寶可夢、數碼寶貝分類器
淺談機器學習原理
Step 1: function with unknown

Step 2: define loss

Step 3: optimization

Review: Basic Idea of ML

https://youtu.be/Ye018rCVvOo

https://youtu.be/bHcJCp2Fyx5s
Review: Strategy

More parameters, easier to overfit. Why?

https://youtu.be/WeHM2xpYQpw
Case Study: Pokémon v.s. Digimon

https://medium.com/@tyreeoestevenson/teaching-a-computer-to-classify-anime-8c77bc89b881
Pokémon vs. Digimon

這是數碼寶貝的蟲蟲獸

這才是寶可夢的綠毛蟲
Pokémon vs. Digimon

小智身邊有小火龍

太一身邊有亞古獸
Pokémon/Digimon Classifier

- We want to find a function......

\[ f(\text{Pokémon or Digimon}) = \text{Determine a function with unknown parameters (based on domain knowledge)} \]
Observation

**Digimon**

**Pokémon**

線條較複雜？

線條較簡單？
Observation

Edge detection

\[ e(\cdot) = 3558 \]

\[ e(\cdot) = 7389 \]
Function with Unknown Parameters

\[ f(\ ) = \begin{cases} \text{Digimon} & \text{if } e(\ ) \geq h \\ \text{Pokémon} & \text{if } e(\ ) < h \end{cases} \]

\( f_h \): function with threshold \( h \)

\( \mathcal{H} = \{1, 2, \ldots, 10,000\} \)

\(|\mathcal{H}|\): number of candidate functions (model “complexity”)

\( f_h \): function with threshold \( h \)
Loss of a function (given data)

• Given a dataset \( \mathcal{D} \)

\[
\mathcal{D} = \{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \ldots, (x^N, \hat{y}^N)\}
\]

• Loss of a threshold \( h \) given data set \( \mathcal{D} \)

\[
L(h, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} l(h, x^n, \hat{y}^n)
\]

If \( f_h(x^n) \neq \hat{y}^n \)

Output 1

Otherwise

Output 0

Don’t like it? Of course, you can choose cross-entropy. 😊
Training Examples

• If we can collect all Pokémons and Digimons in the universe $\mathcal{D}_{all}$, we can find the best threshold $h^{all}$

$$h^{all} = \arg\min_h L(h, \mathcal{D}_{all})$$

• We only collect some examples $\mathcal{D}_{train}$ from $\mathcal{D}_{all}$

$$\mathcal{D}_{train} = \{(x^1, \hat{y}^1), (x^2, \hat{y}^2), \ldots, (x^N, \hat{y}^N)\}$$

$$(x^n, \hat{y}^n) \sim \mathcal{D}_{all}$$ independently and identically distributed (i.i.d.)

$$h^{train} = \arg\min_h L(h, \mathcal{D}_{train})$$
Training Examples

• If we can collect all Pokémon and Digimons in the universe $\mathcal{D}_{all}$, we can find the best threshold $h^{all}$

$$h^{all} = \arg\min_{h} L(h, \mathcal{D}_{all})$$

• We only collect some examples $\mathcal{D}_{train}$ from $\mathcal{D}_{all}$

$$h^{train} = \arg\min_{h} L(h, \mathcal{D}_{train})$$

We hope $L(h^{train}, \mathcal{D}_{all})$ and $L(h^{all}, \mathcal{D}_{all})$ are close.
We hope $L(h^{train}, D_{all})$ and $L(h^{all}, D_{all})$ are close.

All Pokémon and Digimons we know as $D_{all}$

Pokémon: 819
Digimon: 971

In most applications, you cannot obtain $D_{all}$.

(Testing data $D_{test}$ as the proxy of $D_{all}$)

Source of Digimon:
https://github.com/mrok273/Qiita
Source of Pokémon:
https://www.kaggle.com/kvpratama/pokemon-images-dataset/data

$h^{all} = 4824$
$L(h^{all}, D_{all}) = 0.28$
We hope $L(h^{\text{train}}, D_{\text{all}})$ and $L(h^{\text{all}}, D_{\text{all}})$ are close.

Sample 200 Pokémons and Digimons as $D_{\text{train1}}$

$\begin{align*}
h^{\text{train1}} &= 4727 \\
L(h^{\text{train1}}, D_{\text{train1}}) &= 0.27 \\
\text{Even lower than } L(h^{\text{all}}, D_{\text{all}})?
\end{align*}$

All Pokémons and Digimons we know as $D_{\text{all}}$

$\begin{align*}
h^{\text{all}} &= 4824 \\
L(h^{\text{all}}, D_{\text{all}}) &= 0.28 \\
L(h^{\text{train1}}, D_{\text{all}}) &= 0.28
\end{align*}$
We hope $L(h^{train}, D_{all})$ and $L(h^{all}, D_{all})$ are close.

Sample 200 Pokémon and Digimons as $D_{train2}$

All Pokémon and Digimons we know as $D_{all}$

$h^{train2} = 3642$  
$L(h^{train2}, D_{train2}) = 0.20$

$h^{all} = 4824$  
$L(h^{all}, D_{all}) = 0.28$

$L(h^{train2}, D_{all}) = 0.37$
What do we want? \( L(h_{\text{train}}, D_{\text{train}}) \) can be smaller than \( L(h_{\text{all}}, D_{\text{all}}) \)

We want \( L(h_{\text{train}}, D_{\text{all}}) - L(h_{\text{all}}, D_{\text{all}}) \leq \delta \)

What kind of \( D_{\text{train}} \) fulfill it?

\[ \forall h \in \mathcal{H}, |L(h, D_{\text{train}}) - L(h, D_{\text{all}})| \leq \delta / 2 \]

\( D_{\text{train}} \) is a good proxy of \( D_{\text{all}} \) for evaluating loss \( L \) given any \( h \).
What do we want?

We want \( L(h^{\text{train}}, D_{\text{all}}) - L(h^{\text{all}}, D_{\text{all}}) \leq \delta \)

What kind of \( D_{\text{train}} \) fulfill it?

\[ \forall h \in \mathcal{H}, |L(h, D_{\text{train}}) - L(h, D_{\text{all}})| \leq \delta/2 \]

\[
L(h^{\text{train}}, D_{\text{all}}) \leq L(h^{\text{train}}, D_{\text{train}}) + \delta/2 \\
\leq L(h^{\text{all}}, D_{\text{train}}) + \delta/2 \\
\leq L(h^{\text{all}}, D_{\text{all}}) + \delta/2 + \delta/2 = L(h^{\text{all}}, D_{\text{all}}) + \delta
\]

\( h^{\text{train}} = \arg \min_h L(h, D_{\text{train}}) \)
What do we want?

We want $L(h_{\text{train}}, D_{\text{all}}) - L(h_{\text{all}}, D_{\text{all}}) \leq \delta$

What kind of $D_{\text{train}}$ fulfill it?

$\forall h \in \mathcal{H}, |L(h, D_{\text{train}}) - L(h, D_{\text{all}})| \leq \delta/2$

We want to sample good $D_{\text{train}}$ $\varepsilon = \delta/2$

$\forall h \in \mathcal{H}, |L(h, D_{\text{train}}) - L(h, D_{\text{all}})| \leq \varepsilon$

What is the probability of sampling bad $D_{\text{train}}$?
Very General!

• The following discussion is model-agnostic.
• In the following discussion, we don’t have assumption about data distribution.
• In the following discussion, we can use any loss function.
Probability of Failure

Each point is a training set.

- good $D_{train}$
- bad $D_{train}$

$D_{train1}$  $D_{train2}$

Each point is a training set.
Probability of Failure

Each point is a training set.

\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) \]
Probability of Failure

If a $\mathcal{D}_{\text{train}}$ is bad, at least one $h$ makes $|L(h, \mathcal{D}_{\text{train}}) - L(h, \mathcal{D}_{\text{all}})| > \varepsilon$

$P(\mathcal{D}_{\text{train}} \text{ is bad due to } h_1)$

$P(\mathcal{D}_{\text{train}} \text{ is bad due to } h_2)$

$p_3$

$h_1$

$h_2$
\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) = \bigcup_{h \in \mathcal{H}} P(\mathcal{D}_{\text{train}} \text{ is bad due to } h) \]
\[ \leq \sum_{h \in \mathcal{H}} P(\mathcal{D}_{\text{train}} \text{ is bad due to } h) \]
\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) = \bigcup_{h \in \mathcal{H}} P(\mathcal{D}_{\text{train}} \text{ is bad due to } h) \leq \sum_{h \in \mathcal{H}} P(\mathcal{D}_{\text{train}} \text{ is bad due to } h) \]

\[ |L(h, \mathcal{D}_{\text{train}}) - L(h, \mathcal{D}_{\text{all}})| > \varepsilon \quad L(h, \mathcal{D}) = \frac{1}{N} \sum_{n=1}^{N} l(h, x^n, \hat{y}^n) \]

Loss of an example \( l(h, x^n, \hat{y}^n) \)
\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) = \bigcup_{h \in \mathcal{H}} P(\mathcal{D}_{\text{train}} \text{ is bad due to } h) \leq \sum_{h \in \mathcal{H}} P(\mathcal{D}_{\text{train}} \text{ is bad due to } h) \]

Hoeffding’s Inequality:

\[ P(\mathcal{D}_{\text{train}} \text{ is bad due to } h) \leq 2 \exp(-2N\varepsilon^2) \]

- The range of loss \( L \) is \([0, 1]\)
- \( N \) is the number of examples in \( \mathcal{D}_{\text{train}} \)
How to make $P(\mathcal{D}_{\text{train}} \text{ is bad})$ smaller?

Larger $N$ and smaller $|\mathcal{H}|$
\[ P(D_{\text{train is bad}}) \leq |\mathcal{H}| \cdot 2^{\exp(-2N\varepsilon^2)} \]
\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) \leq |\mathcal{H}| \cdot 2^{\exp(-2N\varepsilon^2)} \]

Smaller |\mathcal{H}|
Example

\[ \mathcal{H} = \{1, 2, \ldots, 10,000 \} \]
\[ \mathcal{D}_{\text{train}} = \{ (x^1, \hat{y}^1), (x^2, \hat{y}^2), \ldots, (x^N, \hat{y}^N) \} \]
\[ \forall h \in \mathcal{H}, |L(h, \mathcal{D}_{\text{train}}) - L(h, \mathcal{D}_{\text{all}})| \leq \varepsilon \]

\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) \leq |\mathcal{H}| \cdot 2 \exp(-2N\varepsilon^2) \]

\[ |\mathcal{H}| = 10000, N = 100, \varepsilon = 0.1 \quad \text{Usually happen QQ} \]
\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) \leq 2707 \]

\[ |\mathcal{H}| = 10000, N = 500, \varepsilon = 0.1 \]
\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) \leq 0.91 \]

\[ |\mathcal{H}| = 10000, N = 1000, \varepsilon = 0.1 \]
\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) \leq 0.00004 \]
Example

\[ P(D_{\text{train}} \text{ is bad}) \leq |\mathcal{H}| \cdot 2 \exp(-2N\varepsilon^2) \]

If we want \( P(D_{\text{train}} \text{ is bad}) \leq \delta \)

How many training examples do we need?

\[ |\mathcal{H}| \cdot 2 \exp(-2N\varepsilon^2) \leq \delta \quad \Rightarrow \quad N \geq \frac{\log(2|\mathcal{H}|/\delta)}{2\varepsilon^2} \]

\[ |\mathcal{H}| = 10000, \delta = 0.1, \varepsilon = 0.1 \]

\[ \Rightarrow \quad N \geq 610 \]
Model Complexity

\[ P(\mathcal{D}_{\text{train}} \text{ is bad}) \leq |\mathcal{H}| \cdot 2^{\exp(-2N\varepsilon^2)} \]

The number of possible functions you can select

What if the parameters are continuous?

• **Answer 1**: Everything that happens in a computer is discrete. 😊

• **Answer 2**: VC-dimension (not this course)
Model Complexity

$$P(\mathcal{D}_{\text{train}} \text{ is bad}) \leq |\mathcal{H}| \cdot 2^{\exp(-2N\varepsilon^2)}$$

Why don’t we simply use a very small $|\mathcal{H}|$?

“$\mathcal{D}_{\text{train}}$ is good” means ...

$$\forall h \in \mathcal{H}, |L(h, \mathcal{D}_{\text{train}}) - L(h, \mathcal{D}_{\text{all}})| \leq \varepsilon$$

Larger loss

$$L(h^{\text{train}}, \mathcal{D}_{\text{all}}) - L(h^{\text{all}}, \mathcal{D}_{\text{all}}) \leq \delta$$

$$\varepsilon = \delta/2$$

$h^{\text{all}} = \arg \min_{h \in \mathcal{H}} L(h, \mathcal{D}_{\text{all}})$

fewer candidates
**Tradeoff of Model Complexity**

Larger $N$ and smaller $|\mathcal{H}|$ \(\Rightarrow\) \(L(h^{\text{train}}, D_{\text{all}}) - L(h^{\text{all}}, D_{\text{all}}) \leq \delta\)

Smaller $|\mathcal{H}|$ \(\Rightarrow\) Larger \(L(h^{\text{all}}, D_{\text{all}})\)

魚與熊掌可以兼得嗎？ Yes, Deep Learning.