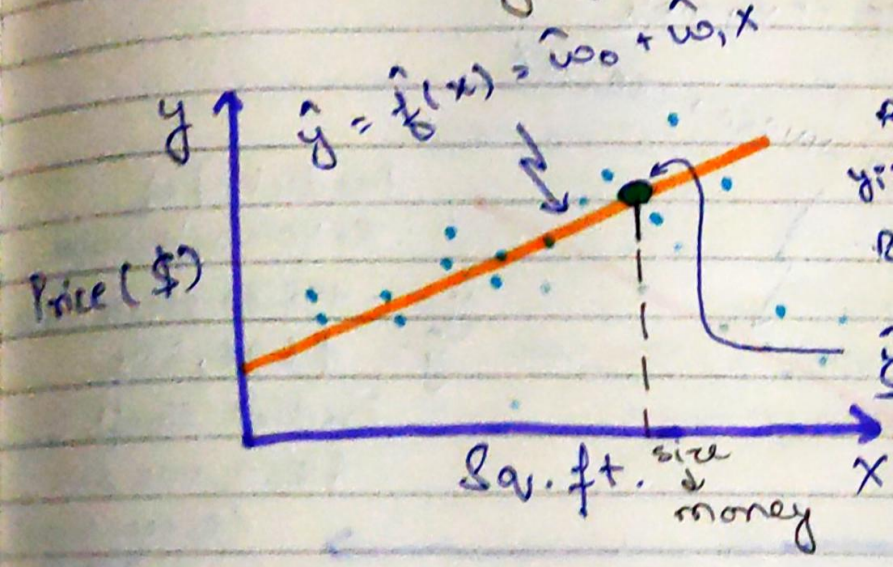
Parameters (unknown variable)

Estimated Parameters:

$-44850 = \hat{w}_0$, $\hat{w}_1 = 280.76$

Take actual values

* Seller: Predicting your house Price



Regression model:

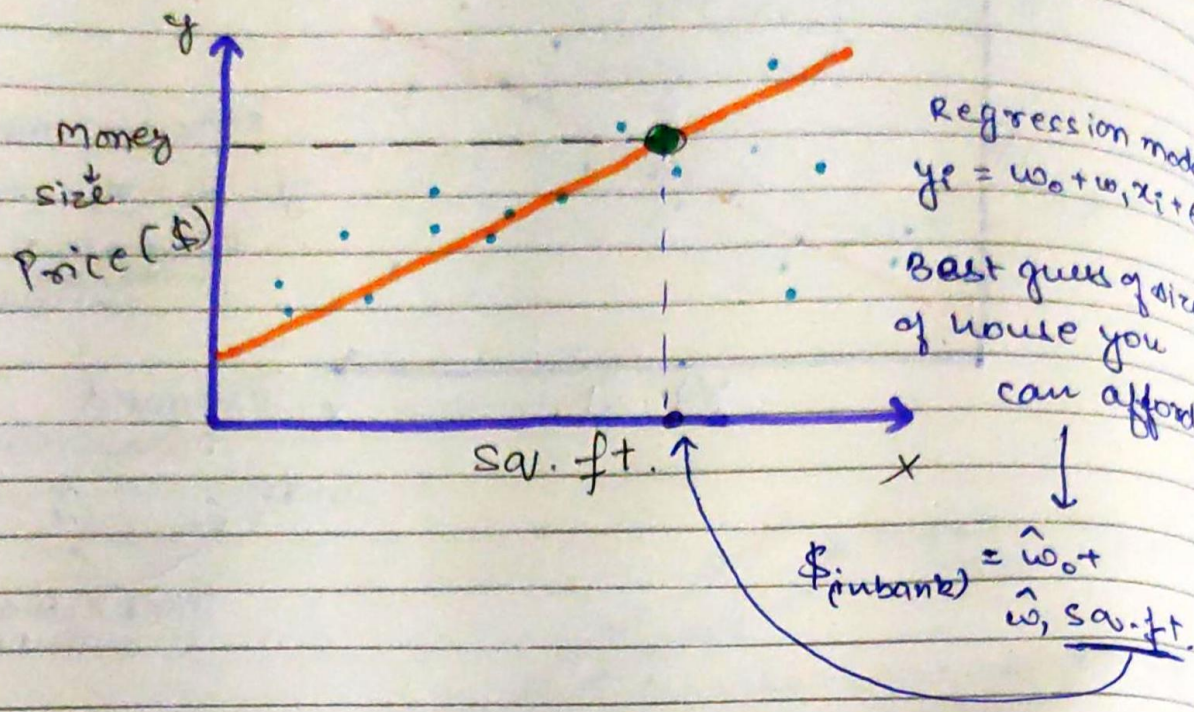
$y_i = f(x_i)$

$y_i = w_0 + w_1 x_i + \epsilon_i$

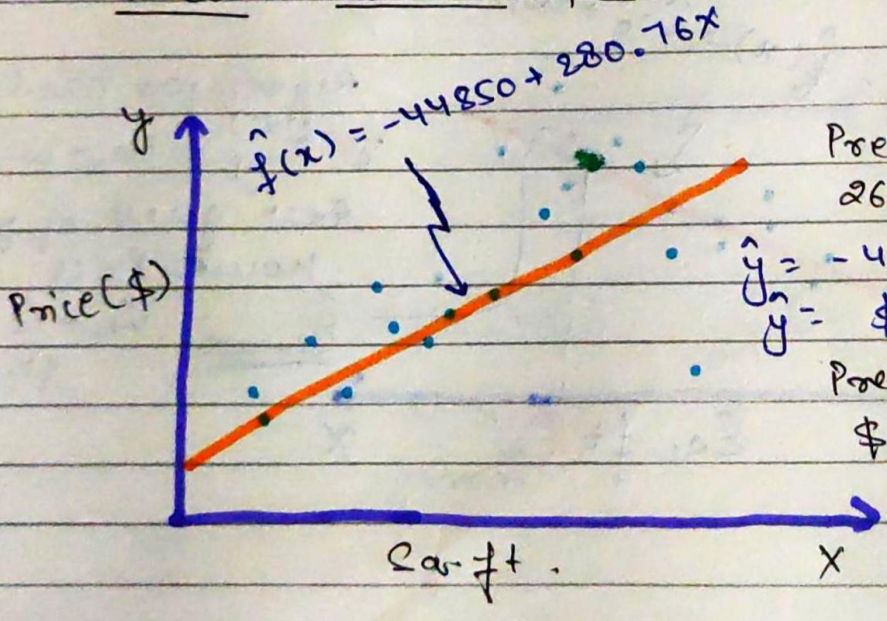
Best guess of your house Price

$\hat{y}_{house} = \hat{w}_0 + \hat{w}_1$ sq. ft. house

* Buyer: Predicting size of house



A concrete example



Predict $ of 2640 sq. ft. house

$\hat{y} = -44850 + 280.76 \times 2640$

$\hat{y} = \$696,356$

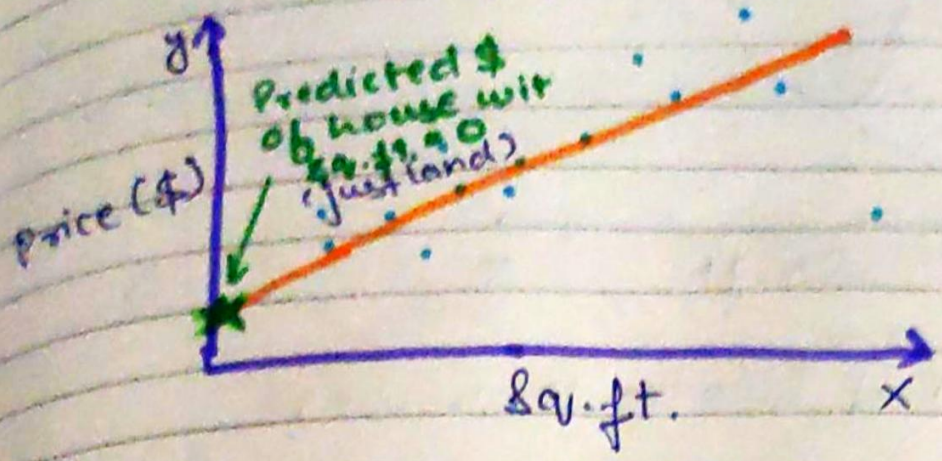
Predict sq. ft. of $859,000 sale:

$\frac{(859000 + 44850)}{280.76}$

$= 3219$ sq. ft.

Interpreting the coefficients

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x$$

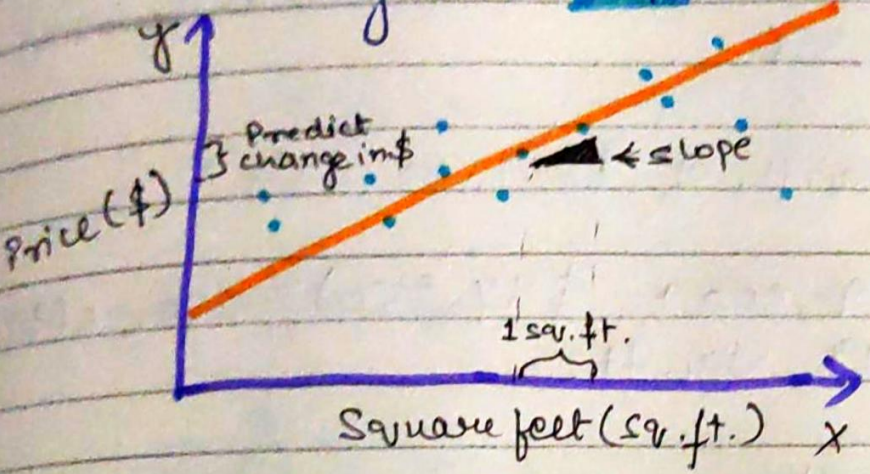


$\hat{y} = \hat{w}_0$ when $x = 0$

↓

This is not very meaningful.

$$\hat{y} = \hat{w}_0 + \hat{w}_1 x$$



$\$_{1001 sq.ft.} - \$_{1000 sq.ft.}$

$= \hat{w}_0 + \hat{w}_1 \cdot 1001_{sq.ft.} - (\hat{w}_0 + \hat{w}_1 \cdot 1000 sq.ft)$

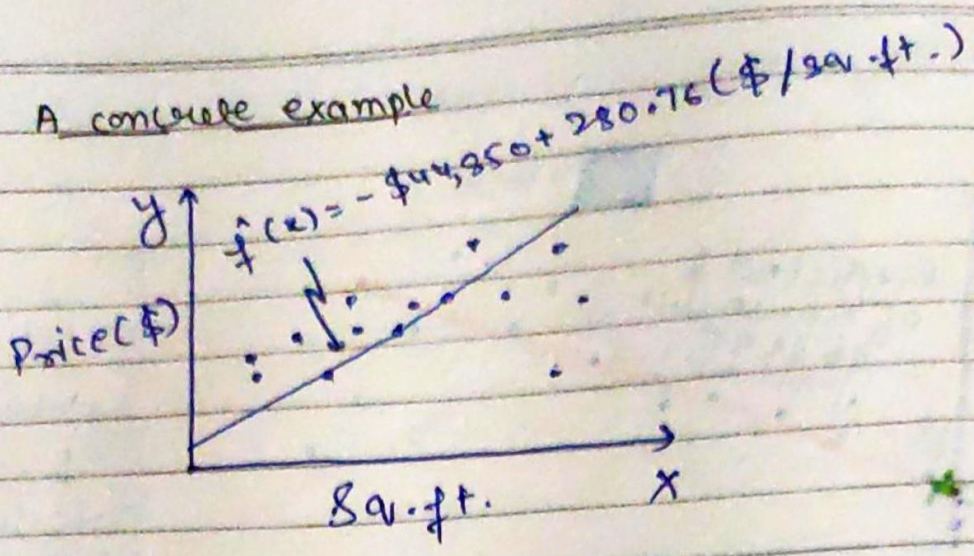
$= \hat{w}_1$

↓

Predict change in the O/P per unit change in I/P.

warning: magnitude depends on unit of both features & observations

A concrete example



Predict $ of 2,640 sq.ft. house:

$-\$44,850 + 280.76(\$/\text{sq.ft.}) * 2640 \text{ sq.ft.}$
$= \$696,356$

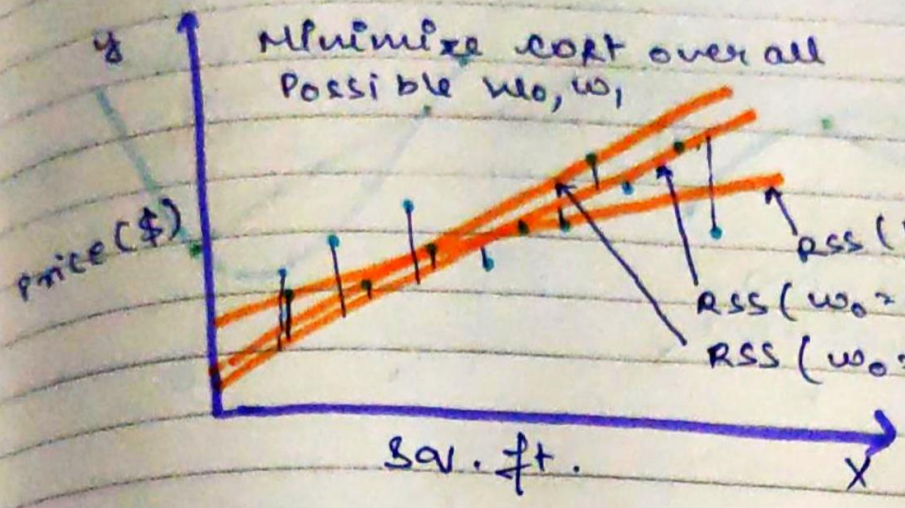
Predict sq.ft. of $859,000 sale:

$= (\$859,000 + \$44,850) / 280.76(\$/\text{sq.ft.})$
$= 3,219 \text{ sq.ft.}$

- But what if:
  - House was measured in sq. meters?
  - Price was measured in RMB?

Algorithm for fitting the model

- find the "best" line



$$RSS(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

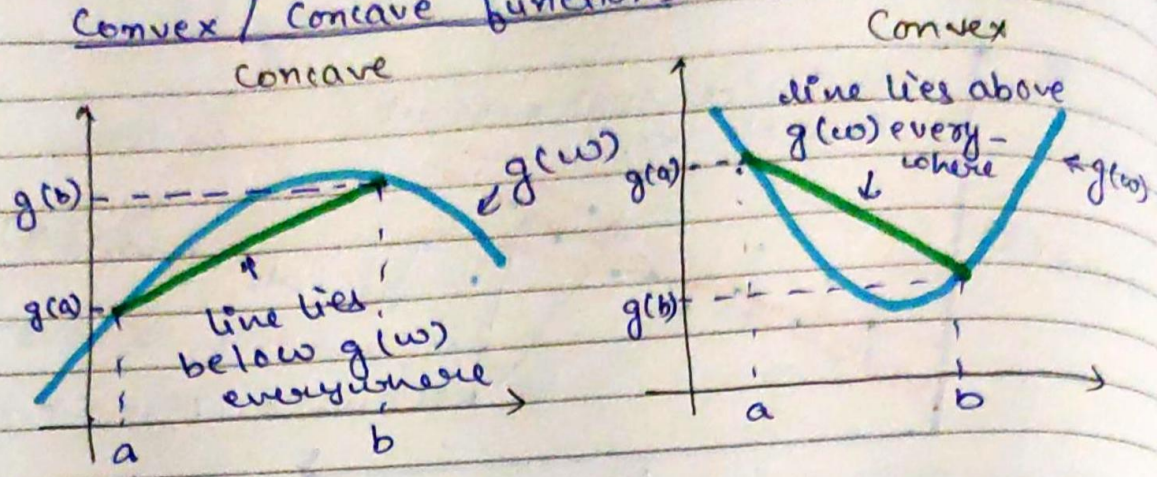
→ minimize function overall possible w_0, w_1

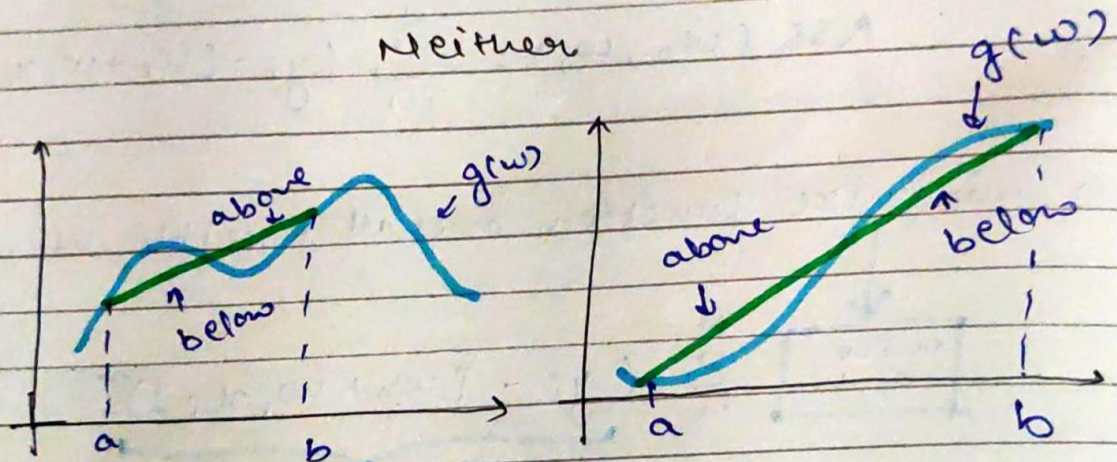
$$\min_{w_0, w_1} \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

RSS (w_0, w_1) is a function of 2 variables = $g(w_0, w_1)$

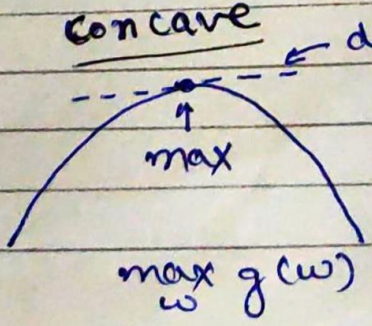
An aside on optimization :

Convex / Concave functions

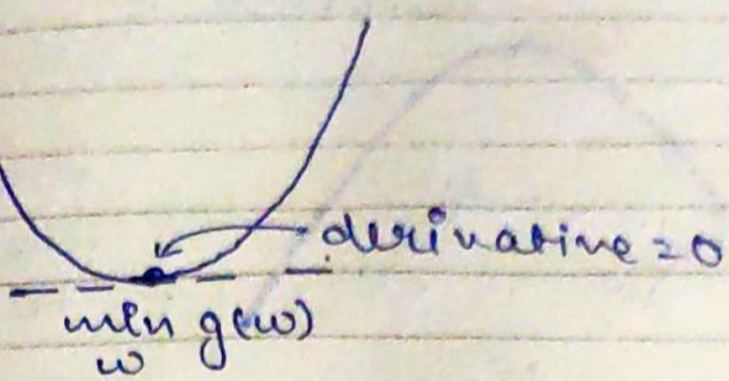




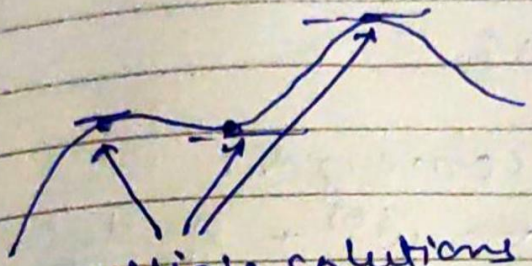
Finding the max or min analytically

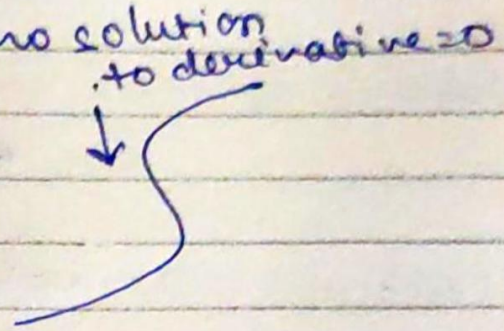


convex



Neither





Example

$$g(w) = 5 - (w-10)^2$$

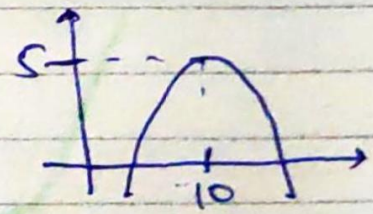
$$\frac{d(g(w))}{dw} = 0 - 2\times(w-10)^1 \times 1$$

$$= -2w + 20$$

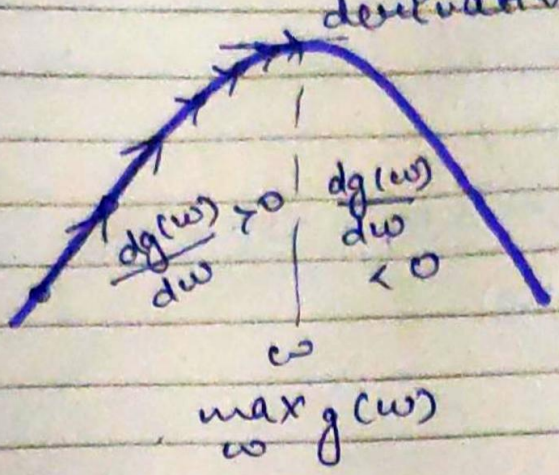
Set derivative = 0

$$-2w + 20 = 0$$

$$\boxed{w = 10}$$



Finding the max via Hill climbing



How does knows whether to move w to right or left?

↓

increase or decrease the value of w?

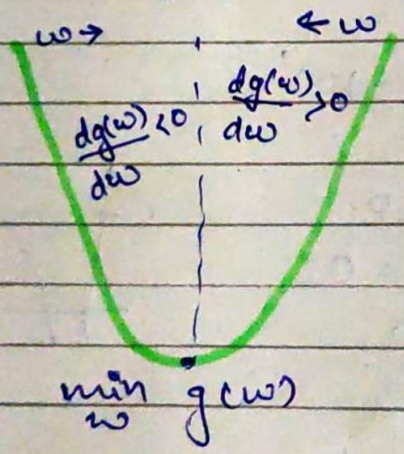
Algorithm:

while not converged:

$$w^{(t+1)} \leftarrow w^{(t)} + \eta \frac{dg(w)}{dw}$$

$w^{(t)}$: iteration; η: step size

Finding the min via Hill descent



when derivative is positive, w want to decrease w.
and when derivative is negative we want to increase w.

Algorithm:

while not converged

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \left.\frac{dg}{dw}\right|_{w^{(t)}}$$

Choosing the stepsize -

1) Fixed stepsize

η

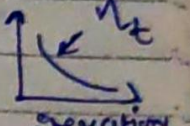
$\eta = 0.1$

2) Decreasing stepsize or stepsize schedule

common choices:

$\eta_t = \frac{\alpha}{t}$

$\eta_t = \frac{\alpha}{\sqrt{t}}$



Convergence criteria :

- For convex functions, optimum occurs when $\frac{dg(w)}{dw} = 0$

- In practice stop when

$$\left| \frac{dg(w)}{dw} \right| < \epsilon$$

ϵ ↑ threshold to be set

Algorithm:

while not converged

$$w^{(t+1)} \leftarrow w^{(t)} - \eta \left. \frac{dg}{dw} \right|_{w^{(t)}}$$

3. Moving to multiple dimensions: Gradients

$$\nabla g(w) = \begin{bmatrix} \frac{\partial g}{\partial w_0} \\ \frac{\partial g}{\partial w_1} \\ \vdots \\ \frac{\partial g}{\partial w_p} \end{bmatrix}$$

gradient of $[w_0, w_1, \ldots, w_p]$

$(p+1)$ - dimensional vector

partial derivative is like a derivative w.r.t. w_1, treating all other variables as constant.

Gradient example:

$$g(w) = 5w_0 + 10w_0w_1 + 2w_1^2$$

$$\frac{\partial g}{\partial w_0} = 5 + 10w_1$$

$$\frac{\partial g}{\partial w_1} = 10w_0 + 4w_1$$

$$\nabla g(w) = \begin{bmatrix} 5 + 10w_1 \\ 10w_0 + 4w_1 \end{bmatrix}$$

NOTE: Contour Plots is a bird's eye view of 3D plot of RSS with tangent plane at minimum.

- Gradient Descent Algorithm:

while not converged

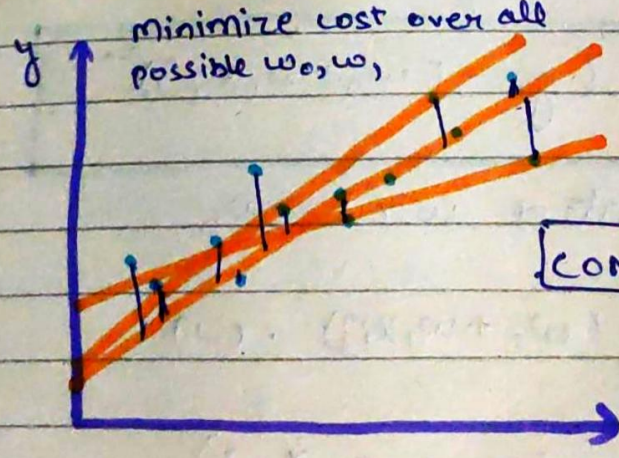
$$w^{(t+1)} \leftarrow w^{(t)} - \eta \nabla g(w^{(t)})$$

$$\begin{bmatrix} \vdots \end{bmatrix} \leftarrow \begin{bmatrix} \vdots \end{bmatrix} - \eta \begin{bmatrix} \vdots \end{bmatrix}$$

convergence: $\| \nabla g(w) \| < \epsilon$

2. Finding the least square lines:

Find "best" line



$$\min_{w_0, w_1} \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

CONVEX $\Rightarrow$ solution is unique + gradient descent algorithm will converge to minimum

Compute the Gradient

→ $RSS(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$

→ $\frac{d}{dw} \sum_{i=1}^{N} g_i(w) = \frac{d}{dw} (g_1(w) + g_2(w) + \dots + g_N(w))$

$= \frac{d}{dw} g_1(w) + \frac{d}{dw} g_2(w) + \dots + \frac{d}{dw} g_N(w)$

$= \sum_{i=1}^{N} \frac{d}{dw} g_i(w)$

→ In our case

$g_i(w) = (y_i - [w_0 + w_1 x_i])^2$

$\frac{\partial RSS(w)}{\partial w_0} = \sum_{i=1}^{N} \frac{\partial}{\partial w_0} (y_i - [w_0 + w_1 x_0])^2$

same for w_1

→ $\boxed{RSS(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2}$

⇒ Taking the derivative w.r.t. w_0

$= \sum_{i=1}^{N} 2 (y_i - [w_0 + w_1 x_i]) \cdot (-1)$

$= -2 \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])$

⇒ Taking the derivative w.r.t. w_1,

$$= \sum_{i=1}^{N} 2\left(y_i - [w_0 + w_1 x_i]\right) \cdot (-x_i)$$

$$= -2 \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i]) x_i$$

⇒ Putting it together:

$$\nabla RSS(w_0, w_1) = \begin{bmatrix} -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] \\ -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] \cdot x_i \end{bmatrix}$$

Approach 1: set gradient = 0

$$\nabla RSS(w_0, w_1) = \begin{bmatrix} -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] \\ -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] x_i \end{bmatrix}$$

Top term:

$$\hat{w}_0 = \frac{\sum_{i=1}^{N} y_i}{N} - \hat{w}_1 \frac{\sum_{i=1}^{N} x_i}{N}$$

($\frac{\sum y_i}{N}$ ← average house sales price; $\hat{w}_1$ ← estimate of the slope; $\frac{\sum x_i}{N}$ ← average sq. ft.)

Bottom term:

$$\sum y_i x_i - \hat{w}_0 \sum x_i - \hat{w}_1 \sum x_i^2 = 0$$

$$\hat{w}_1 = \frac{\sum y_i x_i - \frac{\sum y_i \sum x_i}{N}}{\sum x_i^2 - \frac{\sum x_i \sum x_i}{N}}$$

NOTE:
You have to find these 4 terms:
1) $\sum_{i=1}^{N} y_i$ 2) $\sum_{i=1}^{N} x_i$
3) $\sum_{i=1}^{N} y_i x_i$ 4) $\sum_{i=1}^{N} x_i^2$

Approach 2: Gradient Descent

Interpreting the gradient: actual house sales observation; predicted value $\hat{y}_i(w_0, w_1)$

$$\nabla RSS(w_0, w_1) = \begin{bmatrix} -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] \\ -2 \sum_{i=1}^{N} [y_i - (w_0 + w_1 x_i)] x_i \end{bmatrix}$$

$$\nabla RSS(w_0, w_1) = \begin{bmatrix} -2 \sum_{i=1}^{N} [y_i - \hat{y}_i(w_0, w_1)] \\ -2 \sum_{i=1}^{N} [y_i - \hat{y}_i(w_0, w_1)] x_i \end{bmatrix}$$

while not converged

$$\begin{bmatrix} w_0^{(t+1)} \\ w_1^{(t+1)} \end{bmatrix} \leftarrow \begin{bmatrix} w_0^{(t)} \\ w_1^{(t)} \end{bmatrix} + 2\eta \begin{bmatrix} \sum_{i=1}^{N} [y_i - \hat{y}_i(w_0^{(t)}, w_1^{(t)})] \\ \sum_{i=1}^{N} [y_i - \hat{y}_i(w_0^{(t)}, w_1^{(t)})] x_i \end{bmatrix}$$

- If overall, underpredicting $\hat{y}_i$, then $\sum_i [y_i - \hat{y}_i]$ is positive
  → w_0 is going to increase
  similar intuition for w_1, but multiply by x_i

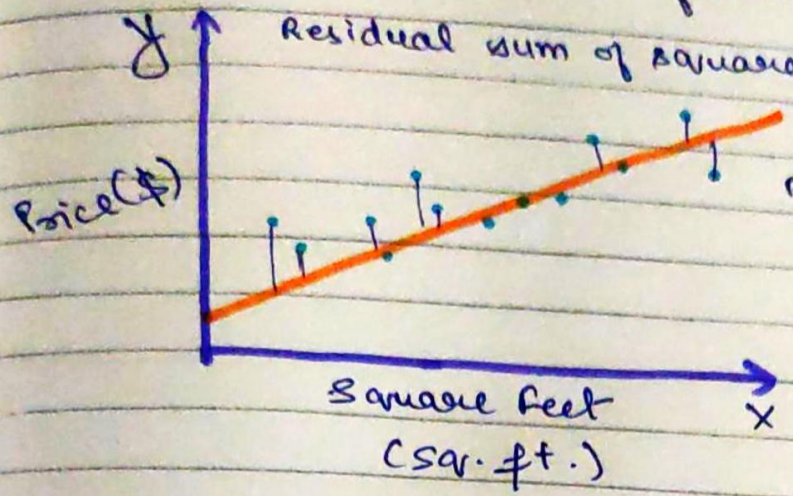
Comparing the approaches

- For most ML problems, cannot solve gradient = 0
- Even if solving gradient = 0 is feasible, gradient descent can be more efficient.
- Gradient Descent relies on choosing stepsize & convergence criteria.

2. Influence of high leverage points

Asymmetric errors

→ Symmetric cost functions

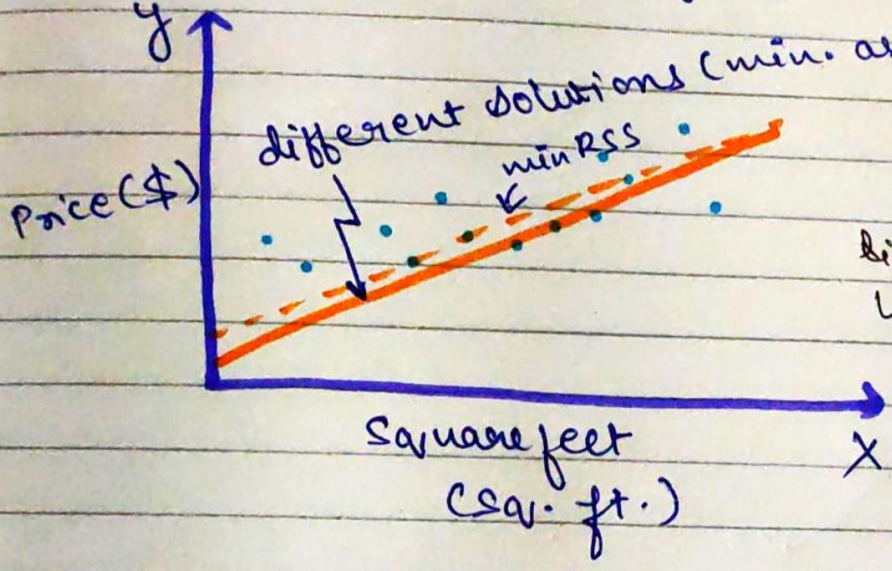


$$RSS(w_0, w_1) = \sum_{i=1}^{N} (y_i - [w_0 + w_1 x_i])^2$$

↑

Assume cost of over-estimating sales price is same as under-estimating.

→ Asymmetric cost functions



What if cost of light house too high has bigger cost?

Too high → no offers ($=0)

Too low → offers for lower $