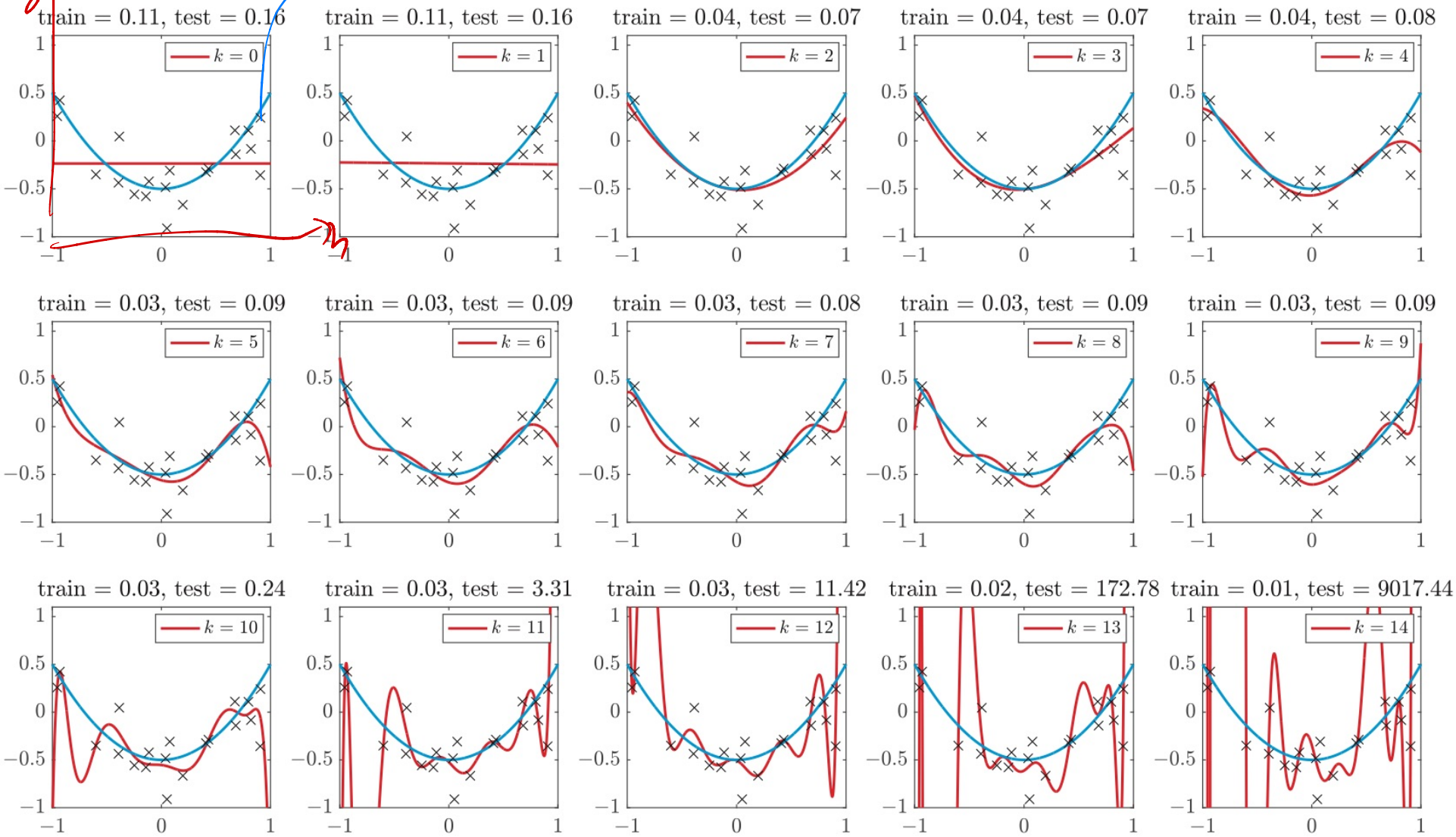
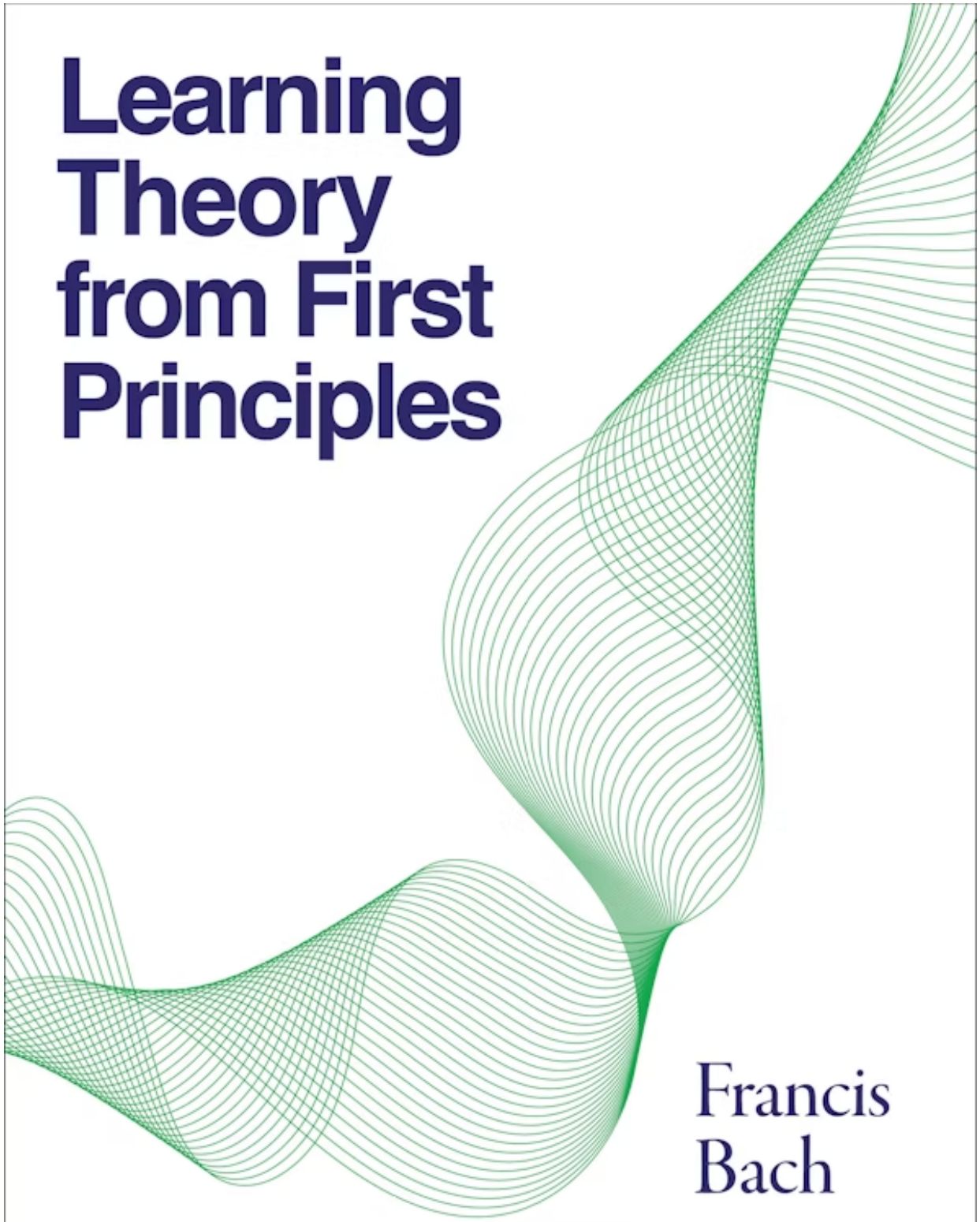

train = 0.11, test = 0.16
k = 0
train = 0.11, test = 0.16
k = 1
train = 0.04, test = 0.07
k = 2
train = 0.04, test = 0.07
k = 3
train = 0.04, test = 0.08
k = 4
train = 0.03, test = 0.09
k = 5
train = 0.03, test = 0.09
k = 6
train = 0.03, test = 0.08
k = 7
train = 0.03, test = 0.09
k = 8
train = 0.03, test = 0.09
k = 9
train = 0.03, test = 0.24
k = 10
train = 0.03, test = 3.31
k = 11
train = 0.03, test = 11.42
k = 12
train = 0.02, test = 172.78
k = 13
train = 0.01, test = 9017.44
k = 14

# Learning Theory from First Principles



Francis Bach

# Supervised learning

Data $(x_i, y_i)$, $i = 1, \dots, n$ — large

Training

$x_i$: inputs, covariates, features

$y_i$: outputs, labels, responses

$x \in \mathcal{X}$

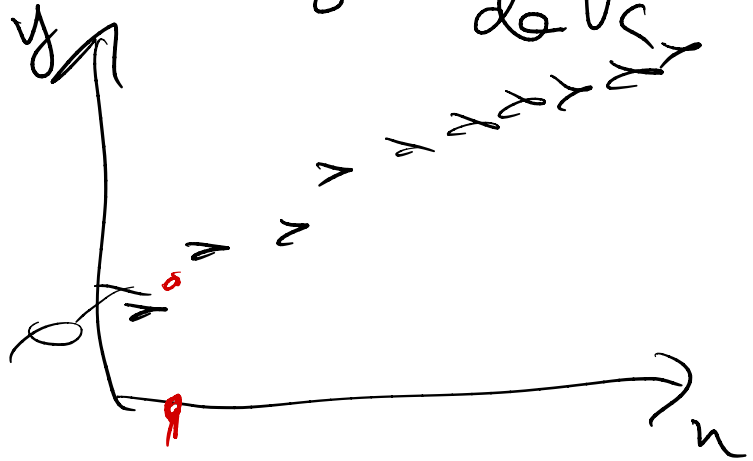
$y \in \mathcal{Y}$

$\mathcal{X} \subseteq \mathbb{R}^d$

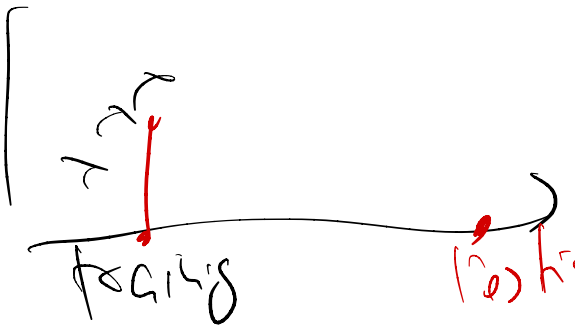
$\varphi : \mathcal{X} \to \mathbb{R}^d$ — large

feature vector

- $y = f(x) + \text{"noise"}$
- scarcity trap




Test performance: $x \in \mathcal{X}$



# Probabilistic formulation

$(x_i, y_i)$ IID $i = 1, \dots, n$

independent and identically distributed from distribution $p(x,y)$ = test distribution

Remark: dependence through Markov chain, domain adaptation

# Practical performance evaluation

test data $(x_i^{test}, y_i^{test})_{i=1,\dots,n_{test}}$ — $\sqrt{\log(\#\text{models})}$ / $n$

$$\left| \mathbb{E}\, \ell(y, f(x)) - \frac{1}{n} \sum_{i=1}^{n} \ell(y_i^{test}, f(x_i^{test})) \right| = O\left(\frac{1}{\sqrt{n}}\right)$$


training
validation
test

Decision theory: learning when data distribution is known

distribution $p(x,y)$ on $X \times Y$

$$\mathbb{E}(f(x)) = \int_X f(x)\, dp(x)$$

$$\mathbb{E}[g(x,y)] = \int_{X\times Y} g(x,y)\, dp(x,y)$$

Loss function: $\ell : Y \times Y \to \mathbb{R}$

$\ell(y,z)$ = loss of predicting $z$ while the observed output is $y$

- Binary classification: 0-1 loss $\ell(y,z) = 1_{y \neq z}$

  $Y = \{0,1\}$, $\{-1,1\}$, $\{1,2\}$

- Multicategory: $Y = \{1, \dots, k\}$

- Regression: $Y = \mathbb{R}$ $\quad \ell(y,z) = (y-z)^2$ or $\frac{1}{2}(y-z)^2$

  $\log\left(1 - e^{-y f(x)}\right)$

Expected risk : $f: X \to Y$

generalization perf

test error

$$R(f) = \mathbb{E}\left[ e(y, f(x)) \right]$$

testing data

testing error

"functional"

two sources of randomness $\begin{cases} \text{train data} \\ \text{testing data} \end{cases}$

Empirical risk

training error

$$\hat{R}(f) = \frac{1}{n} \sum_{i=1}^{n} e(y_i, f(x_i))$$

$\leftarrow$ training data.

Examples : classification $f: X \to Y$ finite with 0-1 loss

$$R(f) = \mathbb{E}\left(1_{y \neq f(x)}\right) = \mathbb{P}(y \neq f(x))$$ error rate

Regression : $R(f) = \mathbb{E}\left(y - f(x)\right)^2$ "least-squares" $\quad 1 - \text{accuracy}$

# Bayes Risk and Bayes predictor

$R^* = R(f^*)$ — $f^*$ optimal $f \sim f^*$

$$R(f) = \mathbb{E}\left[\ell(y, f(x))\right] = \mathbb{E}\left[\underbrace{\mathbb{E}(\ell(y, f(x)) \mid x)}_{\text{function of } x}\right]$$

To simplify: $\mathcal{X}$ = finite.

$$= \sum_{x' \in \mathcal{X}} \underbrace{\mathbb{E}\left[\ell(y, f(x')) \mid x = x'\right]}_{\text{minimize separately}} \cdot p(x = x')$$

$f : \mathcal{X} \to \mathcal{Y}$

$f(x') \quad \forall x' \in \mathcal{X}$

$$f^*(x') \in \operatorname*{argmin}_{z \in \mathcal{Y}} \mathbb{E}\left(\ell(y, z) \mid x = x'\right)$$

Classif

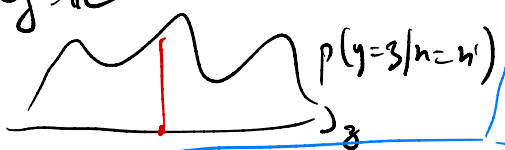
$\mathcal{Y}$ finite

$$f^*(x') = \operatorname*{argmin}_{z} \mathbb{P}(y \neq z \mid x = x')$$

$$= \operatorname*{argmax}_{z \in \mathcal{Y}} \mathbb{P}(y = z \mid x = x')$$

Regression: $\mathcal{Y} = \mathbb{R}$

$$f^*(x') \in \operatorname*{Argmin}_{z \in \mathbb{R}} \mathbb{E}\left[(y - z)^2 \mid x = x'\right) = \mathbb{E}[y \mid x = x']$$

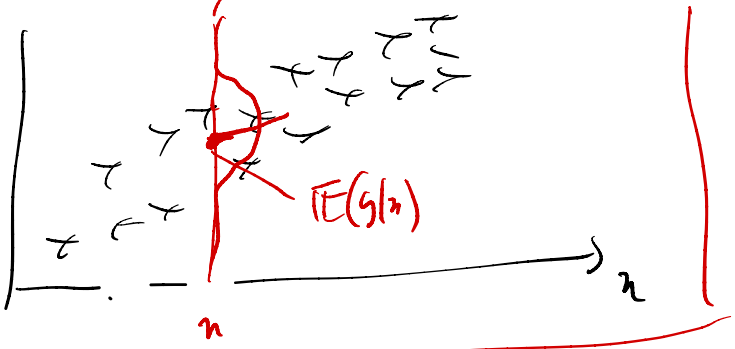



law of total variance: $\mathbb{E}(y-a)^2 = (\mathbb{E}y - a)^2 + \underbrace{\sigma^2}_{\mathrm{Var}(y)}$

Lemma: $\operatorname*{argmin}_a \mathbb{E}\left((y-a)^2\right) = \mathbb{E}y$

$\mathbb{E}\left((y-a)^2\right) = \mathbb{E}\left(y^2 - 2ay + a^2\right) = (\mathbb{E}y)^2 + \sigma^2 - 2a\,\mathbb{E}y + a^2$

$\partial/\partial a = -2\mathbb{E}y + 2a = 0$




Bayes predictor $f^*(x) = \underset{z \in Y}{\operatorname{argmin}}\ \mathbb{E}\left[\ell(y,z) \mid x = x\right]$

**Bayes risk**: $R(f^*) = \mathbb{E}_{x}\left[\min \mathbb{E}\left(\ell(y,z) \mid x = x\right)\right] = R^*$

Calvin: $R(f) - R^*$ **excess risk**

$\frac{\log n}{n} \to \frac{1}{n^{\alpha}}$

Two classes:

- local averaging
- Empirical risk min.

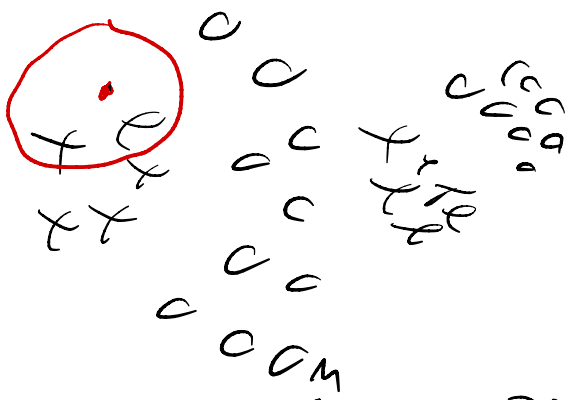
## local averaging

$$f^*(x) = \operatorname{argmin}_z \mathbb{E}\left[\ell(y,z) \mid x = x\right] = \int \ell(y,z)\, dp(y \mid x = x)$$

$$\downarrow$$

$$d\hat{p}(y \mid x = x)$$

training data dependent estimate



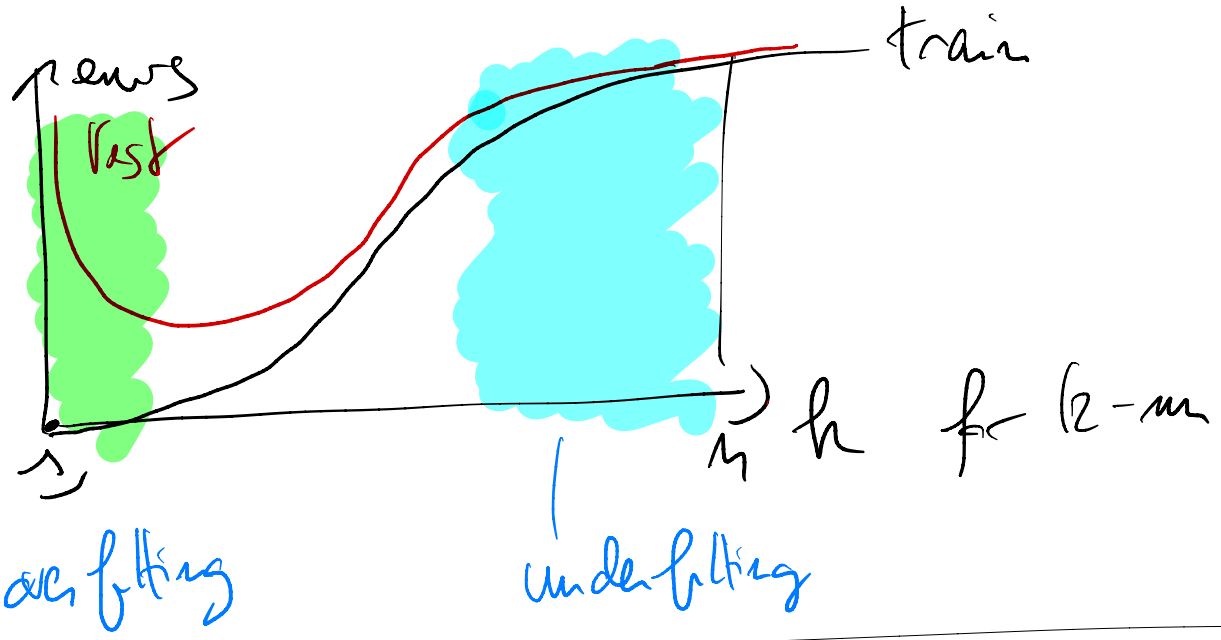
$$d\hat{p}(y \mid x) = \sum_{i=1}^{n} w_i(x)\, \text{Dirac}(y \mid y_i)$$

$\rightarrow$ K-nearest neighbors

$\rightarrow$ Nadaraya-Watson

Ex: square loss: $\hat{f}(x) = \sum w_i(x)\, y_i$

0-1 loss: majority vote weighted by $w_i$

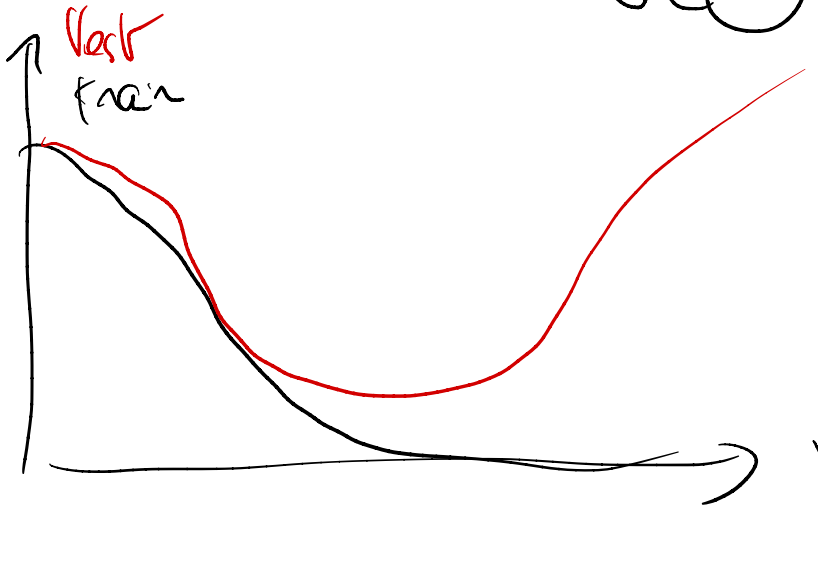


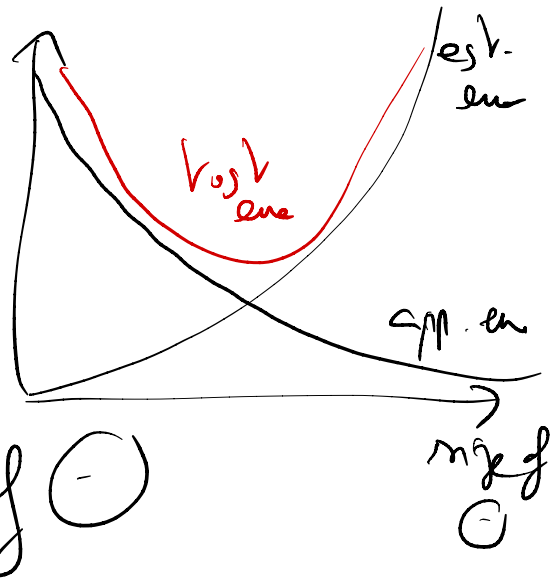

---

Empirical Risk Minimization : ERM

Model $f_\theta : \mathcal{X} \to \mathcal{Y}$ of functions parameterized by $\theta \in \Theta$

$$\hat{f} = f_{\hat{\theta}} \quad \text{s.t} \quad \hat{\theta} \in \underset{\theta \in \Theta}{\operatorname{argmin}} \; \hat{\mathcal{R}}(f_\theta) = \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f_\theta(x_i))$$



Decomposition between estimation error & approximation error

Model $\{f_\theta, \theta \in \Theta\}$.

Criteria: $R(f_{\hat{\theta}}) - R_* = \underbrace{R(f_{\hat{\theta}}) - \inf_{\theta' \in \Theta} R(f_{\theta'})}_{\geq 0} + \underbrace{\inf_{\theta' \in \Theta} R(f_{\theta'}) - R^*}_{\geq 0}$

estimation error
$\geq 0$
random
decreasing in increasing of $(-)$

approximation error
$\geq 0$
deterministic
independent in decreasing $(-)$

# goal of supervised learning

Data $\mathcal{D}_n(p) = (x_i, y_i)_{i=1,\dots,n}$ $\overset{\text{iid dist. } p}{\sim}$ $\Rightarrow$ ML algo $\mathcal{A}$ $\Rightarrow$ prediction function $\hat{f}: \mathcal{X} \to \mathcal{Y}$

$\mathcal{A}$ : datasets $\longrightarrow$ functions

Criteria $\mathcal{R}_p(\hat{f}) - \mathcal{R}_p^* = \mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)) - \mathcal{R}_p^*$

2 ways of removing randomness $\Rightarrow$ ① the expectation

$$\mathbb{E}_{\text{training data}}\, \mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)) - \mathcal{R}^* \leq \mathcal{B}(p)$$

↳ $\mathbb{E}$ testing data

$\Rightarrow$ ② high proba $\forall \delta \in (0,1)$, with proba $\geq 1-\delta$, $\mathcal{R}_p(\mathcal{A}(\mathcal{D}_n(p)) - \mathcal{R}^* \leq \mathcal{B}(p, \delta)$

PAC

Markov inequality : $X \geq c$

$$\mathbb{P}(X \geq \varepsilon) \leq \frac{\mathbb{E}X}{\varepsilon}$$

$\forall \varepsilon$, $\in \forall \delta$ with prob $1-\delta$, $X < \frac{\mathbb{E}X}{\delta}$

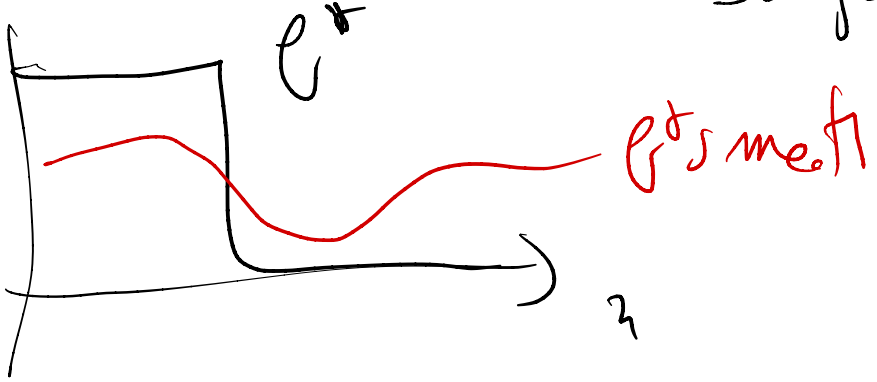
$= \delta$

---

$$\sup_{p \in \mathcal{P}} \mathbb{E}_{\text{training data}} \mathcal{R}_p\left(\mathcal{A}(\mathcal{D}_n, G)\right) - \mathcal{R}_p^*$$

No free lunch

$\mathcal{P} : \{ q(z, \cdot),\ \rho^*(x)$ smooth enough $\}$



- learning spaces
  - ERM
  - uniform LLN
  - local averages
    - kernel methods
    - sparsity
    - NN