Noise-Tolerant Distribution-Free Learning of General Geometric Concepts

NADER H. BSHOUTY

Technion, Haifa, Israel

SALLY A. GOLDMAN

Washington University, St. Louis, Missouri

H. DAVID MATHIAS

Ohio State University, Columbus, Ohio

SUBHASH SURI

Washington University, St. Louis, Missouri

AND

HISAO TAMAKI

IBM Tokyo Research Laboratory, Yamato, Japan

Abstract. We present an efficient algorithm for PAC-learning a very general class of geometric concepts over $\Re^d$ for fixed d. More specifically, let $\mathcal{T}$ be any set of s halfspaces. Let $x = (x_1, \ldots,$

A preliminary version of this work appeared in *Proceedings of the 28th Annual ACM Symposium on the Theory of Computing* (Philadelphia, Pa., May 22–24). ACM, New York, pp. 151–160.

S. A. Goldman would like to dedicate this paper to Paris Kanellakis, who first introduced her to computer science research.

This research was carried out while D. Mathias was at Washington University and N. Bshouty was at University of Calgary, Calgary, Canada and supported in part by NSERC of Canada.

S. Goldman and D. Mathias were supported in part by National Science Foundation (NSF) NYI grant CCR 93-57707 with matching funds provided by Xerox PARG and WUTA.

S. Suri was supported in part by NSF grant CCR 05-01494.

Authors' addresses: N. H. Bshouty, Computer Science Department, Technion, Technion City, Haifa, 32000 Israel, bshouty@csa.technion.ac.il; S. A. Goldman and S. Suri, Department of Computer Science, Washington University, One Brookings Drive (Campus Box 1045), St. Louis, MO 63130-4899, {sg, suri}@cs.wustl.edu; H. D. Mathias, Department of Computer and Information Science, Ohio State University, 395 Dreese Lab, 2015 Neil Avenue, Columbus, OH 43210, dmath@cis.ohio-state.edu; and H. Tamaki, IBM Tokyo Research Laboratory, Yamamoto, 242, Japan, htmaki@trl.ibm.co.jp.

x_d) be an arbitrary point in $\mathcal{R}^d$. With each $t \in \mathcal{T}$ we associate a boolean indicator function $I_t(x)$ which is 1 if and only if x is in the halfspace t. The concept class, $\mathcal{C}_s^d$, that we study consists of all concepts formed by any Boolean function over $I_{t_1}, \ldots, I_{t_s}$ for $t_i \in \mathcal{T}$. This class is much more general than any geometric concept class known to be PAC-learnable. Our results can be extended easily to learn efficiently any Boolean combination of a polynomial number of concepts selected from any concept class $\mathcal{C}$ over $\mathcal{R}^d$ given that the VC-dimension of $\mathcal{C}$ has dependence only on d and there is a polynomial time algorithm to determine if there is a concept from $\mathcal{C}$ consistent with a given set of labeled examples. We also present a statistical query version of our algorithm that can tolerate random classification noise. Finally we present a generalization of the standard ϵ-net result of Haussler and Welzl [1987] and apply it to give an alternative noise-tolerant algorithm for $d = 2$ based on geometric subdivisions.

Categories and Subject Descriptors: F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Computational learning, geometric concepts

1. *Introduction*

We present an efficient algorithm for PAC-learning a broad class of geometric concepts over $\mathcal{R}^d$ for fixed d against arbitrary distributions. More specifically, given a set $\mathcal{T}$ of s halfspaces, associate a boolean indicator function $I_t(\cdot)$, with each $t \in \mathcal{T}$, such that, for an arbitrary point $x = (x_1, \ldots, x_d)$ in $\mathcal{R}^d$, $I_t(x) = 1$ if and only if $x \in t$. The concept class, $\mathcal{C}_s^d$, that we study consists of all concepts formed by any boolean function over $I_{t_1}, \ldots, I_{t_s}$ for $t_i \in \mathcal{T}$ where $\mathcal{T}$ is any set of s halfspaces. In other words, $\mathcal{C}_s^d$ consists of all boolean labelings of the $O(s^d)$ subspaces created by any s halfspaces. (See Figure 1 for an illustration.) While the time complexity of our algorithm has exponential dependence on d, the sample complexity is polynomial in d and s.

The concept class we study is more general than any geometric concept class known to be PAC-learnable. (See Section 5 for a discussion of related work.) Special cases of our main theorem result in learning algorithms for several new geometric concept classes. For example, the concept class defined by the parity function over the indicator variables of s hyperspaces is learnable by our algorithm. While we consider geometric concepts defined by halfspaces, any polyhedron (not necessarily convex) defined by f faces can be formed by combining f halfspaces. Thus, our algorithm can also learn any Boolean combination of a polynomial number of polyhedra each with a polynomial number of faces. Even further, our results can be easily extended to learn efficiently any Boolean combination of a polynomial number of concepts selected from any concept class $\mathcal{C}$ over $\mathcal{R}^d$ given that the VC-dimension of $\mathcal{C}$ has dependence only on d (and is thus constant for any constant d), and there is a polynomial time algorithm to determine if there is a concept from $\mathcal{C}$ consistent with a given set of labeled examples. Standard techniques cannot be applied to learn many such complex geometric concepts and, in fact, prior to this work, no algorithms were known for these problems.

In addition to presenting a polynomial time algorithm to learn $\mathcal{C}_s^d$ (for d constant), we present a variation of our algorithm that can tolerate random noise

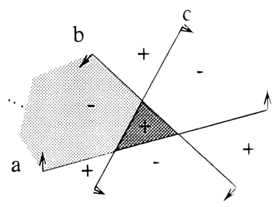

FIG. 1. A target concept from $\mathscr{C}_3^2$ formed from the three halfspaces a, b, and c. For any point x in the lightly shaded region $I_a(x) = I_b(x) = 1$ and $I_c(x) = 0$. Likewise, for any point x' in the darkly shaded region $I_a(x') = I_b(x') = I_c(x') = 1$. The Boolean function over I_a, I_b, I_c that is illustrated in this figure is the parity function. Namely, x is a positive point exactly when an odd number of the indicator functions are 1 for x. Observe that a positive point in the darkly shaded region cannot be separated from all negative points by a single hyperplane.

in the labels for any noise rate strictly less than 1/2. This variation takes advantage of the noise tolerance inherent in the statistical query (SQ) model [Kearns 1993]. Finally we present a generalization of the standard ϵ-net result of Haussler and Welzl [1987] and apply it to give an alternative noise-tolerant algorithm for $d = 2$ based on geometric subdivisions.

2. *Limitations of the Standard Covering Technique*

Considerable work has been done on learning geometric concepts in the PAC model. To illustrate the key technique used in most of this work, consider the problem of learning unions of s halfspaces in d-dimensional space for any constant d [Blumer et al. 1989; Baum 1990a; Brönnimann and Goodrich 1994].[1] The standard Occam algorithm draws a sufficiently large sample S of m points (where m is chosen to satisfy the bound of Blumer et al. [1989]) and then finds a hypothesis consistent with the sample by formulating a set covering problem. For the learning problem, one first generates a set $\mathscr{F}$ containing $O(m^d)$ halfspaces such that each possible partition of the points in S into two sets by a halfspace is realized. Note that, when generating $\mathscr{F}$, the labels of the examples are ignored. During the covering phase (described below), the labels are used. After generating $\mathscr{F}$, the learner initializes the hypothesis h to be the always false hypothesis, and P to be all positive points from S. Then the learner repeats the following step until $P = \emptyset$:

Find a halfspace $f \in \mathscr{F}$ consistent with all of the negative examples in S and the maximum number of examples from P. Replace h by $h \cup f$ and remove from P the points correctly classified by f.

Since the target concept gives a covering of size s, it follows that the greedy set covering algorithm [Chvatal 1979] produces a cover of size $O(s \ln|P|) = O(s \lg m)$. Thus the representation size of the final hypothesis is sub-linear in m as required by the result of Blumer et al. [1989].

To see a fundamental limitation of the standard technique, consider the geometric concept shown in Figure 1. Notice that there is no hyperplane that separates a positive point in the darkly shaded region from all negative points.

[1] By a dual argument, everything discussed applies to intersections of halfspaces.

When the target is not a union or an intersection of halfspaces then it is not clear what the right notion of covering is, or if it is applicable at all.

This standard covering technique can be applied to learn unions (and intersections) of base classes, $\mathscr{C}$, other than halfspaces. However, for a concept class $\mathscr{C}$, $|\mathscr{F}|$ has an exponential dependence on the VC-dimension of $\mathscr{C}$. Thus, this approach yields a polynomial-time algorithm only when the VC-dimension of $\mathscr{C}$ is constant. Blumer et al. [1989] prove that this set covering approach can be used to PAC-learn the union (or intersection) of concepts from a base class $\mathscr{C}$ efficiently given that the VC-dimension of $\mathscr{C}$ is constant and there is a polynomial time algorithm to determine if there is a concept from $\mathscr{C}$ consistent with a given set of points. Our technique, introduced here, generalizes this standard technique by allowing us to learn classes defined by arbitrary Boolean combinations of concepts from $\mathscr{C}$ versus just intersections or unions.

If one tries to apply the standard technique to Boolean combinations more general than union or intersection, then the fundamental problem illustrated in Figure 1 returns. Although for some base classes, such as axis-aligned boxes, any single positive example can be separated from all the negative regions by a box–it is essential to argue that there exists a covering whose size is sublinear in m. When learning a union, it is easily seen that each of the s concepts in $\mathscr{C}$ used to form the target concept correctly classify *all* negative examples. Thus those s concepts form a covering. However, when we consider more complex geometric concepts (i.e., concepts for which the combination of the indicator functions is not simply a union or intersection), a small covering of this form may not exist. Thus, the standard techniques cannot be applied. As another example illustrating the limitations of the known techniques, consider the class defined by the union of a polynomial number of convex polyhedra each with a polynomial number of faces. Here the standard technique would fail to give an algorithm whose time complexity is polynomial in the total number of faces in the polyhedra defining the target concept because the VC-dimension of a polyhedron depends on the number of faces.

3. *Our Contributions*

In this paper, we describe several contributions. First, we give an algorithm to learn $\mathscr{C}_s^d$ in the PAC model.[2] Our work introduces a new learning technique that uses the following novel, though simple, application of set covering. While (as previously pointed out) it is not true in general that every positive example in the sample must be separated from *all* negative examples by a *single* hyperplane "in" the target, it is true that every positive example must be separated from *each* negative example by some hyperplane (and so it should be separated from all negative points by a subset of these hyperplanes). Alternatively, we must ensure that for each positive example and negative example, there is a hyperplane in the hypothesis that separates (covers) them. To achieve this goal, we construct from the sample a complete bipartite graph by adding an edge between every pair of

[2] While we use $\mathscr{C}_s^d$ to illustrate our technique, the result generalizes to any Boolean combination of a polynomial number of concepts selected from any concept class $\mathscr{C}$ over $\mathscr{R}^d$ given that the VC-dimension of $\mathscr{C}$ depends only on d (and thus is constant for d constant), and there is a polynomial-time algorithm to determine if there is a concept from $\mathscr{C}$ consistent with a given set of labeled examples.

points $\langle x_+, x_- \rangle$ where x_+ is positive and x_- is negative. The edges in this graph form the set $\mathcal{U}$ to be covered. The family of subsets of $\mathcal{U}$ is comprised of hyperplanes where a hyperplane covers every edge that it properly intersects. Since the VC-dimension of a hyperplane is $d + 1$ (which is constant for fixed d), $|\mathcal{F}|$ is polynomial in the size of the sample.[3] By applying a greedy covering technique, we obtain a set of hyperplanes that separates every pair $\langle x_+, x_- \rangle$. By using the set covering algorithm of Brönnimann and Goodrich [1994] (versus the standard greedy covering) we obtain a cover of size $O(sd \lg(sd))$. Our approach does not have the limitations discussed above, yet maintains the desirable feature that the sample complexity is polynomial in s and d. It is only the time complexity that depends exponentially on d. This contrasts recent results described in Section 5. Furthermore, notice that the hyperplanes that define the *target concept* divide $\mathcal{R}^d$ into *regions* (or connected components) that are individually labeled. (As an example the target concept shown in Figure 1 has 3 hyperplanes that divide $\mathcal{R}^2$ into 7 regions.) Since the number of different regions in the *target concept* could depend exponentially on d, the size of the target concept itself could be exponential in d. Thus, with the exception of some subclasses of $\mathcal{C}_s^d$ with polynomially sized concepts, our algorithm runs in time polynomial in the size of the target. Note, we can use parallel set covering techniques [Berger et al. 1994] to get our algorithm to run efficiently in parallel.

A second contribution of our work is the conversion of our basic algorithm to a statistical query algorithm giving noise tolerance. Due to our covering approach, the statistical queries we make come from a modified distribution (that we simulate from our given example source). This modification of the distribution causes the desired accuracy that the learner must reach to get smaller with each step of the covering. Thus, the conversion to a statistical query algorithm is more complex than that needed with the standard covering approach.

Finally, we present a generalization of the standard ϵ-net result of Haussler and Welzl [1987] and apply it to give an alternative noise-tolerant algorithm for $d = 2$ based on geometric subdivisions.

4. *Preliminaries*

The learning model we use in this work is the *probably approximately correct* (PAC) model of Valiant [1984]. In this model, the learner is presented with examples, chosen randomly from *instance space* $\mathcal{X}$ according to unknown probability distribution $\mathcal{D}$. Let f be an unknown target function from known concept class $\mathcal{C}$. The learner must return a hypothesis h that classifies at least $(1 - \epsilon)$ of $\mathcal{X}$ consistent with f, with probability at least $(1 - \delta)$, where ϵ and δ are given constants. That is, with high probability, the hypothesis must correctly classify most of the instances (by weight under distribution $\mathcal{D}$). We now describe several relevant results relating learnability and the Vapnik-Chervonenkis (VC) dimension, an important complexity measure. As is commonly done, we treat a concept as a subset of the instance space that contains exactly the positive examples. When speaking of a concept as a function, we use the corresponding indicator

[3] Applying a result from Blumer et al [1989] gives that for classes $\mathcal{C}$ with finite VC-dimension and a polynomial time consistency algorithm, the elements of $\mathcal{F}$ can be listed in polynomial time. While this result applies for $\mathcal{C}_s^d$, in Section 6 we describe a more efficient approach based on a geometric duality argument.

function that evaluates to 1 for an example in the concept and 0 for an example not in the concept.

The paper of Blumer et al. [1989] identifies a combinatorial parameter of a class of hypotheses called the *Vapnik-Chervonenkis* (*VC*) *dimension*, which originated in the paper of Vapnik and Chervonenkis [1971]. The VC-dimension characterizes the sample size required in order to have enough information for accurate generalization. The VC dimension of concept class $\mathcal{C}$ (which we denote VCD($\mathcal{C}$)) is the size of a largest set $S \subseteq \mathcal{X}$ such that any subset of S is of the form $S \cap C$, for some $C \in \mathcal{C}$, or ∞ if such sets can be arbitrarily large. In other words, it is the cardinality of the largest set S of examples where for *each* of the $2^{|S|}$ labelings (or subsets) of S, there is a concept in $\mathcal{C}$ that gives that labeling.

The results of Blumer et al. [1989] give a sufficient condition for a prediction algorithm to generalize successfully from example data, in terms of the VC dimension. Namely, they prove the following result.

THEOREM 1 [BLUMER ET AL. 1989]. *Let A be a learning algorithm for concept class $\mathcal{C}$ that has hypothesis space $\mathcal{H}$. Let d be the VC-dimension of $\mathcal{H}$. Any concept $f \in \mathcal{H}$ consistent with a sample of size max* $(4/\epsilon)log(2/\delta)(8d/\epsilon)log(13/\epsilon)$ *will have error at most ϵ with probability at least* $1 - \delta$.

Furthermore, Ehrenfeucht and Haussler [1989] prove that any concept class $\mathcal{C}$ of VC dimension d must use $\Omega(1/\epsilon)\log(1/\delta) + (d/\epsilon)$ examples in the worst case.

One drawback with the above approach is that the hypothesis must be drawn from a fixed hypothesis class $\mathcal{H}$. In particular, the complexity of the hypothesis class cannot depend on the size of the sample. However, when using a set covering approach, the size of the hypothesis often depends on the size of the sample as well as the target complexity (as defined by the number of bits needed to represent the target concept). Hence, to reflect the dependence of the results on the target complexity size and sample size, let $\mathcal{H}^A_{s,m}$ be the hypothesis space used by algorithm A for a target of complexity s and sample size m. Blumer et al. [1987, 1989] show that finding a hypothesis whose size is sublinear in the sample size is sufficient to guarantee polynomial PAC learnability. More formally, we say that algorithm A is an *Occam Algorithm* for concept class $\mathcal{C}$ if there exists a polynomial $p(s)$ and a constant α, $0 \leq \alpha < 1$, such that for all s, $m \geq 1$, the VC dimension of $\mathcal{H}^A_{s,m}$ is at most $p(s)m^{\alpha}$. Since the VC dimension of our hypothesis class grows like poly(s) and polylog(m), we can apply the following result of Blumer et al. [1989].

THEOREM 2 [BLUMER ET AL. 1989]. *Let A be a learning algorithm for concept class $\mathcal{C}$ that has hypothesis space $\mathcal{H}^A_{s,m}$ where s is the target complexity. If the VC dimension of $\mathcal{H}^A_{s,m}$ is at most $p(s)(lg\ m)^{\ell}$ for some polynomial* $p(s) \geq 2$ *and* $\ell \geq 1$, *then A is a PAC-learning algorithm for $\mathcal{C}$ using sample size*

$$m = max\left(\frac{4}{\epsilon}\, lg\,\frac{2}{\delta},\ \frac{2^{\ell+4}p(s)}{\epsilon}\left(lg\frac{8(2\ell+2)^{\ell+1}p(s)}{\epsilon}\right)^{\ell+1}\right).$$

The basic PAC model assumes that the examples given to the learner are drawn randomly from $\mathcal{D}$ and labeled *correctly* based on the target concept. In this work we also consider a variant of the PAC model in which the labeled examples that the learner receives are corrupted by *random classification noise* [Angluin and Laird 1988]. In this noise model, each example is still drawn at random from $\mathcal{D}$. However, with probability η (where $0 \leq \eta < 1/2$ is called the *noise rate*), the learner receives the incorrect label, and with probability $1 - \eta$, the learner receives the correct label. Thus, the example drawn is labeled incorrectly, at random, with probability η. In the *malicious classification noise* model [Sloan 1995], each example is still drawn at random from $\mathcal{D}$. Here, with probability η, an adversary can select the label. In the *malicious noise* model [Valiant 1985], with probability η the adversary can provide an example and label of its choice.

To obtain a noise-tolerant algorithm, we use the *statistical query* model introduced by Kearns [1993] and further studied by Decatur [1993] and Aslam and Decatur [1993, 1995]. In this model, rather than sampling labeled examples, the learner requests the value of various statistics on the distribution from an oracle. A statistical query oracle returns the probability, within some additive constant, that some predicate is true relative to the distribution. The particular queries we use are known as *relative statistical queries* [Aslam and Decatur 1995]. These take the form rel-$\text{STAT}_{\mathcal{D}}(\chi, \mu, \theta)$ where χ is a predicate over labeled examples drawn from $\mathcal{D}$, μ is the relative error bound, and θ is the threshold. For target function f, let $P_\chi = \Pr_{\mathcal{D}}[\chi(x, f(x)) = 1]$. If $P_\chi < \theta$ then rel-$\text{STAT}_{\mathcal{D}}(\chi, \mu, \theta)$ may return $\perp$. If $\perp$ is not returned, then rel-$\text{STAT}_{\mathcal{D}}(\chi, \mu, \theta)$ must return an estimate $\hat{P}_\chi$ such that $P_\chi(1 - \mu) \leq \hat{P}_\chi \leq P_\chi(1 + \mu)$. The learner may also request unlabeled examples. Aslam and Decatur [1995] have shown that all relative error SQ algorithms are robust against random classification noise for any noise rate $\eta < 1/2$.

THEOREM 3 [ASLAM AND DECATUR 1995]. *The total sample complexity required to simulate a relative SQ algorithm in the presence of classification noise at a rate of $\eta \leq \eta_b < 1/2$ is*

$$\tilde{O}\left(\frac{VCD(\mathcal{Q}) + \log(1/\delta)}{\mu_*^2\theta_*\rho(1 - 2\eta_b)^2}\right),$$

where μ_* (respectively, θ_*) is the minimum value of μ (respectively, θ) across all queries and $\rho \in [\theta_*, 1]$. (The soft-oh notation ($\tilde{O}$) is similar to the standard big-oh notation except low order factors are also removed.)

They have also shown that relative error SQ algorithms are robust against small amounts of malicious errors.

THEOREM 4 [ASLAM AND DECATUR 1995]. *The total sample complexity required to simulate a relative SQ algorithm in the presence of malicious noise with error rate $\beta \leq \mu_*\theta_*/32$ is*

$$O\left(\frac{VCD(\mathcal{Q})}{\mu_*^2\theta_*}\log\frac{1}{\mu_*\theta_*} + \frac{1}{\mu_*^2\theta_*}\log\frac{1}{\delta}\right),$$

where μ_* (respectively, θ_*) is the minimum value of μ (respectively, θ) across all queries.

Finally, when converting an SQ algorithm to a PAC algorithm, the time required to evaluate each statistical query is linear in the sample size.

Let $\mathscr{X}$ be the set of integers. Let $\vec{y} = (y_1, \ldots, y_d)$ denote the d dimension variables and $\vec{x} = (x_1, \ldots, x_d)$ denote an element of $[0, 1]^d$, the instance space. Let $\vec{a} = (a_1, \ldots, a_d)$ be a vector where $a_i \in \mathscr{X}$. A d-dimensional *hyperplane* is $\{\vec{y}|\vec{a} \cdot \vec{y} = b\}$ for some $b \in \mathscr{X}$. A d-dimensional *halfspace* is $\{\vec{y}|\vec{a} \cdot \vec{y} \succ b\}$ where $\succ \in \{>, \geq, <, \leq\}$ and b is a constant. Let H be a finite set of hyperplanes in $[0, 1]^d$. The *arrangement* $\mathscr{A}(H)$ is the decomposition of $[0, 1]^d$ into features of dimension k, $0 \leq k \leq d$. We define a region as a d-dimensional feature in $\mathscr{A}(H)$. For each set S of sample points, we say that two hyperplanes p_i and p_j are equivalent, with respect to S, if the partition of S imposed by p_i is identical to the partition of S imposed by p_j.

An instance of the set covering problem is a ground set $\mathscr{U}$ and a family $\mathscr{F}$ of subsets of $\mathscr{U}$ such that $\bigcup_{f \in \mathscr{F}} f = \mathscr{U}$. A solution is a smallest cardinality subset $\mathscr{G}$ of $\mathscr{F}$ such that $\bigcup_{g \in \mathscr{G}} g = \mathscr{U}$. The greedy approximation algorithm [Chvatal 1979] for this problem has a ratio bound of $\ln|\mathscr{U}| + 1$ on the size of the approximation.

5. *Previous Work*

In this section, we highlight some of the relevant learning results for geometric concepts. There have been many results for the classes of unions and intersections of halfspaces. Blum and Rivest [1989] show that there does not exist an efficient proper[4] learning algorithm for unions of s halfspaces, unless $RP = NP$. (That is, no such algorithm can have polynomial dependence on d.) They also give an algorithm to PAC-learn the xor of two halfspaces by transforming the examples so that positive and negative points are linearly separable. Baum [1990a] gives an algorithm that efficiently learns a union of s halfspaces in constant dimensional space. Blumer et al. [1989] give a similar result. Both algorithms return hypotheses containing $O(s \lg m)$ halfspaces where m is the size of the sample. Brönnimann and Goodrich [1994] present a set covering algorithm that allows them to return a hypothesis containing $O(ds \lg(ds))$ halfspaces. Baum gives efficient algorithms for learning several classes with infinite VC-dimension (such as convex polyhedral sets) under uniform distributions [Baum 1990b]. Haussler [1989] also gives distribution specific algorithms for several classes of functions. Bshouty et al. [1998] have given noise-tolerant algorithms for several geometric classes. In particular, they studied $\mathscr{C}_s^d$ against the product distribution and a restricted version of this class, in which the hyperplanes have slopes from a set of r known slopes, against arbitrary distributions. They also considered a non-linear geometric class against the product distribution. All of these algorithms have time and sample complexity exponential in the number of dimensions.

The learnability of unions of axis-parallel boxes has also been considered. Blumer et al. [1989] present an algorithm to PAC-learn an s-fold union of boxes in E^d by drawing a sufficiently large sample of size $m = poly((1/\epsilon), lg(1/\delta), s, d)$, and then performing a greedy covering over the at most $(em/2d)^{2d}$ boxes defined by the sample. Thus for d constant this algorithm runs in polynomial time. Long and Warmuth [1994] present an algorithm to PAC-learn this same class by again drawing a sufficiently large sample and constructing a hypothesis

[4] A learning algorithm is proper if all hypotheses come from the concept class.

that consists of at most $s(2d)^s$ boxes consistent with the sample. Thus both the time and sample complexity of their algorithm depend polynomially on s, d^s, $(1/\epsilon)$, and $\lg(1/\delta)$. So for s constant this yields an efficient PAC algorithm.

In summary, here is the relationship of our work to previous work for learning geometric concepts under the PAC model for an arbitrary distribution (and without membership queries). Most previous work only considered learning the union (or intersection) of either halfspaces or axis-aligned boxes. For these results, polynomial time algorithms are obtained only for either a constant number of halfspaces/boxes or constant dimension. Furthermore, these results can be formulated as SQ algorithms and thus noise tolerance can be obtained. Our work handles a much more general combination of halfspaces (or boxes) for constant dimension. Roughly, the computational cost is that instead of having a covering algorithm cover a set of m points, it must cover up to $O(m^2)$ pairs. We note that it is not known if Baum's algorithm to learn the xor of two halfspaces can tolerate even random classification noise. Finally, we note that the discretized version of the union of axis-aligned boxes generalizes DNF. In particular, a polynomial time algorithm for arbitrary d and s would solve the problem of learning DNF.

Under a variation of the PAC model (with some added restrictions) in which membership queries can be made, Baum [1991] gives an algorithm that PAC-learns the union of s halfspaces in $\mathcal{R}^d$ in time polynomial in s, d. Recently, Kwek and Pitt [1996] give a result in the standard PAC-model with membership queries to learn the union of s halfspaces in $\mathcal{R}^d$ in time polynomial in s, d, and a parameter that characterizes the number of bits of accuracy with which the target hyperplanes are specified. Also, Frazier et al. [1996] have given an algorithm to PAC-learn the s-fold union of boxes in E^d for which each box is entirely contained within the positive quadrant *and* contains the origin. Their algorithm learns this subclass of general unions of boxes in time polynomial in both s and d. In the algorithms presented here, no membership queries are used and there are no parameters needed that relate to the number of bits of precision to specify the hyperplanes defining the target. However, we require that d be constant.

There has been substantial work on exactly learning using equivalence queries (and in some cases also membership queries) unions of boxes [Chen and Homer 1994; Goldberg et al. 1993] and other more complex discretized geometric concepts [Bshouty et al. 1994; 1999]. While most such work assumes that there are a constant number of dimensions, recently Auer et al. [1996] gave a polynomial time algorithm that is robust against noise to learn the class of depth two linear threshold circuits with a polynomial number of variables given that the input gates have constant fan-in. This class is broader than the class of union or intersections of halfspaces, but not as broad as the class we consider here. Their algorithm does have time polynomial in the number of dimensions. However, the requirement that each input gate has constant fan-in equates to only allowing halfspaces that have a constant number of non-zero coefficients. In contrast, we allow arbitrary halfspaces and an arbitrary circuit, but have time complexity that is exponential in the number of dimensions. Finally, following the original conference publication of our work, Ben-David et al. [1997] and Bshouty [1998] have developed algorithms to learn exactly the discretized version of the class we study using only equivalence queries.

6. *PAC-Learning* $\mathcal{C}_s^d$

In this section we present an algorithm for learning $\mathcal{C}_s^d$ using a set covering approach. As before we use s to denote the number of hyperplanes defining the target concept. We note that the algorithm we present in this section has polynomial sample complexity. The time used by this algorithm has an exponential dependence on d. In Section 7, we convert this algorithm to an algorithm in the relative statistical query model.

6.1. OUR ALGORITHM. We begin by discussing the algorithm at a high level. We first draw a labeled sample S of size

$$m = O\left(\frac{1}{\epsilon}\lg\frac{1}{\delta} + \frac{ds}{\epsilon}\left(\lg\frac{ds}{\epsilon}\right)^3\right).$$

In Corollary 6, we show that this sample complexity can be improved in terms of s and ϵ. (For ease of exposition we assume that the learner knows s. If s is unknown, standard doubling techniques can be applied.) We then create a complete bipartite graph on this sample, treating the sample points as vertices, by introducing a 1-dimensional edge for each $+/-$ pair in the sample. That is, for each positively labeled point we add edges connecting it to every negatively labeled point. This yields a graph with $O(m^2)$ edges. We then perform a set covering of the edges in this graph using $(d - 1)$-dimensional hyperplanes.

We say that an edge is covered by a hyperplane if the hyperplane intersects the relative interior of the edge. We will shortly describe our choice of the set $\mathcal{F}$ of covering hyperplanes, which ensures that $\mathcal{F}$ contains a subset of s hyperplanes that separate every $+/-$ pair in the sample. The set covering problem, however, is NP-complete, and so we use a greedy approximation algorithm [Johnson 1974; Lovász 1975; Chvatal 1979], which gives an approximation ratio bound of $O(\lg m)$ for our application. The set of hyperplanes returned by the set covering algorithm partitions the space into $O((s \lg m)^d)$ regions. Since the algorithm separates each $+/-$ pair by some hyperplane(s), the label for each region containing sample points is easily determined: all of the points in the region are identically labeled. Any region not containing any sample points can be labeled arbitrarily.

This set of hyperplanes returned by the set covering algorithm, along with the region labels, forms our hypothesis. Since the hypothesis has size that is polynomial in the size of the target and sublinear in the size of the sample, the algorithm is an Occam algorithm, which by Theorem 2, is a PAC algorithm for this class. What remains is to show that we can guarantee that the set of candidate hyperplanes we use for the greedy covering contains a subset of s hyperplanes that separate every $+/-$ pair in the sample.

We now argue that we can construct an appropriate set of candidate hyperplanes. (Recall that the results of Blumer et al. [1989] allow us to build $\mathcal{F}$ in the more general setting where the VC-dimension of the class $\mathcal{C}$ from which we combine concepts is constant and there is a polynomial time algorithm to determine if there is a concept from $\mathcal{C}$ consistent with a given set of points.) A hyperplane partitions the set S of sample points into three subsets: one set on the hyperplane and one set on each side of the hyperplane. We call two hyperplanes *equivalent with respect to S* if they induce an identical partition of S. To ensure

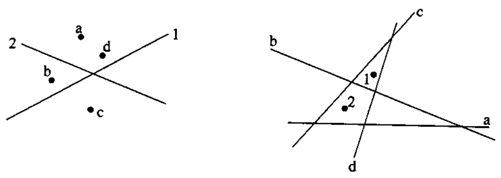

FIG. 2. A 2-dimensional primal space (left) and its 2-dimensional dual space (right).

that our set cover problem has an optimal solution of size at most s, we want our collection of hyperplanes to include at least one hyperplane equivalent, with respect to S, to each hyperplane in the target concept. For this purpose, we pick one representative hyperplane from each equivalence class. To achieve this goal, we use a geometric duality argument (see, e.g., Edelsbrunner [1987]). Each point p in our original or *primal space* is mapped to a hyperplane $g(p)$ in the *dual space*, and each hyperplane q in the primal space is mapped to a point $g'(q)$ in the dual space. Note that g' is the inverse of g and the mappings maintain relative position. That is, if point p is above hyperplane q in the primal space, then point $g'(q)$ is above hyperplane $g(p)$ in the dual space. (See Figure 2.) Thus, each equivalence class is represented by the interior of a $(d - k)$-dimensional face of the arrangement $\mathcal{A}$ consisting of dual hyperplanes $g(p)$, $p \in S$, where k is the number of points of S on a hyperplane in the equivalence class. The *centroid* of a finite set of points is simply the arithmetic mean of the points and is known to be within the convex hull of the points. Since every feature of $\mathcal{A}$ is convex, the centroid of a feature in $\mathcal{A}$ yields an appropriate hyperplane in the primal space. Therefore, the number of those equivalence classes is at most $O(m^d)$ and the representative for each can be listed (and thus placed in $\mathcal{F}$) in polynomial time.[5]

Since $\mathcal{F}$ contains s hyperplanes consistent with the target concept, the greedy set covering algorithm is guaranteed to return a hypothesis of size $O(s \lg m)$. In the next section, we analyze this algorithm.

6.2. ANALYSIS.

In this section we prove the following result:

THEOREM 5. *There exists an algorithm that PAC-learns $\mathcal{C}_s^d$. The algorithm draws a sample of size*

$$m = O\left(\frac{1}{\epsilon}\lg\frac{1}{\delta} + \frac{ds}{\epsilon}\left(\lg\frac{ds}{\epsilon}\right)^3\right)$$

and uses time polynomial in m^d, s^d, $1/\epsilon$ and $1/\delta$. The hypothesis returned contains $O(s \lg m)$ hyperplanes.

PROOF. Our algorithm, LEARN-$\mathcal{C}_s^d$, is shown in Figure 3. We first show that the stated sample complexity is sufficient. Since the size of the covering is $O(s \lg m)$, the final hypothesis is the Boolean combination of $O(s \lg m)$ hyperplanes. The VC dimension of a single hyperplane in d-dimensional space is easily shown to be $d + 1$. Thus, by applying a variation of Lemma 3.2.3 of Blumer et al. [1989] (see the appendix), the VC dimension of the hypothesis class is at most $O(ds \lg$

[5] Any interior point in the convex hull would suffice. We picked the centroid simply because it is easy to compute.

```
LEARN-$\mathcal{C}_s^d(s, \epsilon, \delta)$:
  Draw labeled sample $S$ of size
      $m = O\left(\frac{1}{\epsilon}\lg\frac{1}{\delta} + \frac{ds}{\epsilon}\left(\lg\frac{ds}{\epsilon}\right)^3\right)$
  Create $\mathcal{U}$ by adding edge for every $+/-$ pair in $S$
  $\mathcal{F} \leftarrow \emptyset$
  Create a geometric dual space containing a hyperplane
    for each point in $S$
  Calculate centroid $\gamma$ of each feature in the dual space
  $\mathcal{F} \leftarrow \mathcal{F} \cup \{g'(\gamma)\}$
  $F \leftarrow$ greedy covering of $\mathcal{U}$ using $\mathcal{F}$
  $h \leftarrow$ regions formed by $F$
  For each region $\rho \in h$
      If $\rho \cap S \neq \emptyset$ then label $\rho$ according to $S$
      Else label $\rho$ arbitrarily
  Return $h$
```

FIG. 3. Pseudocode for the algorithm to LEARN-$\mathcal{C}_s^d$.

$m \lg(s \lg m)) = O(ds(\lg m)^2)$. Thus, by Theorem 2 [Blumer et al. 1989], we get that the stated sample size suffices.

We now argue that the running time of this algorithm is polynomial in m^d, s^d, $1/\epsilon$ and $1/\delta$. We first note that drawing a sample of size m requires $O(m)$ time. The algorithm then creates a complete bipartite graph on the sample, which takes $O(m^2)$ time since the graph has $O(m^2)$ edges. Creating the arrangement $\mathcal{A}$ of the dual hyperplanes requires $O(m^d)$ time. The total complexity of the arrangement, counting cells of all dimensions i, for $0 \le i \le d$, is also $O(m^d)$.

Next we examine the time required to create $\mathcal{F}$. As described above, to build $\mathcal{F}$, we work in the dual space. We must find an interior point of each $(d - k)$-dimensional feature. If there are no interior points, as for a one-dimensional object, then a point on the boundary of the feature is sufficient. Note that when we construct the arrangement $\mathcal{A}$, we calculate the vertices of all of the features. Thus, we have the convex hull of each feature and can calculate the centroid in constant time. Therefore, since there are $O(m^d)$ features of which we must find the centroid, each requiring constant time, the time required to construct $\mathcal{F}$ is $O(m^d)$. Note that this is also the number of hyperplanes in $\mathcal{F}$. (We can calculate the centroids of the features as $\mathcal{A}$ is being constructed without changing asymptotically the time required for this operation, rendering the separate calculation of the centroids unnecessary.)

The time complexity of the greedy set covering algorithm is $O(\Sigma_{p\in\mathcal{F}}|p|)$, where $|p|$ is the size of the set p (that is, the number of edges of our bipartite graph intersected by p). Since the family $\mathcal{F}$ has $O(m^d)$ sets, and $|p| = O(m^2)$, for all $p \in \mathcal{F}$, the set covering algorithm runs in time $O(m^{d+2})$, and produces a hypothesis containing $O(s \lg m)$ hyperplanes. The set covering is the dominant step in our learning algorithm, and since m is polynomial in s, d, $1/\epsilon$ and $1/\delta$, the theorem has been established. □

Replacing the standard greedy set covering algorithm with one due to Brönnimann and Goodrich [1994] we obtain the following result. (Note that here the complexity of the hypothesis class does not depend on the sample size m. Thus instead of using an Occam algorithm as above, we can apply Theorem 1 [Blumer et al. 1989].)

COROLLARY 6. *There exists an algorithm that PAC-learns $\mathcal{C}_s^d$. The algorithm draws a sample of size*

$$m = O\left(\frac{1}{\epsilon}\lg\frac{1}{\delta} + \frac{sd^2\log^2(sd)}{\epsilon}\lg\frac{1}{\epsilon}\right)$$

and uses time polynomial in m^d, s^d, $1/\epsilon$ and $1/\delta$ to return a hypothesis containing $O(sd \lg(sd))$ hyperplanes.

Finally, by applying Lemma 3.2.2 from Blumer et al. [1989] instead of using the geometric duality construction to build $\mathcal{F}$ we get the following generalization.[6]

COROLLARY 7. *Let $\mathcal{C}$ be a concept class over $\mathcal{R}^d$ for which VCD($\mathcal{C}$) has dependence only on d, and for which there is a polynomial time algorithm to determine if there is a concept from $\mathcal{C}$ consistent with a set of labeled examples. Then there exists a PAC-learning algorithm for the concept class defined by the Boolean combination of any s concepts from $\mathcal{C}$. The sample complexity of the algorithm is polynomial in s, d, $1/\epsilon$, and $1/\delta$, and the time complexity of the algorithm is polynomial in s, $1/\epsilon$, and $1/\delta$.*

Note that the sample complexity of our algorithm is polynomial in all of the relevant parameters, *including* d. It is only the time complexity that has an exponential dependence on d. We also note that it is not necessary for the algorithm to label every region of the hypothesis. We label only those regions containing sample points. Since all other regions can be labeled arbitrarily, we leave them unlabeled and supply an arbitrary classification for a point in one of these regions when required. Thus, our hypothesis has size polynomial in s and d. In the next section, we present a statistical query version of this algorithm that is robust against random classification noise for any noise rate $\eta < 1/2$.

7. *PAC-Learning $\mathcal{C}_s^d$ with Noisy Data*

In this section, we extend our algorithm of the previous section to allow random classification noise of any rate $\eta < 1/2$. We do this by converting our algorithm to one in the relative error SQ model. The advantage of providing an SQ algorithm is that there is no need to explicitly deal with noise–by a standard conversion result, we obtain a PAC algorithm that is provably correct when the data is corrupted with a high rate of random classification noise (as well as small rates of other forms of noise). As in the noise-free case, we form a set $\mathcal{F}$ of hyperplanes for the covering. In the covering stage, the goal is to ensure that there is a very small probability that a random positive and random negative point are not separated by a hyperplane in our covering. Since the provided labels are unreliable, we use a statistical query (which can tolerate labeling noise) to estimate the probability p that a random positive and negative point are not yet separated. Given a hypothesis h, let the $+/-$ *error* of h, denoted $E_{+/-}(h)$, be the

[6] In contrast to using the approach described above of finding the centroid of each feature in the dual space, the general technique of Blumer et al. [1989] when $\mathcal{C}$ is the class of halfspaces would involve solving $\Omega(m^d)$ linear programs where each linear program is needed to determine if a set of up to m labeled points are linearly separable. Thus, the geometric duality approach gives a significant improvement in the complexity of computing $\mathcal{F}$.

probability that a randomly drawn positive point and a randomly drawn negative point are in the same region of the hypothesis. Our goal is thus to reduce $E_{+/-}(h)$ by adding additional hyperplanes from $\mathcal{F}$ to h. Then we can use statistical queries to determine the classification for each region in h. Because the error measure, $E_{+/-}(h)$ we are reducing (by adding hyperplanes to h) is different from the probability that a randomly selected point is misclassified, we cannot just use the standard method of transforming a set covering based algorithm into a SQ algorithm in which a statistical query is used to determine which halfspace in $\mathcal{F}$ reduces the false positive error the most (i.e., covers the most uncovered negative weight).

7.1. A NOISE-TOLERANT ALGORITHM. Before describing our algorithm, we give some definitions. Let h denote the hypothesis of the learning algorithm and r denote the number of regions in h. Let $\mathcal{D}$ be the probability distribution on $\mathcal{R}^d$. We define $\mathcal{D}'$ to be the *filtered distribution* producing the pair $\langle x_+, x_- \rangle$, where x_+ is a random positive example drawn according to $\mathcal{D}$ and x_- is a random negative example drawn according to $\mathcal{D}$. By a filtered distribution, we mean a modification of the original distribution created by the learner by removing (or filtering out) some of the examples. Here, the learner will draw, independently, two random points from $\mathcal{D}$ and then filter out those examples in which the two points have the same labels. The examples that are not filtered (i.e., those where the pair have different labels) form the distribution $\mathcal{D}'$. We call $\langle x_+, x_- \rangle$ a $+/-$ *pair*.

Our algorithm first draws an unlabeled sample S_u of size m_u. As in the noise-free algorithm, we create a family of hyperplanes, $\mathcal{F}$, producing all possible behaviors of m_u. Since the first stage (i.e., creating $\mathcal{F}$) used an unlabeled sample, no change is needed. We now describe the second stage of our algorithm. We want to use statistical queries to create a set $F \subseteq \mathcal{F}$ of halfspaces to separate the positive weight in $\mathcal{D}$ from the negative weight in $\mathcal{D}$. Initially $F = \emptyset$ (and so the initial hypothesis h has one region). Then in each step of the loop (as discussed in more detail below), we add a hyperplane to F (and update h accordingly) until $\hat{P}_{\chi_1^h}$ drops below a certain threshold. More specifically, let χ_1^h be the predicate "*x_+ and x_- are in same region of h*" and let χ_2^p be the predicate "*x_+ and x_- are separated by hyperplane p*". We perform the following loop at most t times (we derive t in Section 7.2). We first use the rel-STAT oracle to estimate $P_{\chi_1^h} = E_{+/-}(h)$. Thus, by the definition of the rel-STAT oracle, the estimate $\hat{P}_{\chi_1^h}$ returned (if not $\perp$) by rel-STAT$_{\mathcal{D}'}(\chi_1^h, 1/2, (2\epsilon/t^d + 1))$ satisfies

$$\frac{1}{2} P_{\chi_1^h} \le \hat{P}_{\chi_1^h} \le \frac{3}{2} P_{\chi_1^h}.$$

We will exit the repeat loop when $\hat{P}_{\chi_1^h} \le (\epsilon/(t^d + 1))$, and thus it follows that in this case $P_{\chi_1^h} \le 2\hat{P}_{\chi_1^h} \le (2\epsilon/(t^d + 1))$. Likewise, if $\perp$ is returned then, by definition, $P_{\chi_1^h} \le (2\epsilon/(t^d + 1))$. Thus if we exit the repeat loop after less than t executions, then $E_{+/-}(h) \le (2\epsilon/(t^d + 1))$.

If $\hat{P}_{\chi_1^h} > (\epsilon/(t^d + 1))$, then we continue the repeat loop and thus select the next hyperplane to add to our hypothesis. To achieve this, for each candidate hyperplane in $\mathcal{F}$ we call rel-STAT$_{\mathcal{D}'}$ $(\chi_1^h \wedge \chi_2^p, 1/2, (\epsilon/(4s(t^d + 1)))$ which estimates the probability that adding the given hyperplane to h would create a hypothesis h' that separates a random pair $\langle x_+, x_- \rangle$ that was not separated by h.

```
LEARN-NOISY-$\mathcal{C}_s^d(s, \epsilon, \delta)$:

  Draw unlabeled sample $S_u$ of size
    $m_u = \frac{16(t^d+1)}{3\epsilon} \cdot \left(2(d+1)t \lg(3t) \lg \frac{48(t^d+1)}{3\epsilon} + \lg \frac{6}{\delta}\right)$
  Create set of candidates, $\mathcal{F}$, as in the noise free case
  $F = \emptyset$
  Repeat $t$ times
    If rel-STAT$_{\mathcal{D}'}(\chi_1^h, 1/2, \frac{2\epsilon}{t^d+1}) > \frac{\epsilon}{(t^d+1)}$

      $p_{max} = \mathrm{argmax}_{p \in \mathcal{F}} \left\{ \text{rel-STAT}_{\mathcal{D}'} \left(\chi_1^h \wedge \chi_2^p, 1/2, \frac{\epsilon}{4s(t^d+1)}\right)\right\}$
      $F \leftarrow F \cup p_{max}$
      $\mathcal{F} \leftarrow \mathcal{F} - \{p_{max}\}$
      $h \leftarrow$ regions formed by $F$
    Else exit the repeat loop
  For each region $\rho$ of $h$
    If rel-STAT$_{\mathcal{D}}(\chi_3, 1/4, 1/2) \geq 1/2$
      Label $\rho$ positive
    Else
      Label $\rho$ negative
  Return $h$
```

FIG. 4. Our noise-tolerant algorithm for $\mathscr{C}_s^d$. Recall that $t = \max\{6s \ln(4/\epsilon), 35sd \ln(sd)\}$, $\langle x_+, x_- \rangle$ is a $+/-$ *pair*, χ_1^h is the predicate "x_+ and x_- *are in same region of* h", and χ_2^p is the predicate "x_+ *and* x_- *separated by hyperplane* p". χ_3 is the predicate "$x \in \rho$ and $f(x) = 1$". So $\langle x_+, x_- \rangle$ denotes a pair of examples where x_+ is a random positive example and x_- is a random negative example, each drawn according to $\mathfrak{D}$. Recall that $\mathfrak{D}'$ outputs $\langle x_+, x_- \rangle$.

We then add to h the hyperplane $p \in \mathscr{F}$ for which the highest estimate was returned. In Section 7.2, we prove that $t = \max\{6s \ln(4/\epsilon)\ 35sd \ln(sd)\}$ rounds are sufficient to ensure that $E_{+/-}(h) = P_{\chi_1^h} \leq (2\epsilon/(t^d + 1))$. Thus, after at most t repetitions of the repeat loop, we are guaranteed that $P_{\chi_1^h} \leq (2\epsilon/(t^d + 1))$. As we will show, it then follows that if the regions in h are labeled appropriately (i.e., based on the majority of the weight), the classification error of h will be at most $\epsilon/2$.

Once the $+/-$ error becomes sufficiently low, we stop even if fewer than t covering rounds were used. Otherwise, the threshold for the statistical query to estimate $P_{\chi_1^h \wedge \chi_2^p}$ would need to be too small. Also, as long as the positive and negative weight in $\mathfrak{D}$ is greater than $\epsilon/2$, it is easily seen that $\mathfrak{D}'$ can be efficiently simulated. For ease of exposition, we have not explicitly included a test to see if the positive or negative weight in $\mathfrak{D}$ is too small.

Finally, for each region ρ of h, we use statistical queries of the form rel-STAT$_{\mathfrak{D}}((x \in \rho$ and $f(x) = 1), 1/4, 1/2)$ to determine the label of ρ as given by the majority of the weight in ρ. Pseudocode for the entire statistical query algorithm appears in Figure 4.

In the next section, we derive the sample complexity and the bound, t, on the number of hyperplanes added to the hypothesis. We also examine the time complexity of this algorithm.

7.2. ANALYSIS.

In this section, we prove the following theorem:

THEOREM 8. *There exists an algorithm to PAC-learn* $\mathscr{C}_s^d$ *in the presence of random classification noise with a rate* $(1/2) - \gamma$ $(\gamma > 0)$, *with time and sample complexity polynomial in* s, $1/\epsilon$, *log* $1/\delta$, *and* $1/\gamma$ *for fixed* d.

The repeat loop of our algorithm is executed at most t times. Thus, t is an upper bound on the number of hyperplanes added to the hypothesis. This gives us that the maximum number of regions in the hypothesis is $r \leq t^d + 1$. Before proving the main result of this section, we first derive the value for t. To distinguish the standard error measure from $E_{+/-}(h)$, we refer to the probability that a randomly drawn point is misclassified by the hypothesis as the *classification error*, denoted $E_C(h)$. Since the metric used during the greedy covering is $E_{+/-}(h)$, we must relate this to the classification error so we can ensure that when the algorithm halts $E_C(h) \leq \epsilon$.

LEMMA 1. *Suppose* $E_C(h) > \epsilon$. *Then* $E_{+/-}(h) > (4\epsilon(1 - \epsilon)/r)$, *where r is the number of regions in h.*

PROOF. We use P_+ to denote the total weight of positive instances and P_- the total weight of negative instances under distribution $\mathcal{D}$. We use $x \in \mathcal{D}$ to denote an instance x drawn at random from $\mathcal{D}$. Our goal is to express $E_{+/-}(h)$ in terms of ϵ for $E_C(h) > \epsilon$. To achieve this goal we take the summation over all regions i (for $1 \leq i \leq r$) of h of the $+/-$ error due to region i. Notice that the $+/-$ error due to region i is the probability that a random positive example is in region i times the probability that a random negative example is in region i. Let $p_i = Pr_{x \in \mathcal{D}}(x \in i \wedge f(x) = 1)$ and let $n_i = Pr_{x \in \mathcal{D}}(x \in i \wedge f(x) = 0)$. Thus, we have that

$$E_{+/-}(h) = \sum_{i=1}^{r} Pr_{x \in \mathcal{D}}(x \in i | f(x) = 1) \cdot Pr_{x \in \mathcal{D}}(x \in i | f(x) = 0)$$

$$= \sum_{i=1}^{r} \frac{Pr_{x \in \mathcal{D}}(x \in i \wedge f(x) = 1)}{P_+} \cdot \frac{Pr_{x \in \mathcal{D}}(x \in i \wedge f(x) = 0)}{P_-}$$

$$= \sum_{i=1}^{r} \frac{p_i}{P_+} \cdot \frac{n_i}{(1 - P_+)} \geq 4 \sum_{i=1}^{r} p_i n_i > \frac{4\epsilon(1 - \epsilon)}{r}$$

where the last step follows by using Lagrange multipliers. □

Therefore, we know that if $E_C(h) > (\epsilon/2)$, then

$$E_{+/-}(h) > \frac{4 \cdot (\epsilon/2)(1 - (\epsilon/2)}{r} = \frac{2\epsilon(1 - (\epsilon/2))}{t^d + 1}.$$

Thus, if

$$E_{+/-}(h) \leq \frac{2\epsilon(1 - (\epsilon/2))}{t^d + 1},$$

then $E_C(h) \leq (\epsilon/2)$. Observe that since the number of regions in the hypothesis increases with each hyperplane added, the value of $E_{+/-}(h)$ that must be obtained decreases with each hyperplane added to the hypothesis. The following observation bounds the $+/-$ error of h in terms of the classification error. The worst case occurs for a target concept when, without loss of generality, all examples are positive. Hence, the ϵ weight of examples misclassified by h are all false negatives and create $+/-$ error with the $1 - \epsilon$ weight of positive examples that are correctly classified by h.

OBSERVATION 1. *Let ϵ represent the classification error of hypothesis h. Then $E_{+/-}(h) \le \epsilon(1 - \epsilon) \le \epsilon$.*

Let e_i be the value $E_{+/-}(h)$ after the ith hyperplane has been added to the hypothesis. By the construction of $\mathcal{F}$ and the fact that m_u was chosen (based on the bounds of Blumer et al. [1989]) so that, with probability at least $1 - \delta/3$, any hypothesis consistent with S_u (if we knew the correct labels) would have error at most $(3\epsilon/4(t^d + 1))$, we know that with probability at least $1 - \delta/3$, there exist s hyperplanes in $\mathcal{F}$, that when properly oriented to form halfspaces, generate a hypothesis with classification error at most $(3\epsilon/4(t^d + 1))$. Thus, there exist s hyperplanes in $\mathcal{F}$ that have $+/-$ error at most $(3\epsilon/4(t^d + 1))$. By an averaging argument (over the error reduction obtained by including these s hyperplanes), when selecting the ith hyperplane to add, there exists a $p \in \mathcal{F}$ such that

$$\begin{aligned} \mathrm{P}_p &= \Pr_{\langle x_+, x_- \rangle \in \mathcal{D}'}((p \text{ separates } x_+ \text{ and } x_-) \\ &\quad \wedge (x_+ \text{ and } x_- \text{ are not separated by } h)) \\ &\ge \left(e_{i-1} - \frac{3\epsilon}{4(t^d + 1)} \right) \frac{1}{s}. \end{aligned}$$

Since we stay in the repeat loop only if

$$e_i \ge \frac{\epsilon}{(t^d + 1)}$$

we know that

$$\mathrm{P}_p \ge \left(\frac{\epsilon}{(t^d + 1)} - \frac{3\epsilon}{4(t^d + 1)} \right) \frac{1}{s} = \frac{\epsilon}{4s(t^d + 1)}.$$

Thus, we let

$$\theta = \frac{\epsilon}{4s(t^d + 1)}$$

for the statistical queries to estimate P_p since we need no estimates when $\mathrm{P}_p < \theta$ (since, in this case, p is not an optimal choice). We can now solve for t.

LEMMA 2. *Let t be the maximum number of iterations of the greedy covering algorithm (thus, the maximum number of hyperplanes in h). Choosing $t = max\{6s\ ln(4/\epsilon), 35sd\ ln(sd)\}$ suffices to ensure that $E_{+/-}(h) \le (2\epsilon/(t^d + 1))$, and thus, $E_C(h) \le (\epsilon/2)$.*

PROOF. Let $\hat{\mathrm{P}}_p$ represent the estimate of P_p obtained from the SQ oracle. Since we use a relative error bound of 1/2,

$$\hat{\mathrm{P}}_p \geq \left(e_i - \frac{3\epsilon}{4(t^d+1)}\right)\frac{1}{2s}.$$

Thus, for p_{max},

$$\hat{P}_{p_{max}} \geq \left(e_i - \frac{3\epsilon}{4(t^d+1)}\right)\frac{1}{2s}.$$

Once again, because we use a relative error bound of 1/2, it follows that

$$P_{p_{max}} \geq \hat{P}_{p_{max}} \cdot \frac{2}{3} = \left(e_i - \frac{3\epsilon}{4(t^d+1)}\right)\frac{1}{3s}.$$

Therefore,

$$e_{i+1} \leq e_i - \left(e_i - \frac{3\epsilon}{4(t^d+1)}\right)\frac{1}{3s}.$$

Solving this recurrence yields

$$e_i \leq \left(1 - \frac{1}{3s}\right)^i + \frac{3\epsilon}{4(t^d+1)}.$$

To ensure that $E_C(h) \leq (\epsilon/2)$ it suffices to select t such that

$$\left(1 - \frac{1}{3s}\right)^t + \frac{3\epsilon}{4(t^d+1)} \leq \frac{2\epsilon(1-(\epsilon/2))}{r} = \frac{2\epsilon(1-(\epsilon/2))}{t^d+1}.$$

Since $2\epsilon(1 - (\epsilon/2)) \geq \epsilon$, it suffices to let

$$\left(1 - \frac{1}{3s}\right)^t + \frac{3\epsilon}{4(t^d+1)} \leq \frac{\epsilon}{t^d+1}.$$

Equivalently, we must pick t such that

$$\left(1 - \frac{1}{3s}\right)^t \leq \frac{\epsilon}{4(t^d+1)}.$$

Since $\exp(-t/(3s)) \geq (1 - (1/3s))^t$, it suffices if

$$\exp(-t/(3s)) \leq \frac{\epsilon}{4(t^d+1)}.$$

Taking the logarithms of both sides and solving for t yields $t \geq 3s(\ln 4/\epsilon + \ln(t^d + 1))$ and hence it suffices if

$$t \geq 6s \cdot \max\left\{ \ln\frac{4}{\epsilon},\ \ln(t^d + 1) \right\}.$$

It can be verified that for $t = 35sd \ln(sd)$, $t \geq 6sd \ln(t + 1) = 6s \ln((t + 1)^d) \geq 6s \ln(t^d + 1)$. Hence, we get the final result that $t = \max\,\{6s \ln(4/\epsilon), 35sd \ln(sd)\}$ suffices. □

We are now ready to prove the main result of this section. Since we used the SQ model in designing our noise-robust algorithm, LEARN-NOISY, we must now explicitly convert LEARN-NOISY into a noise-tolerant PAC algorithm. In this conversion, a labeled sample (corrupted by the random classification noise) is used to estimate the values for the statistical queries. Since the examples used by LEARN-NOISY are drawn from $\mathcal{D}'$, we must take this into account when simulating our SQ algorithm by a PAC algorithm.

PROOF OF THEOREM 8. The confidence and accuracy parameters are allocated in the following way. We use $\delta/3$ when drawing sample S_u and $\delta/3$ in the filtering used to create $\mathcal{D}'$ (with probability $\geq 1 - \delta/3$ the filtering does not take too long). This leaves $\delta/3$ for the simulations of the statistical queries from the PAC example oracle. This gives a total probability of failure of δ. For the accuracy bound, recall that by Lemma 2, and the observation that we exit the repeat loop prematurely only if $E_{+/-}(h) \leq (2\epsilon/(t^d + 1))$, it follows that once the regions of h are labeled appropriately $E_C(h) \leq \epsilon/2$.

For each region of h a statistical query is performed to determine its classification. Since $E_C(h) \leq \epsilon/2$ when the labels are picked based on the majority of the weight in the region, for each region we must have that either $P_{\chi 3} \leq \epsilon/2$ or $P_{\chi 3} \geq 1 - \epsilon/2$. Without loss of generality we assume that $0 < \epsilon < 1/2$. Then, we have that either $P_{\chi 3} \leq 1/4$ or $P_{\chi 3} \geq 3/4$. If $P_{\chi 3} \leq 1/4$, then $\hat{P}_{\chi 3} \leq 5/16 < 1/2$ and thus the region is correctly classified as negative. Likewise, if $P_{\chi 3} \geq 3/4$, then $\hat{P}_{\chi 3} \geq 9/16 > 1/3$ and thus the region is correctly classified as positive. Finally, if $\perp$ is returned, then $P_{\chi 3} < 1/2$ and thus the correct classification of negative is given.

We now argue that the time and sample complexity are polynomial in the stated parameters. Since h can be viewed as a Boolean combination of at most

$$t = \max\left\{ 6s \ln\frac{4}{\epsilon},\ 35sd \ln(sd) \right\}$$

halfspaces (obtained by arbitrarily orienting each of the hyperplanes), the VCD($\mathcal{H}$) is at most $2(d + 1)t \lg(3t)$ (by Blumer et al. [1989]). After substituting the above values for ϵ and

$$\delta,\ m_u = \frac{16(t^d + 1)}{3\epsilon} \cdot \left(\mathit{VCD}(\mathcal{H}) \lg\frac{48(t^d + 1)}{3\epsilon} + \lg\frac{6}{\delta} \right)$$

is sufficient to ensure a hypothesis consistent with the sample has error greater than $(3\epsilon/4(t^d + 1))$ with probability at most $\delta/3$.

Finally, we consider the time complexity of this algorithm. By the argument presented in the proof of Theorem 5, it takes $O((m_u)^d)$ time to construct candidate set $\mathcal{F}$. Simulating each statistical query, in the PAC model, requires polynomial time since for all statistical queries made $1/\mu$ and $1/\theta$ are polynomial in s, $1/\epsilon$, and $1/\delta$.

For each of the t rounds of the covering we perform $O(m^d)$ statistical queries to select the hyperplane to add to h. Finally, we must label each of the regions in h. There are $r = t^d + 1$ regions, each requiring a statistical query. Thus, the total time required by this algorithm is polynomial. □

8. *A Planar Subdivision Approach*

In the case $d = 2$, we explore a method of recursively subdividing the plane. In this approach for learning $\mathcal{C}_s^2$ $(d = 2)$ with *malicious* classification noise of rate $(1/2) - \gamma$ (for $\gamma > 0$), the learner takes $mq = poly(1/\epsilon, 1/\delta, 1/\gamma, s)$ labeled examples according to a distribution $\mathcal{D}$. The learner subdivides the space in the following manner. Given that the points are in general position (namely, no three points are collinear), there exist two lines that divide the points into four groups of equal size [Willard 1982]. We build the subdivision by finding such a pair of lines, and then we recursively apply the same technique (independently) to each of the four regions obtained until there are only q points in each region. Thus, after $\log_4 m$ division we obtain m polyhedra of at most $\log_4 m$ sides where each contains exactly q points. Furthermore, since each of the s lines defining the target concept can intersect at most three of the four regions created, the number of polyhedra intersected by a single line is at most $3^{\log_4 m} = m^{\log_4 3} = m^\alpha$ where $\alpha = \log_4 3 < 0.8$.

Since our sample may not have points in general position, when three or more points lie on a line, then with high probability this line has high (i.e., at least $1/poly$) weight under the distribution and we can learn the target function on this line separately (it will just be a union of intervals). Notice that only a polynomial number of lines can have large weight. Thus, with probability at least $1 - \delta/2$, we can learn all such lines of high weight with a total error of at most $\epsilon/2$.

Each region of the subdivision is labeled according to a majority of the q points in that region. Since each line intersects at most m^α regions, and by the lemma in Appendix B, each region has distribution at most $\xi = \tilde{O}(m^{-1})$ where $\tilde{O}$ means that we are omitting log factors. Therefore, we need to have $sm^\alpha\xi = \tilde{O}(s/m^{1-\alpha}) \leq (\epsilon/2)$. So it is enough to choose $m = \tilde{O}(s/\epsilon)^{(1/(1-\alpha))}$. Notice that the condition that $\alpha < 1$ is critical. Finally, by applying Chernoff bounds, given that

$$q = O\left(\frac{1}{\gamma^2}\left(\lg m + \lg\frac{1}{\delta}\right)\right),$$

with probability at least $1 - \delta/2$, the label of all nonintersected regions will be correct and thus the total error is at most ϵ.

9. *Concluding Remarks*

We have described a simple algorithm for efficiently PAC-learning an extremely general geometric concept class in constant-dimensional space even when there is a high rate of random classification noise against an arbitrary unknown distribution. This concept class, defined by taking Boolean combinations of half-spaces, is the most general geometric class for which we know algorithms to exist. Although we consider geometric concepts defined by half-spaces, any polyhedron (not necessarily convex) defined by f faces can be formed by combining f half-spaces. Thus, our algorithm can also learn any Boolean combination of a polynomial number of polyhedra each with a polynomial number of faces. Even further, our results can be easily extended to efficiently learn any Boolean combination of a polynomial number of concepts selected from any concept class $\mathscr{C}$ given that the VC-dimension of $\mathscr{C}$ is constant (for d constant) and there is a polynomial time algorithm to determine if there is a concept from $\mathscr{C}$ consistent with a given set of points.

A nice feature of our noise-tolerant algorithm is that it can take advantage of unlabeled as well as labeled data which may be of value for some real-life applications, especially when combined with the simplicity and robustness of the algorithms.

Appendix A. Variation of Lemma 3.2.3 *of Blumer et al.*

We begin with the statement and proof of Lemma 3.2.3 directly from Blumer et al.

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 $C_s = \{\bigcup_{i=1}^{s} c_i : c_i \in C, 1 \leq i \leq s\}$ (respectively, $C_s = \{\bigcap_{i=1}^{s} c_i : c_i \in C, 1 \leq i \leq s\}$). Then for all $s \geq 1$, the VC dimension of C_s is less than $2ds \log_2(3s)$.*

PROOF. 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 $(\Phi_d(m))^2 < 2^m$, 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 $(em/d)^d s < 2^m$ for $m = 2ds \log_2(3s)$ which is equivalent to $\log_2(3s) < 9s/(2e)$. 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 $s = 2$. □

While not explicitly stated by Blumer et al., the following corollary is implicit in the above proof.

COROLLARY TO 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 be any Boolean combination of s concepts from C. Then for all $s \geq 1$, the VC dimension of C_s is less than $2ds \log_2(3s)$.*

PROOF. The only place in the above proof where the fact that C_s is an intersection of s half-spaces is used is in stating that $|\Pi_{C_s}(S)| \leq (|\Pi_C(S)|)^s$. However, clearly this same inequality holds for any Boolean combination of s half-spaces. No other changes are required. □

Appendix B. A Generalization of ϵ-Nets

Let $(X, \mathcal{R})$ be a range space and $\mathcal{D}$ be a distribution on X. We say that a set of points $N \subset X$ is an (ϵ, p)-net if any $R \in \mathcal{R}$ satisfying $\mathbf{Pr}_{\mathcal{D}}[R] > \epsilon$ contains at least p points from N. For a range space $(X, \mathcal{R})$ with $|X| = n$ points of VC-dimension d, $|\mathcal{R}| \leq g(d, n) = \sum_{i=0}^{d}\binom{n}{i} \leq n^d$. We use lg for the base-2 logarithm.

LEMMA B.1. *Let $(X, \mathcal{R})$ be a range space of VC-dimension d. Let $\mathcal{D}$ be a distribution over X. Let N be a sequence of points obtained by $m \geq 16/\epsilon$ random independent draws from X according to the distribution $\mathcal{D}$ where*

$$2g(d, 2m)\frac{\sum_{i=0}^{r-1}\binom{\epsilon m/2}{i}\binom{2m - \epsilon m/2}{m - i}}{\binom{2m}{m}} < \delta$$

Then with probability at least $1 - \delta$ we have that N is an (ϵ, r)-net for X.

LEMMA B.2. *For $r \leq \epsilon m/5$ and $m = O(max((d/\epsilon)log(1/\epsilon), (1/\epsilon)log(1/\delta)))$, we have*

$$2g(d, 2m)\frac{\sum_{i=0}^{r-1}\binom{\epsilon m/2}{i}\binom{2m - \epsilon m/2}{m - i}}{\binom{2m}{m}} \leq \delta.$$

We first give the proof of Lemma B.1.

PROOF OF LEMMA B.1. Let $\mathcal{R}_\epsilon$ be the set of all $R \in \mathcal{R}$ with $\Pr_{\mathcal{D}}[R] \geq \epsilon$. Define the random variable $A = [(\exists R \in \mathcal{R}_\epsilon)\ |R \cap N| < r]$. That is, $A = 1$ if the statement in the brackets is true and 0 otherwise. We will write $\Pr_N[A]$ for $\Pr_N[A = 1]$. To prove the lemma, we need to prove that

$$\Pr_N[A] \leq \delta.$$

Now we change the probability space to an equivalent one as follows. Instead of choosing m points in X according to the distribution $\mathcal{D}$ we choose $2m$ points W from X according to the distribution $\mathcal{D}$ and then uniformly choose m points N from W. Obviously,

$$\Pr_N[A] = \Pr_{W,N}[A].$$

Now we use the following result in probability. Let B be an event. First notice that $\Pr_N[A] = \mathbf{E}_N[A]$. Then

$$\mathbf{E}_N[B] = \mathbf{E}_{W,N}[B] = \mathbf{E}_W[\mathbf{E}_N[B|W]].$$

The inner expectation is $\mathbf{E}_N[B|W]$ is the expectation of the event B when W is a fixed set. Now, it is easier to handle this expectation because W is finite (unlike

X). For the proof we choose B to be the event

$$B = [(\exists R \in \mathcal{R}_\epsilon)|R \cap N| < r \text{ and } |R \cap W| \ge \epsilon m/2].$$

Notice that B is A with the extra condition that $|R \cap W| \ge \epsilon m/2$. We now prove the following claim.

CLAIM B.3. $\Pr_N[A] \le 2 \Pr_{W,N}[B]$.

PROOF. We have

$$\Pr_{W,N}[\bar{B}|A] = \Pr[\underline{(\forall R \in \mathcal{R}_\epsilon)|R \cap N| \ge r \text{ or } |R \cap W| < \epsilon m/2} \mid \underline{(\exists R \in \mathcal{R}_\epsilon)|R \cap N| < r}].$$

Let $R_0 \in \mathcal{R}_\epsilon$ such that $|R_0 \cap N| < r$. Then the above probability is

$$\begin{aligned}\Pr_{W,N}[\bar{B}|A] &\le \Pr_{W,N}[\underline{|R_0 \cap N| \ge r \text{ or } |R_0 \cap W| < \epsilon m/2} \mid \underline{|R_0 \cap N| < r}] \\ &= \Pr_{W,N}\left[|R_0 \cap W| \le \frac{\epsilon m}{2} \,\middle|\, |R_0 \cap N| < r \right] \\ &\le \Pr_{W,N}\left[|R_0 \cap (W \backslash N)| \le \frac{\epsilon m}{2} \,\middle|\, |R_0 \cap N| < r \right] \\ &= \Pr_{W,N}\left[|R_0 \cap (W \backslash N)| \le \frac{\epsilon m}{2} \right] \le LE\left(\epsilon, m, \frac{\epsilon m}{2}\right),\end{aligned}$$

where $LE(\epsilon, m, \epsilon m/2)$ is the probability that at most $\epsilon m/2$ points from $W \backslash N$ are in R_0 assuming $D(R_0) = \epsilon$. Since each point can be in R_0 with probability ϵ, by Chebychev bound and since $m \ge 8/\epsilon$, we have

$$LE\left(\epsilon, m, \frac{\epsilon m}{2}\right) \le \frac{m\epsilon}{(m\epsilon - \epsilon m/2)^2} = \frac{4}{m\epsilon} \le \frac{1}{2}.$$

Now

$$\begin{aligned}\Pr_{W,N}[B] &= \Pr_{W,N}[A \text{ and } B] \\ &= \Pr_{W,N}[B|A] \Pr_{W,N}[A] \\ &= (1 - \Pr_{W,N}[\bar{B}|A]) \Pr_{W,N}[A] \ge \left(\frac{1}{2}\right) \Pr_{W,N}[A].\end{aligned}$$ □

Now we prove the following claim.

CLAIM B.4.

$$\Pr_{W,N}[B] \le g(d, 2m)\frac{\sum_{i=0}^{r-1}\binom{\epsilon m/2}{i}\binom{2m-\epsilon m/2}{m-i}}{\binom{2m}{m}}.$$

PROOF. For each $R \in \mathcal{R}_\epsilon$ let $B_R = [|R \cap N| < r \text{ and } |R \cap W| \ge \epsilon m/2]$. Then

$$B = \bigvee_{R\in\mathcal{R}_\epsilon} B_R.$$

Now, if we fix $R \in \mathcal{R}_\epsilon$ we have $\mathbf{E}_{W,N}[B_R] = \mathbf{E}_W[\mathbf{E}_N[B_R|W]]$. Now if we fix W, then the event A_R depends only on $R \cap W$. This is because $R \cap N = (R \cap W) \cap N$. So if, for R and R', we have $R \cap W = R' \cap W$, then B_R and $B_{R'}$ are the same event. By the Sauer lemma, the number of different events is at most

$$|\{R \cap W | R \in \mathcal{R}\}| \le |P_W(\mathcal{R})| \le g(d, 2m).$$

Now for a fixed W we have

$$\begin{aligned}
\mathbf{E}_{W,N}[B_R] &= \Pr[B_R] \\
&= \Pr\left[|R \cap N| < r \text{ and } |R \cap W| \ge \frac{\epsilon m}{2}\right] \\
&\le \Pr\left[|R \cap N| < r \;\middle|\; |R \cap W| \ge \frac{\epsilon m}{2}\right] \\
&\le \Pr\left[|R \cap N| < r \;\middle|\; |R \cap W| = \frac{\epsilon m}{2}\right] \\
&= \frac{\sum_{i=0}^{r-1}\binom{\epsilon m/2}{i}\binom{2m-\epsilon m/2}{m-i}}{\binom{2m}{m}}.
\end{aligned}$$

Since this is for any W, we now have

$$\begin{aligned}
\Pr[B] &= \mathbf{E}_{W,N}\left[\bigvee_{R\in\mathcal{R}_\epsilon} B_R\right] \\
&\le \mathbf{E}_W\left[\mathbf{E}_N\left[\sum_{R\in\mathcal{R}_\epsilon} B_R\right]\middle| W\right] \\
&\le g(d, 2m)\mathbf{E}_W[\mathbf{E}_N[B_R|W]]
\end{aligned}$$

$$\leq g(d, 2m)\frac{\sum_{i=0}^{r-1}\binom{\epsilon m/2}{i}\binom{2m-\epsilon m/2}{m-i}}{\binom{2m}{m}}. \qquad \square$$

We now combine Claims B.3 and B.4 to complete the proof of Lemma B.1. So

$$\Pr_N[A] \leq 2\Pr_{W,n}[B] \leq 2g(d, 2m)\frac{\sum_{i=0}^{r-1}\binom{\epsilon m/2}{i}\binom{2m-\epsilon m/2}{m-i}}{\binom{2m}{m}}$$

completing the proof of Lemma B.1. □

In order to find an m that satisfies the inequality in Lemma B.1, we prove the following propositions.

PROPOSITION B.5.

$$\frac{\binom{2m-\epsilon m/2}{m}}{\binom{2m}{m}} \leq 2^{-\epsilon m/2}.$$

PROOF. We have

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

and since $(m-i)/(2m-i) \leq 1/2$ the result follows. □

PROPOSITION B.6. *For $\epsilon m > 16$ and $r \leq \epsilon m/5$, we have*

$$\frac{\sum_{i=0}^{r-1}\binom{\epsilon m/2}{i}\binom{2m-\epsilon m/2}{m-i}}{\binom{2m}{m}} \leq 2^{-\epsilon m/4}.$$

PROOF

$$\frac{\sum_{i=0}^{r-1}\binom{\epsilon m/2}{i}\binom{2m-\epsilon m/2}{m-i}}{\binom{2m}{m}} = \sum_{i=0}^{r-1}\frac{\binom{\epsilon m/2}{i}\binom{2m-\epsilon m/2}{m-i}}{\binom{2m-\epsilon m/2}{m}}\,\frac{\binom{2m-\epsilon m/2}{m}}{\binom{2m}{m}}$$

$$\le \sum_{i=0}^{r-1}\frac{(\epsilon m/2)(\epsilon m/2-1)\cdots(\epsilon m/2-i+1)}{(m-\epsilon m/2+1)(m-\epsilon m/2+2)\cdots(m-\epsilon m/2+i)}$$

$$\cdot\binom{m}{i}2^{-\epsilon m/2}$$

$$\le \sum_{i=0}^{r-1}\binom{m}{i}2^{-\epsilon m/2} \le \sum_{i=0}^{r-1}(\epsilon m)^i 2^{-\epsilon m/2} \le 2^{-\epsilon m/2 + r\lg(\epsilon m)} \le 2^{-\epsilon m/4}$$

□

We now are ready to prove Lemma B.2.

PROOF OF LEMMA B.2. By Propositions B.5 and B.6, we have

$$2g(d,2m)\frac{\sum_{i=0}^{r-1}\binom{\epsilon m/2}{i}\binom{2m-\epsilon m/2}{m-i}}{\binom{2m}{m}} \le 2\left(\frac{2em}{d}\right)^{d+1}2^{-\epsilon m/4}$$

Therefore it is sufficient if

$$2\left(\frac{2em}{d}\right)^{d+1}2^{-\epsilon m/4} \le \delta.$$

Taking logarithms yields

$$\frac{\epsilon m}{4} \ge (d+1)\lg\frac{2em}{d} + \lg\frac{2}{\delta}.$$

Hence, it suffices if

$$\frac{\epsilon m}{4} \ge 2(d+1)\lg\frac{2em}{d} \text{ and } \frac{\epsilon m}{4} \ge \lg\frac{4}{\delta}.$$

The latter inequality gives $m = O((1/\epsilon)\log(1/\delta))$. The former gives $m = O((d/\epsilon)\log(1/\epsilon))$. □

ACKNOWLEDGMENTS. We thank Lisa Hellerstein for her assistance with a proof in Section 7.

REFERENCES

ANGLUIN, D. 1988. Queries and concept learning. *Mach. Learn. 2*, 4, 319–342.

ANGLUIN, D., AND LAIRD, P. 1988. Learning from noisy examples. *Mach. Learn. 2*, 4, 343–370.

ASLAM, J. A., AND DECATUR, S. E. 1993. General bounds on statistical query learning and PAC learning with noise via hypothesis boosting. In *Proceedings of the 34th Annual Symposium on Foundations of Computer Science* (Nov.). pp. 282–291.

ASLAM, J. A., AND DECATUR, S. E. 1995. Specification and simulation of statistical query algorithms for efficiency and noise tolerance. In *Proceedings of the 8th Annual ACM Conference on Computational Learning Theory* (Santa Cruz, Calif., July 5–8). ACM, New York, pp. 437–446.

AUER, P., KWEK, S., MAASS, W., AND WARMUTH, M. K. 1996. Learning of depth two neural networks with constant fan-in at the hidden nodes. In *Proceedings of the 9th Annual ACM Conference on Computational Learning Theory* (Desenzano del Garda, Italy, June 28–July 1). ACM, New York, pp. 333–343.

BAUM, E. B. 1990a. On learning a union of half spaces. *J. Complexity 6*, 1 (March), 67–101.

BAUM, E. B. 1990b. The perceptron algorithm is fast for nonmalicious distributions. *Neural Comput. 2*, 248–260.

BAUM, E. B. 1991. Neural net algorithms that learn in polynomial time from examples and queries. *IEEE Trans. Neural Netw. 2*, 1 (Jan.), 5–19.

BEN-DAVID, S., BSHOUTY, N., AND KUSHILEVITZ, E. 1997. A composition theorem for learning algorithms with applications to geometric concept classes. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing* (El Paso, Tex. May 4–6). ACM, New York, pp. 324–333.

BERGER, B., ROMPEL, J., AND SHOR, P. W. 1994. Efficient NC algorithms for set cover with applications to learning and geometry. *J. Comput. Syst. Sci. 49*, 2, 454–477.

BLUM, A., AND RIVEST, R. L. 1989. Training a 3-node neural net is NP-Complete. In *Advances in Neural Information Processing Systems I.* Morgan-Kaufmann, San Mateo, Calif., pp. 494–501.

BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., AND WARMUTH, M. K. 1987. Occam's razor. *Inf. Proc. Lett. 24*, 377–380.

BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., AND WARMUTH, M. K. 1989. Learnability and the Vapnik-Chervonenkis dimension. *J. ACM 36*, 4 (Oct.), 929–965.

BRÖNNIMANN, H., AND GOODRICH, M. T. 1994. Almost optimal set covers in finite VC-dimension. In *Proceedings of the 10th Annual Symposium on Computational Geometry* (May). pp. 293–302.

BSHOUTY, N. 1998. A new composition theorem for learning algorithms. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing* (Dallas, Tex., May 23–26). ACM, New York, pp. 583–589.

BSHOUTY, N., CHEN, Z., AND HOMER, S. 1994. On learning discretized geometric concepts. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science* (Oct.). IEEE Computer Science Press, Los Alamitos, Calif., pp. 54–63.

BSHOUTY, N. H., GOLDBERG, P. W., GOLDMAN, S. A., AND MATHIAS, H. D. 1999. Exact learning of discretized geometric concepts. *SIAM J. Comput. 28,* 2, 674–699.

BSHOUTY, N. H., GOLDMAN, S. A., AND MATHIAS, H. D. 1998. Noise-tolerant parallel learning of geometric concepts. *Inf. Comput.,* to appear.

CHEN, Z., AND HOMER, S. 1994. The bounded injury priority method and the learnability of unions of rectangles. Unpublished manuscript.

CHVATAL, V. 1979. A greedy heuristic for the set covering problem. *Math. Op. Res. 4*, 3, 233–235.

DECATUR, S. E. 1993. Statistical queries and faulty PAC oracles. In *Proceedings of the 6th Annual ACM Conference on Computational Learning Theory* (Santa Cruz, Calif., July 26–28). ACM, New York, pp. 262–268.

EDELSBRUNNER, H. 1987. *Algorithms in Combinatorial Geometry.* Springer-Verlag, New York.

EHRENFEUCHT, A., AND HAUSSLER, D. 1989. Learning decision trees from random examples. *Inf. Comput. 82*, 3 (Sept.), 231–246.

FRAZIER, M., GOLDMAN, S., MISHRA, N., AND PITT, L. 1996. Learning from a consistently ignorant teacher. *J. Comput. Syst. Sci. 52,* 3 (June), 472–492.

GOLDBERG, P. W., GOLDMAN, S. A., AND MATHIAS, H. D. 1993. Learning unions of rectangles with membership and equivalence queries. Tech. Rep. WUCS-93-46 (Nov.), Washington Univ., St. Louis, MO.

HAUSSLER, D. 1989. Generalizing the PAC model: sample size bounds from metric dimension-based uniform convergence results. In *Proceedings of the 30th Annual Symposium on Foundations of Computer Science* (Oct.). IEEE Computer Society Press, Los Alamitos, Calif., pp. 40–45.

HAUSSLER, D., AND WELZL, E. 1987. Epsilon-nets and simplex range queries. *Disc. Comput. Geom. 2*, 127–151.

JOHNSON, D. S. 1974. Approximation algorithms for combinatorial problems. *J. Comput. Syst. Sci.* 256–278.

KEARNS, M. 1993. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing* (San Diego, Calif., May 16–18). ACM, New York, pp. 392–401.

KWEK, S., AND PITT, L. 1996. PAC learning intersections of halfspaces with membership queries. In *Proceedings of the 9th Annual ACM Conference on Computational Learning Theory* (Desenzuno del Garda, Italy, June 28–July 1). ACM, New York, pp. 244–254.

LONG, P. M., AND WARMUTH, M. K. 1994. Composite geometric concepts and polynomial predictability. *Inf. Comput. 113,* 2, 203–252.

LOVÁSZ, L. 1975. On the ratio of optimal integral and fractional covers. *Disc. Math. 13*, 383–390.

SLOAN, R. H. 1995. Four types of noise in data for PAC learning. *Inf. Proc. Lett. 54,* 157–162.

VALIANT, L. G. 1984. A theory of the learnable. *Commun. ACM 27*, 11 (Nov.), 1134–1142.

VALIANT, L. 1985. Learning disjunctions of conjunctions. In *Proceedings of 9th International Joint Conference on Artificial Intelligence*.

VAPNIK, V. N., AND CHERVONENKIS, A. Y. 1971. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications XVI*, 2, 264–280.

WILLARD, D. E. 1982. Polygon retrieval. *SIAM J. Comput. 11*, 1 (Feb.), 149–165.

RECEIVED JULY 1996; REVISED MARCH 1997; ACCEPTED JUNE 1998