Some notes on Statistical Learning Theory

Course of Machine Learning
Master Degree in Computer Science
University of Rome "Tor Vergata"
a.a. 2024-2025

Giorgio Gambosi

Statistical learning theory studies from the theoretical point of view the effect of the chosen set of hypotheses and of the training set size on the quality (in terms of risk) of the predictor derived from learning.

As said above, a *learning algorithm* $\mathcal{A}$ takes a set $\mathcal{T}$ of pairs in $\mathcal{X} \times \mathcal{Y}$ and returns a predictor $A_{\mathcal{T}}$ computing a function $h_{\mathcal{T}} : \mathcal{X} \mapsto \mathcal{Y}$.

Indeed, $\mathcal{A}$ requires the specification of a search space, that is a class of functions $\mathcal{H}$ over which the selection of $h_{\mathcal{T}}$ is performed: this is called *Hypothesis class* or Inductive bias.

A particular and relevant learning algorithm is *Empirical risk minimization* (ERM), which consists in returning, given a dataset $\mathcal{T}$, the predictor $\mathcal{H}$ which minimizes the training error,

$$ERM(\mathcal{T}) = h_{\mathcal{T}} = \operatorname*{argmin}_{h} \overline{\mathcal{R}}_{\mathcal{T}}(h)$$

Being a particular learning algorithm, ERM requires itself the specification of the class of functions $\mathcal{H}$ over which the minimization is performed

$$ERM(\mathcal{T}, \mathcal{H}) = h_{\mathcal{T},\mathcal{H}} = \operatorname*{argmin}_{h \in \mathcal{H}} \overline{\mathcal{R}}_{\mathcal{T}}(h)$$

A fundamental question in learning theory is over which hypothesis classes applying a learning algorithm $\mathcal{A}$, and in particular ERM, will not result in a likely limited risk, for different training sets.

Finite search space, realizability, and 0-1 loss

The simplest type of restriction on $\mathcal{H}$ is assuming the number of possible predictors is upper bounded. In this case, choosing $h_{\mathcal{T}}$ will not overfit, provided $\mathcal{T}$ is sufficiently large, that is a sufficient number of examples of the application of f is available[1].

Let us limit first ourselves to the case that there exists a predictor in $h^* \in \mathcal{H}$ which does not make any error in classifying items in $\mathcal{X}$, that is such that

$$\mathcal{R}_{p_M,f}(h^*) = \mathop{\mathbb{E}}_{p_M,f}[\, L(h^*(\mathbf{x}), f(\mathbf{x}))\,] = \mathop{\mathbb{E}}_{p_M,f}[\, |\mathbf{x} \in \mathcal{X} : h^*(\mathbf{x}) \neq f(\mathbf{x})|\,] = 0$$

This is denoted as **realizability assumption** and, since the 0-1- loss is a non negative function, it implies that[2] it correctly classifies all the elements in any subset of $\mathcal{X}$, that is for any possible set $\mathbf{X}$ of elements sampled from $\mathcal{X}$, which implies that for any training set $\mathcal{T} = (\mathbf{X}, \mathbf{t})$, where $t_i = f(\mathbf{x}_i)$ by assumption, it results

[1] Our considerations will be limited here to the task of binary classification with $0-1$ loss function, and we will refer to the simpler scenario where there is a functional, albeit unknown, relation between items and target values, i.e. classes.

[2] With probability 1, since the expectation equal to 0 does not rule out the possibility of single points (that is sets of measure 0) in $\mathcal{X}$ misclassified by h^*.

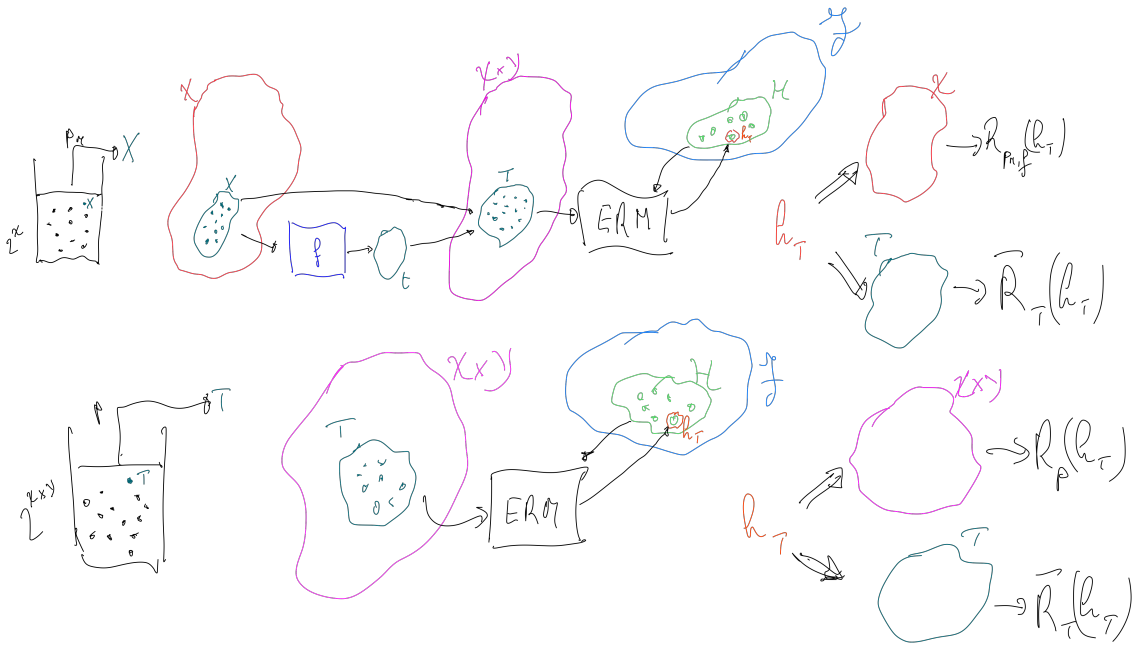

Figure 1: Sketch of the situation. Above, the case of a deterministic relation between item and target value. We assume a set $\mathbf{X}$ of items has been sampled from the population $\mathcal{X}$ under probability distribution $p_M(\mathbf{x})$ and the corresponding target values $\mathbf{t}$ are associated by applying the (unknown) function f. ERM (indeed, any learning algorithm) selects, given the resulting training set $\mathcal{T} = (\mathbf{X}, \mathbf{t})$, a predictor $h_\mathcal{T}$ from the hypothesis class $\mathcal{H}$, which is a subset of the set $\mathcal{F}$ of all possible predictors. When applied to $\mathcal{T}$, the selected predictor has minimum empirical risk $\overline{\mathcal{R}}_\mathcal{T}(h_\mathcal{T})$, while when the expectation over all possible training sets, sampled selecting $\mathbf{X}$ according to p_M, is taken into account, the risk $\mathcal{R}_{p_M,f}(h_\mathcal{T})$ is not necessarily the minimum one: this means that $h_\mathcal{T}$ behaves well on known data and more poorly when applied to new data (overfitting). Observe that under the realizability hypothesis, $\overline{\mathcal{R}}_\mathcal{T}(h_\mathcal{T}) = 0$. Below, the same is shown in the more general case when a probabilistic relation is assumed between items and target values. Here the same training set $\mathcal{T}$ is assumed sampled from the population $\mathcal{X} \times \mathcal{Y}$ under distribution $p(\mathbf{x}, t) = p_M(\mathbf{x})p_C(t|\mathbf{x})$.

$$\overline{\mathcal{R}}_\mathcal{T}(h^*) = \frac{1}{|\mathcal{T}|} \sum_{(x,t)\in\mathcal{T}} L(h^*(x), t) = \frac{|(\mathbf{x}, t) \in \mathcal{T} : h^*(\mathbf{x}) \neq t|}{|\mathcal{T}|} = 0$$

Realizability also implies that for any training set $\mathcal{T}$ the predictor returned by ERM is optimal when applied to $\mathcal{T}$, that is $\overline{\mathcal{R}}_\mathcal{T}(h_\mathcal{T}) = 0$: this derives from the observation that, given $\mathcal{T}$, there exists at least one predictor (that is h^*) which correctly classifies all elements in it. Since ERM returns a predictor, say it $h_\mathcal{T}$, which best classifies such elements it must be $\overline{\mathcal{R}}_\mathcal{T}(h_\mathcal{T}) = 0$. Note that, as a special case, it might be $h_\mathcal{T} = h^*$, that is ERM returns precisely h^*: in this case, the predictor returned by ERM on $\mathcal{T}$ is optimal on the whole elements population $\mathcal{X}$. However, it may also happen that $h_\mathcal{T} \neq h^*$, which implies that the predictor returned by ERM on $\mathcal{T}$ is optimal on $\mathcal{T}$ (that is $\overline{\mathcal{R}}_\mathcal{T}(h_\mathcal{T}) = 0$), but it may return incorrect predictions on some elements not in $\mathcal{T}$ (that is, $\mathcal{R}_{p_M,f}(h_\mathcal{T}) > 0$).

Given $\varepsilon > 0$ let us make the following definitions:

- a predictor $h \in \mathcal{H}$ is *bad* if its risk is greater than ε, i.e. $\mathcal{R}_{p_M,f}(h) > \varepsilon$; that is, a bad predictor is one that it is expected to misclassify an unacceptable fraction of items (where unacceptable stands for greater than ε);

let us denote as $\mathcal{H}_B$ the set of predictors in $\mathcal{H}$ which are bad.

- a set $\mathbf{X} \subset \mathcal{X}$ is *bad* if there exists at least one optimal predictor $\mathcal{H}$ on the training set $\mathcal{T} = (\mathbf{X}, \mathbf{t})$, where items in that is with $\overline{\mathcal{R}}_{\mathcal{T}}(h) = 0$, which is bad, that is such that $\mathcal{R}_p(h_{\mathcal{T}}) > \varepsilon$. If that predictor is the one actually returned by ERM, i.e. $h_{\mathcal{T}} = h$, than the dataset is *very bad*.

Assume now that, given $|\mathcal{H}|$, we want to study for which values of $\mathcal{T}$ size n the probability that our training set (assuming that it results from independently sampling each element from $\mathcal{X}$ with distribution p_M and setting $t_i = f(\mathbf{x}_i)$) is a bad one is sufficiently small, for example, less than a given $\delta \in (0, 1)$. Hence,

$$\mathop{\mathbb{P}}_{\mathcal{T} \sim p^n} \left[\exists \tilde{h} \in \mathcal{H}_B : \overline{\mathcal{R}}_{\mathcal{T}}(\tilde{h}) = 0 \right] \leq \delta$$

this can be proved to be true if

$$\delta \geq |\mathcal{H}| e^{-\varepsilon n}$$

that is, if

$$n \geq \frac{1}{\varepsilon} \ln \frac{|\mathcal{H}|}{\delta} \tag{1}$$

This tells us that the probability that $\mathcal{T}$ is a bad dataset can be made arbitrary small by (logarithmically) increasing the dataset size. Observe that n has to be increased (logarithmically) also if the hypotheses class size increases or if the definition of bad predictor is made stricter by decreasing the amount ε of accepted misclassifications.

PAC Learning

The considerations above can be made more precise by introducing the concept of **Probably Approximatey Correct (PAC) Learning**, with respect to the case of a binary classification problem using 0-1 loss as a measure of error.

Definition 1 (PAC Learnability). *A hypothesis class $\mathcal{H}$ is* **PAC learnable** *if there exists a function $m_{\mathcal{H}} : (0, 1)^2 \mapsto \mathbb{N}$ and a learning algorithm $\mathcal{A}$ such that for every $\varepsilon, \delta \in (0, 1)$[3] for every distribution p_M over $\mathcal{X}$ and for every function $\mathcal{X} \mapsto \mathcal{Y}$ if the realizable assumption holds with respect to $\mathcal{H}$, p_M and f (that is $\mathcal{R}_{p_M, f}(h^*) = 0$) then when $\mathcal{A}$ is applied on a training set $\mathcal{T}$ of size $n \geq m_{\mathcal{H}}(\varepsilon, \delta)$, generated sampling n i.i.d. pairs from p, the algorithm returns a predictor $h_{\mathcal{T}}$ that, with probability at least $1 - \delta$ (over the choice of $\mathcal{T}$), has risk $R_{p_M, f}(h_{\mathcal{T}}) \leq \varepsilon$.*

The accuracy parameter ε determines how far the output classifier can be from the optimal one (this corresponds to the "approximately correct"), and a confidence parameter δ indicating how likely the classifier is to meet that accuracy requirement (corresponds to the "probably" part of "PAC").

The sample complexity $m_{\mathcal{H}}$ is a function of the accuracy (ε) and the confidence (δ) and determines the minimum number of examples which are required to guarantee that an approximately (ε) correct predictor is probably, that is with probability ($1 - \delta$), selected out of $\mathcal{H}$. Observe that the sample complexity also depends on properties of $\mathcal{H}$: for example, as seen above, for a finite class it grows as $\ln|\mathcal{H}|$.

In particular, equation **??** states that, if $\mathcal{H}$ is finite and the realizability assumption is verified, by having at our disposal a dataset whose size n is large enough we are sure that with probability $1 - \delta$ a good predictor (whose risk is less than ε) is returned by applying a specific algorithm $\mathcal{A}$, that is ERM. This implies that if no assumption is made on the learning algorithm applied the number of required examples (that is the sample complexity) could be smaller, thus resulting into

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq \left\lceil \frac{1}{\varepsilon} \ln \frac{|\mathcal{H}|}{\delta} \right\rceil$$

[3] Note that, since the 0-1 loss function is bounded in $(0, 1)$, the risk cannot be larger than 1, which results in $\varepsilon < 1$.

Optimal Prediction and Risk Minimization

Let us now try to extend the PAC-learnability definition to more general frameworks: in particular, we wish to generalize to the probabilistic case when items and corresponding target values are related only through a conditional distribution $p_C(\mathbf{x}, t)$. In this probabilistic framework, the optimal prediction is one that minimizes the risk:

$$h^*(\mathbf{x}) \stackrel{\Delta}{=} \operatorname*{argmin}_{y \in \mathcal{Y}} \mathop{\mathbb{E}}_{t \sim p_C(\cdot|\mathbf{x})} [\, L(y,t) \,] = \operatorname*{argmin}_{y \in \{0,1\}} p_C(t \neq y|\mathbf{x})$$

The optimal predictor h^* is also called **Bayes predictor**, and denoted as h_{Bayes}. However, observe that applying h_{Bayes} requires the knowledge of the conditional distribution $p_C(t|\mathbf{x})$ (or of the function f, if the assumption of the existence of function f is made), which we assume unknown. This makes the bayesian predictor out of reach given the assumptions hypothesized in ML, where we only assume that a training sample $\mathcal{T}$ of $p(\mathbf{x}, t)$ is given.

Since the bayesian predictor is optimal, we know that, for any learning algorithm $\mathcal{A}$ (including ERM) and for any training set $\mathcal{T}$, the risk of the predictor $h_{\mathcal{T}}$ returned by $\mathcal{A}$ when applied on $\mathcal{T}$ will be greater then (or equal at least) than the minimal possible error, that of h_{Bayes}, that is $\mathcal{R}_p(h_{\mathcal{T}}) \geq \mathcal{R}_p(h_{\text{Bayes}})$.

Moreover, it is possible to prove (by the No Free Lunch theorem which will be introduced shortly) that if no prior assumptions about $p(\mathbf{x}, t)$ is made, then there exists no learning algorithm that guarantees that, for any $\mathcal{T}$, the predictor $h_{\mathcal{T}}$ returned is as good as the bayesian one. In this situation, what we may require is that the learning algorithm for most datasets returns a predictor $h_{\mathcal{T}}$ with risk greater, but not too much greater, than $\mathcal{R}_p(h^*)$, the risk of the best predictor $h^* \in \mathcal{H}$, which in general has itself risk greater than h_{Bayes}. In doing this, we also generalize to the case when the realizability assumption does not hold.

Definition 2. *A hypothesis class $\mathcal{H}$ is* **agnostic PAC learnable** *if there exists a function $m_{\mathcal{H}} : (0,1)^2 \mapsto \mathbb{N}$ and a learning algorithm such that for every $\varepsilon, \delta \in (0,1)$ and for every distribution p over $\mathcal{X} \times \mathcal{Y}$, when the learning algorithm is applied on a training set $\mathcal{T}$ of size $n \geq m_{\mathcal{H}}(\varepsilon, \delta)$ generated by sampling n i.i.d. pairs from p, the algorithm returns a predictor $\mathcal{H}$ that, with probability of at least $1 - \delta$ (over the choice of $\mathcal{T}$), has risk*

$$\mathcal{R}_p(h^*) \leq \mathcal{R}_p(h) \leq \mathcal{R}_p(h^*) + \varepsilon$$

where $R_p(h^*) = \min_{h' \in \mathcal{H}} R_p(h')$.

Agnostic PAC learning offers a more general framework than PAC learning. Clearly, if the realizability assumption holds, Agnostic PAC Learnability reduces to PAC Learnability. However, when the realizability assumption does not hold, no learning algorithm can guarantee an arbitrarily small error for all $\mathcal{T}$. However, if agnostic PAC learnability holds, some algorithm is able to return, in most cases, a predictor from $\mathcal{H}$ not much worse that the best one in the class. On the contrary, in PAC learning we require that $h_{\mathcal{T}}$ behaves well in absolute terms.

The above definition can be further extended to the case of general (not 0-1) loss function as follows:

Definition 3 (Agnostic PAC Learnability for General Loss Functions)**.** *A hypothesis class $\mathcal{H}$ is agnostic PAC learnable with respect to a loss function l, if there exists a function $m_{\mathcal{H}} : (0,1)^2 \mapsto \mathbb{N}$ and a learning algorithm such that for every $\varepsilon, \delta \in (0,1)$ and for every distribution p over $\mathcal{X} \times \mathcal{Y}$, when the learning algorithm is applied on a training set S of size $n \geq m_{\mathcal{H}}(\varepsilon, \delta)$ generated sampling n i.i.d. pairs from p, the algorithm returns a predictor $\mathcal{H}$ that, with probability of at least $1 - \delta$ (over the choice of S), has risk*

$$R_p(h^*) \leq R_p(h) \leq R_p(h^*) + \varepsilon$$

where $R_p(h) = \mathbb{E}_{(x,y)\sim p}[\, l(h(x), y) \,]$.

Recall now that ERM, given a hypothesis class $\mathcal{H}$, operates, at least conceptually, as follows: upon receiving a training set $\mathcal{T}$, the learner evaluates for each predictor $h \in \mathcal{H}$ the risk $\overline{\mathcal{R}}_{\mathcal{T}}(h)$ and outputs a predictor $h_{\mathcal{T}} \in \mathcal{H}$ that minimizes this value. The underlying assumption is that the predictor $h_{\mathcal{T}} \in \mathcal{H}$ will also minimize (or closely approximate the minimum of) the true risk $\mathcal{R}_p(h)$ (or $\mathcal{R}_{p_m,f}(h)$) with respect to the actual data probability distribution.

For this assumption to hold, it is crucial to ensure that the empirical risks of all predictors in $\mathcal{H}$ are good approximations of their true risks. In other words, we require that, uniformly over all predictors in the hypothesis class, the empirical risk closely aligns with the true risk. This concept is formalized in the following definition:

A training set $\mathcal{T}$ is said ε-representative (with respect to $\mathcal{H}$, the loss function l, and the distribution $p(\mathbf{x}, t)$) if, for every $h \in \mathcal{H}$, the following condition holds:

$$|\overline{\mathcal{R}}_{\mathcal{T}}(h) - \mathcal{R}_p(h)| \leq \varepsilon$$

This formulation establishes a quantitative measure of how well the empirical risk approximates the true risk across the entire hypothesis class, providing a foundation for analyzing the effectiveness of the ERM learning paradigm.

Definition 4 (ε-representative sample)**.** *A training set $\mathcal{T}$ is considered ε-representative (with respect to domain $\mathcal{X} \times \mathcal{Y}$, hypothesis class $\mathcal{H}$, loss function l, and distribution $p(\mathbf{x}, t)$) if the following condition holds for all predictors in $\mathcal{H}$:*

$$\forall h \in \mathcal{H}, |\overline{\mathcal{R}}_{\mathcal{T}}(h) - \mathcal{R}_p(h)| \leq \varepsilon$$

Assume now that a training set $\mathcal{T}$ is $\frac{\varepsilon}{2}$-representative (w.r.t. $\mathcal{H}, l, p$): then for every $h \in \mathcal{H}$

$$\mathcal{R}_p(h_{\mathcal{T}}) \leq \mathcal{R}_p(h) + \varepsilon$$

That is, if $h_{\mathcal{T}}$ is returned by ERM on $\mathcal{T}$, its behaviour is not too worse than the best predictor in $\mathcal{H}$

$$\mathcal{R}_p(h_{\mathcal{T}}) \leq \overline{\mathcal{R}}_p(h^*) + \varepsilon$$

where as usual $h^* = \underset{h \in \mathcal{H}}{\operatorname{argmin}}\, \mathcal{R}_p(h)$.

This result establishes a crucial link between the representativeness of the training sample and the performance of the hypothesis selected by the ERM algorithm, providing a theoretical foundation for the effectiveness of this learning approach.

How can be exploit this observation to ensure that ERM is a good learning algorithm, at least in the PAC sense? It is sufficient to show that with probability $1 - \delta$ the sampling of the dataset from $\mathcal{X} \times \mathcal{Y}$ with probability distribution $p(\mathbf{x}, t)$ results in a training set which is $\frac{\varepsilon}{2}$-representative. The following definition will be useful to this task.

Definition 5 (Uniform Convergence)**.** *A hypothesis class $\mathcal{H}$ has the* **uniform convergence** *property (w.r.t. a loss function l) if there exists a function $m_{\mathcal{H}}^{UC} : (0,1)^2 \mapsto \mathbb{N}$ such that for every $\varepsilon, \delta \in (0,1)$ and for every probability distribution $p(\mathbf{x}, t)$, if $\mathcal{T}$ is a dataset of $n \geq m_{\mathcal{H}}^{UC}(\varepsilon, \delta)$ i.i.d. pairs, each one sampled from $\mathcal{X} \times \mathcal{Y}$ with probability p, then, with probability of at least $1 - \delta$, $\mathcal{T}$ is ε-representative.*

The function $m_{\mathcal{H}}^{UC}$ returns the (minimal) sample complexity necessary for the uniform convergence property. That is, the minimum number of examples necessary to ensure that with probability at least $1 - \delta$ the predictor obtained by ERM on $\mathcal{T}$ has risk equal to the risk of the best predictor in $\mathcal{H}$, plus at most 2ε.

In other terms, if a class of predictors $\mathcal{H}$ has the uniform convergence property with a function $m_{\mathcal{H}}^{UC}$ then the class is agnostically PAC learnable with sample complexity $m_{\mathcal{H}}(\varepsilon, \delta) \leq m_{\mathcal{H}}^{UC}(\frac{\varepsilon}{2}, \delta)$. Furthermore, in that case, ERM is a successful agnostic PAC learner for $\mathcal{H}$.

In summary, we have seen that finite hypothesis classes possess the uniform convergence property. This means that if we have a finite set of candidate predictors, we can guarantee that with high probability the empirical risk will be representative of the true risk. Consequently, the ERM rule is a sound approach for learning from finite hypothesis classes in an agnostic PAC framework.

ection*PAC learnability of finite classes

Let $\mathcal{H}$ be a finite hypothesis class, let $\mathcal{X} \times \mathcal{Y}$ be a domain, and let $l : \mathcal{X} \times \mathcal{Y} \mapsto [0, 1]$ be a loss function. Then, $\mathcal{H}$ enjoys the uniform convergence property with sample complexity

$$m_{\mathcal{H}}^{UC}(\varepsilon, \delta) \leq \left\lceil \frac{1}{2\varepsilon^2} \ln \frac{2|\mathcal{H}|}{\delta} \right\rceil$$

As a consequence, $\mathcal{H}$ is agnostically PAC learnable using the ERM algorithm with sample complexity

$$m_{\mathcal{H}}(\varepsilon, \delta) \leq m_{\mathcal{H}}^{UC}\left(\frac{\varepsilon}{2}, \delta\right) \leq \left\lceil \frac{1}{\varepsilon^2} \ln \frac{2|\mathcal{H}|}{\delta} \right\rceil$$

That is, if the number of possible predictors is m, then by sampling

$$\left\lceil \frac{1}{\varepsilon^2} \ln \frac{2|\mathcal{H}|}{\delta} \right\rceil = \left\lceil \frac{2(\ln m - \ln \delta + \ln 2)}{\varepsilon^2} \right\rceil$$

i.i.d. examples from $\mathcal{X} \times \mathcal{Y}$, each one under probability distribution $p(x, y)$, and applying ERM on the resulting dataset, we obtain with probability $1 - \delta$ a predictor whose expected loss over pairs in $\mathcal{X} \times \mathcal{Y}$ is the one of best predictor among the n considered, plus at most ε.

Observe that the number of examples grows logarithmically wrt n while decreases (logarithmically) wrt to δ and (as a square root) wrt ε.

While this result only applies to finite hypothesis classes, there is a simple trick that allows us to get a very good estimate of the practical sample complexity of infinite hypothesis classes.

Consider a hypothesis class that is parameterized by d parameters. For example, let $\mathcal{X} = \mathbb{R}, \mathcal{Y} = \{-1, +1\}$, and the hypothesis class, $\mathcal{H}$, be all functions of the form $h_\theta(x) = \text{sgn}(x - \theta)$. That is, each hypothesis is parameterized by one parameter, $\theta \in \mathbb{R}$, and the hypothesis outputs 1 for all instances larger than θ and outputs -1 for instances smaller than θ. This is a hypothesis class of an infinite size. However, if we are going to learn this hypothesis class in practice, using a computer, we will probably maintain real numbers using floating point representation, say, of 64 bits. It follows that in practice, our hypothesis class is parameterized by the set of scalars that can be represented using a 64 bits floating point number. There are at most 2^{64} such numbers; hence the actual size of our hypothesis class is at most 2^{64}. More generally, if our hypothesis class is parameterized by d numbers, in practice we learn a hypothesis class of size at most 2^{64d}. By the above considerations, we obtain that the sample complexity of such classes is bounded by

$$\frac{128d + 2\ln\frac{2}{\delta}}{\varepsilon^2} = \frac{128d + 2\ln 2 - 2\ln\delta}{\varepsilon^2}$$

This upper bound on the sample complexity has the deficiency of being dependent on the specific representation of real numbers used by our machine. In the folllowing we will introduce a rigorous way to analyze the sample complexity of infinite size hypothesis classes. Nevertheless, the discretization trick can be used to get a rough estimate of the sample complexity in many practical situations.

Relevance of the inductive bias

Whenever we choose a hypothesis class $\mathcal{H}$, we actually make use of some prior knowledge about our data. We only choose the class because we believe (or hope) that it contains a good (i.e. low-risk) predictor for the task we are considering.

This begs the question whether such prior knowledge is a necessary condition for learning, or whether there can exist a "universal learner". Recall that a learning task is defined by an unknown probability distribution p over the set of all possible examples and labelings $\mathcal{X} \times \mathcal{Y}$, and given a training set of size n, we want to find a hypothesis $h \in \mathcal{H}$ that has a low risk with high probability. So a universal learner would correspond to an algorithm $\mathcal{A}^*$ and a

sample complexity n such that $\mathcal{A}^*$ finds a low risk predictor $\mathcal{H}$ with high probability, whatever is the distribution p. The **No-Free-Lunch theorem** we now introduce states that no universal learner exists. More precisely, the theorem asserts that, for binary classification tasks, there is a distribution on which every learner fails. By failing we mean that the learner, after receiving i.i.d. examples from that distribution, returns with "high" probability a predictor whose risk is itself "high", while in the same situation, some other learner could return a predictor with a much lower risk. In other words, the theorem shows that no learner can succeed on all learnable tasks—every learner fails on some tasks where others succeed.

Theorem 1 (No Free Lunch)**.** *Let $\mathcal{A}$ be any learning algorithm (for binary classification with $0-1$ loss) over the domain $\mathcal{X}$, and let n be an integer $n < \frac{|\mathcal{X}|}{2}$. Then there exists a distribution $\overline{p}_{\mathcal{A}}$ over $\mathcal{X} \times \{0,1\}$ such that:*

1. *There exists a predictor $h^* : \mathcal{X} \mapsto \{0,1\}$ with $R_{\overline{p}_{\mathcal{A}}}(h^*) = 0$ (that is the realizability assumption holds on $\mathcal{X} \mapsto \{0,1\}$ if pairs are distributed according to $\overline{p}_{\mathcal{A}}$).*
2. *With probability at least $1/7$ over the choice of a dataset $\mathcal{T}$ of size n of i.i.d. pairs, each sampled according to $\overline{p}_{\mathcal{A}}$, we have that $R_{\overline{p}_{\mathcal{A}}}(h_{\mathcal{A},\mathcal{T}}) \geq 1/8$, where $h_{\mathcal{A},\mathcal{T}}$ is the predictor returned by $\mathcal{A}$ when applied on $\mathcal{T}$.*

This theorem states that for every learner, there exists a task (a distribution on $\mathcal{X} \times \mathcal{Y}$) on which it fails, even though that task can be successfully learned by another learner.

How does the No-Free-Lunch result relate to the need for prior knowledge?

Let us consider an ERM predictor over the hypothesis class $\mathcal{F}$ of all the functions f from an infinite-size $\mathcal{X}$ to $\{0,1\}$. This class represents lack of prior knowledge: every possible function from $\mathcal{X}$ to $\mathcal{Y} = \{0,1\}$ is considered.

According to the No Free Lunch theorem, any learning algorithm that chooses a predictor from hypotheses in $\mathcal{F}$, and in particular the ERM algorithm, will fail on some learning task. Therefore, the absence of prior knowledge results in the class $\mathcal{F}$ that is not PAC learnable.

Assume in fact, by contradiction, that $\mathcal{F}$ is PAC learnable. Recalling the definition of PAC learnability, this means that there must exist a learning algorithm $\mathcal{A}_{\mathcal{F}}$ and a function $m_{\mathcal{F}} : (0,1)^2 \mapsto \mathbb{N}$ such that for any ε, δ with $0 < \varepsilon, \delta < 1$ and for any distribution p on $\mathcal{X} \times \{0,1\}$ for which there exists an optimal predictor $h^* \in \mathcal{F}$ (with risk $R_p(h^*) = 0$) then, if $\mathcal{A}_{\mathcal{F}}$ is applied to a dataset of size $n \geq m_{\mathcal{F}}(\varepsilon, \delta)$ distributed according to p^n, the algorithm returns with probability greater than $1-\delta$ a predictor whose risk is "small", i.e. at most ε. That is, it returns with probability smaller than δ a predictor whose risk is "high", i.e. greater than ε.

Let us now assume $\varepsilon < 1/8$ and $\delta < 1/7$. By the No Free Lunch theorem, for any $n < \frac{|\mathcal{X}|}{2}$ (that is for any n, since $\mathcal{X}$ has infinite size) and for every learning algorithm $\mathcal{A}$ there must exist a distribution $\overline{p}_{\mathcal{A}}$ such that there exists $h^* \in \mathcal{F}$ with risk $R_{\overline{p}_{\mathcal{A}}}(h^*) = 0$ (as required in the definition of PAC learning) and that, with probability at least $1/7 > \delta$, has risk $R_{\overline{p}_{\mathcal{A}}}(\mathcal{A}(S)) \geq 1/8 > \varepsilon$.

This clearly holds also for the algorithm $\mathcal{A}_{\mathcal{F}}$ whose existence is assumed by the hypothesis of PAC learnability of $\mathcal{F}$: as a consequence, there exists a distribution $\overline{p}_{\mathcal{A}_{\mathcal{F}}}$ such that there exists a predictor h^* with null risk and that the predictor returned by $\mathcal{A}_{\mathcal{F}}$ has probability greater than δ to have "high", that is greater than ε, risk. This clearly contradicts the assumption of PAC learnability of $\mathcal{F}$.

Recall that PAC-learnability requires that there is an algorithm $\mathcal{A}$ and a sample complexity $m_H(\varepsilon, \delta)$ for any $\varepsilon > 0$, $0 < \delta < 1$, such that $R_p(h_{\mathcal{A},S}) \leq \varepsilon$ with probability at least $1-\delta$. The above theorem tells us that, if we do not restrict ourselves to a subset of all functions from $\mathcal{X}$ to $\{0,1\}$ (i.e. choose a hypothesis space), there will always be a probability distribution $\overline{p}$ that makes any learning algorithm return a "bad" predictor with high probability, even though there exists one with zero error. This implies that no algorithm will be able to PAC-learn this target function.

As a direct consequence if follows that, for some infinite domain $\mathcal{X}$, the set of all functions from $\mathcal{X}$ to $\{0,1\}$ cannot be PAC-learnable. No matter what training set size n we pick, $|\mathcal{X}|$ will always be larger than $2n$. So we can always apply the above theorem.

From the No Free Lunch Theorem we can conclude that choosing a suitable hypothesis class is crucial for learning a given concept. This way we restrict ourselves to a subset of all possible functions from $\mathcal{X}$ $\{0,1\}$, which helps us avoiding unfavourable distributions and might allow us to find a low-error hypothesis with high probability.

On the other hand we might exclude the best hypotheses from our set of candidates, as it might not be a member of our hypothesis class. So we might find a good approximation for the best hypothesis in our class, but this best hypothesis in the class might be a bad approximation for the true target predictor. This dilemma is often referred to as the Bias-Complexity Tradeoff .

If we choose a hypothesis class $\mathcal{H}$ and get a training set $\mathcal{T}$ as input, ERM returns a predictor $h_{\mathcal{T}}$ minimizing such that the empirical risk $\mathcal{R}_p(h)$ is minimal.

Now we can decompose this risk as follows:

$$\mathcal{R}_p(h_{\mathcal{T}}) - \mathcal{R}_p(h_{\text{Bayes}}) = \underbrace{(\mathcal{R}_p(h_{\mathcal{T}}) - \mathcal{R}_p(h^*)}_{\text{estimation error}} + \underbrace{(\mathcal{R}_p(h^*) - \mathcal{R}_p(h_{\text{Bayes}}))}_{\text{approximation error}} = \varepsilon_V + \varepsilon_B$$

where h^* is the best predictor in $\mathcal{H}$ (that is such that $\mathcal{R}_p(h^*) = \min_{h \in \mathcal{H}} \mathcal{R}_p(h^*)$, while h_{Bayes} is the absolute best predictor for the task (that is such that $\mathcal{R}_p(h_{\text{Bayes}}) = \min_{h \in \mathcal{F}} \mathcal{R}_p(h^*)$, where $\mathcal{F}$ is the set of all possible predictors).

More in detail:

- ϵ_B is the minimum risk achievable by any $h \in \mathcal{H}$: this is only determined by the inductive bias, and independent from the training set. It is a property of the class of hypotheses considered with respect to the prediction task. This is called **bias**
- ϵ_V is the difference between the above minimum risk in $\mathcal{H}$ and the risk associated to the best predictor in $\mathcal{H}$ with respect to the training set: it is related to the fact that empirical risk minimization only provides an estimate of the best predictor achievable for the given inductive bias. It is a measure of how well the predictor computed from a particular training set approximates the best possible one. Its expectation with respect to all possible training sets is a measure of how much a predictor derived from a random training set may result in poorer performances with respect to the best possible one. This is called **variance**

The choice of $\mathcal{H}$ is subject to a bias-variance tradeoff: higher bias tend to induce lower variance, and vice versa (see Figure **??**).

- High bias and low variance implies that all predictors which can be obtained from different training sets tend to behave similarly, with a similar risk (low variance). However, all of them then to behave poorly (high bias), since $\mathcal{H}$ is too poor to include a satisfactory predictor for the task considered. This results into underfitting
- Low bias and high variance implies that lot of predictors are available in $\mathcal{H}$, and among them a good one is usually avaliable (low bias). However, quite different predictors can be obtained from different training sets, which implies that it may easily happen that, while a very good performance can be obtained on the training set, the resulting predictor can behave quite differently and more poorly that the best possible one, which implies overfitting

See Figure **??** for an illustration. If we choose a very large space $\mathcal{H}$, then the approximation term will become small (the Bayes classifier might even be contained in $\mathcal{H}$ or can be approximated closely by some element in it). The estimation error, however, will be rather large in this case: the space $\mathcal{H}$ will contain complex functions which will lead to overfitting. The opposite effect will happen if the function class $\mathcal{H}$ is very small.

Learning theory studies how rich we can make the class $\mathcal{H}$ while still maintaining a reasonable estimation error. In many cases, empirical research focuses on designing good classes of predictors for a given domain. Here, "good" means classes for which the approximation error should not be excessively high. The idea is that, even if we are not experts and do not know how to construct the optimal predictor, we still have some preliminary knowledge of the specific problem, which allows us to design classes of predictors for which both the approximation error and the estimation error are not too high.

In practice, one typically has a set of predictors with associated sets of hyper-parameters: a type of predictor and an assignment of values to the corresponding hyper-parameters define a class of parametric predictors, from which a specific predictor is selected through a learning algorithm such as ERM.

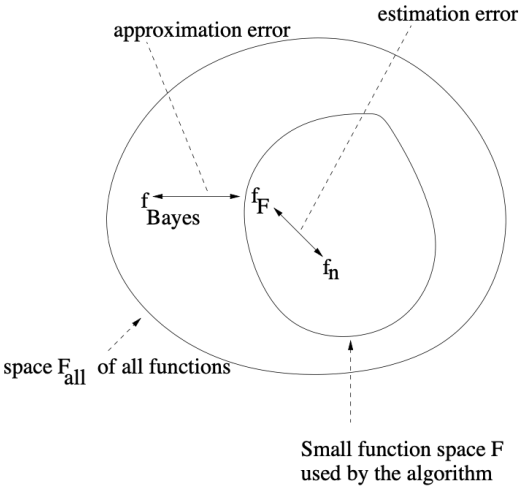

Figure 2: Illustration of estimation and approximation error.

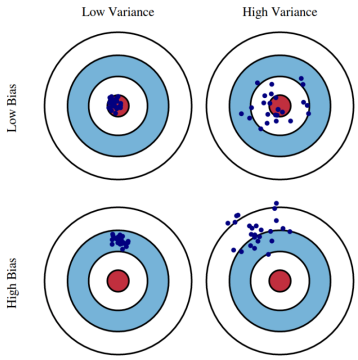

Figure 3: Graphical representation of bias and variance

The identification of this class, by evaluating and selecting a type of predictor and the values of the corresponding hyper-parameters, is called model selection.

The Vapnik-Červonenkis Dimension

The *Vapnik-Červonenkis dimension* (VC-dimension) is a measure of the complexity of a given hypothesis class. The measure is introduced to characterize infinite hypothesis classes in terms of their learnability.

We know that finite hypothesis classes are PAC-learnable and that the sample complexity depends on the size of the class. However, there also exist infinite hypothesis classes that are PAC-learnable, such as the class of threshold functions on real numbers

$$\mathcal{H}_\theta = \{\mathbb{1}[x < \theta]; \theta \in \mathbb{R}\}.$$

This hypothesis class is PAC-learnable with a sample complexity of $m_{\mathcal{H}}(\varepsilon, \delta) \leq \lceil \frac{1}{\varepsilon} \log \frac{2}{\delta} \rceil$.

Thus, finiteness is sufficient but not necessary for learnability. To define a more general and useful measure of complexity, we first need a few other definitions.

Given a subset $C = \{c_1, ..., c_m\} \subset \mathcal{X}$ of $\mathcal{X}$, we define the *restriction* $\mathcal{H}_C$ of $\mathcal{H}$ to C as the set of functions $f : C \mapsto \{0, 1\}$ that can be derived from predictors in $\mathcal{H}$ (i.e., such that for each $f \in C$ there exists a predictor

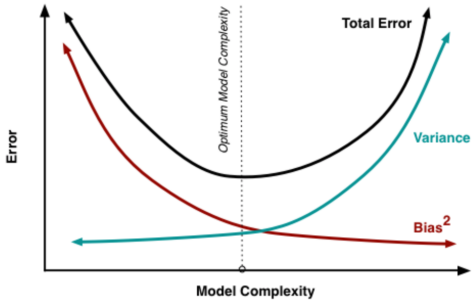

Figure 4: Bias and variance vs model complexity

$h \in \mathcal{H}$ for which $f(c_i) = h(c_i), i = 1, \ldots, m$). If we describe each function from C to $\{0,1\}$ as a vector in $\{0,1\}^{|C|}$, we can formally write it as

$$\mathcal{H}_C = \{(h(c_1), \ldots h(c_m)) : h \in \mathcal{H}\}.$$

If $\mathcal{H}_C$ is the set of all functions from C to $\{0,1\}$ (and thus $|\mathcal{H}_C| = 2^{|C|}$), this means that for every binary labeling of the elements of C (and thus for every separation of the elements into two distinct classes, or even for every possible binary classification task on C), there exists a predictor in $\mathcal{H}$ that separates the two classes, in the sense that it correctly predicts the target values of each element c_i. In this case, we say that $\mathcal{H}$ *shatters* C.

For example, consider the class $\mathcal{H}_\theta$ of threshold functions introduced above. Consider a set $C = \{c_1\}$: then, if we set $\theta = c_1 + 1$, we have $h_\theta(c_1) = 1$, and if we take $\theta = c_1 - 1$, we have $h_\theta(c_1) = 0$. Therefore, $\mathcal{H}_C$ is the set of all functions from C to $\{0,1\}$, and $\mathcal{H}$ shatters C. If we now consider a set $C = \{c_1, c_2\}$, where $c_1 \leq c_2$, no $h \in \mathcal{H}$ can represent the labeling $c_1 = 0, c_2 = 1$, because any threshold that assigns label 0 to c_1 must also assign label 0 to c_2. Therefore, not all functions from C to $\{0,1\}$ are included in $\mathcal{H}_C$, so C is not shattered by $\mathcal{H}$.

We now define the VC-Dimension $\text{VCdim}(\mathcal{H})$ of a hypothesis class $\mathcal{H}$ as the size of the largest subset $C \subset \mathcal{X}$ that is shattered by $\mathcal{H}$.

The motivation behind this definition is the following. From the No-Free-Lunch theorem, we know that the set of all functions from a domain to $\{0,1\}$ is not PAC-learnable. However, the proof of this statement is based on the assumption that we are considering all possible functions: it is reasonable to assume that introducing limitations on the hypothesis class might bring advantages.

To illustrate the concept of VC-Dimension, let us consider a few examples:

- Let $\mathcal{H}^{\text{thr}}$ be the class of threshold functions on $\mathbb{R}$ (tahat is functions h_θ such that $h_\theta(x) = 1$ if $x \geq \theta$ and $h_\theta(x) = 0$ otherwise). It is easy to see that for an arbitrary set $C = c_1$ of size 1, $\mathcal{H}_{\text{thr}}$ shatters C (just set $\theta = c_1$ if c_1 has label 1 and $\theta = c_1 + 1$ if the label is 0); therefore, $\text{VCdim}(\mathcal{H}^{\text{thr}}) \geq 1$. It is also possible to check that there exist sets $C = \{c_1, c_2\}$ of size 2 that $\mathcal{H}$ does not shatter (consider the case $c_1 > c_2$ with labeling $c_1 = 1, c_2 = 0$). Therefore, we conclude that $\text{VCdim}(\mathcal{H}^{\text{thr}}) = 1$.

- Let $\mathcal{H}^{\text{rect}}$ be the set of all axis-aligned rectangles in the Euclidean plane. If we want to prove that a certain number d is indeed the VC-Dimension of a hypothesis class $\mathcal{H}^{\text{rect}}$, we need to prove two things: $\text{VCdim}(\mathcal{H}^{\text{rect}}) \geq d$ and $\text{VCdim}(\mathcal{H}^{\text{rect}}) < d + 1$.

  In this case, $\text{VCdim}(\mathcal{H}^{\text{rect}}) = 4$. For the first inequality, we simply need to find a set of 4 points shattered by axis-aligned rectangles. Consider 4 equidistant points on a circle.

  It is easy to see (Figure **??**) that any partition of the 4 points into positive (green) and negative (red) can be shattered by an axis-aligned rectangle.

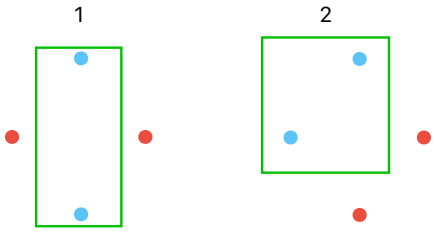

Figure 5: The two possible cases of shattering a set of 4 elements using axis-aligned rectangles.

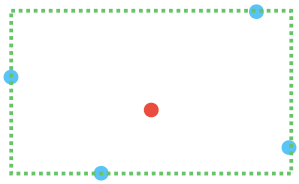

Figure 6: The impossibility of shattering a set of 5 elements using axis-aligned rectangles.

To prove that $\text{VCdim}(\mathcal{H}^{\text{rect}}) < 5$, we need to show that *no* set of size 5 is shattered. To this end, consider an arbitrary set of 5 points (Figure **??**) and their bounding box (indicated by the dashed green rectangle). There must be at least one point inside the bounding box (or on one of its sides). Now, if we label the other 4 points as positive and the fifth as negative, any axis-aligned rectangle that contains all the positive points must also contain the negative point. Therefore, no set of size 5 is shattered by $\mathcal{H}^{\text{rect}}$.

Let $\mathcal{H}^{\text{int}}$ be the class of intervals on $\mathbb{R}$, precisely $\mathcal{H}^{\text{int}} = \{h_{a,b} : a, b \in \mathbb{R}, a < b\}$, where $h_{a,b} : \mathbb{R} \to \{0, 1\}$ is a function such that $h_{a,b}(x) = \mathbb{1}[a < x < b]$. Consider the set $C = \{1, 2\}$. In this case, $\mathcal{H}^{\text{int}}$ shatters C (Fig. **??**), and therefore $\text{VCdim}(\mathcal{H}^{\text{int}}) \geq 2$. Now, take an arbitrary set $C = \{c_1, c_2, c_3\}$ and assume without loss of generality that $c_1 \leq c_2 \leq c_3$. In this case, the labeling $(1, 0, 1)$ cannot be obtained from an interval, so $\mathcal{H}^{\text{int}}$ does not shatter C. Therefore, we conclude that $\text{VCdim}(\mathcal{H}^{\text{int}}) = 2$.

- Let $\mathcal{H}^{\text{fin}}$ be a finite class. As observed, in order to shatter a set C we need $2^{|C|}$ predictors (since $2^{|C|}$ is the number of different labelings). Also, observe that $|\mathcal{H}^{\text{fin}}_C| \leq |\mathcal{H}^{\text{fin}}|$. As a consequence, C cannot be shattered by $\mathcal{H}^{\text{fin}}$ if $|\mathcal{H}^{\text{fin}}| < 2^{|C|}$, since this implies $|\mathcal{H}^{\text{fin}}_C| < 2^{|C|}$. This results into $\text{VCdim}(\mathcal{H}^{\text{fin}}) \leq \log_2|\mathcal{H}|$. The PAC learnability of finite classes then derives from the more general property PAC learnability of classes with finite VC-dimension. However, note that the VC-dimension of a finite class $\mathcal{H}^{\text{fin}}$ can be significantly smaller than $\log_2(|\mathcal{H}^{\text{fin}}|)$. For example, let $\mathcal{X} = \{1, \ldots, k\}$ for some integer k, and consider the class of threshold functions on $\mathcal{H}$. Then, $|\mathcal{H}| = k$ but $\text{VCdim}(\mathcal{H}) = 1$. Since k can be arbitrarily large, the difference between $\log_2(|\mathcal{H}|)$ and $\text{VCdim}(\mathcal{H})$ can be arbitrarily large.

Figure 7: Shattering of a 2-element set using intervals.

The main reason for interest in studying the VC-dimension is the existence of the so-called **Fundamental Theorem of Statistical Learning**, which provides a direct connection between the VC-dimension and the PAC learnability of a hypothesis class.

Theorem 2 (Fundamental Theorem of Statistical Learning)**.** *Let $\mathcal{H}$ be a class of hypotheses $h : \mathcal{X} \to \{0, 1\}$ for binary classification, and let the $0-1$ loss be the considered cost function. Then, the following statements are equivalent:*

1. *$\mathcal{H}$ has a finite VC-dimension.*
2. *$\mathcal{H}$ is agnostic PAC-learnable, and there exist constants $c_1 < c_2$ such that its sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$ is upper and lower bounded as*
$$\frac{c_1}{\varepsilon^2}\left(d + \ln\frac{1}{\delta}\right) \leq m_{\mathcal{H}}(\varepsilon, \delta) \leq \frac{c_2}{\varepsilon^2}\left(d + \ln\frac{1}{\delta}\right)$$
Moreover, this property holds also when ERM is applied (that is, it is a successful agnostic PAC-learning algorithm for $\mathcal{H}$).
3. *$\mathcal{H}$ is PAC-learnable, and its sample complexity $m_{\mathcal{H}}(\varepsilon, \delta)$ is upper and lower bounded as*
$$\frac{c_1}{\varepsilon}\left(d + \ln\frac{1}{\delta}\right) \leq m_{\mathcal{H}}(\varepsilon, \delta) \leq \frac{c_2}{\varepsilon}\left(d + \ln\frac{1}{\delta}\right)$$
Moreover, this property holds also when ERM is applied (that is, it is a successful PAC-learner for $\mathcal{H}$).