Review of
Probably Approximately Correct
by Leslie Valiant
Basic Books, 2013
195 pages, Hardcover

Review by
Joshua Brulé jtcbrule@gmail.com
University of Maryland, College Park

1 Introduction

Probably approximately correct (PAC) learning is a mathematical framework for analyzing the limits of what is 'learnable', not unlike how computational complexity theory acts as a mathematical framework for analyzing the ideas of 'computable' or 'tractable'. PAC learning can also be compared to the (perhaps better known) Bayesian framework which uses a prior distribution on hypotheses and updates via Bayes rule to represent learning.

Remarkably, PAC learning allows learning *without* assuming a prior probability distribution over hypotheses (one of the common criticisms of Bayesian inference). Instead, all that is required is the existence of distribution from which data is drawn, and that the sample data that the learner sees is drawn from the same distribution. Learning a hypothesis that is correct *all* the time on *all* the data is, of course, impossible. But given a polynomial number of examples to learn from, a PAC learner can find a hypothesis that is usually (probably) correct over most of (approximately) all the future data.

For inventing PAC (and other significant contributions to theory of computation), Leslie Valiant was awarded the 2010 Turing Award.

2 Summary

Probably Approximately Correct (the book) is Valiant's exposition on PAC learning written for a general (i.e. non-computer scientist) audience. However, *Probably Approximately Correct* stands out from most other 'popular science' books by explaining not only the theory itself, but the remarkable advances made in the rest of computer science theory that were required to make PAC possible. Valiant also discusses some of more interesting applications of PAC, specifically, how evolution can be viewed as a PAC learning algorithm and how human level AI is most likely not outside of our reach, seeing as we have no reason to believe that human beings learn in some computationally special way.

Chapters 1 and 2 discuss some of the inherent difficulties in machine learning. Valiant coins the term 'ecorithm' to describe an algorithm designed to solve a problem that we lack a quantitative theory for (the 'theoryless'). This acts as a nice distinction from traditional algorithms (e.g. running an n-body simulation) which can designed around a more traditional scientific theory such as Newtonian mechanics (which Valiant refers to as the 'theoryful').

Chapter 3 is a very accessible discussion computational complexity theory, including Turing machines, computability and classic complexity classes such as P, BPP, BQP and NP. Valiant also

provides an excellent example of a learning algorithm in his description of the classic perceptron algorithm. Unlike most popular science books (whose publishers seem to be afraid of introducing too many numbers or equations), Valiant uses just enough mathematics to accurately describe the algorithm, while still keeping the description accessible to anyone with a high school background.

Chapter 4 is a brief argument regarding the importance of studying how information is processed in biological systems rather than studying just the substrate of life, similar to how computer science focuses on studying algorithms independently of hardware.

Chapter 5 is a description of the actual PAC learning framework in which Valiant again does a remarkable job of using just enough math to explain the basic ideas while remaining accessible to most readers. Special mention should go to using cryptography as an intuitive example of how not all of P is learnable (or rather, as an example of why we should not *believe* all of P is learnable). For if we could learn the details behind any cryptosystem constructed with polynomial resources in polynomial time, there would be no such thing as a secure cryptosystem.

Chapter 6 is an argument that evolution might be better viewed as an instantiation of a PAC learning process and that perhaps evolution is not as widely accepted as other physical theories (e.g. gravity) because it lacks more quantitative predictions. This foray into analyzing evolution from a PAC standpoint is intended to act as a starting place for more research on evolution with the additional goal of gaining wider acceptance of the theory. (I don't know how convincing this will be to an intelligent design advocate, but it's a worthy effort.)

Chapter 7 is a discussion of classic first order logic's brittleness in building intelligent systems. The first half of the chapter is quite valuable in explaining to a general audience why AI has appeared to fail in delivering on it's early promises and sets up for the next chapter in discussing why we should still have hope. Valiant also argues that an adequate robust logic can be derived from PAC learning. The discussion of robust logic feels less clear (compared to the earlier explanation of PAC learning), but should still be accessible to a general audience.

Finally, Chapter 8, 9 and 10 discuss viewing humans and machines as 'ecorithms'. The overall view is quite optimistic: that we should have strong hopes for advancement in AI since we have no reason to believe that humans have access to a form of inference more powerful than PAC learning. Valiant quite convincing argues that the main difficulties seem to come from the fact that AI is 'competing' against billions of years of evolution. And considering that computer science as a field of study has been around for less then a century, the field has made remarkable progress in such a short time.

3 Opinion

For the mathematically literate, *Probably Approximately Correct* is likely not the best introduction to PAC learning, especially considering that a large part of the book is spent discussing concepts that virtually all computer scientists will be strongly familiar with, only in less detail than a more technical introduction.

However, the book really shines as an introduction to computer science theory to the general public, providing a compact and accessible description of basic, *important* results that are sadly not widely known outside the field. This is a book that should be on every computer scientist's shelf so that when someone asks, "Why is computer science theory important?" the three word response can be, "Read this book".