17

Information Processing Letters 24 (1987) 377–380
North-Holland

6 April 1987

OCCAM'S RAZOR

Anselm BLUMER *

Department of Mathematics and Computer Science, University of Denver, Denver, CO 80208, U.S.A.

Andrzej EHRENFEUCHT **

Department of Computer Science, University of Colorado, Boulder, CO 80302, U.S.A.

David HAUSSLER *

Department of Mathematics and Computer Science, University of Denver, Denver, CO 80208, U.S.A.

Manfred K. WARMUTH ***

Department of Computer and Information Sciences, University of California, Santa Cruz, CA 95064, U.S.A.

Communicated by M.A. Harrison
Received 21 February 1986
Revised 18 July 1986

We show that a polynomial learning algorithm, as defined by Valiant (1984), is obtained whenever there exists a polynomial-time method of producing, for any sequence of observations, a nearly minimum hypothesis that is consistent with these observations.

Keywords: Machine learning, induction, inductive inference, Occam's Razor, methodology of science

"Entities should not be multiplied unnecessarily"
William of Occam, c. 1320

1. Introduction

Although William of Occam first wielded his famous razor against the superfluous elaborations of his Scholastic predecessors, his principle of parsimony has since been incorporated into the methodology of experimental science in the following form: given two explanations of the data, all other things being equal, the simpler explanation is preferable. This principle is very much alive today in the emerging science of machine learning, whose expressed goal is often to discover the simplest hypothesis that is consistent with the sample data [1]. As laudable as this goal may seem, whether in the area of machine learning or in science as a whole, one can still ask why the simplest hypothesis for a given sequence of observations should perform well on further observations taken from the same source. After all, the

* These authors were financially supported by the National Science Foundation under Grant IST-8317918.
** This author was financially supported by the National Science Foundation under Grant MCS-8305245.
*** Part of this work was done while this author was visiting the University of Denver and was supported by Research funds of the University of Denver. M.K. Warmuth would also like to express his thanks for Faculty Research funds granted by the University of California, Santa Cruz.

real value of a scientific explanation lies not in its ability to explain events past, but in predicting events that have yet to occur. We show that, under very general assumptions, Occam's Razor produces hypotheses that with high probability will be predictive of future observations. As a consequence, when hypotheses of minimum or near minimum complexity can be produced efficiently (i.e., in polynomial time) from sample data, this process leads to a polynomial learning algorithm, as defined by Valiant [5]. Our numerical results improve on related (but more general) results given in [3] and are derived using a simpler argument.

2. The Razor applied

Following [5], we consider the problem of learning a class H of functions that map from a fixed domain X into a fixed finite range. H will be called the *hypothesis class*. As an example, X may be the set of all finite strings of 0's and 1's and H the class of Boolean functions [1] or the class of $\{0, 1\}$-valued functions representing regular languages over X (see, e.g., [1]). Given a function f in H, an *observation* of f is a point $x \in X$, along with the value f(x). A sequence of m observations constitutes a *sample* (of f) of *size* m. The problem of learning is the problem of recovering f, or at least a function that approximates f, from a sample of f. We define a *learning algorithm* for H as an algorithm that takes as input a sample of an unknown function $f \in H$ and produces as output a hypothesis consistent with this sample that is itself a function in H. [2]

We assume that observations of f are made independently according to some fixed probability distribution P on X. Thus, a sample of size m is chosen according to the product probability distribution P^m on X^m. The *error* of a hypothesis produced from such a sample is the probability that it disagrees with f on a single observation (chosen according to the distribution P). A successful learning algorithm is one that with high probability (with respect to P^m) finds a hypothesis whose error is small. For an arbitrary learning algorithm and for a finite H, the following lemma relates the number of hypotheses and the sample size to the error.

2.1. Lemma. *Given any function* f *in a hypothesis class of* r *hypotheses, the probability that any hypothesis with error larger than* ε *is consistent with a sample of* f *of size* m *is less than* $(1-\varepsilon)^m r$.

Proof. Let E_i be the event that hypothesis h_i agrees with an observation of f, and E_i^m be the event that h_i agrees with all m observations of f. The probability we are trying to bound is

$$P^m\left(\bigcup E_i^m\right),$$

where the union is over all i such that h_i has error larger than ε. For such an i, by independence and by the definition of error,

$$P^m(E_i^m) < (1-\varepsilon)^m.$$

The subadditivity of probability measures now gives the desired results. □

The lemma can be restated: the probability that all consistent hypotheses have error at most ε is larger than $1-(1-\varepsilon)^m r$. Furthermore, the bounds on the m are independent of the function f to be learned and independent of the probability distribution that governs the samples. As an immediate consequence of the above, we get an upper bound on the sample size m needed to assure that the hypothesis produced by any learning algorithm has error at most ε with probability larger than $1-\delta$. It suffices to have m such that

$$(1-\varepsilon)^m r \leqslant \delta,$$

which implies that

$$m \geqslant \frac{1}{-\ln(1-\varepsilon)}\left(\ln(r) + \ln\left(\frac{1}{\delta}\right)\right),$$

[1] Here we may assume that a Boolean function of v variables has some standard value representing 'undefined' on any input string which does not consist of exactly v bits.

[2] In this paper we ignore the issue of errors in the sample, and the possible computational advantages of allowing the algorithm to produce hypotheses that are either not in H or are only consistent with some observations in the sample.

and this certainly holds if

$$m > \frac{1}{\varepsilon}\left(\ln(r) + \ln\left(\frac{1}{\delta}\right)\right).$$

For a countably infinite hypothesis class H, we follow [3] and let the *complexity* of a hypothesis $h \in H$ be defined as the number of bits needed to represent h in some fixed encoding of the hypothesis in H. The complexity of a hypothesis will be denoted with the parameter n. In this setting, assuming that the domain X is countable as well, Valiant's notion of (polynomial) learnability [5] can be stated as follows.

H is *polynomially learnable* (with respect to a fixed encoding) if there exists a learning algorithm for H and a minimal sample size $m(\varepsilon, \delta, n)$, polynomial in $1/\varepsilon$, $1/\delta$, and n, such that:

(a) for all $f \in H$ of complexity at most n and all distributions P on X, given $m(\varepsilon, \delta, n)$ independent observations of f, the algorithm produces a hypothesis with error at most ε with probability at least $1 - \delta$, and

(b) the algorithm produces its hypothesis in time polynomial in the length of the given sample. [3]

A polynomial learning algorithm for an infinite hypothesis class cannot in general afford to choose a hypothesis arbitrarily. Occam's Razor would suggest that a learning algorithm should choose its hypothesis among those that are consistent with the sample and have minimum complexity. However, this is not always practical. For example, finding a minimum length DNF expression consistent with a sample of a Boolean function and finding a minimum size deterministic finite automaton consistent with positive and negative examples of a regular language are both NP-hard problems under standard encodings ([2, problems LO9 and AL8]; see also [4]). To obtain polynomial algorithms, we will weaken this criterion of minimality as follows.

[3] We must assume some standard encoding of the observations as well. In practice it should be required that both hypothesis and observation encodings have the property that a hypothesis can be checked for consistency against any observation in polynomial time.

2.2. Definition. An *Occam-algorithm* for H with constant parameters $c \geqslant 1$ and $0 \leqslant \alpha < 1$ is a learning algorithm that:

(i) produces a hypothesis of complexity at most $n^c m^\alpha$ when given a sample of size m of any function in H of complexity at most n, and

(ii) runs in time polynomial in the length of the sample.

We now show that the existence of an Occam-algorithm for H implies polynomial learnability.

2.3. Theorem. *Given independent observations of any function in* H *of complexity at most* n, *an Occam-algorithm with parameters* $c \geqslant 1$ *and* $0 \leqslant \alpha < 1$ *produces a hypothesis of error at most* ε *with probability at least* $1 - \delta$ *using sample size polynomial in* n, $1/\varepsilon$, *and* $1/\delta$, *independent of the function and of the probability distribution. The sample size required is*

$$O\left(\ln(1/\delta)/\varepsilon + (n^c/\varepsilon)^{1/(1-\alpha)}\right).$$

Proof. We will show that the sample size

$$m \geqslant \max\left\{\frac{2\ln(1/\delta)}{-\ln(1-\varepsilon)}, \left(\frac{2n^c \ln(2)}{-\ln(1-\varepsilon)}\right)^{1/(1-\alpha)}\right\}$$

is sufficient. The result then follows from the definition of Occam-algorithm. The second lower bound on m implies that

$$m^{1-\alpha} \geqslant \frac{2n^c \ln(2)}{-\ln(1-\varepsilon)},$$

or

$$n^c m^\alpha \ln(2) \leqslant -\tfrac{1}{2} m \ln(1-\varepsilon).$$

Hence, since the hypotheses under consideration are given by binary strings of length at most $n^c m^\alpha$, the number of hypotheses, r, is at most

$$2^{n^c m^\alpha} \leqslant (1-\varepsilon)^{-m/2}.$$

By Lemma 2.1 and the first lower bound on m, the probability of producing a hypothesis with error larger than ε is less than

$$(1-\varepsilon)^m r \leqslant (1-\varepsilon)^{m/2} \leqslant \delta. \qquad \square$$

An important special case of this theorem occurs when $\alpha = 0$, i.e., the Occam-algorithm gives consistent hypotheses of complexity at most n^c, independent of the sample size (where n is the complexity of the function to be learned). In this case, the sample size needed is

$$O\left(\frac{1}{\varepsilon}(n^c + \ln(1/\delta))\right).$$

The n^c term represents the maximum number of bits required to specify a hypothesis in the range of the learning algorithm when its domain is restricted to functions represented with at most n bits.

Finally, we note that results similar to those presented here can also be derived from the results given by Pearl [3], but the sample size required is significantly larger: at the very least, the ε's in the above theorem must be replaced by their squares.

3. Open problems

While it is known that finding a minimum hypothesis consistent with a sample is NP-hard for many common classes of functions, it appears that little is known about the existence of Occam-algorithms for these classes of functions. Can it be shown that if $P \neq NP$, then there are no Occam-algorithms for Boolean functions under standard encodings, nor are there any for deterministic finite automata? Results of this type would be consistent with the hypothesis that these classes are not polynomially learnable under standard encodings. Stronger evidence for this hypothesis would be obtained if it can also be shown that the converse of the above learnability result holds, i.e., if polynomial learnability can actually be reduced to the problem of finding Occam-algorithms. In this case, these classes would not be polynomially learnable under standard encodings unless $P = NP$.

Acknowledgment

We would like to thank Les Valiant for stimulating discussions of these ideas that led to the definition of Occam-algorithms we have given here. We would also like to thank Jan Mycielski, Bill Reinhart and Nick Littlestone for their critiques on an earlier draft of this paper.

References

[1] D. Angluin and C.H. Smith, Inductive inference: Theory and methods, Comput. Surv. 15 (3) (1983) 327–369.

[2] M.R. Garey and D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness (Freeman, San Francisco, CA, 1979).

[3] J. Pearl, On the connection between the complexity and credibility of inferred models, Internat. J. General Systems 4 (1978) 255–264.

[4] L. Pitt and L.G. Valiant, Computational Limits on Learning from Examples, Tech. Rept., Dept. of Computer Science, Harvard Univ., to appear.

[5] L.G. Valiant, A theory of the learnable, Comm. ACM 27 (11) (1984) 1134–1142.