NONUNIFORM LEARNABILITY

by

Gyora M. Benedek and Alon Itai

Technical Report #474

November 1987
Abstract

Valiant described a protocol for learning Boolean formulae from examples produced by an arbitrary distribution $D$. His protocol is probabilistic and the learner may fail with probability $\varepsilon$. The computation time depends on the size of the concept and on $\varepsilon$. He has shown certain concept classes to be learnable in polynomial time and if $RP \neq P$ other concept classes to require more than polynomial time.

Blumer et al. considered more general universes and concept classes. They considered only uniform learning, i.e., learning when the number of examples is independent of the concept to be learned.

Since many concept classes which are intuitively learnable are not uniformly learnable, we allow the number of examples to depend also on the concept to be learned and call this variant nonuniform learning.

Our main results are the following:

1. A characterization of nonuniform learning.
2. An exhibition of concept classes which cannot be learned nonuniformly.
3. Showing concept classes (Boolean formulae, recursive and r.e. sets) which can be learned nonuniformly by a polynomial number of examples (but not necessarily in polynomial time).
4. Restricting the learning protocol such that the learner has to commit himself after a finite number of examples does not effect the concept classes which can be learned.
5. An extension of nonuniform learnability to nonuniform learnability w.r.t. specific distributions.
1. Introduction:

In his seminal paper [V1], Valiant based the notion of learnability of Boolean formulae on complexity. In his model the teacher selects one concept (Boolean formula) from a commonly known concept class and provides the student with examples selected at random by some distribution $D$. An example is one element of the domain and a label specifying whether it is contained in the concept or not. A concept class is (polynomially) learnable if there is a (polynomial) algorithm that after receiving (polynomially many) examples finds, with high probability, the concept regardless of the distribution used by the teacher. He actually allowed finding an approximation to the concept, i.e., finding a set whose symmetric difference with the teacher's concept has probability less than a predetermined error parameter.

Blumer et al [BEHW1] generalized the notion of learnability to arbitrary domains and concept classes. In their context, the number of examples depends on the concept class and the error parameter but is independent of the concept to be learned. They give necessary and sufficient conditions for learnability by using the Vapnik-Chervonenkis dimension [VC]. This dimension depends only on the structure of the concept class (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 parameter but also on the concept to be learned. We give necessary and sufficient conditions for this case. I.e., a concept class is nonuniformly learnable for every distribution 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, by demanding from the student to "know" that he succeeded to learn, the definition is strengthened. We show that both definitions are equivalent.

1 see exact definition in the next section
The papers ([V1],[V2],[V3],[N],[PV],[KLPV]) deal with nonuniform learnability for Boolean formulae only. They define polynomial learnability where both the number of examples and the running time of the learning algorithm are polynomial in the error parameter and in the length of some encoding of the Boolean formula to be learned. In [BEHW2] polynomial learnability for countable domains and countably many functions from the domain to a finite set. They defined an Occam algorithm as an algorithm that, when receiving a sample of a function \( f \), produces a hypothesis subject to certain constraints. The algorithms' complexity is polynomial in the number of examples and the complexity of \( f \). The main result of [BEHW2] is that the existence of an Occam algorithm implies polynomial learnability. We deal with recursive functions and allow the range to be infinite. Furthermore, by our results, the constraints on the Occam algorithm can be relaxed. None of the aforementioned papers refer to the question whether the student "knows" when to stop or not.

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 [V1] 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.

In a recent paper [BI] we defined uniform learnability for an arbitrary distribution \( D \) and gave necessary and sufficient conditions for a concept class \( C \) to be uniformly learnable w.r.t. \( D \). Here we broaden this notion by considering nonuniform learnability w.r.t. \( D \). We give necessary and sufficient conditions for this case also.

2. Definitions:

Following [BEHW1], let \( X \) be a set and \( D \) a distribution over \( X \). A concept class is a set \( C \subseteq \mathcal{P}^X \) of concepts. For \( x=(x_1,x_2,\ldots,x_l)\in X^l \), the labeled \( l \)-sample of \( c \in C \) is given by \( \text{sam}_c(x) = (\langle x_1,J_c(x_1) \rangle,\ldots,\langle x_l,J_c(x_l) \rangle) \) where \( J_c(x) \) equals 1 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 \( c \in C \) and all \( x \in X^l \) for all \( l \geq 1 \).
Let $C$ be a concept class on $X$ and $H$ an algebra of Borel sets on $X$. Then $F_{CH}$ is the set of all functions $f : S_C \rightarrow H \cup \{\bot\}$ where $\bot$ is a special symbol, introduced to account for the possibility that $f$ does not reach a conclusion based on the current information. In the sequel we omit $C$ and $H$ when understood from the context.

We use the following protocol ([V1] and others) between two agents, $T$ (teacher) and $L$ (learner): $T$ (who wants to teach $L$ the concept $c$) repeatedly picks, at random according to some distribution $D$, an element $x$ from a set $X$ and sends to $L$ the pair $<x, I_c(x)>$. $L$, after receiving sufficiently many examples, uses a function $f \in F_{CH}$ to return the set $f(<x_1, I_c(x_1)>, \cdots, <x_l, I_c(x_l)>)$.

Let $Y_1, Y_2 \subseteq X$ we say that $Y_1$ and $Y_2$ are $\varepsilon$-close w.r.t. the distribution $D$ if $Pr_D(Y_1 \oplus Y_2) < \varepsilon$ ($\oplus$ denotes the symmetric difference). Otherwise, $Y_1$ and $Y_2$ are $\varepsilon$-far w.r.t. the distribution $D$.

Following [BEHW1], throughout the paper, we assume that $X$ is a fixed set, usually finite, countable, or $E^r$ (Euclidean $r$ dimensional space) for some $r \geq 1$. In the latter case, we assume that each $c \in C$ and $h \in H$ is a Borel set.

Let $C$ be some given concept class over $X$. For $f \in F_{CH}$, distribution $D$, concept $c \in C$, $\varepsilon > 0$ and positive integer $l$, let $x=(x_1, \cdots, x_l) \in X^l$ be independently and randomly selected by $D$. We define the following probabilities taken over the random selection of $x$: $r^+ f(D, c, l, \varepsilon) = Pr(f(\text{sam}_x(x)) \text{ is a subset of } X \text{ e-close to } c \text{ w.r.t. } D)$ and it is the probability that $f$ finds a "good" approximation for $c$ using an $l$-sample of $c$. Similarly $r^- f(D, c, l, \varepsilon) = Pr(f(\text{sam}_x(x)) \text{ is a subset of } X \text{ e-far from } c \text{ w.r.t. } D)$ and is the probability that $f$ "fails" using an $l$-sample of $c$. Thus $1-r^- f \rightarrow r^+ f$ is the probability that $f$ "realizes" that it is unable to find a concept $\varepsilon$-close to $c$ using an $l$-sample of $c$ and thus returns $\bot$.

Uniform learnability for every distribution: [BEHW1] Let $C$ be a concept class. A function $f \in F$ uniformly learns $C$ for every distribution if for every $\varepsilon, \delta > 0$ there is an $l=l(\varepsilon, \delta) > 0$ such that for every $D$ and every $c \in C$, $r^+ f(D, c, l, \varepsilon) > 1-\delta$. $C$ is uniformly learnable for every distribution if there exists an $f \in F$ that uniformly learns $C$ for every distribution.

The definition above is quite narrow:
Example 1: Let $X = (0,1)$ and for every $n$ let $C_n$ be the set of all unions of $n$ open segments over $X$. $C_n$ is uniformly learnable for every distribution see [BEHW1]. Let $C = \bigcup_{n=1}^\infty C_n$ then $C$ is not uniformly learnable for every distribution [BEHW1] (for further details see section 3).

On the other hand, intuitively, since every $c \in C$ belongs to some $C_i$ it is learnable by 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, the more difficult it is to learn it. Therefore, in the following definition we allow the number of examples to depend on the concept to be learned.

Nonuniform learnability for every distribution: [BEHW1] Let $C$ be a concept class. A function $f \in F$ nonuniformly learns $C$ for every distribution if for every $\epsilon, \delta > 0$ and every $c \in C$ there is an $l = l(\epsilon, \delta) > 0$ such that for every $D$, $r_f^c (D, c, l, \epsilon) > 1 - \delta$. Note that here $l$ depends also on $c$. $C$ is nonuniformly learnable for every distribution if there exists an $f \in F$ that nonuniformly learns $C$ for every distribution.

In section 4 we show that $C$, of example 1, is nonuniformly learnable for every distribution.

Example 2: Let $X = (0,1)$ and $C_\infty$ the set of all open sets over $X$. In section 4 we shall show that $C_\infty$ is not nonuniformly learnable.

3. Previous results - Uniform learnability for every distribution.

In this section we quote the result presented in [BEHW1] (see also [VC]), 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 dimension $d$ ($\dim(C) = d$) 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 dimension. The main result of [BEHW1] is that $C$ is learnable for every distribution if and only if $\dim(C)$ is finite.

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 dimension.
Theorem 1 [BEHW1]: If $\dim(C)=d$ then

1. For every $\epsilon, \delta > 0$ there exists a function that uniformly learns $C$ using
   \[ \max\{\left(\frac{4}{e}\right)\log(2\delta)/(8d/e)\log(8d/e)\} \] examples.

2. For every $0 < \epsilon < \frac{1}{2}$ and $0 < \delta < 1$ there exists no function that can uniformly learn $C$ using at most
   \[ \max\{\left(\frac{1}{2}\epsilon\right)\log(1/\delta), d(1-2(\epsilon(1-\delta)+\delta))\} \] examples.

4. A necessary and sufficient condition for nonuniform learnability.

Theorem 2: $C$ is nonuniformly learnable for every distribution 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 $f \in F$, such that every $c \in C$ is learnable by $f$
using a finite number $i$ of examples. Let $\epsilon = \delta = \frac{1}{4}$ then for $i = 5, 6, 7, \ldots$ let $C_i$ be the set of concepts
learnable by $i$ or less examples. By the second claim of theorem 1 every $C_i$ has finite dimension (at
most $2i$). Since $C$ is nonuniformly learnable every $c \in C$ belongs to some $C_i$.

Let $C = \bigcup_{i=1}^{\infty} C_i$ such that $\dim C_i < \infty$ for every $i = 1, 2, \ldots$ and let $c$ be some concept in $C$ then there
is a $j$ such that $c \in C_j$ and thus for every $\epsilon, \delta > 0$ there is an $l = l(c, \epsilon, \delta)$ such that $c$ is learnable by $l$
examples. □

Example 4: as example 1. We have $\dim C_n = 2n + 1$ and $C = \bigcup_{i=1}^{\infty} C_i$, thus $C$ is nonuniformly learnable.

Claim: The concept class $C_\omega$ of example 2 is not nonuniformly learnable for every distribution.

Proof: First notice that the set $\{1/n : n \in \mathbb{N}\}$ is shattered by $C_\omega$.

Lemma 1: (proof Shai Ben-david) 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: 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 an (infinite) subset \( Y \) such that \( C \cap T \neq Y \) and thus \( C \)'s dimension is not infinite. Let \( A_1 \) be an arbitrary set of \( d_1 + 2 \) elements of \( T \). Since \( \dim C_1 = d_1 \) there is a nonempty set \( B_1 \subseteq A_1 \), such that for every \( c \in C_1, c \cap A_1 \neq B_1 \). For every \( n \) let \( A_n \) be a set of \( d_n + 2 \) elements of \( T - \bigcup_{i=1}^{n-1} B_i \). Since \( \dim C_n = d_n \) there is a nonempty subset of \( A_n, B_n \) such that for every \( c \in C_n, c \cap A_n \neq B_n \). Let \( Y = \bigcup_{i=n}^{\infty} B_i \). It remains to show that \( Y \in C \). This follows since for every \( c \in C \) there exists an \( n \) such that \( c \in C_n \) but \( A_n \cap c \neq A_n \cap Y \) thus \( c \neq Y \). ■

The definition of nonuniform learnability has a considerable practical disadvantage. Even though the function needs only a finite number of examples to learn a concept \( c \) it never "knows" whether it has already learned or not. It might be tempted to search for better and better hypotheses (ask for more examples) even though it already reached one that sufficiently approximates \( c \). In other words, an algorithm that uses such a function to learn a concept has no indication when to stop. This resembles learning in the limit of Inductive Inference [AS]. To overcome this difficulty we present the following:

**Definition:** A concept class \( C \) is **nonuniformly strongly learnable for every distribution** if there is an \( f \) in \( F \) such that for all \( \varepsilon, \delta > 0 \) and every \( c \in C \) there is an \( l > 0 \) such that for every \( D \), \( \sum_{i=1}^{l} r_{i}^{f}(D, c, i, \varepsilon) < \delta \) and \( r^{\ast f}(D, c, l, \varepsilon) > 1 - \delta \).

Notice that here, even though the number of examples needed for learning is not known a priori, a set \( \varepsilon \)-close to \( c \) can be found with high probability by subsequently increasing the number of examples and applying \( f \).

**Theorem 3:** The two definitions are equivalent.

**Proof:** Obviously, if \( C \) is nonuniformly strongly learnable then it is nonuniformly learnable. Prior to the formal proof of the other direction, we give an intuitive sketch. Let \( C \) be a nonuniformly learnable
concept class. Using the functions that learn the \( C_i \)'s we build a function \( f \in F \) that nonuniformly strongly learns \( C \). In the construction of \( f \), an initial segment of examples of a certain length (\( K \)) is used to infer a Turing machine \( T \), the remaining examples serve to verify that the language of \( T \) is \( \epsilon \)-close to the concept to be learned. If the verification fails \( f \) returns \( \perp \). For technical reasons, if the examples cannot be exactly divided into two segments of the appropriate length \( f \) returns \( \perp \).

Formally: Let \( C \) be a nonuniformly learnable concept class then by theorem 2 there are \( C_1, C_2, \ldots \) such that \( C = \bigcup_{i=1}^{\infty} C_i \) and \( d_i = \dim C_i < \infty \) for every \( i \). W.l.g. let \( d_i \) be a nondecreasing series.

(If this does not hold let \( C' = \bigcup_{i=1}^{\infty} C_i \).) Let \( K(\epsilon, \delta, i) = \max \left( \frac{4 \log \frac{2}{\delta} + 8d_i}{\epsilon}, \frac{8d_i}{\epsilon} \right) \), then by \([\text{BEHW1}]\) there exists a function \( f_i \) which uniformly learns every \( c \in C_i \) and for every \( \epsilon, \delta > 0 \) it uses at most \( K(\epsilon, \delta, i) \) examples.

In the following we build a function \( f \in F \) that nonuniformly strongly learns \( C \). Let \( \epsilon, \delta > 0 \) be given and let \( c \in C \) be the concept to be learned. Let \( L(\epsilon, \delta) \) be the minimal number of independent Bernoulli trials, each with probability of success \( \epsilon \), necessary to have at least one success with probability at least \( 1 - \delta \). It is easy to see that \( L(\epsilon, \delta) \leq \frac{\ln(\delta)}{\ln(1 - \epsilon)} \).

For every integer \( i > 0 \) let \( M(\epsilon, \delta, i) = \max \left( K(\epsilon, \delta, i), L(\epsilon, 6\delta/(\pi i)^2) \right) \) where \( \epsilon_i = \frac{-\ln(1 - \delta/4)}{L(\epsilon, 6\delta/(\pi i)^2)} \).

Note that \( M(\epsilon, \delta, i) \) grows unboundedly as \( i \) grows.

Let \( \text{sam}_c(x) \) be a labeled \( l \)-sample of \( c \). If there is no integer \( i \) such that \( l = M(\epsilon, \delta, i) \) then we define \( f(\text{sam}_c(x)) = \perp \). Otherwise, let \( i \) be such an integer and let \( K_i = K(\epsilon_i, \delta/4, i) \) and \( L_i = L(\epsilon, 6\delta/(\pi i)^2) \). If \( f_i(\text{sam}_c(x_1, \ldots, x_K)) \) is consistent with \( \text{sam}_c(x_{K+1}, \ldots, x_l) \) then \( f(\text{sam}_c(x)) = f_i(\text{sam}_c(x_1, \ldots, x_K)) \) otherwise \( f(\text{sam}_c(x)) = \perp \).

Next, we prove that \( f \) nonuniformly strongly learns \( C \). Let \( j \) be the smallest integer such that \( c \in C_j \). We prove first that if \( l = M(\epsilon, \delta, j) \) then \( r_f(D, c, l, \epsilon) > 1 - \delta \) for every \( D \). By definition \( \tilde{c} = f_j(\text{sam}_c(x_1, \ldots, x_K)) \) is \( \epsilon_j \)-close to \( c \) with probability at least \( 1 - \delta/4 \). Note that \( \epsilon_j = \frac{-\ln(1 - \delta/4) \ln(1 - \epsilon)}{2 \ln n_j - \ln 6\delta} < \frac{\delta \epsilon}{4(2 \ln n_j - \ln 6\delta)} < \epsilon \). The probability that \( \text{sam}_c(x_{K+1}, \ldots, x_l) \) is consistent with \( \tilde{c} \)
is greater than \[ \left( 1 + \frac{\ln(1-\delta/4)}{L} \right)^{L} \geq \left( 1 - \frac{2\ln(1-\delta/4)}{L} \right)^{-L} > e^{2\ln(1-\delta/4)} > 1 - \frac{\delta}{2}. \]

It is left to show that \[ \sum_{i=1}^{j} r_{ij}^{-1}(D,c,m) < \delta \] for every \( D \) where \( l = M(e,\delta,j) \). If there is no \( i \) such that \( n = K(e,\delta/2,i) + L(e,\delta/(\pi i)^{2}) \) then \( f \) returns \( \bot \) and \( r_{ij}^{-1}(D,c,m) = 0 \) thus we can rewrite the sum as \[ \sum_{i=1}^{j} r_{ij}^{-1}(D,c,l,i) \] where \( l = K(e,\delta/2,i) + L(e,\delta/(\pi i)^{2}) \). But by construction \( r_{ij}^{-1}(D,c,l,i) < 6\delta/(\pi i)^{2} \) which concludes the proof. \( \blacksquare \)

5. Classes of languages learnable by polynomially many examples.

Theorem 4: Let \( X \) be the set of all finite bit-strings (i.e. \( X = \{0,1\}^{*} \)) and \( C \) be the set of recursive languages. Let \( \varepsilon, \delta > 0, c \in C \) and \( M \) a Turing machine with \( k \) states that recognizes \( c \). Then \( c \) is nonuniformly strongly learnable for every distribution by at most

\[ m(\varepsilon, \delta, k) = \max \left\{ \frac{4d_{k}}{\varepsilon_{k}} \log \frac{8d_{k}}{\varepsilon_{k}}, \frac{8d_{k}}{\varepsilon_{k}} \log \frac{8d_{k}}{\varepsilon_{k}} \right\} + \frac{2\ln nk - \ln 6\delta}{-\ln(1-\varepsilon)} \]

where \( \varepsilon_{k} = \frac{\ln(1-\delta/4)\ln(1-\varepsilon)}{2\ln nk - \ln 6\delta} \) and \( d_{k} \leq 2k \log_{2} 6k \).

Proof: Let \( C_{k} \) be the set of recursive languages recognizable by Turing machines with \( k \) states or less. Clearly \( C = \bigcup_{i \in \mathbb{N}} \cup C_{i} \) and since \( C_{k} \) is finite so is its dimension. Thus \( C \) is nonuniformly learnable for every distribution.

To show that the number of examples needed is polynomial we find the dimension of \( C_{k} \). The number of languages in \( C_{k} \) is bounded by the number of Turing machines with \( k \) states or less. Using the standard model for Turing machines [HU] (the head can move to the left, right or remain in the same place) and \( \Sigma = \{0,1\} \) there are at most \((6k)^{2k}\) Turing machines with \( k \) or less states. (Notice that any Turing machine with less than \( k \) states is equivalent to one with exactly \( k \) states and a transition function that never reaches some of the states.) Thus \( d_{k} = \dim C_{k} \leq 2k \log_{2} 6k \).

By theorem 3, \( m(\varepsilon, \delta, k) = K(e, \delta/4,k) + L(e, 6\delta/(\pi k)^{2}) \) examples are enough to learn \( c \in C_{k} \). \( \blacksquare \)

\(^{3}\) The minimum such \( k \) is related to the Kolmogorov complexity of \( c \).
Remark 1: As defined, learnability does not require computability. In the above proof we use a function \( f \) that, given \( m(\varepsilon, \delta, k) \) examples, returns a Turing machine with \( k \) or less states consistent with the examples, or returns \( \bot \) if no such machine exists. It is an open question whether \( f \) is recursive.

Remark 2: The same proof holds for other concept classes, e.g., the class of r.e. languages.

In [BEHW2] Blumer et al. discuss polynomial learnability of a set of countably many functions with finite range. Here we show a similar but more relaxed condition for polynomial learnability of recursive languages which can be easily extended to recursive functions. First note that for fixed \( \varepsilon, \delta > 0 \) \( m(\varepsilon, \delta, k) = O(k \ln^3 k) \).

Corollary 1: Let \( C \) be a set of recursive languages and \( \alpha > 1 \). For every \( c \in C \) let \( k(c) \) be the smallest number such that there is a Turing machine with \( k(c) \) states that recognizes \( c \). If there is a polynomial algorithm\(^3\) that for every \( c \in C \) given \( m = k^\alpha(c) \) examples produces a Turing machine with \( m/\ln^2(m) \) states consistent with the examples then \( C \) is polynomially learnable.

Notice that the following "naive" algorithm does not define a function that learns the set of recursive languages:

Using some enumeration of the Turing machines such that all Turing machines with \( k \) states come before those with \( k+1 \) states let \( N(l) \) be the number of states in Turing machine number \( l \). Let \( L \) be the recursive language to be learned and let \( T_x \) be the smallest Turing machine that recognizes \( L \).

Procedure A
for \( i=1, 2, \ldots \)
do begin
let \( T_i \) be the \( i^{th} \) Turing machine;
for \( j=m(\varepsilon, \delta, N(i-1))+1 \) to \( m(\varepsilon, \delta, N(i)) \) do ask for examples \( \langle x_j, y_j \rangle \);
Perform one step on every pair \( \langle T_i, x_r \rangle \) where \( 1 \leq r \leq i \) and \( 1 \leq r \leq m(\varepsilon, \delta, N(l)) \).
if there is an \( l \) s.t. \( T_i(x_r) = y_r \) for every \( 1 \leq r \leq m(\varepsilon, \delta, N(l)) \) then stop and output \( T_i \);
end;

For every \( \varepsilon, \delta > 0 \) A outputs, with probability at least \( 1-\delta \), a Turing machine \( T_i \) such that the language of \( T_i \) is \( \varepsilon \)-close to \( L \). However, the number of examples needed depends on the outcome of the selection of the examples. The difficulty to find a computable function which learns the set of recursive languages arises from the fact that there is no bound on the computing time needed to check

\(^3\) This is similar to the Occam algorithm [BEHW2].
whether a certain word belongs to a certain language.

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

1. $C$ is recursive i.e. for every Turing machine $T$, if $T$ is the smallest machine that recognizes a language $L$, then given a coding of $T$ it is decidable whether $L \in C$.

2. $C$ has bounded complexity i.e. there is a function $f$ such that if $T$ is the smallest machine that recognizes a language $L \in C$ then for every word $w$ $T$ stops after at most $f(|w|)$ steps. ($|w|$ is the length of $w$.)

The conditions above can be relaxed by allowing $T$ to be some "polynomially small" Turing machine rather than the smallest. Also note that the above discussion and theorem hold for recursive functions as well.

**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 $|b| \geq v$ 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 $b$ iff $f(b) = \text{true}$. For every $f$ there is such an $M_f$ with $n+O(1)$ states. (Simply write on the input tape the description of $f$ ($n$ states) and then activate a bounded time complexity Turing machine that given a $v$-variable Boolean formula and a Boolean vector of length $v$ evaluates the value of the formula w.r.t. the vector.) Furthermore, the time complexity of an $M_f$, built as above, is bounded. Thus, by condition (2) above, Boolean formulae are learnable by a computable function.

Pitt and Valiant [PV] showed several classes of Boolean formulae such that if $RP \neq P$ then there is no polynomial algorithm that finds a bounded function in that class consistent with a set of examples. In particular: $k$-TERM-DNF, $k$-CLAUSE-CNF (and their monotonic forms) and $\mu$-expressions. Thus even though, the number of examples is polynomial the computation time needed for learning is
6. Nonuniform learnability for a given distribution.

In order to widen the notion of learnability Benedek and Itai [BI] have defined:

Learnability for a given distribution $D$: $C$ is learnable w.r.t. $D$ if there is a function $f \in F$ such that for every $\epsilon, \delta > 0$ there is an $l > 0$ such that for every $c \in C$, $r_f^+(D, c, l, \epsilon) < \delta$.

Finite cover: A subset $C_\epsilon$ of $2^X$ is an $\epsilon$-cover of $C$ w.r.t. $D$ if for every $c \in C$ there is a $c' \in C_\epsilon$ $\epsilon$-close to it. $C$ is finitely coverable (w.r.t. $D$) if for every $\epsilon > 0$ there is a finite $\epsilon$-cover $C_\epsilon$ of $C$. $C$ is countably coverable (w.r.t. $D$) if for every $\epsilon > 0$ there is a countable $\epsilon$-cover $C_\epsilon$ of $C$. In the sequel we omit $D$ when understood from the context.

Theorem 5: $C$ is finitely coverable (w.r.t. $D$) if and only if $C$ is learnable (w.r.t. $D$).

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

Nonuniform learnability for a given distribution $D$: $C$ is nonuniformly learnable w.r.t. distribution $D$ if there is a function $f \in F$ such that for every $\epsilon, \delta > 0$ and every $c \in C$ there is an $l = l(c, \epsilon, \delta) > 0$ such that $r_f^+(D, c, l, \epsilon) < \delta$.

Theorem 6: $C$ is nonuniformly learnable w.r.t. $D$ if and only if $C$ is countably coverable w.r.t. $D$.

Lemma 2: $C$ is finitely or countably coverable w.r.t. iff for every $\epsilon$ there are $C_1^\epsilon, C_2^\epsilon, \cdots$ such that $C = \bigcup_{i=1}^{\infty} C_i^\epsilon$ and $C_i$ has finite $\epsilon$-cover w.r.t. $D$ for every $i = 1, 2, \cdots$.

The proofs of theorem 6 and lemma 2 are similar to the proof of theorem 2.

In the same manner we define nonuniform strong learnability for a fixed distribution:

Nonuniform strong learnability for a given distribution $D$: $C$ is nonuniformly strongly learnable w.r.t. $D$ if there is a function $f \in F$ such that for every $\epsilon, \delta > 0$ and every $c \in C$ there is an $l = l(c, \epsilon, \delta) > 0$ such that $\sum_{i=1}^{l-1} r_f^+(D, c, i, \epsilon) < \delta$ and $r_f^+(D, c, l, \epsilon) > 1 - \delta$. 
Similar to theorem 3 we can prove:

Theorem 7: The two definitions are equivalent.

Example 6: Let $C_n$ be as in example 2 and let $D$ be the uniform distribution over $(0,1)$. It can be shown that for every $\epsilon > 0$ $C$ (of example 1) is an $\epsilon$-cover of $C_n$ w.r.t. $D$ thus $C_n$ is nonuniformly learnable w.r.t. $D$. (The above is true for every continuous distribution.)

Remark 1: In the full version of this paper, the definition of learnability is extended to allow random algorithms and "random functions". All the theorems mentioned hold also for this case.

Remark 2: In [Bl] the authors prove that the same conditions for learnability w.r.t. a given distribution hold even if the examples contain errors. This is also the case for nonuniform learnability w.r.t. given distribution.

7. Conclusion

In this paper we have refined the notion of learnability and have exhibited how concept classes which were not learnable by previous definitions became learnable. We found a general theorem that allows us to decide whether a concept class is learnable.
8. REFERENCES


[PV] Pitt L. and Valiant, L.G., "Computational limitations on learning from examples" Aiken Computation Laboratory, Harvard University, Cambridge, MA 02138, July, 1986


