Learnability and the Vapnik–Chervonenkis Dimension

ANSELM BLUMER

Tufts University, Medford, Massachusetts

ANDRZEJ EHRENFEUCHT

University of Colorado at Boulder, Boulder, Colorado

AND

DAVID HAUSSLER AND MANFRED K. WARMUTH

University of California at Santa Cruz, Santa Cruz, California

Abstract. Valiant's learnability model is extended to learning classes of concepts defined by regions in Euclidean space E^n. The methods in this paper lead to a unified treatment of some of Valiant's results, along with previous results on distribution-free convergence of certain pattern recognition algorithms. It is shown that the essential condition for distribution-free learnability is finiteness of the Vapnik–Chervonenkis dimension, a simple combinatorial parameter of the class of concepts to be learned. Using this parameter, the complexity and closure properties of learnable classes are analyzed, and the necessary and sufficient conditions are provided for feasible learnability.

Categories and Subject Descriptors: F.2.2 **[Analysis of Algorithms and Problem Complexity]**: Nonnumerical Algorithms and Problems—*computations on discrete structures, geometrical problems and computations*; G.3 **[Probability and Statistics]**: *probabilistic algorithms*; I.2.6 **[Artificial Intelligence]**: Learning—*concept learning, induction*; I.5.0 **[Pattern Recognition]**: General

General Terms: Algorithms, Theory, Verification

Additional Key Words and Phrases: Capacity, learnability theory, learning from examples, Occam's razor, PAC learning, sample complexity, Vapnik–Chervonenkis classes, Vapnik–Chervonenkis dimension

---

An extended abstract of this research appeared in *Proceedings of the 18th ACM Symposium on Theory of Computing* (Berkeley, Calif., May 28–30). ACM, New York, 1986, pp. 273–282, and a preliminary version of this paper appeared as BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., AND WARMUTH, M. K. Classifying Learnable Geometric Concepts with the Vapnik–Chervonenkis Dimension. Tech. Rep. UCSC-CRL-86-5. Univ. Calif. at Santa Cruz, Santa Cruz, CA, 1986.

The work of D. Haussler and M. K. Warmuth was supported by the Office of Naval Research grant N00014-86-K-0454. The work of A. Blumer was supported by the National Science Foundation grant IST 83-17918. The work of E. Ehrenfeucht was supported by the National Science Foundation grant MCS 83-05245.

Authors' addresses: A. Blumer, Department of Mathematics and Computer Science, Tufts University, Medford, MA 02155; A. Ehrenfeucht, Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80302; D. Haussler and M. K. Warmuth, Department of Computer and Information Sciences, University of California at Santa Cruz, Santa Cruz, CA 95064.

1. *Introduction*

Valiant has recently introduced a new complexity-based model of learning from examples and illustrated this model by exhibiting and analyzing several learning algorithms for classes of Boolean functions [59, 60]. In this paper we extend Valiant's model to learning concepts defined by regions in Euclidean n-dimensional space E^n, $n \geq 1$. The general techniques we develop lead to new results in Boolean domains as well. Our methods are based on the pioneering work of Vapnik and Chervonenkis [61–63] on the distribution-free convergence of empirical probability estimates and its application to the theory of pattern recognition. These methods provide a unified treatment of some of Valiant's results, and extend previous results of Pearl [50, 51] and Devroye and Wagner ([15], see also [14]), along with our results from [10].

In learning a class C of concepts (e.g., subsets of E^n) from examples, a single *target concept* is selected from C and we are given a finite sequence of points in E^n, each labeled "1" if it is in the target concept (a positive example) and "0" if it is not (a negative example). This set is called a *sample* of the target concept. A *learning function* for C is a function that, given a large enough randomly drawn sample of any target concept in C, returns a region in E^n (a *hypothesis*) that is with high probability a good approximation to the target concept. More precisely:

(1) We let P be a fixed probability distribution on E^n and assume that examples are created by drawing points independently at random according to P.
(2) The *error* of a hypothesis is taken to be the probability that it disagrees with the target concept on a randomly drawn example, that is, the error is just the probability (according to P) of the symmetric difference between the hypothesis and the target concept.
(3) We demand that for a large enough sample size we get a hypothesis with arbitrarily small error with arbitrarily high probability, no matter which concept from C we are trying to learn. The bounds on the sample size must be independent of the underlying distribution P.

This notion of learning is formalized in Section 2. A class of concepts with a learning function that satisfies (3) is called *uniformly learnable*. Condition (3) is formalized by demanding that the hypothesis has error greater than ϵ with probability at most δ for small ϵ and δ, uniformly for all concepts in C. The smallest sample size that achieves this for all distributions and all target concepts in C is called the *sample complexity* of the learning function. This general definition of uniform learnability imposes no restrictions of feasibility or even computability of the learning function; these considerations are postponed until Section 3.

In Section 2 (Theorem 2.1) we give necessary and sufficient conditions on a class of concepts C for the existence of a learning function satisfying (3). This result is based directly on the work of Vapnik and Chervonenkis [61–63]; following [29], we have simplified some of their more general arguments to obtain sharper bounds on the sample complexity of functions satisfying (3) in the special case we consider. We have also added lower bounds for the sample complexity.

Our characterization of learnability uses a simple combinatorial parameter called the *Vapnik–Chervonenkis (VC) dimension* of the class C of concepts [29].[1] We show that there is a learning function satisfying (3) if and only if the VC dimension of C is finite. Moreover, if C does have finite VC dimension d then there is a

[1] This parameter is called the *capacity* of C in [61] (named after a similar notion from [13]) and $S(C)$ in [17].

learning function for C, uniformly achieving error no more than ϵ with probability at least $1 - \delta$, using sample size $m(\epsilon, \delta) = \max(4/\epsilon \log(2/\delta), 8d/\epsilon \log(13/\epsilon))$. In fact, *any* function from samples into the class C that always gives hypotheses consistent with the sample is a learning function for C and has sample complexity bounded by $m(\epsilon, \delta)$. Applications of this result in Euclidean domains are given. We also give an example in a Boolean domain where this result gives significantly better bounds on the sample complexity than the simpler counting arguments used in [10] (Example 2.4).

In Section 3 we introduce considerations of computational feasibility, investigating the implementation of learning functions by computationally efficient *learning algorithms*.

We study two types of learning algorithms. One type works for all domains of Euclidean dimension $n \geq 1$ and for each n, learns concepts in a class $C_n \subseteq 2^{E^n}$, or $C_n \subseteq 2^{\{0,1\}^n}$ in the Boolean case. For example, a pattern recognition algorithm that finds linear separators might be used to learn the class C_n of all half-spaces in E^n for each $n \geq 1$ (see Example 3.1.2).

For the other type of learning algorithm, the domain is fixed, but the "complexity" of the target concept may vary. For example, the domain may be E^2 and the class of target concepts C may be all convex polygons. Since the VC dimension of C is infinite, no algorithm can define a uniform learning function for C in the sense defined above. However, there are efficient learning algorithms for C if we allow the sample size to depend on the number of edges in the target concept, a natural measure of target complexity (see Example 3.2.2).

These two types of learning algorithms lead to two notions of feasible (but nonuniform) learnability for concept classes: polynomial learnability with respect to domain dimension, in which the sample size is allowed to grow polynomially in the Euclidean dimension of the domain, and polynomial learnability with respect to target complexity, in which the sample size is allowed to grow polynomially in the complexity of the target concept. In both cases we insist that the sample size also be polynomially bounded in the inverses of the error and confidence parameters ϵ and δ, and that the learning algorithm run in time polynomial in the sample size. These notions of polynomial learnability, both closely related to the model introduced in [59] and elaborated in [36] and [52], are discussed in Sections 3.1 and 3.2, respectively.

The main result of Section 3.1 gives a characterization of polynomial learnability with respect to domain dimension. We show that the concept classes C_n, $n \geq 1$, are polynomially learnable if and only if the VC dimension of C_n grows polynomially in n and there exists a polynomial time probabilistic algorithm for finding a consistent hypothesis in C_n for any sample of a target concept in C_n (Theorem 3.1.1). As a corollary, we get a result of Natarajan [48] that in the Boolean case, the concept classes $C_n \subseteq 2^{\{0,1\}^n}$, $n \geq 1$, are polynomially learnable if and only if $\log | C_n |$ grows polynomially in n and there exists a polynomial time probabilistic algorithm for finding a consistent hypothesis in C_n for any sample of a target concept in C_n (Corollary 3.1.3). Related results are given in [30]. We give several examples that illustrate these results.

In Section 3.2 we give a sufficient condition for polynomial learnability with respect to target complexity based on the principle of preferring the simpler hypothesis, usually called Occam's Razor (Theorem 3.2.1). Essentially, this result shows that if we can efficiently produce a hypothesis that explains (i.e., is consistent with) the sample data, and is sufficiently more compact than the sample data, then we can feasibly learn. In this sense the result may be interpreted as showing a

relationship between a kind of data compression and learning (see also [55] and [65]).

We use this result to study learning in concept classes that are formed by taking either finite unions or finite intersections of a fixed base class of finite VC dimension. For example, convex polygons are defined by finite intersections of half-planes. We show that classes of this type are polynomially learnable with respect to target complexity whenever there is a polynomial-time algorithm for finding a consistent hypothesis in the base class (Theorem 3.2.4). In obtaining this result, we employ the greedy algorithm for set cover to obtain a sufficiently simple explanation of the sample data. We do not attempt to find the *simplest* explanation, as this is, in general, NP-hard.

Finally, we note that this paper is mostly self-contained in that complete proofs for all the probabilistic and combinatorial lemmas are provided. However, we have relegated many of them to the Appendix. We close with a brief overview of this Appendix.

Section A1 extends the notion of an ϵ-*transversal* for a class of regions $R \subseteq 2^{E^n}$ introduced in [29][2] to arbitrary probability distributions on E^n. For a fixed distribution, an ϵ-transversal for R is a finite set of points $N \subseteq E^n$ such that every region in R of probability at least ϵ contains at least one point in N. Section A2 uses the notion of an ϵ-transversal to provide the primary machinery for Theorem 2.1, following [29] and [62].

In Section A3 we briefly explore the more general problem of learning "stochastic" target concepts, that is, concepts in which the classification of each point in the domain is defined probabilistically rather than deterministically. This is a fairly standard assumption in the pattern recognition literature (e.g., [16]). Here we provide some sufficient conditions for learning such concepts derived directly from results in [61]. These results can also be viewed in terms of a strengthened notion of an ϵ-transversal. Finally, we discuss the relationship between the problem of learning stochastically defined target concepts and some recent extensions of Valiant's learning model that allow misclassifications in the training examples [4, 34, 39, 58, 60].

Notation. $S \Delta T$ denotes the symmetric difference of sets S and T, and $|S|$ the cardinality of S. For $S \subseteq X$, I_S denotes the indicator function for S on X, that is, $I_S(x) = 1$ if $x \in S$, $I_S(x) = 0$, otherwise. X^m denotes the m-fold Cartesian product of X. Elements of X^m will be denoted by barred variables, for example, $\bar{x}$ or $\bar{y}$. We assume $\bar{x}$ denotes $(x_1, \ldots, x_m)$ where $x_i \in X$, $1 \leq i \leq m$, when m is clear from the context, and similarly for other barred variables. If P is a probability distribution on X, then P^m denotes the m-fold product probability distribution on X^m. Natural logarithms are written "ln", all other logarithms are base 2. $\mathbf{Z}$ denotes the integers and $\mathbf{Z}^+$ the positive integers.

2. *Learnability*

We use the following notions of learning functions and learnability.

Definitions. A *concept class* is a nonempty set $C \subseteq 2^X$ of *concepts*. In this paper it is assumed that X is a fixed set, either finite, countably infinite, $[0, 1]^n$, or E^n (Euclidean n-dimensional space) for some $n \geq 1$. In the latter cases, we assume that each $c \in C$ is a Borel set. For $\bar{x} = (x_1, \ldots, x_m) \in X^m$, $m \geq 1$, the *m-sample* of

[2] What we call an ϵ-transversal is called an ϵ-*net* in [29]. Here we have loosely borrowed some terminology from [11, page 65] to avoid confusion with the topological notion of an ϵ-net used in [61] and elsewhere.

$c \in C$ *generated by* $\bar{x}$ is given by $sam_c(\bar{x}) = (\langle x_1, I_c(x_1)\rangle, \ldots, \langle x_m, I_c(x_m)\rangle)$. The *sample space* of C, denoted S_C, is the set of all m-samples over all $c \in C$ and all $\bar{x} \in X^m$, for all $m \geq 1$.

$\mathbf{A}_{C,H}$ denotes the set of all functions $A\colon S_C \to H$, where H is a set of Borel sets on X. H is called the *hypothesis space.* Elements in H are called *hypotheses.* Usually we would like the hypothesis space to be C itself, but in some cases it is computationally advantageous to allow A to approximate concepts in C using hypotheses from a different class H. $A \in \mathbf{A}_{C,H}$ is *consistent* if its hypothesis always agrees with the sample, that is, whenever $h = A(\langle x_1, a_1\rangle \ldots, \langle x_m, a_m\rangle)$ then for all i, $1 \leq i \leq m$, $a_i = I_h(x_i)$. For any $A \in \mathbf{A}_{C,H}$, probability distribution P on X, $c \in C$, and $\bar{x} \in X^m$, the *error* of A for concept c on $\bar{x}$ (with respect to P) is given by $error_{A,c,P}(\bar{x}) = P(c \Delta h)$, where $h = A(sam_c(\bar{x}))$. Thus, A's error is measured as the probability of the region that forms the symmetric difference between the target concept and A's hypothesis, which is just the probability that A's hypothesis will be inconsistent with the target concept on a randomly drawn point (with respect to P). When c and P are clear from the context, we refer to $P(c \Delta h)$ simply as the *error* of h.

Let $m(\epsilon, \delta)$ be an integer-valued function of ϵ and δ for $0 < \epsilon, \delta < 1$ and P be a probability distribution on X. $A \in \mathbf{A}_{C,H}$ is a *learning function for* C (with respect to P) *with sample size* $m(\epsilon, \delta)$ if for all $0 < \epsilon, \delta < 1$ and for all $c \in C$, $P^{m(\epsilon,\delta)}(W) \leq \delta$, where $W = \{\bar{x} \in X^{m(\epsilon,\delta)} : error_{A,c,P}(\bar{x}) > \epsilon\}$. If A is defined in such a way that W is not measurable for some $c \in C$, then we require that there exist a measurable W' such that $W \subseteq W'$ and $P^{m(\epsilon,\delta)}(W') \leq \delta$. Essentially, we insist that using a randomly drawn sample of size $m(\epsilon, \delta)$ of any target concept in C, A produces, with probability at least $1 - \delta$, a hypothesis in H with error no more than ϵ. If such an A exists, we say that C is *uniformly learnable by H under the distribution P.*

Finally, we say $A \in \mathbf{A}_{C,H}$ is a *learning function for C with sample size* $m(\epsilon, \delta)$ only when A is a learning function for C with respect to P with sample size $m(\epsilon, \delta)$ for *all* probability distributions P on X. The smallest such sample size $m(\epsilon, \delta)$ is called the *sample complexity* of A. If such an A exists, we say C is *uniformly learnable* by H. When we say "C is uniformly learnable," we mean that C is uniformly learnable by H for some hypothesis space H.

Example 2.1. Consider the problem of learning concepts such as the concept of "medium build," defined (for men) as having weight between 150 and 185 pounds and height between 5′4″ and 6′0″. By looking at a finite database of randomly chosen men that gives their weight, height, and classification (medium build or not), we want to form a rule that approximates the true concept of medium build, and we want our approximation to be accurate independently of the underlying distribution on height-weight pairs (height and weight values are *not* assumed to be independent). This type of learning problem is formalized and solved as follows:

Let X be E^2 and C be the set of all axis–parallel rectangles, that is, products of intervals on the x-axis with intervals on the y-axis. Let P be any probability distribution on E^2. A simple algorithm to learn a concept $c \in C$ is the following. Keep track of the minimum and maximum x and y coordinates of any positive example. Let l', r', b', t' denote these four values, respectively. After drawing a number of examples, predict that the concept is $h = [l', r'] \times [b', t']$. If no positive examples are drawn, let $h = \varnothing$. Call the learning function defined by this algorithm A.

We claim that A is a learning function for C with sample complexity at most $4/\epsilon \ln(4/\delta)$.

Assume the concept c to be learned is the product of the intervals $[l, r]$ on the x axis and $[b, t]$ on the y axis. Since A's hypothesis h is always contained in c, if $P(c) < \epsilon$, then $error_{A,c,P}$ of any sample of c is always less than ϵ. Otherwise, we define four minimal side rectangles within c that each cover an area of probability at least $\epsilon/4$:

$$left = [l, x] \times [b, t], \quad \text{where } x = \inf\{x : P([l, x] \times [b, t]) \geq \epsilon/4\}$$

and *right*, *bottom*, and *top* are defined similarly. If the sample size is m, the probability that a particular side rectangle contains no example is at most $(1 - \epsilon/4)^m$. The probability that some side rectangle contains no example is bounded by 4 times this quantity. This latter quantity is smaller than $4e^{-m\epsilon/4}$, so if the sample size m is at least $4/\epsilon \ln(4/\delta)$, then with probability at least $1 - \delta$ we draw an example in each of these four side rectangles. If this occurs, then the probability of the region given by the symmetric difference of A's hypothesis and c will be less than ϵ, thus bounding the error of the hypothesis. Since this bound is independent of the particular distribution P, the claim follows.

This example readily generalizes to n-dimensional rectangles for $n > 2$, with a bound of $2n/\epsilon \ln(2n/\delta)$, on the sample complexity.

It is not clear even in two dimensions how to generalize the above example to other types of concepts, for example, circles, half-planes or rectangles of arbitrary orientation. To show that these classes are also uniformly learnable, we use a concept first introduced in [62].

Definition. Given a nonempty concept class $C \subseteq 2^X$ and a set of points $S \subseteq X$, $\Pi_C(S)$ denotes the set of all subsets of S that can be obtained by intersecting S with a concept in C, that is, $\Pi_C(S) = \{S \cap c : c \in C\}$. If $\Pi_C(S) = 2^S$, then we say that S is *shattered* by C. The *Vapnik–Chervonenkis (VC) dimension* of C is the cardinality of the largest finite set of points $S \subseteq X$ that is shattered by C. If arbitrarily large finite sets are shattered, the VC dimension of C is infinite.

Note that the empty set is always shattered, so the VC dimension is well defined. The following additional notation will also be useful.

Definition. For any integer $m \geq 0$, $\Pi_C(m) = \max(|\Pi_C(S)|)$ over all $S \subseteq X$ of cardinality m.

Using this notation, the VC dimension of C can be defined as the largest integer d such that $\Pi_C(d) = 2^d$, or infinity.

Example 2.2. Let X be the real line and let C be the set of all intervals (open or closed) on X. Then given any set S consisting of two points $x_1, x_2 \in X$, we can find concepts $c_1, c_2, c_3, c_4 \in C$ such that $c_1 \cap S = \{x_1\}$, $c_2 \cap S = \{x_2\}$, $c_3 \cap S = \emptyset$, and $c_4 \cap S = S$. Hence, S is shattered by C. However, if S consists of three points $x_1 \leq x_2 \leq x_3$, then there is no concept $c \in C$ that contains x_1 and x_3 but not x_2, and hence S is not shattered. Thus the VC dimension of C is 2.

More generally, let X be the real line and let C be the set of all unions of up to s intervals for some fixed $s \geq 1$. If S consists of points $x_1, \ldots, x_{2s}$, where $x_i \leq x_{i+1}$, $1 \leq i < 2s$, then it is easily verified that S is shattered by C. However, if S consists of points $x_1, \ldots, x_{2s}, x_{2s+1}$, where $x_i \leq x_{i+1}$, $1 \leq i \leq 2s$, then no concept in C contains $x_1, x_3, \ldots, x_{2s+1}$ without also containing a point in $x_2, x_4, \ldots, x_{2s}$. Thus S is not shattered and hence the VC dimension of C is $2s$.

Generalizing in a different manner, let X be E^n for some $n \geq 1$ and C be the set of axis-parallel rectangles on X as in Example 2.1 above. It is easily verified that if

S consists of the $2n$ points at the centers of the faces of the unit n-cube, then S is shattered by C. However, if S consists of any $2n + 1$ points, then consider the smallest closed axis-parallel rectangle R that contains the points of S. Since R has only $2n$ faces, there must be some point $x \in S$ such that either x lies in the interior of R, or x lies on the face of R along with another point of S. Hence, any concept in C that includes all the other points of S must include x. Thus S is not shattered by C and therefore the VC dimension of C is $2n$.

Example 2.3. Let C be any finite concept class. Then since it requires 2^d distinct concepts to shatter a set of d points, no set of cardinality larger than $\log|C|$ can be shattered. Hence, the VC dimension of C is at most $\log|C|$.

Vapnik and Chervonenkis [63], Dudley [17], Wenocur and Dudley [66], Assouad [6], and Haussler and Welzl [29] give numerous additional examples of concept classes of finite VC dimension. For example, the VC dimension of half-spaces and balls of E^n is $n + 1$. In general, whenever C is of finite VC dimension, then C_k, the set of all Boolean combinations formed from at most k concepts in C, is also of finite VC dimension [17]. Thus, for example, since the set C_k of convex k-gons in E^n for fixed $k > n$ is formed by k-fold intersections of half-spaces, C_k is of finite VC dimension for any finite k (see Example 3.2.2). Wenocur and Dudley [66] also prove more general results that imply, for example, that the concept class formed by the set of all half-spaces bounded by polynomial curves of fixed degree also has finite VC dimension (see Example 3.1.3). On the other hand, the set of all finite unions of intervals, like the set of all open sets or all Borel sets, obviously has infinite VC dimension.

Definition. A concept class $C \subseteq 2^X$ is *trivial* if C consists of one concept, or two disjoint concepts c_1 and c_2 such that $c_1 \cup c_2 = X$.

It is clear that a sample size of one is the most that is required to learn C when C is trivial.

We can now state the main result of this section.

THEOREM 2.1. *Let C be a nontrivial, well-behaved*[3] *concept class.*

(i) *C is uniformly learnable if and only if the VC dimension of C is finite.*

(ii) *If the VC dimension of C is d, where $d < \infty$, then*

(a) *for $0 < \epsilon < 1$ and sample size at least*

$$\max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{8d}{\epsilon}\log\frac{13}{\epsilon}\right),$$

any consistent function $A: S_C \to C$ is a learning function for C and

(b)[4] *for $0 < \epsilon < \frac{1}{2}$ and sample size less than*

$$\max\left(\frac{1-\epsilon}{\epsilon}\ln\frac{1}{\delta}, d(1 - 2(\epsilon(1-\delta) + \delta))\right),$$

no function $A: S_C \to H$, for any hypothesis space H, is a learning function for C.

[3] This is a relatively benign measure-theoretic assumption discussed in Section A1 of the Appendix.

[4] This lower bound has recently been improved [20] to $\max((1-\epsilon)/\epsilon \ln 1/\delta, (d-1)/32\epsilon)$ for $\epsilon \le \frac{1}{8}$ and $\delta \le \frac{1}{100}$ which asymptotically meets the upper bound of (a), except for an $O(\log 1/\epsilon)$ factor.

The proof will be given in a series of lemmas and theorems, most of which are given in the Appendix. We first observe that if (ii)(a) holds then the "if" part of (i) follows. This is because we can always produce a consistent function that maps from S_C into C by simply well-ordering the concepts in C and choosing for each sample in S_C the first concept that is consistent with this sample. In this section we are not concerned with the computability of A. Similarly, if (ii)(b) holds, then the "only if" part of (i) follows. This is because the second lower bound of (ii)(b) grows arbitrarily large with d for appropriate choice of ϵ and δ. Hence, we need only establish (ii). The proof of (ii)(a) is given in Section 2 of the Appendix. Here we give only the following proof:

PROOF OF (ii)(b). Let C be a nontrivial concept class of finite VC dimension d, $0 < \epsilon < \frac{1}{2}$, and $m = \max((1 - \epsilon)/\epsilon \ln 1/\delta, d(1 - 2(\epsilon(1 - \delta) + \delta)))$. We must show that any learning function for C must use sample size at least m. We begin with the first term of the lower bound, considering two cases for C.

Case 1. C contains two distinct concepts c_1 and c_2 that have a nonempty intersection.

Let A be a learning function for C and let $a \in c_1 \cap c_2$ and $b \in c_2 - c_1$. Let P be the probability distribution such that $P(b) = \epsilon$, $P(a) = 1 - \epsilon$, and $P(x) = 0$ for any other point $x \in X$. With respect to this distribution, we can effectively replace X with the set $\{a, b\}$, C with $\{\{a\}, \{a, b\}\}$, and H with the four subsets of $\{a, b\}$, modifying A accordingly. (C may contain other subsets of X as well, but this would only make the learning problem harder.)

It is easily verified that if the sample size m is less than $1/(-\ln(1 - \epsilon))\ln(1/\delta)$, then the probability of drawing the point a each time is greater than δ. Since $1/(-\ln(1 - \epsilon)) > (1 - \epsilon)/\epsilon$ for all $0 < \epsilon < 1$, this holds for $m \leq (1 - \epsilon)/\epsilon \ln(1/\delta)$. We can divide the possible learning functions for C into four types, depending on how they respond to an m-sample in which each point is a. Since both concepts in C contain a, all examples of the sample will be positive. If the learning function responds by producing the hypothesis $\{a\}$, then for this sample it has error ϵ if the target concept is $\{a, b\}$. If it responds $\{a, b\}$, $\{b\}$, or $\emptyset$, then, because $\epsilon < \frac{1}{2}$, it has error at least ϵ if the target concept is $\{a\}$. In any case, there is a concept in C such that the probability that A produces a hypothesis of error at least ϵ with sample size m is greater than δ. By decreasing the probability of a slightly and increasing the probability of b by the same amount it follows that the previous statement also holds for error strictly larger than ϵ instead of error at least ϵ. We conclude A cannot be a learning function for C with sample size $m < (1 - \epsilon)/\epsilon \ln(1/\delta)$.

Case 2. C contains at least two disjoint concepts c_1 and c_2 such that $c_1 \cup c_2 \neq X$.

Let $a \in X - (c_1 \cup c_2)$ and $b \in c_1$. Given any learning function A for C, we let the distribution P be defined as above in Case 1 and replace X by $\{a, b\}$, C by $\{\{b\}, \emptyset\}$, H by $2^{\{a,b\}}$, and modify A accordingly. The remainder of the analysis is the same, except that we consider the case when A receives a sequence of negative examples, all of the point a.

This completes the verification of the first term of the lower bound.

For the second term of the lower bound, note that since C is nontrivial, the VC dimension d of C is at least 1. There must exist a set X_d of d points in X that are shattered by C. Let the probability distribution P on X be uniform on these points and 0 everywhere else. With respect to this distribution, we may replace X with X_d and C with 2^{X_d}.

Suppose we draw a sequence $\bar{x} \in X^m$ and l different points are observed in this sample. For each of the 2^l possible labelings of $\bar{x}$, there are 2^{d-l} concepts consistent with this labeling. Whatever the hypothesis of the learning function, for every point of X not observed in $\bar{x}$, it will be correct for exactly half these concepts. Thus, the average error of the learning function on $\bar{x}$ over all concepts in C is at least $(d - l)/(2d) \geq (d - m)/(2d)$. This implies that the average error of the learning function over all $\bar{x} \in X^m$ and all concepts in C is at least $(d - m)/(2d)$. Hence, there must be a concept with average error at least $(d - m)/(2d)$. To make the frequency of m-sequences in which the error on this concept is greater than ϵ at most δ, the error can be greater than ϵ (i.e., 1) on at most $\delta 2^m$ of the m-sequences, and must be at most ϵ on the remainder. Hence, the average error can be at most $\epsilon(1 - \delta) + \delta$, which gives the second part of the lower bound.

Combined with the proof of (ii)(a) given in the Appendix, this completes the proof of Theorem 2.1. □

The above theorem shows a significant gap in the inherent sample complexity of learning problems: depending on whether or not the VC dimension of a concept class C is finite, either C is uniformly learnable with sample size $O(1/\epsilon \log(1/(\epsilon\delta)))$ or C is not uniformly learnable at all. Furthermore, if C is uniformly learnable, then there exists one sample size proportional to the VC dimension of C that any learning algorithm for C must use, and another sample size proportional to the VC dimension of C that is sufficient for any consistent algorithm using hypothesis space C. Hence, the sample complexity of learning C is linear in the VC dimension of C in a strong sense.

It also has immediate consequences for several well-known pattern recognition algorithms. For example, since the VC dimension of half-spaces in E^n is $n + 1$, the sample complexity of the classical perceptron learning algorithm (see, e.g., [16]) is at most $\max(4/\epsilon \log(2/\delta), 8(n + 1)/\epsilon \log(13/\epsilon))$, that is, at most this many examples are required to achieve error at most ϵ with probability at least $1 - \delta$, independent of the underlying distribution governing the selection of examples. (Here we assume that one random sample of an unknown half-space is drawn and the algorithm cycles through this sample until it finds a hyperplane that correctly separates the positive from the negative examples.) Since δ appears only in a log term, this implies that good linear separators can be found with very high probability from relatively small random samples. Furthermore, the same bound applies to *any* learning algorithm for half-spaces that produces consistent hypotheses that are half-spaces, for example, the Ho–Kashyap algorithm or linear programming [16] (see also Example 3.1.2 below).

Since the above bound is $O(1/\epsilon(\log(1/\delta) + n \log(1/\epsilon)))$, for small ϵ it is significantly better than the $O(1/\epsilon^2(\log(1/\delta) + n \log(n/\epsilon)))$ bound given in [51] (eq. 29), derived directly from [15] and [62]. However, the latter bound is more robust in that with this sample size, an accurate estimate of the true error of any half-space that is hypothesized can be obtained from its observed rate of disagreement with the target. This may be necessary in cases where no half-space is consistent with the sample data, for example, because of noise in the data, or because the target concept is not a half-space. In a recent overview paper, Devroye [14] discusses further applications of results of this type, derived from the original work of Vapnik and Chervonenkis, to a variety of other pattern recognition methods. In Appendix A3 we indicate how the bounds given by Vapnik [61] on the relative deviation of empirical estimates from true probabilities provide an alternate analysis in cases where the hypothesis does not fit the training data exactly. This analysis gives

smaller required sample sizes when the observed rate of disagreement between target and hypothesis is small.

Other general techniques for obtaining distribution-free learning results for the case when the hypothesis space is finite, for example, for learning problems in Boolean domains, have been described in [10] (see also [61]). In analogy with Theorem 2.1(ii)(a), we have the following result.

THEOREM 2.2 [10, 61]. *Let $C \subseteq 2^X$ be any finite concept class. Then for sample size greater than $1/\epsilon\, ln(|C|/\delta)$, any consistent function A: $S_C \rightarrow C$ is a learning function for C.*

PROOF. Let P by any probability distribution on X and c be any target concept in C. Any hypothesis $h \in C$ with error greater than ϵ will be inconsistent with an m-sample of c unless all examples in it land in the region outside the symmetric difference of h and c, a region of probability less than $1 - \epsilon$. The probability of this occurring for a particular $h \in C$ is at most $(1 - \epsilon)^m \leq e^{-\epsilon m}$. So the probability of it occurring for any $h \in C$ is at most $|C|e^{-\epsilon m}$. If $m \geq 1/\epsilon \ln(|C|/\delta)$, then this probability is at most δ. Hence, since A produces consistent hypotheses in C, for this sample size the probability that A's hypothesis has error greater than ϵ is less than δ. □

Theorem 2.1(ii)(a) can also be used to obtain bounds of the type given in the above result, since the VC dimension of C is at most $\log |C|$ for finite C. However, in many cases Theorem 2.2 is easier to apply than Theorem 2.1 (you do not need to calculate the VC dimension of C) and yields slightly better bounds on the sample complexity. Its proof is also considerably simpler. On the other hand, in some cases merely counting the number of different possible concepts is too crude a measure to use in estimating the sample complexity of a learning algorithm.

Example 2.4. Consider the case of pattern recognition with linear separators on Boolean domains: Let $X = \{0, 1\}^n$ and C be the class of concepts on X defined by linearly separable Boolean functions on n variables (i.e., intersections of half-spaces with the Boolean n-cube) [25, 42]. In [46], it is shown that $2^{n(n-1)/2} \leq |C| \leq 2^{n^2}$. So by applying Theorem 2.2 we cannot show that C is uniformly learnable with a sample size that is better than $O(1/\epsilon(n^2 + \log(1/\delta)))$. In fact, since half-spaces in E^n have VC dimension $n + 1$, the VC dimension of C is at most $n + 1$ and thus by Theorem 2.1, C is uniformly learnable with sample size $O(1/\epsilon(n \log(1/\epsilon) + \log(1/\delta)))$ by any consistent learning function. For fixed ϵ and δ this is a reduction in the sample complexity bound from $O(n^2)$ to $O(n)$.

3. *Polynomial Learnability and Occam's Razor*

In this section we examine the computational feasibility of learning various concept classes. Part of this involves looking at how learning functions can be implemented as algorithms that take samples and return hypotheses. However, to be useful, a theory of learning *algorithms* must start with a somewhat broader notion than that of a learning function as given in the previous section.

First, learning algorithms are often defined for domains of arbitrary Euclidean dimension. For example, the techniques for finding linear separators discussed in the previous section work for any dimension. It is best to think of such a technique, that is, the technique of finding a hypothesis with linear programming or that of cycling through the data with the perceptron algorithm, as a single learning

algorithm that works for the class of half-spaces in any Euclidean dimension. In this way we can focus on how the sample size required for learning and the computational efficiency of the hypothesis-producing procedure are affected as the dimension of the domain increases. Because the sample size required grows with the domain dimension, this type of learning result is not uniform in the strong sense defined in the previous section, where the sample size was bounded only in terms of the accuracy and confidence parameters ϵ and δ.

Another type of nonuniform learning is also of interest in a computational theory of learnability. As an example, consider again the algorithm for learning axis-parallel rectangles. In Euclidean dimension 1, this reduces to an algorithm for learning intervals on the real line. Without increasing the dimension of the domain, this algorithm can be generalized to learn target concepts that are unions of intervals of the real line (see Example 3.2.1 below). Since the VC dimension of the class C of all finite unions of intervals on the real line is infinite, Theorem 2 shows that C is not uniformly learnable. Nevertheless, the algorithm in Example 3.2.1 is a very practical learning algorithm. Given a relatively small sample of any target concept defined by a union of a small number s of intervals, it produces, with high probability, a hypothesis with small error, independent of the distribution. It does not learn C uniformly because the sample size needed grows with s.

The above example shows that it is not only useful to parameterize learning algorithms and learnability results by the dimension of the domain, but also by some natural measure of the syntactic complexity of the target concept, in this case the number of intervals used to define it. Both of these considerations are emphasized in [36] and [52] in the investigation into the learnability of Boolean functions. We treat both of these issues formally in the next two subsections. The consequences of introducing syntactic complexity into an abstract theory of learnability have recently been explored independently from our work in [8] and [41].

Before proceeding, we note that the efficiency of a learning algorithm in a real-valued domain will depend on which of the standard computational models is adopted. We can choose either the *logarithmic cost* model, in which real numbers are represented in finite precision and operations on them are charged time proportional to the number of bits of precision, or the *uniform cost* model, in which real numbers occupy one unit of space and standard operations of addition, multiplication, etc. take one unit of time (see [1]). We deem it unwise at this point to attempt to dictate which model is correct for the study of computational learnability, so we shall leave this aspect of our basic model unspecified. Our theorems hold in either model, but specific examples are occasionally model-dependent.

3.1 Polynomial Learnability with Respect to Domain Dimension. For the purposes of computation, we must assume some representation for the hypotheses produced by a learning algorithm, and in addition, we may assume some representation for the target concepts. Usually the class of target concepts and hypothesis space are the same and the same representation is used, but this is not always so (see, e.g., [36]).

Definition. For each $n \geq 1$ let X_n be a learning domain, which in this section is either E^n, $[0, 1]^n$ or $\{0, 1\}^n$. For computational purposes, we assume that points in X_n are represented as n-tuples in a standard way. Let $C_n \subseteq 2^{X_n}$ be a class of target concepts on X_n and let $H_n \subseteq 2^{X_n}$ be a hypothesis space. We assume that H_n is well-behaved and that both $\varnothing$ and X_n are members of H_n.

Let there be associated with $\{(X_n, H_n)\}_{n\geq 1}$ a set of representations for concepts in each H_n, $n \geq 1$, given in some representation language. We assume only the following properties of this language.

(1) Each string in the representation language uniquely represents a concept in H_n for some $n \geq 1$, and each such concept has a representation in the language.
(2) The language is in **P**, that is, there is a polynomial-time algorithm to decide if a string is in the language or not.
(3) There is a polynomial-time algorithm that, given a string r in the language and a point $x \in X_n$ for some $n \geq 1$, decides if x is in the concept represented by r or not.

Let there be a similar representation associated with $\{(X_n, C_n)\}_{n\geq 1}$.

By **C** we denote $\{(X_n, C_n)\}_{n\geq 1}$, along with its representation, and by **H** we denote $\{(X_n, H_n)\}_{n\geq 1}$, along with its representation. We use the term *concept class* to refer to **C** and also to refer to individual sets C_n. We say that a concept c is in **C** if $c \in \bigcup C_n$. Similar conventions are adopted for **H**, except that, following established convention, **H** is referred to as the *hypothesis space.*

Refining the learnability definitions from the previous two sections, we now give a definition of polynomial learnability with respect to the dimension n of the domain. Because in this definition a learning algorithm implements a function from samples to hypotheses, we call this the *functional* model of polynomial learnability.

Definition. Let **C** and **H** be defined as above. We say that **C** is *polynomially learnable* (*poly-learnable*) *by* **H** if there exists a polynomial-time algorithm A that takes as input a sample of a concept in **C**, outputs a hypothesis in **H**, and has the property that for all $0 < \epsilon, \delta < 1$ and $n \geq 1$ there exists a sample size $m(\epsilon, \delta, n)$, polynomial in $1/\epsilon$, $1/\delta$, and n, such that for all target concepts $c \in C_n$, and all probability distributions P on X_n, given a random sample of c of size $m(\epsilon, \delta, n)$ drawn independently according to P, A produces, with probability at least $1 - \delta$, a hypothesis $h \in H_n$ that has error at most ϵ (i.e., a hypothesis h such that $P(h \Delta c) \leq \epsilon$). The smallest such $m(\epsilon, \delta, n)$ is called the *sample complexity* of A.

If **C** is poly-learnable by **C**, then we say **C** *is properly poly-learnable.* □

Essentially this definition requires that the algorithm A define for each $n \geq 1$ a learning function for C_n by H_n, and that the sample complexity and computation time for A be polynomial in the appropriate parameters. A similar definition of polynomial learnability is used in [56].

In the above definition the computation time of the learning algorithm is measured as a function of input length. It is also possible to allow the computation time to depend explicitly on the accuracy and confidence parameters ϵ and δ. Since this, and other extensions of the above model, are allowed in the definition of polynomial learnability in [52] and [59], we now introduce a second model of polynomial learnability, which we call the *oracle* model (see also [3] and [36]).

In the oracle model the learning algorithm receives the parameters ϵ, δ, and n as input and has access to an oracle $EX()$ that with each call returns a point in X_n drawn independently at random according to a fixed distribution P, along with a label 0 or 1 indicating whether or not the point is in a fixed target concept $c \in C_n$. Each oracle call takes unit time. After some time A halts and outputs a hypothesis in H_n. In addition to allowing A access to ϵ, δ, n, and an oracle for examples, we

also allow A to be randomized, in the sense that A can in unit time flip a fair coin to decide its next move. An algorithm of this type will be called a (*randomized*) *oracle algorithm.*

Definition. We say that $\mathbf{C}$ is *polynomially learnable* (*poly-learnable*) *by* $\mathbf{H}$ *in the oracle model* if there exists a (possibly randomized) oracle algorithm A that has the property that for all $0 < \epsilon, \delta < 1$ and $n \geq 1$ there exists a time bound[5] $T_A(\epsilon, \delta, n)$, polynomial in $1/\epsilon$, $1/\delta$, and n, such that for all target concepts $c \in C_n$, and all probability distributions P on X_n, A runs in time $T_A(\epsilon, \delta, n)$ and produces, with probability at least $1 - \delta$, a hypothesis $h \in H_n$ that has error at most ϵ.

The functional and oracle models of polynomial learnability are shown to be equivalent in [30], along with another variant of the oracle model in which there are two probability distributions on the domain X_n and two oracles, one for positive examples of the target concept and one for negative examples (e.g., [36] and [52]). However, if randomized oracle algorithms are allowed, the latter proof of equivalence also requires the assumption that H_n includes $\{x\}$ for every $x \in X_n$. Because of this close relationship between the models, we use the term *poly-learnable* to mean polynomially learnable in any one of these models, and similarly for *properly poly-learnable.*

The results of Section 2 show that a number of interesting concept classes are properly poly-learnable. In the following examples we can assume any of the standard representations for the concepts.

Example 3.1.1. Define $\mathbf{C}$ by letting C_n be the class of all axis–parallel rectangles in E^n. Then the algorithm given in Example 2.1, extended appropriately to arbitrary dimension n, has the property that given any sample of size $2n/\epsilon \ln(2n/\delta)$ of a target concept that is an axis–parallel rectangle in E^n, it produces a hypothesis that is an axis–parallel rectangle in E^n and, with probability at least $1 - \delta$ this hypothesis has error at most ϵ, independent of the distribution. Since, in addition, the time required to compute the hypothesis is polynomial in the length of the input (the algorithm merely computes the smallest and largest value of coordinate i among all positive examples for each $1 \leq i \leq n$), this shows that $\mathbf{C}$ is properly poly-learnable. Note that using the fact that the VC dimension of C_n is $2n$ (Example 2.2) and that the algorithm always returns a consistent hypothesis in C_n, this result can also be derived directly using Theorem 2.1(ii)(a).

Example 3.1.2. Define $\mathbf{C}$ by letting C_n be the class of all half-spaces (open or closed) in E^n. We can find a hypothesis in C_n consistent with a sample of a concept in C_n by finding a hyperplane separating the positive from the negative examples. This problem can be reduced to a linear programming problem in $n + 1$ dimensions in which each example forms a constraint. Thus using a polynomial-time algorithm for linear programming, e.g., Karmarkar's algorithm [33], we have a polynomial-time algorithm that always produces a consistent hypothesis in C_n. Since the VC dimension of C_n is $n + 1$, Theorem 2.1(ii)(a) shows that the function defined by any such algorithm is a learning function for C_n by C_n using sample-size polynomial in $1/\epsilon$, $1/\delta$, and n. Hence, $\mathbf{C}$ is properly poly-learnable.

Note that in this case Megiddo's technique for linear programming [44] would not suffice because its time complexity grows exponentially in the dimension of

[5] When the domain is real-valued and the logarithmic cost model is adopted, we also let the time bound depend polynomially on the length of the longest example returned by the oracle.

the linear programming problem (see, e.g., [40]). The perceptron algorithm and related pattern recognition techniques will also take exponential time to produce a separating hyperplane in some cases [25]. On the other hand, the polynomial-time bound for Karmarkar's algorithm holds only in the logarithmic cost model. It is still an open problem to determine if the class of half-spaces is poly-learnable by any hypothesis space in the uniform cost model.

Example 3.1.3. Let k be any positive integer constant and define $\mathbf{C}$ by letting C_n be the set of all half-spaces in E^n defined by surfaces of degree at most k, that is, regions of the form $p(x_1, \ldots, x_n) \geq 0$ or $p(x_1, \ldots, x_n) > 0$ for some k-degree polynomial p. For $k = 1$, C_n is just the class of half-spaces given in Example 3.1.2. For $k = 2$, C_n contains n-dimensional half-spaces, balls, and all other types of half-space regions defined by quadratic surfaces. As in Example 3.1.2, a consistent hypothesis for any sample of a concept in C_n can always be found by linear programming. In this case the dimension of the linear programming problem is $O(n^k)$, which is polynomial in n for fixed k. Furthermore, the results of [66] (also of [13]) imply that the VC dimension of C_n is $O(n^k)$ as well. (This also follows from the fact that surfaces of degree k in n dimensions can be represented as hyperplanes in an $O(n^k)$ dimensional space.) Hence, again using Karmarkar's algorithm, $\mathbf{C}$ is properly poly-learnable.

Example 3.1.4. Let k be any positive integer constant and define $\mathbf{C}$ by letting C_n be the class of concepts on $\{0, 1\}^n$ defined by k-DNF formulas. These are Boolean formulas in disjunctive normal form (i.e., a disjunction of terms, each term a conjunction of literals) in which each term contains at most k literals. Valiant gives a learning algorithm which shows that $\mathbf{C}$ is properly poly-learnable [59, 60]. This result also holds for the dual class defined by k-CNF formulas. However, by a simple counting argument, $|C_n| \leq 2^{((2n)^k)}$. Hence, by Theorem 2.2, *any* algorithm that always produces a consistent k-DNF (k-CNF) hypothesis (and in particular, the algorithm given by Valiant) defines a learning function for C_n by C_n with sample size $O(1/\epsilon(n^k + \log(1/\delta)))$ (the sample size used by Valiant's algorithm). Hence, any such algorithm that runs in polynomial time can be used to show that $\mathbf{C}$ is properly poly-learnable.

All of these examples rely on two basic properties of $\mathbf{C}$. The first is that each class C_n has finite VC dimension and the VC dimension grows only polynomially with n, so that by Theorem 2.1, learning is possible with a reasonably small sample size. In Example 3.1.4 this is guaranteed by the fact that $\log|C_n|$ grows polynomially in n. The second is the existence of an efficient algorithm for producing consistent hypotheses in C_n from samples of target concepts in C_n. Using the techniques of [30], [47], [48], and [52], we can cast these requirements in the form of a characterization of proper polynomial learnability.

Definition. Let $\mathbf{C} = \{(X_n, C_n)\}_{n\geq 1}$, along with some representation. A *randomized polynomial hypothesis finder* (*r-poly hy-fi*) *for* $\mathbf{C}$ is a randomized polynomial time algorithm that takes as input a sample of a concept in $\mathbf{C}$ and for some $\gamma > 0$, with probability at least γ produces a hypothesis in $\mathbf{C}$ that is consistent with this sample. We refer to γ as the *success rate* of the r-poly hy-fi.

THEOREM 3.1.1. *For any concept class* $\mathbf{C} = \{(X_n, C_n)\}_{n\geq 1}$, $\mathbf{C}$ *is properly poly-learnable if and only if there is an r-poly hy-fi for* $\mathbf{C}$ *and the VC dimension of* C_n *grows only polynomially in n.*

PROOF. For the "if" part, assume that the VC dimension of C_n is bounded by $p(n)$ for some polynomial p and that we are given an r-poly hy-fi for $\mathbf{C}$ with success rate γ. Let A be a randomized oracle algorithm defined as follows:

On input (ϵ, δ, n), A calls the oracle EX( ) for

$$\max\left(\frac{4}{\epsilon}\log\frac{4}{\delta}, \frac{8p(n)}{\epsilon}\log\frac{13}{\epsilon}\right)$$

random examples of the target concept $c \in C_n$ drawn according to some distribution P on X_n. Let Q be the resulting sample of c. Then A repeats the following:

(1) simulate the r-poly hy-fi on Q
(2) check if the output of the r-poly hy-fi is consistent with Q until either a consistent hypothesis is found or the number of repetitions exceeds $1/\gamma \ln(2/\delta)$.

In the first case, A returns the hypothesis found. In the second case A returns some default hypothesis in C_n.

It is easily verified that the running time of A is polynomial in $1/\epsilon$, $1/\delta$, and n (and, if we are using the logarithmic cost model, in the maximum size of any example returned by the oracle). Furthermore, A fails to produce a hypothesis of error at most ϵ only if it is forced to return the default hypothesis, or it returns a hypothesis that is consistent with Q but has error greater than ϵ. Since the success rate of the r-poly hy-fi is γ, A is forced to return the default hypothesis with probability at most

$$(1-\gamma)^{1/\gamma \ln 2/\delta} \leq \exp\left(-\ln\frac{2}{\delta}\right) = \frac{\delta}{2}.$$

By Theorem A2.2, the probability that any hypothesis in C_n that is consistent with Q has error greater than ϵ is at most $\delta/2$. Thus A returns a hypothesis that is consistent with Q but has error greater than ϵ with probability at most $\delta/2$. Hence, A produces, with probability at least $1-\delta$, a hypothesis that has error at most ϵ and therefore $\mathbf{C}$ is properly poly-learnable.

For the "only if" part, note first that by Theorem 2.1(ii)(b), any learning algorithm for $\mathbf{C}$ must use a sample size that grows linearly in the VC dimension of C_n, and hence if the VC dimension of C_n is not polynomial in n, then $\mathbf{C}$ is not poly-learnable by any hypothesis space $\mathbf{H}$. To show that $\mathbf{C}$ being poly-learnable implies that there is an r-poly hy-fi for $\mathbf{C}$, we use a construction from [52].

Let A be a learning algorithm for $\mathbf{C}$ in the functional model and let $m(\epsilon, \delta, n)$ be a polynomial in $1/\epsilon$, $1/\delta$, and n such that given $m(\epsilon, \delta, n)$ random examples, A produces a hypothesis that has error at most ϵ with probability at least $1-\delta$ for any $c \in C_n$ and any P on X_n. Using A, we define an r-poly hy-fi B for $\mathbf{C}$ with success rate $\frac{1}{2}$ as follows:

Suppose that B is given a nonempty sample Q of some concept $c \in C_n$, for some $n \geq 1$. Let P be the distribution on X_n that is uniform on all the points of X_n that appear in examples in Q, and 0 elsewhere. Let m be the number of examples in Q, $\epsilon = 1/(m+1)$, and $\delta = \frac{1}{2}$. B determines n and then produces a sample Q' of size $m(\epsilon, \delta, n)$ by drawing points from X_n independently according to P and labeling them with the same labels they had in Q. B then simulates the learning algorithm A on Q' and returns the output of A.

It is easily verified that B is a polynomial-time algorithm. Since $\delta = \frac{1}{2}$ and B produces a sample of the target concept c of size $m(\epsilon, \delta, n)$ independently drawn according to the distribution P, by our assumptions on the learning algorithm A, B's simulation of A produces a hypothesis that has error at most ϵ with respect to c and P with probability at least $\frac{1}{2}$. Since every point of X_n that appears in Q has probability at least $1/m$ according to P, any hypothesis that is not consistent with Q has error greater than ϵ. Hence, B produces a hypothesis that is consistent with Q with probability at least $\frac{1}{2}$. Therefore B is an r-poly hy-fi for $\mathbf{C}$ with success rate at least $\frac{1}{2}$. □

As a corollary of Theorem 3.1.1, we also obtain a useful characterization of proper polynomial learnability in the Boolean case.

LEMMA 3.1.2 [48]. *Assume that $X_n = \{0, 1\}^n$ and $C_n \subseteq 2^{X_n}$ for each $n \geq 1$. Then the VC dimension of C_n grows polynomially in n if and only if $\log | C_n |$ grows polynomially in n.*

PROOF. For all $n \geq 1$, let $q(n)$ be the VC dimension of C_n. Since $C_n = \Pi_{C_n}(X_n)$, Proposition A2.1 shows that $| C_n | \leq (2^n)^{q(n)} + 1 = 2^{nq(n)} + 1$. Thus when $q(n)$ grows polynomially, $\log | C_n |$ grows polynomially. On the other hand, if $| C_n | \leq 2^{p(n)}$, then no subset of X_n of cardinality larger than $p(n)$ can be shattered by C_n, and hence $q(n) \leq p(n)$. Thus when $\log | C_n |$ grows polynomially, $q(n)$ grows polynomially. □

COROLLARY 3.1.3. *For any concept class $\mathbf{C} = \{(\{0, 1\}^n, C_n)\}_{n \geq 1}$, $\mathbf{C}$ is properly poly-learnable if and only if there is an r-poly hy-fi for $\mathbf{C}$ and $\log | C_n |$ grows polynomially in n.*

PROOF. Follows directly from Theorem 3.1.1 and Lemma 3.1.2. □

This characterization is useful both for showing classes to be properly poly-learnable, as demonstrated in Examples 3.1.1 to 3.1.4 above, and for showing that they are not properly poly-learnable unless **RP** = **NP**, as demonstrated in [52]. Here **RP** is the class of languages accepted by randomized polynomial-time algorithms (see, e.g., [22]) and **NP** is the class of languages accepted by nondeterministic polynomial-time algorithms. The negative results are obtained by reducing a known NP-complete problem to the problem of finding a hypothesis in $\mathbf{C}$ that is consistent with a given sample, or to the following decision problem:

Definition. For any concept class $\mathbf{C}$, the *consistency problem for* $\mathbf{C}$ is the problem of determining if there is a concept in $\mathbf{C}$ that is consistent with a given sample.

Clearly, if there is an r-poly hy-fi for $\mathbf{C}$, then the consistency problem for $\mathbf{C}$ is solvable by a randomized polynomial-time algorithm. (Here we use the assumption that the representation language for $\mathbf{C}$ is in **P** and there is a polynomial-time algorithm to check for any given c in $\mathbf{C}$ and point x, if $x \in c$.) Hence if **RP** $\neq$ **NP**, then when the consistency problem for $\mathbf{C}$ is NP-hard, Theorem 3.1.1 shows that $\mathbf{C}$ is not properly poly-learnable.

Results in [52] show that in the Boolean domain, when $\mathbf{C}$ consists of concepts represented by either DNF expressions with at most k terms or CNF expressions with at most k clauses for some fixed $k \geq 2$ (called k-term DNF and k-clause CNF concepts, resp.), or by Boolean threshold functions (i.e., all concepts of the form $\{x \in \{0, 1\}^n : a \cdot x \geq y\}$, for some $a \in \{0, 1\}^n$ and integer $y \geq 0$, where $\cdot$ denotes the inner product), then the consistency problem for $\mathbf{C}$ is NP-complete, and hence

it is unlikely that **C** is properly poly-learnable. However, they also show that k-term DNF concepts can also be represented by k-CNF expressions, and k-clause CNF concepts can be represented by k-DNF expressions. Hence, by the result given in Example 3.1.4 above, in either of these cases **C** is poly-learnable by a larger hypothesis space **H** (as was first shown in [52]). Since the class of concepts represented by Boolean threshold functions is contained in the class of linearly separable Boolean concepts, by Example 3.1.2, this is true for Boolean threshold concepts as well.

Results in [45] show that when C_n is the set of all concepts on E^n defined by the union of two half-spaces, the consistency problem for **C** is NP-complete. This result holds in both the uniform and logarithmic cost models. Hence it is unlikely that **C** is properly poly-learnable.[6]

To the best of our knowledge, it is unknown if this class is poly-learnable by any hypothesis space H.

Haussler et al. [30] give an analysis of the more general case when **C** is poly-learnable by **H** for distinct **C** and **H**, with an appropriately generalized definition of hypothesis finder.

3.2 Polynomial Learnability with Respect to Concept Complexity and Occam's Razor. We now turn to learnability results of the second type mentioned above in the introduction to this section. Here the learning domain is fixed to a single space X, but the class $C \subseteq 2^X$ of concepts is graded according to some concept complexity measure. For simplicity, we consider only the case when concepts in C are learned by hypotheses in C, although the definitions are easily extended to allow a distinct hypothesis space H.

Definition. Let X be a learning domain that for this section is either a finite set, a countably infinite set, or equal to E^n, for some fixed n. Let $C \subseteq 2^X$ be a well-behaved class of concepts on X and let **size** be a function from C into $\mathbf{Z}^+$. The function **size** will be called a *concept complexity measure.* Let there also be associated with C a set of representations for concepts in C given in some representation language. As in the definition at the beginning of the previous section, we assume that there is a function from the set of representations in the language onto C, that the language is in **P**, and that given $x \in X$ and a string in the language, we can decide in polynomial time if x is in the concept represented by the string.

By **C** we denote (X, C), along with the function **size** and the representation for C. We use the term *concept class* to refer to **C** as well as to C. We say a concept c is in **C** if $c \in C$.

In analogy with the definition of polynomial learnability in the functional model defined in the previous section, we make the following definition:

Definition. Let **C** be defined as above. We say that **C** is (*properly*) *polynomially learnable* (*poly-learnable*) if there exists a polynomial-time algorithm A that takes as input a sample of a concept in **C**, outputs a hypothesis in **C**, and has the property that for all $0 < \epsilon, \delta < 1$ and $s \geq 1$ there exists a sample size $m(\epsilon, \delta, s)$, polynomial in $1/\epsilon$, $1/\delta$, and s, such that for all target concepts $c \in C$ with $\mathbf{size}(c) \leq s$, and all probability distributions P on X, given a random sample of c of size $m(\epsilon, \delta, s)$ drawn independently according to P, A produces, with probability at least $1 - \delta$, a

[6] A. Blum and R. Rivest have recently sharpened this result to the Boolean domain [9].

hypothesis $h \in C$ that has error at most ϵ. The smallest such $m(\epsilon, \delta, s)$ is called the *sample complexity* of A.

Since we only deal with the case of **C** poly-learnable by **C**, we drop the adjective "properly" in this section. As in the previous section, there is also an oracle model of polynomial learnability in this case. In fact, the notion of polynomial learnability can be defined for situations when the size of the domain grows with a parameter n as in the previous section, and for each C_n there is also a concept complexity measure (see [30] and [36]).

Below we give a few examples to illustrate the notion of polynomial learnability with respect to concept complexity. But first we introduce some useful definitions.

Definition. Let $C \subseteq 2^{\mathcal{X}}$ be a concept class. By $\mathbf{U}(C)$ we denote the closure of C under finite unions, that is,

$$\mathbf{U}(C) = \{c_1 \cup \cdots \cup c_s : s \geq 1 \text{ and } c_i \in C,\ 1 \leq i \leq s\}.$$

Similarly, $\mathbf{I}(C)$ denotes the closure of C under finite intersections.

Let us assume that there is a standard representation associated with concepts in C. This induces a *standard representation* for $\mathbf{U}(C)$ in which the concept $c = c_1 \cup \cdots \cup c_s$, where $c_i \in C$, $1 \leq i \leq s$, is given as a concatenation of the representations of the c_i's. The *standard concept complexity measure* for $\mathbf{U}(C)$ is the function $\mathbf{size}: \mathbf{U}(C) \to \mathbf{Z}^+$ defined by letting $\mathbf{size}(c)$ be the smallest s such that $c = c_1 \cup \cdots \cup c_s$, where $c_i \in C$, $1 \leq i \leq s$. When we say "let $\mathbf{C} = (X, \mathbf{U}(C))$", where C has some standard representation, we assume the standard representation and concept complexity measure for $\mathbf{U}(C)$. The class $\mathbf{I}(C)$ is treated similarly.

Example 3.2.1. Let $\mathbf{C} = (X, \mathbf{U}(C))$, where X is the real line and C is the set of intervals on X. Consider the following learning algorithm A for **C**.

> Given a sequence of examples of a target concept c in **C**, sort them in increasing order according to the value of their points. Partition this ordering into alternating segments of positive and negative examples, that is, a run of consecutive negative examples, followed by a run of consecutive positive examples, etc. For each segment of positive examples form the closed interval with endpoints consisting of the smallest and largest points in the segment. Return the hypothesis h that is the union of these intervals.

It is easily verified that algorithm A always returns a hypothesis h that is consistent with the examples of c and consists of a union of the fewest possible number of intervals. Hence, $\mathbf{size}(h) \leq \mathbf{size}(c)$. In Example 2.2 above, we calculated the VC dimension of the class C_s of all concepts c in **C** such that $\mathbf{size}(c) \leq s$ to be $2s$. Hence, by Theorem A2.2, for any distribution P on X, given

$$\max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{16s}{\epsilon}\log\frac{13}{\epsilon}\right)$$

independent random examples of c drawn according to P, with probability at least $1 - \delta$, every hypothesis in C_s that is consistent with all of these examples has error at most ϵ. Since A returns a consistent hypothesis in C_s and runs in polynomial time, it follows that **C** is poly-learnable.

Example 3.2.2. Let $\mathbf{C} = (X, \mathbf{I}(C))$, where $X = E^2$ and C is the set of half-planes. In this case the concept class $\mathbf{I}(C)$ is the set of (possibly unbounded) convex polygons. In [19] an algorithm is developed that, given any m-sample of a

convex polygon, will construct in time $O(m \log m)$ a convex polygon that is consistent with the sample and is formed by intersecting the minimal number of half-planes, that is, a consistent hypothesis h such that **size**(h) is minimal.

As above, let C_s be the class of all concepts c in **C** such that **size**(c) $\leq s$, that is, polygons formed by intersecting at most s half-spaces. It is easily verified that the VC dimension of C_s is $2s + 1$: First check that $2s + 1$ points evenly spaced on the unit circle can be shattered by C_s.

Then note that for any set S of $2s + 2$ points,

(1) either one point lies in the convex hull of the other points, in which case, since concepts in C_s are convex, no concept in C_s contains the other points without containing this point, or
(2) the points of S form the vertices of a convex polygon with $2s + 2$ edges, in which case the subset of S consisting of every other point in the clockwise ordering of these vertices cannot be obtained by intersecting S with the intersection of less than $s + 1$ half-planes.

Hence, $2s + 2$ points cannot be shattered and thus the VC dimension of C_s is $2s + 1$.

Again by Theorem A2.2, this implies that for any distribution P on X and any $c \in C_s$, given

$$\max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{16s + 8}{\epsilon}\log\frac{13}{\epsilon}\right)$$

independent random examples of c drawn according to P, the algorithm of [19] produces a hypothesis (in C_s) that, with probability at least $1 - \delta$, has error at most ϵ. Thus **C** is poly-learnable.

Note that in both of the above examples, the VC dimension of the entire concept class (i.e., all unions of intervals or all convex polygons) is infinite. Hence, it does not suffice to merely find consistent hypotheses. In these cases we provided an algorithm with the stronger property that it finds consistent hypotheses with minimal complexity, and then used the fact that the VC dimension of the class C_s of hypotheses of complexity at most s grows only polynomially in s. This is a concrete case where it is provably sufficient to employ the principle of always preferring the simplest hypothesis that explains the data, usually called *Occam's Razor.*

However, there are simple examples where this strategy does not work because of the computational difficulty of producing consistent hypotheses of minimal complexity. Let $\mathbf{C} = (X, \mathbf{U}(C))$, where $X = E^2$ and C is the set of all axis-parallel rectangles. Given a set of points in E^2 labeled with 0's and 1's, it is NP-hard to determine the smallest s such that the set of 1-labeled points can be covered by s axis-parallel rectangles, where none of these rectangles contains a 0-labeled point [43]. Any learning algorithm for **C** that always produces hypotheses of minimal complexity could be used to solve this problem. Hence, finding a hypothesis of minimal complexity is NP-hard in this case.

To address the cases when it is not feasible to find the simplest hypotheses, we show that it suffices to settle for simpl*er* rather than simpl*est* hypotheses, that is, it suffices to produce hypotheses that are significantly simpler than the sample data itself.

Definition. Let $\mathbf{C} = (X, C)$ be a concept class with concept complexity measure **size**. Let A be a polynomial-time algorithm that, given a sample of a concept in **C**,

produces a consistent hypothesis in $\mathbf{C}$. For every s, $m \geq 1$, let $S_{\mathbf{C},s,m}$ denote the set of all m-samples of concepts $c \in C$ such that $\mathbf{size}(c) \leq s$. Let $\mathbf{C}^A_{s,m} \subseteq C$ denote the A-image of $S_{\mathbf{C},s,m}$, that is, the set of all hypotheses produced by A when A is given as input an m-sample of a concept $c \in C$ with $\mathbf{size}(c) \leq s$. We call $\mathbf{C}^A_{s,m}$ the *effective hypothesis space of A for target complexity s and sample size m*. We say A is an *Occam algorithm* for $\mathbf{C}$ if there exist a polynomial $p(s)$ and a constant α, $0 \leq \alpha < 1$, such that for all s, $m \geq 1$, the VC dimension of $\mathbf{C}^A_{s,m}$ is at most $p(s)m^\alpha$.

In this version of Occam's Razor, the VC dimension of the effective hypothesis space $\mathbf{C}^A_{s,m}$ measures how well the Razor is applied by a learning algorithm A. By allowing the hypothesis space of an Occam algorithm to grow with both target complexity and sample size, we make the search for a consistent hypothesis easier. On the other hand, we show that by restricting the VC dimension of this hypothesis space as above we guarantee polynomial learnability.

THEOREM 3.2.1. *Let* $\mathbf{C}$ *be a concept class with a given concept complexity measure.*

(i) *If there is an Occam algorithm for* $\mathbf{C}$, *then* $\mathbf{C}$ *is poly-learnable.*

(ii) *Let* A *be an Occam algorithm for* $\mathbf{C}$ *with effective hypothesis space* $\mathbf{C}^A_{s,m}$ *for target complexity s and sample size m. Then*

(a) *if the VC dimension of* $\mathbf{C}^A_{s,m}$ *is at most* $p(s)m^\alpha$ *for some polynomial* $p(s) \geq 1$ *and* $0 \leq \alpha < 1$, *then* A *is a polynomial-time learning algorithm for* $\mathbf{C}$ *using sample size*

$$m = max\left(\frac{4}{\epsilon}log\frac{2}{\delta}, \left(\frac{8p(s)}{\epsilon}log\frac{13}{\epsilon}\right)^{1/(1-\alpha)}\right)$$

(b) *if the VC dimension of* $\mathbf{C}^A_{s,m}$ *is at most* $p(s)(\log m)^l$ *for some polynomial* $p(s) \geq 2$ *and* $l \geq 1$, *then the same result holds with the second term of the bound replaced by*

$$\frac{2^{l+4}p(s)}{\epsilon}\left(log\frac{8(2l+2)^{l+1}p(s)}{\epsilon}\right)^{l+1}$$

PROOF. Since part (i) follows from part (ii)(a), it suffices to prove part (ii). The result follows if we can show that

$$2\Pi_{\mathbf{C}^A_{s,m}}(2m)2^{-\epsilon m/2} \leq \delta, \qquad (*)$$

since by Theorem A2.1, for any target concept and distribution, the left side of this inequality is a bound on the probability that there is any hypothesis in $\mathbf{C}^A_{s,m}$ of error greater than ϵ that is consistent with a random m-sample of this target. For part (a), using Proposition A2.1 (iii) and the fact that $p(s) \geq 1$, to prove $(*)$ it suffices to show that $2(2em/(p(s)m^\alpha))^{p(s)m^\alpha}2^{-\epsilon m/2} \leq \delta$. For part (b), using $p(s) \geq 2$ and Proposition A2.1(ii), to prove $(*)$ it suffices to show that $2(2m)^{p(s)(\log m)^l}2^{-\epsilon m/2} \leq \delta$. These calculations are given in Lemma A2.5 in Appendix A2.

In some ways Theorem 3.2.1 can be viewed as showing a relationship between learning and data compression.

Example 3.2.3. Let $\mathbf{C} = (X, C)$, where X is some countable domain, $C \subseteq 2^X$, and for all $c \in C$, $\mathbf{size}(c)$ is the number of bits needed to represent the concept c in some fixed representation language. For example, X could be the set of words

over a finite alphabet and C the class of languages represented by regular expressions, or X could be $\{0, 1\}^n$ for some large n and $C = 2^X$ might be all Boolean concepts on X represented by DNF expressions, decision trees, etc.

Let A be a polynomial-time algorithm that, given any m-sample of a concept $c \in C$ that can be described in s bits, produces a hypothesis in C that "explains" (i.e., is consistent with) the sample and is described in at most $p(s)m^\alpha$ bits, for some polynomial $p(s)$ and $0 \leq \alpha < 1$. For fixed s, this amounts to a kind of data compression on the sample.

Let $\mathbf{C}^A_{s,m}$ be the effective hypothesis space of A for target complexity s and sample size m. Since $|\mathbf{C}^A_{s,m}| \leq 2^{p(s)m^\alpha}$, the VC dimension of $\mathbf{C}^A_{s,m}$ is as most $p(s)m^\alpha$, and hence A is an Occam algorithm for $\mathbf{C}$. Thus Theorem 3.2.1 shows that A is a learning algorithm for $\mathbf{C}$ with reasonably small sample complexity when α is not close to 1.

As demonstrated in the above example, Theorem 3.2.1 shows that efficient data compression via hypothesis generation is sufficient for learning. Numerical bounds on sample complexity of Occam algorithms like the ones in that example that are slightly better in some cases than those given in Theorem 3.2.1 are derived in [10], using a simpler argument, akin to that given in Theorem 2.2 above.

We now use Theorem 3.2.1 to demonstrate the learnability of many concept classes of the form $(X, \mathbf{U}(C))$ and $(X, \mathbf{I}(C))$ for C of finite VC dimension, including the case when C is the class of axis-parallel rectangles discussed above.

Definition. Let $\mathbf{C} = (X, C)$ be a concept class along with some representation as described above. A *polynomial hypothesis finder* for $\mathbf{C}$ is a polynomial algorithm that, given a sample of a target concept in $\mathbf{C}$, returns a hypothesis in $\mathbf{C}$ that is consistent with the sample. Note that, in contrast to the previous section, we do not consider randomized hypothesis finders here. The *consistency problem for* $\mathbf{C}$ (or C) is the problem of determining if there is a concept in C that is consistent with a given sample over X.

As in the previous section, given our assumptions on the representation for $\mathbf{C}$, the existence of a polynomial hypothesis finder for $\mathbf{C}$ implies that the consistency problem for C is in $\mathbf{P}$.

LEMMA 3.2.2. *If C has finite VC dimension and the consistency problem for C is in $\mathbf{P}$, then for any finite set $S \subseteq X$, the sets of $\Pi_C(S)$ can be listed in time polynomial in the cardinality of S.*

PROOF. Assume $S = \{x_1, x_2, \ldots, x_m\}$. The size of $\Pi_C(S)$ is polynomial in m by Proposition A2.1. To produce a list L of $\Pi_C(S)$ we proceed as follows. Initialize L to the one element list consisting of just the empty set. This corresponds to the case $m = 0$. Now by induction, assume that the list $L = \Pi_C(\{x_1, x_2, \ldots, x_i\})$ has been produced for some i, $0 \leq i < m$. L is updated to the list $\Pi_C(\{x_1, x_2, \ldots, x_{i+1}\})$ as follows. For each element T of L, test the sets T and $T \cup \{x_{i+1}\}$ for membership in $\Pi_C(\{x_1, x_2, \ldots, x_{i+1}\})$. Since the consistency problem for C is in $\mathbf{P}$, this can be done in polynomial time by creating the appropriate samples and checking if there is a concept in C that is consistent with them. Now replace the element T in L with either one or both of these sets, according to the outcome of this test. (Note that it is possible that $T \in \Pi_C(\{x_1, x_2, \ldots, x_i\})$ but $T \notin \Pi_C(\{x_1, x_2, \ldots, x_{i+1}\})$.) The time for each complete update of the list L is polynomial since by Proposition A2.1 the size of L remains polynomial in m. Hence, the entire procedure is polynomial time. □

LEMMA 3.2.3. *Let $C \subseteq 2^X$ be a concept class of finite VC dimension $d \geq 1$. For all $s \geq 1$ let $C_s = \{\bigcup_{i=1}^{s} c_i : c_i \in C,\ 1 \leq i \leq s\}$ (resp., $C_s = \{\bigcap_{i=1}^{s} c_i : c_i \in C,\ 1 \leq i \leq s\}$). Then for all $s \geq 1$, the VC dimension of C_s is less than $2ds \log(3s)$.*

PROOF. The proof is analogous to that of Lemma 4.5 of [29], which gives a slightly weaker bound. We consider only the case for unions. The case of intersections is treated similarly.

Clearly we may assume $s \geq 2$. Consider a finite set $S \subseteq X$ with $|S| = m \geq 1$. By Proposition A2.1(i), $|\Pi_C(S)| \leq \Phi_d(m)$. Every set in $\Pi_{C_s}(S)$ is of the form $\bigcup_{i=1}^{s} S_i$, with $S_i \in \Pi_C(S)$, $1 \leq i \leq s$. This shows that

$$|\Pi_{C_s}(S)| \leq (|\Pi_C(S)|)^s \leq (\Phi_d(m))^s.$$

If $(\Phi_d(m))^s < 2^m$, then S cannot be shattered by C_s and the VC dimension of C_s is less than m. Thus by Proposition A2.1(iii) it suffices to prove that $(em/d)^{ds} < 2^m$ for $m = 2ds \log(3s)$, which is equivalent to $\log(3s) < 9s/(2e)$. If the last inequality holds for some value of s, then it holds for all larger values as well. It is easy to verify it for $s = 2$. □

THEOREM 3.2.4. *Let $\mathbf{C} = (X, C)$ be a concept class and associated representation such that there exists a polynomial hypothesis finder for $\mathbf{C}$ and C has finite VC dimension. Then $\mathbf{C}' = (X, \mathbf{U}(C))$ (resp., $\mathbf{C}'' = (X, \mathbf{I}(C))$) is polynomially learnable.*

PROOF. We consider only the case $\mathbf{C}' = (X, \mathbf{U}(C))$, the other case being similar. Let S be the set of points in an m-sample of a target concept c in $\mathbf{C}'$. Our strategy will be to find a hypothesis consistent with S that is formed from the union of relatively few concepts in C, that is, not many more than **size**(c). This problem can be formulated as a set cover problem. The set to be covered is the set of positive points of S and the sets allowed in the cover are the elements of $\Pi_C(S)$ that contain only positive points. To find the smallest set cover is NP-hard [21] and remains NP-hard for simple geometric versions such as covering with rectangles [43]. Fortunately, there is a simple greedy algorithm [32, 49] that produces a cover using at most $s \ln(p) + 1$ sets, where s is the minimum number of sets needed for any cover and p is the size of the set to be covered: pick the set that covers the largest number of points; after this, pick the set that covers the largest number of points that have not been covered previously, and so forth.

Since the sets of $\Pi_C(S)$ can be listed in polynomial time (Lemma 3.2.2), the largest set that contains only positive points can be found in polynomial time. Given this set, by labeling the other points negative we can use the polynomial hypothesis finder for $\mathbf{C}$ to produce a hypothesis in $\mathbf{C}$ that includes only these points of the sample. By deleting these points and then iterating this procedure, we obtain a greedy cover for the positive examples in S expressed as the union of concepts in $\mathbf{C}$, which we use as a hypothesis. Call this algorithm A.

We have shown that A is polynomial time and that given any m-sample of a concept c in $\mathbf{C}'$ with **size**$(c) \leq s$, A produces a consistent hypothesis h for this sample with **size**$(h) \leq s \ln(m) + 1$. Hence, the effective hypothesis space $\mathbf{C}'^{A}_{s,m}$ of A for target complexity s and sample size m contains only hypotheses such that **size**$(h) \leq s \ln(m) + 1$. By Lemma 3.2.3, the VC dimension of $\mathbf{C}'^{A}_{s,m}$ is $O(s \log(m)(\log s + \log\log m))$. Hence, A is an Occam algorithm for $\mathbf{C}'$ and thus by Theorem 3.2.1, $\mathbf{C}'$ is polynomially learnable. □

Example 3.2.4. Let $\mathbf{C}' = (X, \mathbf{U}(C))$, where $X = E^n$ for some fixed n and C is the set of axis–parallel rectangles on E^n. Then $\mathbf{C}'$ is poly-learnable by Theorem 3.2.4 (see Examples 2.1 and 2.2).

Example 3.2.5. Let $\mathbf{C}' = (X, \mathbf{U}(C))$ and $\mathbf{C}'' = (X, \mathbf{I}(C))$, where $X = E^n$ for some fixed n and C is the set of half-spaces defined by surfaces of degree at most k for some fixed k (see Example 3.1.3). Then $\mathbf{C}'$ and $\mathbf{C}''$ are poly-learnable by Theorem 3.2.4.

4. *Summary, Open Problems, and Further Research*

We have shown that the VC dimension is a useful combinatorial parameter in the context of Valiant's model of learnability by using it to give necessary and/or sufficient conditions for various types of learnability. Although we have distinguished between feasible and infeasible learning problems, we have not attempted to provide tight bounds on the number of examples and the computation time needed for various learning problems. A more refined analysis for some cases is given in [20], and in [64], where the parallel computational complexity of learning is investigated. However, considerable further research remains to be done in this area. In particular, there may be interesting general trade-offs between the sample size required for learning and the computational effort required to produce a consistent hypothesis that are yet to be discovered (see [12]).

These issues are important if this theory of learnability is to find useful applications, for example to learning problems arising in Artificial Intelligence [26–28, 37, 56, 60]. In many of the AI models of learning from examples the domain is defined by n multivalued attributes that can range from Boolean to real-valued. Attributes whose values are organized into certain types of hierarchies are also used. These domains tend to have a structure that is roughly intermediate between the Boolean domains considered in [37] and the continuous domains considered here. The techniques we have described in this paper are easily applied to these domains, and generally give better results than the simple counting argument of Theorem 2.2 [28]. More complex learning problems in which the domain consists of a set of labeled graphs representing descriptions of visual scenes and the target classes are defined by certain types of first-order formulas (e.g., Winston's "arch" concept in a blocks world domain [67]) are considered in [26] and [60]. The application of the results given here to learning methods that use connectionist or neural network representations is discussed in [7].

As for other research directions, AI applications also bring up the question of incremental learning, in which individual examples are processed one at a time and only the current hypothesis is maintained and updated [31, 42]. They also bring up the issues of misclassification in the examples and the possibility of stochastically defined target concepts, discussed in Section A3 of the Appendix, and the possibility of combining random examples with other types of information, for example, the various oracles and queries discussed in [2], [3], and [59].

Another important issue is that of the representation chosen for hypotheses. In Section 3 we prove a number of results on learning algorithms that represent their hypotheses in the same form that the target concept is represented. This is fine for the positive learnability results. However, for negative learnability results, one would often like something stronger, something that shows that the concept class is not polynomially learnable by any hypothesis space of the type described in Section 3.1. Using results from [24] on poly-random collections, it can be shown that there are concept classes $\mathbf{C} = \{(\{0, 1\}^n, C_n)\}_{n \geq 1}$ with the VC dimension of C_n growing only polynomially in n that are not polynomially learnable in this strong sense, given the existence of 1–1 one-way functions [53]. Given more specific cryptographic assumptions, Kearns and Valiant have shown that such "strongly

hard to learn" classes include the class of all concepts represented by Boolean formulas of size bounded by a fixed polynomial in n [35]. In [53] a notion of reduction among learning problems is developed that, in conjunction with the above result, implies that regular languages (represented by deterministic finite state automata) are also probably "strongly hard to learn."

Finally, we note that here we have only considered the problem of learning indicator functions of sets. Other variants of the model will be required to handle the problem of learning real-valued target functions. This problem is addressed in [17], [18], [23], [54], and [61] from a purely statistical point of view. A comprehensive overview of methods that have been proposed for generalizing the VC dimension to classes of real-valued functions are contained in [18]. These results should be combined with considerations of computational complexity at some point, laying the groundwork for a more general computational learning theory.[7]

We close with a few concrete open problems:

(1) Let $\mathbf{C} = \{(\{0, 1\}^n, C_n)\}_{n \geq 1}$, where C_n is the class of concepts represented by n-term n-variable DNF expressions. Is $\mathbf{C}$ polynomially learnable by $\mathbf{H}$ for any hypothesis space $\mathbf{H}$ as defined in Section 3.1? This is a variant of one of the problems posed in [59].

(2) Can we, perhaps by allowing probabilistic Occam algorithms in analogy with the r-poly hy-fi's of Section 3.1, obtain a converse of Theorem 3.2.1(i)? (I.e., does learnability with respect to target complexity imply the ability to efficiently find simple hypotheses?)

(3) Can Theorem 3.2.4 be extended to $\mathbf{C} = (X, \mathbf{A}(C))$, where $\mathbf{A}(C)$ is the closure of C under finite unions, intersection and complement, and concept complexity is measured as the length of the smallest expression for $c \in \mathbf{A}(C)$? For example, if $X = E^n$ for some fixed n and C is the set of half-spaces, then concepts in $\mathbf{C} = (X, \mathbf{A}(C))$ can be represented as (small enough) unions of simplices, and from this is can be shown that $\mathbf{C}$ is poly-learnable using Theorem 3.2.4.

Appendix A

A1. *Definition of Well-Behaved Classes and ϵ-Transversals*

For Theorem 2.1 to apply, we require that the concept class C have some additional properties related to measurability, beyond the assumption that all sets in C are Borel. The properties we need are related to the following definitions, which will be used in the proof of Theorem 2.1(ii)(a) given below.

Definition. For any class of regions $R \subseteq 2^X$, probability distribution P on X, and $\epsilon > 0$, let $R_{P,\epsilon} = \{r \in R : P(r) > \epsilon\}$. $N \subseteq X$ is an *ϵ-transversal* for R (with respect to P) if N contains a point in every $r \in R_{P,\epsilon}$.

This definition of an ϵ-transversal generalizes the notion of an ϵ-net from [29] to arbitrary probability distributions. (We changed the notation here to avoid confusion with the topological notion of an ϵ-net used in [61] and elsewhere.)

Example A1.1. If X is the interval $[0, 1]$, P is the uniform distribution and R is the set of closed intervals in X, then the set of all points ϵk, for natural numbers k in the range $0 \leq k \leq 1/\epsilon$, is an ϵ-transversal for R for any $\epsilon > 0$. In fact, R has an ϵ-transversal of this size for any distribution P on X. On the other hand, if R is all open sets, then clearly there are no finite ϵ-transversals for R with respect to the uniform distribution.

[7] Some small progress along these lines in [28a].

We are concerned with the probability of drawing an ϵ-transversal for a class of regions R by independently drawing random points from X. In particular, we need to measure the probabilities of the following events.

Definition. For any $m \geq 1$ and $\epsilon > 0$, Q_ϵ^m denotes the set of all $\bar{x} \in X^m$ such that the set of distinct elements of $\bar{x}$ does not form an ϵ-transversal for R with respect to P, that is, such that there exists $r \in R_{P,\epsilon}$ with $\bar{x} \cap r = \emptyset$. J_ϵ^{2m} denotes the set of all $\overline{xy} \in X^{2m}$, where $\bar{x}, \bar{y} \in X^m$, such that there exists $r \in R_{P,\epsilon}$, where $\bar{x} \cap r = \emptyset$ and $|\{i : y_i \in r,\ 1 \leq i \leq m\}| \geq \epsilon m/2$, that is, no element of r occurs in the first half of the sequence, but elements of r occur with frequency at least $\epsilon m/2$ in the second half.

In our learning application, the class of regions R will be formed by taking symmetric differences between hypotheses in a hypothesis space $H \subseteq 2^X$ and a fixed target concept $c \subseteq X$, that is, we have $R = \{h \Delta c : h \in H\}$. Each of these regions in R represents the set of points in X that are counterexamples to a hypothesis $h \in H$ for the target concept c, that is, the error region of that hypothesis. Thus, if the sequence of points in a sample is an ϵ-transversal of R, then the sample contains counterexamples to every hypothesis that has error greater than ϵ.

Definition (Shai Ben-David). H is *well behaved* if the sets Q_ϵ^m and J_ϵ^{2m} defined above are measurable for every class of regions $R = \{h \Delta c : h \in H\}$ for any Borel set c, $\epsilon > 0$, $m \geq 1$ and distribution P on X.

An example of a hypothesis space H that is not well behaved is the following (see [66]). Let X be the closed interval $[0, 1]$ and let X be well-ordered such that all prefixes of the well-ordering are countable.[8] Let H consist of all suffixes of the well-ordering, including the empty set. Note that every set in H is the complement of a countable set, hence, it is a Borel set. Let the target concept c be $\emptyset$, so that $R = H$ and P be the uniform distribution. It can be shown that in this case J_ϵ^{2m} is not measurable for all $0 < \epsilon < 1$, even for $m = 1$. Hence, H is not well behaved.

This example also shows that the well-behaved condition is required for Theorem 2.1: It is readily verified that the VC dimension of H is 1, yet Theorem 2.1(ii)(a) fails for $C = H$. To see this, let P be the uniform distribution on X, let the target concept c be $\emptyset$ as above, and consider the following learning function A: Given any finite sequence of examples (all necessarily negative), form the hypothesis consisting of the largest (by set containment) suffix in H that contains no point from the sequence of examples. Clearly, A is consistent. However, since the target concept is $\emptyset$ and A's hypothesis always has measure 1, A is not a learning function for H with respect to P for any sample size. Theorem A.3 of [61] (described below in Proposition A3.1) also fails for this case.

On the other hand, virtually any concept class that one might consider in the context of machine-learning applications will be well behaved. Proofs of good behavior for most common concept classes can be derived by showing that they satisfy Dudley's condition of *universal separability* [7a, 54] (this notion is called *countably S-coverable* in [7a]; see the appendix of [54] for a discussion of other approaches).

Definition. A hypothesis space $H \subseteq 2^X$ is *universally separable* if there exists a countable subset H_0 of H such that each set h in H can be written as the pointwise limit of some sequence of sets in H_0, that is, for all $h \in H$ there is a sequence h_1,

[8] The existence of such a well-ordering requires the Continuum Hypothesis.

$h_2, \ldots$ in H_0 such that for every $x \in X$, there exists n such that for all $i \geq n$, $x \in h_i$ if and only if $x \in h$.

Classes of rectangles, half-spaces, etc. can easily be shown to be universally separable (see exercises 4, 5 and 7 in chapter II of [54]).

LEMMA A1.1 *(Shai Ben-David). If H is universally separable, then H is well behaved.*

PROOF. Fix a Borel set c and let $R = \{h \Delta c : h \in H\}$. It suffices to show that the sets Q_ϵ^m and J_ϵ^{2m} are Borel sets. We show this for Q_ϵ^m, the argument for J_ϵ^{2m} being similar.

Since H is universally separable, R is universally separable. Let T be a countable subset of R such that every set in R is the pointwise limit of a sequence of sets in T. Let $\gamma_1, \gamma_2, \ldots$ be a decreasing sequence of strictly positive real numbers converging to 0 and let $\epsilon_1, \epsilon_2, \ldots$ be a decreasing sequence of strictly positive real numbers converging to ϵ. For every $i, j \geq 1$, let

$$T_{i,j} = \{t \in T : \text{there exists } r \in R \text{ with } P(r) \geq \epsilon_i \text{ and } P(t \Delta r) \leq \gamma_j\}.$$

We claim that

$$Q_\epsilon^m = \bigcup_{i=1}^{\infty} \bigcap_{j=1}^{\infty} \bigcup_{t \in T_{i,j}} \{\bar{x} \in X^m : \bar{x} \cap t = \varnothing\},$$

and hence, Q_ϵ^m is a Borel set.

To see this, note that if $\bar{x}$ is in the right hand set above then $\bar{x} \cap t = \varnothing$ for some $t \in T_{i,j}$ where $\epsilon_i - \gamma_j > \epsilon$. For any such t we have $t \in R$ and $P(t) > \epsilon$. Thus $\bar{x} \in Q_\epsilon^m$. On the other hand, if $\bar{x} \in Q_\epsilon^m$, then there exists $i \geq 1$ such that $\bar{x} \cap r = \varnothing$ for some $r \in R$ with $P(r) \geq \epsilon_i$. Since there is a sequence of sets in T that converge pointwise to r, for every $j \geq 1$ there is a set t in T with $P(t \Delta r) \leq \gamma_j$ and $\bar{x} \cap t = \bar{x} \cap r = \varnothing$. Thus, $\bar{x}$ is the right hand set above. □

Although Lemma A1.1 is useful in proving that most common hypothesis spaces are well behaved, it is not always sufficient. For example, if $X = [0, 1]$ and $H = \{\{x\} : x \in X\}$, then H is well behaved but not universally separable.

We are now ready to proceed with Section A2:

A2. *Proof of Theorem* 2.1(ii)(a)

For the next two lemmas let $R \subseteq 2^X$ be a fixed nonempty class of sets and P be a distribution on X such that Q_ϵ^m and J_ϵ^{2m} are measurable for all $m \geq 1$ and $\epsilon > 0$. The proofs of these lemmas are analogous to those of Lemma and Theorem 2 of [62]. Using Proposition A2.5, they generalize Lemmas 3.4 and 3.5 of [29] to arbitrary probability distributions.

LEMMA A2.1. *For any $\epsilon > 0$ and $m \geq 2/\epsilon$, $P^m(Q_\epsilon^m) < 2P^{2m}(J_\epsilon^{2m})$.*

PROOF. We show that $P^{2m}(J_\epsilon^{2m}) > \frac{1}{2}P^m(Q_\epsilon^m)$. By Fubini's theorem (see, e.g., [57])

$$P^{2m}(J_\epsilon^{2m}) = \int_{X^{2m}} I_{J_\epsilon^{2m}}(x_1 \cdots x_{2m})\, dP^{2m} = \int_{\bar{x} \in X^m} \left(\int_{\bar{y} \in X^m} I_{J_\epsilon^{2m}}(\bar{x}, \bar{y})\, dP^m \right) dP^m$$

and this is

$$\geq \int_{\bar{x} \in Q_\epsilon^m} \left(\int_{\bar{y} \in X^m} I_{J_\epsilon^{2m}}(\bar{x}, \bar{y})\, dP^m \right) dP^m,$$

since $Q_\epsilon^m \subseteq X^m$. For each $\bar{x} \in Q_\epsilon^m$ let $r_{\bar{x}}$ be a region in $R_{P,\epsilon}$ such that $\bar{x} \cap r = \emptyset$. Let K_ϵ^{2m} be the set of all $\overline{\bar{x}\bar{y}} \in X^{2m}$, where $\bar{x}, \bar{y} \in X^m$ and $|\{i : y_i \in r_{\bar{x}}\}| \geq \epsilon m/2$. Obviously, $J_\epsilon^{2m} \supseteq K_\epsilon^{2m}$ and thus

$$P^{2m}(J_\epsilon^{2m}) \geq \int_{\bar{x} \in Q_\epsilon^m} \left(\int_{\bar{y} \in X^m} I_{K_\epsilon^{2m}}(\bar{x}, \bar{y})\, dP^m \right) dP^m.$$

For each $\bar{x} \in Q_\epsilon^m$, the inner integral is just the probability that an event with probability at least ϵ occurs with frequency at least $\epsilon m/2$ in m independent Bernoulli trials. This probability is greater than $\frac{1}{2}$ for any $m \geq 2/\epsilon$: For $2/\epsilon \leq m < 8/\epsilon$ this can be shown by a case analysis using the exact formula for the binomial distribution; for $m \geq 8/\epsilon$, this is easy to prove by applying Chebyshev's inequality. Hence

$$P^{2m}(J_\epsilon^{2m}) > \int_{\bar{x} \in Q_\epsilon^m} \frac{1}{2}\, dP^m = \frac{1}{2} P^m(Q_\epsilon^m). \qquad \square$$

LEMMA A2.2. *$P^{2m}(J_\epsilon^{2m}) \leq \Pi_R(2m) 2^{-\epsilon m/2}$ for all $m \geq 1$ and $\epsilon > 0$.*

PROOF. For each j, $1 \leq j \leq (2m)!$, let σ_j be a distinct permutation of the indices $1, \ldots, 2m$. It is clear that

$$P^{2m}(J_\epsilon^{2m}) = \int_{X^{2m}} I_{J_\epsilon^{2m}}(\bar{x})\, dP^{2m} = \int_{X^{2m}} I_{J_\epsilon^{2m}}(\sigma_j(\bar{x}))\, dP^{2m}$$

for all permutations σ_j. Hence

$$P^{2m}(J_\epsilon^{2m}) = \int_{X^{2m}} \frac{1}{(2m)!} \sum_{j=1}^{(2m)!} I_{J_\epsilon^{2m}}(\sigma_j(\bar{x}))\, dP^{2m}.$$

Thus, it suffices to show that

$$\frac{1}{(2m)!} \sum_{j=1}^{(2m)!} I_{J_\epsilon^{2m}}(\sigma_j(\bar{x})) \leq \Pi_R(2m) 2^{-\epsilon m/2},$$

for all $\bar{x} \in X^{2m}$.

Consider a fixed $\bar{x} \in X^{2m}$. Let S be the set of distinct elements of X that appear in $\bar{x}$. For each permutation $\sigma_j(\bar{x})$ in J_ϵ^{2m} there is a subset T of S that is a witness to the fact that $\sigma_j(\bar{x}) \in J_\epsilon^{2m}$ in the sense that there exists $r \in R_{P,\epsilon}$ such that $T = r \cap S$, all occurrences of members of T appear in the second half of $\sigma_j(\bar{x})$ and there are at least $\epsilon m/2$ such occurrences. However, for a given T, this can occur in only a small fraction of all permutations of $\bar{x}$. In particular, if there are l occurrences of members of T in $\bar{x}$, then T is a witness for at most a fraction

$$\frac{\binom{m}{l}}{\binom{2m}{l}} = \frac{m(m-1)\cdots(m-l+1)}{2m(2m-1)\cdots(2m-l+1)} \leq 2^{-l} \leq 2^{-\epsilon m/2}$$

of all permutations of $\bar{x}$. Since $|S| \leq 2m$, there are at most $\Pi_R(2m)$ distinct subsets T of S induced by intersections with regions $r \in R$, and, hence, at most $\Pi_R(2m)$ distinct subsets induced by intersections with $r \in R_{P,\epsilon}$. Thus, there are at most $\Pi_R(2m)$ distinct witnesses. It follows that

$$\frac{1}{(2m)!} \sum_{j=1}^{(2m)!} I_{J_\epsilon^{2m}}(\sigma_j(\bar{x})) \leq \Pi_R(2m) 2^{-\epsilon m/2}. \qquad \square$$

The above two lemmas can be combined to give an upper bound on the probability of not getting an ϵ-transversal for a class R of regions in terms of the

function $\Pi_R(m)$ (see [29]). To apply this to learning we need to use regions that form the symmetric differences between the target concept and the various possible hypotheses. The following lemma is useful.

LEMMA A2.3. *For any $m \geq 1$, $c \subseteq X$, and $H \subseteq 2^X$, $\Pi_H(m) = \Pi_R(m)$, where $R = \{h \Delta c : h \in H\}$.*

PROOF. For any subset $t \in X$, let t' denote the complementary subset $X - t$. The following holds for any $h_1, h_2 \in H$ and for any S, $c \subseteq X$:

$$\begin{aligned} h_1 \cap S = h_2 \cap S &\Leftrightarrow h_1 \cap c' \cap S = h_2 \cap c' \cap S \quad \text{and} \\ &\quad h_1' \cap c \cap S = h_2' \cap c \cap S \\ &\Leftrightarrow ((h_1' \cap c) \cup (h_1 \cap c')) \cap S = ((h_2' \cap c) \cup (h_2 \cap c')) \cap S \\ &\Leftrightarrow (h_1 \Delta c) \cap S = (h_2 \Delta c) \cap S. \end{aligned}$$

This implies that $|\Pi_H(S)| = |\Pi_R(S)|$ and the lemma follows. □

THEOREM A2.1. *Let H be any nonempty well-behaved hypothesis space contained in 2^X, P be any probability distribution on X and the target concept c be any Borel set contained in X. Then for any $\epsilon > 0$ and $m \geq 1$, given m independent random examples of c drawn according to P, the probability that there exists a hypothesis in H that is consistent with all of these examples and has error greater than ϵ is at most*

$$2\Pi_H(2m)2^{-\epsilon m/2}.$$

PROOF. Let $R = \{h \Delta c : h \in H\}$. Each region in R represents the symmetric difference between the target concept c and a hypothesis $h \in H$. A hypothesis h has error greater than ϵ only if its symmetric difference with c has probability greater than ϵ. Hence, if the points from the examples that are drawn form an ϵ-transversal for R, every hypothesis in H that has error greater than ϵ will have an example drawn from its symmetric difference with c. Since this implies that the hypothesis is inconsistent with the example, no such hypothesis will be consistent with the entire sample. Hence, there exists a hypothesis in H that has error greater than ϵ and is consistent with all the examples only if the points of the examples do not form an ϵ-transversal of R.

Since H is well behaved and c is a Borel set, the sets Q_ϵ^m and J_ϵ^m defined from R are measurable. If $m \geq 2/\epsilon$, then, by Lemmas A2.1 and A2.2, the probability that the points drawn in the m random examples do not form an ϵ-transversal for R with respect to P (i.e., the probability of Q_ϵ^m) is less than

$$2\Pi_R(2m)2^{-\epsilon m/2}.$$

By Lemma A2.3, $\Pi_R(2m) = \Pi_H(2m)$, hence, the probability of not getting an ϵ-transversal for R is less than

$$2\Pi_H(2m)2^{-\epsilon m/2}.$$

If $1 \leq m < 2/\epsilon$, then $2\Pi_H(2m)2^{-\epsilon m/2} > 1$, and the bound holds trivially. □

We now bound $\Pi_H(m)$ using the VC dimension of H.

Definition. For all $d \geq 0$ and $m \geq 0$, $\Phi_d(m) = \sum_{i=0}^{d} \binom{m}{i}$ if $m \geq d$, and $\Phi_d(m) = 2^m$, otherwise.

PROPOSITION A2.1

(i) *If the VC dimension of H is d, where $d \geq 0$, then $\Pi_H(m) \leq \Phi_d(m)$ for all $m \geq 0$.*

(ii) *$\Phi_d(m) \leq m^d + 1$ for all $d \geq 0$ and $m \geq 0$. $\Phi_d(m) \leq m^d$ for all $d \geq 2$ and $m \geq 2$.*

(iii) *$\Phi_d(m) \leq 2(m^d/d!) \leq (em/d)^d$ for all $m \geq d \geq 1$.*

PROOF OF PART (i). A short inductive proof of part (i) above can be found in [6], along with its history of independent discoveries. We sketch the proof for completeness, using the notation from [29].

We show that for any set S with $|S| = m$ and any family F of subsets of S that has VC dimension d, $|F| \leq \Phi_d(m)$. Letting $F = \Pi_H(S)$ for arbitrary $S \subseteq X$, we obtain the result.

The assertion is trivially true for $d = 0$ and any $m \geq 0$, since $|F| = 1$ in this case. It is also trivially true for $m = 0$ and any $d \geq 0$. Assume the assertion is true for all $m \geq 0$ when F has VC dimension at most $d - 1$, and for $m - 1$ when F has VC dimension d, where $d \geq 1$ and $m \geq 1$.

Consider a particular set S with $|S| = m$ and a family F of subsets of S of VC dimension d. Choose any point $x \in S$. Let

$$F - x = \{f - \{x\} : f \in F\}$$

and

$$F^{(x)} = \{f \in F : x \notin f, f \cup \{x\} \in F\}.$$

Note that both $F - x$ and $F^{(x)}$ are families of subsets of $S - \{x\}$ and that $|F| = |F - x| + |F^{(x)}|$. (In mapping f to $f - \{x\}$ for each $f \in F$, pairs of the form $\{f, f \cup \{x\}\}$ map to the same set. These are correctly accounted for by adding $|F^{(x)}|$.) Obviously $F - x$ has VC dimension at most d; hence, by assumption, $|F - x| \leq \Phi_d(m - 1)$. We show that $F^{(x)}$ has VC dimension at most $d - 1$ and hence, $|F^{(x)}| \leq \Phi_{d-1}(m - 1)$.

Let A be a subset of $S - \{x\}$ that can be shattered by $F^{(x)}$. Then it is easy to see that $A \cup \{x\}$ can be shattered by F: For $A' \subseteq A$ there is an $f \in F^{(x)}$ with $A' = A \cap f$. Since $x \notin f$, $A' = (A \cup \{x\}) \cap f$ and $A' \cup \{x\} = (A \cup \{x\}) \cap (f \cup \{x\})$, where both f and $f \cup \{x\}$ are in F. It follows that $A \cup \{x\}$ can be shattered by F. Since the VC dimension of F is d, we must have $|A \cup \{x\}| \leq d$, so $|A| \leq d - 1$. Thus $F^{(x)}$ has VC dimension at most $d - 1$.

It follows that $|F| \leq \Phi_d(m - 1) + \Phi_{d-1}(m - 1)$. It is easily verified that $\Phi_{d-1}(m - 1) + \Phi_d(m - 1) = \Phi_d(m)$ for d, $m \geq 1$, so this completes the induction.

PROOF OF PART (ii). This part is easily verified.

PROOF OF PART (iii). The second inequality clearly holds for $d = 1$. To show that it holds for $d \geq 2$ we use Stirling's Approximation [38, p. 111]:

$$2\frac{m^d}{d!} < \frac{2m^d}{\sqrt{2\pi d}d^d e^{-d}} = \sqrt{\frac{2}{\pi d}}\left(\frac{em}{d}\right)^d < \left(\frac{em}{d}\right)^d.$$

The proof of the first inequality of Part (iii) is analogous to the proof of a similar bound given in [61, p. 166] (see also [17]). It is by induction on m and d:

If $d = 1$, $\Phi_d(m) = m + 1 \leq 2m$, so the result follows. If $m = d > 1$, $\Phi_d(m) = 2^d$. Observe that for $d > 1$, $2 \leq (1 + 1/(d - 1))^{d-1}$ by the binomial theorem. Thus,

using induction on d we obtain

$$\begin{aligned} 2^d &\le \left(\frac{d}{d-1}\right)^{d-1} 2^{d-1} && \text{(by the above observation)} \\ &\le 2\left(\frac{d}{d-1}\right)^{d-1} \frac{(d-1)^{d-1}}{(d-1)!} && \text{(by ind. hyp.)} \\ &= 2\,\frac{d^d}{d!}, \end{aligned}$$

which verifies the result for the case $m = d > 1$.

Now assume $m > d > 1$. Since $\Phi_d(m) = \Phi_{d-1}(m-1) + \Phi_d(m-1)$, it suffices to show that

$$\begin{aligned} 2\,\frac{(m-1)^{d-1}}{(d-1)!} + 2\,\frac{(m-1)^d}{d!} &\le 2\,\frac{m^d}{d!}, \\ \Leftrightarrow d(m-1)^{d-1} + (m-1)^d &\le m^d \\ \Leftrightarrow (d+m-1)(m-1)^{d-1} &\le m^d \\ \Leftrightarrow \frac{(d+m-1)}{(m-1)} &\le \frac{m^d}{(m-1)^d} \\ \Leftrightarrow 1 + \frac{d}{m-1} &\le \left(1 + \frac{1}{m-1}\right)^d. \end{aligned}$$

The last inequality follows from the binomial theorem. □

From this proposition, it follows that whenever the VC dimension of H is finite, then $\Pi_H(m)$ grows only polynomially in m. Since the negative exponential term in the bound of Theorem A2.1 eventually dominates the polynomial $\Pi_H(2m)$, this shows that the probability that there exists a hypothesis of error greater than ϵ that is consistent with an m-sample goes very rapidly to zero for large m. We now estimate the sample size m required to make this probability less than δ.

LEMMA A2.4. *If*

$$m \ge \max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{8d}{\epsilon}\log\frac{13}{\epsilon}\right), \quad \textit{then } 2\Phi_d(2m)2^{-\epsilon m/2} \le \delta.$$

PROOF. By Proposition A2.1(iii), $\Phi_d(2m) \le (2em/d)^d$, thus it suffices to show that $2(2em/d)^d \le \delta 2^{\epsilon m/2}$, which is equivalent to $\epsilon m/2 \ge d\log(2em/d) + \log(2/\delta)$. The first of the two bounds on m implies $\epsilon m/4 \ge \log(2/\delta)$. Thus, it suffices to show that $\epsilon m/4 \ge d\log(2em/d)$. With $q = 4d/\epsilon$ and $t = 2e/d$, this inequality is expressed as $m \ge q\log(tm)$. If this inequality holds for some value of m, it will also hold for larger values; so suppose m is equal to the second bound in the statement of the theorem. We need to show that $2q\log(13/\epsilon) \ge q\log(2qt\log(13/\epsilon))$, which is equivalent to $13^2/(2qt\epsilon^2) = 13^2/(16e\epsilon) \ge \log(13/\epsilon)$. Again, if the latter inequality holds for some value of $1/\epsilon$, it will also hold for larger values. The inequality is easily verified for $\epsilon = 1$. □

THEOREM A2.2. *Let H be any well-behaved hypothesis space of finite VC dimension d contained in 2^X, P be any probability distribution on X and the target concept c be any Borel set contained in X. Then for any $0 < \epsilon, \delta < 1$, given*

$$m \ge \max\left(\frac{4}{\epsilon}\log\frac{2}{\delta}, \frac{8d}{\epsilon}\log\frac{13}{\epsilon}\right)$$

independent random examples of c drawn according to P, with probability at least $1 - \delta$, *every hypothesis in H that is consistent with all of these examples has error at most* ϵ.

PROOF. This follows directly from Theorem A2.1, Proposition A2.1, and Lemma A2.4. □

Part (ii)(a) of Theorem 2.1 follows directly from the above theorem. □

Now, in order to complete the proof of Theorem 3.2.1, we close this section with the following lemma:

LEMMA A2.5

(a) *If* $0 < \epsilon, \delta < 1$, $0 \le \alpha < 1$, $k \ge 1$ *and*

$$m = max\left(\frac{4}{\epsilon}log\frac{2}{\delta}, \left(\frac{8k}{\epsilon}log\frac{13}{\epsilon}\right)^{1/(1-\alpha)}\right),$$

then $2(2em/(km^\alpha))^{km^\alpha}2^{-\epsilon m/2} \le \delta$.

(b) *If* $l \ge 1$ *and the second term of the bound on m above is replaced by*

$$\frac{2^{l+4}k}{\epsilon}\left(log\frac{8(2l+2)^{l+1}k}{\epsilon}\right)^{l+1},$$

then $2(2m)^{k(\log m)^l}2^{-\epsilon m/2} \le \delta$.

PROOF. We first prove part (a). To simplify expressions in the proof, we let $q = 4k/\epsilon$ and $t = 2e/k$. Using the first bound on m as in the proof of Lemma A2.4, it suffices to show that $\epsilon m/4 \ge km^\alpha \log 2(em/(km))^\alpha$, that is, $m^{1-\alpha} \ge q\log(tm^{1-\alpha})$. Again, we suppose that m is equal to its second bound, since the inequality only improves as m increases. We must show

$$2q\log\frac{13}{\epsilon} \ge q\log\left(2qt\log\frac{13}{\epsilon}\right),$$

which as in Lemma A2.4 is equivalent to $13^2/(16e\epsilon) \ge \log(13/\epsilon)$ and this holds for all $\epsilon \le 1$.

For part (b), we let $q = 4k/\epsilon$ as above and $r^{l+1} = 2q(2l+2)^{l+1}$. Using the first bound on m as in the proof of Lemma A2.4, it suffices to show that $m \ge 2q(\log m)^{l+1}$. Again, we suppose that m is equal to its second bound, since the inequality only improves as m increases. We must show

$$2^{l+2}q(\log r^{l+1})^{l+1} \ge 2q[\log(2^{l+2}q(\log r^{l+1})^{l+1})]^{l+1}.$$

Canceling $2q$ and removing the $l+1$ powers gives

$$2\log r^{l+1} \ge \log(2^{l+2}q(\log r^{l+1})^{l+1}) = \log\left(2q(2l+2)^{l+1}\left(\frac{1}{l+1}\right)^{l+1}(\log r^{l+1})^{l+1}\right)$$

$$= \log\left(r^{l+1}\left(\frac{1}{l+1}\log r^{l+1}\right)^{l+1}\right) = \log r^{l+1} + \log((\log r)^{l+1}).$$

This inequality reduces to $r \ge \log r$, which certainly holds. □

A3. *Generalizations*

A more general result than that of Theorem A2.1 above is needed if our learning model allows for the possibility of misclassification in the examples given to the

learning function. This possibility is considered in several recent papers on Valiant's learnability model [4, 34, 39, 58, 60]. When misclassifications are present, it may not even be possible to find a hypothesis that is consistent with all of the examples.

This can also occur when the target concept is not in the hypothesis space used by the learning algorithm, or more generally, when the target concept itself is defined stochastically, as is a common assumption in the pattern recognition literature (e.g., [14, 16]). Here both the distribution on the learning domain X and the target concept $c \subseteq X$ are replaced by a single distribution on $X \times \{0, 1\}$ which gives the probability of drawing any given example (x, a), where $x \in X$ and $a \in \{0, 1\}$. For any given $x \in X$ it is possible that both $(x, 0)$ and $(x, 1)$ will have positive probability, so the "target concept" will not in general be representable as a subset of the learning domain X.

The notion of a stochastic target concept can be used to model the case in which there is a (deterministic) underlying target concept and a fixed distribution on the learning domain, but the random examples received by the learning algorithm are modified by an additional random process that may change the instance points and/or their classifications, as in [4], [39], and [58]. However, here the action of this secondary "noise" process is allowed to depend on the nature of the example it intercepts (e.g., it may be more apt to change the classification of examples that are "close to the border" of the underlying target concept). On the other hand, stochastic target concepts cannot be used to model the adversarial noise processes considered in [34], [58], and [60].

In the standard pattern recognition approach, the goal of the learning algorithm is to find a (deterministic) hypothesis that is a good appproximation to the stochastic target concept. When the stochastic element of the target concept is due to noise, this means finding a hypothesis that will, with high probability, agree with random examples generated by the composition of two random processes: the random process generating examples of the underlying target concept and the random noise process. However, Angluin and Laird [4] have argued that in some situations where noise is present in the training examples it is more appropriate to try to find a hypothesis that is a good approximation of the underlying target concept, that is, a hypothesis that with high probability will agree with random ("noise-free") examples of the underlying target concept, as in the definition of learning we have used in previous sections of this paper. They also exhibit a simple learning situation in which these goals are incommensurate.

In this final section we give some generalizations of the results of the previous section that apply when learning stochastic target concepts. Here we take the standard pattern recognition approach, showing that certain deterministic hypotheses will be good approximations to stochastic target concepts with high probability. However, the results we give are quite general, and can also be used indirectly in other approaches (see, e.g., [34, Theorem 7]; generalize by replacing the Chernoff bound used with Corollary A3.1 below). The results are all relatively straightforward corollaries of a theorem of Vapnik [61, Theorem A.3, page 176].

Definition. Let P be any probability distribution on X, and $\bar{x} = (x_1, \ldots, x_m) \in X^m$, $m \geq 1$. For any measurable $r \subseteq X$, by $\hat{P}_{\bar{x}}(r)$ we denote the empirical estimate of $P(r)$ based on the sample $\bar{x}$, that is,

$$\hat{P}_{\bar{x}}(r) = \frac{|\{i : x_i \in r,\ 1 \leq i \leq m\}|}{m}.$$

PROPOSITION A3.1 [61]. *Let $R \subseteq 2^X$ be a class of regions of X with suitable measurability properties, P be any probability distribution on X, $m \geq 1$, $1 < q \leq 2$, and $\chi > 0$. If $\bar{x} \in X^m$ is chosen randomly according to P^m, then the probability that there exists $r \in R$, $P(r) \neq 0$, such that*

$$\frac{P(r) - \hat{P}_{\bar{x}}(r)}{P(r)^{1/q}} > \chi$$

is less than

$$8\Pi_R(2m)exp\left(\frac{-\chi^2 m^{2-2/q}}{4}\right).$$

Because the proof is lengthy, we do not sketch it here. The measurability properties required are similar to those given in our notion of well-behaved classes. We note only that it employs techniques similar to those used in [62], which form the basis of Lemmas A2.1 and A2.2 given above. A useful corollary of this result is the following:

COROLLARY A3.1. *Let R, P and m be as in Proposition A3.1, and $0 < \epsilon, \gamma \leq 1$. If $\bar{x} \in X^m$ is chosen randomly according to P^m, then the probability that there exists a region $r \in R$ with $P(r) > \epsilon$ such that*

$$\hat{P}_{\bar{x}}(r) \leq (1 - \gamma)P(r)$$

is less than

$$8\Pi_R(2m)exp\left(\frac{-\gamma^2 \epsilon m}{4}\right).$$

PROOF. Let $q = 2$ and $\chi = \gamma\sqrt{\epsilon}$. Note that $\chi > 0$. For any $r \in R$, if $\hat{P}_{\bar{x}}(r) \leq (1 - \gamma)P(r)$, then $P(r) - \hat{P}_{\bar{x}}(r) \geq \gamma P(r)$, hence, $(P(r) - \hat{P}_{\bar{x}}(r))/\sqrt{P(r)} \geq \gamma\sqrt{P(r)}$ when $P(r) > 0$. If $P(r) > \epsilon$, then $\gamma\sqrt{P(r)} > \gamma\sqrt{\epsilon} = \chi$. Hence, by Proposition A3.1, the probability that there exists $r \in R$ such that $P(r) > \epsilon$ and $\hat{P}_{\bar{x}}(r) \leq (1 - \gamma)P(r)$ is at most

$$8\Pi_R(2m)\exp\left(\frac{-\chi^2 m^{2-2/q}}{4}\right),$$

which is

$$8\Pi_R(2m)\exp\left(\frac{-\gamma^2 \epsilon m}{4}\right). \qquad \square$$

This result generalizes one of the Chernoff bounds that is frequently used in papers on learnability in discrete domains [4, 5, 34, 36, 39, 52, 60]. Also note that in terms of ϵ-transversals, letting $\gamma = \frac{1}{2}$ it shows that the probability of not getting in m random draws an ϵ-transversal in which each region r of R with $P(r) \geq \epsilon$ is hit with frequency at least $P(r)/2$ is at most $8\Pi_R(2m)\exp(-\epsilon m/16)$. Using Proposition A2.1(i), this is comparable to the bound given in [26, Theorem 3.6] on the probability of getting any ϵ-transversal.

We can adapt Corollary A3.1 for application to the problem of learning stochastic target functions as follows:

Definition. Let X be a learning domain and $H \subseteq 2^X$ be a hypothesis space. As usual we assume that each $h \in H$ is a Borel set. Let $Y = X \times \{0, 1\}$ and P be a

distribution on Y. We denote by *disagree*(h) the set of all points in Y that disagree with h, that is, *disagree*(h) = $\{(x, a) \in Y : x \in h$ and $a = 0$ or $x \notin h$ and $a = 1\}$. The *error* of h (with respect to P), denoted $er(h)$, is P(*disagree*(h)). For any $\tilde{y} \in Y^m$ we denote by $\hat{er}_{\tilde{y}}(h)$ the empirical estimate of $er(h)$ based on $\tilde{y}$, that is, $\hat{er}_{\tilde{y}}(h) = \hat{P}_{\tilde{y}}$(*disagree*($h$)).

In analogy to Theorem A2.1 above, we have

THEOREM A3.1. *Let H, P, and Y be as above, where H has suitable measurability properties, $m \geq 1$ and $0 < \epsilon, \gamma \leq 1$.*

(i) *If $\tilde{y} \in Y^m$ is chosen randomly according to P^m, then the probability that there exists a hypothesis $h \in H$ with $er(h) > \epsilon$ such that*

$$\hat{er}_{\tilde{y}}(h) \leq (1 - \gamma) er(h)$$

is less than

$$8\Pi_H(2m) exp\left(\frac{-\gamma^2 \epsilon m}{4}\right). \qquad (*)$$

(ii) *If the VC dimension of H is d, then for any $0 < \delta < 1$, the quantity $(*)$ is at most δ for any sample size m greater than*

$$max\left(\frac{8}{\gamma^2 \epsilon} ln \frac{8}{\delta}, \frac{16d}{\gamma^2 \epsilon} ln \frac{16}{\gamma^2 \epsilon}\right).$$

PROOF. For part (i), let $R = \{$*disagree*(h): $h \in H\}$. We claim that $\Pi_R(m) = \Pi_H(m)$ for all $m \geq 1$. The argument is similar to that used in the proof of Lemma A2.3. Choose any $S \subseteq Y$ with $|S| = m$. Let $T = \{x \in X : (x, a) \in S$ for some $a \in \{0, 1\}\}$. Since $\Pi_R(S)$ is maximal when $|T| = m$, we assume this is the case. It is clear that for any $h_1, h_2 \in H$, $h_1 \cap T = h_2 \cap T$ if and only if *disagree*(h_1) $\cap S =$ *disagree*(h_2) $\cap S$. From this it follows that $\Pi_R(m) = \Pi_H(m)$. The result then follows directly from Corollary A3.1.

The calculation of the bound in part (ii) is similar to the calculation in Lemma A2.4, using Proposition A2.1(i) and (iii). □

Letting $\gamma = \frac{1}{2}$, the above result shows that if a learning algorithm takes a random sample $\tilde{y}$ of size

$$\max\left(\frac{32}{\epsilon} \ln \frac{8}{\delta}, \frac{64d}{\epsilon} \ln \frac{64}{\epsilon}\right)$$

and only entertains hypotheses $h \in H$ with empirical error $\hat{er}_{\tilde{y}}(h) \leq \epsilon/2$, then the probability that it returns a hypothesis h with actual error $er(h)$ more than ϵ is at most δ. This is true for any stochastic target concept. By letting $\gamma = 1$ and restricting ourselves to deterministic target concepts chosen from H, we can use Theorem A3.1 to show that any consistent function $A: S_H \rightarrow H$ is a learning function for H using sample size

$$\max\left(\frac{8}{\epsilon} \ln \frac{8}{\delta}, \frac{16d}{\epsilon} \ln \frac{16}{\epsilon}\right).$$

Thus, Theorem 2.1(ii)(a) is almost a special case of Theorem A3.1, but for the fact that its simpler proof yields slightly better constants.

ACKNOWLEDGMENTS. We thank Shai Ben-David for pointing out an error in our original definition of well-behaved classes and in Lemma A1.1, and for suggesting

the versions used here. We would like to thank Emo Welzl for many helpful discussions and ideas on ϵ-transversals and classes of finite VC dimension, and for pointing out [6] to us. The bound on the VC dimension of s-gons given in Example 3.2.2 is also due to him. Thanks to Leen Stougie for making us aware of [61] and to Nick Littlestone for numerous important observations, especially regarding r-poly hy-fi's and Theorem 3.1.1. Les Valiant, Dana Angluin, Lenny Pitt, Jan Mycielski, John Cherniavsky, Janet Blumer, Herbert Edelsbrunner, and Sally Floyd also provided valuable suggestions at various stages of this investigation.

REFERENCES

1. AHO, A., HOPCROFT, J., AND ULLMAN, J. *The Design and Analysis of Computer Algorithms.* Addison-Wesley, London, 1974.
2. ANGLUIN, D. Learning regular sets from queries and counterexamples. *Inf. Comput. 75* (1987), 87–106.
3. ANGLUIN, D. Queries and concept learning. *Mach. Learning 2*, 2 (1988), 319–342.
4. ANGLUIN, D., AND LAIRD, P. D. Learning from noisy examples. *Mach. Learning 2*, 2 (1988), 343–370.
5. ANGLUIN, D., AND VALIANT, L. Fast probabilistic algorithms for Hamiltonian circuits and matchings. *J. Comput. Syst. Sci. 19* (1979), 155–193.
6. ASSOUAD, P. Densité et Dimension. *Ann. Inst. Fourier, Grenoble 33*, 3 (1983), 233–282.
7. BAUM, E., AND HAUSSLER, D. What size net gives valid generalization. *Neural Comput. 1*, 1 (1989) 151–160.

7a. BEN-DAVID, S., BENEDEK, G., AND MANSOUR, Y. A parameterization scheme for classifying models of learnability. In *Proceedings of the 2nd Workshop of Computational Learning Theory* (Santa Cruz, Calif., July 31–Aug. 2). Morgan Kaufman, San Mateo, Calif., 1989, to appear.

8. BENEDEK, G., AND ITAI, A. Nonuniform learnability. In *Proceedings of the 15th International Conference on Automata, Languages and Programming* (July). 1988, pp. 82–92.
9. BLUM, A., AND RIVEST, R. Training a 3-node neural network is NP-complete. In *Proceedings of the 1st Workshop on Computational Learning Theory* (Cambridge, Mass., Aug. 3–5). Morgan Kaufmann, San Mateo, Calif., 1988, pp.9–18.
10. BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., AND WARMUTH, M. K. Occam's Razor. *Inf. Process. Lett. 24* (1987), 377–380.
11. BOLLOBÁS, B. *Combinatorics.* Cambridge Univ. Press, Cambridge, Mass., 1986.
12. BOUCHERON, S., AND SALLANTIN, J. Some remarks about space-complexity of learning and circuit complexity of recognizing. In *Proceedings of the 1st Workshop on Computational Learning Theory* (Cambridge, Mass., Aug. 3–5). Morgan Kaufmann, San Mateo, Calif., 1988, pp. 125–138.
13. COVER, T. Geometrical and statistical properties of systems of linear inequalities with applications to pattern recognition. *IEEE Trans. Elect. Comp. 14* (1965), 326–334.
14. DEVROYE, L. P. Automatic pattern recognition: A study of the probability of error. *IEEE Trans. Pattern Analysis and Mach. Intell. 10*, 4 (1988), 530–543.
15. DEVROYE, L. P., AND WAGNER, T. J. A distribution-free performance bound in error estimation. *IEEE Trans. Inf. Theorem 22* (1976), 586–587.
16. DUDA, R., AND HART, P. *Pattern Classification and Scene Analysis.* Wiley, New York, 1973.
17. DUDLEY, R. M. A course on empirical processes. In *Lecture Notes in Mathematics*, vol. 1097. Springer-Verlag, New York, 1984.
18. DUDLEY, R. M. Universal Donsker classes and metric entropy. *Ann. Prob. 15*, 4 (1987), 1306–1326.
19. EDELSBRUNNER, H., AND PREPARATA, F. P. Minimal polygonal separation. *Inf. Comput. 77* (1988), 218–232.
20. EHRENFEUCHT, A., HAUSSLER, D., KEARNS, M., AND VALIANT, L. G. A general lower bound on the number of examples needed for learning. *Inf. Comput.*, to appear.
21. GAREY, M., AND JOHNSON, D. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, San Francisco, 1979.
22. GILL, J. Probabilistic Turing machines. *SIAM J. Comput. 6*, 4 (1977), 675–695.
23. GINÉ, E., AND ZINN, J. Lectures on the central limit theorem for empirical processes. In *Lecture Notes in Mathematics*, vol. 1221. Springer-Verlag, New York, 1986.
24. GOLDREICH, O., GOLDWASSER, S., AND MICALI, S. How to construct random functions. *J. ACM 33*, 4 (Oct. 1986), 792–807.
25. HAMPSON, S. E., AND VOLPER, D. Linear Function neurons: Structure and training. *Biol. Cyber. 53* (1986), 203–217.

26. HAUSSLER, D. Learning conjunctive concepts in structural domains. *Mach. Learning 4*, 1 (1989)
27. HAUSSLER, D. Applying Valiant's learning framework to AI concept learning problems. In *Proceedings of the 4th International Workshop on Machine Learning.* Morgan Kaufmann, San Mateo, Calif., 1987, pp. 324–336.
28. HAUSSLER, D. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artif. Intell. 36* (1988), 177–221.
28a. HAUSSLER, D. Generalizing the PAC model: Sample size bounds from metric dimension-based uniform convergence results. In *Proceedings of the 30th IEEE Symposium on Foundations of Computer Science* (Research Triangle Park, N.C., Oct. 30–Nov. 1). IEEE, New York, 1989, to appear.
29. HAUSSLER, D., AND WELZL, E. Epsilon-nets and simplex range queries. *Disc. Comput. Geometry 2* (1987), 127–151.
30. HAUSSLER, D., KEARNS, M., LITTLESTONE, N., AND WARMUTH, M. K. Equivalence of models for polynomial learnability. Tech. Rep. UCSC-CRL-88-06. Univ. California at Santa Cruz, Santa Cruz, Calif., 1988.
31. HAUSSLER, D., LITTLESTONE, N., AND WARMUTH, M. K. Predicting {0, 1}-functions on randomly drawn points. In *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science* (White Plains, N.Y., Oct.). IEEE, New York, 1988, pp. 100–109.
32. JOHNSON, D. S. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci. 9* (1974), 256–278.
33. KARMARKAR, N. A new polynomial-time algorithm for linear programming. *Combinatorica 4* (1984), 373–395.
34. KEARNS, M., AND LI, M. Learning in the presence of malicious errors. In *Proceedings of the 20th Symposium on Theory of Computing* (Chicago, Ill., May 2–4). ACM, New York, 1988, pp. 267–280.
35. KEARNS, M., AND VALIANT, L. Cryptographic limitations on learning Boolean formulae and finite automata. In *Proceedings of the 21st ACM Symposium on Theory of Computing* (Seattle, Wash., May 15–17). ACM, New York, 1989, pp. 433–444.
36. KEARNS, M., LI, M., PITT, L., AND VALIANT, L. On the learnability of Boolean formulae. In *Proceedings of the 19th ACM Symposium on Theory of Computation* (New York, N.Y., May 25–27). ACM, New York, 1987, pp. 285–295.
37. KEARNS, M., LI, M., PITT, L., AND VALIANT, L. Recent results on Boolean concept learning. In *Proceedings of the 4th International Workshop on Machine Learning.* Morgan-Kaufmann, San Mateo, Calif., 1987, pp. 337–352.
38. KNUTH, D. E. *The Art of Computer Programming*, 2nd ed., vol. 1. Addison-Wesley, Reading, Mass., 1973.
39. LAIRD, P. D. Learning from good data and bad. Tech. Rep. YALEU/DCS/TR-551. Yale Univ., New Haven, Conn., 1987.
40. LEE, D. T., AND PREPARATA, F. P. Computational geometry—A survey. *IEEE Trans. Comput. 33*, 12 (1984), 1072–1101.
41. LINIAL, N., MANSOUR, Y., AND RIVEST, R. Results on learnability and the Vapnik–Chervonenkis dimension. In *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science* (White Plains, N.Y., Oct.). IEEE, New York, 1988, pp. 120–129.
42. LITTLESTONE, N. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Mach. Learning 2*, 2 (1988), 285–318.
43. MASEK, W. J. Some NP-complete set cover problems. MIT Laboratory for Computer Science, unpublished manuscript.
44. MEGIDDO, N. Linear programming in linear time when the dimension is fixed. *J. ACM 31*, 1 (Jan. 1984), 114–127.
45. MEGIDDO, N. On the complexity of polyhedral separability. *Discrete Comput. Geometry 3* (1988), 325–337.
46. MUROGA, S. *Threshold Logic and Its Applications.* Wiley, New York, 1971.
47. NATARAJAN, B. K. On learning Boolean functions. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computation* (New York, N.Y., May 25–27). ACM, New York, 1987, pp. 296–304.
48. NATARAJAN, B. K. Learning functions from examples. Tech. Rep. CMU-RI-TR-87-19. Carnegie Mellon Univ., Pittsburgh, Pa., Aug. 1987.
49. NIGMATULLIN, R. G. The fastest descent method for covering problems (in Russian). In *Proceedings of a Symposium on Questions of Precision and Efficiency of Computer Algorithms*, Book 5. Kiev, 1969, pp. 116–126.

50. PEARL, J. On the connection between the complexity and credibility of inferred models. *Int. J. Gen. Syst. 4* (1978), 255–264.
51. PEARL, J. Capacity and error estimates for Boolean classifiers with limited capacity. *IEEE Trans. Pattern Analysis Mach. Intell. 1*, 4 (1979), 350–355.
52. PITT, L., AND VALIANT, L. G. Computational limitations on learning from examples. *J. ACM 35*, 4 (Oct. 1988), 965–984.
53. PITT, L., AND WARMUTH, M. Reductions among prediction problems, on the difficulty of predicting automata. In *Proceedings of the 3rd IEEE Structure in Complexity Theory Conference* (Washington, D.C.). IEEE, New York, 1988, pp. 62–69.
54. POLLARD, D. *Convergence of Stochastic Processes.* Springer-Verlag, New York, 1984.
55. QUINLAN, R., AND RIVEST, R. Inferring decision trees using the minimum description length principle. *Inf. Comput.*, to appear.
56. RIVEST, R. Learning decision-lists. *Mach. Learning 2*, 3 (1987), 229–246.
57. ROYDEN, H. L. *Real Analysis*, 2nd ed. MacMillan, New York, 1968.
58. SLOAN, R. Types of noise in data for concept learning. In *Proceedings of the 1st Workshop on Computational Learning Theory* (Cambridge, Mass., Aug. 3–5). Morgan Kaufmann, San Mateo, Calif., 1988, pp. 91–96.
59. VALIANT, L. G. A theory of the learnable. *Commun. ACM 27*, 11 (Nov. 1984), 1134–1142.
60. VALIANT, L. G. Learning disjunctions of conjunctions. In *Proceedings of the 9th International Conference on Artificial Intelligence* (Los Angeles, Calif., Aug.), vol. 1. Morgan Kaufmann, San Mateo, Calif., 1985, pp. 560–566.
61. VAPNIK, V. N. *Estimation of Dependences Based on Empirical Data.* Springer Verlag, New York, 1982.
62. VAPNIK, V. N., AND CHERVONENKIS, A. YA. On the uniform convergence of relative frequencies of events to their probabilities. *Theoret. Probl. and Its Appl. 16*, 2 (1971), 264–280.
63. VAPNIK, V. N., AND CHERVONENKIS, A. YA. *Theory of Pattern Recognition* (in Russian). Nauka, Moscow, 1974.
64. VITTER, J., AND LIN, J. H. Learning in parallel. In *Proceedings of the 1st Workshop on Computational Learning Theory* (Cambridge, Mass., Aug. 3–5). Morgan Kaufmann, San Mateo, Calif., 1988, pp. 106–124.
65. WATANABE, S. Pattern recognition as information compression. In *Frontiers of Pattern Recognition*, S. Watanabe, Ed. Academic Press, Orlando, Fla., 1972.
66. WENOCUR, R. S., AND DUDLEY, R. M. Some special Vapnik–Chervonenkis classes. *Discrete Math. 33* (1981), 313–318.
67. WINSTON, P. Learning structural descriptions from examples. In *The Psychology of Computer Vision.* McGraw-Hill, New York, 1975.

RECEIVED MARCH 1986; REVISED NOVEMBER 1987 AND OCTOBER 1988; ACCEPTED NOVEMBER 1988