

# Machine Learning Foundations

## (機器學習基石)



### Lecture 4: Feasibility of Learning

Hsuan-Tien Lin (林軒田)

htlin@csie.ntu.edu.tw

Department of Computer Science  
& Information Engineering

National Taiwan University  
(國立台灣大學資訊工程系)



# Roadmap

## ① When Can Machines Learn?

### Lecture 3: Types of Learning

focus: **binary classification** or **regression** from a **batch** of **supervised** data with **concrete** features

### Lecture 4: Feasibility of Learning

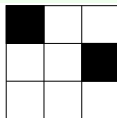
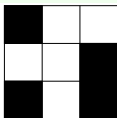
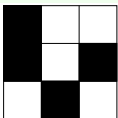
- Learning is Impossible?
- Probability to the Rescue
- Connection to Learning
- Connection to Real Learning

## ② Why Can Machines Learn?

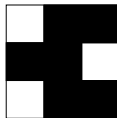
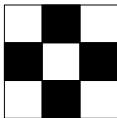
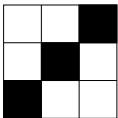
## ③ How Can Machines Learn?

## ④ How Can Machines Learn Better?

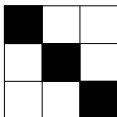
# A Learning Puzzle



$$y_n = -1$$



$$y_n = +1$$



$$g(\mathbf{x}) = ?$$

**let's test your 'human learning'  
with 6 examples :-)**

## Two Controversial Answers

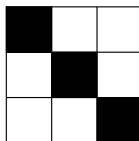
whatever you say about  $g(\mathbf{x})$ ,



$$y_n = -1$$



$$y_n = +1$$



$$g(\mathbf{x}) = ?$$

truth  $f(\mathbf{x}) = +1$  because ...

- symmetry  $\Leftrightarrow +1$
- (black or white count = 3) or (black count = 4 and middle-top black)  $\Leftrightarrow +1$

truth  $f(\mathbf{x}) = -1$  because ...

- left-top black  $\Leftrightarrow -1$
- middle column contains at most 1 black and right-top white  $\Leftrightarrow -1$

all valid reasons, your **adversarial teacher**  
can always call you 'didn't learn'. :-)

# A 'Simple' Binary Classification Problem

$\mathbf{x}_n$	$y_n = f(\mathbf{x}_n)$
0 0 0	○
0 0 1	×
0 1 0	×
0 1 1	○
1 0 0	×

- $\mathcal{X} = \{0, 1\}^3$ ,  $\mathcal{Y} = \{\text{○}, \text{×}\}$ , can enumerate all candidate  $f$  as  $\mathcal{H}$

pick  $g \in \mathcal{H}$  with all  $g(\mathbf{x}_n) = y_n$  (like PLA),  
**does  $g \approx f$ ?**

## No Free Lunch

$\mathcal{D}$

$\mathbf{x}$	$y$	$g$	$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$	$f_7$	$f_8$
0 0 0	○	○	○	○	○	○	○	○	○	○
0 0 1	×	×	×	×	×	×	×	×	×	×
0 1 0	×	×	×	×	×	×	×	×	×	×
0 1 1	○	○	○	○	○	○	○	○	○	○
1 0 0	×	×	×	×	×	×	×	×	×	×
1 0 1		?	○	○	○	○	×	×	×	×
1 1 0		?	○	○	×	×	○	○	×	×
1 1 1		?	○	×	○	×	○	×	○	×

- $g \approx f$  inside  $\mathcal{D}$ : sure!
- $g \approx f$  outside  $\mathcal{D}$ : **No!** (but that's really what we want!)

If we learned from  $\mathcal{D}$ , it would be irrational to infer something that is outside from  $\mathcal{D}$ .

learning from  $\mathcal{D}$  (to infer something outside  $\mathcal{D}$ )  
is doomed if **any 'unknown'  $f$  can happen**. :-)

# Fun Time

This is a popular 'brain-storming' problem, with a claim that 2% of the world's cleverest population can crack its 'hidden pattern'.

$$(5, 3, 2) \rightarrow 151022, \quad (7, 2, 5) \rightarrow ?$$

It is like a 'learning problem' with  $N = 1$ ,  $\mathbf{x}_1 = (5, 3, 2)$ ,  $y_1 = 151022$ .  
Learn a hypothesis from the one example to predict on  $\mathbf{x} = (7, 2, 5)$ .  
What is your answer?

- ① 151026
- ② 143547
- ③ I need more examples to get the correct answer
- ④ there is no 'correct' answer

# Fun Time

This is a popular ‘brain-storming’ problem, with a claim that 2% of the world’s cleverest population can crack its ‘hidden pattern’.

$$(5, 3, 2) \rightarrow 151022, \quad (7, 2, 5) \rightarrow ?$$

It is like a ‘learning problem’ with  $N = 1$ ,  $\mathbf{x}_1 = (5, 3, 2)$ ,  $y_1 = 151022$ .  
Learn a hypothesis from the one example to predict on  $\mathbf{x} = (7, 2, 5)$ .  
What is your answer?

- ① 151026
- ② 143547
- ③ I need more examples to get the correct answer
- ④ there is no ‘correct’ answer

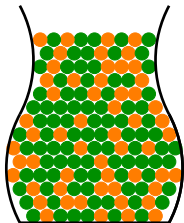
Reference Answer: ④ The answer from Internet is 2,  
but it's just one explanation.

Following the same nature of the no-free-lunch problems discussed,  
we cannot hope to be correct under this ‘adversarial’ setting. BTW,  
② is the designer’s answer: the first two digits =  $x_1 \cdot x_2$ ; the next two  
digits =  $x_1 \cdot x_3$ ; the last two digits =  $(x_1 \cdot x_2 + x_1 \cdot x_3 - x_2)$ .



# Inferring Something Unknown

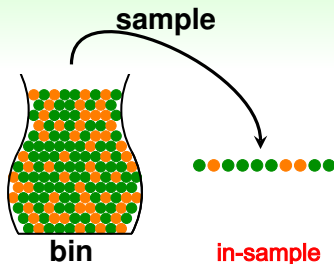
difficult to infer **unknown target  $f$  outside  $\mathcal{D}$**  in learning;  
can we infer **something unknown** in **other scenarios**?



- consider a bin of many many **orange** and **green** marbles
- do we **know** the **orange** portion (probability)? **No!**

can you **infer** the **orange** probability?

**by sample**

Statistics 101: Inferring **Orange** Probability**bin**

assume

orange probability =  $\mu$ ,green probability =  $1 - \mu$ ,with  $\mu$  **unknown****sample** $N$  marbles sampled independently, withorange fraction =  $\nu$ ,green fraction =  $1 - \nu$ ,now  $\nu$  **known**

does **in-sample**  $\nu$  say anything about  
out-of-sample  $\mu$ ?

# Possible versus Probable

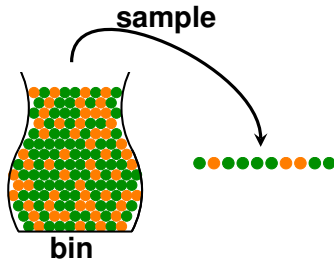
does **in-sample**  $\nu$  say anything about out-of-sample  $\mu$ ?

**No!**

possibly not: sample can be mostly **green** while bin is mostly **orange**

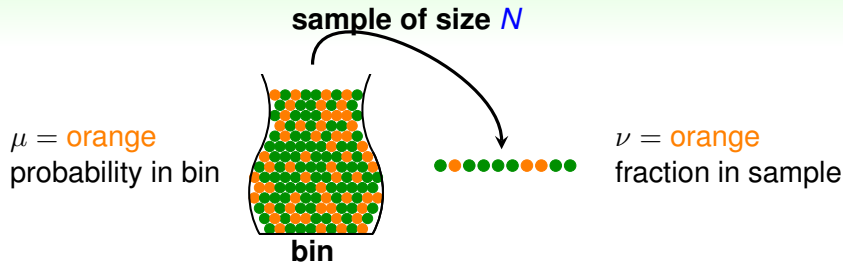
**Yes!**

probably yes: in-sample  $\nu$  likely **close** to unknown  $\mu$



formally, **what does**  $\nu$  **say about**  $\mu$ ?

# Hoeffding's Inequality (1/2)



- in big sample ( $N$  large),  $\nu$  is probably close to  $\mu$  (within  $\epsilon$ )

The probability of not being close

$$\mathbb{P} [|\nu - \mu| > \epsilon] \leq 2 \exp(-2\epsilon^2 N)$$

If sample size is very big, the probability of not being close is very small.

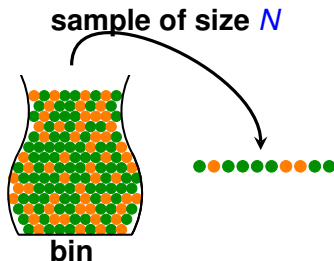
- called Hoeffding's Inequality, for marbles, coin, polling, ...

the statement ' $\nu = \mu$ ' is  
probably approximately correct (PAC)

## Hoeffding's Inequality (2/2)

$$\mathbb{P} [|\nu - \mu| > \epsilon] \leq \underline{2 \exp(-2\epsilon^2 N)}$$

- valid for all  $N$  and  $\epsilon$
- does not depend on  $\mu$ ,  
no need to 'know'  $\mu$
- larger sample size  $N$  or  
looser gap  $\epsilon$   
 $\Rightarrow$  higher probability for ' $\nu \approx \mu$ '



if **large  $N$** , can **probably** infer  
unknown  $\mu$  by known  $\nu$

# Fun Time

Let  $\mu = 0.4$ . Use Hoeffding's Inequality

$$\mathbb{P} [|\nu - \mu| > \epsilon] \leq 2 \exp(-2\epsilon^2 N)$$

to bound the probability that a sample of 10 marbles will have  $\nu \leq 0.1$ . What bound do you get?

- ① 0.67
- ② 0.40
- ③ 0.33
- ④ 0.05

## Fun Time

$v \leq 0.1$  and we take 10 marbles, it means the probability of taking 1 green marble ( $v = 0.1$ ) and taking zero green ball ( $v = 0$ )

Let  $\mu = 0.4$ . Use Hoeffding's Inequality

$$\mathbb{P}[|\nu - \mu| > \epsilon] \leq 2 \exp(-2\epsilon^2 N)$$

to bound the probability that a sample of 10 marbles will have  $\nu \leq 0.1$ . What bound do you get?

- ① 0.67
- ② 0.40
- ③ 0.33
- ④ 0.05

$$C_1^{10} \cdot 0.4^1 \cdot C_9^9 \cdot 0.6^9 + C_0^{10} \cdot 0.4^0 \cdot C_{10}^{10} \cdot 0.6^{10}$$

orange " green

0.04636

overestimate (because it's upper bound)



$\therefore \mu = 0.4$  and  $v \leq 0.1$   
 $\therefore$  We can set  $\epsilon$  to 0.3

Reference Answer: ③

Set  $N = 10$  and  $\epsilon = 0.3$  and you get the answer. BTW, ④ is the actual probability and Hoeffding gives only an upper bound to that.

$$2e^{-2 \cdot (0.3)^2 \cdot 10} \approx 0.33$$

# Connection to Learning

## bin

- unknown **orange** prob.  $\mu$
- marble  $\bullet \in \text{bin}$
- **orange**  $\bullet$
- **green**  $\bullet$
- size- $N$  sample from bin

of i.i.d. marbles

## learning

- fixed hypothesis  $h(\mathbf{x}) \stackrel{?}{=} \text{target } f(\mathbf{x})$
- $\mathbf{x} \in \mathcal{X}$
- $h$  is **wrong**  $\Leftrightarrow h(\mathbf{x}) \neq f(\mathbf{x})$
- $h$  is **right**  $\Leftrightarrow h(\mathbf{x}) = f(\mathbf{x})$
- check  $h$  on  $\mathcal{D} = \{(\mathbf{x}_n, \underbrace{y_n}_{f(\mathbf{x}_n)})\}$

with i.i.d.  $\mathbf{x}_n$

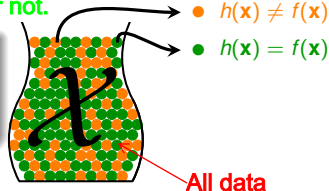
**sample**

**(training data)**

All marbles mean all data, and the vase means the  $h$  we chose.

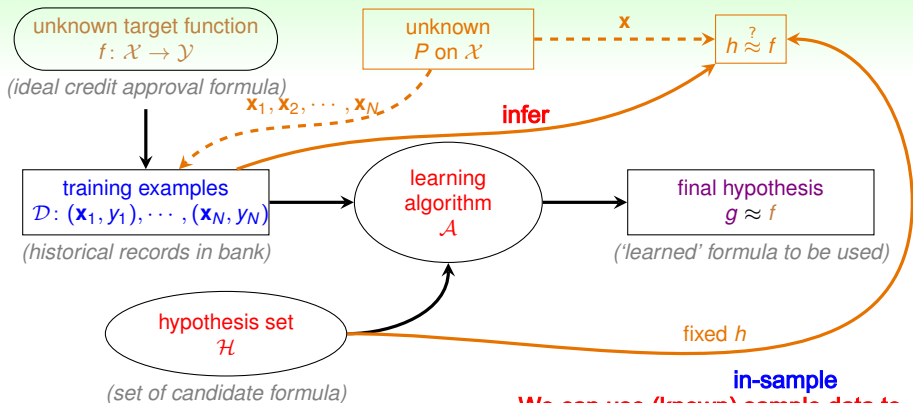
The color of marble means the result of  $h(\mathbf{x})$  is correct or not.

if **large**  $N$  & **i.i.d.**  $\mathbf{x}_n$ , can **probably** infer  
unknown  $\llbracket h(\mathbf{x}) \neq f(\mathbf{x}) \rrbracket$  probability  
by known  $\llbracket h(\mathbf{x}_n) \neq y_n \rrbracket$  fraction





# Added Components



for any fixed  $h$ , can probably infer

$$\text{unknown } E_{\text{out}}(h) = \mathcal{E}_{\mathbf{x} \sim P} [h(\mathbf{x}) \neq f(\mathbf{x})]$$

$$\text{by known } E_{\text{in}}(h) = \frac{1}{N} \sum_{n=1}^N [h(\mathbf{x}_n) \neq y_n].$$

**in-sample**  
We can use known sample data to  
infer unknown the whole data.

**out-of-sample**

# The Formal Guarantee

for any fixed  $h$ , in ‘big’ data ( $N$  large),

in-sample error  $E_{\text{in}}(h)$  is probably close to

out-of-sample error  $E_{\text{out}}(h)$  (within  $\epsilon$ )

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

same as the ‘bin’ analogy ...

- valid for all  $N$  and  $\epsilon$
- does not depend on  $E_{\text{out}}(h)$ , **no need to ‘know’**  $E_{\text{out}}(h)$   
—  $f$  and  $P$  can stay unknown
- ‘ $E_{\text{in}}(h) = E_{\text{out}}(h)$ ’ is **probably approximately correct (PAC)**

We have a model with good results.

This model should  
also do well on  
unseen data.

if ‘ $E_{\text{in}}(h) \approx E_{\text{out}}(h)$ ’ and ‘ $E_{\text{in}}(h)$  **small**’  
 $\implies \underline{E_{\text{out}}(h) \text{ small}} \implies h \approx f$  with respect to  $P$

# Verification of One $h$

for any fixed  $h$ , when data large enough,

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

$h$ : one model in  $H$

$g$ : a well-trained model in  $H$

Can we claim 'good learning' ( $g \approx f$ )?

$E_{\text{in}}(h)$  needs to be very small.

Yes!

if  $E_{\text{in}}(h)$  small for the fixed  $h$   
and  $\mathcal{A}$  pick the  $h$  as  $g$   
 $\Rightarrow$  ' $g = f$ ' PAC

No!

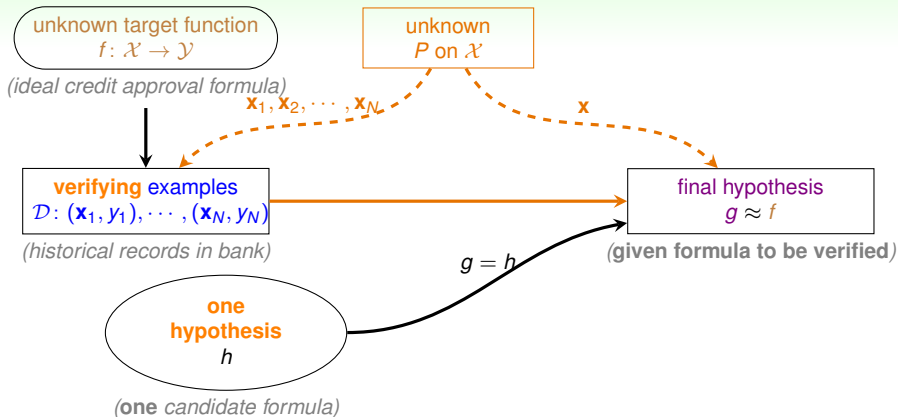
if  $\mathcal{A}$  forced to pick THE  $h$  as  $g$   
 $\Rightarrow E_{\text{in}}(h)$  almost always not small  
 $\Rightarrow$  ' $g \neq f$ ' PAC!

real learning:

$\mathcal{A}$  shall **make choices**  $\in \mathcal{H}$  (like PLA)  
rather than **being forced to pick one  $h$** . :-)

If algorithm  $\mathcal{A}$  always gives us the same  $h$  regardless of different sample data, it is NOT machine "learning"!

# The 'Verification' Flow



can now use 'historical records' (data) to  
**verify 'one candidate formula'  $h$**

Given  $h$  ( $\equiv$  we don't have  $A$  to choose the  $g$  from  $H$ ), we can verify this  $h$ 's performance.

$\Rightarrow$  This is not learning but verification.

# Fun Time

Your friend tells you her secret rule in investing in a particular stock: 'Whenever the stock goes down in the morning, it will go up in the afternoon; vice versa.' **To verify the rule, you chose 100 days uniformly at random from the past 10 years of stock data, and found that 80 of them satisfy the rule.** What is the best guarantee that you can get from the verification?

- 1 You'll definitely be rich by exploiting the rule in the next 100 days.
- 2 You'll likely be rich by exploiting the rule in the next 100 days, if the market behaves similarly to the last 10 years.
- 3 You'll likely be rich by exploiting the 'best rule' from 20 more friends in the next 100 days.
- 4 You'd definitely have been rich if you had exploited the rule in the past 10 years.

# Fun Time

Your friend tells you her secret rule in investing in a particular stock: 'Whenever the stock goes down in the morning, it will go up in the afternoon; vice versa.' **To verify the rule, you chose 100 days uniformly at random from the past 10 years of stock data, and found that 80 of them satisfy the rule.** What is the best guarantee that you can get from the verification?

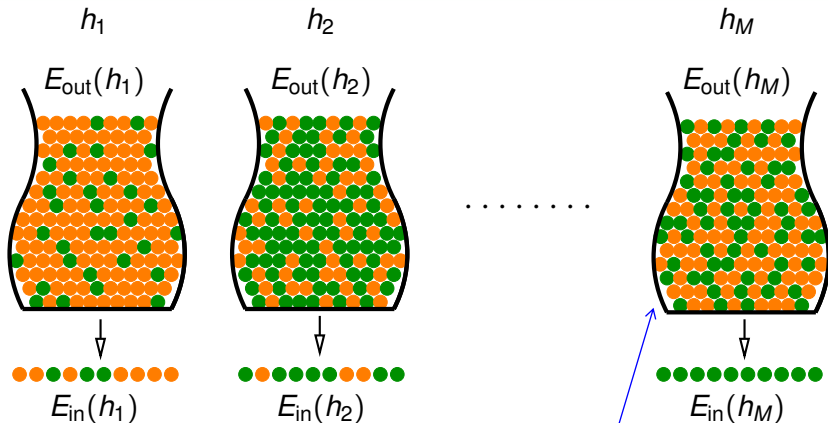
- ① You'll definitely be rich by exploiting the rule in the next 100 days.
- ② You'll likely be rich by exploiting the rule in the next 100 days, if the market behaves similarly to the last 10 years.
- ③ You'll likely be rich by exploiting the 'best rule' from 20 more friends in the next 100 days.
- ④ You'd definitely have been rich if you had exploited the rule in the past 10 years.

Reference Answer: ②

①: no free lunch; ③: no 'learning' guarantee in verification; ④: verifying with only 100 days, possible that the rule is mostly wrong for whole 10 years.

Multiple  $h$ 

top

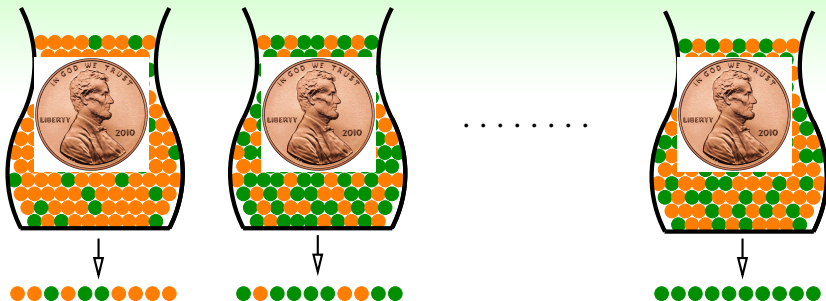
orange: wrong  
green: correct

real learning (say like PLA):

**BINGO** when getting .....?

It's actually a bad hypothesis.

## Coin Game



or flip 150 times by yourself

Q: if everyone in size-150 NTU ML class flips a coin 5 times, and **one of the students gets 5 heads for her coin 'g'**. Is 'g' really magical?

A: No. Even if all coins are fair, the probability that **one of the coins** results in **5 heads** is  $1 - \left(\frac{31}{32}\right)^{150} > 99\%$ . If we had so many vases (h), the possibility of getting a " $E_{in}$  that is a very small one" is very high.

extremely low  
possibility  
in average!

→ **BAD sample:**  $E_{in}$  and  $E_{out}$  far away  
—can get **worse** when involving 'choice'

Toss 1 time

↓  
Toss 150 times

The possibility of getting a bad sample becomes way more larger.



# BAD Sample and BAD Data

## BAD Sample

e.g.,  $E_{\text{out}} = \frac{1}{2}$ , but getting all heads ( $E_{\text{in}} = 0$ )!

## BAD Data for One $h$

$E_{\text{out}}(h)$  and  $E_{\text{in}}(h)$  far away:

e.g.,  $E_{\text{out}}$  big (far from  $f$ ), but  $E_{\text{in}}$  small (correct on most examples)

one bundle of data

	$\mathcal{D}_1$	$\mathcal{D}_2$	...	$\mathcal{D}_{1126}$	...	$\mathcal{D}_{5678}$	...	Hoeffding
$h$	<b>BAD</b>	GOOD		GOOD		<b>BAD</b>	GOOD	$\mathbb{P}_{\mathcal{D}} [\text{BAD } \mathcal{D} \text{ for } h] \leq \dots$

When we grab a small amount of data every time, the chance of catching a handful of bad data is very small.

Hoeffding: small

For one  $h$

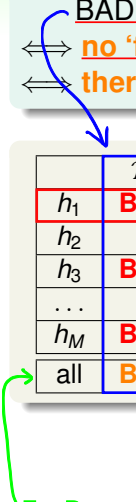
$$\mathbb{P}_{\mathcal{D}} [\text{BAD } \mathcal{D}] = \sum_{\text{all possible } \mathcal{D}} \mathbb{P}(\mathcal{D}) \cdot \llbracket \text{BAD } \mathcal{D} \rrbracket$$

# BAD Data for Many $h$

BAD data for many  $h$

$\Leftrightarrow$  no 'freedom of choice' by  $\mathcal{A}$

$\Leftrightarrow$  there exists some  $h$  such that  $E_{\text{out}}(h)$  and  $E_{\text{in}}(h)$  far away



	$\mathcal{D}_1$	$\mathcal{D}_2$	...	$\mathcal{D}_{1126}$	...	$\mathcal{D}_{5678}$	Hoeffding
$h_1$	<b>BAD</b>					<b>BAD</b>	$\mathbb{P}_{\mathcal{D}} [\text{BAD } \mathcal{D} \text{ for } h_1] \leq \dots$
$h_2$		<b>BAD</b>					$\mathbb{P}_{\mathcal{D}} [\text{BAD } \mathcal{D} \text{ for } h_2] \leq \dots$
$h_3$	<b>BAD</b>	<b>BAD</b>				<b>BAD</b>	$\mathbb{P}_{\mathcal{D}} [\text{BAD } \mathcal{D} \text{ for } h_3] \leq \dots$
...							
$h_M$	<b>BAD</b>					<b>BAD</b>	$\mathbb{P}_{\mathcal{D}} [\text{BAD } \mathcal{D} \text{ for } h_M] \leq \dots$
all	<b>BAD</b>	<b>BAD</b>				<b>BAD</b>	?

Good data

for  $M$  hypotheses, bound of  $\mathbb{P}_{\mathcal{D}} [\text{BAD } \mathcal{D}]$ ?

For  $\mathcal{D}_i$  :  
all =  $h_1$  or  $h_2$  or ... or  $h_m$

## Bound of BAD Data

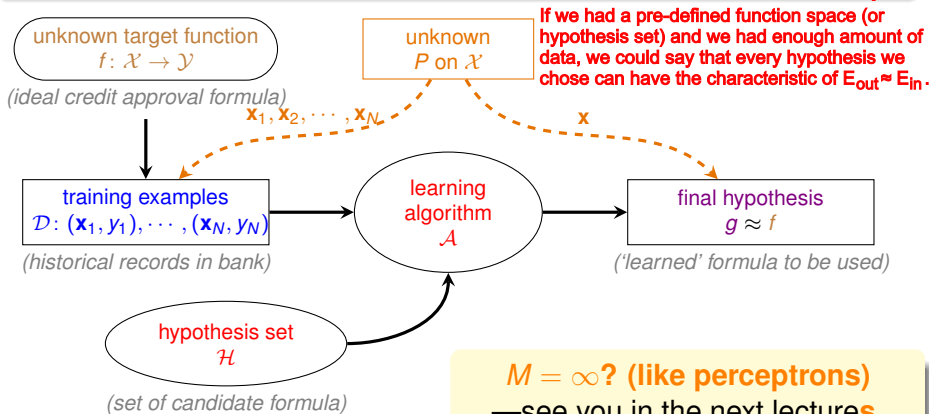
$$\begin{aligned}
& \mathbb{P}_{\mathcal{D}}[\text{BAD } \mathcal{D}] \\
&= \mathbb{P}_{\mathcal{D}}[\text{BAD } \mathcal{D} \text{ for } h_1 \text{ or BAD } \mathcal{D} \text{ for } h_2 \text{ or } \dots \text{ or BAD } \mathcal{D} \text{ for } h_M] \\
&\leq \mathbb{P}_{\mathcal{D}}[\text{BAD } \mathcal{D} \text{ for } h_1] + \mathbb{P}_{\mathcal{D}}[\text{BAD } \mathcal{D} \text{ for } h_2] + \dots + \mathbb{P}_{\mathcal{D}}[\text{BAD } \mathcal{D} \text{ for } h_M] \\
&\quad (\text{union bound}) \quad \text{worse case: not over-lapping} \\
&\leq 2 \exp(-2\epsilon^2 N) + 2 \exp(-2\epsilon^2 N) + \dots + 2 \exp(-2\epsilon^2 N) \\
&= 2M \exp(-2\epsilon^2 N) \quad M \searrow, N \nearrow \Rightarrow P \searrow
\end{aligned}$$

- finite-bin version of Hoeffding, valid for all  $M$ ,  $N$  and  $\epsilon$
- does not depend on any  $E_{\text{out}}(h_m)$ , no need to 'know'  $E_{\text{out}}(h_m)$   
—  $f$  and  $P$  can stay unknown
- ' $E_{\text{in}}(g) = E_{\text{out}}(g)$ ' is **PAC**, regardless of  $\mathcal{A}$

'most reasonable'  $\mathcal{A}$  (like PLA/pocket):  
pick the  $h_m$  with **lowest**  $E_{\text{in}}(h_m)$  as  $g$

# The 'Statistical' Learning Flow

if  $|\mathcal{H}| = M$  finite,  $N$  large enough,  
 for whatever  $g$  picked by  $\mathcal{A}$ ,  $E_{\text{out}}(g) \approx E_{\text{in}}(g)$   
 if  $\mathcal{A}$  finds one  $g$  with  $E_{\text{in}}(g) \approx 0$ ,  
 PAC guarantee for  $E_{\text{out}}(g) \approx 0 \Rightarrow$  **learning possible :-)**



# Fun Time

Consider 4 hypotheses.

$$h_1(\mathbf{x}) = \text{sign}(x_1), \quad h_2(\mathbf{x}) = \text{sign}(x_2),$$

$$h_3(\mathbf{x}) = \text{sign}(-x_1), \quad h_4(\mathbf{x}) = \text{sign}(-x_2).$$

For any  $N$  and  $\epsilon$ , which of the following statement is not true?

- ① the **BAD** data of  $h_1$  and the **BAD** data of  $h_2$  are exactly the same
- ② the **BAD** data of  $h_1$  and the **BAD** data of  $h_3$  are exactly the same
- ③  $\mathbb{P}_{\mathcal{D}}[\text{BAD for some } h_k] \leq 8 \exp(-2\epsilon^2 N)$
- ④  $\mathbb{P}_{\mathcal{D}}[\text{BAD for some } h_k] \leq 4 \exp(-2\epsilon^2 N)$

## Fun Time

Consider 4 hypotheses.

$$h_1(\mathbf{x}) = \text{sign}(x_1), \quad h_2(\mathbf{x}) = \text{sign}(x_2),$$

$$h_3(\mathbf{x}) = \text{sign}(-x_1), \quad h_4(\mathbf{x}) = \text{sign}(-x_2).$$

For any  $N$  and  $\epsilon$ , which of the following statement is not true?

- ① the **BAD** data of  $h_1$  and the **BAD** data of  $h_2$  are exactly the same
- ② the **BAD** data of  $h_1$  and the **BAD** data of  $h_3$  are exactly the same
- ③  $\mathbb{P}_{\mathcal{D}}[\text{BAD for some } h_k] \leq 8 \exp(-2\epsilon^2 N)$
- ④  $\mathbb{P}_{\mathcal{D}}[\text{BAD for some } h_k] \leq 4 \exp(-2\epsilon^2 N)$

Reference Answer: ①

The important thing is to note that ② is true, which implies that ④ is true if you revisit the union bound. Similar ideas will be used to conquer the  $M = \infty$  case.

# Summary

## 1 When Can Machines Learn?

Lecture 3: Types of Learning

Lecture 4: Feasibility of Learning

- Learning is Impossible?  
**absolutely no free lunch outside  $\mathcal{D}$**
- Probability to the Rescue  
**probably approximately correct outside  $\mathcal{D}$**
- Connection to Learning  
**verification possible if  $E_{\text{in}}(h)$  small for fixed  $h$**
- Connection to Real Learning  
**learning possible if  $|\mathcal{H}|$  finite and  $E_{\text{in}}(g)$  small**

## 2 Why Can Machines Learn?

- **next: what if  $|\mathcal{H}| = \infty$ ?**

## 3 How Can Machines Learn?

## 4 How Can Machines Learn Better?