

Washington University in St. Louis

Washington University Open Scholarship

All Computer Science and Engineering
Research

Computer Science and Engineering

Report Number: WUCS-97-19

1997-01-01

Noise-Tolerant Distribution-Free Learning of General Geometric Concepts

Nader H. Bshouty, Sally A. Goldman, H. David Mathias, Subhash Suri, and Hisao Tamaki

We present an efficient algorithm for PAC-learning a very general class of geometric concepts over R^d for fixed d . More specifically, let T be any set of s halfspaces. Let $x = (x_1, \dots, x_d)$ be an arbitrary point in R^d . With each $t \in T$ 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 C_d that we study consists of all concepts formed by any boolean function over I_{t_1}, \dots, I_{t_s} for $t_i \in T$. This class is much more general than any geometric concept class known... [Read complete abstract on page 2.](#)

Follow this and additional works at: https://openscholarship.wustl.edu/cse_research



Part of the [Computer Engineering Commons](#), and the [Computer Sciences Commons](#)

Recommended Citation

Bshouty, Nader H.; Goldman, Sally A.; Mathias, H. David; Suri, Subhash; and Tamaki, Hisao, "Noise-Tolerant Distribution-Free Learning of General Geometric Concepts" Report Number: WUCS-97-19 (1997). *All Computer Science and Engineering Research*.
https://openscholarship.wustl.edu/cse_research/435

Department of Computer Science & Engineering - Washington University in St. Louis
Campus Box 1045 - St. Louis, MO - 63130 - ph: (314) 935-6160.

Noise-Tolerant Distribution-Free Learning of General Geometric Concepts

Nader H. Bshouty, Sally A. Goldman, H. David Mathias, Subhash Suri, and Hisao Tamaki

Complete Abstract:

We present an efficient algorithm for PAC-learning a very general class of geometric concepts over \mathbb{R}^d for fixed d . More specifically, let T be any set of s halfspaces. Let $x = (x_1, \dots, x_d)$ be an arbitrary point in \mathbb{R}^d . With each $t \in T$ 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 C_d that we study consists of all concepts formed by any boolean function over I_{t_1}, \dots, I_{t_s} for $t_i \in T$. 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 C over \mathbb{R}^d given that the VC-dimension of C has dependence only on d and there is a polynomial time algorithm to determine if there is a concept from 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.

**Noise-Tolerant Distribution-Free Learning of
General Geometric Concepts**

**Nader H. Bshouty, Sally A. Goldman, H. David
Mathias, Subhash Suri and Hisao Tamaki**

WUCS-97-19

April 1997

**Department of Computer Science
Washington University
Campus Box 1045
One Brookings Drive
St. Louis MO 63130**

Noise-Tolerant Distribution-Free Learning of General Geometric Concepts

Nader H. Bshouty

The University of Calgary, Calgary, Alberta, Canada T2N 1N4

and

Sally A. Goldman

Washington University, St. Louis, MO 63130

and

H. David Mathias

Washington University, St. Louis, MO 63130

and

Subhash Suri

Washington University, St. Louis, MO 63130

and

Hisao Tamaki

IBM Tokyo Research Laboratory, Yamato 242, Japan

We present an efficient algorithm for PAC-learning a very general class of geometric concepts over \mathbb{R}^d for fixed d . More specifically, let \mathcal{T} be any set of s halfspaces. Let $x = (x_1, \dots, x_d)$ be an arbitrary point in \mathbb{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}, \dots, 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 \mathbb{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 [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 \mathbb{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 $x = (x_1, \dots, x_d)$ in \mathbb{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}, \dots, 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. 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 efficiently learn any boolean combination of a polynomial number of concepts selected from any concept class \mathcal{C} over \mathbb{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 in the labels for any noise rate strictly less than $1/2$. This variation takes advantage of the noise tolerance inherent in the statistical (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;

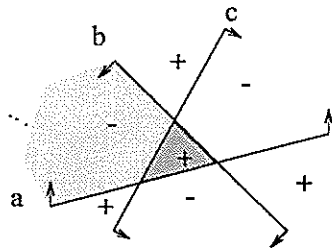


Fig. 1. A target concept from \mathcal{C}_3^2 formed from the three halfspaces a , b , and c . For any point x in the lightly shaded region $I_a(x) = I_b(x) = 1$ and $I_c(x) = 0$. Likewise, for any point x' in the darkly shaded region $I_a(x') = I_b(x') = I_c(x') = 1$. The boolean function over I_a, 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.

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, Ehrenfeucht, Haussler, and Warmuth [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 \mathcal{F} of halfspaces such that each possible partition of the points in S into two sets by a halfspace is realized. Then 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 $f \in \mathcal{F}$ 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, Ehrenfeucht, Haussler, and Warmuth [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. Thus for this situation no covering of the form used above is possible.

This standard covering technique can be applied to learn unions (and intersections) of base classes, \mathcal{C} , other than halfspaces. However, for a concept class \mathcal{C} , $|\mathcal{F}|$ has an exponential

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

dependence on the VC-dimension of \mathcal{C} . Thus, this approach yields a polynomial time algorithm only when the VC-dimension of \mathcal{C} is constant. Blumer, Ehrenfeucht, Haussler, and Warmuth [1989] prove that this set covering approach can be used to efficiently PAC-learn the union (or intersection) of concepts from a base class \mathcal{C} given that the VC-dimension of \mathcal{C} 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. Our technique introduced here, generalizes this standard technique by allowing us to learn classes defined by arbitrary boolean combinations of concepts from \mathcal{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 sub-linear in m . When learning a union it is easily seen that each of the s concepts in \mathcal{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.

There are several important contributions of this paper. First, we give an algorithm to learn \mathcal{C}_s^d in the PAC model². Our work introduces a new learning technique that uses the following novel, though simple, application of set covering. Presented with a labeled sample of points we know that any positively labeled point must be separated from all negatively labeled points by some subset of the hyperplanes corresponding to the halfspaces defining the target concept. We construct from the sample a complete bipartite graph by adding an edge between every pair of points (x_+, x_-) 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

²While we use \mathcal{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 \mathcal{C} over \mathbb{R}^d given that the VC-dimension of \mathcal{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 \mathcal{C} consistent with a given set of labeled examples.

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 sample³. By applying a greedy covering technique, we obtain a set of hyperplanes that separates every pair (x_+, x_-) . 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 4. Furthermore, 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.

3. 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

³Applying a result from Blumer, Ehrenfeucht, Haussler, and Warmuth [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 5 we describe a more efficient approach based on a geometric duality argument.

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.

The paper of Blumer, Ehrenfeucht, Haussler, and Warmuth [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 $\text{VCD}(\mathcal{C})$) 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.

The results of Blumer *et al.* 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 $\max\left(\frac{4}{\epsilon} \log \frac{2}{\delta}, \frac{8d}{\epsilon} \log \frac{13}{\epsilon}\right)$ 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 $\Omega\left(\frac{1}{\epsilon} \log \frac{1}{\delta} + \frac{d}{\epsilon}\right)$ 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. Blumer, Ehrenfeucht, Haussler, and Warmuth [1987, 1989] show that finding a hypothesis whose size is sublinear in the sample size is sufficient to guarantee polynomial PAC learnability. Let $\mathcal{H}_{s,m}^A$ be the hypothesis space used by algorithm A for a target of complexity s and sample size m . 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 $\mathcal{H}_{s,m}^A$ is at most $p(s)m^\alpha$.

THEOREM 2. ([Blumer et al. 1989]) *Let A be a learning algorithm for concept class \mathcal{C} that has hypothesis space $\mathcal{H}_{s,m}^A$. If the VC dimension of $\mathcal{H}_{s,m}^A$ 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 assume 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 1988], 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 [Kearns 1993; Decatur 1993; Aslam and Decatur 1993; Aslam and Decatur 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 $\text{rel-STAT}_{\mathcal{D}}(\chi, \mu, \theta)$ 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 $\text{rel-STAT}_{\mathcal{D}}(\chi, \mu, \theta)$ may return \perp . If \perp is not returned, then $\text{rel-STAT}_{\mathcal{D}}(\chi, \mu, \theta)$ must return an estimate \hat{P}_{χ} 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 $\eta < 1/2$. They have also shown that relative error SQ algorithms are robust against small amounts of malicious errors.

Let \mathcal{Z} be the set of integers. Let $\vec{y} = (y_1, \dots, y_d)$ denote the d dimension variables and $\vec{x} = (x_1, \dots, x_d)$ denote an element of $[0, 1]^d$, the instance space. Let $\vec{a} = (a_1, \dots, a_d)$ be a vector where $a_i \in \mathcal{Z}$. A d -dimensional *hyperplane* is $\{\vec{y} \mid \vec{a} \cdot \vec{y} = b\}$ for some $b \in \mathcal{Z}$. A d -dimensional *halfspace* is $\{\vec{y} \mid \vec{a} \cdot \vec{y} \succ b\}$ where $\succ \in \{>, \geq, <, \leq\}$ and b is a constant. Let H be a finite set of hyperplanes in $[0, 1]^d$. The *arrangement* $\mathcal{A}(H)$ 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 $\mathcal{A}(H)$. For each set S of sample points, we say that two hyperplanes p_i and p_j are equivalent 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 \mathcal{U} and a family \mathcal{F} of subsets of \mathcal{U} such that $\bigcup_{f \in \mathcal{F}} f = \mathcal{U}$. A solution is a smallest cardinality subset \mathcal{G} of \mathcal{F} such that $\bigcup_{g \in \mathcal{G}} g = \mathcal{U}$. The greedy approximation algorithm [Chvatal 1979] for this problem has a ratio bound of $\ln |\mathcal{U}| + 1$ on the size of the approximation.

4. 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 proper⁴ learning algorithm for unions of s halfspaces, unless $RP = NP$. (That is, any such algorithm must have exponential 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, Ehrenfeucht, Haussler, and Warmuth [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, Goldman, and Mathias [1995] have given noise-tolerant algorithms for several geometric classes. In particular, they studied 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.* present an algorithm to PAC-learn an s -fold union of boxes in E^d by drawing a sufficiently large sample of size $m = \text{poly}(\frac{1}{\epsilon}, \lg \frac{1}{\delta}, s, d)$, and then performing a greedy covering over the at most $(\frac{\epsilon m}{2d})^{2d}$ boxes defined by the sample. Thus for d constant this algorithm runs in polynomial time. Long and Warmuth [1990] present an algorithm to PAC-learn this same class by again drawing a sufficiently large sample and constructing a hypothesis 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 $s, d^s, \frac{1}{\epsilon}$, and $\lg \frac{1}{\delta}$. So for s constant this yields an efficient PAC algorithm. We note that either of these PAC algorithms can

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

be applied to the class $\bigcup_m \text{BOX}_n^d$ giving efficient PAC algorithms for this class for either d constant or s constant.

Finally, 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 \mathfrak{R}^n in time polynomial in s, n . Recently, Kwek and Pitt [1996] give a result in the standard PAC-model with membership queries to learn the union of s halfspaces in \mathfrak{R}^n in time polynomial in s, n , and a parameter that characterizes the number of bits of accuracy with which the target hyperplanes are specified.

Also, Frazier, Goldman, Mishra, and Pitt [1994] 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.

Recall that since $\bigcup_m \text{BOX}_n^d$ generalizes DNF, a polynomial-time algorithm for arbitrary d and s would solve the problem of learning DNF. Observe that the class considered by Frazier *et al.* is a generalization of the class of DNF formulas in which all variables only appear negated. Bshouty, Chen, and Homer [1994] give PAC algorithm to learn the discretized version of R-linear geometric concepts with random classification noise.

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, Chen, and Homer 1994; Bshouty, Goldberg, Goldman, and Mathias 1994]. While most such work assumes that there are a constant number of dimensions, recently Auer, Kwek, Maass, and Warmuth [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.

5. 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 6 we convert this algorithm to an algorithm in the relative statistical query model.

5.1 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 4, we show that this sample complexity can be improved. (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 describe shortly 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, Ehrenfeucht, Haussler, and Warmuth [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

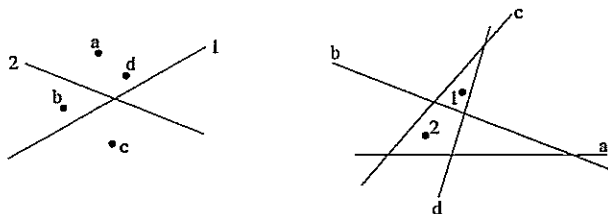


Fig. 2. A 2-dimensional primal space (left) and its 2-dimensional dual space (right).

partition of S . To ensure 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 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, for example, Edelsbrunner [1987]). Each point p in our original or *primal space* is mapped into 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.

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.

5.2 Analysis

In this section we prove the following result.

THEOREM 3. *There exists an algorithm that PAC-learns C_s^d . The algorithm draws a sample of size*

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

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 γ of each feature in the dual space

$\mathcal{F} \leftarrow \mathcal{F} \cup \{g'(\gamma)\}$

$h \leftarrow$ greedy covering of \mathcal{U} using \mathcal{F}

For each region $\rho \in h$

 If $\rho \cap S \neq \emptyset$ then label ρ according to S

 Else label ρ arbitrarily

Return h

Fig. 3. Pseudocode for the algorithm to learn \mathcal{C}_s^d .

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

PROOF. 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 the results of Blumer, Ehrenfeucht, Haussler, and Warmuth [1989], the VC dimension of the hypothesis class is at most $O(ds \lg m \lg(s \lg m)) = O(ds(\lg m)^2)$. Thus by applying Theorem 2 [Blumer *et al.*] 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 $0 \leq i \leq d$, 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 $O(\sum_{p \in \mathcal{F}} |p|)$, 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. \square

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.*]).

COROLLARY 4. *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.* instead of using the geometric duality construction to build \mathcal{F} we get the following generalization.

COROLLARY 5. *Let \mathcal{C} be a concept class over \mathbb{R}^d for which $\text{VCD}(\mathcal{C})$ 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

is 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 $\eta < 1/2$.

6. 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. 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 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 can not just use the standard method of transforming a set covering based algorithm into a SQ algorithm.

6.1 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 \mathfrak{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} . 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 . Let χ_1 be the predicate " x_+ and x_- are in same region of h " and let χ_2 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 6.2). We first use the rel-STAT oracle to estimate $P_{\chi_1} = E_{+/-}(h)$. Thus, by the definition of the rel-STAT oracle, the estimate \hat{P}_{χ_1} returned (if not \perp) by $\text{rel-STAT}_{\mathcal{D}'}(\chi_1, 1/2, \frac{2\epsilon}{t^{d+1}})$ satisfies

$$\frac{1}{2}P_{\chi_1} \leq \hat{P}_{\chi_1} \leq \frac{3}{2}P_{\chi_1}.$$

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

If $\hat{P}_{\chi_1} > \frac{\epsilon}{(t^d+1)}$ 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 $\text{rel-STAT}_{\mathcal{D}'}$ $(\chi_1 \wedge \chi_2, 1/2, \frac{\epsilon}{4s(t^d+1)})$ 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 . We then add to h the hyperplane $p \in \mathcal{F}$ for which the highest estimate was returned. In Section 6.2, we prove that $t = \max \{(3sd)^{3/2}, 3s \ln \frac{8}{\epsilon}\}$ rounds are sufficient to ensure that $E_{+/-}(h) = P_{\chi_1} \leq \frac{2\epsilon}{(t^d+1)}$. Thus after at most t repetitions of the repeat loop we are guaranteed that $P_{\chi_1} \leq \frac{2\epsilon}{(t^d+1)}$. 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 $\epsilon/2$.

Notice that fewer than t covering rounds are used if the $+/-$ error becomes sufficiently low. If we did not stop at this point then the threshold for the statistical query to estimate $P_{\chi_1 \wedge \chi_2}$ would need to be too small. Note that as long as the positive and negative weight in \mathcal{D} is greater than $\epsilon/2$ it is also easy to see that \mathcal{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 \mathcal{D} is too small.

Finally, for each region ρ of h , we use statistical queries of the form $\text{rel-STAT}_{\mathcal{D}}((x \in \rho \text{ and } f(x) = 1), 1/4, 1/2)$ 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.

6.2 Analysis

In this section we prove the following theorem.

THEOREM 6. *There exists an algorithm to PAC-learn \mathcal{C}_s^d in the presence of random classification noise with a rate $\frac{1}{2} - \gamma$ ($\gamma > 0$), with time and sample complexity polynomial in s , $1/\epsilon$ 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

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

$h = \emptyset$

Repeat t times

If $\text{rel-STAT}_{\mathcal{D}'}(\chi_1, 1/2, \frac{2\epsilon}{t^d+1}) > \frac{\epsilon}{(t^d+1)}$

$$p_{max} = \operatorname{argmax}_{p \in \mathcal{F}} \left\{ \text{rel-STAT}_{\mathcal{D}'} \left(\chi_1 \wedge \chi_2, 1/2, \frac{\epsilon}{4s(t^d+1)} \right) \right\}$$

$h \leftarrow h \cup p_{max}$

$\mathcal{F} \leftarrow \mathcal{F} - \{p_{max}\}$

Else exit the repeat loop

For each region ρ of h

If $\text{rel-STAT}_{\mathcal{D}}(\chi_3, 1/4, 1/2) \geq 1/2$

Label ρ positive

Else

Label ρ negative

Return h

Fig. 4. Our noise-tolerant algorithm for \mathcal{C}_s^d . Recall that $t = \max \{(3sd)^{3/2}, 3s \ln \frac{6}{\epsilon}\}$, χ_1 is the predicate “ x_+ and x_- are in same region of h ” and χ_2 is the predicate “ x_+ and x_- separated by hyperplane p ”. χ_3 is the predicate “ $x \in \rho$ and $f(x) = 1$ ”. Let $\langle x_+, x_- \rangle$ denote a pair of examples where x_+ is a random positive example and x_- is a random negative example, each drawn according to \mathcal{D} . Recall that \mathcal{D}' outputs $\langle x_+, x_- \rangle$.

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 the $E_{+/-}(h) > \frac{4\epsilon(1-\epsilon)}{r}$ 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 . Thus we have that $E_{+/-}(h)$

$$\begin{aligned} &= \sum_{i=1}^r Pr_{x \in \mathcal{D}}(x \in i | f(x) = 1) \cdot Pr_{x \in \mathcal{D}}(x \in i | f(x) = 0) \\ &= \sum_{i=1}^r \frac{Pr_{x \in \mathcal{D}}(x \in i \wedge f(x) = 1)}{P_+} \cdot \frac{Pr_{x \in \mathcal{D}}(x \in i \wedge f(x) = 0)}{P_-} \\ &= \sum_{i=1}^r \frac{p_i}{P_+} \cdot \frac{n_i}{(1 - P_+)} \geq 4 \sum_{i=1}^r p_i n_i > \frac{4\epsilon(1 - \epsilon)}{r} \end{aligned}$$

where the last step is shown using Lagrange multipliers. Note that p_i and n_i are simply notation for the probabilities in the numerators of line 2. \square

Therefore, we know that if $E_C(h) > \frac{\epsilon}{2}$, 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 $E_C(h) \leq \frac{\epsilon}{2}$. 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.

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

Let e_i be the value $E_{+/-}(h)$ after the i th 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 $\frac{3\epsilon}{4(t^d + 1)}$, 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 $\frac{3\epsilon}{4(t^d + 1)}$. Thus, there exist s hyperplanes in \mathcal{F} that have $+/-$ error at most $\frac{3\epsilon}{4(t^d + 1)}$. By an averaging argument, when selecting the i th hyperplane to add, there exists a $p \in \mathcal{F}$ such that

$$\begin{aligned} P_p &= Pr_{(x_+, x_-) \in \mathcal{D}'}((p \text{ separates } x_+ \text{ and } x_-) \wedge (x_+ \text{ and } x_- \text{ are not separated by } h)) \\ &\geq \left(e_i - \frac{3\epsilon}{4(t^d + 1)} \right) \frac{1}{s}. \end{aligned}$$

Since we stay in the repeat loop only if $e_i \geq \frac{\epsilon}{(t^d + 1)}$ we know that $P_p \geq \left(\frac{\epsilon}{(t^d + 1)} - \frac{3\epsilon}{4(t^d + 1)} \right) \frac{1}{s} = \frac{\epsilon}{4s(t^d + 1)}$. Thus, we let $\theta = \frac{\epsilon}{4s(t^d + 1)}$ for the statistical queries to estimate P_p since we need no estimates when $P_p < \theta$ (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 $t = \max \{(3sd)^{3/2}, 3s \ln \frac{8}{\epsilon}\}$ suffices to ensure that $E_{+/-}(h) \leq \frac{2\epsilon}{t^d + 1}$, and thus, $E_C(h) \leq \frac{\epsilon}{2}$.*

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

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

Thus for p_{max} , $\hat{P}_{p_{max}} \geq \left(e_i - \frac{3\epsilon}{4(t^d + 1)} \right) \frac{1}{2s}$. Thus since 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 $e_i \leq (1 - \frac{1}{3s})^i + \frac{3\epsilon}{4(t^d + 1)}$. To ensure that $E_C(h) \leq \frac{\epsilon}{2}$ it suffices to select t such that

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

Since $2\epsilon(1 - \frac{\epsilon}{2}) \geq \epsilon$ it suffices to let

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

It can be shown that $t = \max \{ (3sd)^{3/2}, 3s \ln \frac{8}{\epsilon} \}$ is sufficient provided that $s \geq 4$ and $d \geq 2$. We note that increasing to 2 the exponent on the first term is sufficient to ensure a valid value for t for smaller values of s and d . \square

We are now ready to prove the main result of this section.

PROOF THEOREM 6. 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 $E_{+/-}(h) \leq \frac{2\epsilon}{t^d + 1}$, 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 $P_{X_3} \leq \epsilon/2$ or $P_{X_3} \geq 1 - \epsilon/2$. Without loss of generality we assume that $0 < \epsilon < 1/2$. Then we have that either $P_{X_3} \leq 1/4$ or $P_{X_3} \geq 3/4$. If $P_{X_3} \leq 1/4$, then $\hat{P}_{X_3} \leq 5/16 < 1/2$ and thus the region is correctly classified as negative. Likewise, if $P_{X_3} \geq 3/4$, then $\hat{P}_{X_3} \geq 9/16 > 1/3$ and thus the region is correctly classified as positive. Finally, if \perp is returned then $P_{X_3} < 1/2$ 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

$$t = \max \left\{ (3sd)^{3/2}, 3s \ln \frac{8}{\epsilon} \right\}$$

halfspaces (obtained by arbitrarily orienting each of the hyperplanes), the $\text{VCD}(\mathcal{H})$ is at most $2(d+1)t \lg(3t)$ (by [Blumer et al. 1989]). After substituting the above values for ϵ and δ , $m_u = \frac{16(t^d+1)}{3\epsilon} \cdot \left(\text{VCD}(\mathcal{H}) \lg \frac{48(t^d+1)}{3\epsilon} + \lg \frac{6}{\delta} \right)$ is sufficient to ensure a hypothesis consistent with the sample has error greater than $\frac{3\epsilon}{4(t^d+1)}$ with probability at most $\delta/3$.

Finally, we consider the time complexity of this algorithm. By the argument presented in the proof of Theorem 3, 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. \square

7. 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 $\frac{1}{2} - \gamma$ (for $\gamma > 0$), the learner takes $m_q = \text{poly}(1/\epsilon, 1/\delta, 1/\gamma, s)$ labeled examples A 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 divides 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 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 $sm^\alpha\xi = \tilde{O}\left(\frac{s}{m^{1-\alpha}}\right) \leq \frac{\epsilon}{2}$. So it is enough to choose $m = \tilde{O}\left(\frac{s}{\epsilon}\right)^{\frac{1}{1-\alpha}}$. 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 non-intersected regions will be correct and thus the total error is at most ϵ .

8. 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 halfspaces, is the most general geometric class for which we know algorithms to exist. 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 efficiently learn any boolean combination of a polynomial number of concepts selected from any concept class \mathcal{C} given that the VC-dimension of \mathcal{C} is constant (for d 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 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.

One significant open problem is whether or not an algorithm for \mathcal{C}_ϵ^d exists in the query learning model of Angluin [1988]. Since in this model the learner is required to achieve exact identification of the target concept (as opposed to the approximation achieved in the PAC model), the domain must be discretized. We currently know of no efficient query algorithms for this class.

ACKNOWLEDGMENTS

The research of Nader Bshouty was supported in part by the NSERC of Canada. The research of Sally Goldman and David Mathis was supported in part by NSF NYI Grant CCR-9357707 with matching funds provided by Xerox PARC and WUTA. The research of Subash Suri was supported in part by NSF Grant CCR-9501494.

Sally Goldman would like to dedicate this paper to Paris Kanellakis, who first introduced her to computer science research. We thank Lisa Hellerstein for her assistance with a proof in Section 6.

References

- ALON, N., SPENCER, J. H., AND ERDŐS, P. 1992. *The Probabilistic Method*. Wiley.
- ANGLUIN, D. 1988. Queries and concept learning. *Machine Learning* 2, 4, 319–342.
- ANGLUIN, D. AND LAIRD, P. 1988. Learning from noisy examples. *Machine Learning* 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 *34th Annual Symposium on Foundations of Computer Science* (Nov. 1993). 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 Eighth Annual ACM Conference on Computational Learning Theory* (July 1995). pp. 437–446.
- AUER, P., KWEK, S., MAASS, W., AND WARMUTH, M. 1996. Learning of depth two neural networks with constant fan-in at the hidden nodes. In *Proceedings of the Ninth Annual ACM Conference on Computational Learning Theory* (June 1996). pp. 333–343.
- BAUM, E. B. 1990a. On learning a union of half spaces. *Journal of Complexity* 6, 1 (March), 67–101.
- BAUM, E. B. 1990b. The perceptron algorithm is fast for nonmalicious distributions. *Neural Computation* 2, 248–260.
- BAUM, E. B. 1991. Neural net algorithms that learn in polynomial time from examples and queries. *IEEE Transactions on Neural Networks* 2, 1 (Jan.), 5–19.
- BERGER, B., ROMPEL, J., AND SHOR, P. W. 1994. Efficient NC algorithms for set cover with applications to learning and geometry. *Journal of Computer and System Sciences* 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*, pp. 494–501. Morgan Kaufmann.
- BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., AND WARMUTH, M. K. 1987. Occam’s razor. *Information Processing Letters* 24, 377–380.
- BLUMER, A., EHRENFEUCHT, A., HAUSSLER, D., AND WARMUTH, M. K. 1989. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the Association for Computing Machinery* 36, 4 (Oct.), 929–965.
- BRÖNNIMANN, H. AND GOODRICH, M. T. 1994. Almost optimal set covers in finite VC-dimension. In *10th Annual Symposium on Computational Geometry* (May 1994). pp. 293–302.
- BSSHOUTY, N. H., CHEN, Z., AND HOMER, S. 1994. On learning discretized geometric concepts. In *35th Annual Symposium on Foundations of Computer Science* (Nov. 1994). pp. 54–63.
- BSSHOUTY, N. H., GOLDMAN, S. A., AND MATHIAS, H. D. 1995. Noise-tolerant parallel learning of geometric concepts. In *Proceedings of the Eighth Annual ACM Conference on Computational Learning Theory* (July 1995). pp. 345–352.
- 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. *Mathematics of Operations Research* 4, 3, 233–235.

- DECATUR, S. E. 1993. Statistical queries and faulty PAC oracles. In *Proceedings of the Sixth Annual ACM Conference on Computational Learning Theory* (1993). ACM Press, New York, NY, 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. *Information and Computation* 82, 3 (Sept.), 231–246.
- FRAZIER, M., GOLDMAN, S., MISHRA, N., AND PITT, L. 1994. Learning from a consistently ignorant teacher. In *Proceedings of the Seventh Annual ACM Conference on Computational Learning Theory* (July 1994).
- GOLDBERG, P. W., GOLDMAN, S. A., AND MATHIAS, H. D. 1993. Learning unions of rectangles with membership and equivalence queries. Technical Report WUCS-93-46 (Nov.), Washington University.
- HAUSSLER, D. 1989. Generalizing the PAC model: sample size bounds from metric dimension-based uniform convergence results. In *30th Annual Symposium on Foundations of Computer Science* (Oct. 1989). pp. 40–45.
- HAUSSLER, D. AND WELZL, E. 1987. Epsilon-nets and simplex range queries. *Discrete Computational Geometry* 2, 127–151.
- JOHNSON, D. S. 1974. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 256–278.
- KEARNS, M. 1993. Efficient noise-tolerant learning from statistical queries. In *Proc. 25th Annu. ACM Sympos. Theory Comput.* (1993). ACM Press, New York, NY, pp. 392–401.
- KWEK, S. AND PITT, L. 1996. Pac learning intersections of halfspaces with membership queries. In *Proceedings of the Ninth Annual ACM Conference on Computational Learning Theory* (June 1996). pp. 244–254.
- LONG, P. M. AND WARMUTH, M. K. 1990. Composite geometric concepts and polynomial predictability. In *Proceedings of the Third Annual Workshop on Computational Learning Theory* (Aug. 1990). Morgan Kaufmann, pp. 273–287.
- LOVÁSZ, L. 1975. On the ratio of optimal integral and fractional covers. *Discrete Mathematics*, 383–390.
- SLOAN, R. H. 1988. Types of noise in data for concept learning. In *Proceedings of the 1988 Workshop on Computational Learning Theory* (1988). Morgan Kaufmann, pp. 91–96.
- VALIANT, L. 1984. A theory of the learnable. *Communications of the ACM* 27, 11 (Nov.), 1134–1142.
- VALIANT, L. 1985. Learning disjunctions of conjunctions. In *Proceedings of Ninth International Joint Conference on Artificial Intelligence* (1985).
- 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 Journal on Computing* 11, 1 (Feb.), 149–165.

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 $\Pr_{\mathcal{D}}[R] > \epsilon$ contains at least p points from N . For a range space (X, \mathcal{R}) with $|X| = n$ points of VC-dimension d , $|\mathcal{R}| \leq g(d, n) = \sum_{i=0}^d \binom{n}{i} \leq n^d$.

LEMMA 3. *Let (X, \mathcal{R}) be a range space of VC-dimension d . Let \mathcal{D} be a distribution over X . Suppose $0 < \epsilon, \delta < 1$. Let N be a sequence of points obtained by m random independent draws from X according to the distribution \mathcal{D} , where*

$$m \geq \max \left(\frac{4}{\epsilon} \lg \frac{2}{\delta}, \frac{8(d+p)}{\epsilon} \lg \frac{8(d+p)}{\epsilon} \right).$$

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

PROOF. We assume the reader is familiar with the standard ϵ -net proof (e.g. See Alon, Spencer, and Erdős [1992]).

Let $W = (z_1, \dots, z_{2m})$ be $2m$ random independent draws from X according to distribution \mathcal{D} . Let $N = (x_1, \dots, x_m)$ be m points from W and $T = (y_1, \dots, y_m)$ the other points in W . Notice that the probability space of N is equivalent to the space in which you just choose m random independent elements from X according to the distribution \mathcal{D} .

For a sequence $H = (h_1, \dots, h_r)$, we use $|H| = r$ to denote the length of the sequence H . For a set $A \subset X$, let $A \cap H$ be the sequence of all elements in the sequence H that are also in A .

Let \mathcal{R}_ϵ be the set of all $R \in \mathcal{R}$ with $\Pr_{\mathcal{D}}[R] > \epsilon$. Define the following events

$$A_1 = [(\exists R \in \mathcal{R}_\epsilon) |R \cap N| < p]$$

and

$$A_2 = [(\exists R \in \mathcal{R}_\epsilon) |R \cap N| < p \text{ and } |R \cap T| \geq (\epsilon m)/2].$$

□

CLAIM 1. $\Pr_{W,N}[A_2] \geq \frac{1}{2} \Pr_N[A_1]$.

PROOF.

$$\begin{aligned} \Pr_{W,N}[\overline{A_2} \mid A_1] &= \Pr[(\forall R \in \mathcal{R}_\epsilon) |R \cap N| \geq p \text{ or} \\ &\quad |R \cap T| < (\epsilon m)/2 \mid \\ &\quad (\exists R_0 \in \mathcal{R}_\epsilon) |R_0 \cap N| < p] \\ &\leq \Pr_T[|R_0 \cap T| < (\epsilon m)/2] \\ &\leq \frac{\mathbf{E}_T[|R_0 \cap T|]}{(\epsilon m)/2} \leq \frac{\epsilon m}{(\epsilon m/2)^2} \end{aligned}$$

$$= \frac{4}{\epsilon m} \leq \frac{1}{2}$$

where we use Chebyshev's inequality.

Now $\Pr_{W,N}[A_2] = \Pr_{W,N}[A_2 \wedge A_1] = \Pr_{W,N}[A_2 \mid A_1] \Pr_N[A_1] \geq \frac{1}{2} \Pr_N[A_1]$. \square

CLAIM 2. $\Pr_{W,N}[A_2] \leq g(d, 2m)g(p, 2m)2^{-\epsilon m/2}$.

PROOF. For each $R \in \mathcal{R}_\epsilon$ let $A_R = [|R \cap N| \leq p \text{ and } |R \cap T| > (\epsilon m)/2]$. Obviously, $A_2 = \bigvee_{R \in \mathcal{R}_\epsilon} A_R$. Notice that $E_{W,N}[A_R] = E_W[E_N[A_R|W]]$. By fixing the set of examples W , the event A_R depends only on $R \cap W$. This is because (N, T) satisfies A_R if and only if

$$|R \cap N| < p \text{ and } |R \cap T| \geq \frac{\epsilon m}{2}.$$

Therefore if $R \cap W = R' \cap W$ then

$$(R \cap T) \cup (R \cap N) = (R' \cap T) \cup (R' \cap N)$$

and since N, T are disjoint we have $(R \cap T) = (R' \cap T)$ and $(R \cap N) = (R' \cap N)$. Therefore both A_R and $A_{R'}$ are the same event. Now notice that the number of different events do not exceed the number of different $R \cap W$ where $R \in \mathcal{R}$. This number is

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

Now for a fixed W let $q = |R \cap W|$. Then

$$\begin{aligned} E_N[A_R] &= \Pr_N[A_R] \\ &= \Pr[|R \cap N| < p \text{ and } |R \cap T| > (\epsilon m)/2] \\ &\leq \Pr[|R \cap N| < p \text{ and } |R \cap W| > (\epsilon m)/2] \\ &\leq \Pr[|R \cap N| < p \mid |R \cap W| > (\epsilon m)/2] \\ &\leq \frac{\binom{2m-p}{m} \binom{p}{0} + \binom{2m-p}{m-1} \binom{p}{1} + \dots + \binom{2m-p}{m-q} \binom{p}{q}}{\binom{2m}{m}} \\ &\leq \frac{\binom{2m-p}{m} \left(\binom{p}{0} + \dots + \binom{p}{q} \right)}{\binom{2m}{m}} \\ &\leq g(p, q)2^{-\epsilon m/2} \leq g(p, 2m)2^{-\epsilon m/2}. \end{aligned}$$

Since this is true for any W we have

$$\begin{aligned} \Pr[A_2] &= E_{W,N} \left[\bigvee_{R \in \mathcal{R}_\epsilon} A_R \right] \\ &\leq E_W \left[E_N \left[\bigvee_{R \in \mathcal{R}_\epsilon} A_R \right] \mid W \right] \end{aligned}$$

$$\begin{aligned}
&\leq E_W \left[E_N \left[\sum_{R \in \mathcal{R}_\epsilon} A_R \right] \mid W \right] \\
&\leq E_W [E_N [g(d, 2m)A_R] \mid W] \\
&\leq g(d, 2m)g(p, 2m)2^{-\epsilon m/2}
\end{aligned}$$

Now using claim 1 we have

$$\Pr[A_1] \leq 2g(d, 2m)g(p, 2m)2^{-\epsilon m/2} \leq \delta$$

giving the desire result. \square