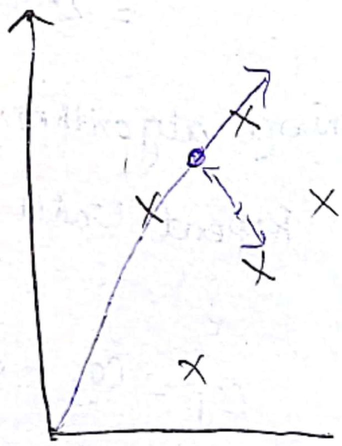
Day-2

Agenda

1. Ridge and Lasso Regression
2. Assumption of Linear Regression
3. Logistic Regression
4. Confusion Matrix
5. Practicals for Linear, Ridge, Lasso & Logistic

Ridge And Lasso Regression

CO



{Traning Data}

Underfitting
(High Bias)

Model fails performs → Training Data

fails in performs well → Testing Data
(High Variance)

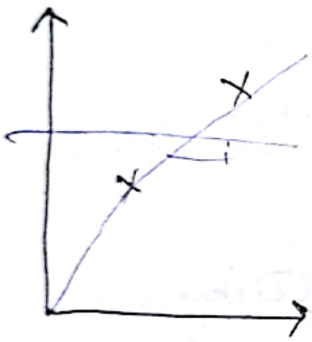
Overfitting
(Low Bias)

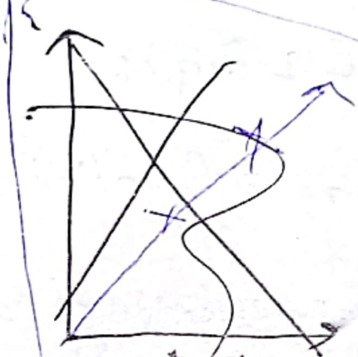
Model perform well → training Data

fails in perform well → Testing Data
(High Variance)

[∵ Bias - will Come only for Traning Data
it it's peforms well it Comes Low &
less performances it comes high Same for
Variance stuff too [Variance - only for
Testing Data]]

MODEL-1	MODEL-2	MODEL-3
Training accuracy = 90%	Train = 92%	Train = 70%
Test accuracy = 80%	Test = 91%	Test = 65%
⇓	⇓	⇓
Overfitting	Generalized Model	Underfitting
Low Bias	Low Bias	High Bias
High Variance	Low Variance	High Variance





Cost function

$$= \frac{1}{2m} \sum^{m} (h_\theta(x^i) - y^{(i)})^2 = (\hat{y}^i - y^{(i)})^2$$

$$=$$

$h_\theta(x) = \hat{y}, \theta_0 = 0$

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

$h_\theta(x) = \theta_1 x$ ↳ slope

Let's $\theta_1 = 2$

$$= (\hat{y}_i - y^{(i)})^2 + \lambda(\text{slope})^2$$

(Lasso)

$$= 0 + 1(2)^2$$

$$= 4$$

Ridge (L2 Regularization) [PREVENT OVERFITTING

$$= (\hat{y}^{(i)} - y^{(i)})^2 + \lambda(\text{slope})^2$$

⇓

$$(\text{Small value}) + 1(1.356)^2$$

$$\simeq 3$$

iteration of Hyperparameter to change slope value

R^2, adjust R^2

Lasso (L1 Regularization) → feature Selection

$= (\hat{y}-y)^2 + \lambda |Slope|$

$h_\theta(x) = \hat{y} = \theta_0 + \theta_1 x + \theta_2 x + \ldots + \theta_n x_n$

① Prevents Overfitting } → L1 } Regulization
② feature Selection } → L2 }

{ λ → Cross Validation }

Ridge Regression (L2 Reg):-

Cost function $= (h_\theta(x)^{(i)} - y^{(i)})^2 + \lambda (Slope)^2$

Purpose:- preventing Overfitting

Lasso Regression (L1 Reg):-

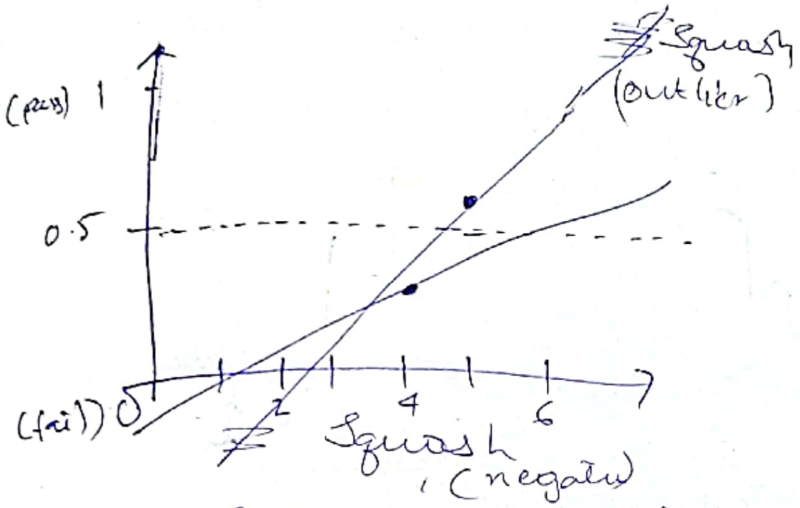
Cost function $= (h_\theta(x^{(i)}) - y^{(i)})^2 + \lambda |Slope|$

Purpose:- i) preventing Overfitting
ii) feature Selection

Assumptions of Linear Regression:-
① Normal (Gaussian Distribution → Model will train well)
② Standarization (Scaloing Data → Z-Score)
③ Linearity
④ Multi-Collinearity

Scanned with OKEN Scanner

Logistic Regression:-
(Binary Classification)



$h_\theta(x) < 0.5 \Rightarrow 0$ (fail)

$h_\theta(x) \geq 0.5 \Rightarrow 1$ (pass)

Decision Boundary Logistic Regression

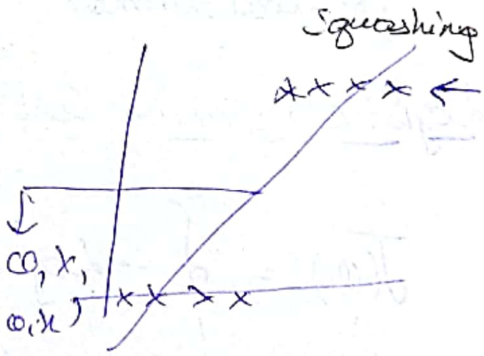
$h_\theta(x) = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n$

$\boxed{h_\theta(x) = \theta^T X}$

1) why don't we use linear (in) this?

2) (squashing is method: overcomes the outlier issues)

[Becoz some values are below 0 & some above 1, (of range 0-1)]



$h_\theta(x) = \theta_0 + \theta_1 x_1$

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

Let $z = \theta_0 + \theta_1 x$

$h_\theta(x) = g(z)$

$h_\theta(x) = \frac{1}{1+e^{-z}}$ (Sigmoid function)

$$\boxed{h_\theta(x) = \frac{1}{1+e^{-(\theta_0+\theta_1 x)}}}$$

Training set

$\{(x^1 y^1), (x^2 y^2) \cdots (x^n y^n)\}$

$y \in \{0, 1\} \rightarrow$ 2 o/p (Binary classif)

$h_\theta(z) = \frac{1}{1+e^{-z}}$

Change parameter θ_1!

Cost function

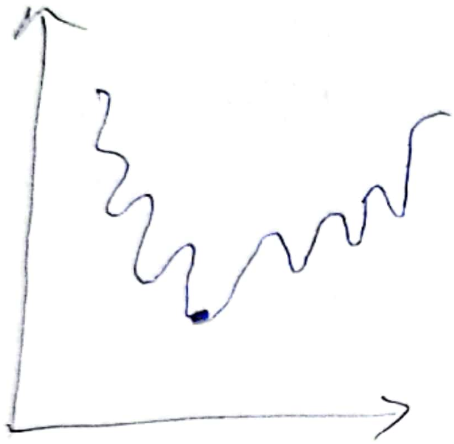
Linear reg $\Rightarrow J(\theta_1) = \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} (h_\theta(x)^i - y^i)^2$

Logistic reg $\Rightarrow h_\theta(x) = \frac{1}{1+e^{-(\theta_1 x)}}$

non-convex func of log reg.

gradient descent

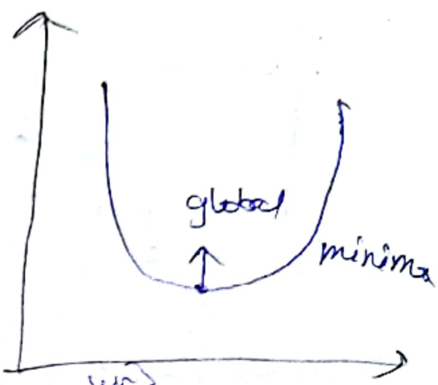
Non-Convex function



(no-local minima)

gradient descent

Convex function.



Logistic regression Cost function :-

[main cost fun]

$h_\theta(x) = \dfrac{1}{1+e^{-(\theta_1 x)}}$

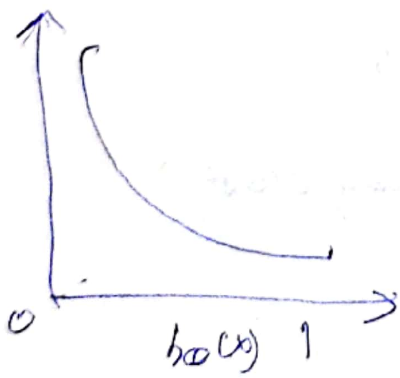
$$J(\theta_1) = \begin{cases} -\log(h_\theta(x)) & y=1 \\ -\log(1-h_\theta(x)) & y=0 \end{cases}$$ ✓

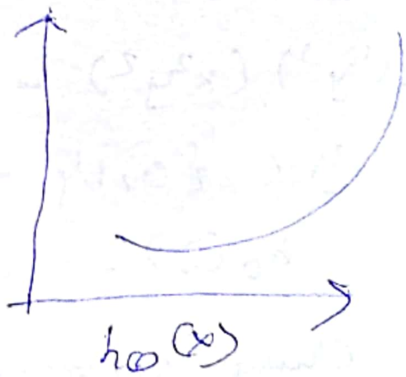
To get local minima & gradient descent we used above one not the $h_\theta(x) = \dfrac{1}{1+e^{(-\theta_1 x)}}$ ×

eg:- Cost = 0

if $y=1$, $h_\theta(x)=1$

if $y=0$





Combine both to get ~~[illegible]~~ &g.d

So for the Example

Cost function given by

$$J(\theta_1) = -\frac{1}{2m} \sum_{i=1}^{m} \left(y^i \log(h_\theta(x)) + (1-y) \log(1-h_\theta(x^i)) \right)$$

$$\left[\because h_\theta(x) = \frac{1}{1+e^{-\theta_1 x}} \right.$$

→ repeat until converge

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

$$\}$$

Performance metrics

① accuracy $= \dfrac{TP+TN}{TP+FN+FP+TN}$

② precision $= \dfrac{TP}{TP+FP}$

③ Recall $= \dfrac{TP}{TP+FN}$

④ f-score $= \dfrac{2(\text{Precision} \times \text{Recall})}{\text{Precision} + \text{recall}}$

	1	0
1	TP	FP
0	FN	TN