LIMITATIONS ON INDUCTIVE LEARNING*
(Extended Abstract)

Thomas G. Dietterich
Department of Computer Science
Oregon State University
Corvallis, OR 97331
tgd@cs.orst.edu

ABSTRACT

This paper explores the proposition that inductive learning from examples is fundamentally limited to learning only a small fraction of the total space of possible hypotheses. We begin by defining the notion of an algorithm *reliably* learning a good approximation to a concept C. An empirical study of three algorithms (the classical algorithm for maximally specific conjunctive generalizations, ID3, and back-propagation for feed-forward networks of logistic units) demonstrates that each of these algorithms performs very poorly for the task of learning concepts defined over the space of Boolean feature vectors containing 3 variables. Simple counting arguments allow us to prove an upper bound on the maximum number of concepts reliably learnable from m training examples.

INTRODUCTION

How good are current inductive learning algorithms? How well can any inductive learning algorithm perform? This paper addresses these questions for the case of learning concepts defined over the universe of Boolean n-tuples.

Most work in the probably-approximately correct (PAC) learning theory yields results of the form "If the learning algorithm searches a space of hypotheses H and finds an hypothesis $\hat{h} \in H$ consistent with all m given training examples, and if m is large enough, then $\hat{h}$ is probably approximately correct." The goal of this paper is to turn these results around and ask "Suppose we are given m training examples, what is the size of the largest space of concepts H such that if $h \in H$ is the correct hypothesis, a given learning algorithm will find an hypothesis $\hat{h}$ that is probably approximately correct?"

We approach this question by first defining a new notion, frequently approximately correct (FAC) learning, that assumes the uniform probability distribution over the space of training examples (sampling without replacement). Then, we report the results of an experiment on three existing learning algorithms to determine the number of hypotheses that each algorithm can FAC-learn. Finally, we derive an upper bound on the maximum number of concepts FAC-learnable by any algorithm. The results suggest either that the upper bound is not tight or else that current algorithms are not very good. In either case, the upper bound demonstrates that only a small fraction of the space of possible hypotheses is FAC-learnable by any inductive learning algorithm.

NOTATION

Following the usual practice in PAC-learning theory, we define the set U to be the space of all Boolean n-tuples. A *concept* h is a subset of U, so there are $2^{|U|} = 2^{2^n}$ possible concepts definable over U. An

*This work was supported in part by NSF under grant numbers IRI-86-57316 (Presidential Young Investigator Award) and CCR-87-16748 and by gifts from Tektronix and SUN Microsystems. Thanks also to Hussein Almuallim for assisting with Corollary 1.

example of a concept h is a pair of the form $(u,+)$ if $u \in h$ and $(u,-)$ otherwise. A training sample of size m is a set S of m distinct examples.

Suppose a learning algorithm is given a sample S and produces as output the hypothesis $\hat{h}$. We say that the error of $\hat{h}$ is the fraction of U that is incorrectly classified by $\hat{h}$. This is equal to $\frac{|h \oplus \hat{h}|}{2^n}$, where $\oplus$ denotes the disjoint union. This error measure is a special case of the PAC error measure for learning problems where the examples are drawn without replacement according to the uniform distribution.

Let T be the total number of possible training sets of size m for a given concept h. Since there are 2^n possible training examples, there are $T = \binom{2^n}{m}$ possible training sets.

We say that an algorithm *frequently approximately correctly* (FAC) learns a concept h if for $(1-\delta)T$ training sets, the guess $\hat{h}$ returned by the algorithm has error at most ϵ. Let $F_A(m,\epsilon,\delta)$ be number of distinct concepts FAC learnable by learning algorithm A. One goal of inductive learning research is to find an algorithm A that maximizes $F_A(m,\epsilon,\delta)$ for typical values of ϵ and δ.

EXPERIMENTAL RESULTS

We have experimentally measured the F_A of three popular learning algorithms for the case $n = 3, \epsilon = \frac{1}{8}$, and $\delta = \frac{1}{10}$. This case is admittedly small, since there are only 8 possible training examples and 256 possible hypotheses. However, it is the largest case that it has thus far been practical to compute. The three algorithms are

CONJ: the classical algorithm for computing the maximally specific conjunctive Boolean formula consistent with the training set. If there are no positive examples in the training sample, then the algorithm returns the concept NIL (the empty set). If there is no conjunctive concept, the algorithm is considered to have returned a concept with error greater than ϵ.

ID3: a version of Quinlan's popular algorithm for constructing decision trees (Quinlan, 1986). This version employs the information gain criterion to select the root feature for each decision (subtree). Windowing is not performed.

BACK: the version of the error back-propagation algorithm described in (Rumelhart, Hinton, and Williams, 1986). This version employs a learning rate of 0.25 and a momentum term of 0.9. An architecture consisting of 2 hidden units (fully connected to the 3 inputs) and 1 output unit (fully connected to the hidden units) is trained until minimum error is attained (change in total error of less than .0001 after a complete pass over the training set) and no classification errors are made on the training set. Each unit computes the logistic function. For training purposes, an output value of .9 or greater is considered a one; an output .1 or less is considered a zero; and all other output values are indeterminate. For testing purposes, an output is a one if it is greater than .5 and a zero otherwise. If the algorithm is unable to find a consistent network after 10 attempts (each attempt starting with randomized weights), then the algorithm is considered to have returned a concept with error greater than ϵ.

For each possible concept h defined over 3 Boolean features, all $\binom{2^3}{4} = 70$ training sets of size 4 were generated and processed by each algorithm. If on at least 63 of those training sets the algorithm returned an hypothesis that incorrectly classified at most one of the 8 possible examples, then the concept h was FAC-learned by the algorithm. The results are summarized in Table 1.

The results show that only a very small fraction of the 256 possible concepts are FAC-learned by these algorithms. The relative order of the three algorithms is probably not generalizable to larger n, and the reader should not conclude from this experiment that CONJ is superior to ID3 or that ID3 is superior to BACK. The surprising result is that *none* of the algorithms performs very well.

Table 1: Number of concepts FAC-learnable when $n = 3, m = 4, \epsilon = \frac{1}{8}$, and $\delta = \frac{1}{10}$.

Algorithm	Number of FAC-learned concepts
ID3	8
BACK	2
CONJ	10

This demonstrates the fallacy of the following argument: (a) ID3 learns decision trees, (b) any Boolean concept can be represented by a decision tree, therefore (c) ID3 can learn any Boolean concept. This is true only if all of the possible training examples are given to the algorithm. In practice, it is rare for a learning algorithm to have even 50% of the possible training examples available for learning. Similar arguments have been put forward concerning the learning power of back propagation. It should be clear that the expressive power of the hypothesis space is not the only factor to consider in assessing the ability of a learning algorithm to FAC-learn an unknown concept.

To obtain the data for Table 1, each learning algorithm was executed 1,120 times. Unfortunately, to obtain data for the analogous case where $n = 4$ and $m = 8$ would require executing each algorithm 3,294,720 times. Statistical approximations do not substantially decrease this number. We are currently reimplimenting our code on a connection machine to perform these runs.

AN UPPER BOUND

To determine how well any algorithm could do, it is useful to view a learning algorithm as a mapping from training sets to concepts. For a given training set of size m, there are 2^{2^n-m} possible consistent concepts. This is because there are $2^n - m$ remaining examples in U, and each one of them could be classified in 2 possible ways. A learning algorithm must choose one of these consistent concepts (or possibly some inconsistent concept!) as its guess $\hat{h}$.

Now the $\hat{h}$ that it guesses will be a good approximation (error $\leq \epsilon$) for some of the 2^{2^n-m} hypotheses and a bad approximation for the others. From the definition of FAC learning, we see that a concept h is FAC-learnable only if for most of the training sets consistent with h, the guess $\hat{h}$ is a good approximation to h. An algorithm will perform badly if it tends to "scatter" its guesses, so that for some training sets consistent with h, the guess $\hat{h}$ is good and for others it is bad. An algorithm will perform well if it can more-or-less concentrate its guesses on a subset of the possible hypotheses. This perspective allows us to prove the following theorem.

Theorem 1 *If $m \leq (1-\epsilon)2^n$, then no learning algorithm can FAC-learn more than*

$$\frac{2^m \sum_{i=0}^{\epsilon 2^n} \binom{2^n-m}{i}}{1-\delta}$$

concepts from m training examples, for error parameter ϵ and confidence parameter δ.

Proof:

For a training set S, when a learning algorithm A makes a guess, $\hat{h}$, there are at most

$$Ball(\epsilon) = \sum_{i=0}^{\epsilon 2^n} \binom{2^n - m}{i}$$

concepts that are within ϵ of $\hat{h}$ and consistent with S. This is because there are exactly $\binom{2^n-m}{i}$ concepts at Hamming distance i from $\hat{h}$, and we sum for Hamming distances from 0 up to $\epsilon 2^n$. The binomial coefficient is well-defined only when $2^n - m \geq \epsilon 2^n$, or $m \leq (1-\epsilon)2^n$. Let us call these ϵ-close concepts "wins." Similarly, there are at least

$$2^{2^n-m} - Ball(\epsilon)$$

concepts that have error more than ϵ from $\hat{h}$. Let us call these "losses."

Because there are $\binom{2^n}{m}2^m$ training sets, no learning algorithm can create more than $\binom{2^n}{m}2^m Ball(\epsilon)$ wins. We will call a concept h a winner if in at least $(1-\delta)\binom{2^n}{m}$ of the training sets with which it is consistent, it receives a "win". A winner is therefore FAC-learnable. An optimal FAC algorithm can do no better than to allocate exactly $(1-\delta)\binom{2^n}{m}$ wins to each winner. This spreads the wins as widely as possible and therefore maximizes the number of winners. Let W be the maximum number of winners created by any FAC-learning algorithm. By dividing the maximum number of wins by the minimum number of wins needed to create a winner, we obtain the following bound:

$$W \leq \frac{\binom{2^n}{m}2^m Ball(\epsilon)}{(1-\delta)\binom{2^n}{m}}.$$

Simplifying, this gives us

$$W \leq \frac{2^m Ball(\epsilon)}{1-\delta} = \frac{2^m \sum_{i=0}^{\epsilon 2^n}\binom{2^n-m}{i}}{1-\delta}.\square$$

When $m = 4, n = 3, \epsilon = \frac{1}{8}$, and $\delta = \frac{1}{10}$, this quantity is 88. Comparison with Table 1 suggests either that our bound is too high or else that existing learning algorithms could stand significant improvement. In either case, however, this theorem puts a bound on the fraction of the 2^{2^n} concepts that can be FAC-learned from examples.

While Theorem 1 gives useful answers for small values of n, it is surely an overestimate for large n, since it grows as $O(2^{n2^n})$. Another way of deriving a bound is to apply the following theorem proved by Ehrenfeucht, Haussler, Kearns, and Valiant (1988):

Theorem 2 *Assume $0 < \epsilon \leq \frac{1}{8}, 0 < \delta \leq \frac{1}{100}$, and $VCdim(H) \geq 2$. Then any learning algorithm A that PAC learns every concept in H for any probability distribution P over U must use sample size*

$$m \geq \frac{VCdim(H) - 1}{32\epsilon}.$$

Here the $VCdim(H)$ is the Vapnik-Chervonenkis dimension of H (Blumer, Ehrenfeucht, Haussler, and Warmuth, in press). Natarajan (in press) has proved that $VCdim(H) \geq \left\lceil \frac{1}{n+2} \lg |H| \right\rceil$. Hence, by combining these results and solving for $|H|$, we obtain the following bound:

Corollary 1 *Assume $0 < \epsilon \leq \frac{1}{8}, 0 < \delta \leq \frac{1}{100}$, and $VCdim(H) \geq 2$. Then given m training examples, the number of hypotheses any algorithm A can PAC learn is bounded by*

$$|H| \leq 2^{(n+2)(32\epsilon m)}.$$

For practical cases, m will be a polynomial function of $n, \frac{1}{\epsilon}$, and $\frac{1}{\delta}$. Hence, Corollary 1 states that $|H|$ can only grow as $2^{poly(n)}$ for some polynomial. Since there are 2^{2^n} possible concepts, this means that for reasonable sample sizes, only a small fraction of the possible concepts can be learned from examples under arbitrary distributions.

It is important to realize that the bound in Corollary 1 is not directly comparable to Theorem 1, because Corollary 1 requires that every concept in H be learnable from any probability distribution P over U. Theorem 1, on the other hand, is only concerned with the case where the probability distribution P is uniform. It is likely that fewer examples are required to learn under a fixed distribution than under an unknown distribution. Hence, for a given number of training examples m, it is likely that a larger number of concepts is learnable from a fixed distribution.

IMPLICATIONS

The fact that inductive learning methods are fundamentally limited to learning only a small fraction of all possible hypotheses has many implications.

First, it means that there are no general purpose learning methods that can learn any concept (from a sample of reasonable size). Instead, different classes of learning problems may call for different learning algorithms. An important problem for future research is to attempt to identify relationships between types of learning problems (e.g., problems in speech understanding) and types of hypothesis spaces (e.g., decision trees, neural nets, etc.).

Second, the results suggest that human learning involves much more than learning from positive and negative training examples. It is unlikely that human learning is limited in the way that these inductive learning algorithms are limited, since people seem to be able to learn well in a wide variety of domains. If these results accurately modeled human learning situations, one would expect that people would only succeed in learning a small proportion of the "concepts" that they face in daily life.

Third, these results underline the importance of studying actual learning situations to determine what prior knowledge and sources of information (including training examples) are available to the learner. Research aimed at understanding how prior knowledge and other sources of information can be exploited by the learning process is also very important.

Fourth, if the upper bounds can be tightened, and I believe they can, then the results would indicate that further work on inductive learning methods—including methods that construct "new terms"—is unlikely to produce significant improvements in learning performance. In any case, algorithms that introduce new terms cannot overcome these upper bounds.

Future work must focus on reducing the difference between the performance of existing algorithms and the upper bound.

BIBLIOGRAPHY

Blumer, A., Ehrenfeucht, A., Haussler, D., and Warmuth, M. K., (in press). Learnability and the Vapnik-Chervonenkis dimension. To appear in *Journal of the ACM.*

Ehrenfeucht, A., Haussler, D., Kearns, M., and Valiant, L., (1988). A general lower bound on the number of examples needed for learning. *COLT 88: Proceedings of the Conference on Learning Theory*, Los Altos, CA: Morgan-Kaufmann. 110–120.

Natarajan, B. K. (In press). On learning sets and functions. Unpublished manuscript.

Quinlan, J. R. (1986). Induction of Decision Trees, *Machine Learning,* 1(1), 81–106.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning internal representations by error propagation. In Rumelhart, D. E., and McClelland, J. L., (eds.) *Parallel Distributed Processing*, Vol 1. 318–362.