Math Camp for Machine Learning

Synferlo

Mar. 22, 2021

1 Statistical Learning

1.1 Elements in ML

1. Instance/example:

x, $x \in X$

2. Instance space/domain:

X (where the instance comes from).

3. Label:

Each instance has a label, or class. Label can be 0/1 or +/-.

4. Concept:

There's a function c, called concept, that tells the **true** relationship between instance and label.

$$\text{concept } c : X \to \{0, 1\}$$

Each instance x is labeled by $c(x)$. Our goal is to find this $c(\cdot)$.

5. Hypothesis:

Note, this is NOT the same one as we say in econometrics. Hypothesis, $h(\cdot)$, is a function that the machine use to do the prediction given an instance x.

$$h : X \to \{0, 1\}$$

6. Concept VS Hypothesis:

Concept is the TRUE relationship between x and label.

Hypothesis is the GUESS of our machine given the training data.

7. Concept class:

C is where concept c comes from, $c \in C$.

8. Distribution:

All instances are generated from a particular distribution D. We call it target distribution or distribution for short.

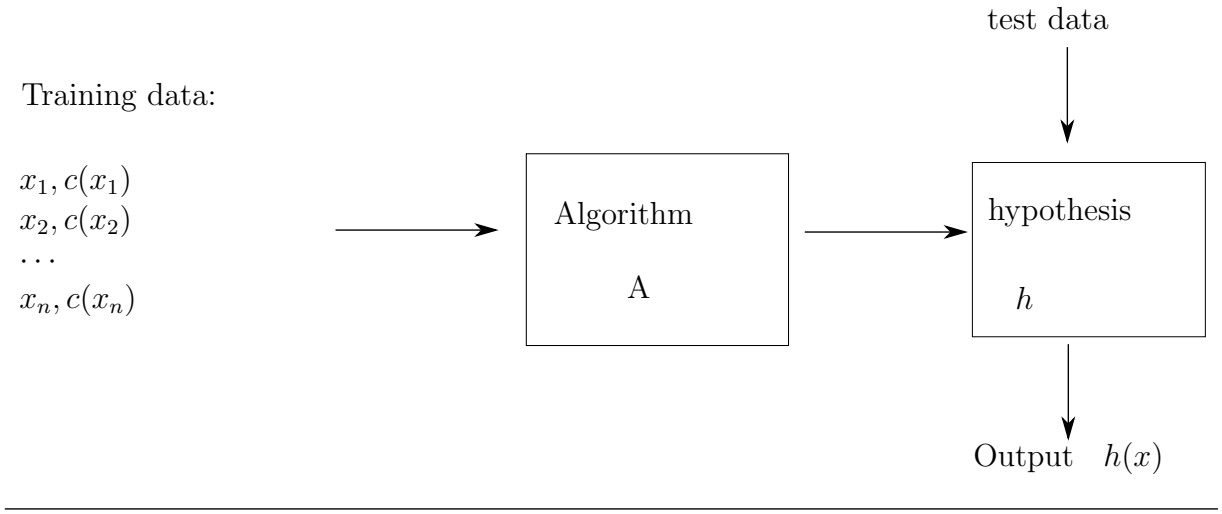
$$x_i \sim D, i.i.d.$$

Hypothesis class:

It tells where the hypothesis comes from. We allow h and c come from different classes.

$$h \in \mathcal{H}$$

1.2 ML Process



$x_i \in X, \quad x_i \sim D, \quad c \in C, \quad h \in \mathcal{H}$

Figure 1: ML process

1.3 PAC Learning

We want to see $h(x) = c(x)$

We DO NOT want to see $h(x) \neq c(x)$

1.3.1 How we measure error:

$$err_D(h) = Pr_{x \sim D}\left[h(x) \neq c(x)\right]$$

We want this,

$$err_D(h) \leq \varepsilon,$$

where ε is a small positive number.

To guarantee the machine work well, we require the following condition,

$$Pr\left(err_D(h) \leq \varepsilon\right) \geq 1 - \delta$$

where δ is a small positive number.

Hence, $err_D \leq \varepsilon$ requires algorithm to be more accurate. $Pr(err \leq \varepsilon) \geq 1 - \delta$ requires the probability of this correction to be high.

This method is called Probability approximately correct, or PAC for short.

==========

We say concept space C is PAC-learnable by $\mathcal{H}$,

if there exist an algorithm (alg.) A, $\forall c \in C$, $\forall$ distribution D, $\forall \varepsilon > 0, \delta > 0$,

A takes $m = poly(\frac{1}{\varepsilon}, \frac{1}{\delta}, \cdots)$ random examples $x_i \sim D$,

that it makes output hypothesis $h \in \mathcal{H} \quad s.t.\ Pr(err_D(h) \leq \varepsilon) \geq 1 - \delta$.

NB: m is sample size. The more data we have, the higher accuracy that $h(\cdot)$ will be. Hence, m is negative correlated with ε and δ.

==========

Here's an example: For $X \in \mathbb{R}$

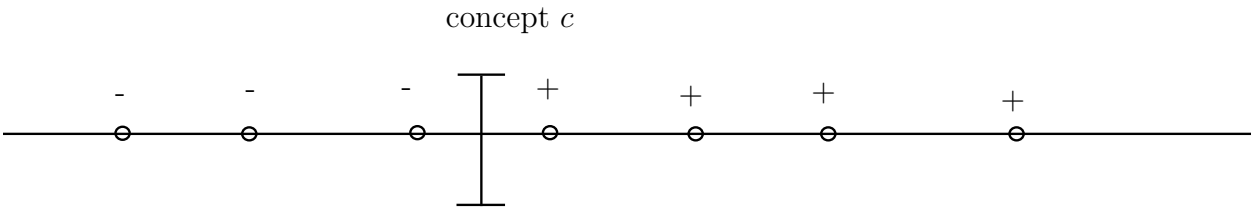


Figure 2: Threshold

There are many instances on the real line. Concept c is a threshold Function.

All instances on the right are labeled by + (True)

All instances on the left are labeled by - (False)

What we are doing is like this,

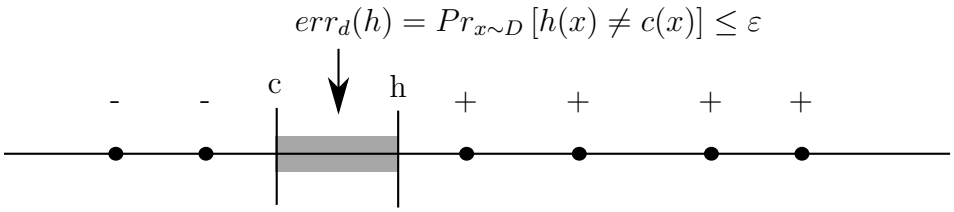


Figure 3: hyp. and concept

So, output $h(x)$ would be

$$h(x) = \begin{cases} + & \text{if } x \geq b \\ - & \text{O.W.} \end{cases}$$

In this case, we say $\mathcal{H} = C$.

Now, let's consider a general case.

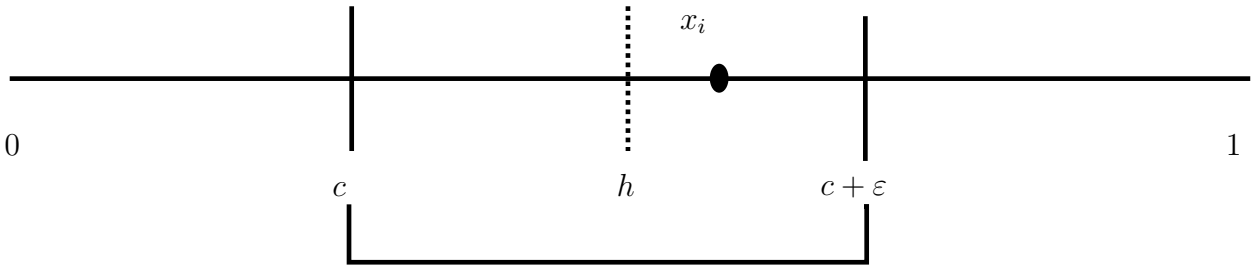


Figure 4: General case

If a training data example x_i falls in $(c, c+\varepsilon)$, region R, we have to shift $h(\cdot)$ to the left of x_i, so that

$$Pr\,[\quad err_D(h) > \varepsilon \quad] \leq Pr\,(\quad \text{no } x_i \text{ in } R \quad) = Pr\,(x_1 \notin R, x_2 \notin R, \cdots x_m, \notin R)$$

Since x_i is *i.i.d.*, we can write

$$Pr(x_1 \notin R, \cdots, x_m \notin R) = \prod_{i=1}^{m} Pr(x_i \notin R) = \prod_{i=1}^{m} (1-\varepsilon) = (1-\varepsilon)^m$$

Recall from basic calculus, $1 + x \leq e^x$, so we can rewrite

$$(1-\varepsilon)^m \leq (e^{-\varepsilon})^m = e^{-\varepsilon m}$$

Also, since

$$Pr(err_D(h) \leq \varepsilon) \geq 1 - \delta$$

we have

$$Pr(err_D(h) > \varepsilon) \leq \delta$$

So, we end up with

$$\begin{aligned} Pr(err_D(h) > \varepsilon) &\leq e^{-\varepsilon m} \\ &\leq \delta \end{aligned}$$

We need to solve for m, so that we can know the condition for sample size. Because we need to make sure this upper bound, $e^{-\varepsilon m}$, no greater than δ, we can write this,

$$\begin{aligned} e^{-\varepsilon m} &\leq \delta \\ -\varepsilon m &\leq \ln \delta \\ m &\geq -\frac{\ln \delta}{\varepsilon} \\ m &\geq \frac{\ln \frac{1}{\delta}}{\varepsilon} \end{aligned}$$

It says at least you should have sample size greater than this lower bound to guarantee $Pr(err_D(h) \leq \varepsilon) \geq 1 - \delta$.

1.4 Finite Hypothesis Space

Let's start with finite hypothesis space case, $|\mathcal{H}| < \infty$.

==========

Theorem:

Suppose hypothesis space $\mathcal{H}$ is finite, Algorithm A finds hypothesis $h_A \in \mathcal{H}$ is consistent with m random training examples where

$$m \geq \frac{1}{\varepsilon}(\ln |\mathcal{H}| + \ln \frac{1}{\delta}) \tag{1}$$

Then we can say

$$Pr\left[err_D(h_A) > \varepsilon\right] \leq \delta \tag{2}$$

or it is PAC learnable.

Equivalently, we can write,

$$\text{with } prob. \geq 1 - \delta,$$
$$err_D(h) \leq \frac{\ln |\mathcal{H}| + \ln \frac{1}{\delta}}{m} = \varepsilon$$

where h is a random variable.

==========

Notice, consistent here is NOT what we mean in Econometrics!!

Here, consistent means hyp. makes NO mistake in training examples.

==========

In last section we have

$$m \geq \frac{\ln \frac{1}{\delta}}{\varepsilon}.$$

Now, we add $\ln |\mathcal{H}|$ to the RHS,

$$m \geq \frac{1}{\varepsilon}(\ln |\mathcal{H}| + \ln \frac{1}{\delta})$$

Term $\ln |\mathcal{H}|$ measures the complexity of the hypothesis space.

==========

Complexity:

If we want to give each hyp. a name in $\mathcal{H}$, how many bits we need to do that?

The number of bits you need would be $log_2 |\mathcal{H}|$, where $|\mathcal{H}|$ stands for the number of hypothesis in this class. So, here, we use $\ln |\mathcal{H}|$ roughly measures the complexity of $\mathcal{H}$.

==========

Recall three conditions we need to do machine learning:

1. enough data
2. fit the training set well
3. use simple classifier

Equation (2) gives us an accurate classifier. Equation (1) tells us we have enough data. Remember, Alg. A finds a h_A is consistent with training data. And $\ln|\mathcal{H}|$ bound the complexity. The more complex it is, the higher value would $\ln|\mathcal{H}|$ be. Then, m will also go up. It means with more complex hypothesis space, we need more data to get an accuracy prediction. So, we need to limit the complexity. Note, $|\mathcal{H}|$ is the cardinality, not absolute value.

==========

Theorem:

Assume given $m \geq \frac{1}{\varepsilon}(\ln|\mathcal{H}| + \ln\frac{1}{\delta})$ examples, with *prob.* $\geq 1-\delta$,

$\forall h \in \mathcal{H}$:

if h is consistent, then $err_D(h) \leq \varepsilon$.

Note,

if $err_D(h) \leq \varepsilon$, we say h is ε-good

if $err_D(h) \leq \varepsilon$, we say h is ε-bad

==========

Proof:

$Pr(\forall h \in \mathcal{H} : h \text{ consistent } \Rightarrow h \text{ is } \varepsilon\text{-good}) \geq 1-\delta$

can be written as

$Pr(\exists h \in \mathcal{H} : h \textit{ cons.} \text{ and } h \text{ is } \varepsilon\text{-bad}) \leq \delta$.

==========

Remember h is a random variable because it depends on training samples. But "e is ε-bad" is not a RV.

==========

Let's go back to the proof.

Define

$$B = \{h \in \mathcal{H} : \ h \text{ is } \varepsilon\text{-bad } \}$$

So we can write

$$\begin{aligned}
& Pr(\exists h \in \mathcal{H} : \ h \text{ cons. } \Rightarrow \ h \text{ is } \varepsilon\text{-bad }) && (3)\\
=& Pr(\exists h \in B : h \text{ is cons. }) && (4)\\
=& Pr\left(\bigcup_{h \in B} (h \text{ is cons. })\right) && (5)\\
=& (h_1 \in B \text{ cons. }) \cup (h_2 \in B \text{ cons. }) \cdots (h_n \in B \text{ cons. }) && (6)\\
\leq& \sum_{h \in B} Pr(h \text{ cons. }) \quad \text{recall, } Pr(a \cup b) \leq Pr(a) + Pr(b) && (7)
\end{aligned}$$

Then what is the $Pr(h \text{ cons. })$?

Because x_i are *i.i.d.*,

$$\begin{aligned}
Pr(h \text{ cons. }) &= Pr(h(x_1) = c(x_1) \cap \cdots \cap h(x_m) = c(x_m)) && (8)\\
&= \prod_{i=1}^{m} Pr(h(x_i) = c(x_i)) && (9)\\
&\leq (1 - \varepsilon)^m && (10)
\end{aligned}$$

Note, since $Pr(h(x_i) \neq c(x_i)) \leq \varepsilon$, so $Pr(h(x_i) = c(x_i)) \leq 1 - \varepsilon$.

Hence, from equation (7) and (10), we can write

$$
\begin{aligned}
& Pr\left(\bigcup_{h\in B}(h \text{ cons. })\right) \\
& \leq \sum_{h\in B} Pr(h \text{ cons. }) \\
& \leq |B|\,(1-\varepsilon)^m \quad \text{Note, } |B| \leq |\mathcal{H}| \\
& \leq |\mathcal{H}|\, e^{-\varepsilon m} \\
& \leq \delta
\end{aligned}
$$

How do get δ from $|\mathcal{H}|\, e^{-\varepsilon m}$?

Take log, we have $\ln|\mathcal{H}| - \varepsilon m$. Given $m \geq \frac{1}{\varepsilon}(\ln|\mathcal{H}|)$

$$
\begin{aligned}
em &\geq \ln|\mathcal{H}| + \ln\frac{1}{\delta} \\
-\varepsilon m &\leq -\ln|\mathcal{H}| - \ln\frac{1}{\delta} = -\ln|\mathcal{H}| + \ln\frac{1}{\delta} \\
\ln|\mathcal{H}| - \varepsilon m &\leq \ln\delta \\
|\mathcal{H}|\, e^{-\varepsilon m} &\leq \delta \quad \text{take exponential} \\
& Q.E.D.
\end{aligned}
$$

1.5 Infinite hypothesis space

Suppose we have four examples on the real line. h_1 and h_2 give us the same prediction.

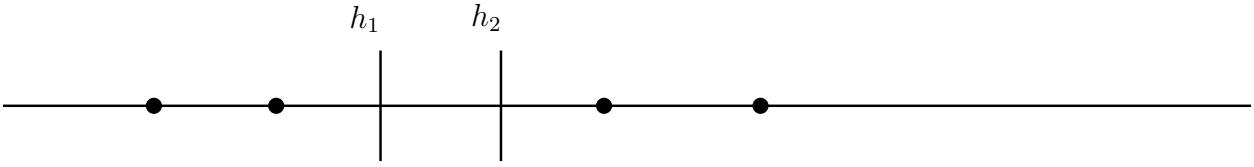


Figure 5: same prediction

The prediction would be different if we do this,

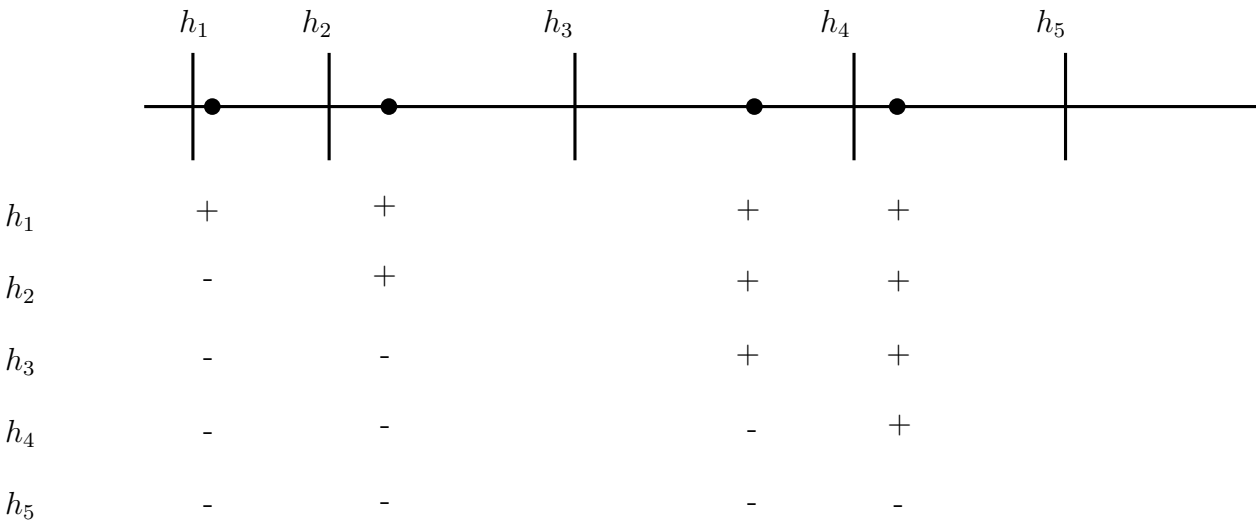


Figure 6: different hyp.

For each of the four instances, hypothesis's behavior can be either labeling instance as + or -. So, for m instances, the max behavior is 2^m. Here we have $m+1$, 5, hypothesis.

1.5.1 Growth Function

A set of examples $\mathcal{S} = \{x_1, \cdots, x_m\}$ contains m labeled instances. For particular hypothesis h,

$$\Pi_{\mathcal{H}}(\mathcal{S}) = \{\{h(x_1), \cdots, h(x_m)\} : h \in \mathcal{H}\} \tag{11}$$

where $\Pi_{\mathcal{H}}(\mathcal{S})$ is a set of hypothesis behaviors.

We can look at how bad (maximum number) this set can be.

$$\Pi_H(m) = \max_{|\mathcal{S}|=m} |\Pi_{\mathcal{H}}(\mathcal{S})|$$

These are all cardinality, not absolute value. They measures the size.

The Growth function tells us the maximum number (the worst case) of labeling behavior a hypothesis space can make given a sample with size m.

In figure 6, we have $m+1$ hypothesis to make it an effective hypothesis space. And $\Pi_{\mathcal{H}}(m)$

is called the growth function.

Recall, when $|\mathcal{H}| < \infty$, we have

$$err_D(h) \leq \frac{\ln |\mathcal{H}| + \ln \frac{1}{\delta}}{m}$$

Now, we relax this finite assumption, and we can use the growth function to measure the complexity of the hypothesis space.

Theorem:

Given m training examples, with *prob.* $\geq 1 - \delta$, $\forall h \in \mathcal{H}$,

if h is consistent, then

$$err_D(h) \leq O\left(\frac{\ln \Pi_{\mathcal{H}}(2m) + \ln \frac{1}{\delta}}{m}\right) \tag{12}$$

If the growth function is polynomial (this is a nice case),

$$\Pi_{\mathcal{H}}(m) = O(m^d)$$

where d is a constant. We use big O here to hide constant terms. Hence the error becomes,

$$err_D(h) \leq \frac{d \ln m + \ln \frac{1}{\delta}}{m} \tag{13}$$

In the nice case,

$$\Pi_{\mathcal{H}}(m) = O(m^d),$$

it is the finite case.

In the worst case, the growth function realizes all possible behaviors,

$$\Pi_H(m) = 2^m,$$

it is an infinite case.

For any hypothesis space $\mathcal{H}$, it will be either the nice case, or the worst case. There's NO other case!