Foundations for Machine Learning

L. Y. Stefanus

TU Dresden, June-July 2018

Reference

- Shai Shalev-Shwartz and Shai Ben-David. UNDERSTANDING MACHINE LEARNING: From Theory to Algorithms. Cambridge University Press, 2014.

Definition of PAC (Probably Approximately Correct) Learnability

A hypothesis class H is PAC learnable if there exists a function $m_H: (0,1) \times (0,1) \rightarrow \mathbb{N}$ and a learning algorithm with the following property:

For every $\epsilon, \delta \in (0,1)$, for every distribution D over X, and

for every labeling function $f: X \rightarrow \{0,1\}$, if the realizable assumption holds with respect to $\mathrm{H}, \mathrm{D}, \mathrm{f},$ then when running the learning algorithm on $m \geq m_H(\epsilon, \delta)$ i.i.d. samples generated by D and labeled by f, the algorithm returns

a hypothesis $h \in H$ such that, with probability of at least $1 - \delta$ (over the choice of the samples), $L_{(D,f)}(h) \leq \epsilon$.

- The accuracy parameter ϵ determines how far the resulted classifier can be from the optimal one (this corresponds to the “approximately correct” part of PAC), and a confidence parameter δ indicating how likely the classifier is to meet that accuracy requirement (corresponds to the “probably” part of PAC).
- Under the data access model that we are investigating, these approximations are inevitable. Since the training set is randomly generated, there may always be a small chance that it will happen to be non-representative (for example, the training set might contain only one domain point, sampled over and over again). Furthermore, even when we are lucky enough to get a training sample that does faithfully represent D, because it is just a finite sample, there may always be some fine details of D that it fails to reflect. The accuracy parameter, ϵ, allows “forgiving” the learner's classifier for making minor errors.

Sample Complexity

- The function $m_H(\epsilon, \delta)$ determines the sample complexity of learning H: that is, how many samples are required to guarantee a probably approximately correct solution.
- Note that if H is PAC learnable, there are many functions $m_H(\epsilon, \delta)$ that satisfy the requirements given in the definition of PAC learnability. Therefore, to be precise, we will define the sample complexity of learning H to be the minimal function, in the sense that $m_H(\epsilon, \delta)$ gives the minimal integer that satisfies the requirements of PAC learning.
- Using the terminology of sample complexity, Theorem 1 can be expressed as follows:

Corollary 2

- Every finite hypothesis class is PAC learnable with sample complexity

$$m_H(\epsilon,\delta) \le \left\lceil \frac{\ln(|\mathrm{H}|/\delta)}{\epsilon} \right\rceil$$

Releasing the Realizability Assumption

- Recall that the realizability assumption requires that there exists $h^* \in H$ such that $\mathbb{P}_{x\backsim D}[h^*(x) = f(x)] = 1$. In many practical problems this assumption does not hold. Furthermore, it is maybe more realistic not to assume that the labels are fully determined by the features we measure on input elements (in the case of the papayas learning, it is plausible that two papayas of the same color and softness will have different taste). In the following, we relax the realizability assumption by replacing the “target labeling function” with a more flexible notion, a data-labels generating distribution.

- Formally, let D be a probability distribution over $X \times Y$, where, X is our domain set and Y is a set of labels (usually $Y = \{0,1\}$). That is, D is a joint distribution over domain points and labels.
- One can view such a distribution as being composed of two parts: a distribution D_x over unlabeled domain points (sometimes called the marginal distribution) and a conditional probability over labels for each domain point, $D((x,y)|x)$. In the papaya example, D_x determines the probability of encountering a papaya whose color and softness fall in some color-softness values domain, and the conditional probability is the probability that a papaya with color and softness represented by x is tasty. Indeed, such modeling allows for two papayas that share the same color and softness to belong to different taste categories.

The Revised True Error

- For a probability distribution, D, over X × Y, one can measure how likely h is to make an error when labeled points are randomly drawn according to D. We redefine the true error (or risk or loss) of a prediction rule h to be

$$L_D(h) \overset{\text{def}}{=} \mathbb{P}_{(x,y)\sim D}[h(x) \neq y] \overset{\text{def}}{=} D(\{(x,y): h(x) \neq y\})$$

- We would like to find a predictor, h, for which this error will be minimized. However, the learner does not know the data generating D. What the learner does have access to is the training data, S.

- The definition of the empirical risk remains the same as before, namely,

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

- Given S, a learner can compute $L_S(h)$ for any function $h: X \rightarrow Y$.
- Note that $L_S(h) = L_{D(\text{uniform over } S)}(h)$.
- We wish to find some hypothesis, $h: X \rightarrow Y$, that probably approximately minimizes the true risk, $L_D(h)$.

Definition of Agnostic PAC Learnability

A hypothesis class H is Agnostic PAC learnable if there exists a function $m_H : (0,1) \times (0,1) \rightarrow \mathbb{N}$ and a learning algorithm with the following property:

For every $\epsilon, \delta \in (0,1)$, for every distribution D over $\mathrm{X} \times \mathrm{Y}$, when running the learning algorithm on $m \geq m_H(\epsilon, \delta)$ i.i.d. samples generated by D, the algorithm returns a hypothesis $h \in H$ such that, with probability of at least $1 - \delta$ (over the choice of the *m* training samples),

$$L_D(h) \leq \min_{g \in H} L_D(g) + \epsilon.$$

Agnostic PAC learning generalizes PAC learning

- If the realizability assumption holds, agnostic PAC learning provides the same guarantee as PAC learning. In that sense, agnostic PAC learning generalizes PAC learning.
- When the realizability assumption does not hold, no learner can guarantee an arbitrarily small error. Nevertheless, under the definition of agnostic PAC learning, a learner can still declare success if its error is not much larger than the best error achievable by a predictor from the class H.
- This is in contrast to PAC learning, in which the learner is required to achieve a small error in absolute terms and not relative to the best error achievable by the hypothesis class.

Extension to a variety of learning tasks

- Multiclass Classification
  - Our classification does not have to be binary. Take, for example, the task of document classification according to topics.
- Regression
  - We wish to find some simple pattern in the data, i.e., a functional relationship between the X and Y components of the data. Here the target set Y is the the set of real numbers.

Generalized Loss Functions

- To accommodate a wide range of learning tasks we generalize our formalism of the measure of success as follows:
  - Given any set H (that plays the role of our hypotheses, or models) and some domain Z let ℓ be any function from H × Z to the set of nonnegative real numbers,

$$\ell : H \times Z \to \mathbb{R}_+$$

  - The risk function is defined to be the expected loss of a classifier, $h \in H$, with respect to a probability distribution D over Z, namely,

$$L_D(h) \overset{\text{def}}{=} \underset{z \sim D}{\mathbb{E}} [\ell(h, z)] .$$

- Similarly, we define the empirical risk to be the expected loss over a given sample $S = (z_1, \dots, z_m) \in Z^m$, namely,

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

- For example, the loss function used in the binary and multiclass classification tasks is the 0-1 loss:

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

The random variable z ranges over the set of pairs $X \times Y$.

Definition of Agnostic PAC Learnability for General Loss Functions

A hypothesis class H is Agnostic PAC learnable with respect to a set Z and a loss function $\ell: H \times Z \to \mathbb{R}_+$, if there exists a function $m_H: (0,1) \times (0,1) \to \mathbb{N}$ and a learning algorithm with the following property:

For every $\epsilon, \delta \in (0,1)$, for every distribution D over Z, when running the learning algorithm on $m \geq m_H(\epsilon, \delta)$ i.i.d. samples generated by D, the algorithm returns a hypothesis $h \in H$ such that, with probability of at least $1 - \delta$ (over the choice of the m training samples),

$$L_D(h) \leq \min_{g \in H} L_D(g) + \epsilon,$$

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

A Bit of History

- PAC learning was introduced by Leslie Valiant (1984).
  - Valiant, L. G. (1984), "A theory of the learnable," *Communications of the ACM* 27(11), 1134-1142.
- Valiant was named the winner of the 2010 Turing Award for the introduction of the PAC model.

Machine Learning

+

Big Data

⇓

Industrially Useful Predictor

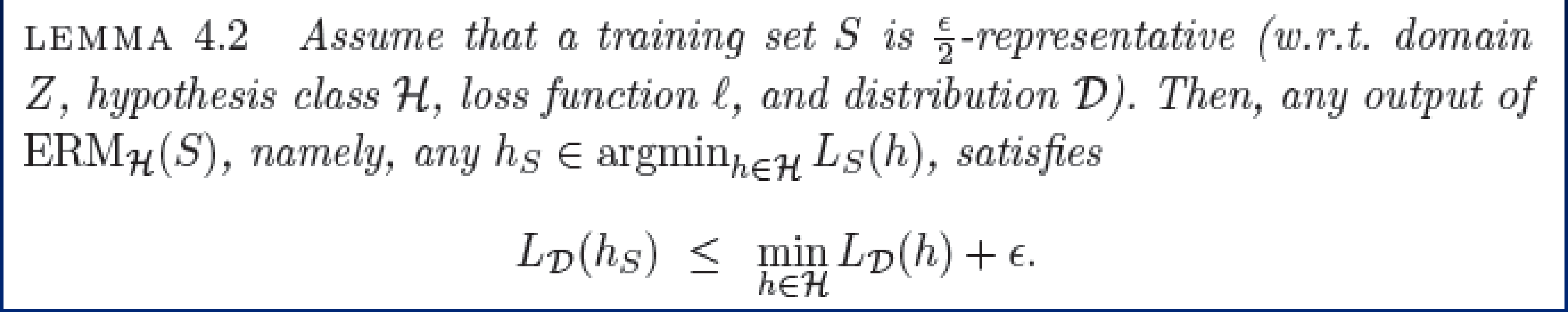
Learning via Uniform Convergence

- The idea behind uniform convergence is very simple.
- Recall that, given a hypothesis class, H, the ERM learning paradigm works as follows: Upon receiving a training sample, S, the learner evaluates the risk (or error) of each h in H on the given sample and outputs a member of H that minimizes this empirical risk.
- The hope is that an h that minimizes the empirical risk with respect to S is a risk minimizer w.r.t. the true data probability distribution as well. For that, it suffices to ensure that the empirical risks of all members of H are good approximations of their true risk. This condition is formalized as follows.

definition of ∈-representative sample

A training set S is called ∈-representative (w.r.t. domain Z, hypothesis class H, loss function ℓ, and distribution D) if

$$\forall h \in H, |L_S(h) - L_D(h)| \le \epsilon.$$



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*

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

Proof

For every $h \in H$,

$$L_D(h_S) \leq L_S(h_S) + \frac{\epsilon}{2} \leq L_S(h) + \frac{\epsilon}{2} \leq L_D(h) + \frac{\epsilon}{2} + \frac{\epsilon}{2}$$

$$= L_D(h) + \epsilon,$$

where the first and third inequalities are due to the assumption that S is $\frac{\epsilon}{2}$-representative and the second inequality holds since h_S is an ERM predictor.

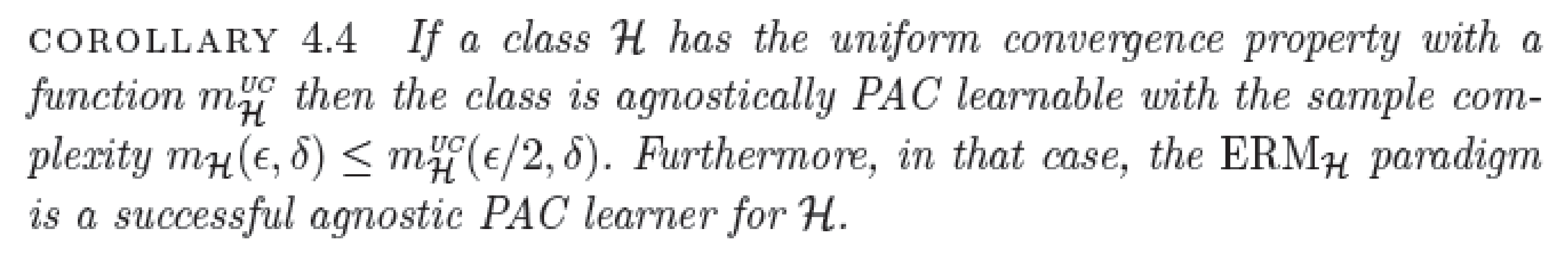
definition of Uniform Convergence

A hypothesis class H has the uniform convergence property (w.r.t. a domain Z and a loss function ℓ) if there exists a function $m_H^{\mathrm{UC}}: (0,1) \times (0,1) \rightarrow \mathbb{N}$ such that for every $\epsilon, \delta \in (0,1)$ and for every probability distribution D over Z, if S is a sample of $m \geq m_H^{\mathrm{UC}}(\epsilon, \delta)$ instances drawn i.i.d. according to D, then, with probability of at least $1 - \delta$,

S is ϵ-representative.

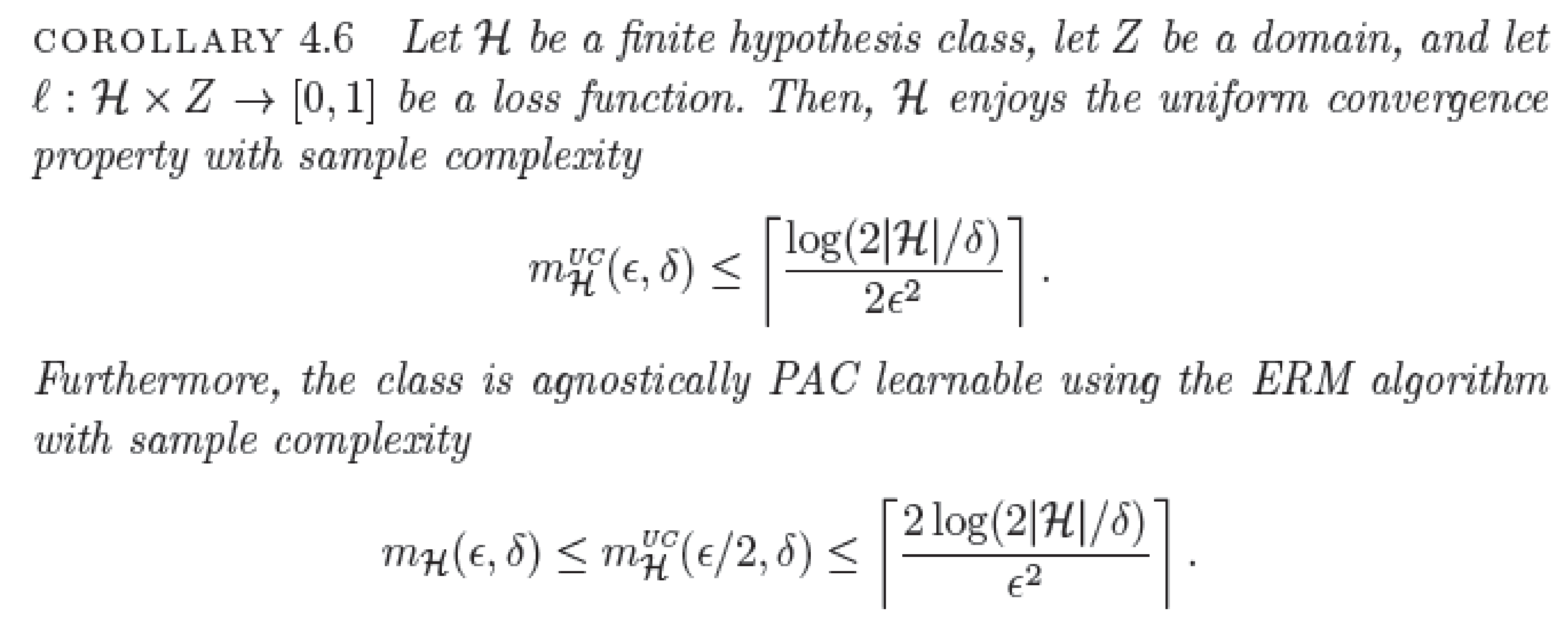
- Note: The term uniform here refers to having a fixed sample size that works for all members of H and over all possible probability distributions over the domain.

- The following corollary follows directly from Lemma 4.2 and the definition of uniform convergence.



COROLLARY 4.4 *If a class $\mathcal{H}$ has the uniform convergence property with a function $m_{\mathcal{H}}^{UC}$ then the class is agnostically PAC learnable with the sample complexity $m_{\mathcal{H}}(\epsilon,\delta) \leq m_{\mathcal{H}}^{UC}(\epsilon/2,\delta)$. Furthermore, in that case, the* $\mathrm{ERM}_{\mathcal{H}}$ *paradigm is a successful agnostic PAC learner for $\mathcal{H}$.*

every finite hypothesis class is agnostic PAC learnable



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*

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

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

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