Name: L. Vyraswathaman, ID: 2019AIML618, Course: PCAM ZC211
Regression EC-3R C1

Q:01.

Given dataset with size: n

No. of predictors $\vec{x}$ = 10 [vector of 10; dimension 10×1]

$\Rightarrow x^{(i)} = \begin{bmatrix} x_1^{(i)} \\ \vdots \\ x_{10}^{(i)} \end{bmatrix}_{(10\times1)}$ where $^{(i)}$ → superscript is the first sample out of 'n'

$y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(n)} \end{bmatrix}_{(10\times1)}$; we add $x_0 = 1$ as a standard feature.

$\Rightarrow x^{(i)} = [x_0^{(i)} \dots x_{10}^{(i)}]^T$

Thus, we have the data matrix $X = \begin{bmatrix} - x^{(1)T} - \\ - x^{(2)T} - \\ \vdots \\ - x^{(n)T} - \end{bmatrix}_{(11\times n+1)}$

Simple regression model is given as $\hat{y} = \theta^T x$. where θ is the model parameter vector for each feature $\Rightarrow \theta = [\theta_0\ \theta_1 \dots \theta_{10}]^T_{(11\times1)}$

$\Rightarrow \hat{y} = \theta_0 x_0 + \theta_1 x_1 \dots + \theta_{10} x_{10} = \theta^T x$ [in matrix-vector notation]

The empirical risk for the model is defined as:

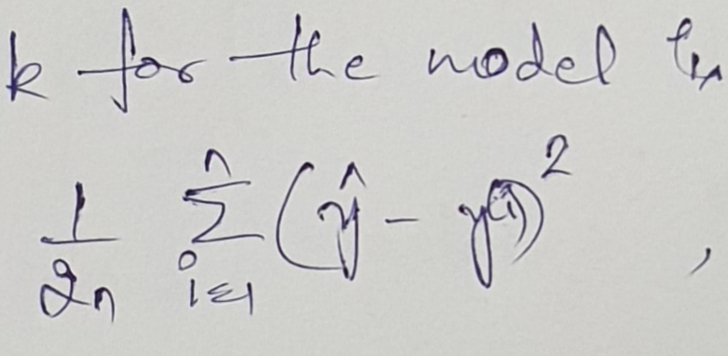
$$J(\theta) = \frac{1}{2n} \sum_{i=1}^{n} (\hat{y} - y^{(i)})^2$$, $\hat{y}$ = fn. of x; parameterized by θ

$\Rightarrow \hat{y} = h_\theta(x)$.

The goal of linear regression is to minimize the empirical risk

$\Rightarrow$ ERM [Emp. Risk minimization] of $J(\theta) \Rightarrow \min_\theta J(\theta) := \frac{1}{2n} \sum_{i=1}^{n} [h_\theta(x^{(i)}) - y^{(i)}]^2$

#1 If 'n' is small ≈ O(10) $\Rightarrow$ there is a chance of overfitting



given 2 points A & B, a line / quadric curve of any shape can fit A & B

#2 n is large, overfitting can still occur with a large degree of polynomial chosen.

Hence we use regularization / shrinkage / stabilization to add bias to the model.

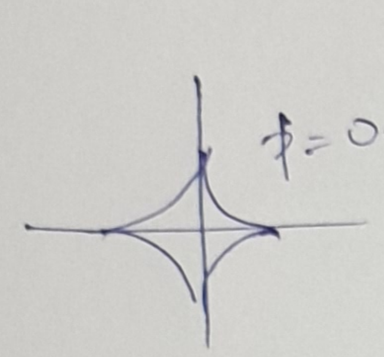
Name : Vyraswathaman, ID: 2019AIML612, Course: PCAMLC211
Regression EC-3R C1

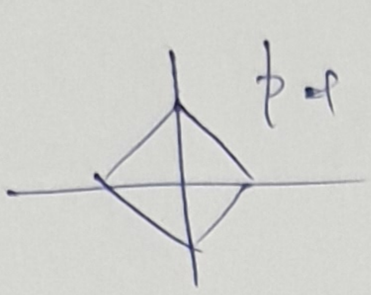
Q: a Contd...

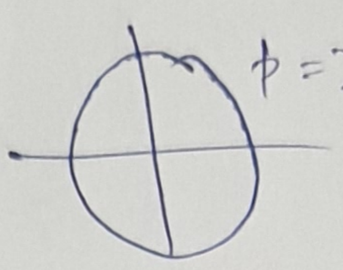
Hence, we minimize variance with added bias & the modelling is called | Regularized Empirical Risk Minimization. |

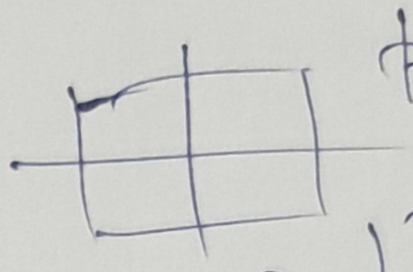
Regularizer: An Lp norm is defined as $\left[\left(\sum_{i=1}^{n} \theta_i\right)^{\phi}\right]^{1/\phi}$

where ϕ can range from $0 \to \infty$.









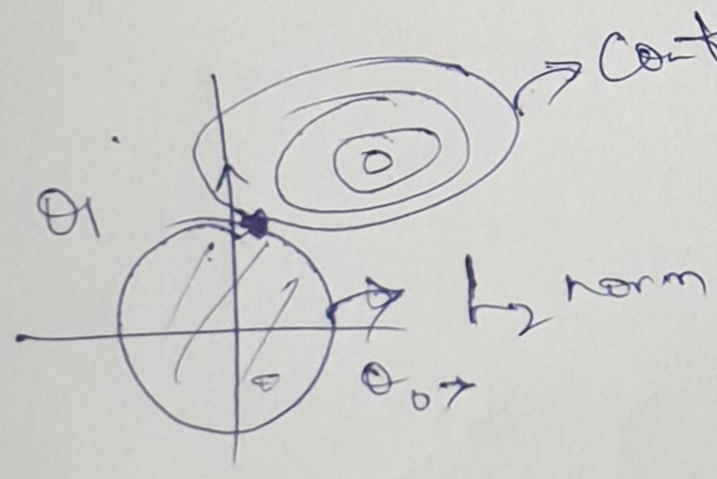
for convexity | $\phi \geq 1$ |

A) Ridge Regression: In Ridge regression, the norm $\phi = 2$ & the shape formed by the constraint is a circle.

$\Rightarrow \sqrt{\sum_{i=1}^{n} \theta_i^2} = L_2$ & the constraint minimization fn.

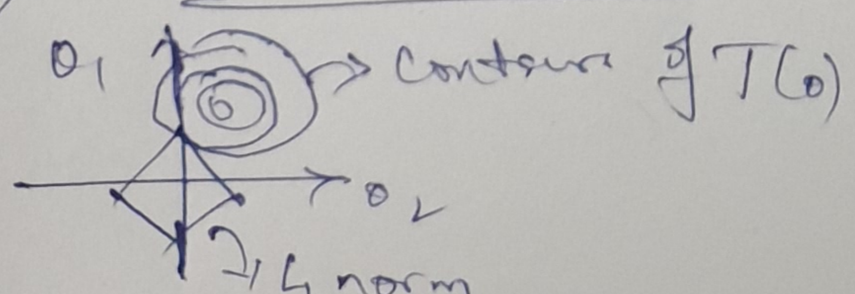
$$J(\theta) = \frac{1}{2n}\left[\sum_{i=1}^{n}\left[h_\theta(x^{(i)}) - y^{(i)}\right]^2 + \lambda \sum_{i=1}^{n} \theta_i^2\right]$$

→ we do not ~~bias~~ regularize θ_0 & λ is the Lagrange multiplier for the constraint.



∴ Note that the solution is when the contour of J touches the circle & the gradients are proportional

B) Lasso Regression: Norm = 1 $\Rightarrow L_1 = \left|\sum_{i=1}^{n} \theta_i\right| =$



$$J(\theta) = \frac{1}{2n}\left[\sum_{i=1}^{n}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \lambda \sum_{i=1}^{n} |\theta_i|\right]$$

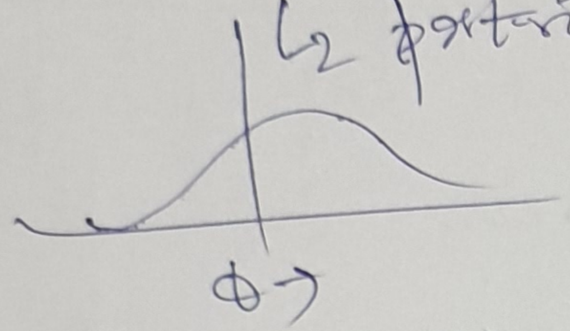
again we do not penalize θ_0 / [redacted] intercept term

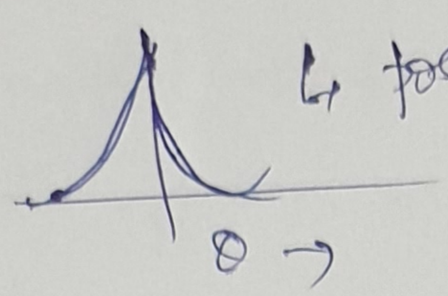
Name
Nyamathulla, ID: 2019AIML618, Course: PGA+C211
Regression EC-3RC4

Q.07 Cont'd...

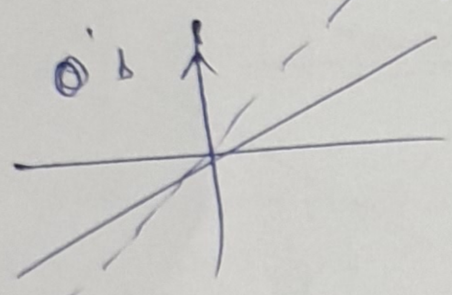
Comparison: L_2 allows smooth solutions & convergence as it is differentiable & solutions are dense

L_1 allows sparse solutions with some θ_i soft thresholded to 0.

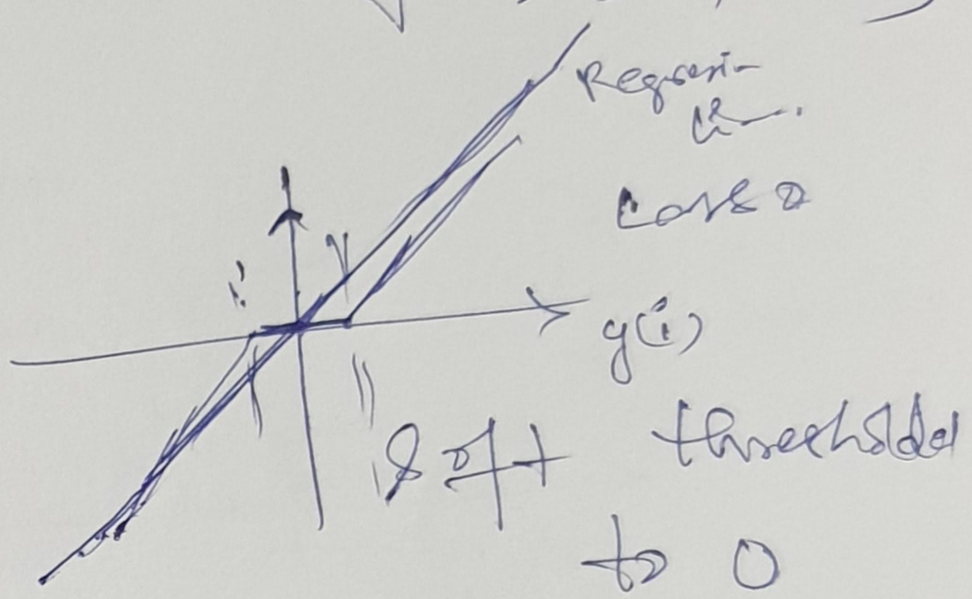




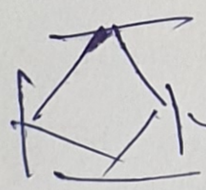
[posterior distributions of L_2 & L_1 norms]



θ_i all θ_i are systematically moved towards 0



This is because, from the Regression diagram in 1 & 2) the axis alignment of Ridge is with probability '0' for θ_i to be 0.

While Lasso has  shape & is not differentiable this allows many θ_i's to be 0 like L_1 to still converge.

Name : Vyravasathanam, ID: 2019-AIML-618, Course ID: PGAM ZC244
Regression EC-3RC1

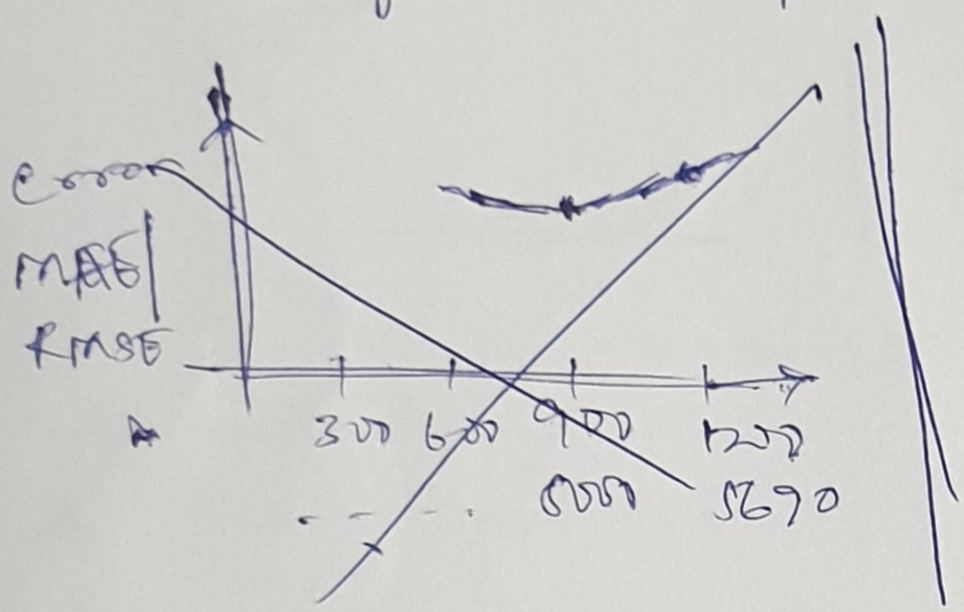
Q.02.

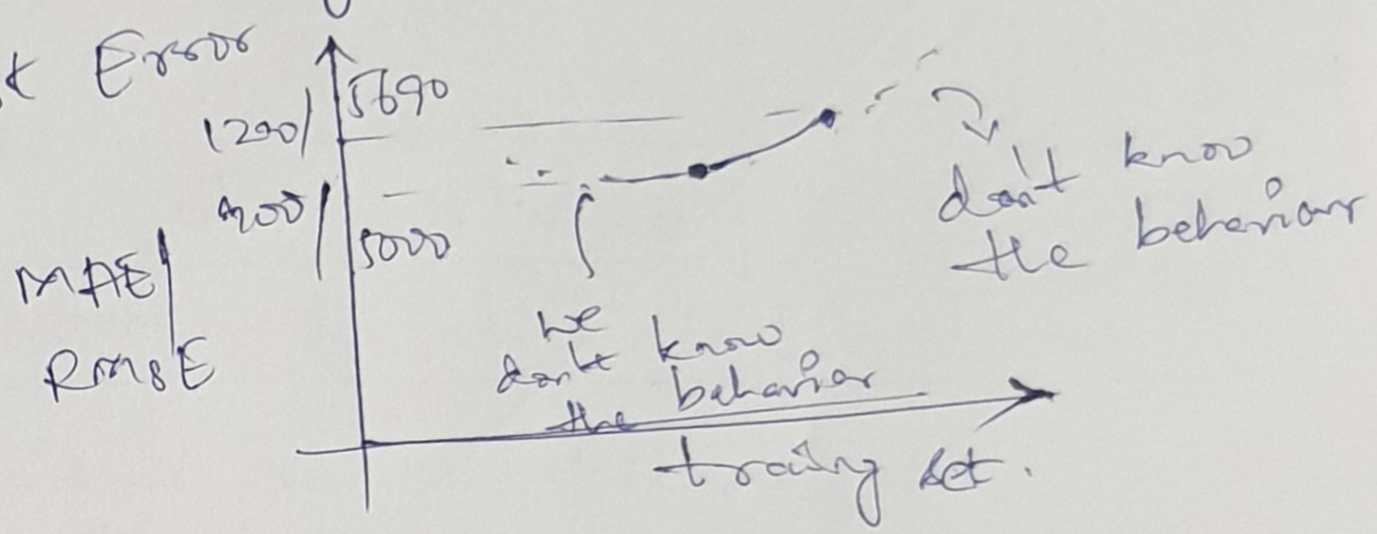
	Model 1	Model 2
MAE :	900	1200
R^2 :	68	71
RMSE :	8000	5690.
F.Score:	23	27.

Assumptions

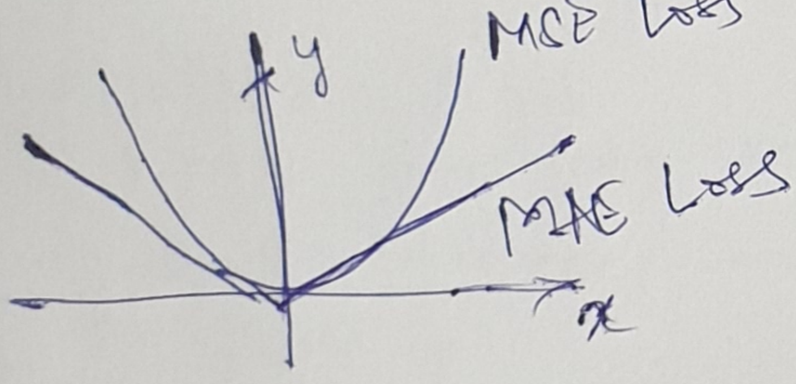
RMSE is calculated correct as there is huge difference b/w MAE & not in RMSE

If we plot the model test error against MAE / RMSE we get this plot approximately.





→ With the info. I cannot claim one model over another
→ the deductions are below with regards to the data given.



→ The MAE & MSE plots are as shown on the left.
→ While MAE deviates less for outliers or wrong samples, MSE grows & penalizes wrong examples / outliers heavily.

CONCLUSIONS

MODEL - 1	MODEL - 2
1) Though R^2 (% variability explained by the model is lesser), so is the MAE & RMSE => lower generalization error / risk	1) R^2 is better than Model 1 but the risk associated is more with > MAE & RMSE
2) F-Score > 1 => H0 (No model parameters are related to response is rejected)	2) F-Score > 1 => H0 rejected. All model parameters are significant.
3) Choose this model to have lower risk & better generalization across other data	3) Choose this model when more variability needs to be explained & model with better performance & higher risk.
4) Slightly lower variance & higher bias model	4) Slightly greater variance & lower bias model

Name: Vyravasthanam, ID: 2019AIM2618, Course: PGAM7CP11 Regression EC-23 C1

Q.03 : 1 billion dataset, Algorithm comparison.

At the outset → for $O(10^9)$, gradient methods will be too slow. Prefer second order over first order gradient methods

---

Ex: Stabilized bundle methods (cutting plane) & LBFGS & [BFGS will not work storing such huge data in memory for calculating (Hessian).]

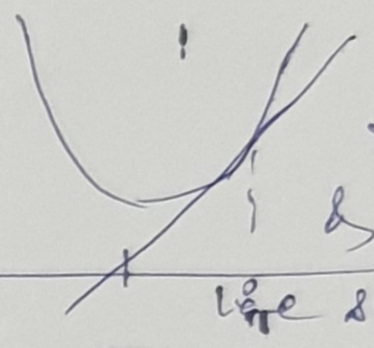
cutting planes.

Newton-Raphson with line search (exact wolfe conditions & necessary & sufficient descent)

works pretty well too

$$\theta^{t+1} = \theta^t - \alpha \frac{f'(\theta)}{f''(\theta)}$$

$$\alpha \left[\theta^{t+1} = \theta^t - \alpha H^{-1} \nabla f(\theta)\right], \quad \alpha = \text{line search obtained by}$$



line search

For the given comparison between batch & mini batch.

→ The algorithms & comparisons are as follows.

Vanilla / Batch G.D	Mini Batch G.D.
i) Algorithm, { for ('N' iterations or till convergence); → calculate cost → update parameter vector. → check tolerance / convergence → repeat }	i) Algorithm, { for ('m' iterations or till convergence); → Create shuffled batches (batch size is a parameter) → { for (each batch); → calalate cost, update parameter vector } → check convergence → repeat.
2) cost m = no. of samples. $J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left[h_\theta(x^{(i)}) - y^{(i)}\right]^2$	2) Cost $J(\theta) = \frac{1}{2 \times \text{batch size}} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$
3) See that batch descent iterates over all billion samples for a single step of descent or parameter calculation / update.	3) Mini. batch iterates only through the batch size, (typically chosen to be around 10-100 & can be tuned) for one step of param update / descent.

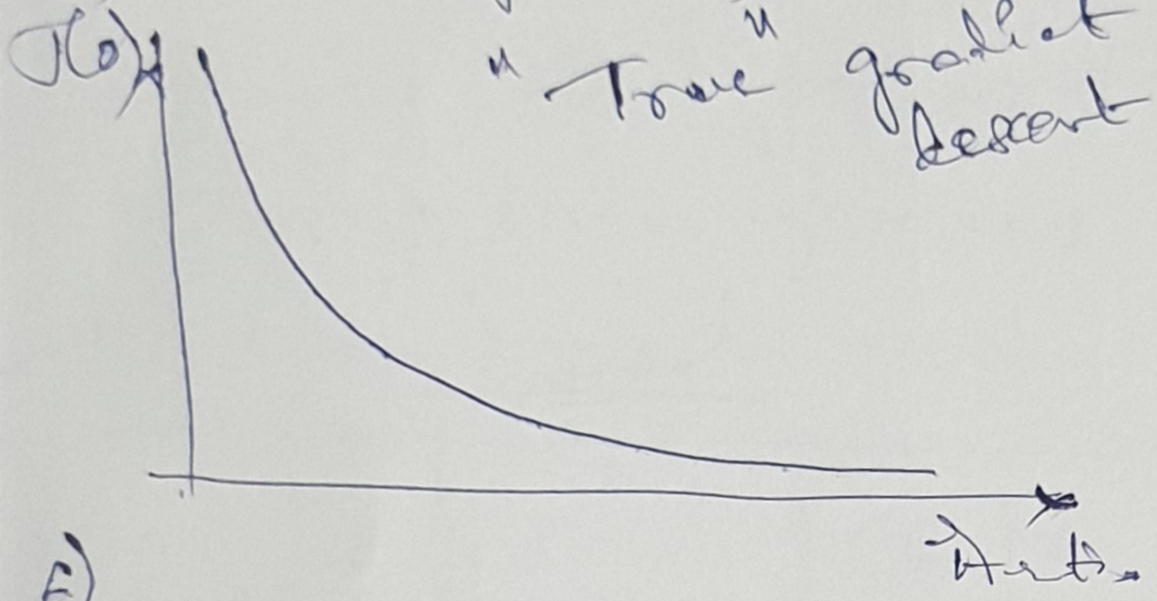
Name: Vyravvatlanar, ID: 2019AIML618, Course: PCAMTC24 Regression EC-R3C1

Q:03 Contd...

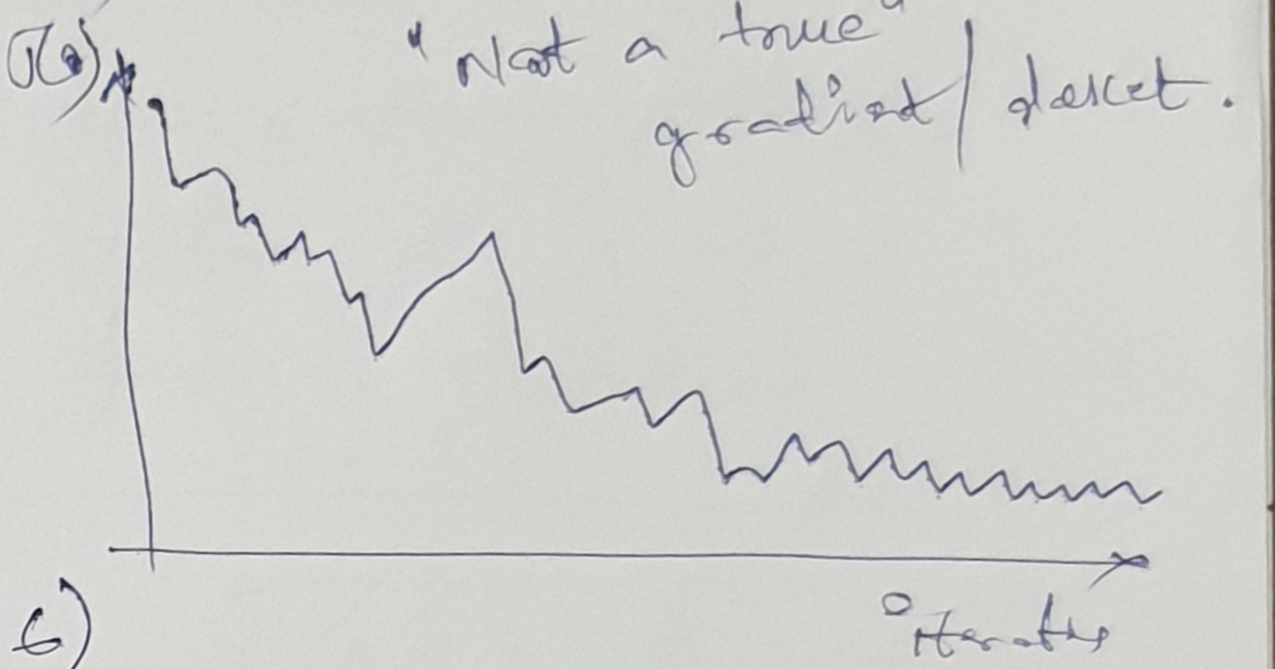
4) Calculating J(θ) will not be feasible & need to use map-reduce & other techniques & parallel for it to work

A) Batch size can vary to get good convergence rates. Can also use map reduce for parallel batch-split-updates.

5) Plot of J(θ) vs. iterations "True" gradient descent.



5) Plot J(θ) vs. iterations "Not a true" gradient/descent.



6) Assuming correct choice of 'α' batch converges to the minimum but it can take forever to converge on 1 billion set. May need ten of thousands of steps to converge ≈ O(10000)

6) Assuming correct 'α' choice, mini-batch converges faster compared to batch descent. Typically, chosen also good batch size, converges in 10-80 iterations (batch iteration)

7) Model created will have slightly different/better params than mini-batch (smooth convergence, see 5)

7) Model params will be slightly different as model, when it approaches minima, it oscillates as it is not a true gradient, but is typically near the minimum, not much variation see (5)

8) This will be very slow to converge if at all & we will not use this for dataset of ≈ O(10^10)

8) Choosing dynamic 'α' to decay too as it approaches minima, can give very close result to batch descent while being very fast as well. That is typically not done.

$$\alpha = \frac{\lambda_1}{\text{iteration} + \lambda_2}$$

Name: Vyrosuothaman, ID: 2019AIML618, Course: PGAIML2U
Regression EC-R3C1

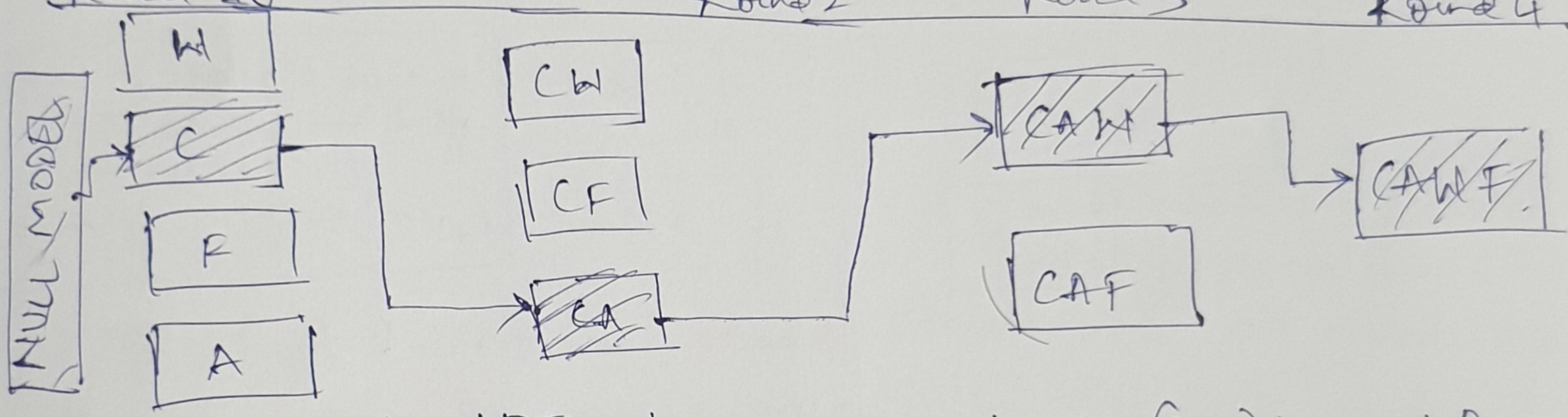
Q: 04

A) Let's say, Body weight = W, Caloric Intake = C, Fat Intake = F
Age = A, Blood Cholestrol = Y

2) features = W, C, F, A, Output = Y

Forward selection algorithm picks best features starting with 1, 2 .. no. of features. Null model is first built assuming all hypotheses to be true & ~~doing Bonty~~ step by step new features are added till a good accuracy score is reached or feature threshold is reached.

Round 1 | Round 2 | Round 3 | Round 4.



→ In General for 'n' predictors, $1 + 1 + (n-1) + \dots + n$
models are built = $1 + \sum_{i=1}^{n} n = 1 + n(n+1)/2$.

→ In this case, 4 features ⇒ $1 + (4 \times 5/2) = 11$ models (including null model)

→ The models are shown above with features at each step (round)

→ Round 0: Null model
→ Round 1: C be the best feature,
→ Round 2: C with A be best,
→ Round 3: C with AG be best,
finally, CAWF are chosen in order.

subject to some heuristic such as AIC, BIC or adjusted R^2 after adjusting for interactions of other predictors

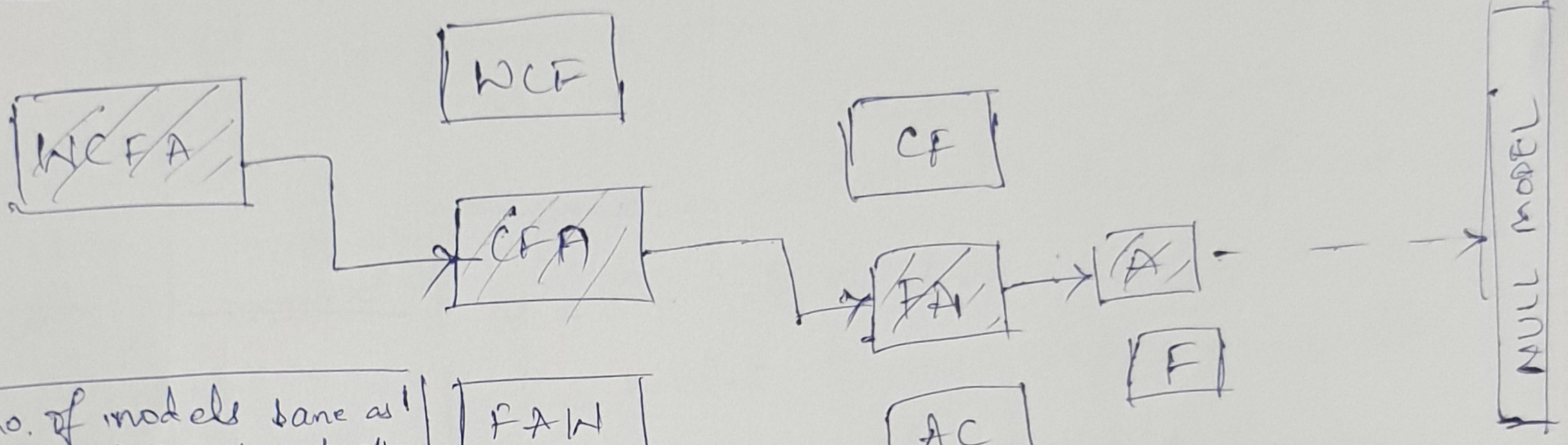
→ The above example was just one of the models that could have been chosen.

→ A best subset of 1 feature would be C, 2 → CA, 3 → CAW & so-on. No. of features needs to be decided

Name: Vyraswathenam, ID: 2019AIML618 Course: PCAM Sec 211
Regression

DS4 contd. —

B) Backward selection fits all features & then sequentially reduces each, based on how many features to keep & a accuracy score based thresholding such as AIC, BIC & adjusted R^2, etc.
Round 1 Round 2 Round 3 Round 4



→ no. of models done as in fwd. selection but in opp. direction.
$n + (n-1) + \cdots + 1 + 1 = \frac{n(n+1)}{2} + 1$

⇒ no. of models $= \left(\frac{4 \times 5}{2}\right) + 1 = 11$ including NULL model

→ Opp. of fwd. selection. Model starts with all features dropping one by one sequentially till null model is reached.
→ E.g. as shown, note, this is just one possible sequence.
→ Round 1: WCFA (all), 2: CFA chosen as best feature (W is dropped)
→ Round 3: FA (C is dropped), 4) F is dropped & A is chosen

Comparison: Unless no. of features is chosen to be a stopping criteria (like 3 best, 2 best, etc.) all features are chosen. Or a threshold accuracy score should be the stopping condition.
Fwd. selection does not is greedy & a feature once selected is fixed & cannot be dropped.
Bck. selection can have max. k features where $k < n$ when no. of features (n) > no. of samples (k).

Name : Vynaenathamam, ID: 2019AIML518 ; Course : PGAM7C24 Regression EC-RB C1

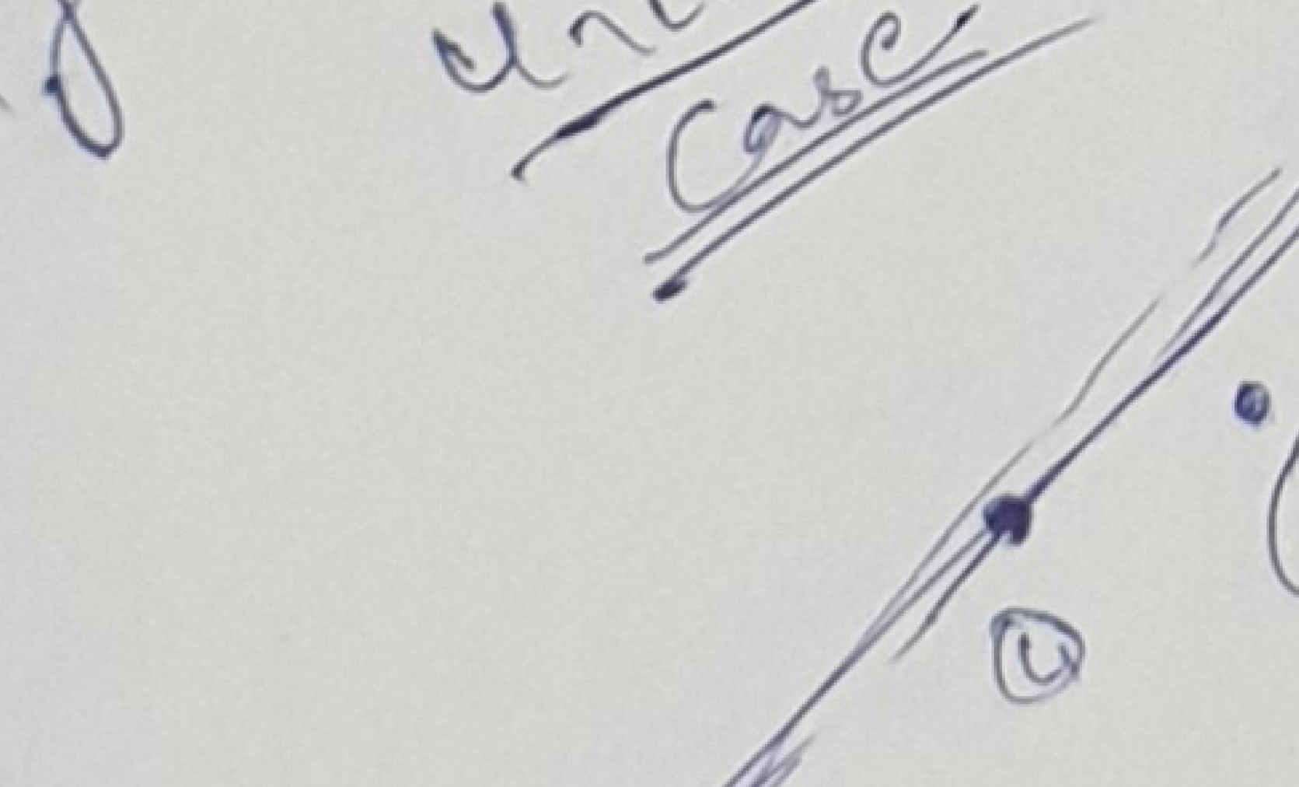
Q.05: Examples of univariate & multiple regression.

Practical example of regression: Predict house price based on 'n' features, n = 1 ... n

1) Simple regression / univariate
Let's assume that house price only depends on the house sq.ft. Let's assume that house price only depends on the feature area. Let's define 'Area' in (m^2) (metres^2) as the feature x. & price of house as feature y which is the target.

	x	y
1.	100	1000
2.	200	2000
3.	500	5000
4.	800	7800
5.	900	9200

We has 5 samples, 1 feature.
Regression model will be defined as
$y = \theta_0 + \theta_1 x$ where θ_0 & θ_1 are the model params representing the y-intercept & slope respectively.
Since this is linear regression, model is the equation of a line.



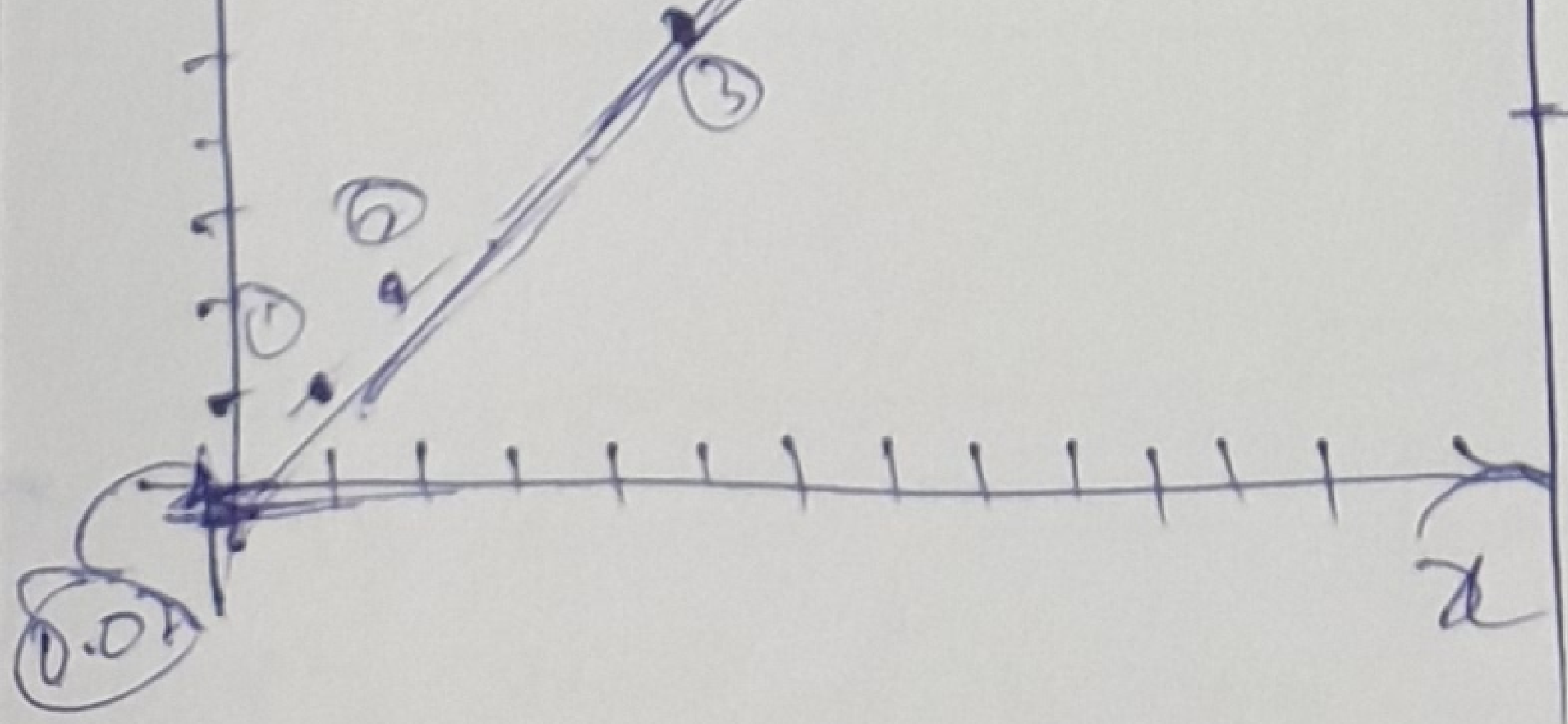
To find the params. normalize the i/p & o/p to '100' scale

x	y
1	1
2	2
5	5
8	7.8
9	9.2

$$\theta_1 = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{\Sigma(x_i - \bar{x})^2}$$

$$\bar{x} = \frac{1+2+5+8+9}{5} = \frac{25}{5} = 5$$

$$\bar{y} = \frac{1+2+5+7.8+9.2}{5} = \frac{25}{5} = 5$$



$$\Rightarrow \theta_1 = \frac{(1-5)(1-5) + (2-5)(2-5) + (5-5)(5-5) + (8-5)(7.8-5) + (9-5)(9.2-5)}{(1-5)^2 + (2-5)^2 + (5-5)^2 + (8-5)^2 + (9-5)^2}$$

$$\Rightarrow \theta_1 = \frac{16+9+0+(3\times 2.8)+(4\times 4.2)}{16+9+0+9+16} = \frac{25+25.2}{50} = \frac{50.2}{50} \approx 1 \; (1.004)$$

Name: Hyderwatlamar, ID: 2017HMLG18 Course: PGAM ZE 211
Regression EC-R3 C1

Q.05. Contd...

$\theta_0 = \bar{y} - \theta\bar{x} = 5 - 5 \times 1.004 = 0.02 \Rightarrow \boxed{y = 0.02 + 1.004x}$

$\Rightarrow$ y/price increases 1.004 times with every unit increase in x of area.

2) Multiple regression: Let's say house price depends also on no. of rooms, , width, height & many other features. This becomes multiple regression. Here let's call

no. of rooms $= x_2$
width $= x_3$
height $= x_4$
(area $= x_1$)

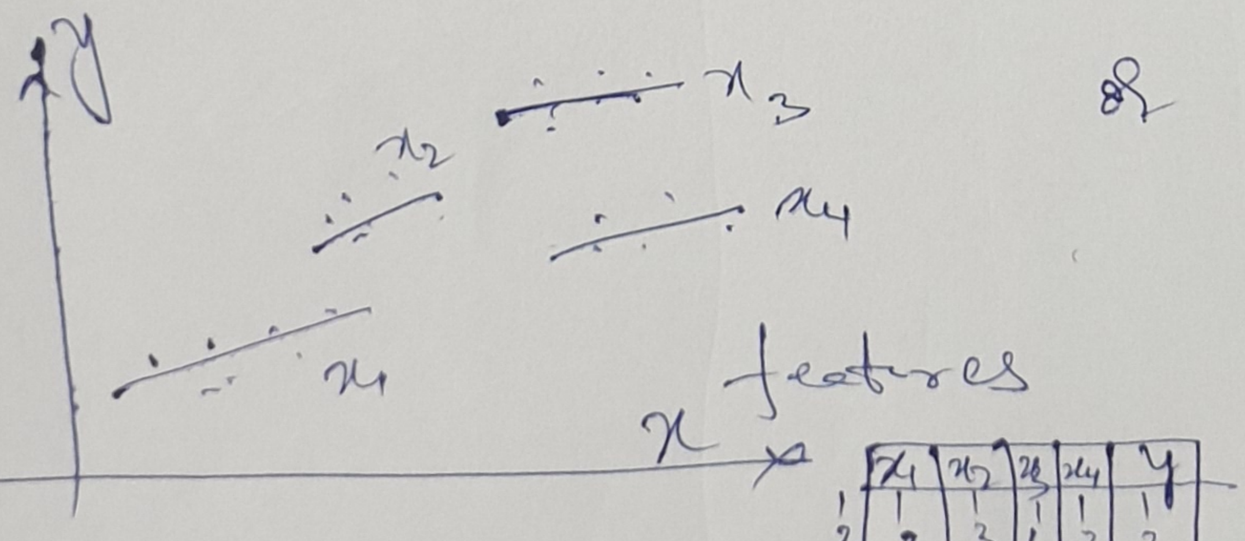
Define $x_0 = 1$ for mathematical convenience

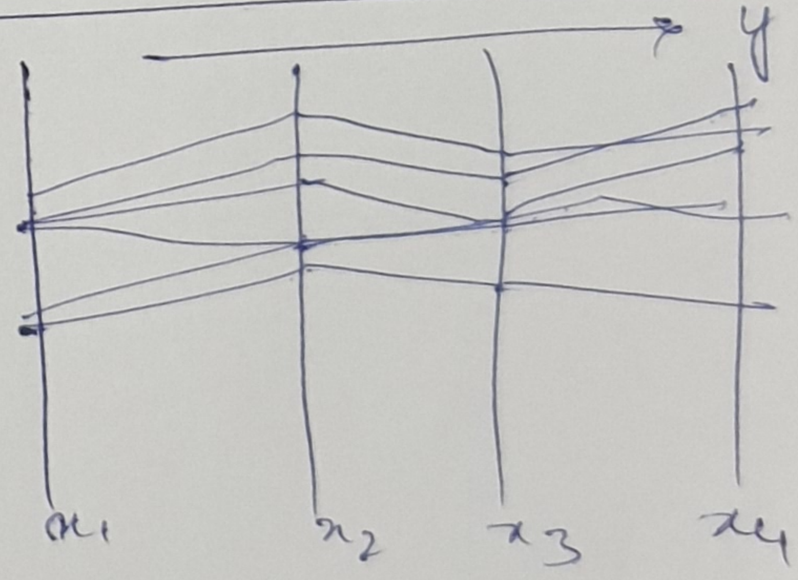
$\Rightarrow y = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4 \dots$

$\Rightarrow \vec{y} = \vec{\theta} \cdot \vec{x}$ or $\vec{y} = \theta^T x$ where.

$y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}_{(m \times 1)}$ m = no. of samples. $x^{(i)} = \begin{bmatrix} x_0^{(i)} \\ \vdots \\ x_n^{(i)} \end{bmatrix}_{(n+1 \times 1)}$ $X = \begin{bmatrix} - & x^{(1)T} & - \\ - & x^{(2)T} & - \\ & \vdots & \\ - & x^{(m)T} & - \end{bmatrix}_{(m+1) \times (n+1)}$

$\Rightarrow \boxed{\vec{y}_{(m*1)} = \theta^T X, \text{ where } \theta = \begin{bmatrix} \theta_0 \\ \vdots \\ \theta_n \end{bmatrix}_{(n+1) \times 1} = \sum_{i=1}^{n} \theta_i x_i}$





	x_1	x_2	x_3	x_4	y
1	1	1	1	1	1
2	2	3	1	2	3
3	2	1	1	3	4
4	3	6	6	1	5
5	1	2	2	1	6

with parallel co-ordinates.

It can be shown that the solution for $\vec{\theta}$ is $\boxed{\vec{\theta} = (X^T X)^{-1} X^T \vec{y}}$

$J(\theta) = \frac{1}{2}(X\theta - y)^T W (X\theta - Y)$ where $w = e^{-\frac{(x_i - x)^T \Sigma^{-1} (x^i - x)}{2}}$ or $\begin{bmatrix} 1 & \dots & 1 \\ 1 & \dots & 1 \end{bmatrix}$ 1's for linear regression
(not deriving due to time constraints)

$\Rightarrow \nabla_\theta J(\theta) = \frac{1}{2}(X\theta - Y)^T (X\theta - Y) = \frac{1}{2}[2X^T X\theta - 2X^T y] = 0 \Rightarrow \boxed{\theta = (X^T X)^{-1} X^T y}$