LEARNING PROBLEM

Bùi Tiến Lên

01/09/2019

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

Contents

Notation

symbol	meaning
$a, b, c, N \ldots$	scalar number
$\boldsymbol{w}, \boldsymbol{v}, \boldsymbol{x}, \boldsymbol{y} \ldots$	column vector
$\boldsymbol{X}, \boldsymbol{Y} \ldots$	matrix
$\mathbb{R}$	set of real numbers
$\mathbb{Z}$	set of integer numbers
$\mathbb{N}$	set of natural numbers
$\mathbb{R}^D$	set of vectors
$\mathcal{X}, \mathcal{Y}, \ldots$	set
$\mathcal{A}$	algorithm

operator	meaning
$\boldsymbol{w}^\intercal$	transpose
$\boldsymbol{X}\boldsymbol{Y}$	matrix multiplication
$\boldsymbol{X}^{-1}$	inverse

Learning Components

Credit Approval

- Suppose that a bank receives thousands of credit card applications every day, and it wants to automate the process of evaluating them.
- Applicant information

age	23 years
gender	male
annual salary	\$30000
years in residence	1 year
years in job	1 year
current debt	\$15000
...	...

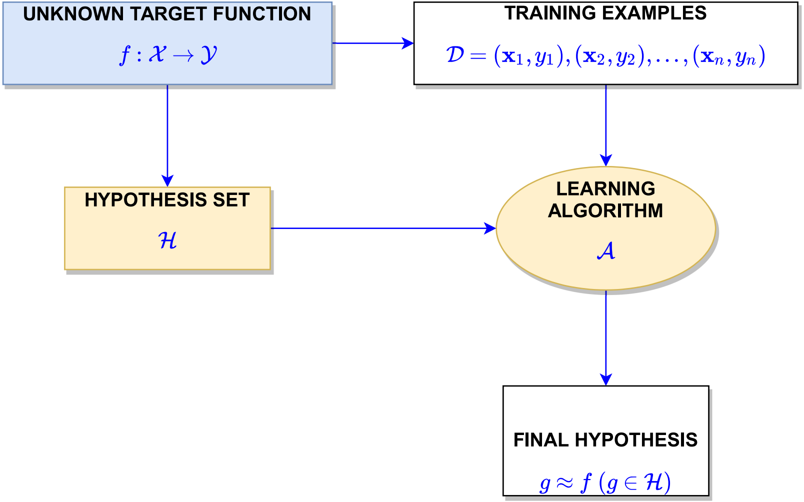
- Approve credit?

Problem Statement

Formalization

- Input: $\boldsymbol{x}$ (*customer application*)
- Output: y (*good/bad customer?* or $\{1, -1\}$)
- Data $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), ...(\boldsymbol{x}_N, y_N)$ (*historical records*)
- Target function: $f : \mathcal{X} \to \mathcal{Y}$ (*ideal credit approval formula*)
- Best approximate function $g : \mathcal{X} \to \mathcal{Y}$ (*formula to be used*)

Components of Learning

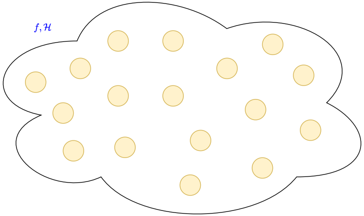


Solution components

The 2 solution components are referred as the **learning model**

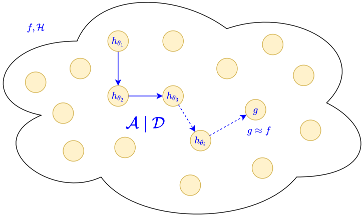
- The **hypothesis set** $\mathcal{H}$ built up from the problem

$$\mathcal{H} = \{h_{\theta_1}, h_{\theta_2}, \ldots\}$$



- The **learning algorithm** $\mathcal{A}$ is a **search algorithm** which finds $g \in \mathcal{H}$ such that

$$g \overset{best}{\approx} f$$



A Simple Learning Model

A Simple Hypothesis Set

We starts with the simple model (**the perceptron model**)

- For input $x = (x_1, ..., x_d)$ (*attributes of a customer*)

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

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

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

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

A Simple Hypothesis Set (cont.)

- Set $w_0 = -threshold$

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

- Introduce an artfificial coordinate $x_0 = 1$

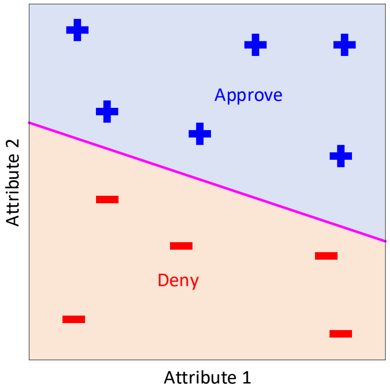
$$h(x) = h_{\mathbf{w}}(x) = sign\left(\sum_{i=0}^{d} w_i x_i\right)$$

- In vector form, the perceptron implements

$$h(x) = h_{\mathbf{w}}(x) = sign\left(\mathbf{w}^{\mathsf{T}} \mathbf{x}\right)$$

2D Model

- **Decision boundaries**: line
- **Decision regions**: approve and deny regions



A Simple Learning Algorithm

We uses the simple learning algorithm (**perceptron learning algorithm - PLA**) to implements

$$h(x) = h_{\mathbf{w}}(x) = sign\left(\mathbf{w}^{\intercal}\mathbf{x}\right)$$

- Given the training set

$$\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ...(\mathbf{x}_N, y_N)\}$$

- pick a *misclassified* point $(\mathbf{x}_i, y_i)$

$$sign(\mathbf{w}^{\intercal}\mathbf{x}_i) \neq y_i$$

- and update the weight vector

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

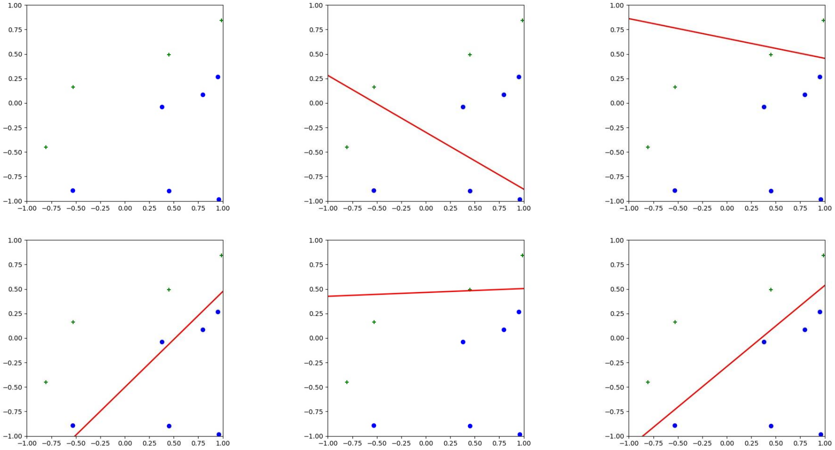
Iterations of PLA

- At iteration $t = 1, 2, 3, \ldots$ pick a misclassified point from

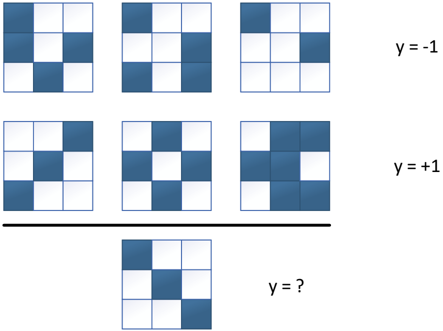
$$\mathcal{D} = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots (\boldsymbol{x}_N, y_N)\}$$

  and run a PLA iteration on it
- That's it

Is It Learning Algorithm?



A Learning Puzzle



Type of Learnings

Basic Premise of Learning

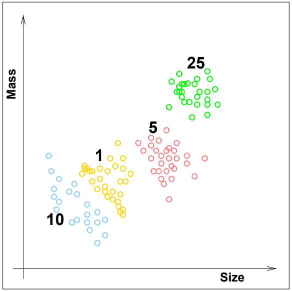
"using a set of observations to uncover an underlying process"

broad premise $\Longrightarrow$ many variations

- Supervise learning
- Unsupervised learning
- Reinforcement learning

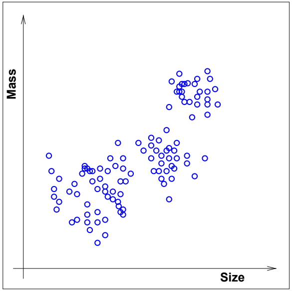
Supervised Learning

- We get data $\mathcal{D}$: (**input**, **correct ouput**)
  - When the **output** is one of *a finite set of values*, the learning problem is called **classification**
  - When the **output** is a *number*, the learning problem is called **regression**
- Example from vending machine - **coin classification**



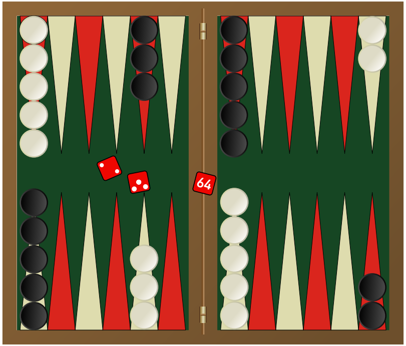
Unsupervised Learning

- Instead of (**input**, **correct input**), we get (**input**, ?)



Reinforcement Learning

- Instead of (**input**, **correct input**),
  we get (**input**, *some* **ouput**, **grade** *for this output*)



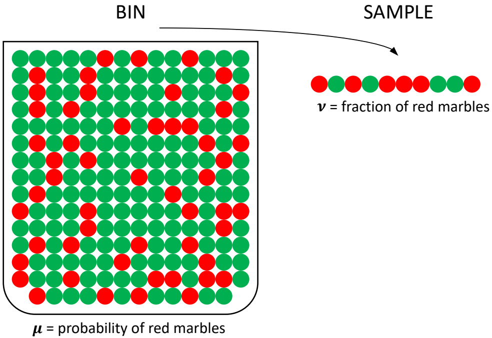
Feasibility Of Learning

A Related Experiment - Bin Problem

- 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 μ is unknown to us
- We pick N marbles independently
- The fraction of red marbles in **SAMPLE** $= \nu$



Does ν say anything about μ?

- **No!** (certain answer)
    - Sample can be mostly red while bin is mostly red
- **Yes!** (uncertain answer)
    - Sample frequency ν is likely close to bin frequency μ

What does ν say about μ?

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

$$P[|\nu - \mu| > \epsilon] \leq 2e^{-2\epsilon^2 N} \texttt{ for any } \epsilon > 0$$

  This is called **Hoeffding's Inequality**
- **Bound** does not depend on μ; tradeoff: N, ϵ and the bound
- We have

$$\nu \approx \mu \Longrightarrow \mu \approx \nu$$

- In other words, the statement "$\mu = \nu$" is **probably approximately correct** (P.A.C)

Connection to Learning

- **Bin problem**: The unknown is a number μ
- **Learning problem**: The unknown is a function $f : \mathcal{X} \to \mathcal{Y}$
- Each marble ● is a point $\boldsymbol{x} \in \mathcal{X}$

Bin problem	**Learning problem**
● (green)	hypothesis got it right $h(x) = f(x)$
● (red)	hypothesis got it wrong $h(x) \neq f(x)$

Connection to Learning (cont.)

- The error rate within the sample, which corresponds to ν in the bin model, will be called the *in-sample error*, (domain $\mathcal{D}$)

$$E_{in}(h) = \text{fraction of } \mathcal{D} \text{ where } f \text{ and } h \text{ disagre}$$
$$= \frac{1}{N} \sum_{n=1}^{N} [\![h(\mathbf{x}_n) \neq f(\mathbf{x}_n)]\!]$$

  where $[\![statement]\!] = 1$ if the statement is true, and $= 0$ if the statement is false
- In the same way, we define the *out-of-sample error* , (domain $\mathcal{X}$)

$$E_{out}(h) = P(h(\mathbf{x}) \neq f(\mathbf{x})), \mathbf{x} \in \mathcal{X}$$

  which corresponds to μ in the bin model.

Connection to Learning (cont.)

- The Hoeffding inequality becomes:

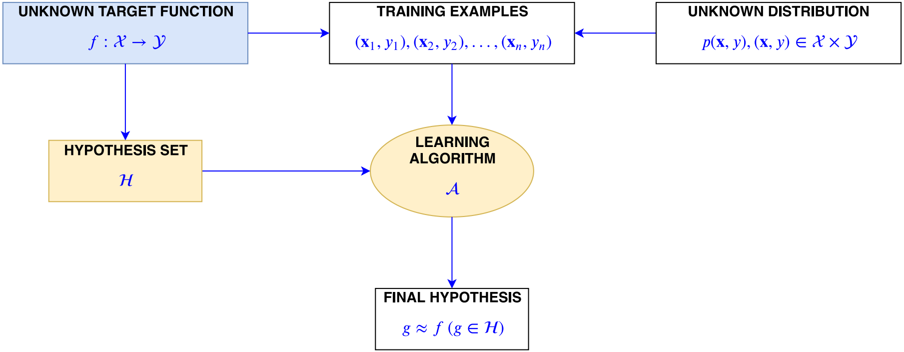
$$P[|E_{in}(h) - E_{out}(h)| > \epsilon] \leq 2e^{-2\epsilon^2 N} \text{ for any } \epsilon > 0$$

Inductive Learning Hypothesis

Generalization is possible.

- If a machine performs well on most **training data** AND it is not too complex, it will probably do well on **similar test data**.

Back to Learning Diagram



References

Goodfellow, I., Bengio, Y., and Courville, A. (2016).
Deep learning.
MIT press.

Lê, B. and Tô, V. (2014).
Cở sở trí tuệ nhân tạo.
Nhà xuất bản Khoa học và Kỹ thuật.

Nguyen, T. (2018).
Artificial intelligence slides.
Technical report, HCMC University of Sciences.

Russell, S. and Norvig, P. (2016).
Artificial intelligence: a modern approach.
Pearson Education Limited.