02/05/2019

$$y = \underbrace{g(\vec{x})}_{\text{Model}} + \underbrace{\underbrace{h^*(\vec{x}) - g(\vec{x})}_{\text{Estimation Error}} + \underbrace{f(\vec{x}) - h^*(\vec{x})}_{\text{Misspecification Error}} + \underbrace{f(\vec{z}) - f(\vec{x})}_{\text{Ignorance Error}}}_{e \text{ (residual)}}$$

For a new observation x^*

$$\hat{y} = g(x^*)$$

In Supervise learning

$$g = \underbrace{\mathcal{A}}_{\text{algorithim.}}(\underbrace{\mathbb{D}}_{\text{historical data}}, \underbrace{\mathcal{H}}_{\text{model space}})$$

Loan Model

$y = \{0, 1\}$ — Model is called "Binary Classification Model"

0: not pay back loan; 1: pay back loan. (credit

Null Model: you have no and you want to create the best model g.

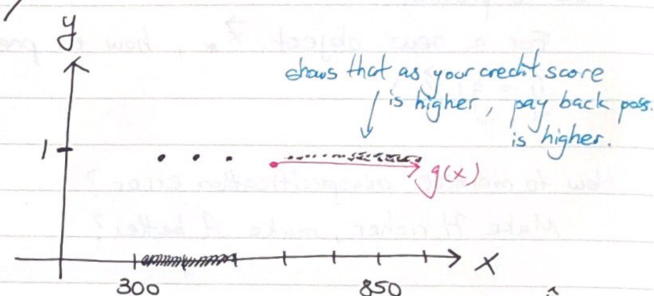
$\mathcal{H} = y$

$g = \text{Mode}(y)$

We now have one x: credit score $\mathcal{X} = [300, 850]$

$$\mathbb{D} = (X, y) = \left\langle \begin{bmatrix} 810 \\ 390 \\ 750 \\ \vdots \\ \vdots \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \\ \vdots \end{bmatrix} \right\rangle$$

$$\mathbb{D} = [X \mid y] = \begin{bmatrix} 810 & 1 \\ 390 & 0 \\ 750 & 1 \\ \vdots & \vdots \\ \vdots & \vdots \end{bmatrix}$$



$e \in \{-1, 0, 1\}$

$y = g(x) + e$ ($\hat{y} = g(x)$)

$\Rightarrow e = y - \hat{y}$

$$\mathcal{H} = \{ \mathbb{1}_{x \geq \theta} : \theta \in \Theta \}$$

θ: parameter; Θ: parameter space

where Θ = unique vectors of X.

eg. $h(x) = \mathbb{1}_{x \geq 500}$,

$h(x) = \mathbb{1}_{x \geq 497.3}$

The algorith $\mathcal{A}$ produces g, g to specified by θ.
$\mathcal{A}$ find θ then pick θ which gives the least prediction errors in $\mathbb{D}$.
(in sample error

ME (missclasification error) $= \cancel{\#/\text{error}}\ \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{g(x_i) \neq y_i}$

ACC (accuracy) $= \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{g(x_i) = y_i}$

$\mathcal{A}: \theta_g = \underset{\theta \in \Theta}{\text{Argmin}} \{ME\}$ ← objective function. fitness function.

Rewrite algorith function using e's ($\overbrace{\text{in-sample}}^{D}$ residuals)

only in binary classification models.

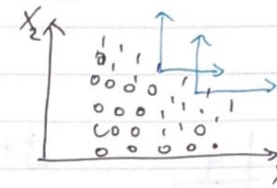
$$ME = \underbrace{\frac{1}{n}\underbrace{\sum_{i=1}^{n}|e_i|}_{\text{SAE (sum absolute error)}}}_{\text{MAE (Mean absolute error)}} = \underbrace{\frac{1}{n}\underbrace{\sum_{i=1}^{n}e_i^2}_{\text{SSE (sum squared error)}}}_{\text{MSE (Mean squared error)}}$$

X_1 : credit score

X_2 : salary

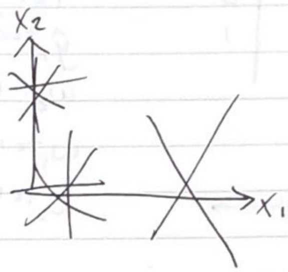
very restrictive set

$\mathcal{H} = \{ \mathbb{1}_{x_1 \geq \theta_1} \text{ and } x_2 \geq \theta_2 : \theta_1, \theta_2 \}$



Linear Threshold Models

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



$-a - bx_1 + x_2 \geq 0$

Bias/intercept

$w_0 + w_1 x_1 + w_2 x_2 \geq 0$

Inputs/ weights

let $\vec{x}_i = [1 \quad x_{i1} \quad x_{i2}]$

$$X = \begin{bmatrix} 1 & \uparrow & \uparrow \\ \vdots & \vec{x}_{1} & \vec{x}_2 \\ 1 & \downarrow & \downarrow \end{bmatrix}$$

$\vec{w} \cdot \vec{x} \geq 0$

$= \{ \mathbb{1}_{\vec{w}\cdot\vec{x} \geq 0} : \vec{w} \in \mathbb{R}^3 \}$ * $\mathbb{R} \to \mathbb{R}^3$ meaning all $c\vec{w}$ are equal solutions $c \in \mathbb{R}$

Perceptron I Learning Algorithm (1957)

let t denote iterator #.

Step 1: $\vec{w}^{t=0}_{\text{Model}} = \vec{0}$ or whatever...

Step 2: Compute $\hat{y}_i = \mathbb{1}_{\vec{w}^{t=0} \cdot \vec{x}_i}$

Step 3: For $j = 0 \ldots p$

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

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

$$\vdots$$

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

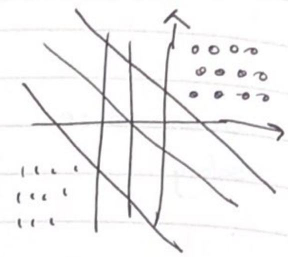
Step 4: Repeat Steps 2, 3 for $i = 1 \ldots n$.

Step 5: Repeat steps 2-4 until no change or some max iterators.

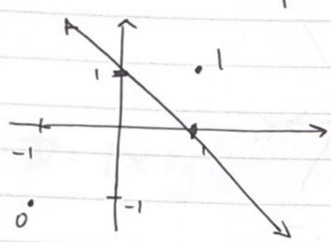
This algorithm is prove to converge if the data $\mathbb{D}$ is Linearly Seporable"
i.e. $\exists \vec{n}$ s.t $ME = 0$
If not, it will linearly fail
i.e produce a very poor model.

---

weakness #1 requires linearly sep.
#2 returns any model that separates.



i	x_1	x_2	y
1	-1	-1	0
2	1	1	1



$\vec{w}^{t=0} = \vec{0}$

$t=1, \quad i=1$

$\hat{y}_1^{t=0} = \mathbb{1}_{\vec{w}\cdot\vec{x} \geq 0} = 1$

$w_0^{t=1} = (0) + (-1)(1) = -1$

$w_1^{t=1} = (0) + (-1)(-1) = +1$

$w_2^{t=1} = (0) + (-1)(-1) = +1$

$t=1 \quad i=2 \quad \hat{y}_2 = \mathbb{1}_{\begin{bmatrix}-1\\+1\\+1\end{bmatrix}\cdot\begin{bmatrix}1\\1\\1\end{bmatrix} \geq 0} = 1$

$W_0^{t=1} = (-1) + (0)(1)$

$W_1^{t=1} = (+1) + (0)(1)$

$W_2^{t=2} = (+1) + (0)(1)$

$t=2, \quad i=1$

$\hat{y}_1 = \mathbb{1}_{\begin{bmatrix}-1\\+1\\+1\end{bmatrix}\cdot\begin{bmatrix}1\\-1\\-1\end{bmatrix} \geq 0} = 0$

Neural Network