PAC learning framework : basic definitions.

We introduce first the PAC learning framework. PAC means "Probably Approximately Correct".

Let $\mathcal{X}$ = domain set of "feature vectors"

(also called instances/patterns) $\underline{x} \in \mathcal{X}$

$\mathcal{Y}$ = "label set" $\quad y \in \mathcal{Y}$

- for classification probl $\mathcal{Y} = \{0, 1\}$
- for regression $\mathcal{Y} = \mathbb{R}$.
- ect ...

$\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$ $\quad z = (x, y)$ a "sample"

- $\underline{x}$ is often seen as an "input" vector/pattern
- y is often seen as an "output" label/value.

$$S = \{(\underline{x}_1, y_1), \dots, (\underline{x}_m, y_m)\}$$
$$= \{z_1, \dots, z_m\}$$

= "Training set" (might appear with repetitions)

The training set will be the input to learning algorithms. We must model where do the samples come from?

$\mathcal{D}(\underline{x}, y) = \mathcal{D}(y \mid \underline{x})\, \mathcal{D}(\underline{x})$.

conditional marginal.

We assume there exists some underlying distribution $\mathcal{D}$ such that $z_i \sim \mathcal{D}$ (say iid for simplicity) $i = 1, \dots, m$. This distribution is assumed to exist in "Nature" or in the "world" producing the samples but is generally unknown to the learner/us.

[Note: we could try to determine some estimate of $\mathcal{D}$ but this is highly sub-optimal a priori. Too many samples might be needed; one might be interested in some simple function of the samples. For example one might be interested in a classification rule; given a new $\underline{x}^{new}$ determine the label $y^{new} \in \{0, 1\}$.]

Learning Task :

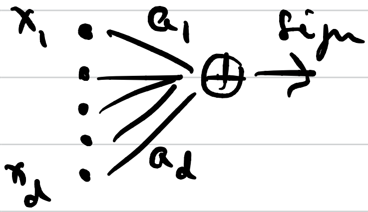
Given a training sequence S we want to have an algorithm A that outputs an "hypothesis" or a "rule" $h = A(S)$. Here an "hypothesis" or "rule" is a function :

$$h : \mathcal{X} \to \mathcal{Y}$$

from feature vectors to label set.

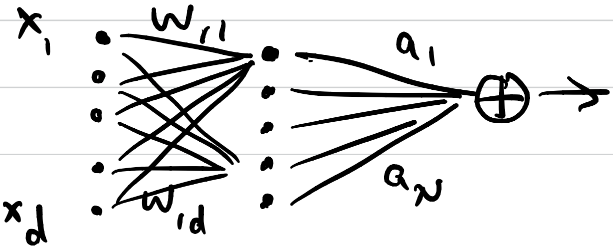
- In classification $h(x) \in \{0, 1\}$ say.
- In regression $h(x) \in \mathbb{R}$ say.

$h(x) = \underline{a}^T . \underline{x}$, $h(x) = \text{sign}(\underline{a}^T . \underline{x})$



perceptron

$$h(x) = \sum_{i=1}^{N} a_i \, \sigma((W \underline{x})_i)$$



two layer network.

ect ...

One often distinguishes two versions of the learning task:

<u>Proper learning</u>: $\mathcal{H}$ is a class of hypothesis given a priori. You ask which $h \in \mathcal{H}$ is the "best" in a sense to be specified (see later on).

Note: $\mathcal{H}$ plays the role of some prior knowledge / we might have specific constraints for possible hypothesis.

<u>Improper learning</u>: We still have some class of hypothesis $\mathcal{H}$ but the "best" h (picked by the algorithm say) might be outside $h \notin \mathcal{H}$.

Now we discuss what we mean by the "best" h.

We usually have a "loss function":

$$\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$$
$$(h, z) \mapsto \ell(h, z)$$

Popular loss fcts are:

$$\ell(h, z) = \ell(h, \underline{x}, y) = (h(\underline{x}) - y)^2$$

"square loss"

$$\ell(h, z) = \ell(h, \underline{x}, y) = \mathbb{1}(h(\underline{x}) \neq y)$$
$$= \begin{cases} 1 & \text{if } h(\underline{x}) \neq y \quad \text{unit cost} \\ 0 & \text{if } h(\underline{x}) = y \quad \text{no cost.} \end{cases}$$

"error fct or indicator loss fct".

Definition: True loss, true risk, true error, generalization error, ...

$$\boxed{L_{\mathcal{D}}(h) \equiv \mathbb{E}_{z \sim \mathcal{D}} \left[\ell(h, z) \right]}$$

This measures how well on average does an hypothesis do on samples $z \sim \mathcal{D}$.

We may define the "best" or "optimal" hypothesis as

$$h^* \in \text{argmin } L_{\mathcal{D}}(h)$$

(or say $h^* = \text{argmin } L_{\mathcal{D}}(h)$ for definiteness)

However there is a conceptual problem here because the rule of the game is that $\mathcal{D}$ is unknown. One possible remedy as we will be using is to minimize a proxy:

Empirical Risk: $L_S(h) \equiv \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i)$

associated to a training set S.

ERM rule or algorithm: The empirical risk minimization rule consists to take

$$A(S) = h_S \in \text{argmin } L_S(h).$$

Now we are ready to adress the question of learnability of an hypothesis class. The philosophy is to take a "competitive point of view" where one tries to achieve something nearly as good as h^*.

Note that the definition here is independent of a specific rule such as ERM or other rule.

Definition : PAC learning.

An hyp class $\mathcal{H}$ is "agnostic PAC learnable" with respect to a set $\mathcal{Z}$ and loss fct $\ell : \mathcal{H} \times \mathcal{Z} \to \mathbb{R}_+$ if $\exists$ a function $m_{\mathcal{H}} : [0,1]^2 \to \mathbb{N}$; $(\epsilon, \delta) \to m_{\mathcal{H}}(\epsilon, \delta)$ and a learning rule or algo A such that :

- $\forall (\epsilon, \delta) \in [0,1]^2$ and $\forall \mathcal{D}$ on $\mathcal{Z}$, when running A on $m > m_{\mathcal{H}}(\epsilon, \delta)$ iid samples $S \sim \mathcal{D}^m$

$$\mathbb{P}\text{rob}_S \left\{ L_{\mathcal{D}}(A(S)) \leq \min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) + \epsilon \right\} \geq 1 - \delta$$

[i.e we succeed with accuracy ϵ with prob $1-\delta$
i.e we succeed probably approximately correctly.]

Remarks:

1) If $\mathcal{H} = \{h_0\}$ $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h) = L_{\mathcal{D}}(h_0)$ and the algorithm $A(S) = h_0$ that trivially outputs h_0 succeeds

So the singleton class is trivially learnable.

If $\mathcal{H}$ grows $\min_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$ might diminish and one might need more and more samples.

Determining if a class is learnable is non trivial and depends on many things:

- size and quality of S
- size of $\mathcal{H}$
- loss fct
- types of A

all these objects interact and it is not easy to determine their interactions.

2) Perhaps the most popular learning algo is ERM.

$$A(S) = \underset{h \in \mathcal{H}}{\operatorname{argmin}} \; L_S(h)$$

which minimizes empirical risk. How is this minimized or even whether we really want to minimize or somehow approximately minimize is a modern question of ML.

3) Above definition of learnability does not address the question of complexity and computational resources. This is an important issue even with ERM if $\mathcal{H}$ is large and/or S is large.

4) $\boxed{\text{Vocabulary}}$: we say that we are in the <u>Realizable case</u> if $\exists\, h_* \in \mathcal{H}$ such that $L_{\mathcal{D}}(h_*) = 0$.

Then $\mathcal{H}$ is <u>learnable</u> if $\Pr_S \{ L_{\mathcal{D}}(A(S)) \leq \epsilon \} \geq 1-\delta$

<u>Agnostic learnable</u> means that we do not know if we are in the realizable case.

Were do we go from here?

→ we will first analyze $\mathcal{H}$ finite. We will show that finite class are learnable by ERM. We will use a notion of uniform convergence of $L_S(h)$ to $L_{\mathcal{D}}(h)$ somehow and will obtain an estimate of $m_{\mathcal{H}}(\epsilon, \delta)$

→ in the exercises you will look at natural $\mathcal{H}$ given by separating planes, circles, threshold fcts and see that infinite $\mathcal{H}$'s can also be learnable. This will be understood in terms of the VC dimension which is an effective measure of the "size" of $\mathcal{H}$. We will see that classes with finite VC dim are learnable.

→ But it is not true that all infinite classes are learnable.

We will prove a "No free lunch thm" which roughly states that there is no universal A: i.e given A one can find $\mathcal{D}$ s.t A will fail (if $|\mathcal{X}|$ is big enough).