Probabilistic Learning Theory, with Emphasis on Sample Complexity

Martin Anthony

Department of Statistical and Mathematical Sciences, London School of Economics and Political Science, Houghton St., London WC2A 2AE, United Kingdom

There are a number of mathematical models of machine and human learning. Here, we survey the 'probably approximately correct' model and some of its modifications and generalisations. Our emphasis is on sample complexity, but we also briefly discuss issues of computational complexity.

PART I: THE BASIC PAC THEORY

In this part of the paper we present the key results of pac learning theory, with emphasis on the sample complexity results. We also briefly discuss questions of computational complexity. This part of the paper is an adaptation of parts of the book (Anthony and Biggs (1992)) by Norman Biggs and the present author; we refer the interested reader there for a more wide-ranging discussion of the basic pac learning model.

1. LEARNING AND TRAINING: A FRAMEWORK

In a general framework for our discussion of learning, we have a 'real world' W containing a set of objects which we shall refer to as examples. We also have a 'pre-processor' P, which takes an example and converts it into a coded message, such as a string of bits or a real vector. This coded version of the example is then presented to M, the 'learner', whose purpose is to recognise certain examples. The output of M is a single bit, either 1 (if the example is recognised as belonging to a certain set), or 0 (if not). The learner

---

This paper was written while the author was visiting Sonderforschungsbereich 343 "Diskrete Strukturen in der Mathematik", University of Bielefeld, Germany.

can be in one of many states. For example, M might be in a state in which it will recognise examples of the letter **A**, coded as a string of bits in which the 1's correspond to the positions of black pixels on a grid; other states of M might enable it to recognise examples of other letters. The learning process we consider involves making changes to the state of the learner on the basis of examples presented to it, so that it achieves some desired classification. For instance, we might be able to change the state in such a way that the more positive and negative examples of the letter **A** that are presented, the better will M recognise future examples of that letter.

For our purposes, a 'concept' can be described by a set of examples. Let Σ be a set, which we shall call the *alphabet* for describing examples. In this article, Σ will be either the *boolean* alphabet $\{0, 1\}$, or the *real* alphabet $\mathbb{R}$. We denote the set of n-tuples of elements of Σ by Σ^n and the set of all non-empty finite strings of elements of Σ by Σ^*. Let X be a subset of Σ^*. We define a *concept*, over the alphabet Σ, to be a function $c : X \to \{0, 1\}$. In some cases X will be the whole of Σ^*; while in other cases X will be taken as Σ^n for one specific value of n, which will be clear from the context.

The set X will be referred to as the *example space*, and its members as *examples*. An example $y \in X$ for which $c(y) = 1$ is known as a *positive example*, and an example for which $c(y) = 0$ is known as a *negative example*. The union of the set of positive examples and the set of negative examples is the domain of c. So, provided that the domain is known, c determines, and is determined by, its set of positive examples. Sometimes it is helpful to think of a concept as a set in that way.

There are two sets of concepts inherent in our framework for learning. First, there is the set of concepts derived from the real world which it is proposed to recognise. This set might contain concepts like 'the letter **A**', 'the letter **B**', 'the letter **C**', and so on, each of which can be coded to determine a set of positive and negative examples. When a set of concepts is determined in this way, we shall use the term *concept space* for it. Secondly, there is the set of concepts which the learner M is capable of recognising. We shall suppose that M can assume various states, and that in a given state it will classify some inputs as positive examples (output 1), and the rest as negative examples (output 0). Thus a state of M determines a concept, which we may think of as a hypothesis. For this reason, the set of all concepts which the learner determines will be referred to as its *hypothesis space*.

Generally speaking, we are given two sets of concepts, C (the concept space) and H (the hypothesis space), and the problem is to find, for each $c \in C$, some $h \in H$ which is a good approximation to c. In realistic situations hypotheses are formed on the basis of certain information which does not amount to an explicit definition of c. In the framework we describe, we shall assume that this information is provided by a sequence of positive and negative examples of c. In practice there are constraints upon our resources, and we have to be content with a hypothesis h which 'probably' represents c 'approximately', in some sense to be defined.