2/7/18

Lecture 4

Supervised Learning

$\mathbb{D}, \mathcal{H}, \mathcal{A}$

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

How to get predictions/fits?

$\mathbb{D} = \{\langle \vec{x}_i, y_i \rangle\}$ $i = 1 \ldots n$

"hat" $\rightarrow \hat{y}_i = g(\vec{x}_i)$ in-sample fit

$\{\hat{y}_1, \ldots, \hat{y}_n\}$ in-sample fit/prediction

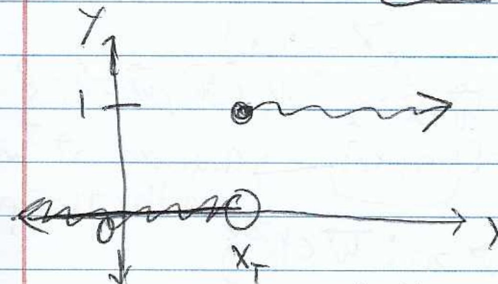
$\approx \{y_1, \ldots, y_n\}$

How to predict for new data/observation $\vec{x}^*$? $\hat{y}^* g(\vec{x}^*)$

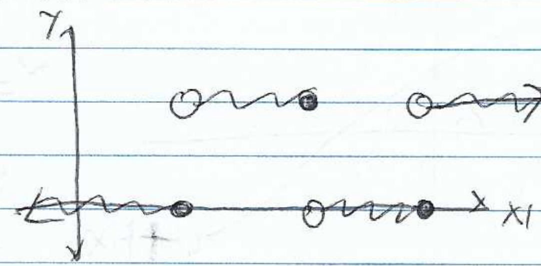
$\mathbb{1}_A := \begin{cases} 1 & \text{if } A \\ 0 & \text{if } A^c \end{cases}$ Ex: $\mathbb{1}_{x > 10} = \begin{cases} 1 & \text{if } x > 10 \\ 0 & \text{if } x \leq 10 \end{cases}$

$y \in \{0, 1\}$ Let's use only x_1 (salary, continuous)

$\mathcal{H} = \{ \mathbb{1}_{x > x_T} : x_T \in \mathbb{R} \}$ $\leftarrow$ called parameter



Threshold



more flexible version

$\text{Err}(\vec{y}, \hat{\vec{y}}) \geq 0$ for all inputs

Sum of abs. error $SAE = \sum_{i=1}^{n} |y_i - \hat{y}_i|$

Mean of abs. error $MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{y_i \neq \hat{y}_i}$
(misclassification error)

Sum of squared error $SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = SAE$

Mean of squared error $MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

A better error is the Asymmetric cost algorithm

$y_{Bob} = 1$, $\hat{y}_{Bob} = 0$

$y_{Judy} = 0$, $\hat{y}_{Judy} = 1$

$$SSE(h) := \sum_{i=1}^{n}\left(y_i - \overbrace{h(\vec{x}_i)}^{\hat{y}_i}\right)^2$$

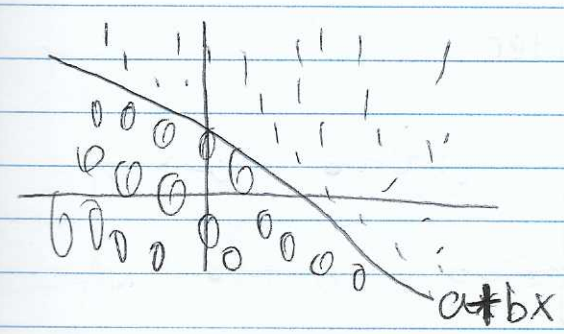
$$g := \underset{h \in \mathcal{H}}{\text{argmin}}\left\{SSE(h)\right\} \iff x_T^* = \underset{x_T}{\text{argmin}}\left\{\sum_{i=1}^{n}\left(y_i - \mathbb{1}_{\vec{x}_i > x_T}\right)^2\right\}$$

A would be a "greedy search" - try everything possible

X_1, X_2 both continuous

$$\mathcal{H} = \left\{\mathbb{1}_{x_2 > a + bx_1} : \begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^2\right\} = \left\{\mathbb{1}_{a + bx_1 - x_2 < 0} : \begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^2\right\}$$

$$= \left\{\mathbb{1}_{-a - bx_1 + x_2 > 0} : \begin{bmatrix} a \\ b \end{bmatrix} \in \mathbb{R}^2\right\} = \left\{\mathbb{1}_{w_0 + w_1 x_1 + w_2 x_2 > 0} : \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix} \in \mathbb{R}^3\right\}$$



(↖ linear classifier)

$\vec{x} = [x_1, x_2]$

$$= \left\{\mathbb{1}_{w_0 + \vec{w} \cdot \vec{x} > 0} : w_0 \in \mathbb{R}, \vec{w} \in \mathbb{R}^2\right\}$$

let $\vec{x} = [1, x_1, x_2]$ → augment $\vec{x}$ with a 1 in the first position

$$= \left\{\mathbb{1}_{\vec{w} \cdot \vec{x} > 0} : \vec{w} \in \mathbb{R}^3\right\}$$

Use same error function, SSE $\quad g = \text{argmin}\{SSE(h)\}$

$$\Updownarrow$$

$$w^* = \text{argmin}\left\{\sum_{i=1}^{n}\left(\overset{y_i -}{\mathbb{1}_{\vec{w} \cdot \vec{x}_i > 0}}\right)^2\right\}$$

$$\vec{w}^* = \text{argmin}\left\{\sum_{i=1}^{n}\left(y_i - \mathbb{1}_{\vec{w} \cdot \vec{x}_i > 0}\right)^2\right\}$$

HARD

Algorithm: Perceptron (1957)