1 Glivenko-Cantelli classes

Let (Ω, P) be a probability space and $\mathcal{F} \subset \{0,1\}^{\Omega}$ be a family of binary functions on Ω. We say that $\mathcal{F}$ is a Universal Uniform Glivenko-Cantelli (UUGC) class if

$$\lim_{n\to\infty} \sup_P \mathbb{E}_{\mathbb{P}} \left[\sup_{f\in\mathcal{F}} |Pf - P_n f| \right] = 0,$$

where $\sup_P$ is the supremum over all the distributions on Ω and

$$P_n f = \frac{1}{n} \sum_{i=1}^{n} f(X_i), \qquad Pf = \mathbb{E} f(X_1)$$

for $X_1, \ldots, X_n$ distributed iid $\sim P$.

[If you're not familiar with the concept of *measurability* from real analysis, you can ignore this comment. If you are familiar with this concept, note that we will ignore all issues of measurability (which are non-trivial; for example, it is not enough to assume that each $f \in \mathcal{F}$ is measurable). Assuming $\mathcal{F}$ to be a countable collection takes care of all the problems, and this assumption may be weakened significantly. See [1] for the gory details.]

2 PAC learnability

Again we start with a probability space $(\mathcal{X}, P)$, except now $\mathcal{X}$ will be called an *instance space*. We will also fix a family of binary functions $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$, and will refer to it as a *concept class*.

Consider a two-player game between a **teacher** and a **learner**. After they agree on $\mathcal{X}$ and $\mathcal{H}$ (but not P, which only the teacher knows!), they play the following game. The teacher picks some $g \in \mathcal{H}$ (without revealing it to the learner) and challenges the learner to produce an ε-good hypothesis $h \in \mathcal{H}$. A hypothesis h is ε-good (with respect to the teacher's *target concept* g) if

$$\mathbb{P}[h(X) \neq g(X)] < \varepsilon,$$

where the probability is over $X \sim P$.

Of course, without further information, the learner has no hope of succeeding. However, he is allowed to request from the teacher a *labeled sample* of size n:

$$S = (X_i, Y_i), \qquad i = 1, \ldots, n,$$

where $X_1, \ldots, X_n$ iid $\sim P$ and $Y_i = g(X_i)$ for $i \in [n]$. Based on the sample S, the learner will construct a hypothesis $h_S \in \mathcal{H}$, which may or may not be ε-good. Note that the learner might

be unlucky and receive a very unrepresentative sample S, so any guarantees he makes can only be probabilistic. We say that the learner is (ε, δ)-successful if h_S is ε-good with probability at least $1 - \delta$. Our learner is said to be *Probably Approximately Correct* (PAC) if for all ε, δ there is an m_0 such that for all $g \in \mathcal{H}$, all distributions P and samples of size $n \geq m_0$, the learner is (ε, δ)-successful.

Formally, this condition may be stated as follows:

$$\forall \varepsilon, \delta > 0 : \ \exists m \in \mathbb{N} : \ \forall n \geq m, \forall P, \forall S \sim P^n, \forall g \in \mathcal{H} : \ \mathbb{P}[\mathrm{err}_P(h_S) > \varepsilon] < \delta, \tag{1}$$

where

$$\mathrm{err}_P(h) = P\{x \in \mathcal{X} : h(x) \neq g(x)\} \tag{2}$$

is the probability (under distribution P) that h disagrees with g on a random $X \sim P$ and h_S is the hypothesis constructed by the learner based on the sample S.

In computational learning theory, one of the main problems studied is how to efficiently construct the hypothesis h_S from the sample. See chapter 2 of [3] or chapter 3 of [4].

We will not be discussing the computational aspects of learning in this course. We will say that $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is PAC-learnable (or just PAC) if there exists a learner (i.e., a mapping $\mathcal{L} : S \mapsto h_S$) that makes (1) hold. Note: in the vast and rich field that modern learning theory has become, we are studying a very small corner, known as *passive supervised learning*. It's passive because the learner has no control over which sample points he'll receive (as opposed to *active learning*, where he does have some control). And supervised means that the learner gets to see labels, which is not always a realistic assumption. Finally, we will only deal with binary functions — mostly because it provides for a very clean and elegant theory.

3 Agnostic PAC

The definition (1) is the "plain vanilla[1]" PAC that's given in [2]. It has since been generalized in the following way. Instead of having a teacher who labels the sample points X_i with $Y_i = g(X_i)$, we will take P to be a distribution over $\mathcal{X} \times \{0,1\}$. As in regular PAC, the learner observes the sample

$$S = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$$

with $X_i \in \mathcal{X}$ and $Y_i \in \{0,1\}$ — except that this time, there is no teacher and no "true" target concept $g \in \mathcal{H}$. Instead, the pairs (X_i, Y_i) are drawn iid $\sim P$. In other words, the example-label is a pair of random variables with some joint distribution. Of course, a particular instance of this setting is when there is some $g \in \mathcal{H}$ such that $Y_i = g(X_i)$; in this case

$$P(Y_i = g(X_i) \,|\, X_i) = 1.$$

[1] known in the literature as "consistent" or "realizable"

For a given hypothesis $h : \mathcal{X} \to \{0,1\}$, we will define its *empirical/sample error* $\mathrm{err}_S(h)$ and *generalization error* $\mathrm{err}_P(h)$. Empirical error is defined exactly as in the regular PAC case:

$$\mathrm{err}_S(h) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{h(X_i)\neq Y_i\}}.$$

The definition of generalization (2) error requires a slight modification:

$$\mathrm{err}_P(h) = P\{(x,y) \in \mathcal{X} \times \{0,1\} : h(x) \neq y\}$$

— since the distribution is now over both the instances X and the labels Y.

What is the learner's goal in agnostic PAC? We cannot require him to achieve a small generalization error (even with high probability), since there might not by **any** good hypothesis. Think of the case where $P(Y = 0 \,|\, X = x) = P(Y = 1 \,|\, X = x) = 1/2$ for all $x \in \mathcal{X}$. In this case, every hypothesis h will have $\mathrm{err}_P(h) = 1/2$. Hence, all we demand from the learner's hypothesis h_S is that it not perform much worse than the best hypothesis in the class:

$$\forall \varepsilon, \delta > 0 : \ \exists m \in \mathbb{N} : \ \forall n \geq m, \forall P, \forall S \sim P^n : \ \mathbb{P}\left[\mathrm{err}_P(h_S) > \mathrm{opt}_P(\mathcal{H}) + \varepsilon\right] < \delta, \tag{3}$$

where

$$\mathrm{opt}_P(\mathcal{H}) = \inf_{g \in \mathcal{H}} \mathrm{err}_P(g).$$

We will say that $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ is agnostic PAC-learnable if there exists a learner (i.e., a mapping $\mathcal{L} : S \mapsto h_S$) that makes (3) hold.

Exercise: Convince yourself that the plain-vanilla PAC model defined in Sec. 2 is a special case of agnostic PAC.

4 UUGC implies PAC

Let $\mathcal{H}$ be a concept class over $(\mathcal{X}, P)$. Define $\mathcal{F} = \mathcal{F}_{\mathcal{H}}$ to be the following class of functions over $\mathcal{X}' = \mathcal{X} \times \{0,1\}$:

$$\left\{ f : (x,y) \mapsto \mathbb{1}_{\{h(x)\neq y\}} : h \in \mathcal{H} \right\}. \tag{4}$$

That is, functions in $\mathcal{F}$ are indexed by members of $\mathcal{H}$, and for every $h \in \mathcal{H}$ there is an $f \in \mathcal{F}$ mapping the pair $(x,y) \in \mathcal{X}'$ to $\{0,1\}$.

Theorem 4.1 *If $\mathcal{F}$ is UUGC then $\mathcal{H}$ is agnostic PAC-learnable.*

Proof: Our learner $\mathcal{L} : S \mapsto h_S$ will be the *empirical risk minimizer* — that is, $\mathcal{L}$ will pick the $h \in \mathcal{H}$ that minimizes the sample error $\mathrm{err}_S(h)$, breaking ties arbitrarily.

Convince yourself that for every sample $Z_1, \ldots, Z_n$, where $Z_i = (X_i, Y_i)$ iid$\sim P$, we have

$$\sup_{f \in \mathcal{F}} |Pf - P_n f| = \sup_{h \in \mathcal{H}} |\mathrm{err}_S(h) - \mathrm{err}_P(h)|$$

— in fact, these are two descriptions of the same random variable.

Let $\hat{h}$ be the minimizer[2] of err_S and h^* be the minimizer[3] of err_P:

$$\mathrm{err}_S(\hat{h}) = \inf_{h \in \mathcal{H}} \mathrm{err}_S(h); \qquad \mathrm{err}_P(h^*) = \inf_{h \in \mathcal{H}} \mathrm{err}_P(h). \tag{5}$$

It's immediate from (5) that

$$\begin{aligned} \mathrm{err}_P(\hat{h}) - \mathrm{err}_P(h^*) &= \mathrm{err}_P(\hat{h}) - \mathrm{err}_S(\hat{h}) + \mathrm{err}_S(\hat{h}) - \mathrm{err}_P(h^*) \\ &\leq \mathrm{err}_P(\hat{h}) - \mathrm{err}_S(\hat{h}) + \mathrm{err}_S(h^*) - \mathrm{err}_P(h^*) \end{aligned}$$

and hence

$$\begin{aligned} \left|\mathrm{err}_P(\hat{h}) - \mathrm{err}_P(h^*)\right| &\leq \left|\mathrm{err}_P(\hat{h}) - \mathrm{err}_S(\hat{h})\right| + |\mathrm{err}_S(h^*) - \mathrm{err}_P(h^*)| \\ &\leq 2 \sup_{f \in \mathcal{F}} |Pf - P_n f| \,. \end{aligned}$$

Since

$$\mathbb{P}\left[\mathrm{err}_P(h_S) > \mathrm{opt}_P(\mathcal{H}) + \varepsilon\right] \leq \mathbb{P}\left[|Pf - P_n f| > \varepsilon/2\right],$$

we have that $\mathcal{F}$ is UUGC $\implies$ $\mathcal{H}$ is agnostic PAC, and therefore also plain-vanilla PAC. ■

5 The equivalence of UUGC and PAC

We have defined a property of concept classes called the VC-dimension, and shown that

(i) if $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ has finite VC-dimension then $\mathcal{F}$ is UUGC

In the upcoming lectures, we will show that

(ii) if $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ has infinite VC-dimension then $\mathcal{H}$ is not PAC-learnable

Finally, it will be very easy to see that when $\mathcal{F}$ is constructed from $\mathcal{H}$ as in (4), they have the same VC-dimension (verify this!).

Conclusion: for binary function classes, PAC (in its two senses) and UUGC are equivalent notions. Actually, from the proof of Thm. 4.1 we gain even more information: the learner who minimizes the empirical error (also known as an Empirical Risk Minimizer [ERM]) is a "universal learner" for binary function classes: $\mathcal{H}$ is PAC iff it is UUGC iff it is learnable by an ERM.

[2] if not unique, pick one arbitrarily

[3] why is there no loss of generality in assuming that (at least) one exists?

References

[1] Richard M. Dudley. A course on empirical processes. In *École d'été de probabilités de Saint-Flour, XII—1982*, volume 1097 of *Lecture Notes in Math.*, pages 1–142. Springer, Berlin, 1984.

[2] Micheal Kearns and Umesh Vazirani. *An Introduction to Computational Learning Theory.* The MIT Press, 1997.

[3] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations Of Machine Learning.* The MIT Press, 2012.

[4] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.