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

2. **Why Can Machines Learn?**

3. **How Can Machines Learn?**

4. **How Can Machines Learn Better?**

   **Lecture 4: Feasibility of Learning**
   - Learning is Impossible?
   - Probability to the Rescue
   - Connection to Learning
   - Connection to Real Learning
Feasibility of Learning

Learning is Impossible?

A Learning Puzzle

\[
\begin{align*}
    y_n &= -1 \\
    y_n &= +1 \\
    g(x) &= ?
\end{align*}
\]

let’s test your ‘human learning’ with 6 examples :-)
Two Controversial Answers

whenever you say about $g(x)$,

\[ \begin{align*}
&yn = -1 \\
&yn = +1 \\
g(x) = ?
\end{align*} \]

truth $f(x) = +1$ because ...

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

truth $f(x) = -1$ because ...

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

all valid reasons, your adversarial teacher can always call you ‘didn’t learn’. :-(

Hsuan-Tien Lin (NTU CSIE)
A ‘Simple’ Binary Classification Problem

<table>
<thead>
<tr>
<th>$x_n$</th>
<th>$y_n = f(x_n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>○</td>
</tr>
<tr>
<td>0 0 1</td>
<td>×</td>
</tr>
<tr>
<td>0 1 0</td>
<td>×</td>
</tr>
<tr>
<td>0 1 1</td>
<td>○</td>
</tr>
<tr>
<td>1 0 0</td>
<td>×</td>
</tr>
</tbody>
</table>

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

pick $g \in \mathcal{H}$ with all $g(x_n) = y_n$ (like PLA),

**does** $g \approx f$?
No Free Lunch

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>g</th>
<th>f₁</th>
<th>f₂</th>
<th>f₃</th>
<th>f₄</th>
<th>f₅</th>
<th>f₆</th>
<th>f₇</th>
<th>f₈</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>0 0 1</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>0 1 0</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>0 1 1</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>1 0 0</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>1 0 1</td>
<td>?</td>
<td>?</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>○</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>1 1 0</td>
<td>?</td>
<td>?</td>
<td>○</td>
<td>○</td>
<td>×</td>
<td>×</td>
<td>○</td>
<td>○</td>
<td>×</td>
<td>×</td>
</tr>
<tr>
<td>1 1 1</td>
<td>?</td>
<td>?</td>
<td>○</td>
<td>×</td>
<td>○</td>
<td>×</td>
<td>○</td>
<td>×</td>
<td>○</td>
<td>×</td>
</tr>
</tbody>
</table>

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

learning from $\mathcal{D}$ (to infer something outside $\mathcal{D}$) is doomed if any ‘unknown’ $f$ can happen. :-(
This is a popular ‘brain-storming’ problem, with a claim that 2% of the world’s cleverest population can crack its ‘hidden pattern’.

\[(5, 3, 2) \to 151022, \quad (7, 2, 5) \to ?\]

It is like a ‘learning problem’ with \(N = 1, \mathbf{x}_1 = (5, 3, 2), \mathbf{y}_1 = 151022\). 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

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?
Statistics 101: Inferring Orange Probability

Assume:
- Orange probability = $\mu$,
- Green probability = $1 - \mu$,
with $\mu$ unknown.

Sample:
- $N$ marbles sampled independently, with
  - Orange 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 $\epsilon$)

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

- called Hoeffding’s Inequality, for marbles, coin, polling, ...
Hoeffding’s Inequality (2/2)

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

- valid for all \( N \) and \( \epsilon \)
- does not depend on \( \mu \), no need to ‘know’ \( \mu \)
- larger sample size \( N \) or looser gap \( \epsilon \)
  \( \longrightarrow \) higher probability for ‘\( \nu \approx \mu \)’

if large \( N \), can probably infer unknown \( \mu \) by known \( \nu \)
Feasibility of Learning

Fun Time

Let \( \mu = 0.4 \). Use Hoeffding’s Inequality

\[
P \left[ |\nu - \mu| > \epsilon \right] \leq 2 \exp \left( -2 \epsilon^2 N \right)
\]

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

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.
Feasibility of Learning

Connection to Learning

**bin**
- unknown orange prob. $\mu$
- marble $\in$ bin
- orange
- green
- size-$N$ sample from bin of i.i.d. marbles

**learning**
- fixed hypothesis $h(\mathbf{x}) \equiv$ target $f(\mathbf{x})$
- $\mathbf{x} \in \mathcal{X}$
- $h$ is wrong $\iff h(\mathbf{x}) \neq f(\mathbf{x})$
- $h$ is right $\iff h(\mathbf{x}) = f(\mathbf{x})$
- check $h$ on $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}$ with i.i.d. $\mathbf{x}_n$

If large $N$ & i.i.d. $\mathbf{x}_n$, can probably infer unknown $[h(\mathbf{x}) \neq f(\mathbf{x})]$ probability by known $[h(\mathbf{x}_n) \neq y_n]$ fraction
Feasibility of Learning

Connection to Learning

**Added Components**

- **unknown target function** $f: \mathcal{X} \to \mathcal{Y}$
- **training examples** $D: (x_1, y_1), \cdots, (x_N, y_N)$
- **learning algorithm** $\mathcal{A}$
- **final hypothesis** $g \approx f$
- **hypothesis set** $\mathcal{H}$

For any fixed $h$, can probably infer

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

by **known** $E_{in}(h) = \frac{1}{N} \sum_{n=1}^{N} [h(x_n) \neq y_n]$. 
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 \( \epsilon \))

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

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)

if ‘\( E_{\text{in}}(h) \approx E_{\text{out}}(h) \)’ and ‘\( E_{\text{in}}(h) \) small’

\( \implies \) \( E_{\text{out}}(h) \) small \( \implies \) \( h \approx f \) with respect to \( P \)
Verification of One $h$

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

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

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

**Yes!**

if $E_{in}(h)$ small for the fixed $h$
and $\mathcal{A}$ pick the $h$ as $g$

$\implies$ ‘$g = f$’ PAC

**No!**

if $\mathcal{A}$ forced to pick THE $h$ as $g$

$\implies$ $E_{in}(h)$ almost always not small

$\implies$ ‘$g \neq f$’ PAC!

real learning:

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

unknown target function $f : \mathcal{X} \rightarrow \mathcal{Y}$

(ideal credit approval formula)

verifying examples $\mathcal{D} : (x_1, y_1), \ldots, (x_N, y_N)$

(historical records in bank)

unknown $P$ on $\mathcal{X}$

$x_1, x_2, \ldots, x_N$ $x$

one hypothesis $h$

(one candidate formula)

g = h

final hypothesis $g \approx f$

(given formula to be verified)

can now use ‘historical records’ (data) to verify ‘one candidate formula’ $h$
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 for whole 10 years.
Feasibility of Learning

Connection to Real Learning

Multiple $h$

real learning (say like PLA):

**BINGO** when getting ···················?
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 $1 - \left(\frac{31}{32}\right)^{150} > 99\%$.

BAD sample: $E_{\text{in}}$ and $E_{\text{out}}$ far away — can get worse when involving ‘choice’
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)

<table>
<thead>
<tr>
<th>$D_1$</th>
<th>$D_2$</th>
<th>$\ldots$</th>
<th>$D_{1126}$</th>
<th>$\ldots$</th>
<th>$D_{5678}$</th>
<th>$\ldots$</th>
<th>Hoeffding</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>BAD</td>
<td>$\ldots$</td>
<td>BAD</td>
<td>$\ldots$</td>
<td>BAD</td>
<td>$\ldots$</td>
<td>$P_D$ [BAD $D$ for $h$] $\leq \ldots$</td>
</tr>
</tbody>
</table>

Hoeffding: small

$$P_D [\text{BAD } D] = \sum_{\text{all possible } D} P(D) \cdot [\text{BAD } D]$$
BAD Data for Many $h$

BAD data for many $h$

$\iff$ no ‘freedom of choice’ by $A$

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

\[
\begin{array}{cccccc}
D_1 & D_2 & \ldots & D_{1126} & \ldots & D_{5678} \\
\hline
h_1 & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} \\
h_2 & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} \\
h_3 & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} \\
\ldots & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} \\
h_M & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} \\
\text{all} & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} & \text{BAD} \\
\end{array}
\]

Hoeffding

$P_D[\text{BAD } D \text{ for } h_1] \leq \ldots$

$P_D[\text{BAD } D \text{ for } h_2] \leq \ldots$

$P_D[\text{BAD } D \text{ for } h_3] \leq \ldots$

$P_D[\text{BAD } D \text{ for } h_M] \leq \ldots$

for $M$ hypotheses, bound of $P_D[\text{BAD } D]$?
Bound of BAD Data

\[ \mathbb{P}_D[\text{BAD } D] = \mathbb{P}_D[\text{BAD } D \text{ for } h_1 \text{ or BAD } D \text{ for } h_2 \text{ or } \ldots \text{ or BAD } D \text{ for } h_M] \leq \mathbb{P}_D[\text{BAD } D \text{ for } h_1] + \mathbb{P}_D[\text{BAD } D \text{ for } h_2] + \ldots + \mathbb{P}_D[\text{BAD } D \text{ for } h_M] \text{ (union bound)} \]

\[ \leq 2 \exp \left( -2\epsilon^2 N \right) + 2 \exp \left( -2\epsilon^2 N \right) + \ldots + 2 \exp \left( -2\epsilon^2 N \right) \]

\[ = 2M \exp \left( -2\epsilon^2 N \right) \]

- 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 \( g \)
Feasibility of Learning

Connection to Real Learning

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 $E_{\text{out}}(g) \approx 0 \implies \text{learning possible :-)}$

unknown target function
$f : \mathcal{X} \rightarrow \mathcal{Y}$
(ideal credit approval formula)

unknown $P$ on $\mathcal{X}$

training examples
$\mathcal{D} : (x_1, y_1), \cdots, (x_N, y_N)$
(historical records in bank)

learning algorithm $\mathcal{A}$

final hypothesis
$g \approx f$
(‘learned’ formula to be used)

hypothesis set
$\mathcal{H}$
(set of candidate formula)

$M = \infty$? (like perceptrons)
—see you in the next lectures
Consider 4 hypotheses.

\[ h_1(x) = \text{sign}(x_1), \quad h_2(x) = \text{sign}(x_2), \]
\[ h_3(x) = \text{sign}(-x_1), \quad h_4(x) = \text{sign}(-x_2). \]

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. \( \mathbb{P}_D[\text{BAD for some } h_k] \leq 8 \exp (-2\epsilon^2 N) \)
4. \( \mathbb{P}_D[\text{BAD for some } h_k] \leq 4 \exp (-2\epsilon^2 N) \)

**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 $E_{in}(h)$ small for fixed $h$
     - Connection to Real Learning 
       - Learning possible if $|\mathcal{H}|$ finite and $E_{in}(g)$ small

2. **Why Can Machines Learn?**
   - *next: what if* $|\mathcal{H}| = \infty$?

3. **How Can Machines Learn?**

4. **How Can Machines Learn Better?**