2/3/21

Let's pretend there are three causal drivers:
z_1: has sufficient funds to pay back loan at the time it's due $z_1 \in \{0,1\}$

z_2: unforeseen emergency?
$z_2 \in \{0,1\}$

z_3: Criminal intent?
$z_3 \in \{0,1\}$

$y = f(z_1, z_2, z_3) = z_1(1-z_2)(1-z_3)$

problem in practice?
1. you don't know the z's because they are realized in the future
2. you may not know the function f which can be very complicated

What is the next best thing since you have to make a decision now and you need a model that works now?

You obtain information that approximates the info in the z's and combine this info to approximate y. We denote these proxies that do this approximation the x's and we denote p to be the number of such proxies: $x_1 .. x_2 ... x_p$. For example

x_1: salary at the time of loan application $\in \mathbb{R}$
x_2: missing payment previously $\in \{0,1\}$
x_3: Criminal charge in the past $\in \{0,1\}$
$\Rightarrow p = 3$

x_j's are called feature, characteristics, attributes, variables, independent variables, regressors, covariates

What is normally done in the real world? You use the features that are available

To learn from data, you measure the x_j's on subjects $i = 1..n$

Let $\vec{x}_i := [x_{i1}, x_{i2}, \ldots, x_{i,p}] \in \mathcal{X}$, the input space

Subjects are also called observation, setting, records, objects inputs.

$x_2 \in \{0, 1\}$ binary variable
$x_1 \in \mathbb{R}$ Continuous variable
x_3 is a binary variable
} types / names of variables

Let's consider measuring x_3 differently:

$x_3 \in \{$ none, infraction, misdemeanor, felony $\}$
(this is an ordered Categorical variable)

How do we make this a metric

(1) Code it in order of severity spacing by 1:
$x_3 \in \{0, 1, 2, 3\}$

Downside: Coding is arbitrary

2. Binarize / dummify this categorical variable:

$x_{3a} \in \{0,1\}$ infraction or not?
$x_{3b} \in \{0,1\}$ misdemeanor or not?
$x_{3c} \in \{0,1\}$ felony or not?

One variable became 3 variables $\Rightarrow p = 5$
We had 4 levels ($L=4$) but now we made $L-1=3$ variables. Why?
You can capture the last category (called the reference category) by setting all "dummies" / binary variable to zero.

If the variable is "nominal categorical" meaning no inherent order, you must do #2 to be able to use it in a model eg

$x \in \{$ red, blue, green, yellow, purple, brown, ... $\}$

Can we say that $y = f(x_1, x_2, \ldots x_p)$?
no! "It is only approximation at best" - Gabriel

$y = t(z_1, \ldots, z_t)$ where you don't know t or the z's

$y \simeq f(x_1, \ldots, x_p)$

or $y = f(x_1, \ldots, x_p) + \delta$, st $\delta = t - f$

What is delta? It's an error, it's error due to... ignorance
"Ignorance of the true causal drivers. It's the error due to the best that the proxies aren't the real thing
You're missing information

How do we decrease delta? Increase p with more useful variables.

How do we get f? Note that there is no "analytical solution" The approach we are in "learning from data". This is an "empirical approach". There are many flavors. We will concentrate on "supervised learning from "historical data", This requires three ingredients!

(1) Training Data

$D = \{\langle \vec{x}_1, y_1 \rangle, \langle \vec{x}_2, y_2 \rangle \ldots \langle \vec{x}_n, y_n \rangle\}$

these are n historical examples of inputs / outputs

Alternate notation:

$D = \langle X, \vec{y} \rangle$ where $X = \begin{bmatrix} \leftarrow \vec{x}_1 \rightarrow \\ \leftarrow \vec{x}_2 \rightarrow \\ \vdots \\ \leftarrow \vec{x}_n \rightarrow \end{bmatrix}$, $\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$

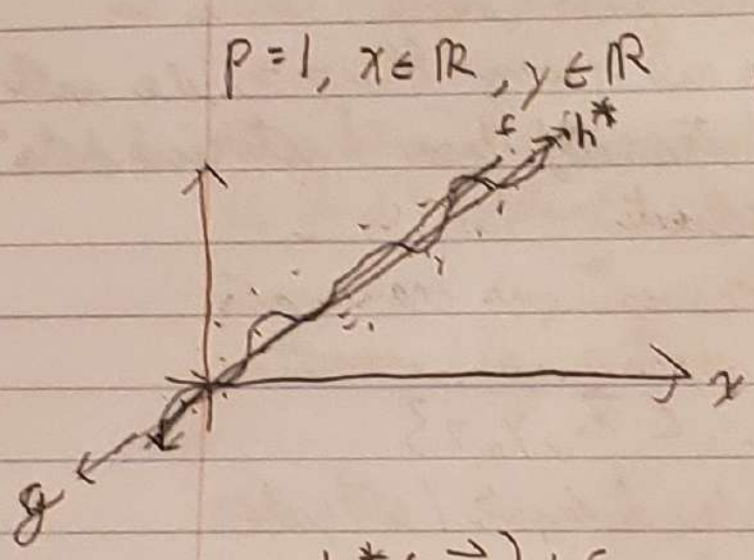
2. H := a set of candidate function with elements h that approximate f. We need this because the space of all function is too large and too ill-defined to directly find the "best one". You need to limit this space!

3. We need A := the algorithm that takes in D, H and returns g, an approximation to f, $g = A(D, H)$.

Is it true that $f \in H$? No, f is arbitrarily complicated and unknown and the set curly-H contains usually simple

function that can be fit with curly-A

However, there is a $h^* \in \mathcal{H}$ which is the candidate model that most closely approximates f. Here is an example:



$f(x) = x + 0.1 \sin(x)$

$\mathcal{H} = \{\text{all linear models}\}$
$= \{b_0 + b_1 x : b_0 \in \mathbb{R}, b_1 \in \mathbb{R}\}$

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

$y = h^*(\vec{x}) + \varepsilon$
$= h^*(\vec{x}) + \underbrace{(f(\vec{x}) - h^*(\vec{x}))}_{\text{model misspecification (ignorance error)}} + \underbrace{(f(\vec{x}) - f(\vec{x}))}_{\text{error}}$

ε

model residual (the "full error": the difference predicted and observed)

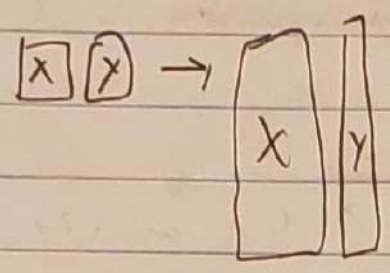
$y = g(\vec{x}) + e$
$= g(\vec{x}) + \underbrace{h^*(\vec{x}) - g(\vec{x})}_{\text{estimation error}} + f(\vec{x}) - h^*(\vec{x}) + \underbrace{f(\vec{x}) - f(\vec{x})}_{\delta}$

ε

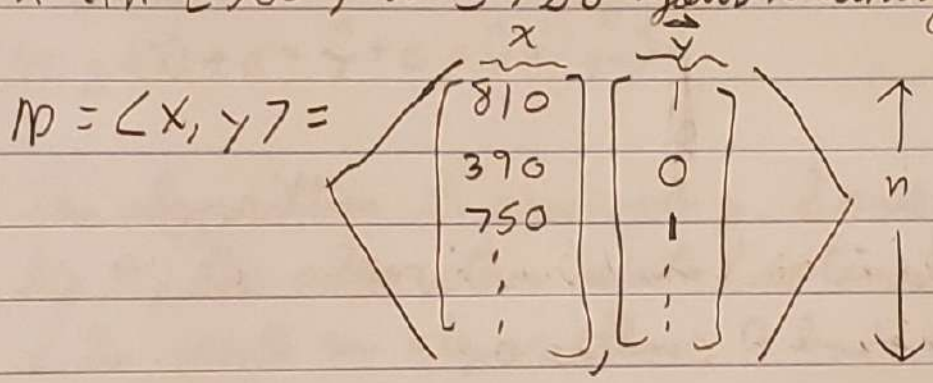
e

How do we decrease model misspecification error?
Expand the set of candidate functions $\mathcal{H}$ to be more complicated and thus more expensive & complex relationship

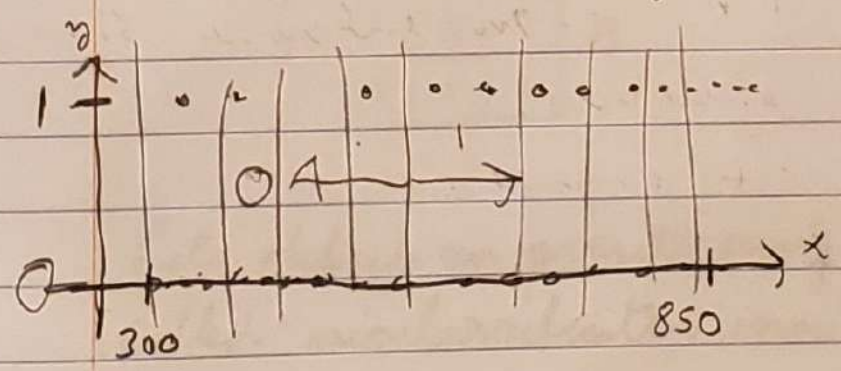
How do we decrease estimation error?
Increase sample size n (more historical examples). The rows in D



Back to the loan example where $y \in \{0, 1\}$
Let's say we have $p=1$ feature, the credit score:
$x \in [300, 850]$. So your training data looks like:



Let's plot the training data.



What is the "null model" g_0 which is the model if you didn't have any x's whatsoever?

$g_0 = \text{mode}[\vec{y}]$ → $\mathcal{H} = \{\mathbb{1}_{x \geq \theta} : \theta \in x\}$ eg $g(x) = \mathbb{1}_{x > 600}$

What is the simplest possible candidate space $\mathcal{H}$?