Math 390.4 Lec 4 3/7/18

Supervised learning: $\mathbb{D}, \mathcal{H}, \mathcal{A}$ then $g = \mathcal{A}(\mathbb{D}, \mathcal{H})$

How to see a prediction for $\vec{x} \in \mathbb{D}$? must have same features as $\vec{x}_i$'s in $\mathbb{D}$, otherwise it is not in the domain of g!!

$\hat{y}_i := g(\vec{x}_i) \quad i \in \{1, \dots, n\}$

Then the next question is mathematically obvious but conceptually quite a leap!!

How to predict for new person? $\vec{x}^*$ ... let $\hat{y}^* = g(\vec{x}^*)$. Hopefully $\hat{y}^* \approx y^*$ most of the time (real, unobserved as of now)

Back to our problem...

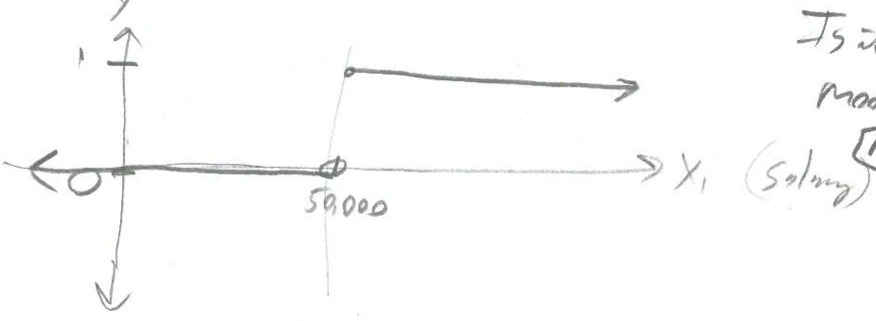
$y \in \{0, 1\}$, $\vec{x} \in \mathcal{X}$ where $p = 3$. To make it even simpler, let's say we only have x_1, salary.

let $\mathcal{H} = \{ \mathbb{1}_{x \geq x_T} \}$

Indicator function is $\mathbb{1}_{x \geq x_T} = \begin{cases} 1 & \text{if } x \geq x_T \\ 0 & \text{if not} \end{cases}$

What would a g look like?

Threshold Model



Is it a good model?? NO

Parameter notation: θ, β, w, any other symbols used (unfortunately)

What do all possible choices of $\mathcal{H}$ look like? Elements in $\mathcal{H}$ are indexed by x_T. Every single value of x_T, the threshold creates a new model. x_T is called a "parameter".

What does A do? It selects the best candidate $g \in \mathcal{H}$.
That is, it selects the "best" x^*. How??

Consider the following... there are n x_i's. Try $x_T = x_1$, $x_T = x_2, \dots, x_T = x_n$,
pick the "best". How to know the best?

Need to determine a way of measuring error. This will be a bit diff when we get to other models

$\hat{y}_i := h(\vec{x}_i)$

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

sum of squared error

i.e. the # of times the hypothesis function disagrees with the response (want it to be small)

$$SAE = \sum_{i=1}^{n} |\hat{y}_i - y_i| = \sum \mathbb{1}_{\hat{y} \neq y_i}$$

# of misclassifications

$$g = \underset{h \in H}{\text{argmin}} \{SSE(h)\} \iff x$$

MSE, MAE, misclassification rate, Any monotonic increasing function

But we never used x_2, x_3 ... wouldn't we do better? Yes... SSE always gets smaller as you increase p.

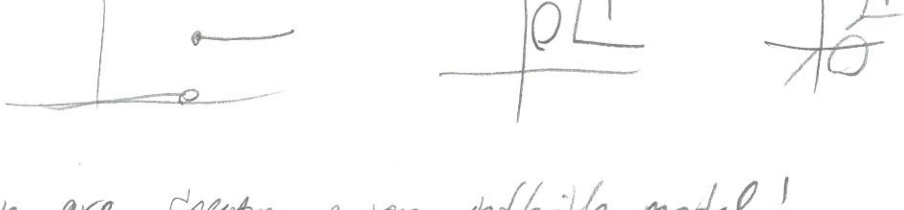
Why don't we try a more complicated model?

How about x_1, x_2, x_3.

we can do a model like

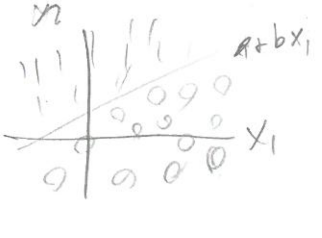
$$H := \{ \mathbb{1}_{x_1 \geq x_1^*} \mathbb{1}_{x_2 \geq x_2^*} \mathbb{1}_{x_3 \geq x_3^*} : \langle x_1^*, x_2^*, x_3^* \rangle \in \mathcal{X} \}$$

Why would this be bad?



You are creating a very inflexible model!

Why not a linear model? Imagine x_1, x_2 (both continuous)



$$h(x_1, x_2) = \mathbb{1}_{x_2 > a + bx_1} = \mathbb{1}_{a + bx_1 - x_2 < 0}$$
$$= \mathbb{1}_{-a - bx_1 + x_2 > 0} = \mathbb{1}_{w_0 + w_1 x_1 + w_2 x_2 > 0}$$

$$\mathcal{H} := \{ \mathbb{1}_{w_0 + w_1 x_1 + w_2 x_2 > 0} : w_0, w_1, w_2 \in \mathbb{R} \}$$

$$= \{ \mathbb{1}_{w_0 + \vec{w} \cdot \vec{x} > 0} : w_0 \in \mathbb{R}, \vec{w} \in \mathbb{R}^2 \}$$ where $\vec{x} = [x_1\ x_2]$, $\vec{w} = [w_1\ w_2]$

if we augment $\vec{x}$ to include an $x_0 = 1 \Rightarrow \vec{x} = [1\ x_1\ x_2]$

" " " $\vec{w}$ " " " " " " " " $w_0 \Rightarrow \vec{w} = [w_0\ w_1\ w_2]$

$$= \{ \mathbb{1}_{\vec{w} \cdot \vec{x} > 0} : \vec{w} \in \mathbb{R}^3 \}$$ Note that $\overset{2}{\overset{\shortparallel}{p}} + 1 = 3$

Here, a 1 is prepended to the feature vector so the intercept w_0 can be part of the dot product. This is purely a convenience for linear models. This is sometimes implied without explicitly telling you!

What is $A(\mathcal{H}, \mathbb{D})$? How to pick $g \in \mathcal{H}$?

$$g = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \{ SSE(h) \} = \mathbb{1}_{\vec{w}^* \cdot \vec{x} > 0}$$

Since $\mathcal{H}$ is parameterized by $\vec{w}$, this is the same problem as: $$\vec{w}^* = \underset{\vec{w} \in \mathbb{R}^3}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} \mathbb{1}_{\vec{w} \cdot \vec{x}_i > 0} \neq y_i \right\}$$

- You can't check all $\vec{w} \in \mathbb{R}^3$ like $x^* \in \{x_1, \ldots, x_n\}$ like last time!
- You can't take the derivative wrt to $\vec{w}$ since the # errors isn't continuous...
- 1957 "Perceptron Learning Algorithm", an iterative, imperfect algorithm