Probably Approximately Correct Learning

Finite Hypothesis Class

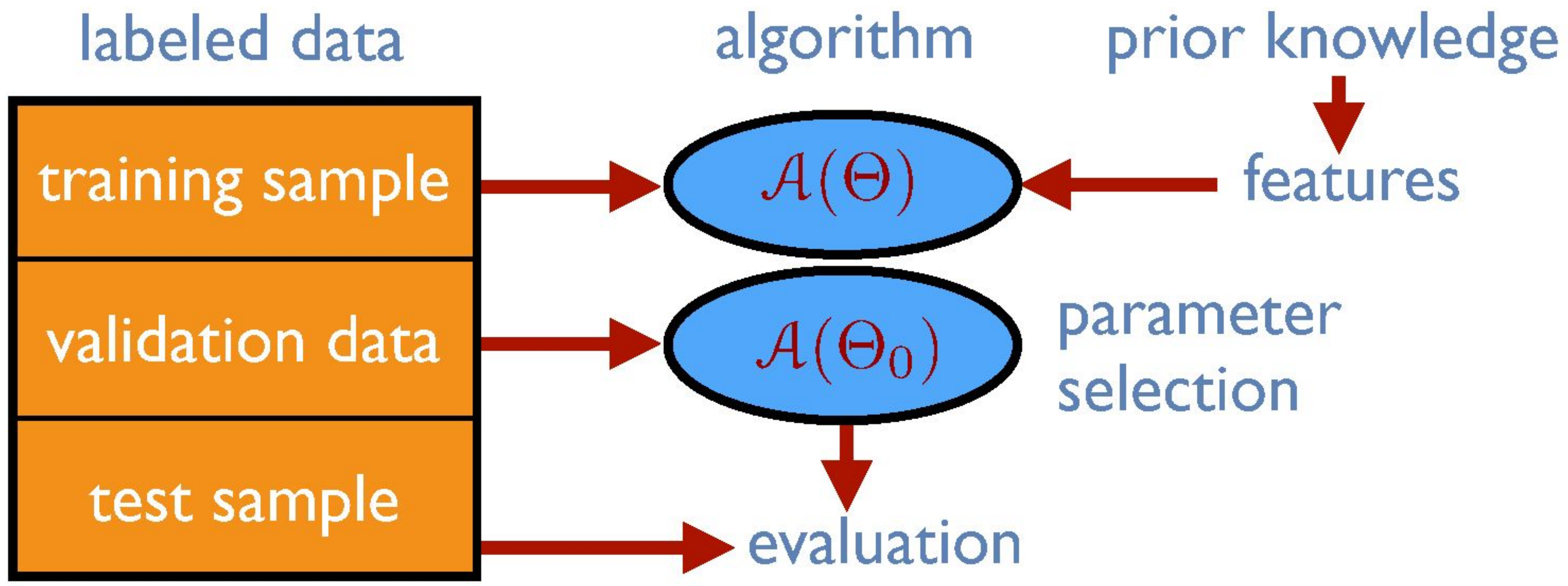
Motivation

- Some computational learning questions
  - What can be learned efficiently?
  - What is inherently hard to learn?
  - A general model of learning?
- Complexity
  - Computational complexity: time and space.
  - Sample complexity: amount of training data needed to learn successfully.
  - Mistake bounds: number of mistakes before learning successfully.

This lecture

- PAC Model
- Sample complexity, finite *H*, consistent case
- Sample complexity, finite *H*, inconsistent case

Learning Stages



Definitions

- Spaces: input space X, output space Y.
- Loss function: $L\colon Y \times Y \to \mathbb{R}$.
  - $L(\widehat{y}, y)$: cost of predicting $\widehat{y}$ instead of y.
  - binary classification: 0-1 loss, $L(y, y') = 1_{y \neq y'}$.
  - regression: $Y \subseteq \mathbb{R}$, $l(y, y') = (y' - y)^2$.
- Hypothesis set: $H \subseteq Y^X$, subset of functions out of which the learner selects his hypothesis.
  - depends on features.
  - represents prior knowledge about task.

Supervised Learning Set-Up

- Training data: sample S of size m drawn i.i.d. from $X \times Y$ according to distribution D:

$$S = ((x_1, y_1), \ldots, (x_m, y_m)).$$

- Problem: find hypothesis $h \in H$ with small generalization error.
    - deterministic case: output label deterministic function of input, $y = f(x)$.
    - stochastic case: output probabilistic function of input.

Errors

- Generalization error: for $h \in H$, it is defined by

$$R(h) = \mathop{\mathrm{E}}_{(x,y)\sim D}[L(h(x), y)].$$

- Empirical error: for $h \in H$ and sample S, it is

$$\widehat{R}(h) = \frac{1}{m} \sum_{i=1}^{m} L(h(x_i), y_i).$$

- Bayes error:

$$R^\star = \inf_{\substack{h \\ h \text{ measurable}}} R(h).$$

  - in deterministic case, $R^\star = 0$.

Definitions and Notation

- X: set of all possible instances or examples, e.g., the set of all men and women characterized by their height and weight.

c(x) (foundations) == f(x) (understanding)

- $c: X \rightarrow \{0, 1\}$: the target concept to learn; can be identified with its support $\{x \in X: c(x) = 1\}$.
- C: concept class, a set of target concepts c.
- D: target distribution, a fixed probability distribution over X. Training and test examples are drawn according to D.

Definitions and Notation

- S : training sample.
- H : set of concept hypotheses, e.g., the set of all linear classifiers.
- The learning algorithm receives sample S and selects a hypothesis h_S from H approximating c .

Errors

- True error or generalization error of h with respect to the target concept c and distribution D:

$$R(h) = \Pr_{x \sim D}[h(x) \neq c(x)] = \mathop{\mathrm{E}}_{x \sim D}[1_{h(x) \neq c(x)}].$$

- Empirical error: average error of h on the training sample S drawn according to distribution D,

$$\widehat{R}_S(h) = \Pr_{x \sim \widehat{D}}[h(x) \neq c(x)] = \mathop{\mathrm{E}}_{x \sim \widehat{D}}[1_{h(x) \neq c(x)}] = \frac{1}{m}\sum_{i=1}^{m} 1_{h(x_i) \neq c(x_i)}.$$

- Note: $R(h) = \mathop{\mathrm{E}}_{S \sim D^m}\left[\widehat{R}_S(h)\right].$

This lecture

- PAC Model
- Sample complexity, finite *H*, consistent case
- Sample complexity, finite *H*, inconsistent case

Finite Hypothesis Classes

The simplest type of restriction on a class is imposing an upper bound on its size (that is, the number of predictors h in $\mathcal{H}$). In this section, we show that if $\mathcal{H}$ is a finite class then $\mathrm{ERM}_{\mathcal{H}}$ will not overfit, provided it is based on a sufficiently large training sample (this size requirement will depend on the size of $\mathcal{H}$).

Limiting the learner to prediction rules within some finite hypothesis class may be considered as a reasonably mild restriction. For example, $\mathcal{H}$ can be the set of all predictors that can be implemented by a C++ program written in at most 10^9 bits of code. In our papayas example, we mentioned previously the class of axis aligned rectangles. While this is an infinite class, if we discretize the representation of real numbers, say, by using a 64 bits floating-point representation, the hypothesis class becomes a finite class.

Generalization Error
True Error
Risk

- **Measures of success:** We define the *error of a classifier* to be the probability that it does not predict the correct label on a random data point generated by the aforementioned underlying distribution. That is, the error of h is the probability to draw a random instance x, according to the distribution $\mathcal{D}$, such that $h(x)$ does not equal $f(x)$.

  Formally, given a domain subset,[2] $A \subset \mathcal{X}$, the probability distribution, $\mathcal{D}$, assigns a number, $\mathcal{D}(A)$, which determines how likely it is to observe a point $x \in A$. In many cases, we refer to A as an event and express it using a function $\pi : \mathcal{X} \to \{0, 1\}$, namely, $A = \{x \in \mathcal{X} : \pi(x) = 1\}$. In that case, we also use the notation $\mathbb{P}_{x \sim \mathcal{D}}[\pi(x)]$ to express $\mathcal{D}(A)$.

  We define the error of a prediction rule, $h : \mathcal{X} \to \mathcal{Y}$, to be

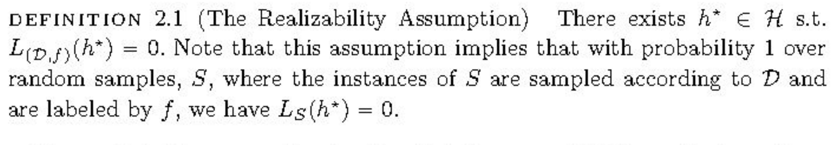
$$L_{\mathcal{D},f}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{P}}_{x \sim \mathcal{D}}[h(x) \neq f(x)] \stackrel{\text{def}}{=} \mathcal{D}(\{x : h(x) \neq f(x)\}). \qquad (2.1)$$

  That is, the error of such h is the probability of randomly choosing an example x for which $h(x) \neq f(x)$. The subscript $(\mathcal{D}, f)$ indicates that the error is measured with respect to the probability distribution $\mathcal{D}$ and the

correct labeling function f. We omit this subscript when it is clear from the context. $L_{(\mathcal{D},f)}(h)$ has several synonymous names such as the *generalization error*, the *risk*, or the *true error* of h, and we will use these names interchangeably throughout the book. We use the letter L for the error, since we view this error as the *loss* of the learner. We will later also discuss other possible formulations of such loss.

Understanding

Realizability Assumption

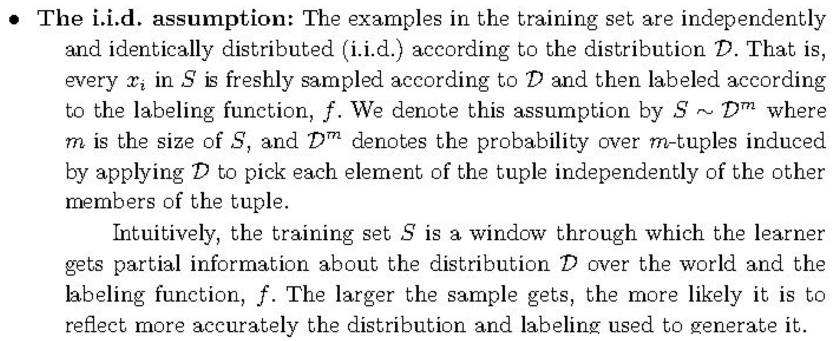

DEFINITION 2.1 (The Realizability Assumption) There exists $h^* \in \mathcal{H}$ s.t. $L_{(\mathcal{D},f)}(h^*) = 0$. Note that this assumption implies that with probability 1 over random samples, S, where the instances of S are sampled according to $\mathcal{D}$ and are labeled by f, we have $L_S(h^*) = 0$.

$=$

Foundations

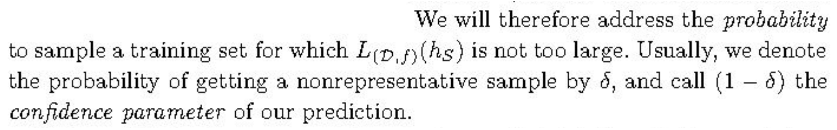
Consistent Case (Συνεπής περίπτωση): the hypothesis h_S returned by the algorithm is always consistent, that is, it admitted no error on the training sample S.

iid


- **The i.i.d. assumption:** The examples in the training set are independently and identically distributed (i.i.d.) according to the distribution $\mathcal{D}$. That is, every x_i in S is freshly sampled according to $\mathcal{D}$ and then labeled according to the labeling function, f. We denote this assumption by $S \sim \mathcal{D}^m$ where m is the size of S, and $\mathcal{D}^m$ denotes the probability over m-tuples induced by applying $\mathcal{D}$ to pick each element of the tuple independently of the other members of the tuple.

  Intuitively, the training set S is a window through which the learner gets partial information about the distribution $\mathcal{D}$ over the world and the labeling function, f. The larger the sample gets, the more likely it is to reflect more accurately the distribution and labeling used to generate it.

δ (Probably)


We will therefore address the *probability* to sample a training set for which $L_{(\mathcal{D},f)}(h_S)$ is not too large. Usually, we denote the probability of getting a nonrepresentative sample by δ, and call $(1-\delta)$ the *confidence parameter* of our prediction.

1 - δ : Confidence

PAC Model

(Valiant, 1984)

- PAC learning: Probably Approximately Correct learning.
- Definition: concept class C is PAC-learnable if there exists a learning algorithm L such that:
  - for all $c \in C, \epsilon > 0, \delta > 0,$ and all distributions D,

$$\Pr_{S \sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta,$$

  - for samples S of size $m = poly(1/\epsilon, 1/\delta)$ for a fixed polynomial.

Remarks

- Concept class C is known to the algorithm.
- Distribution-free model: no assumption on D.
- Both training and test examples drawn $\sim D$.
- Probably: confidence $1-\delta$.
- Approximately correct: accuracy $1-\epsilon$.
- Efficient PAC-learning: L runs in time $poly(1/\epsilon, 1/\delta)$.
- What about the cost of the representation of $c \in C$?

PAC Model - New Definition

- Computational representation:
  - cost for $x \in X$ in $O(n)$.
  - cost for $c \in C$ in $O(size(c))$.
- Extension: running time.

$$O(poly(1/\epsilon, 1/\delta)) \longrightarrow O(poly(1/\epsilon, 1/\delta, n, size(c))).$$

Empirical Risk Minimization

As mentioned earlier, a learning algorithm receives as input a training set S, sampled from an unknown distribution $\mathcal{D}$ and labeled by some target function f, and should output a predictor $h_S : \mathcal{X} \to \mathcal{Y}$ (the subscript S emphasizes the fact that the output predictor depends on S). The goal of the algorithm is to find h_S that minimizes the error with respect to the unknown $\mathcal{D}$ and f.

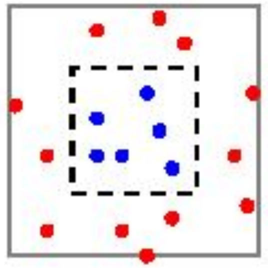
Since the learner does not know what $\mathcal{D}$ and f are, the true error is not directly available to the learner. A useful notion of error that can be calculated by the learner is the *training error* – the error the classifier incurs over the training sample:

$$L_S(h) \stackrel{\text{def}}{=} \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}, \tag{2.2}$$

where $[m] = \{1, \ldots, m\}$.

The terms *empirical error* and *empirical risk* are often used interchangeably for this error.

Since the training sample is the snapshot of the world that is available to the learner, it makes sense to search for a solution that works well on that data. This learning paradigm – coming up with a predictor h that minimizes $L_S(h)$ – is called *Empirical Risk Minimization* or ERM for short.



$$h_S(x) = \begin{cases} y_i & \text{if } \exists i \in [m] \text{ s.t. } x_i = x \\ 0 & \text{otherwise.} \end{cases} \tag{2.3}$$

$$\mathrm{ERM}_{\mathcal{H}}(S) \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h),$$

$$h_S \in \operatorname*{argmin}_{h \in \mathcal{H}} L_S(h). \tag{2.4}$$

ϵ (Approximately)

commonly denoted by ϵ. We interpret the event $L_{(\mathcal{D},f)}(h_S) > \epsilon$ as a failure of the learner, while if $L_{(\mathcal{D},f)}(h_S) \le \epsilon$ we view the output of the algorithm as an approximately correct predictor. Therefore (fixing some labeling function $f : \mathcal{X} \to \mathcal{Y}$), we are interested in upper bounding the probability to sample m-tuple of instances that will lead to failure of the learner. Formally, let $S|_x = (x_1, \ldots, x_m)$ be the instances of the training set. We would like to upper bound

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}).$$

Let $\mathcal{H}_B$ be the set of "bad" hypotheses, that is,

$$\mathcal{H}_B = \{h \in \mathcal{H} : L_{(\mathcal{D},f)}(h) > \epsilon\}.$$

In addition, let

$$M = \{S|_x : \exists h \in \mathcal{H}_B, L_S(h) = 0\}$$

be the set of misleading samples: Namely, for every $S|_x \in M$, there is a "bad" hypothesis, $h \in \mathcal{H}_B$, that looks like a "good" hypothesis on $S|_x$. Now, recall that we would like to bound the probability of the event $L_{(\mathcal{D},f)}(h_S) > \epsilon$. But, since the realizability assumption implies that $L_S(h_S) = 0$, it follows that the event $L_{(\mathcal{D},f)}(h_S) > \epsilon$ can only happen if for some $h \in \mathcal{H}_B$ we have $L_S(h) = 0$. In other words, this event will only happen if our sample is in the set of misleading samples, M. Formally, we have shown that

$$\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\} \subseteq M .$$

Note that we can rewrite M as

$$M = \bigcup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}. \tag{2.5}$$

Hence,

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}) \le \mathcal{D}^m(M) = \mathcal{D}^m(\cup_{h \in \mathcal{H}_B} \{S|_x : L_S(h) = 0\}). \tag{2.6}$$

Next, we upper bound the right-hand side of the preceding equation using the *union bound* – a basic property of probabilities.

LEMMA 2.2 (Union Bound) *For any two sets A, B and a distribution $\mathcal{D}$ we have*

Union Bound

$$\mathcal{D}(A \cup B) \le \mathcal{D}(A) + \mathcal{D}(B).$$

Applying the union bound to the right-hand side of Equation (2.6) yields

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}) \le \sum_{h \in \mathcal{H}_B} \mathcal{D}^m(\{S|_x : L_S(h) = 0\}). \tag{2.7}$$

Next, let us bound each summand of the right-hand side of the preceding inequality. Fix some "bad" hypothesis $h \in \mathcal{H}_B$. The event $L_S(h) = 0$ is equivalent to the event $\forall i, h(x_i) = f(x_i)$. Since the examples in the training set are sampled i.i.d. we get that

$$\begin{aligned}\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) &= \mathcal{D}^m(\{S|_x : \forall i, h(x_i) = f(x_i)\}) \\ &= \prod_{i=1}^m \mathcal{D}(\{x_i : h(x_i) = f(x_i)\}).\end{aligned} \tag{2.8}$$

For each individual sampling of an element of the training set we have

correct probability

$$\mathcal{D}(\{x_i : h(x_i) = y_i\}) = 1 - L_{(\mathcal{D},f)}(h) \le 1 - \epsilon,$$

where the last inequality follows from the fact that $h \in \mathcal{H}_B$. Combining the previous equation with Equation (2.8) and using the inequality $1 - \epsilon \le e^{-\epsilon}$ we obtain that for every $h \in \mathcal{H}_B$,

Bernoulli's inequality

$$\mathcal{D}^m(\{S|_x : L_S(h) = 0\}) \le (1 - \epsilon)^m \le e^{-\epsilon m}. \tag{2.9}$$

Combining this equation with Equation (2.7) we conclude that

$$\mathcal{D}^m(\{S|_x : L_{(\mathcal{D},f)}(h_S) > \epsilon\}) \le |\mathcal{H}_B| \, e^{-\epsilon m} \le |\mathcal{H}| \, e^{-\epsilon m}.$$

A graphical illustration which explains how we used the union bound is given in Figure 2.1.

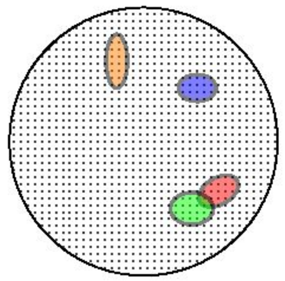


Figure 2.1 Each point in the large circle represents a possible m-tuple of instances. Each colored oval represents the set of "misleading" m-tuple of instances for some "bad" predictor $h \in \mathcal{H}_B$. The ERM can potentially overfit whenever it gets a misleading training set S. That is, for some $h \in \mathcal{H}_B$ we have $L_S(h) = 0$. Equation (2.9) guarantees that for each individual bad hypothesis, $h \in \mathcal{H}_B$, at most $(1 - \epsilon)^m$-fraction of the training sets would be misleading. In particular, the larger m is, the smaller each of these colored ovals becomes. The union bound formalizes the fact that the area representing the training sets that are misleading with respect to some $h \in \mathcal{H}_B$ (that is, the training sets in M) is at most the sum of the areas of the colored ovals. Therefore, it is bounded by $|\mathcal{H}_B|$ times the maximum size of a colored oval. Any sample S outside the colored ovals cannot cause the ERM rule to overfit.

COROLLARY 2.3 *Let $\mathcal{H}$ be a finite hypothesis class. Let $\delta \in (0,1)$ and $\epsilon > 0$*

$$|H| e^{-\varepsilon m} \le \delta$$
$$e^{-\varepsilon m} \le \frac{\delta}{|H|}$$
$$-\varepsilon m \le \log(\delta/|H|)$$
$$\varepsilon m \ge \log(|H|/\delta)$$
$$m \ge \log(|H|/\delta)/\varepsilon$$

and let m be an integer that satisfies

$$m \ge \frac{\log(|\mathcal{H}|/\delta)}{\epsilon}.$$

Then, for any labeling function, f, and for any distribution, $\mathcal{D}$, for which the realizability assumption holds (that is, for some $h \in \mathcal{H}$, $L_{(\mathcal{D},f)}(h) = 0$), with probability of at least $1 - \delta$ over the choice of an i.i.d. sample S of size m, we have that for every ERM hypothesis, h_S, it holds that

$$L_{(\mathcal{D},f)}(h_S) \le \epsilon.$$

The preceeding corollary tells us that for a sufficiently large m, the ERM$_\mathcal{H}$ rule over a finite hypothesis class will be *probably* (with confidence $1-\delta$) *approximately* (up to an error of ϵ) correct. In the next chapter we formally define the model of Probably Approximately Correct (PAC) learning.

Correct

Remarks

- The algorithm can be ERM if problem realizable.
- Error bound linear in $\frac{1}{m}$ and only logarithmic in $\frac{1}{\delta}$.
- $\log_2 |H|$ is the number of bits used for the representation of H.
- Bound is loose for large $|H|$.
- Uninformative for infinite $|H|$.

Sample Complexity
The function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ determines the *sample complexity* of learning $\mathcal{H}$: that is, how many examples are required to guarantee a probably approximately correct solution. The sample complexity is a function of the accuracy (ϵ) and confidence (δ) parameters. It also depends on properties of the hypothesis class $\mathcal{H}$ – for example, for a finite class we showed that the sample complexity depends on log the size of $\mathcal{H}$.

Note that if $\mathcal{H}$ is PAC learnable, there are many functions $m_{\mathcal{H}}$ that satisfy the requirements given in the definition of PAC learnability. Therefore, to be precise, we will define the sample complexity of learning $\mathcal{H}$ to be the "minimal function," in the sense that for any ϵ, δ, $m_{\mathcal{H}}(\epsilon, \delta)$ is the minimal integer that satisfies the requirements of PAC learning with accuracy ϵ and confidence δ.

Let us now recall the conclusion of the analysis of finite hypothesis classes from the previous chapter. It can be rephrased as stating:

COROLLARY 3.2 *Every finite hypothesis class is PAC learnable with sample complexity*

$$m_{\mathcal{H}}(\epsilon, \delta) \le \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil .$$

uniform convergence bound (consistent case)

There are infinite classes that are learnable as well (see, for example, Exercise 3). Later on we will show that what determines the PAC learnability of a class is not its finiteness but rather a combinatorial measure called the *VC dimension*.

This lecture

- PAC Model
- Sample complexity, finite *H*, consistent case
- Sample complexity, finite *H*, inconsistent case

Non - realizable, no $h_s(x)$ with $L(h_s(x))=0$
$\longrightarrow$ Real-world, hard problems

Inconsistent Case

- No $h \in H$ is a consistent hypothesis.
- The typical case in practice: difficult problems, complex concept class.
- But, inconsistent hypotheses with a small number of errors on the training set can be useful.
- Need a more powerful tool: Hoeffding's inequality.

δύσκολη απόδειξη, είναι στα παραρτήματα των βιβλίων

Hoeffding's Inequality

- Corollary: for any $\epsilon > 0$ and any hypothesis $h : X \to \{0, 1\}$ the following inequalities holds:

$$\Pr[R(h) - \widehat{R}(h) \geq \epsilon] \leq e^{-2m\epsilon^2}$$
$$\Pr[\widehat{R}(h) - R(h) \geq \epsilon] \leq e^{-2m\epsilon^2}.$$

- Combining these one-sided inequalities yields

$$\Pr[|R(h) - \widehat{R}(h)| \geq \epsilon] \leq 2e^{-2m\epsilon^2}.$$

Application to Learning Algorithm?

- Can we apply that bound to the hypothesis h_S returned by our learning algorithm when training on sample S?
- No, because h_S is not a fixed hypothesis, it depends on the training sample. Note also that $\mathrm{E}[\widehat{R}(h_S)]$ is not a simple quantity such as $R(h_S)$.
- Instead, we need a bound that holds simultaneously for all hypotheses $h \in H$, a uniform convergence bound.

Generalization Bound - Finite *H*

- **Theorem**: let H be a finite hypothesis set, then, for any $\delta > 0$, with probability at least $1-\delta$,

$$\forall h \in H,\ R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{\log|H| + \log\frac{2}{\delta}}{2m}}.$$

- **Proof**: By the union bound,

$$\Pr\left[\max_{h\in H}\left|R(h) - \widehat{R}_S(h)\right| > \epsilon\right]$$
$$= \Pr\left[\left|R(h_1) - \widehat{R}_S(h_1)\right| > \epsilon \vee \ldots \vee \left|R(h_{|H|}) - \widehat{R}_S(h_{|H|})\right| > \epsilon\right]$$
$$\leq \sum_{h\in H} \Pr\left[\left|R(h) - \widehat{R}_S(h)\right| > \epsilon\right]$$

Hoeffding's Inequality

$$\leq 2|H|\exp(-2m\epsilon^2).$$

Recall

COROLLARY 3.2 *Every finite hypothesis class is PAC learnable with sample complexity*

$$m_{\mathcal{H}}(\epsilon,\delta) \;\le\; \left\lceil \frac{\log(|\mathcal{H}|/\delta)}{\epsilon} \right\rceil .$$

Finally, if we choose

$$m \ge \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2}$$

then

$$\mathcal{D}^m(\{S : \exists h \in \mathcal{H}, |L_S(h) - L_{\mathcal{D}}(h)| > \epsilon\}) \le \delta.$$

COROLLARY 4.6 *Let $\mathcal{H}$ be a finite hypothesis class, let Z be a domain, and let $\ell : \mathcal{H} \times Z \to [0,1]$ be a loss function. Then, $\mathcal{H}$ enjoys the uniform convergence property with sample complexity*

$$m^{UC}_{\mathcal{H}}(\epsilon,\delta) \le \left\lceil \frac{\log(2|\mathcal{H}|/\delta)}{2\epsilon^2} \right\rceil .$$

Furthermore, the class is agnostically PAC learnable using the ERM algorithm with sample complexity

$$m_{\mathcal{H}}(\epsilon,\delta) \le m^{UC}_{\mathcal{H}}(\epsilon/2,\delta) \le \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \right\rceil .$$

ε-αντιπροσωπευτικό training set

DEFINITION 4.1 (ϵ-representative sample) A training set S is called ϵ-representative (w.r.t. domain Z, hypothesis class $\mathcal{H}$, loss function ℓ, and distribution $\mathcal{D}$) if

$$\forall h \in \mathcal{H}, \quad |L_S(h) - L_{\mathcal{D}}(h)| \le \epsilon.$$

The next simple lemma states that whenever the sample is $(\epsilon/2)$-representative, the ERM learning rule is guaranteed to return a good hypothesis.

LEMMA 4.2 *Assume that a training set S is $\frac{\epsilon}{2}$-representative (w.r.t. domain Z, hypothesis class $\mathcal{H}$, loss function ℓ, and distribution $\mathcal{D}$). Then, any output of $\mathrm{ERM}_{\mathcal{H}}(S)$, namely, any $h_S \in \operatorname{argmin}_{h\in\mathcal{H}} L_S(h)$, satisfies*

$$L_{\mathcal{D}}(h_S) \;\le\; \min_{h\in\mathcal{H}} L_{\mathcal{D}}(h) + \epsilon.$$

The Finite Hypothesis Class is **Agnostic PAC Learnable**: can be PAC learnable with error on the training set (inconsistent case)

Agnostic learner: : A learner that **doesn't assume that contains an error-free hypothesis** and that simply finds the hypothesis with minimum training error (ERM)

$$\forall h \in H,\ R(h) \le \widehat{R}_S(h) + \sqrt{\frac{\log|H| + \log\frac{2}{\delta}}{2m}}.$$

$$m_{\mathcal{H}}(\epsilon,\delta) \le m^{UC}_{\mathcal{H}}(\epsilon/2,\delta) \le \left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \right\rceil.$$

Γιατί ισχύει η ανισότητα $m_{\mathcal{H}}(\epsilon,\delta) \le m^{UC}_{\mathcal{H}}(\epsilon/2,\delta)$

$m_H(\varepsilon,\delta)$ $\quad$ $m_H^{UC}(\frac{\varepsilon}{2},\delta)$

$$\frac{\log(\frac{|H|}{\delta})}{\cancel{\varepsilon}} \le \frac{\log(\frac{2|H|}{\delta})}{\cancel{2}(\frac{\varepsilon}{2})^{\cancel{2}}}$$

$$\varepsilon \log\left(\frac{|H|}{\delta}\right) \le 2\log(2) + 2\log\left(\frac{|H|}{\delta}\right)$$

$$(\varepsilon - 2) \le \frac{2\log(2)}{\log(\frac{|H|}{\delta})}$$

$\varepsilon \le 1$

$\varepsilon - 2 \le -1$

Με τί ισούται το σφάλμα $\sqrt{\frac{\log|H| + \log\frac{2}{\delta}}{2m}}$

αν για m βάλουμε το άνω φράγμα για το uniform convergence

$$\left\lceil \frac{2\log(2|\mathcal{H}|/\delta)}{\epsilon^2} \right\rceil$$

$$\sqrt{\frac{\cancel{\log(\frac{2|H|}{\delta})}}{2\cdot\frac{2\cancel{\log(\frac{2|H|}{\delta})}}{\varepsilon^2}}} = \sqrt{\frac{\varepsilon^2}{4}} = \frac{\varepsilon}{2}$$

Remarks

- Thus, for a finite hypothesis set, whp,

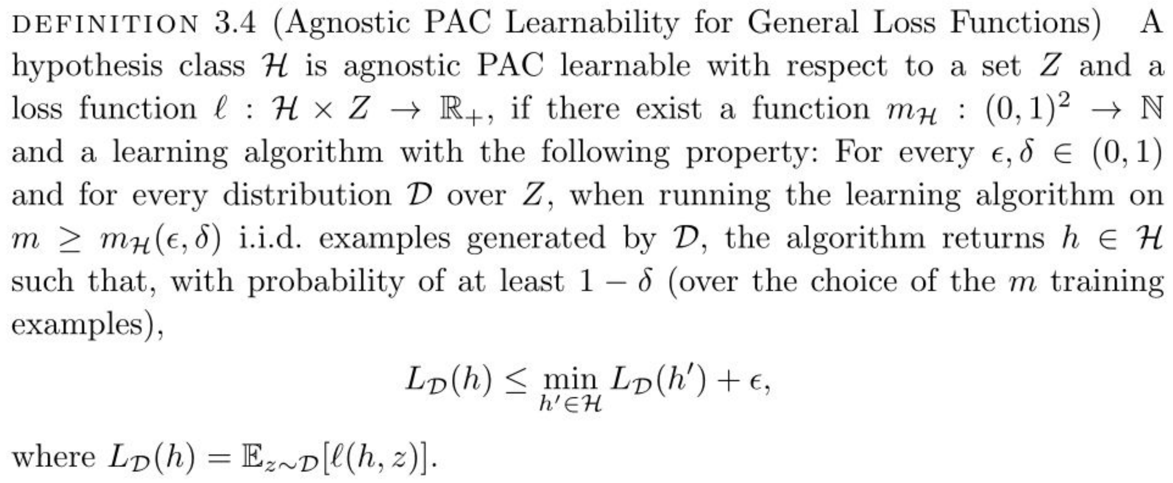
$$\forall h \in H, R(h) \leq \widehat{R}_S(h) + O\left(\sqrt{\frac{\log |H|}{m}}\right).$$

- Error bound in $O(\frac{1}{\sqrt{m}})$ (quadratically worse).
- $\log_2 |H|$ can be interpreted as the number of bits needed to encode H.
- Occam's Razor principle (theologian William of Occam): "plurality should not be posited without necessity".

Occam’s Razor

- Principle formulated by controversial theologian William of Occam: “plurality should not be posited without necessity”, rephrased as “the simplest explanation is best”;
  - invoked in a variety of contexts, e.g., syntax. Kolmogorov complexity can be viewed as the corresponding framework in information theory.
  - here, to minimize true error, choose the most parsimonious explanation (smallest $|H|$).
  - we will see later other applications of this principle.

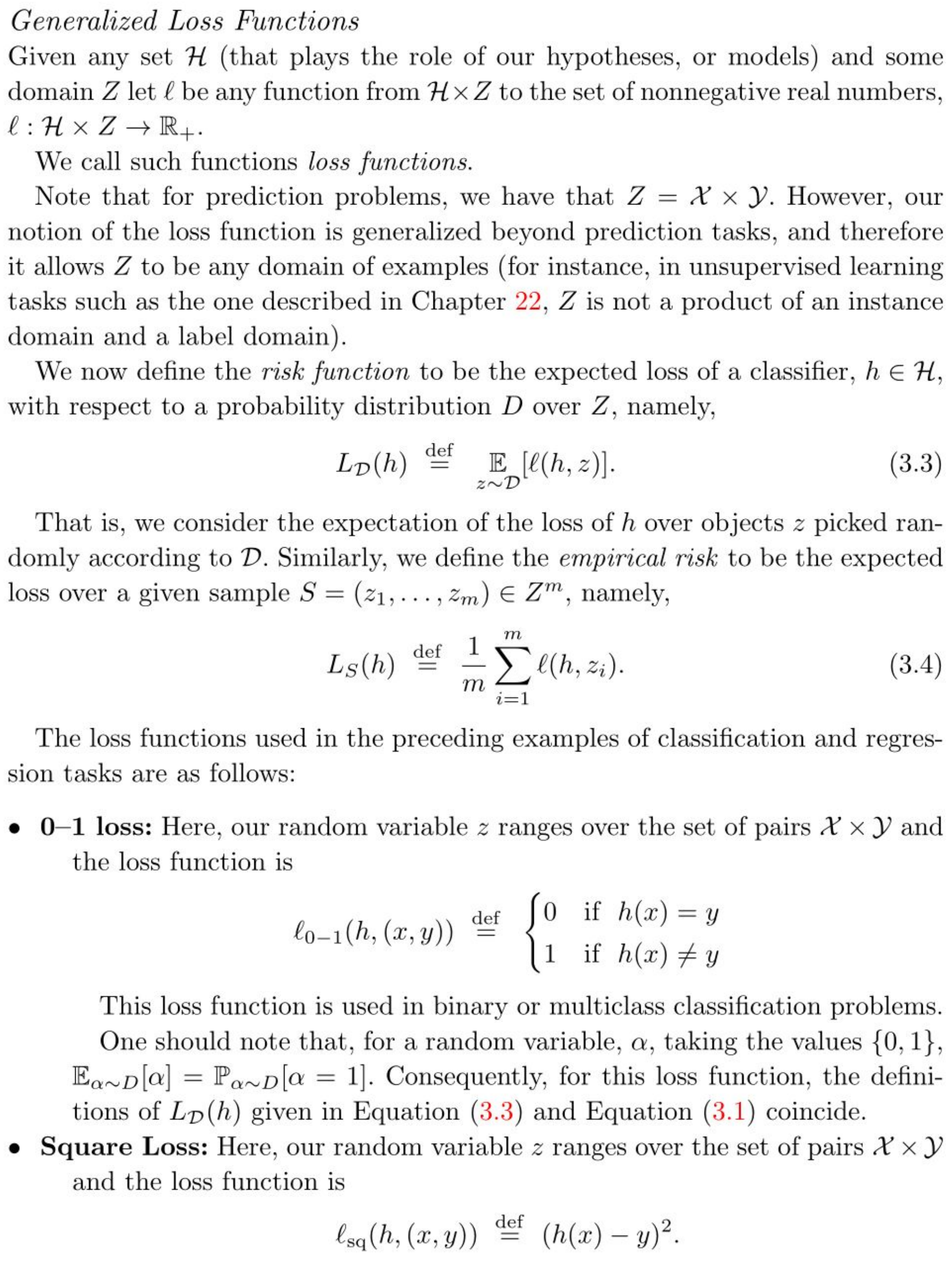
The Finite Hypothesis Class is
Agnostic PAC Learnable for General Loss functions



DEFINITION 3.4 (Agnostic PAC Learnability for General Loss Functions) A hypothesis class $\mathcal{H}$ is agnostic PAC learnable with respect to a set Z and a loss function $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$, if there exist a function $m_{\mathcal{H}} : (0,1)^2 \to \mathbb{N}$ and a learning algorithm with the following property: For every $\epsilon, \delta \in (0,1)$ and for every distribution $\mathcal{D}$ over Z, when running the learning algorithm on $m \geq m_{\mathcal{H}}(\epsilon, \delta)$ i.i.d. examples generated by $\mathcal{D}$, the algorithm returns $h \in \mathcal{H}$ such that, with probability of at least $1 - \delta$ (over the choice of the m training examples),

$$L_{\mathcal{D}}(h) \leq \min_{h' \in \mathcal{H}} L_{\mathcal{D}}(h') + \epsilon,$$

where $L_{\mathcal{D}}(h) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(h, z)]$.



Generalized Loss Functions
Given any set $\mathcal{H}$ (that plays the role of our hypotheses, or models) and some domain Z let ℓ be any function from $\mathcal{H} \times Z$ to the set of nonnegative real numbers, $\ell : \mathcal{H} \times Z \to \mathbb{R}_+$.

We call such functions *loss functions*.

Note that for prediction problems, we have that $Z = \mathcal{X} \times \mathcal{Y}$. However, our notion of the loss function is generalized beyond prediction tasks, and therefore it allows Z to be any domain of examples (for instance, in unsupervised learning tasks such as the one described in Chapter 22, Z is not a product of an instance domain and a label domain).

We now define the *risk function* to be the expected loss of a classifier, $h \in \mathcal{H}$, with respect to a probability distribution D over Z, namely,

$$L_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathop{\mathbb{E}}_{z \sim \mathcal{D}}[\ell(h, z)]. \tag{3.3}$$

That is, we consider the expectation of the loss of h over objects z picked randomly according to $\mathcal{D}$. Similarly, we define the *empirical risk* to be the expected loss over a given sample $S = (z_1, \ldots, z_m) \in Z^m$, namely,

$$L_S(h) \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} \ell(h, z_i). \tag{3.4}$$

The loss functions used in the preceding examples of classification and regression tasks are as follows:

- **0–1 loss:** Here, our random variable z ranges over the set of pairs $\mathcal{X} \times \mathcal{Y}$ and the loss function is

$$\ell_{0-1}(h, (x, y)) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } h(x) = y \\ 1 & \text{if } h(x) \neq y \end{cases}$$

  This loss function is used in binary or multiclass classification problems.

  One should note that, for a random variable, α, taking the values $\{0, 1\}$, $\mathbb{E}_{\alpha \sim D}[\alpha] = \mathbb{P}_{\alpha \sim D}[\alpha = 1]$. Consequently, for this loss function, the definitions of $L_{\mathcal{D}}(h)$ given in Equation (3.3) and Equation (3.1) coincide.

Classification (Binary, Multiclass)

- **Square Loss:** Here, our random variable z ranges over the set of pairs $\mathcal{X} \times \mathcal{Y}$ and the loss function is

$$\ell_{\text{sq}}(h, (x, y)) \stackrel{\text{def}}{=} (h(x) - y)^2.$$

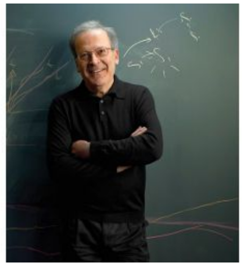
Regression

Lecture Summary

- C is PAC-learnable if $\exists L, \forall c \in C, \forall \epsilon, \delta > 0, m = P\left(\frac{1}{\epsilon}, \frac{1}{\delta}\right)$,
$$\Pr_{S \sim D^m}[R(h_S) \leq \epsilon] \geq 1 - \delta.$$
- Learning bound, finite H consistent case:
$$R(h) \leq \frac{1}{m}(\log |H| + \log \frac{1}{\delta}).$$
- Learning bound, finite H inconsistent case:
$$R(h) \leq \widehat{R}_S(h) + \sqrt{\frac{\log |H| + \log \frac{2}{\delta}}{2m}}.$$
- How do we deal with infinite hypothesis sets?

References

- Anselm Blumer, A. Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, Volume 36, Issue 4, 1989.
- Michael Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory*, MIT Press, 1994.
- Leslie G. Valiant. *A Theory of the Learnable*, Communications of the ACM 27(11):1134–1142 (1984). A theory of the Learnable (ACM paper)

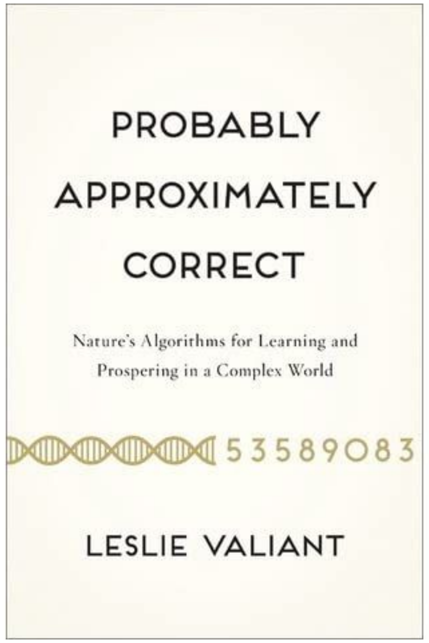


Leslie Gabriel Valiant

J. Coolidge Professor of Computer Science and Applied Mathematics at Harvard University

A.M. Turing Award 2010

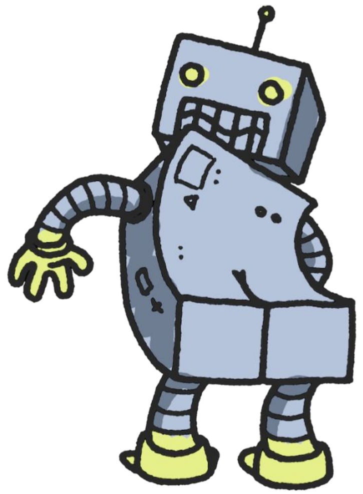
"For transformative contributions to the theory of computation, including the theory of probably approximately correct (PAC) learning"

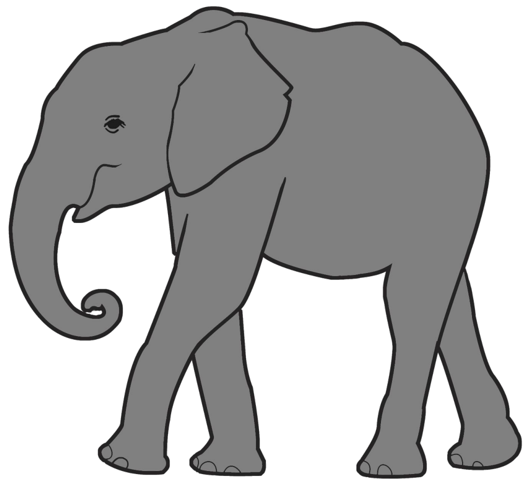

PROBABLY
APPROXIMATELY
CORRECT
Nature's Algorithms for Learning and Prospering in a Complex World
53589083
LESLIE VALIANT

A Theory of the Learnable

In this paper we have considered learning as the process of deducing a program for performing a task, from information that does not provide an explicit description of such a program.
We have given precise meaning to this notion of learning and have shown that in some restricted but nontrivial contexts it is computationally feasible.

Consider a world containing robots and elephants.





Suppose that one of the robots has discovered a recognition algorithm for elephants that can be meaningfully expressed in k-conjunctive normal form. Our Theorem A implies that this robot can communicate its algorithm to the rest of the robot population by simply exclaiming "elephant" whenever one appears. An important aspect of our approach, if cast in its greatest generality, is that we require the recognition algorithms of the teacher and learner to agree on an overwhelming fraction of only the natural inputs.