BIG DIVE

DATA SCIENCE & ANALYTICS

Learning Theory

André Panisson

Organized by top-ix torino piemonte internet exchange

Designed for INTESA SANPAOLO

In collaboration with aizoOn technology consulting, ISI Foundation, TODO

When someone starts using machine learning to deal with a problem, there are two main approaches:

Try everything at hand (brute force approach)

Use the theory as a guide to choose the right strategy

LEARNING THEORY

- Do I have enough data for adequate learning?
- Is the model complexity adequate for the problem?
- What is the best strategy to reduce error/ increase performance?

How can my model generalize better?

- Have a more/less complex model?
- Collect more samples?
- Have more/less dataset features?

<u>SUPPORT FOR STRATEGIC DECISIONS</u>

THE LEARNING PROBLEM

Metaphor: Credit approval

Applicant information:

age	23 years
gender	male
annual salary	$30,000
years in residence	1 year
years in job	1 year
current debt	$15,000
···	···

Approve credit?

Components of learning

Formalization:

- Input: $\mathbf{x}$ (*customer application*)

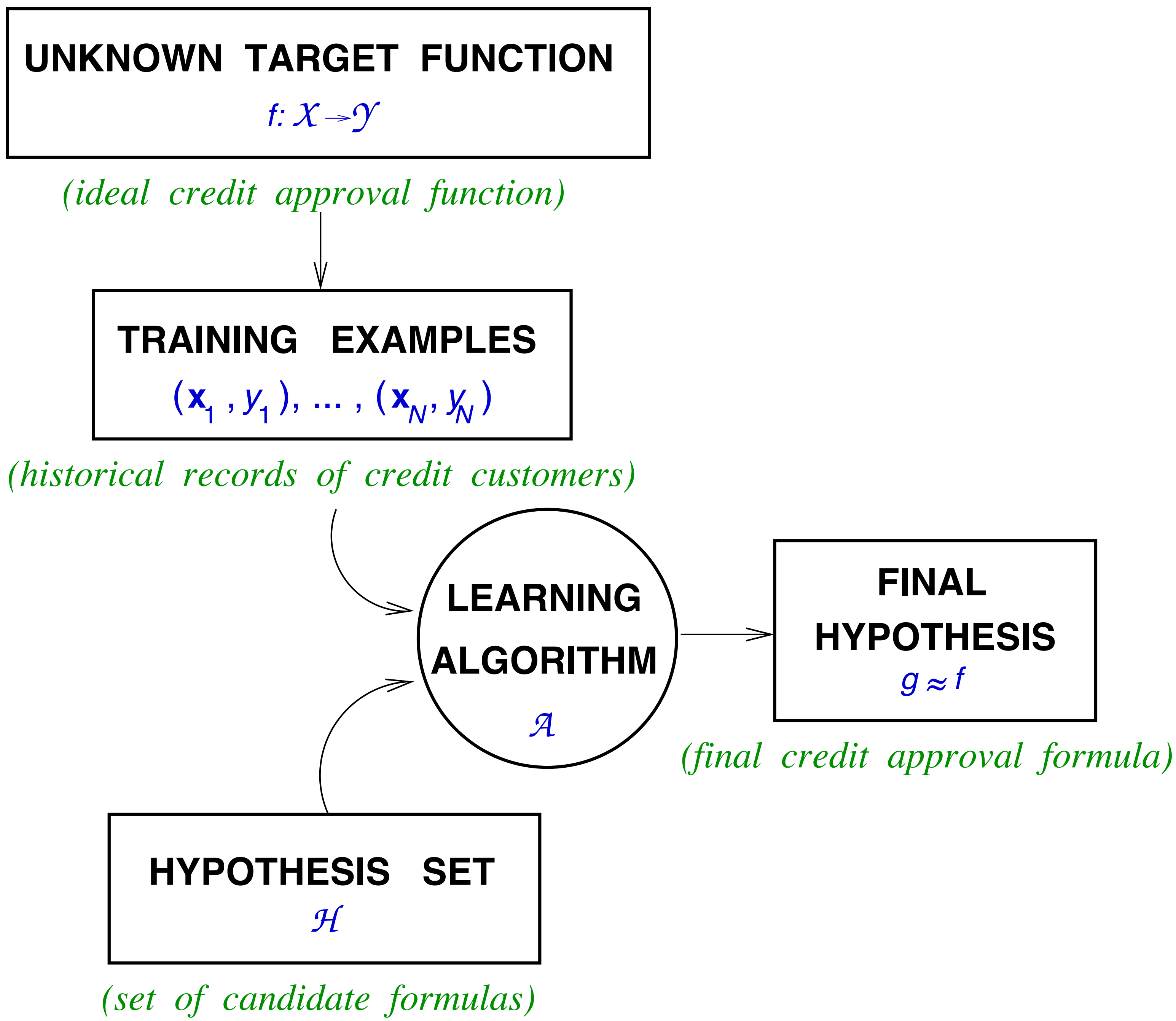
- Output: y (*good/bad customer?*)

- Target function: $f : \mathcal{X} \to \mathcal{Y}$ (*ideal credit approval formula*)

- Data: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$ (*historical records*)

$$\downarrow \quad \downarrow \quad \downarrow$$

- Hypothesis: $g : \mathcal{X} \to \mathcal{Y}$ (*formula to be used*)


UNKNOWN TARGET FUNCTION
$f: \mathcal{X} \to \mathcal{Y}$
(ideal credit approval function)
TRAINING EXAMPLES
$(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$
(historical records of credit customers)
LEARNING ALGORITHM
$\mathcal{A}$
FINAL HYPOTHESIS
$g \approx f$
(final credit approval formula)
HYPOTHESIS SET
$\mathcal{H}$
(set of candidate formulas)

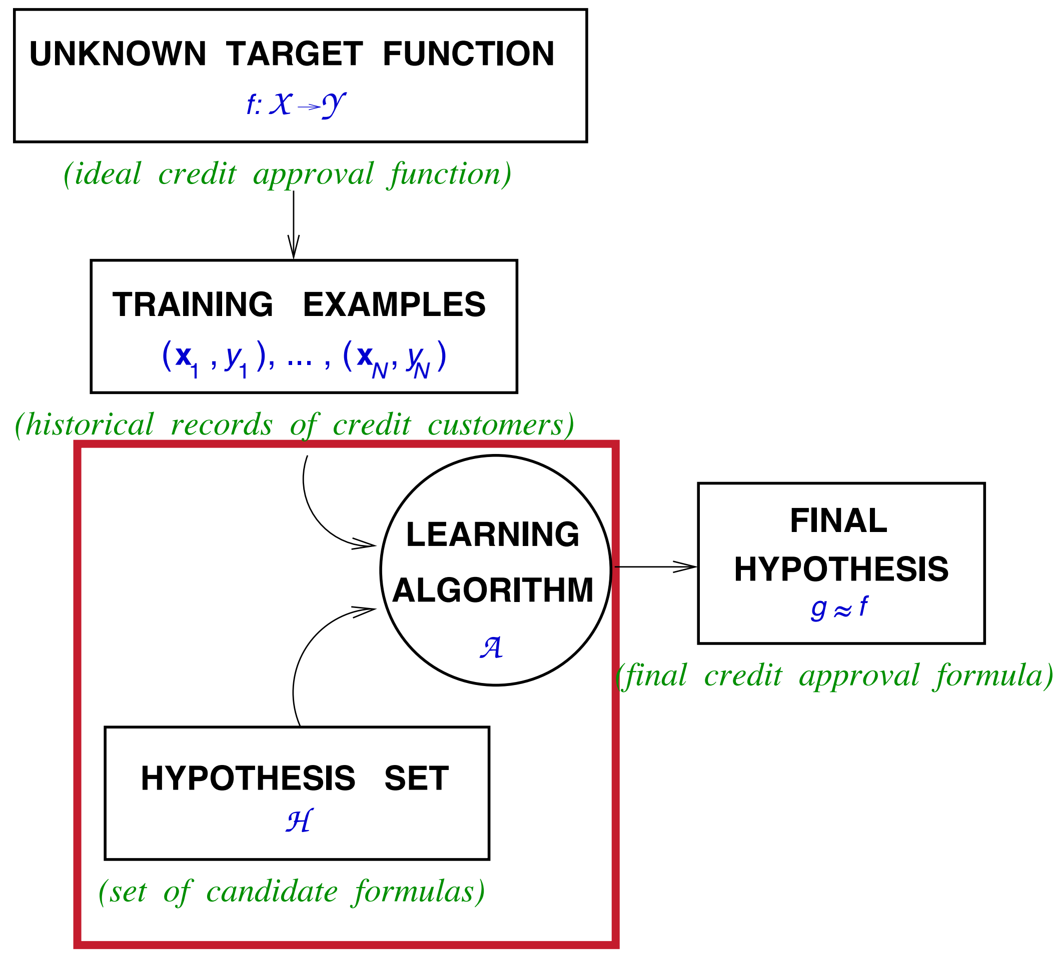
The 2 components of the learning problem:

- The Hypothesis Set

  $$\mathcal{H} = \{h\} \quad g \in \mathcal{H}$$

- The Learning Algorithm $\mathcal{A}$

Together, they are referred as the **Learning Model**



A simple hypothesis set - the 'perceptron'

For input $\mathbf{x} = (x_1, \cdots, x_d)$ 'attributes of a customer'

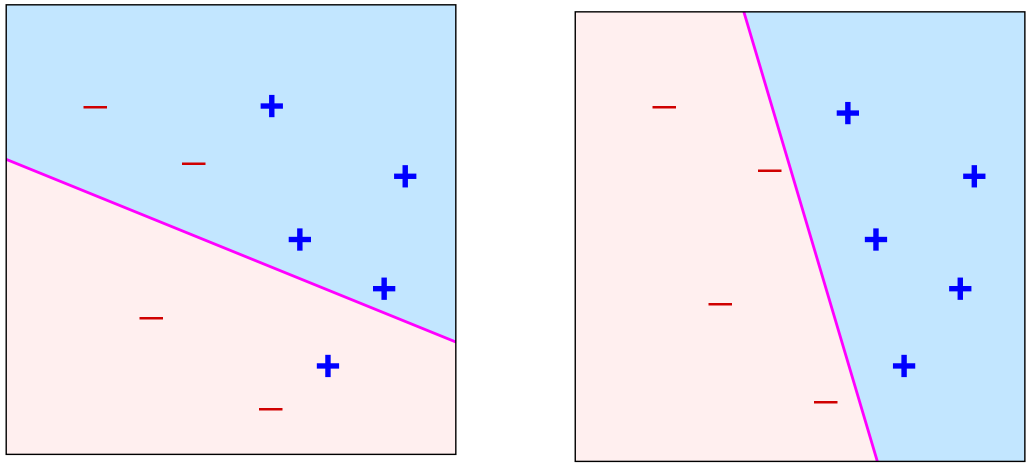
$$\text{Approve credit if} \quad \sum_{i=1}^{d} w_i x_i > \text{threshold},$$

$$\text{Deny credit if} \quad \sum_{i=1}^{d} w_i x_i < \text{threshold}.$$

This linear formula $h \in \mathcal{H}$ can be written as

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - \text{threshold}\right)$$

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i \, x_i\right) + \quad w_0 \quad\right)$$



Introduce an artificial coordinate $x_0 = 1$:

$$h(\mathbf{x}) = \text{sign}\left(\sum_{i=0}^{d} w_i \, x_i\right)$$

In vector form, the perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\intercal \mathbf{x})$$

PLA - The Perceptron Learning Algorithm

The perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\intercal \mathbf{x})$$

Given the training set:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_N, y_N)$$

pick a misclassified point:

$$\text{sign}(\mathbf{w}^\intercal \mathbf{x}_n) \neq y_n$$

and update the weight vector:

$$\boxed{\mathbf{w} \leftarrow \mathbf{w} + y_n \mathbf{x}_n}$$

IS LEARNING FEASIBLE?

A Related Experiment

- Consider a 'bin' with red and green marbles.

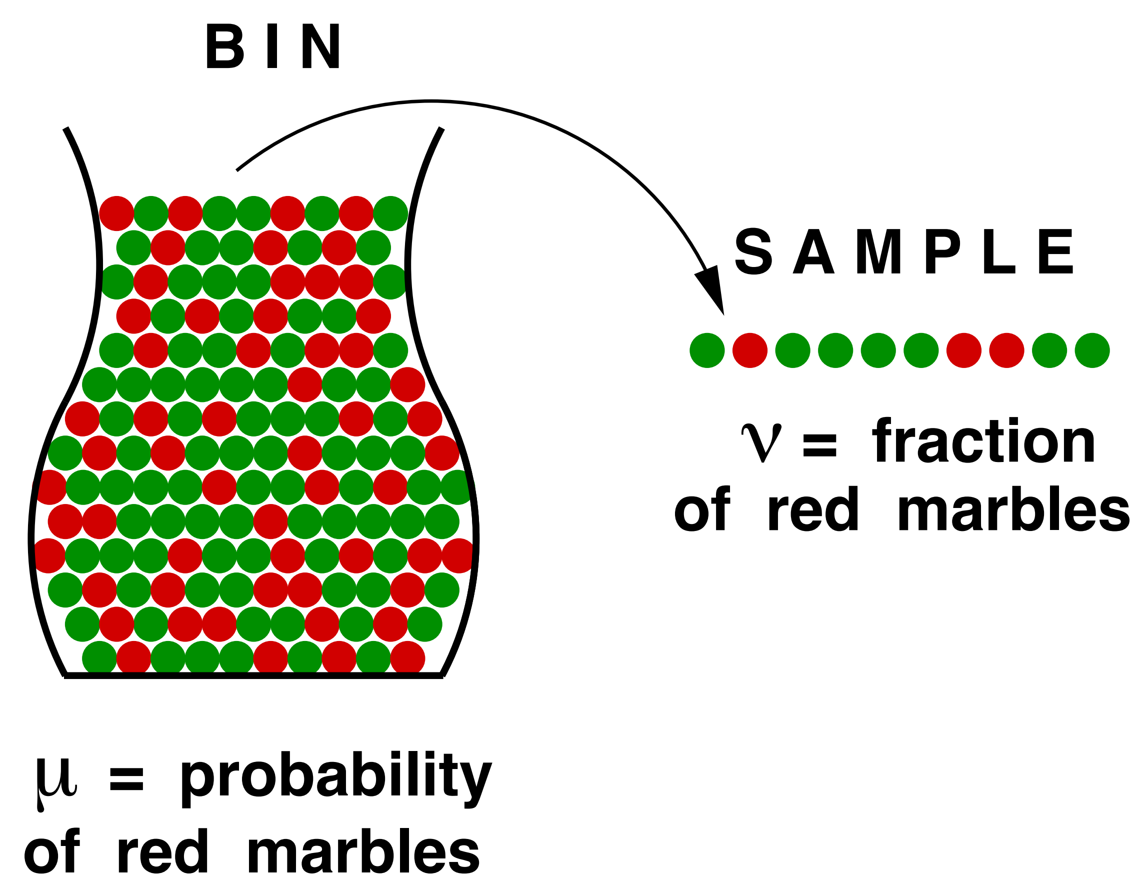
  P[picking a red marble] $= \mu$

  P[picking a green marble] $= 1 - \mu$

- The value of μ is unknown to us.

- We pick N marbles independently.

- The fraction of red marbles in sample $= \nu$



What does ν say about μ?

In a big sample (large N), ν is probably close to μ (within ϵ).

Formally,

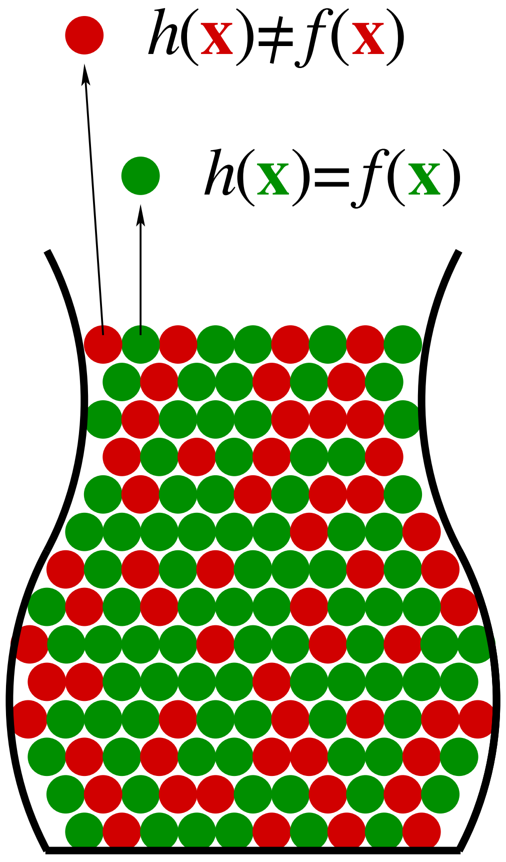
$$\mathbb{P}\left[\,|\nu - \mu| > \epsilon\,\right] \leq 2e^{-2\epsilon^2 N}$$

This is called **Hoeffding's Inequality**.

Connection to Learning

Bin: The unknown is a number μ

Learning: The unknown is a function $f : \mathcal{X} \to \mathcal{Y}$

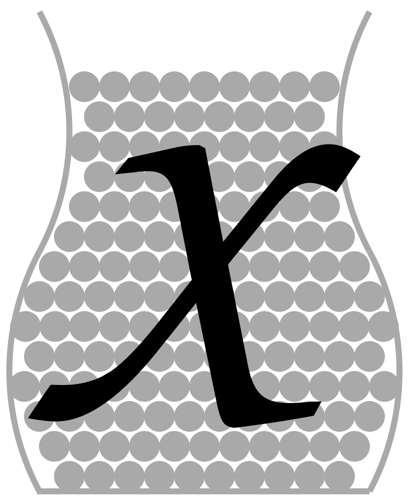


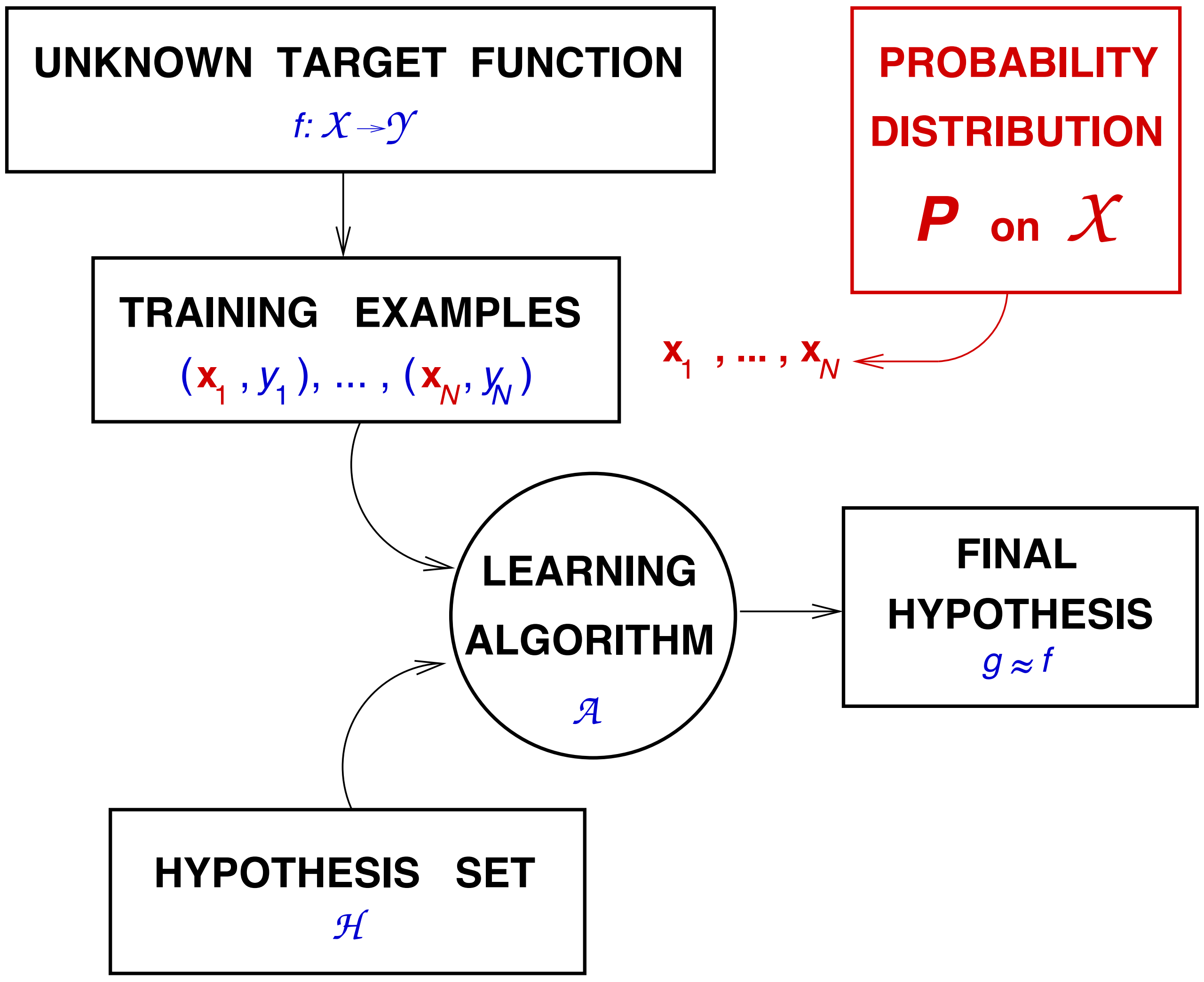
Each marble ● is a point $\mathbf{x} \in \mathcal{X}$

- ● : Hypothesis got it right $h(\mathbf{x}) = f(\mathbf{x})$
- ● : Hypothesis got it wrong $h(\mathbf{x}) \neq f(\mathbf{x})$

Back to the learning diagram

The bin analogy:





Notation for learning

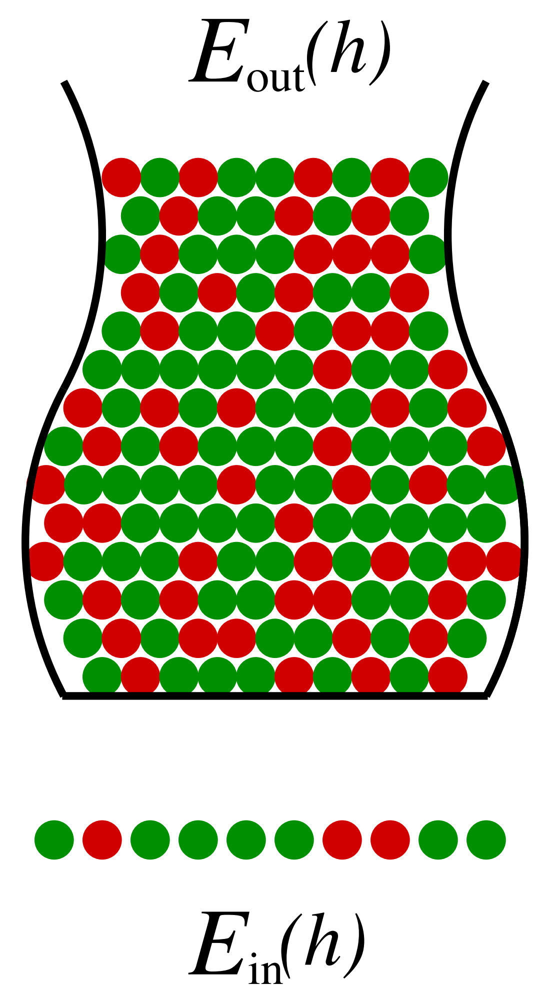
Both μ and ν depend on which hypothesis h

ν is '**in sample**' denoted by $E_{\text{in}}(h)$

μ is '**out of sample**' denoted by $E_{\text{out}}(h)$

The Hoeffding inequality becomes:

$$\mathbf{P}[\ |E_{\text{in}}(h) - E_{\text{out}}(h)| > \epsilon\ ] \ \leq\ 2e^{-2\epsilon^2 N}$$

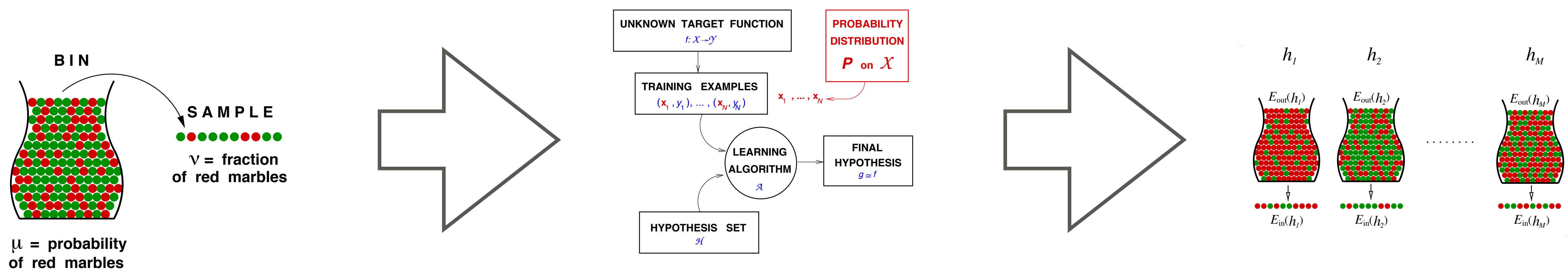


Next Steps

With the Hoeffding inequality, we can estimate $E_{in}(h)$ for any hypothesis h choosen from $\mathcal{H}$, independently from the others.

However, our chosen hypothesis g is one of $h_1, h_2, \ldots, h_M$

Therefore, the Hoeffding inequality doesn't apply to the whole hypothesis set $\mathcal{H}$!



A simple solution: the Union Bound

$$
\begin{aligned}
\mathsf{P}[\ |E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon\ ] \ \le\ \mathsf{P}[\quad & |E_{\text{in}}(h_1) - E_{\text{out}}(h_1)| > \epsilon \\
\textbf{or}\ & |E_{\text{in}}(h_2) - E_{\text{out}}(h_2)| > \epsilon \\
& \cdots \\
\textbf{or}\ & |E_{\text{in}}(h_M) - E_{\text{out}}(h_M)| > \epsilon\ ] \\
\le\ & \sum_{m=1}^{M} \mathsf{P}\left[|E_{\text{in}}(h_m) - E_{\text{out}}(h_m)| > \epsilon\right]
\end{aligned}
$$

$$
\mathsf{P}\left[|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon\right] \le 2{\color{red}M} e^{-2\epsilon^2 N}
$$

In order to make learning feasible, we need:

$$E_{\text{out}}(g) \approx E_{\text{in}}(g)$$

so the out-of-sample error is similar to the in-sample error

At the same time, we need $g \approx f$,

which means $E_{\text{out}}(g) \approx 0$,

and is achieved through

$$E_{\text{out}}(g) \approx E_{\text{in}}(g) \qquad \text{and} \qquad E_{\text{in}}(g) \approx 0$$

Machine Learning: The BIG Questions

Can we make $E_{\text{in}}(g)$ small enough?

APPROXIMATION

Can we make sure that $E_{\text{out}}(g)$ is close enough to $E_{\text{in}}(g)$?

GENERALIZATION

$$\mathsf{P}\left[\,|E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon\,\right] \leq 2{\color{red}M}e^{-2\epsilon^2 N}$$

${\color{red}M}$ represents the complexity of the hypothesis set

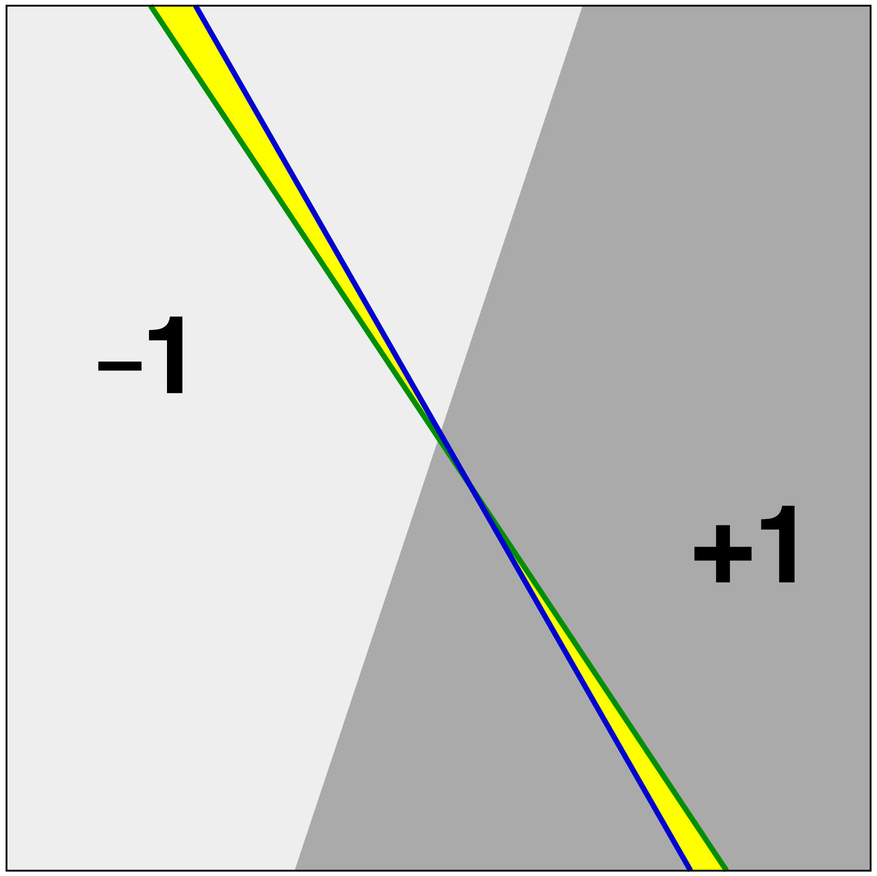
Can we improve on ${\color{red}M}$?

Yes, bad events are *very* overlapping!

ΔE_{out}: change in $+1$ and -1 areas

ΔE_{in}: change in labels of data points

$$|E_{\text{in}}(h_1) - E_{\text{out}}(h_1)| \approx |E_{\text{in}}(h_2) - E_{\text{out}}(h_2)|$$



dichotomy: a binary labeling of $\mathcal{X}$

A hypothesis $h : \mathcal{X} \rightarrow \{-1, +1\}$

A dichotomy $h : \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\} \rightarrow \{-1, +1\}$

Number of hypotheses $|\mathcal{H}|$ can be infinite

Number of dichotomies $|\mathcal{H}(\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N)|$ is at most 2^N

Candidate for replacing M

The growth function

The growth function counts the <u>most</u> dichotomies on any N points

$$m_{\mathcal{H}}(N) = \max_{\mathbf{x}_1, \cdots, \mathbf{x}_N \in \mathcal{X}} |\mathcal{H}(\mathbf{x}_1, \cdots, \mathbf{x}_N)|$$

The growth function satisfies:

$$m_{\mathcal{H}}(N) \leq 2^N$$

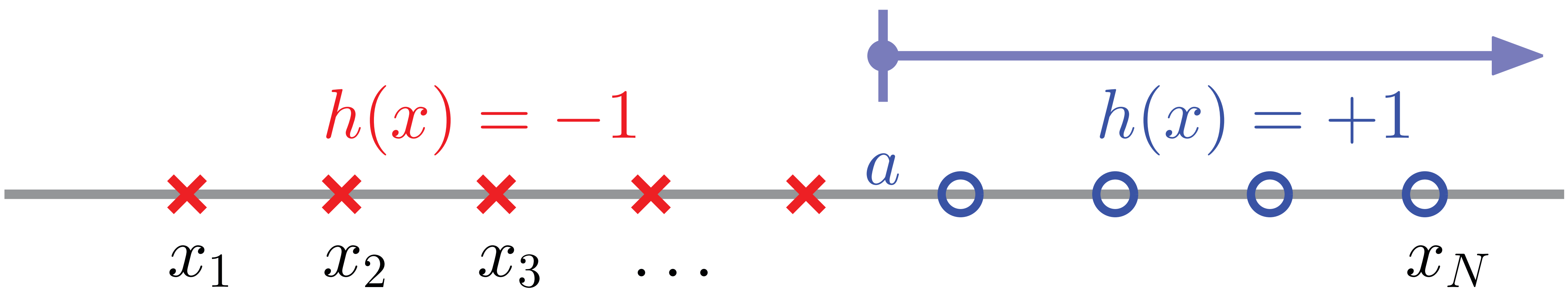
Break Point (k)

If no data set of size k can be **shattered** by $\mathcal{H}$,
then k is a break point of $\mathcal{H}$
(**shatter**: produce all 2^{k} dichotomies)

$$m_{\mathcal{H}}(N) \quad \leq \quad \underbrace{\sum_{i=0}^{k-1} \binom{N}{i}}_{\text{maximum power is } N^{k-1}}$$

The growth function is polinomial!

Example 1: Positive Rays

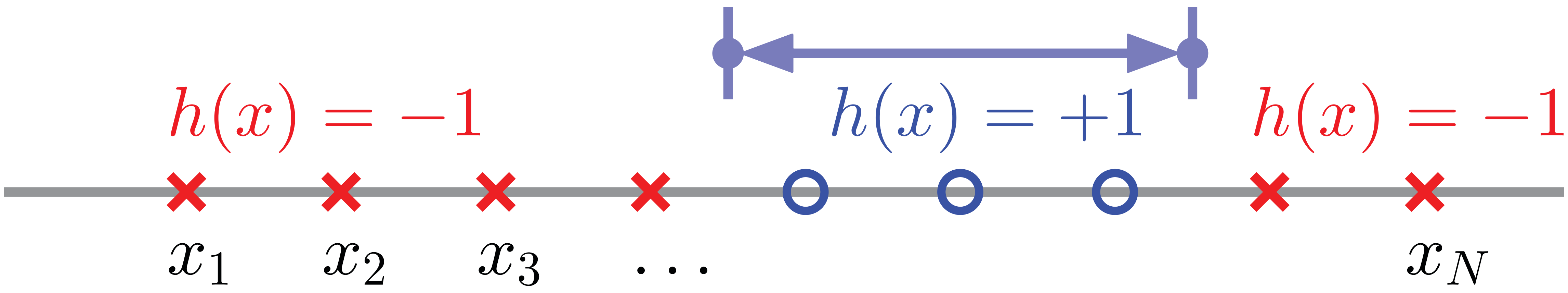


$\mathcal{H}$ is set of $h\colon \mathbb{R} \to \{-1, +1\}$

$h(x) = \mathrm{sign}(x - a)$

$m_{\mathcal{H}}(N) = N + 1$

Example 2: Positive Intervals



$\mathcal{H}$ is set of $h\colon \mathbb{R} \to \{-1, +1\}$

Place interval ends in two of $N+1$ spots

$$m_{\mathcal{H}}(N) = \binom{N+1}{2} + 1 = \tfrac{1}{2}N^2 + \tfrac{1}{2}N + 1$$

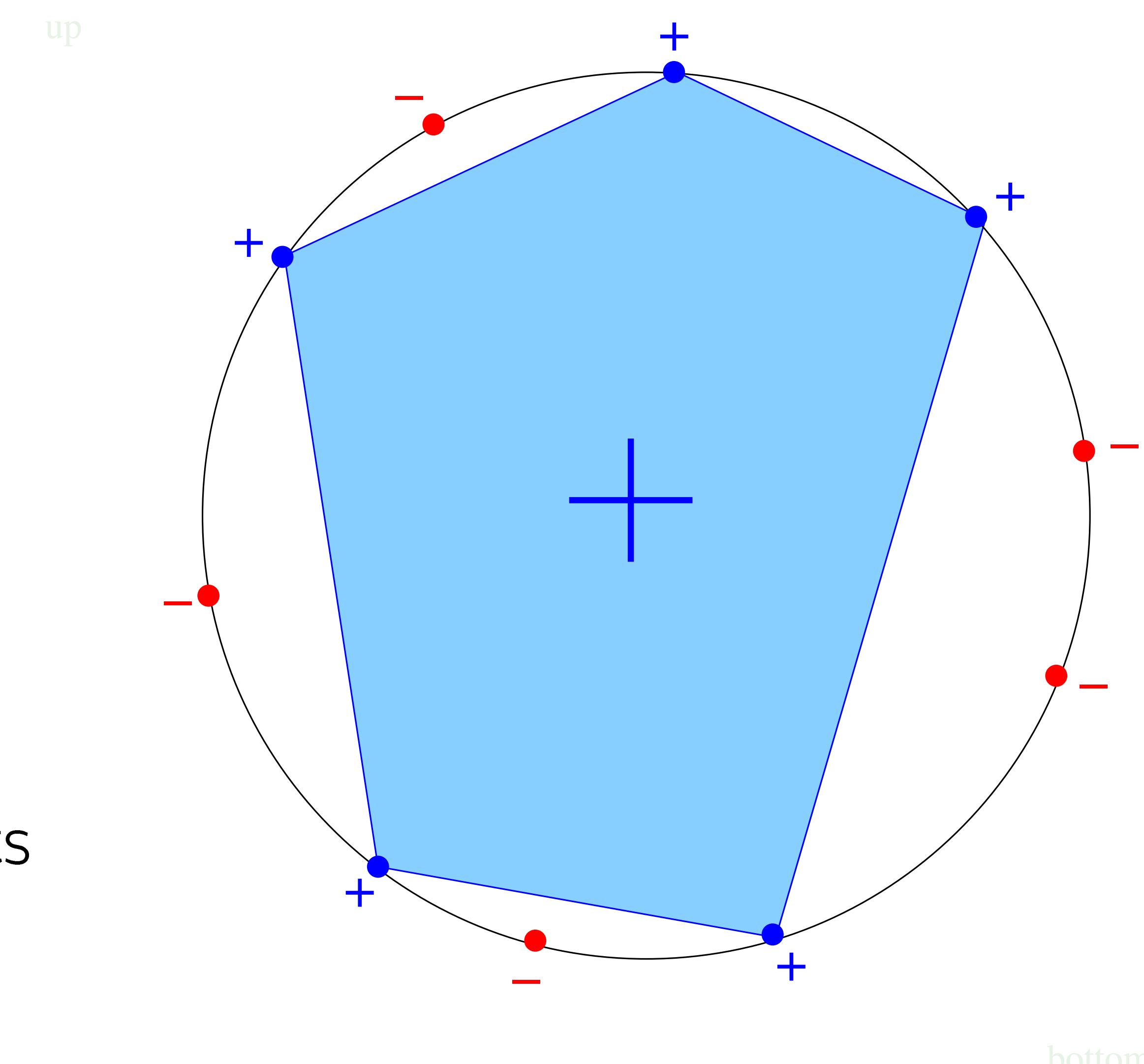
Example 3: Convex Sets

$\mathcal{H}$ is set of $h\colon \mathbb{R}^2 \to \{-1, +1\}$

$h(\mathbf{x}) = +1$ is convex

$m_{\mathcal{H}}(N) = 2^N$

The N points are 'shattered' by convex sets

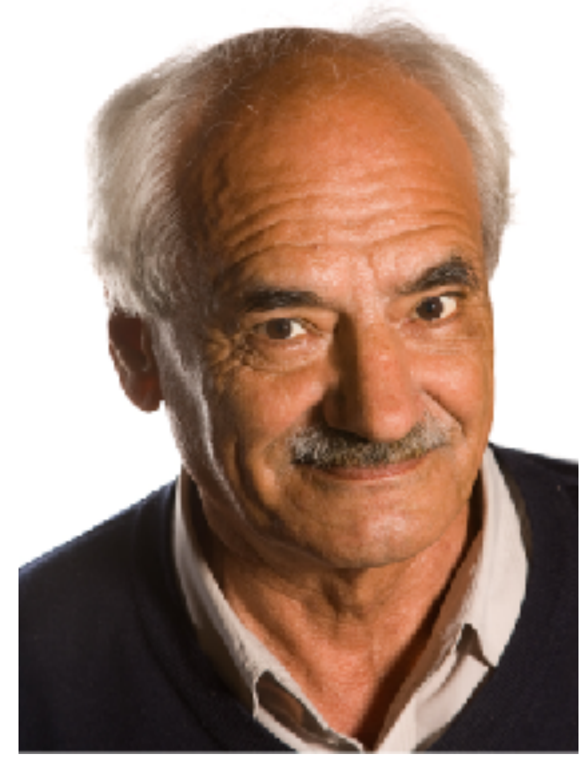


The Vapnic-Chervonenkis Inequality

M is replaced by the **growth function**

$$\mathbb{P}[\ |E_{\text{in}}(g) - E_{\text{out}}(g)| > \epsilon\ ] \ \leq\ 4\ \ m_{\mathcal{H}}(2N)\ \ e^{-\ \frac{1}{8}\ \epsilon^2 N}$$

Vladimir Vapnik



Alexey Chervonenkis

The VC dimension

The hypothesis set $\mathcal{H}$ is said to shatter a set $\mathcal{S} \subset \mathcal{X}$
if $\mathcal{H}$ can realize all $2^{|\mathcal{S}|}$ binary labelings of $\mathcal{S}$.
The Vapnik-Chervonenkis dimension of $\mathcal{H}$ is
the size of the largest subset of $\mathcal{S}$ that $\mathcal{H}$ can shatter.

Definition of VC dimension

The VC dimension of a hypothesis set $\mathcal{H}$, denoted by $d_{\mathrm{VC}}(\mathcal{H})$, is

the largest value of N for which $m_{\mathcal{H}}(N) = 2^N$

"the most points $\mathcal{H}$ can shatter"

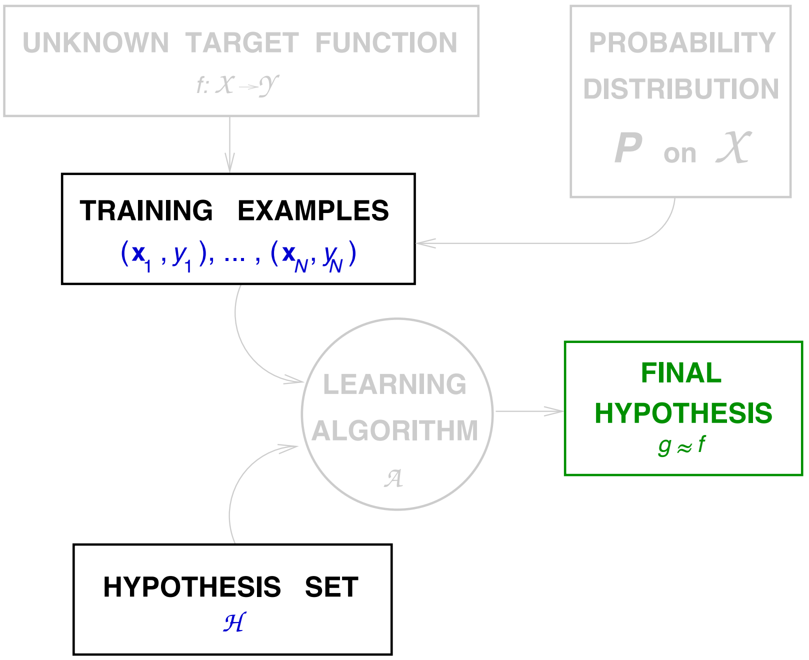
$N \leq d_{\mathrm{VC}}(\mathcal{H}) \implies \mathcal{H}$ can shatter N points

$k > d_{\mathrm{VC}}(\mathcal{H}) \implies k$ is a break point for $\mathcal{H}$

VC dimension and Learning

$d_{\mathrm{VC}}(\mathcal{H})$ is finite $\Longrightarrow$ $g \in \mathcal{H}$ will generalize

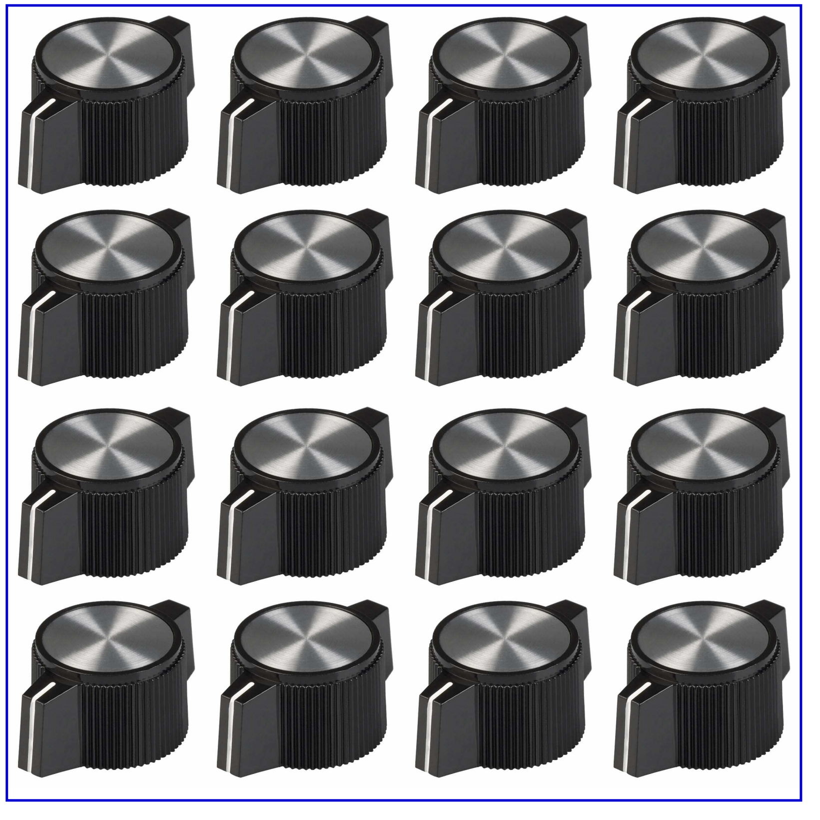
- Independent of the **learning algorithm**
- Independent of the **input distribution**
- Independent of the **target function**



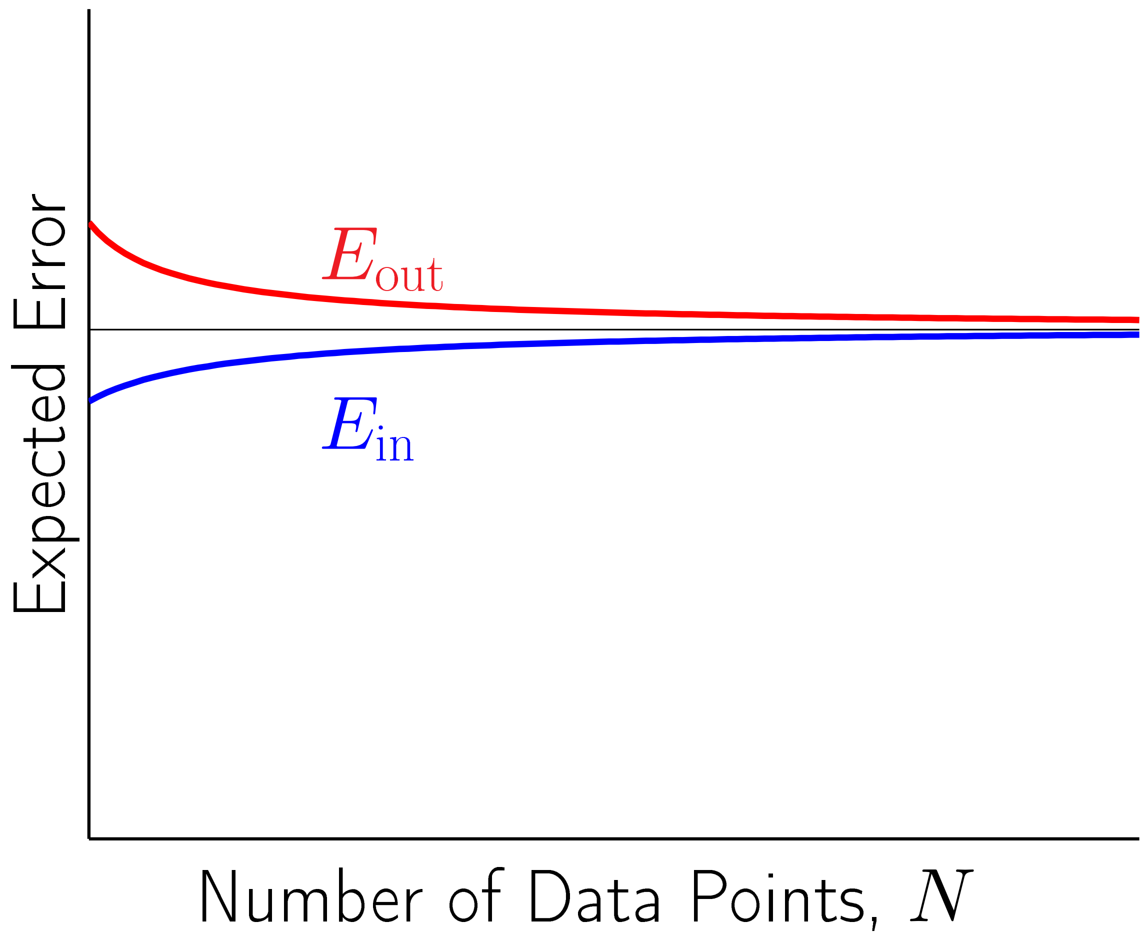
Parameters create degrees of freedom

\# of parameters: **analog** degrees of freedom

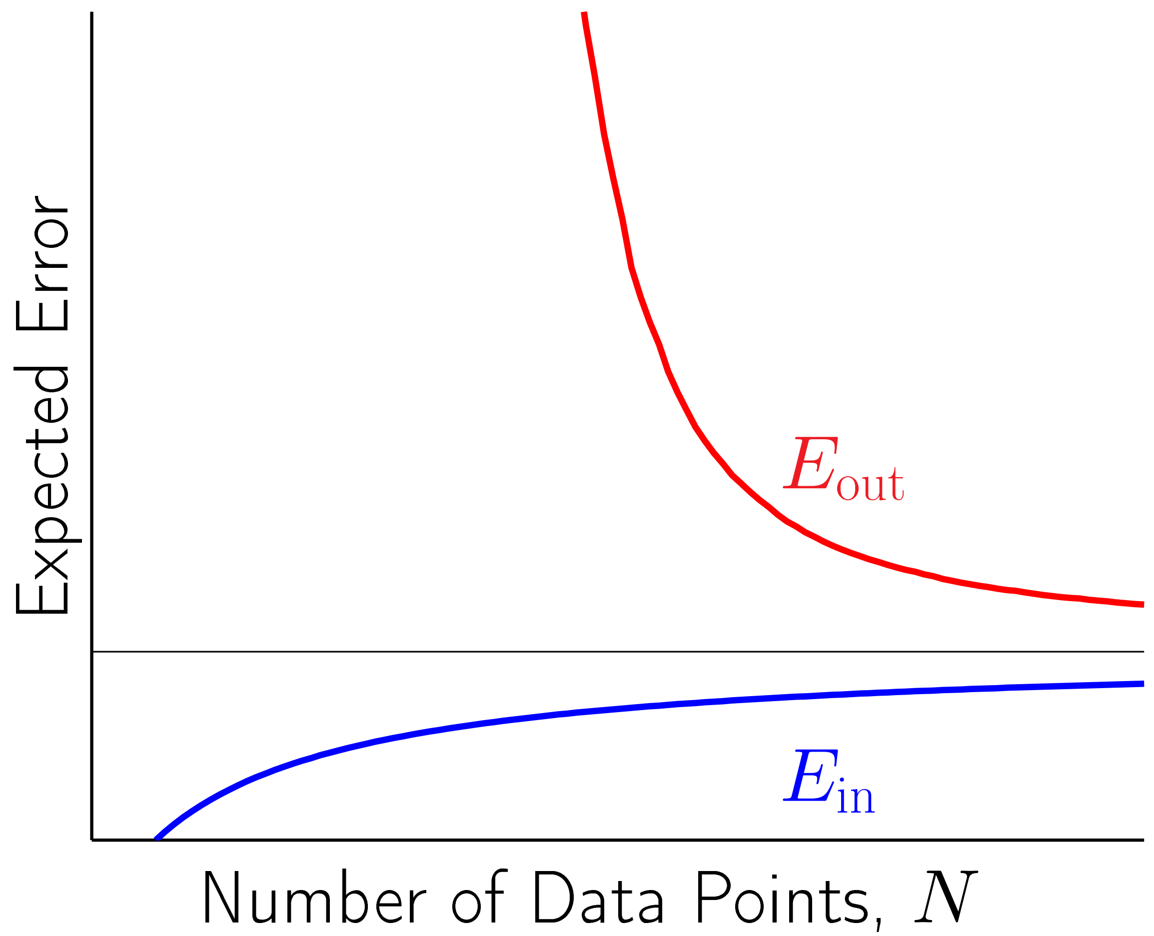
d_{VC}: equivalent **'binary'** degrees of freedom



E_{in} and E_{out} in terms of N

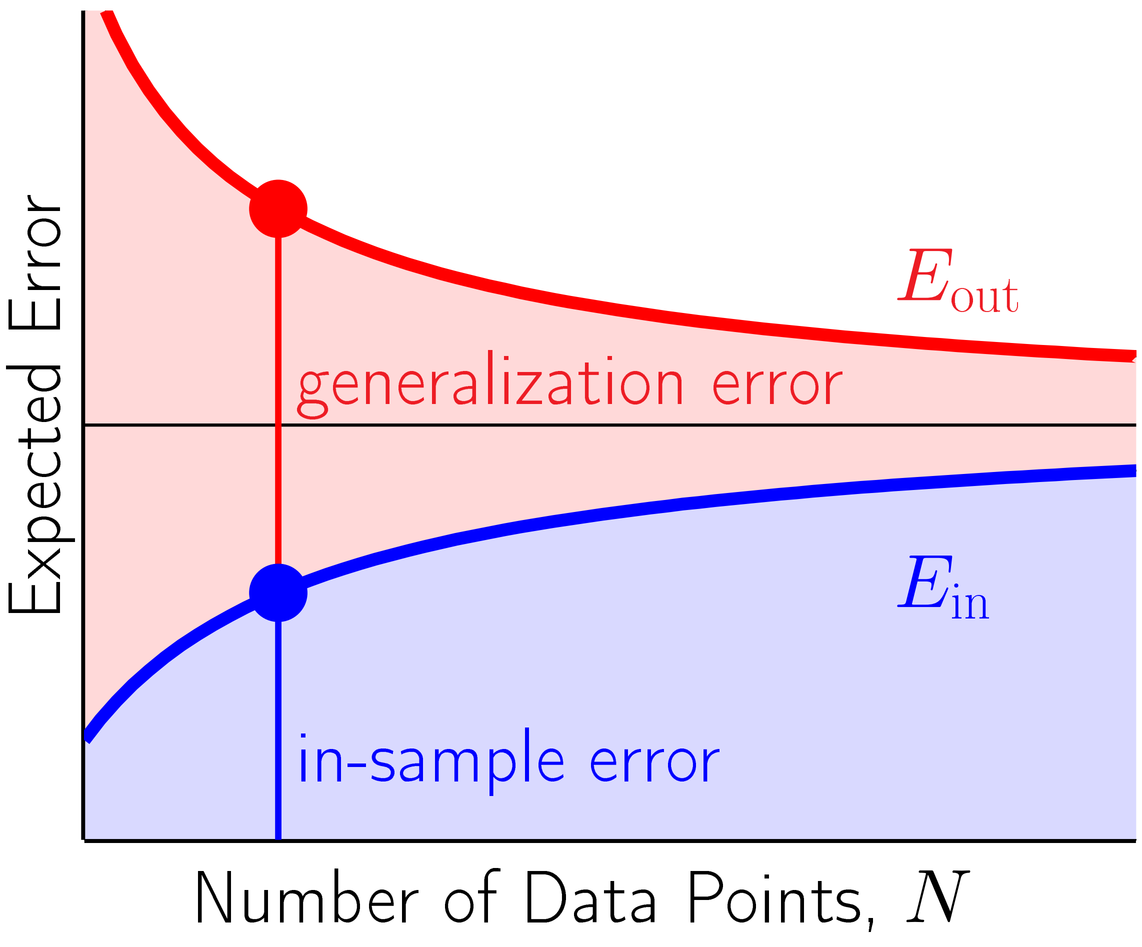


Simple Model

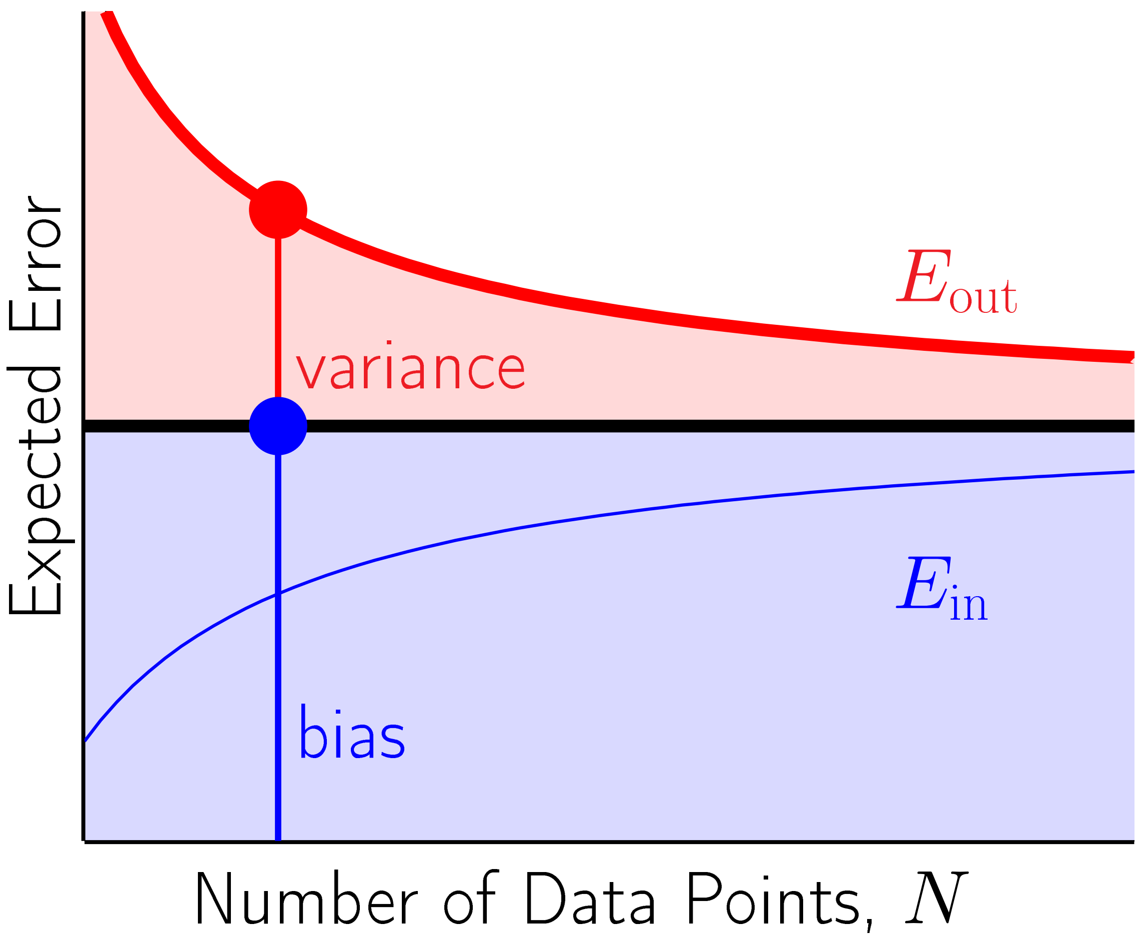


Complex Model

VC versus bias-variance



VC analysis



bias-variance