Feasibility of Learning (LFD 1.3)
A Learning Puzzle

Model 1: left-top black $\Rightarrow -1$, so $g(x) = -1$

Model 2: exists some symmetric pattern $\Rightarrow +1$, so $g(x) = +1$

Not enough information to tell us which one is correct

$y_n = -1$

$y_n = +1$

$g(x) = ?$
A Learning Puzzle

Model 1: left-top black $\Rightarrow -1$, so $g(x) = -1$

Model 2: exists some symmetric pattern $\Rightarrow +1$, so $g(x) = +1$

Not enough information to tell us which one is correct
Outside the Data Set

Another simple binary classification example

<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>

Can we “learn” from this data? (Can we “infer” something outside the training data?)
Learning process:
- Check all the possible functions
- Choose the one that fits all the data

Can’t make any prediction
- Learning from $\mathcal{D}$ (to infer something outside $\mathcal{D}$) is impossible if any $f$ can happen
Inferring Something Unknown

- Consider a bin with red and green marbles
  \[ P[\text{picking a red marble}] = \mu \]
  \[ P[\text{picking a green marble}] = 1 - \mu \]

- The value of \( \mu \) is unknown to us
- How to infer \( \mu \)?
  - Pick \( N \) marbles independently
  - \( \nu \): the fraction of red marbles
Inferring with probability

- Do you know $\mu$?
  - No
  - Sample can be mostly green while bin is mostly red

- Can you say something about $\mu$?
  - Yes
  - $\nu$ is “probably” close to $\mu$
Hoeffding’s Inequality

- In big sample (large $N$), $\nu$ (sample mean) is probably close to $\mu$:

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

This is called **Hoeffding’s inequality**

- The statement “$\mu = \nu$” is probably approximately correct (PAC)
Hoeffding’s Inequality

\[ P[|\nu - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N} \]

- Valid for all \( N \) and \( \epsilon > 0 \)
- Does not depend on \( \mu \) (no need to know \( \mu \))
- Larger sample size \( N \) or looser gap \( \epsilon \)
  \( \Rightarrow \) higher probability for \( \mu \approx \nu \)
Connection to Learning

How to connect this to learning?

- Each marble (uncolored) is a data point $x \in \mathcal{X}$
How to connect this to learning?

- Each marble (uncolored) is a data point $x \in X$
- **red ball**: $h(x) \neq f(x)$ (h is correct)
- **green ball**: $h(x) = f(x)$ (h is wrong)
For a particular function \( h \):

- \( E_{\text{out}}(h) = E_{x \sim P}[h(x) \neq f(x)] \) (out-of-sample, unknown)
  \( \iff \mu \)

- \( E_{\text{in}}(h) = \frac{1}{N} \sum_{n=1}^{N} [h(x_n) \neq y_n] \) (in-sample, known)
  \( \iff \nu \)

- Now we can infer \( E_{\text{out}}(h) \) from \( E_{\text{in}}(h) \)!
Verifying a Hypothesis

- For any \( h \), when sample size \((N)\) is large:

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

- “\( E_{in}(h) = E_{out}(h) \)” is probably approximately correct (PAC)

- How is this useful?
Verifying a Hypothesis

For any $h$, when sample size ($N$) is large:

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

“$E_{in}(h) = E_{out}(h)$” is probably approximately correct (PAC)

How is this useful?

If $E_{in}(h) \approx E_{out}(h)$ and $E_{in}(h)$ is small
\[ \Rightarrow E_{out}(h) \text{ small} \]
\[ \Rightarrow h \approx f \text{ with respect to } P \]
Verifying a Hypothesis

- For any \( h \), when sample size \( (N) \) is large:

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

- “\( E_{\text{in}}(h) = E_{\text{out}}(h) \)” is probably approximately correct (PAC)
- How is this useful?
- If \( E_{\text{in}}(h) \approx E_{\text{out}}(h) \) and \( E_{\text{in}}(h) \) is small
  \[ \Rightarrow E_{\text{out}}(h) \text{ small} \]
  \[ \Rightarrow h \approx f \text{ with respect to } P \]
- **Given a hypothesis** \( h \Rightarrow \) sample \( N \) data \( \Rightarrow E_{\text{in}}(h) \) to “verify” the quality of \( h \)
- Can we apply to multiple hypothesis?
Apply to multiple bins (hypothesis)

Color in each bin depends on different hypothesis

**Bingo** when getting all **green** balls?
If you have 150 fair coins, flip each coin 5 times, and one of them gets 5 heads. Is this coin \((g)\) special?

\[
1 - \left(\frac{31}{32}\right)^{150} > 99\% 
\]

Because: there can exist some \(h\) such that \(E_{in}\) and \(E_{out}\) are far away if \(M\) is large.
If you have 150 fair coins, flip each coin 5 times, and one of them gets 5 heads. Is this coin (g) special?

No. The probability of existing one of the coin results in 5 heads is $1 - \left(\frac{31}{32}\right)^{150} > 99\%$
If you have 150 fair coins, flip each coin 5 times, and one of them gets 5 heads. Is this coin \((g)\) special?

No. The probability of existing one of the coin results in 5 heads is 
\[ 1 - \left(\frac{31}{32}\right)^{150} > 99\% \]

Because: there can exist some \(h\) such that \(E_{in}\) and \(E_{out}\) are far away if \(M\) is large.
A Simple Solution

- For each particular $h$,

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

- We want a “union bound”:

$$P \left[ |E_{in}(h_1) - E_{out}(h_1)| > \epsilon \text{ or } \cdots \text{ or } |E_{in}(h_M) - E_{out}(h_M)| > \epsilon \right]$$

$$\leq \sum_{m=1}^{M} P \left[ |E_{in}(h_m) - E_{out}(h_m)| \right] \leq 2Me^{-2\epsilon^2 N}$$

- So, $P \left[ |E_{in}(g) - E_{out}(g)| > \epsilon \right] \leq 2Me^{-2\epsilon^2 N}$
When is learning successful?

When our Learning Algorithm $\mathcal{A}$ picks the hypothesis $g$:

$$P[|E_{in}(g) - E_{out}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2N}$$

- If $M$ is small and $N$ is large enough:
  - If $\mathcal{A}$ finds $E_{in}(g) \approx 0$
  - $\Rightarrow E_{out}(g) \approx 0$ (Learning is successful!)
Feasibility of Learning

\[ P[|E_{in}(g) - E_{out}(g)| > \epsilon] \leq 2M e^{-2\epsilon^2 N} \]

Two questions:

(1) Can we make sure \( E_{out}(g) \approx E_{in}(g) \)?

(2) Can we make sure \( E_{in}(g) \approx 0 \)?

\( M \): complexity of model

Small \( M \): (1) holds, but (2) may not hold (too few choices)

Large \( M \): (1) doesn't hold, but (2) may hold (over-fitting)

How to handle infinite \( M \)? (will discuss in next lectures)
Feasibility of Learning

\[ P[|E_{in}(g) - E_{out}(g)| > \epsilon] \leq 2M e^{-2\epsilon^2 N} \]

Two questions:

1. Can we make sure \( E_{out}(g) \approx E_{in}(g) \)?
2. Can we make sure \( E_{in}(g) \approx 0 \)?

\( M \): complexity of model

- Small \( M \): (1) holds, but (2) may not hold (too few choices)
  - (under-fitting)
Feasibility of Learning

\[ P[|E_{in}(g) - E_{out}(g)| > \epsilon] \leq 2Me^{-2\epsilon^2N} \]

Two questions:

1. Can we make sure \( E_{out}(g) \approx E_{in}(g) \)?
2. Can we make sure \( E_{in}(g) \approx 0 \)?

\( M \): complexity of model

- Small \( M \): (1) holds, but (2) may not hold (too few choices) (under-fitting)
- Large \( M \): (1) doesn’t hold, but (2) may hold (over-fitting)

How to handle infinite \( M \)? (will discuss in next lectures)
Feasibility of Learning

\[ P[|E_{in}(g) - E_{out}(g)| > \epsilon] \leq 2M e^{-2\epsilon^2 N} \]

Two questions:

1. Can we make sure \( E_{out}(g) \approx E_{in}(g) \)?
2. Can we make sure \( E_{in}(g) \approx 0 \)?

\( M \): complexity of model

- Small \( M \): (1) holds, but (2) may not hold (too few choices) (under-fitting)
- Large \( M \): (1) doesn’t hold, but (2) may hold (over-fitting)

How to handle infinite \( M \)? (will discuss in next lectures)
More on Perceptron (LFD 3.1)
Feature Engineering

- Choosing the “good hypothesis space” is important!
- Feature engineering ⇒ how to select good features?
- Each dimension represents some “sophisticated physical meaning”
Raw features: digit recognition

- Raw features: $16 \times 16 = 256$ pixels
  \[ (\mathbf{x} \in \mathbb{R}^d, \; d = 256) \]
- Perception: need to learn $w_1, \ldots, w_{256}$
  \[ \Rightarrow \text{difficult to learn} \]
Design better features

- For Example: $x = (x_1, x_2)$
  - $x_1$: intensity
  - $x_2$: symmetry (intensity of [image - flipped image])
PLA vs Pocket

- PLA: perceptron learning algorithm
- Pocket: always remember the $\mathbf{w}$ with the best $E_{in}(\mathbf{w})$ (training error) (better for non-separable case)
Conclusions

- When is learning feasible?
  - Training and testing follow the same distribution
  - Size of hypothesis space:
    - Too large $\Rightarrow$ small training error but cannot control test error
    - Too small $\Rightarrow$ training error (may be) large
    - Better features $\Rightarrow$ small hypothesis space and low training error

- Next class: LFD 3.2, 3.3, 3.4 (linear models)

Questions?