MATH 390.4
Lecture 3 ; 2/4/2020

supervised learning (3 ingredients)

$y = t(z)$ $F(\vec{x}) + \int$

$\int$: error due to ignorance

1) training data

$\mathbb{D} = \langle X, Y \rangle$

$X = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$, $Y = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$, $\vec{x_i} \in \mathcal{X}$, $y_i \in \mathcal{Y}$

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

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

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

Is $F \in \mathcal{H}$ generally? NO

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

$y = t(z) = F(x) + \int = h^*(x) + \varepsilon$

$h^*(\vec{x}) + (F(\vec{x}) - h^*(\vec{x})) + (t(\vec{z}) - F(\vec{x}))$

($F(\vec{x}) - h^*(\vec{x})$: misspecification error (model too simple for a complex scenario); $(t(\vec{z}) - F(\vec{x}))$: $\int$ "epsilon error")

ε epsilon error

$y = g(x) + (h^*(x) - g(x)) + ((F(x) - h(x)) + (t(z) - F(x)))$

model; $(h^*(x) - g(x))$: estimation error; $\int$; ε; e residual

* just because $h^* \in \mathcal{H}$ does NOT mean A will locate it. A will not be perfect and the values of ε will confuse A. Then $g \neq h^*$; g is the best A can do

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

$e = y - \hat{y}$ residual if $\vec{x} \in \mathbb{D}$ otherwise they're unknown

How to reduce errors:

1) $\int$, ignorance error can be reduced by: measuring more x_j's (features) of the units that contain information

2) Misspecification error: expand set $\mathcal{H}$ to make more complicated functions } * machine learning

3) Estimation error can be reduced by: increase sample size

example of supervised learning:

(1) $y = \{0, 1\}$ paid back mortgage

# of variables $p = 1$ x_1: credit score

$\mathbb{D} = \langle X, Y \rangle = \left\langle \begin{bmatrix} 810 \\ 570 \\ 750 \\ \vdots \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ \vdots \end{bmatrix} \right\rangle$

$\mathcal{X} = [300, 850]$

$n = 100$

Graph: all same ME (plus 2 points)

(2) $\mathcal{H} = \{ \mathbb{1}_{x \geq \theta} : \theta \in \Theta \}$ (Θ: capital theta, parameter (also called threshold model); $\mathbb{1}$: indicator function)

$\mathbb{1}_A(\omega) = \begin{cases} 1 \text{ IF } \omega \in A \\ 0 \text{ IF } \omega \notin A \end{cases}$ ($B, \omega, \ldots$)

eg: $g(x) = \mathbb{1}_{x \geq 515.3}$

3) Algorithm A

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

A estimates θ (parameter)

"misclassification error"

$ME := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{g(x_i) \neq y_i} = \frac{1}{n} \sum_{i=1}^{n} |e_i|$ (supervised decision error (SAE); mean absolute error (MAE))

accuracy: $1 - ME$

A: minimize ME over $\theta \in \Theta$ $\theta \in$ unique x's

$= \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| = \frac{1}{n} \sum e_i^2$ (sum squared error; mean squared error)

ex 2)

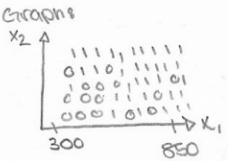
$y \in Y \{0,1\}$

x_1: credit score $\in [300, 850]$

x_2: salary $\in \mathbb{R}$ ($1000s)

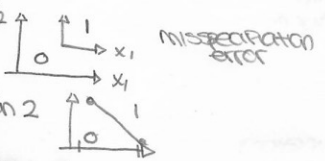
$D = \left\langle \begin{bmatrix} 810 & 63.1 \\ 370 & 58.7 \\ 750 & 137.6 \\ \vdots & \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \right\rangle$

Graphing:



$H = \{ \mathbb{1}_{x_1 \geq \theta_1 \,\&\, x_2 \geq \theta_2} : [\theta_1, \theta_2] \in H \}$

$F(x) = \mathbb{1}_{x_1 \geq 650 \,\&\, x_2 \geq 100}$



misspecification error

$H = \{ \mathbb{1}_{x_2 \geq a + b x_1} : a, b \in \mathbb{R} \}$

parameter space has dimension 2 (degrees of freedom)

$x_2 \geq a + b x_1 \Rightarrow \underbrace{-a}_{w_0} + \underbrace{-b x_1}_{w_1} + \underbrace{(1) x_2}_{w_2} \geq 0$

bias — weights

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

$X = [\vec{1} \; X]$ redefine the matrix by appending a column of 1's on the left

$\vec{x} = [1 \; x_1 \; x_2]$

$H = \{ \mathbb{1}_{\vec{w} \cdot \vec{x} \geq 0} : \vec{w} \in \mathbb{R}^3 \}$

this is overparameterized model each line has infinite $\vec{w}$s that specify it.

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

default score/salary — for the bias term

need algorithm A:

$g = A(D, H)$

assume 0's and 1's are linearly separable

Perception Learning Algorithm (1957)

1) initialize $\vec{w}^{t=0} = \vec{0}$ or random (compute $\hat{y}$)

2) For $j = 0, 1, 2, \ldots, p$ let: (e_i)

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

$w_1^{t=1} = w_1^{t=0} + (y_i - \hat{y}_i) x_{i1}$

$w_2^{t=1} = w_2^{t=0} + (y_i - \hat{y}_i) x_{i2}$

$\vdots$

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

recompute $\hat{y}$

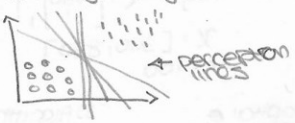
3) Repeat step 2 for $i = 1 \to n$

4) Repeat steps 2 & 3 until no errors

* Perception is proven to converge if the linear separability assumption is true

$X = \begin{bmatrix} 1 & x_{11} & x_{12} & \ldots & x_{1p} \\ 1 & x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & & & & \\ 1 & x_{n1} & x_{n2} & \ldots & x_{np} \end{bmatrix}$

* update weights until right weight



perception lines