Nonuniform Learnability*

GYORA M. BENEDEK

ELBIT Ltd., Haifa, Israel

AND

ALON ITAI[†]

Computer Science Department, Technion, Haifa, Israel

Received January 20, 1991; revised January 20, 1993

The learning model of Valiant is extended to allow the number of examples required for learning to depend on the particular concept to be learned, instead of requiring a uniform bound for all concepts of a concept class. This extension, called *nonuniform learning*, enables learning many concept classes not learnable by the previous definitions. Nonuniformly learnable concept classes are characterized. Some examples (Boolean formulae, recursive, and r.e. sets) are shown to be nonuniformly learnable by a polynomial (in the size of the representation of the concept and in the error parameters) number of examples, but not necessarily in polynomial time. Restricting the learning protocol such that the learner has to commit himself after a finite number of examples does not affect the concept classes which can be learned. An extension of nonuniform learnability to nonuniform learnability with respect to specific distributions is presented. © 1994 Academic Press, Inc.

1. INTRODUCTION

In his seminal paper [21], Valiant introduced the PAC model,[1] according to which the teacher selects a concept (Boolean formula) from a commonly known concept class and provides the student with examples selected at random by some distribution D unknown to the student. An example is one element of the domain and a label specifying whether it is contained in the concept. A concept class is learnable if there is an algorithm that, after receiving sufficiently many examples, finds, with high probability, an approximation to the concept. The success of the algorithm should be independent of the distribution used by the teacher. The learning algorithm is *polynomial* if the sample size and computation time

* A Preliminary version appeared in "ICALP 1988."

[†] This research was supported by the Fund for the Promotion of Research at the Technion.

[1] The exact definition appears in the next section.

0022-0000/94 $6.00

are polynomial in the size of the representation of the concept and the error parameters.

Blumer *et al.* [9] generalized the notion of learnability to arbitrary domains and concept classes. However, they discuss only the number of examples, not the computation time. This number is allowed to depend on the concept class and the error parameters but is required to be independent of the concept to be learned. Necessary and sufficient conditions for learnability using the Vapnik–Chervonenkis dimension [20] are given. This dimension depends only on the structure of the concept class (and is independent of the distribution). They show that a concept class is learnable if and only if it has finite dimension.

In the next section we show intuitively learnable concept classes which are not learnable according to this definition. To broaden the definition of learnability we consider nonuniform learnability, i.e., we allow the number of examples to depend not only on the concept class and error parameters but also on the concept to be learned. We give a necessary and sufficient condition for this case: a concept class is nonuniformly learnable if and only if it is a countable union of subclasses each of finite dimension.

This definition has the weakness that, even though by subsequently increasing the number of examples a "good" approximation of the concept can be found, there is no way to determine when to stop this process. A priori, it seems that to demand that the student "know" when he has succeeded in learning strengthens the definition. We show that both definitions are equivalent.

These general principles yield some interesting results about the learnability of recursive, r.e., context-free languages, and general Boolean formulae. In particular we give a partial solution to an open problem first raised by Valiant [21], i.e., "is the set of Boolean formulae polynomially learnable?" We show that Boolean formulae are learnable by polynomially many examples. However, the learning time of our algorithm is not polynomial.

The papers [21–23, 17, 18, 15] deal with nonuniform learnability of Boolean formulae only. Their definition of polynomial learnability requires both the number of examples and the running time of the learning algorithm to be polynomial in the error parameters and in the length of the Boolean formula to be learned.

Blumer *et al.* [10] investigate learning functions from a countable set into a finite domain. They define a class of algorithms *Occam algorithms* as algorithms that, when receiving a sample of a function f, produce a hypothesis subject to certain constraints. The complexity of the algorithm is polynomial in the number of examples and the complexity of f. The main result of [10] is that the existence of an Occam algorithm implies polynomial learnability. We discuss learnability of recursive functions whose range is infinite. Furthermore, by our results, some of the constraints on the Occam algorithm may be relaxed (e.g., we allow noncountable domains). None of the aforementioned papers refer to the question whether the student "knows" when to stop.

In a recent paper [7] we have defined uniform learnability for an arbitrary distribution D and have given necessary and sufficient conditions for a concept class C to be uniformly learnable with respect to D. Here we broaden this notion by considering nonuniform learnability with respect to D. We give necessary and sufficient conditions for this case also.

After the preliminary version of this paper appeared [8], Lineal *et al.* [16] investigated similar notions. They independently proved the "sufficient" direction of Theorem 2, while we also show that the condition is necessary.

In [6] Ben-David *et al.* parameterize learnability models according to sampling complexity issues. Among other problems they consider the cases where the number of examples may depend on the target concept or on the distribution or on both. They present a classification for families of learnable concept classes using results from the current paper and additional analysis.

2. Definitions

Following [9], let X be a set, $R \subseteq 2^X$ a σ-algebra over X [12], and D a probability measure over R. A *concept class* is a set $C \subseteq R$ of *concepts.* For $\mathbf{x} = (x_1, ..., x_l) \in X^l$, the *labeled l-sample of* $c \in C$ is given by $\mathrm{sam}_c(\mathbf{x}) = (\langle x_1, c(x_1)\rangle, ..., \langle x_l, c(x_l)\rangle)$, where $c(x)$ equals one if $x \in c$ and 0 otherwise. The *sample space of* C, denoted S_C, is the set of all labeled l-samples of c over all $c \in C$, $\mathbf{x} \in X^l$ $(l \geqslant 1)$. For a concept class C on X, F_C denotes the set of all functions $f\colon S_C \to C$.

We use the following protocol ([21] and others) between two agents, T (teacher) and L (learner): T (who wants to teach L the concept c, called the *target* concept) repeatedly picks, at random according to some distribution D, an element x from the set X and sends the pair $\langle x, c(x)\rangle$ to L. L, after receiving sufficiently many examples, returns a concept. We may view L as a function $f \in F_C$, and the set that L returns is $f((\langle x_1, c(x_1)\rangle, ..., \langle x_l, c(x_l)\rangle))$.

Let $Y_1, Y_2 \in R$. Y_1 and Y_2 are *ε-close with respect to the distribution D* if $D(Y_1 \oplus Y_2) < \varepsilon$ ($\oplus$ denotes the symmetric difference). Otherwise, Y_1 and Y_2 are *ε-far with respect to the distrubition D.*

Let $\mathbf{x} = (x_1, ..., x_l)$ be a sequence of l independent randomly selected elements of X. For $f \in F_C$ and $c \in C$ we define the following probabilities $r_f^+(D, c, l, \varepsilon) = \mathrm{Prob}_{\mathbf{x}}(f(\mathrm{sam}_c(\mathbf{x}))$ is a subset of X ε-close to c with respect to D). This is the probability that f finds a "good" approximation for c using an l-sample of c. We now follow [9] and define PAC (probably approximately correct) learnability:

Uniform learnability [9]. A function $f \in F_C$ *uniformly learns a concept class C* if for every $\varepsilon, \delta > 0$ there is an $l = l(\varepsilon, \delta) > 0$ such that for every distribution D over R and every $c \in C$, $r_f^+(D, c, l, \varepsilon) > 1 - \delta$. C is *uniformly learnable* if there exists an $f \in F_C$ that uniformly learns C.

The definition above is quite stringent:

Example 1. Let X be the open segment (0, 1) and for every n let C_n be the set of all unions of n open segments over X. C_n is uniformly learnable, see [9]. Let

$C=\bigcup_{n=1}^{\infty} C_n$ then C is not uniformly learnable [9] (for further details see Section 4).

On the other hand, intuitively, since every $c \in C$ belongs to some C_i and since C_i is uniformly learnable [9], c is also learnable by a finite number of examples. Note that here the number of examples depends on the concept. Thus, it is reasonable to expect that, the more complex the concept $c \in C$, the more difficult it is to learn. This leads to the definition which allows the number of examples to depend on the concept to be learned.

Nonuniform learnability. A function $f \in F_C$ *nonuniformly learns* a concept class $C \subseteq R$ if for every $\varepsilon, \delta > 0$ and every $c \in C$ there is an $l_0 = l(\varepsilon, \delta, c) > 0$ such that for every D over R and $l \geqslant l_0$, $r_f^+(D, c, l, \varepsilon) > 1 - \delta$. Note that here l_0 depends also on c. C is *nonuniformly learnable* if there exists an $f \in F_C$ that nonuniformly learns C. We say that $l(\varepsilon, \delta, c)$ is the *sample complexity* of f.

The intuition behind this definition is that for a fixed target concept $c \in C$, with high probability, the hypotheses output by f become closer to c as the number of examples increases. In Section 4 we show that the concept class of Example 1 is nonuniformly learnable.

EXAMPLE 2. Let $X = (0, 1)$ and C_{OPEN} the set of all open sets over X. In Section 4 we show that C_{OPEN} is not nonuniformly learnable.

3. PREVIOUS RESULTS—UNIFORM LEARNABILITY

In this section we quote the result presented in [9] (see also [20]), which will be used in the next section. Let T be a subset of X, a concept class $C \subseteq 2^X$ *shatters* T if for every subset T' of T there is a concept $c \in C$ such that $T \cap c = T'$. Also, C has Vapnik–Chervonenkis dimension d (VC-dimension) ($\dim(C) = d$) if there is set of d elements of X shattered by C and there is no set of $d+1$ elements shattered by C. If there is no such d then C has infinite VC-dimension.

EXAMPLE 3. Let $X = (0, 1)$ and C be the set of all open intervals over X. It is easy to see that C shatters sets of two points but there is no set of three points shattered by C, thus $\dim(C) = 2$. The concept class C defined in Example 1 shatters any finite set thus has infinite VC-dimension.

The main result of [9] is that C is learnable if and only if $\dim(C)$ is finite. A slightly stronger result appeared in [11].

THEOREM 1 [9, 11]. *If* $\dim(C) = d \geqslant 2$ *then*:

1. *For every* $\varepsilon, \delta > 0$ *any function that receives* $\max\{(4/\varepsilon)\ln(2/\delta), (8d/\varepsilon)\ln(13/\varepsilon)\}$ *examples and returns a concept in C consistent with the examples uniformly learns C.*

2. *For every* $0<\varepsilon\leqslant\frac{1}{8}$ *and* $0<\delta\leqslant\frac{1}{100}$ *there exists no function that can uniformly learn* C *using* $\Omega((1/\varepsilon)\ln(1/\delta)+d/\varepsilon)$ *examples.*

In order for part 1 of the theorem to hold, the pair (C, D) of concept class and distribution, has to be *well behaved.* The precise definition and proofs appear in [5, 9].

4. A Necessary and Sufficient Condition for Nonuniform Learnability

Theorem 2. *C is nonuniformly learnable if and only if there is an infinite sequence $C_1, C_2, \ldots$ such that*

1. $C=\bigcup_{i=1}^{\infty} C_i$,
2. $\dim(C_i)<\infty$ *for every* $i=1, 2, \ldots$.

Proof. If C is nonuniformly learnable then there is an $f\in F_C$, such that every $c\in C$ is learnable by f with sample complexity $l(\varepsilon, \delta, c)$. Let $\varepsilon=\delta=\frac{1}{100}$ then for $i=5, 6, 7, \ldots$ let C_i be the set of concepts c such that $l(\frac{1}{100}, \frac{1}{100}, c)\leqslant i$; i.e., f can learn every $c\in C_i$ from i examples with accuracy and confidence $\frac{1}{100}$. Let k be the implied constant in the lower bound of Theorem 1. Then $i\geqslant k(\frac{1}{100}\ln\frac{1}{100}+\dim(C_i)/100)$. Therefore, $\dim(C_i)\leqslant 100i/k+\ln 100$. Since C is nonuniformly learnable, every $c\in C$ belongs to some C_i, i.e., $C=\bigcup_{i=1}^{\infty} C_i$.

For the other direction, let $C=\bigcup_{i=1}^{\infty} C_i$ and $\dim(C_i)<\infty$. Since the property of having finite VC-dimension is closed under finite union, we may assume that $C_i\supseteq\bigcup_{j=1}^{i} C_j$, and therefore $\dim(C_i)$ is a nondecreasing series. Let $K(\varepsilon, \delta, i)=\max\{(4/\varepsilon)\ln(2/\delta), (8\dim(C_i)/\varepsilon)\ln(13/\varepsilon)\}$. From its definition K is nondecreasing in $1/\varepsilon$, $1/\delta$, and i.

First we describe a function f that receives ε, δ, and l examples and returns a hypothesis $h\in C$:

1. Let i be the largest integer i such that $l\geqslant K(\varepsilon, \delta, i)$.

2. Return a concept $h\in C_i$ consistent with the examples. If no such h exists then return a prespecified $c_0\in C$.

Now we prove that f learns C. Let c be some concept in C. Then there exists an i_0 such that for all $i\geqslant i_0$, $c\in C_i$. Given l, f uses 1 to compute i and returns a concept from C_i consistent with the sample. By part 1 of Theorem 1, if $i>i_0$, f learns c.

We modify f so that it will not need to receive ε and δ as parameters:

1′. Let i be the largest integer i such that $l\geqslant K(i^{-1}, i^{-1}, i)$.

Thus when l increases, i also increases and ε and δ monotonically decrease to zero. For every $\varepsilon, \delta>0$ and $c\in C$ there exists an integer i such that $\varepsilon\geqslant i^{-1}$, $\delta\geqslant i^{-1}$, and $c\in C_i$. It follows that, $l=K(i^{-1}, i^{-1}, i)$ examples are sufficient to learn c. ▌

The equivalence of the models where f does not receive ε and δ as parameters (functional model) and the case where it does was discussed also in [13].

COROLLARY 1. *Any countable class of concepts is nonuniformly learnable.*

EXAMPLE 4. Consider X and C_n of Example 1. We have $\dim(C_n) = 2n$ and $C = \bigcup_{i=1}^{\infty} C_i$; thus C is nonuniformly learnable.

CLAIM 1. *The concept class C_{OPEN} of Example 2 is not nonuniformly learnable.*

Proof. First note that the set $\{1/n : n \in \mathbb{N}\}$ is shattered by C_{OPEN}. The following lemma implies the claim.

LEMMA 1. *If a concept class C shatters an infinite set then there is no sequence $C_1, C_2, \ldots$ such that*

1. $C = \bigcup_{i=1}^{\infty} C_i$
2. $\dim(C_i) < \infty$ *for every* $i = 1, 2, \ldots$.

Proof (Shai Ben-David). Let $C = \bigcup_{i=1}^{\infty} C_i$ such that $d_i = \dim(C_i) < \infty$ for every $i = 1, 2, \ldots$. We show that every infinite set $T \subseteq X$ has a subset B such that for every $c \in C$, $c \cap T \neq B$ and thus C does not shatter T. For every n let A_n be a set of $d_n + 1$ elements of $T - \bigcup_{i=1}^{n-1} A_i$. Since $\dim(C_n) = d_n$, there exists a set $B_n \subseteq A_n$, such that for every $c \in C_n$, $c \cap A_n \neq B_n$. B is defined as $\bigcup_{i=1}^{\infty} B_i$. From the definitions $B_n = B \cap A_n$. Let $c \in C$, we now show that $c \cap T \neq B$. Since $C = \bigcup_{i=1}^{\infty} C_i$ let $c \in C_n$ and consider the two sets $A_n \cap (c \cap T)$ and $A_n \cap B$: Since $A_n \subseteq T$, $A_n \cap (c \cap T) = A_n \cap c$. If $c \cap T = B$ then $A_n \cap (c \cap T) = B \cap A_n$ implying that $A_n \cap c = B_n$—a contradiction. ▮

Shelah [19] showed that the converse of Lemma 1 does not holds.

The definition of nonuniform learnability has a considerable practical disadvantage. A user does not know how many examples to give f. She might be tempted to search for better and better hypotheses (ask for more examples) even though she has already reached a sufficiently close approximation to c. In other words, an algorithm that uses such a function to learn a concept has no indication when to stop. Thus our definition of (nonuniform) learning, resembles learning in the limit of inductive inference [3]. To overcome this difficulty we give the following definition.

DEFINITION. A concept class $C \subseteq R$ is *nonuniformly strongly learnable* if there exists a function $f \in F_C$ and an algorithm $A_f(\varepsilon, \delta)$ which has access to a source of random examples and to f, such that for all $\varepsilon, \delta > 0$ and every $c \in C$ there is an $l > 0$ such that for every distribution D over R, with probability at least $1 - \delta$ the algorithm requires at most l many examples and returns a hypothesis ε-close to c (with respect to D).

THEOREM 3. *A concept class C is nonuniformly strongly learnable if and only if it is nonuniformly learnable.*

Proof. Suppose C is nonuniformly strongly learnable by A_f. Construct a function $g \in F_C$ as follows: Let the examples be $(\langle x_1, c(x_1)\rangle, \ldots, \langle x_l, c(x_l)\rangle)$. For

$i = 1, 2, \ldots$ evaluate $A_f(i^{-1}, i^{-1})$ while providing the examples $\langle x_1, c(x_1)\rangle, \ldots$ to get a hypothesis h_i. Stop when A_f asks for more than l examples, and return the last hypothesis produced by A_f. (If A_f has already asked for more than l examples then return some prespecified concept c_0.) It is easy to see that g learns C nonuniformly.

For the other direction, we use the "generate and test" paradigm as in Angluin *et al.* [1, 2].

Let C be a nonuniformly learnable concept class, and f a learning function for C. Thus, by Theorem 2, $C = \bigcup_{i=1}^{\infty} C_i$, where $\dim(C_i) < \infty$ for every i. As before, we may assume that $C_i = \bigcup_{j=1}^{i} C_j$, thus $\dim(C_i)$ is nondecreasing. Let $L_0 = \lceil (32/\varepsilon)((\ln(3/\delta)) - \ln(1 - e^{-\varepsilon/32}))\rceil$. The algorithm consists of iterations; the ith iteration is

1. Let $\mathbf{x}_i$ consist of 2^i examples.
2. Let $h_i = f(\mathbf{x}_i)$ be the hypothesis of this iteration.
3. Let $\mathbf{y}_i$ consist of $L_i = L_0 + i$ additional examples.
4. Let n_i equal the number of y_k's in $\mathbf{y}_i$ on which c and h_i disagree.
5. If $n_i \leqslant \lfloor \frac{3}{4}\varepsilon L_i \rfloor$ then return h_i. Otherwise continue to the next iteration.

Since C is nonuniformly learnable, for each target concept c, and $\varepsilon, \delta > 0$ there exists an $l = l(\varepsilon/2, \delta/3, c)$ such that when f is given l random examples it produces a concept $\varepsilon/2$-close to c with probability at least $1 - \delta/3$. Let C_j be the first C_i that contains c. Let $l_j = \max\{(12/\varepsilon)\ln(6/\delta), (16\dim(C_j)/\varepsilon)\ln(26/\varepsilon)\}$. By Theorem 1, l_j examples are sufficient to learn any concept of C_j with accuracy $\varepsilon/2$ and confidence $\delta/3$.

The algorithm may fail to learn for the following reasons:

1. It stopped with an h_i which is ε-far from c.
2. It did not halt.

We will use the following Chernoff inequalities (see [4, Proposition 2.4]):
For all n, p, β with $0 \leqslant p \leqslant 1$, $0 \leqslant \beta \leqslant 1$,

$$\sum_{k=0}^{\lfloor (1-\beta)np \rfloor} \binom{n}{k} p^k (1-p)^{n-k} \leqslant e^{-\beta^2 np/2}, \tag{1}$$

$$\sum_{k=\lceil (1+\beta)np \rceil}^{n} \binom{n}{k} p^k (1-p)^{n-k} \leqslant e^{-\beta^2 np/3}. \tag{2}$$

1. Let P_i be the probability that $n_i < \frac{3}{4}\varepsilon L_i$ given that h_i is ε-far from c. From (1) we obtain

$$\begin{aligned} P_i &\leqslant \sum_{j=0}^{\lfloor (3/4)\varepsilon L_i \rfloor} \binom{L_i}{j} \varepsilon^j (1-\varepsilon)^{L_i - j} \\ &\leqslant e^{-(1/4^2)L_i(\varepsilon/2)} = e^{-L_i\varepsilon/32} \\ &= e^{(-L_0 - i)\varepsilon/32} = e^{-L_0\varepsilon/32}(e^{-\varepsilon/32})^i. \end{aligned}$$

Thus the probability of failure in this case is

$$\begin{aligned}
\sum_{i=0}^{\infty} P_i &\leqslant e^{-L_0\varepsilon/32} \sum_{i=0}^{\infty} (e^{-\varepsilon/32})^i \\
&= e^{-L_0\varepsilon/32} \frac{1}{1-e^{-\varepsilon/32}} \\
&= e^{-\lceil (32/\varepsilon)[(\ln(3/\delta)) - \ln(1-e^{-\varepsilon/32})]\rceil(\varepsilon/32)} \frac{1}{1-e^{-\varepsilon/32}} \\
&\leqslant e^{-(32/\varepsilon)[(\ln(3/\delta)) - \ln(1-e^{-\varepsilon/32})](\varepsilon/32)} \frac{1}{1-e^{-\varepsilon/32}} \\
&= \frac{\delta}{3}.
\end{aligned}$$

2. Consider iteration $r = \lceil \log_2 l_j \rceil$. By Theorem 1, there is probability at most $\delta/3$ that h_r is $\varepsilon/2$-far from c. Thus it remains to consider only the case that h_r is $\varepsilon/2$-close and the algorithm did not halt. Let Q be the probability that $n_r > \lfloor \frac{3}{4}\varepsilon L_r \rfloor$. From (2) we obtain

$$\begin{aligned}
Q &\leqslant \sum_{k=\lfloor (3/4)\varepsilon L_r \rfloor + 1}^{L_r} \binom{L_r}{k} \left(\frac{\varepsilon}{2}\right)^k \left(1-\frac{\varepsilon}{2}\right)^{L_r-k} \leqslant \sum_{k=\lceil (3/4)\varepsilon L_r \rceil}^{L_r} \binom{L_r}{k} \left(\frac{\varepsilon}{2}\right)^k \left(1-\frac{\varepsilon}{2}\right)^{L_r-k} \\
&\leqslant e^{-2^{-2}L_r\varepsilon/6} = e^{-L_r\varepsilon/24} \leqslant e^{-L_0\varepsilon/24} < \frac{\delta}{3}.
\end{aligned}$$

Thus the probability of this event is less than $\frac{2}{3}\delta$. ▌

Remark 1. The number of examples needed is

$$\sum_{i=1}^{r} (2^i + L_i) < 2^{r+1} + rL_0 + \tfrac{1}{2}r^2 = O(2^r + rL_0) = O(2^r + rl_j) = O(l_j \log l_j),$$

where l_j was defined in the proof.

Remark 2. If f is computable then so is the algorithm.

5. Classes of Languages Learnable by Polynomially Many Examples

Theorem 4. *Let $X = \{0, 1\}^*$ (the set of all finite bit-strings) and C be any set of recursive languages. Then C is nonuniformly strongly learnable. Moreover, for every target concept $c \in C$ if M is a Turing machine with j states that recognizes c, then c is learnable with accuracy ε and confidence δ by $O((1/\varepsilon)\log^2(1/\delta) + (j/\varepsilon)\log^3(j/\varepsilon))$ many examples.*

Proof. Let C_k be the set of recursive languages recognizable by Turing machines with k states or less. Clearly $C=\bigcup_{i=1}^{\infty} C_i$ and since C_k is finite so is its VC-dimension. Thus C is nonuniformly learnable.

To show that the number of examples needed is polynomial we find the VC-dimension of C_j. The number of languages in C_j is bounded by the number of Turing machines with j states or less. Using the standard model for Turing machines [14] (the head can move to the left, right, or remain stationary) over the alphabet $\Sigma=\{0,1\}$ there are at most $(6j)^{2j}$ Turing machines with j states or less. (Note that any Turing machine with less than j states is equivalent to one with exactly j states and a transition function that never reaches some of the states.) Since for any finite class C, $\dim(C)\leqslant\log_2(|C|)$, $\dim(C_j)\leqslant 2j\log_2 6j$.

By Remark 1, $O(l_j\log l_j)$ examples suffice. Where

$$l_j=O\left(\max\left\{\frac{1}{\varepsilon}\log\frac{1}{\delta},\frac{\dim(C_j)}{\varepsilon}\log\frac{1}{\varepsilon}\right\}\right)=O\left(\max\left\{\frac{1}{\varepsilon}\log\frac{1}{\delta},\frac{j}{\varepsilon}(\log j)\log\frac{1}{\varepsilon}\right\}\right).$$

Case 1. $\log(1/\delta)\geqslant j\log j\log(1/\varepsilon)$.

$$O(l_j\log l_j)=O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\left[\log\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\right)\right]\right)=O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\left[\log\frac{1}{\varepsilon}+\log\log\frac{1}{\delta}\right]\right)$$

$$=O\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\left[\log\frac{1}{\delta}+\log\log\frac{1}{\delta}\right]\right)=O\left(\frac{1}{\varepsilon}\log^2\frac{1}{\delta}\right).$$

Case 2. $\log(1/\delta)<j\log j\log(1/\varepsilon)$.

$$O(l_j\log l_j)=O\left(\frac{j}{\varepsilon}\log j\log\frac{1}{\varepsilon}\left[\log\frac{1}{\varepsilon}+\log j+\log\log j+\log\log\frac{1}{\varepsilon}\right]\right)$$

$$=O\left(\frac{j}{\varepsilon}\log j\log\frac{1}{\varepsilon}\left[\log\frac{1}{\varepsilon}+\log j\right]\right).$$

If $1/\varepsilon\leqslant j$ then $O(l_j\log l_j)=O((j/\varepsilon)\log^3 j)$. Otherwise, $O(l_j\log l_j)=O((j/\varepsilon)\log^3(1/\varepsilon))$. Therefore, in this case $O(l_j\log l_j)=O((j/\varepsilon)\log^3(j/\varepsilon))$. ∎

As defined, learnability does not require computability. In the above proof we use a function f that, given l_j examples, returns a Turing machine with j or less states consistent with the examples. This function is related to the Kolmogorov complexity and is not recursive.

THEOREM 5. *For words x, y let $K_2(x,y)$ denote the minimum k such that there exists a k state Turing machine that halts on both x and y, accepts x, and rejects y. Then $K_2(x,y)$ is not recursive.*

Proof. Assume, for the sake of contradiction, that K_2 is recursive. Let z be a binary word of length l. Define $z^0=[z_1,...,z_{\lfloor l/2\rfloor}]$ and $z^1=[z_{\lfloor l/2\rfloor+1},...,z_l]$. Let

$K^*(z)$ be the miminum k such that there exists a Turing machine with k states that for $z^0 \neq z^1$ accepts z^0 and rejects z^1. The assumption that K_2 is recursive implies that K^* is also recursive.

Let Z_n be the lexicographically first word z such that $K^*(z) > n$. (Z_n is well defined since for all n there exists a word z for which $K^*(z) > n$.) For every n we can construct a Turing machine, T_n, to produce Z_n:

1. Write the binary representation of n on the working tape.
2. Consider all strings z in lexicographic order.
3. If $z^0 \neq z^1$ compute $K^*(z)$ and if $K^*(z) > n$ copy z to the output tape.

Let k^* be the number of states of K^*. There exists a constant c' such that for all n T_n has at most $k^* + c' + \log n$ states.

Let $K(z)$ be the minimum k such that there is a Turing machine with k states that produces z. From the definition of K,

$$K(Z_n) \leqslant k^* + c' + \log n.$$

We now show that there exists a constant c such that

$$K^*(z) \leqslant K(z) + c.$$

Let c be the number of states of a Turing machine $M^1(u, z)$ that accepts if $u = z^0$ and rejects if $u = z^1$. Let M^z be the Turing machine with $K(z)$ states that produces z and M^*, the Turing machine that on input u first applies M^z to produce z then applies $M^1(u, z)$. M^* accepts z^0, rejects z^1, and has $K(z) + c$ states. However, from the definition of Z_n, $n < K^*(Z_n)$, from which we obtain that there exist constants c and c' such that for infinitely many n,

$$n \leqslant \log n + c + c' + k^*,$$

a contradiction. ▮

The previous theorem does not imply that it is undecidable to learn, only that our proof of Theorem 4 does not yield a recursive algorithm.

Remark 3. Results similar to Theorem 4 may be proved for other concept classes, e.g., the class of r.e. languages.

A set C of recursive languages for which one of the following conditions holds is learnable with polynomially many examples by a computable function:

1. There exists a recursive set S of (encodings of) total Turing machines such that for every $L \in C$ there exists a Turing machine T in S whose language is L.

2. C has bounded complexity; i.e., there is a recursive function g such that for every $L \in C$ there exists a Turing machine T that recognizes L and for every word w, T stops after at most $g(|w|)$ steps, where $|w|$ is the length of w.

COROLLARY 2. *There is a computable function that learns Boolean formulae by a polynomial number of examples.*

Proof. Let f be a Boolean formula with v variables such that f can be encoded by n bits. For a word $b \in \{0, 1\}^*$ we define the value of $f(b)$ as follows: if $|b| \geqslant v$ then assign the first v bits of b to the respective variables; otherwise assign 0 for the unassigned variables. We build a Turing machine M_f that on input $b \in \{0, 1\}^*$ accepts b if and only if $f(b) = \text{true}$. For every f there is such an M_f with $n + O(1)$ states. (Simply write on the input tape the description of $f(b)$ (n states) and then activate a Turing machine that given a v-variable Boolean formula and a Boolean vector of length v evaluates the value of the formula applied to the vector.) Furthermore, the time complexity of the above M_f is bounded. Thus, by condition (2) above, Boolean formulae are learnable by a computable function. ▌

Note that we do not address Valiant's problem concerning the computation time. However, Pitt and Valiant [18] showed several classes of Boolean formulae for which there is no polynomial algorithm that finds a bounded function in that class that is consistent with a set of examples unless $RP = P$. In particular: k-TERM-DNF, k-CLAUSE-CNF (and their monotonic forms) and μ-expressions. Thus even though the number of examples is polynomial, the computation time needed for learning is not necessarily polynomial.

6. NONUNIFORM LEARNABILITY FOR A GIVEN DISTRIBUTION

In order to extend the notion of learnability Benedek and Itai [7] have defined:

Uniform learnability for a given distribution D. C is *learnable with respect to D* if there is a function $f \in F_C$ such that for every $\varepsilon, \delta > 0$ there is an $l = l(\varepsilon, \delta) > 0$ such that for every $c \in C$, $r_f^+(D, c, l, \varepsilon) > 1 - \delta$.

Finite and countable covers. A subset C_ε of 2^X is an *ε-cover of C with respect to D* if for every $c \in C$ there is a $\tilde{c} \in C_\varepsilon$ ε-close to it. *C is finitely coverable (with respect to D)* if for every $\varepsilon > 0$ there is a finite ε-cover C_ε of C. *C is countably coverable (with respect to D)* if for every $\varepsilon > 0$ there is a countable ε-cover C_ε of C. In the sequel we omit D when understood from the context.

The main result of [7] is:

THEOREM 6. *C is finitely coverable (with respect to D) if and only if C is learnable (with respect to D).*

Similarly to distribution-free learnability, learnability for a given distribution can also be extended to the nonuniform form.

Nonuniform learnability for a given distribution D. C is nonuniformly learnable with respect to distribution D if there is a function $f \in F_C$ such that for every $\varepsilon, \delta > 0$ and every $c \in C$ there is an $l = l(c, \varepsilon, \delta) > 0$ such that $r_f^+(D, c, l, \varepsilon) > 1 - \delta$.

THEOREM 7. *C is nonuniformly learnable with respect to D if and only if C is countably coverable with respect to D.*

The next lemma follows directly form the definition of countable coverable.

LEMMA 2. *C is countably coverable with respect to D if and only if there exists a countable class C^* such that for all $\varepsilon > 0$, C^* is an ε-cover of C.*

The proof of Theorem 7 is similar to that of Theorem 2. In the same manner we define:

Nonuniform strong learnability for a given distribution D. C is *nonuniformly strongly learnable with respect to D if there is a function $f \in F_C$ such that for every $\varepsilon, \delta > 0$ and every $c \in C$ there is an $l = l(c, \varepsilon, \delta) > 0$ such that $\sum_{i=1}^{l-1} r_f^-(D, c, i, \varepsilon) < \delta$ and $r_f^+(D, c, l, \varepsilon) > 1 - \delta$.*

The following theorem is analogous to Theorem 3.

THEOREM 8. *A concept class is nonuniformly strongly with respect to D learnable if and only if it is nonuniformly learnable with respect to D.*

EXAMPLE 5. Let C_{OPEN} be as in Example 2 and C be the set of all finite unions of rational intervals. Let D be the uniform distribution over (0, 1). It can be shown that for every $\varepsilon > 0$, C is a countable ε-cover of C_{OPEN} with respect to D. Thus C_{OPEN} is nonuniformly learnable with respect to D. (The above is true for every continuous distribution.)

Remark 4. In [7] the authors prove that for every given distribution D over $\{0, 1\}^*$ every concept class over $\{0, 1\}^*$ is uniformly learnable with respect to D. Thus any concept class of recursive or even r.e. concepts is learnable.

Remark 5. The definition of learnability may be extended to allow random algorithms and random functions. All the theorems mentioned hold also for this case.

7. CONCLUSIONS

The notion of PAC learnability has been extended to the case where the sample size may depend on the target concept. We have shown that a concept class can be learned under this definition if and only if it is a countable union of classes with finite VC-dimension. The problem of characterizing concept classes whose learning time is a fixed polynomial of the size of the target concept and the error parameters still remains open.

ACKNOWLEDGMENT

It is a pleasure to thank Shai Ben-David for contributing Lemma 1.

References

1. D. Angluin, Queries and concept learning, *Mach. Learning* **2** (1988), 319–342.
2. D. Angluin and P. Laird, Learning from noisy examples, *Mach. Learning* **2** (1988), 343–370.
3. D. Angluin and C. H. Smith, Inductive inference: Theory and methods, *Comput. Surveys* **15**, No. 3 (1983).
4. D. Angluin and L. G. Valiant, Fast probabilistic algorithms for Hamiltonian circuits and matchings, *J. Comput. System Sci.* **18** (1979), 155–193.
5. S. Ben-David and G. M. Benedek, "Measurability Constraints on PAC Learnability," Tech. Rep. Department of Computer Science, Technion, Haifa, 1992.
6. S. Ben-David, G. M. Benedek, and Y. Mansour, Parameterization scheme for classifying models of learnability, to appear in *Inform.* and *Comput.* **108** (1994).
7. G. M. Benedek and A. Itai, Learnability by fixed distributions, *Theoret. Comput. Sci.* **88** (1991), 377–389; preliminary version appeared in "COLT '88," (1988).
8. G. M. Benedek and A. Itai, Nonuniform learnability, *in* "15th International Colloquium on Automata, Languages and Programming, (ICALP), Tampere, Finland, 1988," Lecture Notes in Computer Science, Vol. 317 (T. Lepistö and A. Salomaa, Eds.), Springer-Verlag, New York/Berlin, 1988.
9. A. Blumer, A. Ehrenfeucht, D. Haussler and M. Warmuth, Learnability and the Vapnik–Chervonenkis dimension, *J. Assoc. Comput. Mach.* **36**, No. 4 (1989), 929–965.
10. A Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth, Occam's razor, *Inform. Process. Lett.* **24** (1987), 377–380.
11. A. Ehrenfeucht, D. Haussler, M. Kearns, and L. Valiant, A general lower bound on the number of examples needed for learning, *in* "COLT '88," (1988).
12. P. R. Halmos, "Measure Theory," Van Nostrand, New York, 1950.
13. D. Haussler, M. Kearns, N. Littlestone, and M. Warmuth, Equivalence of models for polynomial learnability, *in* "COLT '88," (1988).
14. J. E. Hopcroft and J. D. Ullman, "Introduction to Automata Theory, Languages and Computation," Addison–Wesley, Reading, MA, 1979.
15. M. Kearns, L. Ming, L. Pitt, and L. G. Valiant, On the learnability of Boolean formulae, *in* "Proceedings, 19th Symp. Theory of Comput.," pp. 285–295, Assoc. Comput. Mach., New York, 1987.
16. N. Lineal, Y. Mansour, and R. Rivest, Results on learnability and the Vapnik–Chervonenkis dimension, *in* "Proceedings, 29th FOCS," pp. 120–129, (1988).
17. B. K. Natarajan, On learning Boolean functions, *in* "Proceedings, 19th Symp. Theory of Comput.," pp. 296–304, Assoc Comput. Mach., New York, 1987.
18. L. Pitt and L. G. Valiant, "Computational Limitations on Learning from Examples," Aiken Computation Laboratory, Harvard University, Cambridge, MA 02138, July 1986.
19. S. Shelah, unpublished manuscript.
20. V. N. Vapnik and A. Ya. Chervonenkis, On the uniform convergence of relative frequencies of events to their probabilities, *Theory Probab. Appl.* **16**, No. 2 (1971), 264–280.
21. L. G. Valiant, A theory of the learnable, *Comm. ACM* **27**, No. 11 (1984), 1134–1142.
22. L. G. Valiant, Learning disjunctions of conjunctions, *in* "Proceedings, 9th IJCAI, Los Angeles, CA, August 1985," Vol. 1, pp. 560–566.
23. L. G. Valiant, Deductive learning, Aiken Computational Laboratory, Harvard University, 1984.