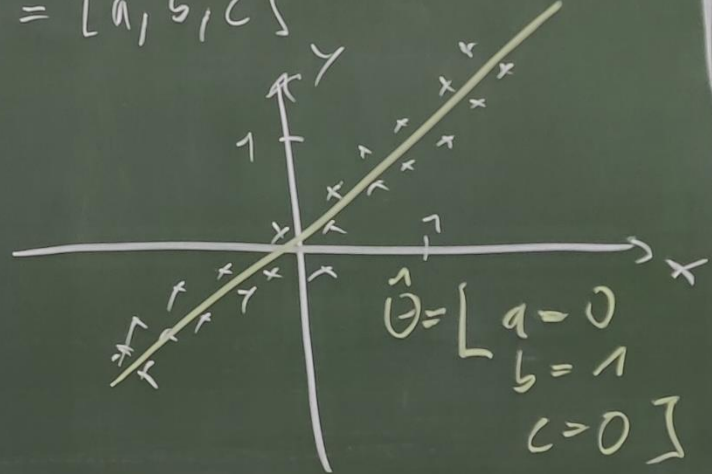
What is ML?

- setting: - quantities Y we would like to know, but cannot (easily) measure
  response
  features
  - quantities X we can measure and are related to Y
- idea: learn function $\hat{Y} = f(X)$ such that $\hat{Y}_i \approx Y_i^*$ ← truth
- traditional science: ask expert to design $f(X)$
- machine learning: - choose a "universal function family" $\mathcal{F}$
  - $\mathcal{F}$ has parameters θ ⇒ find parameter $\hat{\theta}$ that fit data

$$\hat{Y} = f_{\hat{\theta}}(X) \quad \text{s.t.} \quad \hat{Y}_i \approx Y_i^*$$

- example: $\mathcal{F} = \{ a x^2 + b x + c \}$ $\quad \hat{\theta} = [a, b, c]$

$\hat{\theta} = [\, a = 1, \ b = 0, \ c = 0.3 \,]$

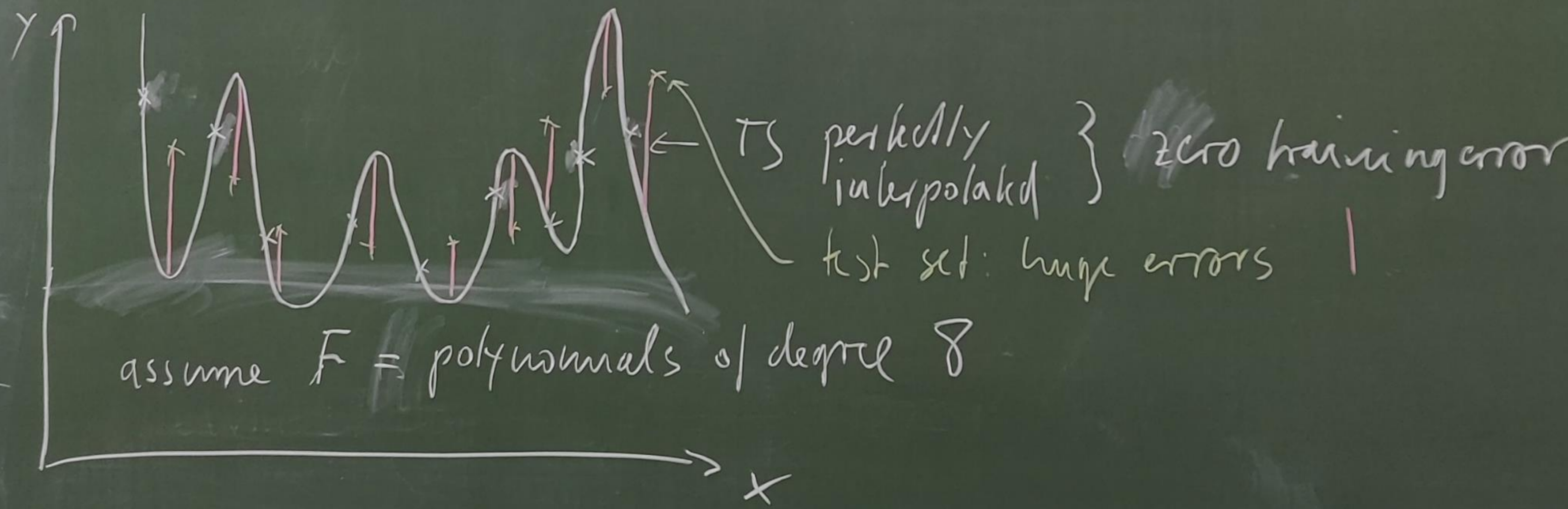


$\hat{\theta} = [\, a = 0, \ b = 1, \ c = 0 \,]$

Basic ML workflow

1. collect data: two datasets: – training dataset (TS) to find $\hat{\theta}$
   – test dataset to validate quality
2. select function family F (prior knowledge, experience, trial-and-error
   upcoming: tools)
3. find best $\hat{\theta}$ by fitting F to <u>training set</u>
4. validate quality of $f_{\hat{\theta}}(x)$ on <u>test set</u>
5. deploy model in practice

separate training & test sets necessary to recognize <u>Overfitting</u>



assume F = polynomials of degree 8

probabilistic prediction:

- reality is often more complicated: no unique true response Y^*
  instead: set of possible values $\{Y_1^*, Y_2^*, \dots\}$
  e.g. $Y^* \in [1, 2.3]$

$\Rightarrow$ learn a conditional probability (instead of function)

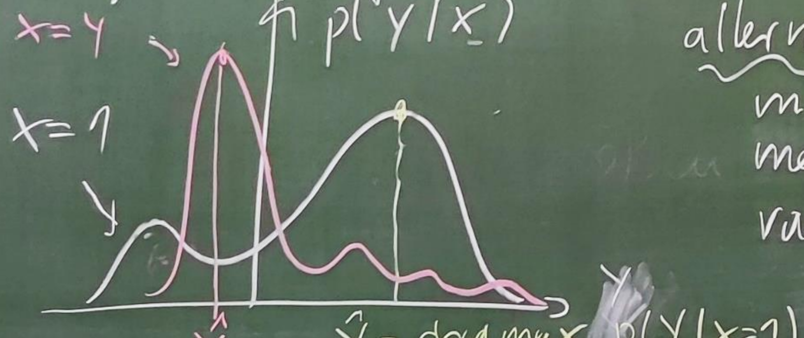
$$\hat{Y} \sim p(Y \mid X)$$

- define "universal probality family" with parameters θ
  choose $\hat{\theta}$ such that $Y \sim p_{\hat{\theta}}(Y \mid X)$ matches the data

- strict generalization: deterministic case is recovered by defining

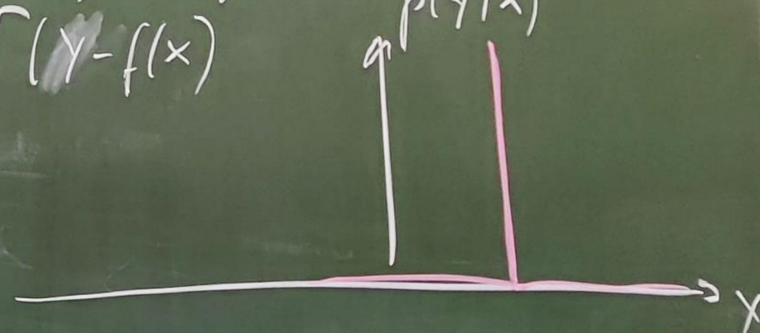
$$p_\theta(Y \mid X) = \delta(Y - f_\theta(x))$$

- often, we derive deterministic results from probabilistic predictions
  e.g. mode of distribution $\hat{=}$ value with highest probability



alternatives:
mean
median
random

$\delta(Y - f(x)$



why is probabilistic prediction important?

- reality is full of uncertainty, on 4 levels
  - fundamental: nature is not fully predictable
    - quantum physics • deterministic chaos • combinatorial explosion
  - data: finite, noisy, incomplete, ambiguous
  - model: solutions $f_{\hat{\theta}}$, $p_{\hat{\theta}}$ are imperfect (function family too small) may not have converged perfectly, nature has changed in the meantime)
  - goal: disagreement about what is relevant / desirable?

$\boxed{\text{handling uncertainty is central challenge of AI and ML}}$

Notation:

features: X_i row vector for each instance i ($i = 1 .. N$), features j ($j = 1, .., D$)

(N: number of instances; D: number of features)

feature matrix X rows instances, columns features; $X \in \mathbb{R}^{N \times D}$

i \ j	1 (name)	2 (height)	3 (gender)
1	Alice	1.7 m	f
2	Bob	1.8 m	m
N = 3	Max	1.9 m	m

in Python all float, discrete labels => one-hot encoding

i \ j	1 (height)	2 (f)	3 (m)	4 (d)
1	1.7	1	0	0
2	1.8	0	1	0
3	1.9	0	1	0

← elements X_{ij}

response: Y_i row vector for instance i, response elements m ($m = 1, \ldots, M$)

mostly: $M = 1$, scalar response (exception: one-hot encoding of discrete labels)

tasks according to type of response

(1) $Y_i \in \mathbb{R}^M$ $\quad Y \approx f_\theta(X)$ "regression"

training set: $TS = \{(X_i, Y_i)\}_{i=1}^{N}$

"supervised learning" $\{$ known response for instance i

(2) Y_i discrete label $Y_i = k,\ k \in \{1, \ldots, C\}$ ← number of categories

"classification"

(2a) labels ordered "ordinal classification"

"tiny" < "small" < "medium" < "large" < "huge"

(2b) labels unordered "categorical classification"

"apples", "oranges", "ananas"

types of learning approaches

(1) supervised learning: true responses are known in $TS = \{(X_i, Y_i)\}_{i=1}^N$

(2) unsupervised learning: only features are know in $TS = \{X_i\}_{i=1}^N$

$\Rightarrow$ learning alg. needs to find structure in the data on its own "data mining", "research"

few solutions that guaranteed to work:
- representation learning: compute better features $\tilde{X} = \varphi(X)$, ex. dimension reduction $\dim(\tilde{X}) < \dim(X)$
- clustering: group similar instances into "clusters"

(3) weakly supervised: in between

(3a) some information about Y_i (ex. know Y_i for some instances, coarse labels: "there is a tumor in CT, but don't know where")

(3b) hot: self-supervised: define auxiliary tasks where Y_i are easy to determine, ex. Large language models: drop words from text