Assumption: $y = t(z_1, \ldots z_t)$ (not model, reality)
Assumes world is deterministic, but we can't know z_i's or t

Next best thing: Obtain $x_1, \ldots, x_p$ which hopefully capture much of the info in the z_i's

then $\vec{x}_i := [x_{i1}, x_{i2}, \ldots, x_{ip}] \in \mathcal{X}$ "input space" / variable space

observation, setting, input, record, object, subject
regressors, variable, covariate, feature, attribute, measurement

recall x_1: credit score $\in \mathbb{R}$

Continuous variable

x_2: Criminality
many metrics
$x_2 \in \{\text{has history}, \text{has no history}\}$
↓ 1 ↓ 0

Dummy / Indicator / Binary variable

or
$x_2 \in \{\text{none}, \text{infraction}, \text{misdemeanor}, \text{felony}\}$

factor / categorical variable
w $L = 4$ levels

$p :=$ number of regressors

Two strategies to factor variables into mathematical models
a) ordinal encoding
$x_2 \in \{0, 1, 2, 3\}$ "ordinal factor variable"
major downside: arbitrarily scaled

b) nominal encoding
$x_{2a} \in \{0,1\}$ $x_{2b} \in \{0,1\}$ $x_{2c} \in \{0,1\}$ ...

downside: $p=3 \rightarrow p=5$

Can you say $y = f(x_1, \ldots, x_p)$?

no, y is $f(x_1, \ldots, x_p) + \delta$ → $\delta = t(\vec{x}) - f(\vec{x})$ Error due to Ignorance

How to minimize δ:
increase # of relevant variables

Find f. The approach we use is called <u>learning from data</u>, an <u>empirical approach</u>. → based on measured data

The type of <u>learning from data</u> we will employ is <u>supervised learning</u>

← historical data oversees the learning

<u>Supervised learning in 3 steps</u>

1) "Training data", "Historical data"

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

n: # of historical examples (sample size)

each $\langle \vec{x}_i, y_i \rangle$ is an input/output pair

$$X := \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \\ \vdots \\ \vec{x}_n \end{bmatrix}, \quad \vec{y} \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad \mathbb{D} = [X, \vec{y}]$$

$\vec{y} \in y^n$ (output space)

Assumption → $\dim = n \times p$

2) $\mathcal{H}$: a set of candidate functions h that can approximate f

3) $\mathcal{A}$: an algorithm that takes $\mathcal{H}$ & $\mathbb{D}$ & provides $g \in \mathcal{H}$ as the best approximation of f. $g = \mathcal{A}(\mathbb{D}, \mathcal{H})$

Is $f \in \mathcal{H}$? Generally speaking, no

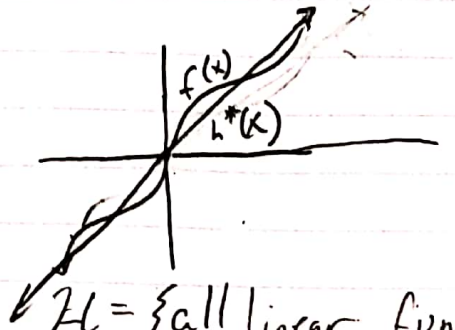
However $\exists\ h^* \in \mathcal{H}$ that is the best approximation of f

$$y = h^*(x_1, \ldots, x_p) + \underbrace{f(\vec{x}) - h^*(\vec{x})}_{\text{misspecification error}} + \underbrace{t(\vec{z}) - f(\vec{x})}_{\text{ignorance error}}$$

↘ machine learning minimizes

↘ ↑ sample size minimizes

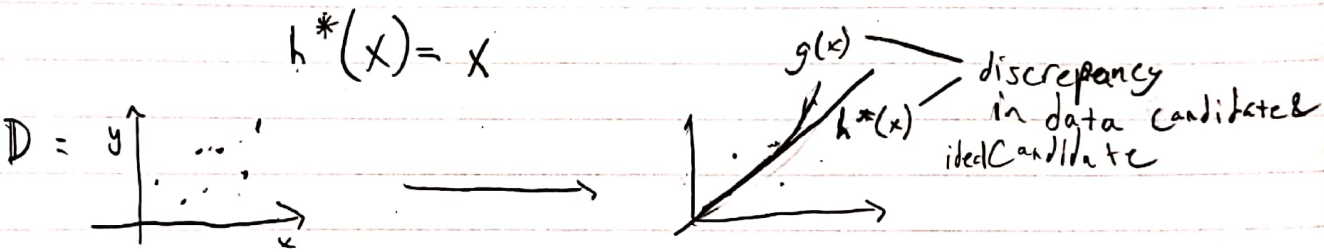
let $f(x) = x + 0.1 \sin(x)$



$\mathcal{H} = \{\text{all linear functions of } x\}$

$= \{\beta_0 + \beta_1 x : \beta_0 \in \mathbb{R}, \beta_1 \in \mathbb{R}\}$

$h^*(x) = x$



A tries to predict h^*

I want $g(x) = b_0 + b_1 x$

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

$$\underbrace{\qquad\qquad\qquad\qquad}_{e\ (\text{residual})}$$

how to predict from a new $\vec{x}$ to y?

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

errors appear random ble they are confounding

$$y = h^*(\vec{x}) + \overbrace{f(\vec{x}) - h^*(\vec{x}) + t(\vec{z}) - f(\vec{x})}^{\mathcal{E}\,(\text{noise})}$$

How to minimize misspecification error?

$\mathcal{H}$ richer, A better (?)

How to minimize estimation error?

Increase n (sample size)