Lecture 3°

② Supervised learning:- has 3 ingredients

① Training Data $D : \langle X, Y \rangle$

$$X = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_n \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, \quad \vec{x}_i \in \mathcal{X}, \quad y_i \in \mathcal{Y}$$

① Misspecification error.

$$Y = t(\vec{z}) = f(\vec{x}) \rightarrow \delta = h^*(\vec{x}) + \underbrace{\epsilon}_{\text{epsilon/error}}$$

$$= h^*(x) + \underbrace{(f(\vec{x}) - h^*(\vec{x}))}_{\delta} + (t(\vec{z}) - f(\vec{x}))$$

δ: error due to ignorance.

Note:- the space of f is very large.

We need to constraint it.

② $\mathcal{H}$: a candidate set of function

③ A, an algorithm which takes in data $\mathbb{D}$ and set $\mathcal{H}$ and produces a model g.

$$g = A(\mathbb{D}, \mathcal{H})$$

Question: is $f \in \mathcal{H}$ generally? NO

However, there is $h^* \in \mathcal{H}$ which is the closest possible model to f.
(function)

Just because $h^* \in \mathcal{H}$ doesnot mean A will locate it. A will not be perfect and the values of ε will confuse A

Thus $g \neq h^*$ g is the best A can do.

$$y = g(x) + (h^*(\vec{x}) - g(\vec{x})) + (f(\vec{x}) - h^*(\vec{x})) + (($$

Arina Rameasa

estimation error

$$Y = g(\vec{x}) + \underbrace{\left(h^*(\vec{x}) - g(\vec{x})\right) + \left(f(\vec{x}) - h^*(\vec{x})\right)}_{\varepsilon} + \underbrace{\left(f(\vec{z}) - f(\vec{x})\right)}_{\delta}$$

model

ε

e "residual"

$\hat{Y} = g(\vec{x})$ ← prediction of y in setting $\vec{x}$

$e = Y - \hat{Y}$ residual if $\vec{x} \in \mathcal{D}$

Otherwise they're unknown

How to reduce errors:

① δ ignorance error can be reduced by measuring more x_j's (features) of the units that contain

information about $\vec{z}$

* ② Misspecification error can be reduced by...

expand H to include more complicated functions

③ Estimation error can be reduced by increase sample size n.

$$SE[\hat{\theta}] = \sigma/\sqrt{n}$$

$y = \{0, 1\}$

↑ ↑

did not pay back — Paid back mortgage

$p = 1$, x_1 is credit score

$x = [300, 850]$

①

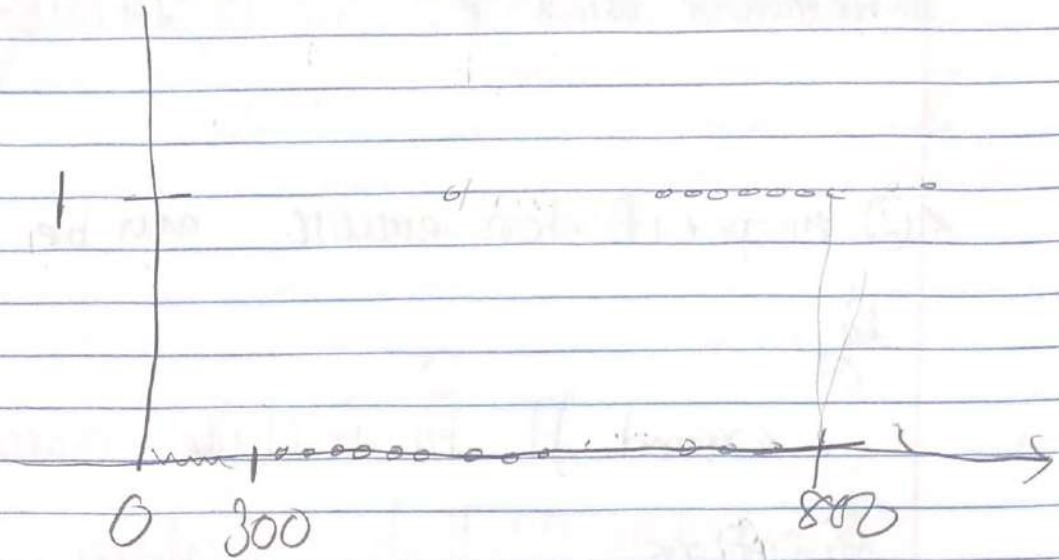
$n = 100$

$D = \langle x, y \rangle =$

y classification

$y = \{0, 1\}$

$$= \left\langle \begin{bmatrix} 810 \\ 390 \\ 750 \\ \vdots \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \end{bmatrix} \right\rangle$$



Threshold model.

$$H = \begin{cases} \mathbb{1}_{x \geq 0} : \theta \in \Theta \end{cases}$$

$\mathbb{1}_{x \geq 0}$ ← indicator function

θ ← Parameter. Sometimes denoted $\beta, w, \ldots$, Others.

Θ ← Capital theta in the greek alphabet

$$\mathbb{1}_A(\omega) = \begin{cases} 1 \text{ if } \omega \in A \\ 0 \text{ if } \omega \notin A \end{cases}$$

e.g. $g(x) = 1_{x \geq 515.3}$

$g(x) = 1_{x \geq 407.9}$

(3) Algorithm A.

Recall,

$g(x) = A(\mathcal{A}, D)$

Let Define

"Misclassification error" —

$$ME := \frac{1}{n} \sum_{i=1}^{n} 1_{g(x_i) \neq y_i}$$

$$= \frac{1}{n} \sum_{i=1}^{n} |\vec{e}_i| \quad \text{residuals.}$$

accuracy :=

$$ACC := 1 - ME$$

A. minimize ME over

$\mathcal{O} \in$ unique x^1s

Sum absolute error (SAE)

$$\frac{1}{n} \sum_{i=1}^{n} |e_i| = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

mean absolute error (MAE)

$$= \frac{1}{n} \sum e_i^2$$

Sum squared error (SSE)

mean squared error (MSE)

Next model

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

X_1: credit score $\in [300, 850]$

X_2: Salary in thousand € R

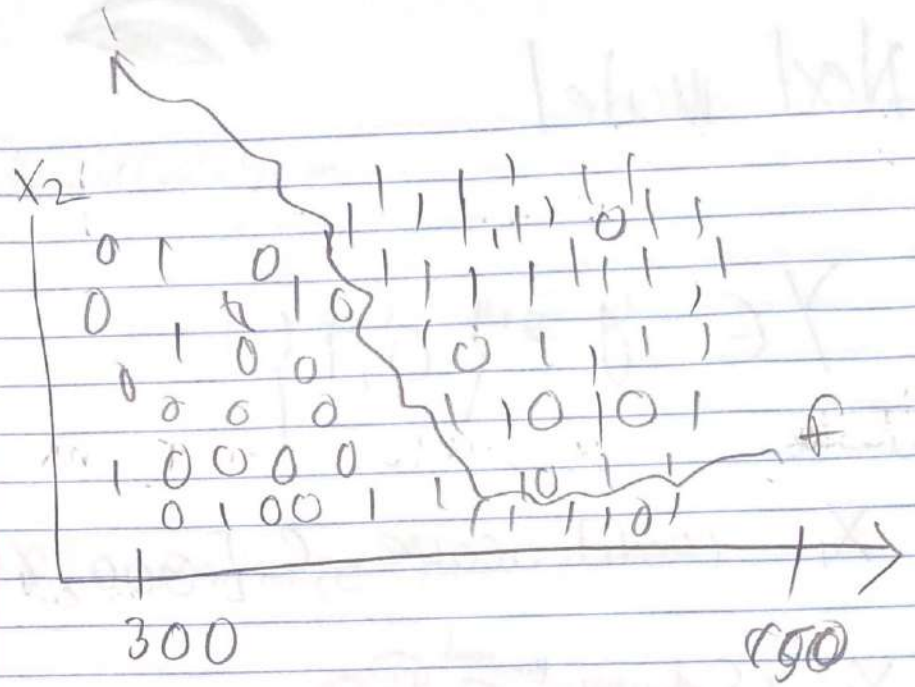
$D =$

x_1	x_2	y
810	63.1	1
390	58.7	0
750	132.6	1
⋮	⋮	⋮

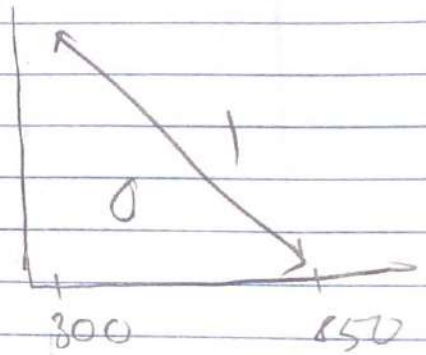
Picture (not to scale)

$$\mathcal{H} = \{ \mathbb{1}_{x_1 > \theta_1 \, \& \, x_2 \geq \theta_2} \; [\text{Not good}] \; \vdots \; \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \in \Theta \}$$

e.g.: $g(x) = \mathbb{1}_{x_1 \geq 650 \; \& \; x_2 \geq 100}$



$$\mathcal{H} = \left\{ \mathbb{1}_{x_2 \geq a + bx_1} : a, b \in \mathbb{R} \right\}$$



Parameter space has dimension 2.

Also known as

2 degrees of freedom

$$\mathcal{H} = \left\{ \mathbb{1}_{X_2 > a + bX_1} \cdot a, b \in \mathbb{R} \right\}$$

~~$X_2 > a$~~

$$X_2 > a + bX_1 = \underbrace{-a}_{w_0} + \underbrace{-bX_1}_{w_1} + \underbrace{(1)X_2}_{w_2} \geq 0$$

"weights"

"bias"

$$\Rightarrow w_0 + w_1 X_1 + w_2 X_2 \geq 0$$

X_1 ← credit score, salary

$p + 1 = 3$ (# of cols in X

↳ bcz the bias term.

$$X = \begin{bmatrix} \vec{1} \\ \vdots \\ 1 \quad X \end{bmatrix} \qquad \vec{X} = [1 \; X_1 \; X_2]$$

↑ a matrix of 1's

Redefine the matrix X by appending a col of 1's on the left

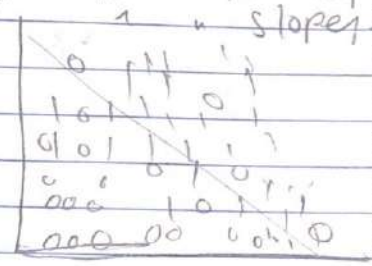
$$\mathcal{H} = \left\{ \exists \vec{w} \cdot \vec{x} \geq 0 : \vec{w} \in \mathbb{R}^3 \right\}$$

This is an "Overparameterized" model

Each line has infinite $\vec{w}$'s that speeding it.

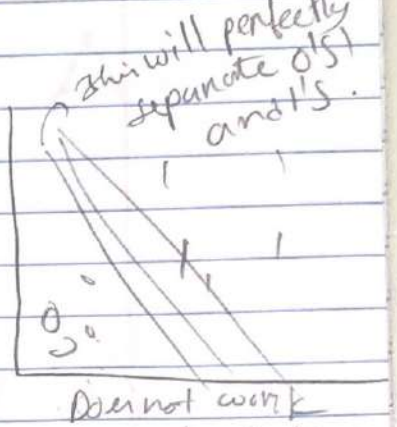
Need algorithm A

$$g = A(D, H)$$



Assume O's and i's are ~~are~~ linearly Seperable.

$\exists \vec{w}$ s.t $g(\vec{x})$ then no error.



Perception learning Algorithm (1957).

① Initialize $\vec{w}^{t=0} = \vec{0}$ or random, Compute $\hat{y}$

② let

$$w_0^{t+1} = w_0^{t=0} + \overbrace{(y_i - \hat{y}_i)}^{e_i}(1)$$

$$w_1^{t+1} = w_1^{t=0} + \underbrace{(y_i - \hat{y}_i)}_{e_i} x_{i,1}$$

$$w_2^{t=1} = w_2^{t=0} + (y_i - \hat{y}_i)\, x_{i,2}$$

$$\vdots$$

$$w_p^{t=1} = w_p^{t=0} + (y_i - \hat{y}_i)\, x_{i,p}$$

Recompute $\hat{y}$

$$X_2 \begin{bmatrix} 1, & x_{1,1} & x_{1,2}, & \cdots & x_{1,p} \\ & x_{2,1} & x_{2,2} & \cdots & x_{2,p} \\ \vdots & & & & \\ 1 & x_{n,1} & x_{n,2} & \cdots & x_{n,p} \end{bmatrix}$$

(3) Repeat 2 for $i = 1 \ldots n$

(4) Repeat steps 2, 3 until no errors

Perception is Proven to converge if linear seperability assumption is true.