Probably Approximately Correct Learning - An Introduction in the Finite Case

Kevin de Haan

Edmonton, Canada
Email: kdehaan@ualberta.ca

***Abstract*—Probably Approximately Correct (PAC) learning is a tool that was first introduced in 1984 by Valiant in order to bridge the gap between computability theory and machine learning [1]. In doing so, Valiant introduced the concept of learnability to a wider group of computer scientists interested in algorithmic efficiency and leading to the combined field of computational learning theory [2].**

***Index Terms*—PAC; Learning Theory; Computability Theory; Computational Learning Theory**

I. Introduction

The Probably Approximately Correct (hereafter PAC) learning framework, first introduced by Valiant [1], is a method for determining the learnability of a problem for a machine classifier. In other words, using the PAC learning framework one can estimate whether a learning classifier will be able to output an approximately correct prediction, and under what conditions the classifier is able to operate. If a concept is known to be PAC-learnable, it means that the learning algorithm must be able to operate in polynomial time [2], and additionally provides an upper bound on the volume of training samples provided to the classifier in order to produce an acceptable prediction [2]. It can also be useful to prove that a target is not PAC-learnable, for example in the case of a cryptographic function [2].

II. Deriving the Key Points of PAC Learning

A. Terminology

PAC and indeed learning theory as a whole has some terminology that varies from traditional machine learning notation. All non-trivial symbols used in this paper can be found in the following list:

- X: the domain space
- Y: the label set space
- f: the mapping function between domain set space to label set space, $f : X \to Y$. Unknowable in its entirety.
- $\mathcal{D}$: the data distribution over X. Unknowable in its entirety.
- S: a sample taken from $\mathcal{D}$
- h: a hypothesis, also referred to as a $model$. A prediction of the real mapping function f
- $\mathcal{H}$: a finite hypothesis space
- L: a $learner$, the algorithm or device that produces a hypothesis h

B. Background: Empirical Risk Minimization

To properly understand the theory behind PAC, it is valuable to have an understanding of Empirical Risk Minimization (hereafter ERM). The fundamental theory behind ERM is that given some $h \in \mathcal{H}$ produced from the training set S, there will be some error between h_S and the real f, demonstrated in equation 1. The true error is equal to the probability of sampling x from $\mathcal{D}$ such that prediction of h is different from that of f [3].

$$L_{\mathcal{D},f}(h) = \mathcal{D}(\{x : h(x) \neq f(x)\}) \tag{1}$$

However, because $\mathcal{D}$ and f are unknown, the full error calculation cannot simply be evaluated by the learner L [3]. The next best thing is to determine the *training error*, the error encountered by the classifier in the process of training (equation 2) [3]. The training error is equal to the real number of times that the prediction from h has been different from the real value in y, divided by the number of samples m. This is also referred to as the *empirical error* or *empirical risk* interchangeably [3].

$$L_S(h) = \frac{|\{i \in [m] : h(x_i) \neq y_i\}|}{m}, [m] = \{1, ..., m\} \tag{2}$$

In contrast to the real error (or *actual risk*) the value of empirical risk is available to the learner and is therefore an obvious choice for minimization - hence, the goal of *empirical risk minimization*.

III. Conclusion and Critical Analysis

References

[1] L. G. Valiant, “A theory of the learnable,” *Commun. ACM*, vol. 27, no. 11, p. 1134–1142, Nov. 1984. [Online]. Available: https://doi.org/10.1145/1968.1972

[2] D. Haussler, “Part 1: Overview of the probably approximately correct (pac) learning framework,” 1995.

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