→ Linear regression

Supervised Learning

θ = # parameters

m = # training examples

n = # features.

h = hypothesis function.

$$\min_{\theta} \frac{1}{2} \sum_{i=1}^{m} (h_\theta(x^i) - y^i)^2.$$

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

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

$$\boxed{\min_{\theta} J(\theta)}$$

↓

objective.

→ Start with some θ ($\theta = \vec{0}$) keep changing θ to reduce $J(\theta)$ (minimize).

→ Gradient descent

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

$$\frac{\partial}{\partial \theta_i}(J(\theta)) = \frac{\partial}{\partial \theta_i} \frac{1}{2}(h_\theta(x) - y)^2$$

$$= 2 \cdot \frac{1}{2} (h_\theta(x) - y) \frac{\partial}{\partial \theta_i}(h_\theta(x) - y)$$

$$= (h_\theta(x) - y) \frac{\partial}{\partial \theta_i}(h_\theta(x) - y)$$

$$= x_i \cdot (h_\theta(x) - y)$$

$$\theta_i := \theta_i - \alpha (h_\theta(x) - y) x_i$$

learning rate { magnitude of step in direction of steepest descent }

for 'm' training examples

$$\theta_i := \theta_i - \alpha \sum_{j=1}^{m} (h_\theta(x^j) - y^j) \cdot x_i^j$$

Batch Gradient descent → takes more time to converge to the minimum.

Stochastic Gradient Descent

Repeat {

for j = 1 to m {

$$\theta_i := \theta_i - \alpha (h_\theta(x^j) - y^j) x_i^j$$

(for all i).

converges to θ which is near to minimum but not exactly minimum.

$$\nabla_\theta J = \begin{bmatrix} \frac{\partial J}{\partial \theta_0} \\ \vdots \\ \frac{\partial J}{\partial \theta_n} \end{bmatrix} \in \mathbb{R}^{n+1}$$

∇_θ — derivative of gradient of J

J — function of vector of parameters.

Gradient Descent

$$\theta := \underbrace{\theta}_{\mathbb{R}^{n+1}} - \alpha \underbrace{\nabla_\theta J}_{\mathbb{R}^{n+1}}$$

Suppose

$$f : \mathbb{R}^{m \times n} \longrightarrow \mathbb{R} \quad f(A) \ni A \in \mathbb{R}^{m \times n}$$

$$\nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{\partial A_{1n}} \\ \vdots & & \\ \frac{\partial f}{\partial A_{m1}} & \cdots & \frac{\partial f}{\partial A_{mn}} \end{bmatrix}$$

$\longrightarrow$ If $A \in \mathbb{R}^{n \times n}$.

$$tr(A) = \sum_{i=1}^{n} A_{ii}$$

$$\boxed{tr(AB) = tr(BA)}$$

→ $f(A) = tr(AB)$.

$$\nabla_A tr(AB) = B^T.$$

$$tr\ A = tr\ A^T.$$

If $a \in \mathbb{R}$ $tr(a) = a$.

→ $$\nabla_A tr\ ABA^TC = CAB + C^TAB^T.$$

→ $X \longrightarrow$ design matrix.

$$X = \begin{bmatrix} (x^1)^T \\ (x^2)^T \\ \vdots \\ (x^m)^T \end{bmatrix}$$

$$X\theta = \begin{bmatrix} (x^1) \\ \vdots \\ (x^m) \end{bmatrix} \theta$$

$$= \begin{bmatrix} (x^1)^T\theta \\ \vdots \\ (x^m)^T\theta \end{bmatrix}.$$

$$\vec{y} = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix}.$$

$$X\theta - y = \begin{bmatrix} h(x^1) - y^{(1)} \\ \\ h(x^m) - y^{(i)} \end{bmatrix} \in \mathbb{R}^m$$

$$z^T z = \sum_i z_i^2$$

$$\frac{1}{2}(X\theta - y)^T (X\theta - y)$$

$$= \frac{1}{2} \sum_{i=1}^{m} (h(x^i) - y^{(i)})^2 = J(\theta).$$

$$\rightarrow \nabla_\theta J(\theta) = \vec{0}$$

$$\nabla_\theta \frac{1}{2}(X\theta - y)^T (X\theta - y)$$

$$= \frac{1}{2} \nabla_\theta \left[(\theta^T X^T - y^T)(X\theta - y) \right]$$

$$= \frac{1}{2} \nabla_\theta \left[\theta^T X^T X \theta - \theta^T X^T y - y^T X \theta + y^T y \right]$$

$$= \frac{1}{2} \left\{ \nabla_\theta \operatorname{tr}(\theta \theta^T X^T X) - \nabla_\theta \operatorname{tr} y^T X \theta - \nabla_\theta y^T X \theta \right\}$$

* $\nabla_\theta \operatorname{tr} \underbrace{y^T}_{B} \underbrace{X}_{A} \underbrace{\theta}_{C} = X^T y$

* $\nabla_\theta \operatorname{tr} \theta I \theta^T X^T X$

* $\nabla_\theta \operatorname{tr} A B A^T C$

$$= \underbrace{X^T}_{C^T} \underbrace{X}_{A} \underbrace{\theta}_{B} \underbrace{I}_{} + \underbrace{X^T}_{C^T} \underbrace{X}_{A} \underbrace{\theta}_{B} \underbrace{I}_{}$$

$$\nabla_\theta J(\theta)$$

$$= \frac{1}{2}\left[X^T X\theta + X^T X\theta - X^T y - X^T y \right]$$

$$= X^T X\theta - X^T y = 0$$

$$\boxed{X^T X\theta = X^T y} \rightarrow \text{normal equn}$$

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

Proofs (Algebra)

$$\nabla_A \operatorname{tr} AB = \begin{bmatrix} | & & | \\ a_1 & \cdots & a_n \\ | & & | \end{bmatrix} \begin{bmatrix} | & & | \\ b_1 & \cdots & b_n \\ | & & | \end{bmatrix}$$

assume $A \& B$ are $n \times n$

$A \in \mathbb{R}^{n \times n}$, $B \in \mathbb{R}^{n \times n}$

$$a_i \cdot b_i = \begin{bmatrix} a_{i1} \cdot b_{i1} \\ \\ a_{in} \cdot b_{in} \end{bmatrix} = \begin{bmatrix} a_1 \cdot b_1 & \cdots & a_n \cdot b_n \end{bmatrix}$$

$$= \begin{bmatrix} a_{11} b_{11} & \cdots & a_{n1} b_{n1} \\ a_{12} b_{12} & & a_{n2} b_{n2} \\ \vdots & & \\ a_{1n} b_{1n} & & a_{nn} b_{nn} \end{bmatrix}$$

$$\nabla_A \operatorname{tr} AB = \begin{bmatrix} b_{11} & \cdots & b_{n1} \\ b_{12} & & b_{n2} \\ \vdots & & \\ b_{1n} & & b_{nn} \end{bmatrix}$$

$$= B^T$$

→ to prove

$$\nabla_{A^T} f(A) = \left(\nabla_A f(A)\right)^T$$

$$A = \begin{bmatrix} A_{11} & \cdots & A_{1n} \\ \vdots & & \vdots \\ A_{n1} & \cdots & A_{nn} \end{bmatrix} \qquad A^T = \begin{bmatrix} A_{11} & \cdots & A_{n1} \\ \vdots & & \vdots \\ A_{1n} & \cdots & A_{nn} \end{bmatrix}$$

$$\nabla_{A^T} f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{A_{n1}} \\ \vdots & & \vdots \\ \frac{\partial f}{A_{1n}} & \cdots & \frac{\partial f}{A_{nn}} \end{bmatrix} \qquad \nabla_A f(A) = \begin{bmatrix} \frac{\partial f}{\partial A_{11}} & \cdots & \frac{\partial f}{A_{1n}} \\ \vdots & & \vdots \\ \frac{\partial f}{A_{n1}} & \cdots & \frac{\partial f}{A_{nn}} \end{bmatrix}$$

① ② by defⁿ of $\frac{\partial f(A)}{\partial A}$.

from observation:- ⇒ and by transposing.

$$\nabla_{A^T} f(A) = \left(\nabla_A f(A)\right)^T$$

$$① = [②]^T$$

→ $\nabla_x x^T A x = Ax + A^T x$

$A \in \mathbb{R}^{n \times n}$, $x \in \mathbb{R}^{n \times 1}$

$$Ax = \begin{bmatrix} \text{—} \tilde{a}_1^T \text{—} \\ \vdots \\ \text{—} a_n^T \text{—} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$$

$$= \begin{bmatrix} \tilde{a}_1^T x \\ \vdots \\ \tilde{a}_n^T x \end{bmatrix} = \begin{bmatrix} \tilde{a}_1^T x & \cdots & \tilde{a}_n^T x \end{bmatrix}^T$$

$$x^T A x = \begin{bmatrix} x_1 & \cdots & x_n \end{bmatrix} \begin{bmatrix} \tilde{a}_1^T x \\ \vdots \\ \tilde{a}_n^T x \end{bmatrix}$$

$$= x_1 \tilde{a}_1^T x + \cdots + x_n \tilde{a}_n^T x$$

$$\frac{\partial}{\partial x_1}(x^T A x) = \frac{\partial}{\partial x_1}\left(x_1 \tilde{a}_1^T x + \cdots + x_n \tilde{a}_n^T x \right)$$

$$= \left(\tilde{a}_1^T x + x_1 \tilde{a}_1^T \frac{\partial x}{\partial x_1} + \cdots + x_n \tilde{a}_n^T \frac{\partial x}{\partial x_1} \right)$$

$$= \tilde{a}_1^T x + \left(x_1 \tilde{a}_1^T + \cdots + x_n \tilde{a}_n^T \right) \frac{\partial x}{\partial x_1}$$

$$\frac{\partial (x^T A x)}{\partial x_1} = \tilde{a}_1^T x + \left(x_1 a_{11} + \cdots + x_n a_{n1} \right)$$

$$\frac{\partial x}{\partial x_1} = \begin{bmatrix} \frac{\partial x_1}{\partial x_1} \\ \frac{\partial x_2}{\partial x_1} \\ \vdots \\ \frac{\partial x_n}{\partial x_1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Illy for all i, $i >= 2$ and $i <= n$.

$$\frac{\partial}{\partial x}(x^T A x) = \begin{bmatrix} \tilde{a}_1^T x + (x_1 a_{11} + \dots x_n a_{n1}) \\ \tilde{a}_2^T x + (x_1 a_{12} + \dots x_n a_{n2}) \\ \vdots \\ \underbrace{a_n^T x}_{Ax} + \underbrace{(x_1 a_{1n} + \dots x_n a_{nn})}_{A^T x} \end{bmatrix}$$

$$\frac{\partial}{\partial x}(x^T A x) = Ax + A^T x.$$

$$\boxed{\nabla_x (x^T A x) = Ax + A^T x}$$

→ Derivative of a product in trace

$$\nabla_A \operatorname{tr}(AB) = B^T$$

$$\operatorname{tr}(AB) = \operatorname{tr} \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vdots \\ \vec{a}_n \end{bmatrix} \begin{bmatrix} \downarrow & & \downarrow \\ b_1 & \dots & b_n \\ \downarrow & & \downarrow \end{bmatrix}$$

$$= \operatorname{tr} \begin{bmatrix} \vec{a}_1^T b_1 & \vec{a}_1^T b_2 & \dots & \vec{a}_1^T b_n \\ \vec{a}_2^T b_1 & \vec{a}_2^T b_2 & & \vdots \\ \vdots & & & \\ \vec{a}_n^T b_1 & \vec{a}_n^T b_2 & \dots & \vec{a}_n^T b_n \end{bmatrix}$$

$$= \sum_{i=1}^{n} a_{1i} b_{i1} + a_{2i} b_{i2} + \dots a_{ni} b_{in}$$

∂

$$\frac{\partial \, tr(AB)}{\partial \, a_{ij}} = b_{ji}$$

$$\nabla_A \, tr \, AB = B^T$$

Lecture - 3

Linear Regression

↓ → locally weighted regression.

probabilistic interpretation

↓

Logistic regression.

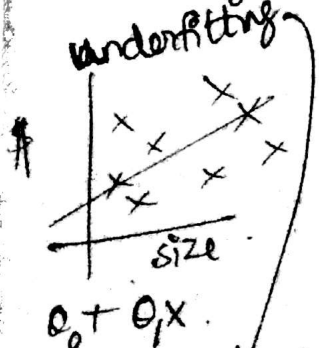
↓ → digression → perceptron learning alg.

Newton's method

$(x^{(i)}, y^{(i)})$ → i^{th} training example

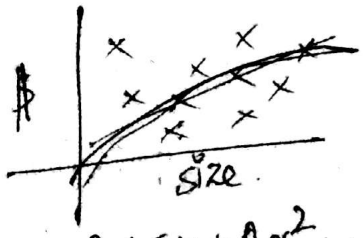
$$h_\theta(x^{(i)}) = \sum_{j=0}^{n} \theta_j x_j^{(i)} = \theta^T x. \quad (x_0 = 1)$$ → convention.

Housing prices.

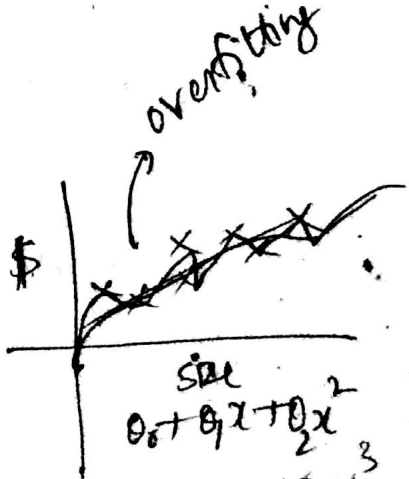


$\theta_0 + \theta_1 x$.

patterns in data that alg. fails to model.



$\theta_0 + \theta_1 x + \theta_2 x^2$



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4 + \ldots \theta_6 x^6$.

algorithm models the idiosyncracies of this particular dataset.

→ "parametric" Learning Algorithm.

Lin. Regression is one example of parametric learning alg.

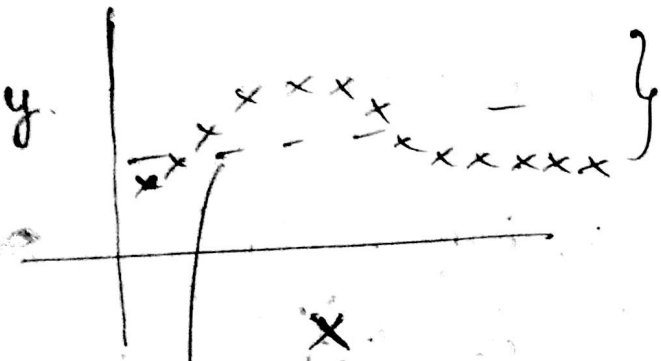
→ "Non-parametric" Learning algorithm.

- no. of parameters grows with training examples (m).

Locally weighted regression (Loess).

↓

We worry less about feature engineering.



} motivating idea for locally weighted regression.

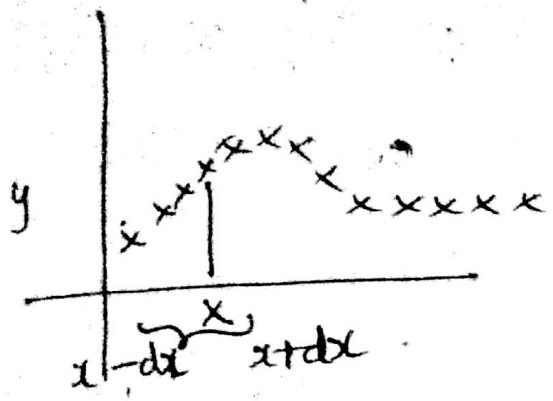
Linear Regression line → poor fit for this data.

* You have to guess/ experiment with features to fit this data.

* To evaluate h at a certain x.

LR → fit θ to minimise $\sum_{i=0}^{i} (y^{(i)} - \theta^T x^{(i)})^2$

and output $\theta^T x^{(i)}$



In LoWoRo, 2) check vicinity of x as well.

ii) You have a bunch of points. ie vicinity of x, then we apply Linear Regression (LoRo) to these points.

iii) You get a LoRo line, evaluate x on the L.R. line.

(2)

* In LoWoRo:

Fit θ to minimize

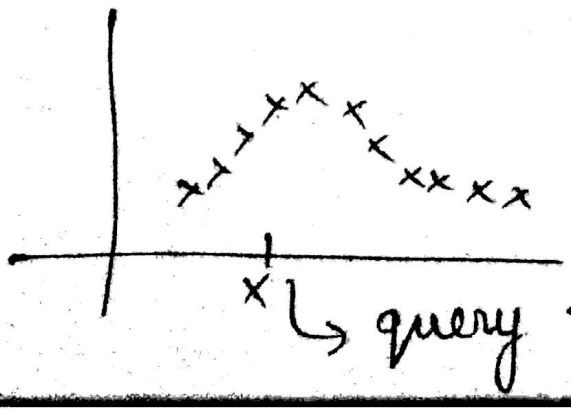
$$\sum_i w^{(i)} \left(y^{(i)} - \theta^T x^{(i)} \right)^2 .$$

where $\boxed{w^{(i)} = e^{\frac{-(x^{(i)} - x)^2}{2\tau^2}}}$ → Not a Normal / Gaussian distribution.

If $|x^{(i)} - x|$ small, Then $w^{(i)} \approx 1$.

If $|x^{(i)} - x|$ large, then $w^{(i)} \approx 0$.

→ τ ↓ bandwidth parameter ↓ controls how fast weights decrease with distance.



→ then points close to x ($|x^{(i)} - x|$ small) get larger weights than ones further ($|x^{(i)} - x|$ large). Points further away from 'x' (query point) would have close to zero contribution.

Research different functions for locally weighted regression. (Andrew Moore K-D Trees for LWR)

→ Probabilistic Interpretation of L.R.

. Why minimize least squares and not any other metric?

Assume.

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$$

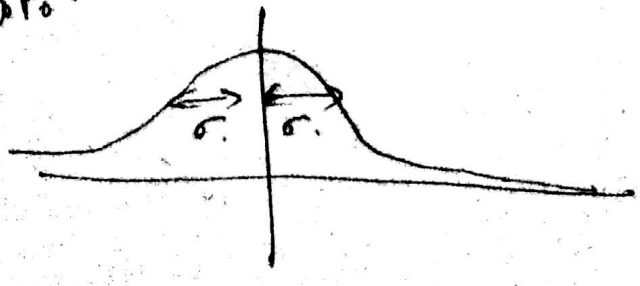
$\varepsilon^{(i)}$ = error term. or random noise.

$\varepsilon^{(i)} \sim \mathcal{N}(0, \sigma^2)$ → Normal distribution.

(0: mean, σ^2: variance.)

$$P(\varepsilon^{(i)}) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\frac{-(\varepsilon^{(i)})^2}{2\sigma^2}}$$

P.D.F.



$$P(y^{(i)} \mid x^{(i)}; \theta) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}}$$

semi-colon

$\varepsilon^{(i)}$

$$y^{(i)} \mid x^{(i)}; \theta \sim N(\mu, \sigma^2).$$

$$\mu = \theta^T x^{(i)}.$$

$$\sigma^2 = \sigma^2.$$

Why is error gaussian?

i) mathematical convenience

ii) distribution of errors is more often Gaussian.

iii) Central limit theorem.

* We are also assuming that error has zero mean.

→ semi-colon, not comma.

$y^i \mid x^i; \theta$ → θ is not an random variable in this case.

We don't know what θ is, but θ is also not a random variable.

Read as:- $y^{(i)}$ given $x^{(i)}$ parametrized by θ.

→ $\varepsilon^{(i)}$ are iid, indepent and identically distributed.

$$L(\theta) = P(\vec{y} \mid x; \theta)$$

↓ Likelihood of θ.

$$= \prod_{i=1}^{m} P(y^{(i)} \mid x^{(i)}; \theta)$$

$$= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\tau^2}}$$

↓ over all training samples.

$$L(\theta) = P(\vec{y} \mid x; \theta)$$

↓ Likelihood means we are viewing $P(\vec{y} \mid \vec{x}; \theta)$ as a function of θ

Maximum Likelihood Estimation:

* Choose θ which maximises ~~probability~~ the likelihood i.e. which makes the data as much as possible.

i.e. choose θ to maximise $L(\theta) = P(\vec{y} \mid \vec{x}; \theta)$

$$l(\theta) = \log L(\theta)$$

$$= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\tau^2}} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\tau^2}}$$

$$= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi\tau^2}} e^{-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\tau^2}}$$

$$\ell(\theta) = m \log\left(\frac{1}{\sqrt{2\pi\tau^2}}\right) + \sum_{i=1}^{m} -\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\tau^2}$$

$$= m \log \frac{1}{\sqrt{2\pi\tau^2}} + \sum_{i=1}^{m} -\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\tau^2}$$

so, maximising $\ell(\theta)$ is the same as minimising $\sum_{i=1}^{m} \frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\tau^2} = J(\theta)$.

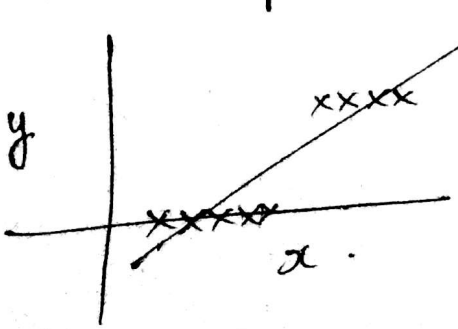
→ So Least Squares regression assuming errors are i.i.d Gaussians is equivalent to M.L.E.

→

Classification

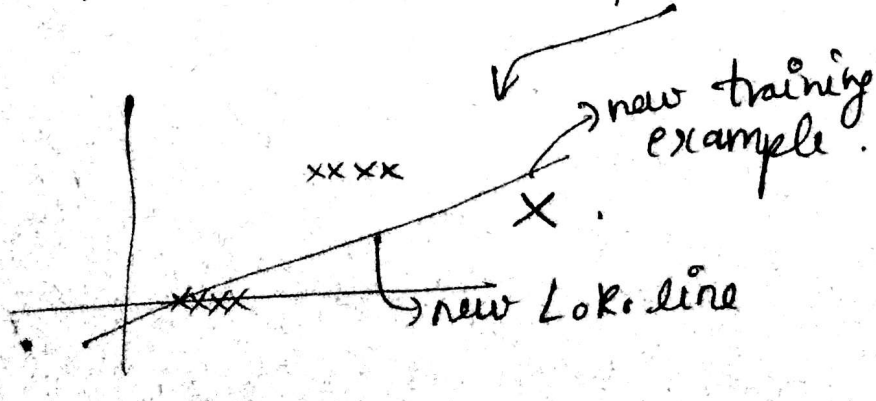
$y \in \{0, 1\}$

↳ to be predicted, discrete values.



→ LR line.

* Bad idea to apply L.R. to classification problems because,



source IP & destination IP → remains same

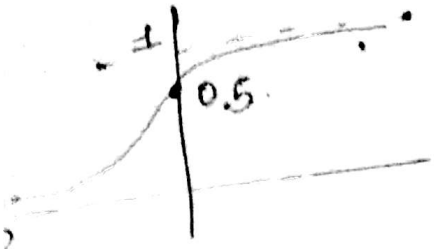
source MAC & destination MAC → change pairwise b/w two hubs

$y \in \{0, 1\}$

$h_\theta(x) \in [0\ 1]$

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$g(z) = \frac{1}{1+e^{-z}} \rightarrow \text{sigmoid function}$$



$$P(y=1 \mid x;\theta) = h_\theta(x)$$

Think of my hypothesis as estimating probability that $y = 1$.

$$P(y=0 \mid x;\theta) = 1 - h_\theta(x)$$

$$P(y \mid x;\theta) = (h_\theta(x))^y (1 - h_\theta(x))^{1-y}$$

$$L(\theta) = P(\vec{y} \mid x;\theta) = \prod_{i=1}^{m} P(y^{(i)} \mid x^{(i)};\theta)$$

$$= \prod_{i=1}^{m} (h_\theta(x^{(i)}))^y (1 - h_\theta(x^{(i)}))^{1-y}$$

→

$$\ell(\theta) = \log(L(\theta))$$

$$= \sum_{i=1}^{m} y^{(i)} (\log h_\theta(x^{(i)})) + (1 - y^{(i)}) \log (1 - h_\theta($$

↓ maximise

Find θ by gradient descent.

$$\theta := \theta + \alpha \nabla_\theta \ell(\theta)$$

$$\frac{\partial}{\partial \theta_j} \ell(\theta) = \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$$

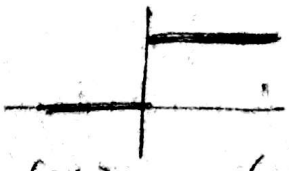
$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} (y^{(i)} - h_\theta(x^{(i)})) x_j^{(i)}$$

different from one in least squares regression.

not a linear function

→ Perceptron

$$g(z) = \begin{cases} 1 & z \geq 0 \\ 0 & \text{otherwise} \end{cases}$$



$$h_\theta(x) = g(\theta^T x)$$

$$\theta_j = \theta_j + \alpha (y^{(i)} - h_\theta(\theta^T x^{(i)})) x_j^{(i)}$$