

## Agnostic Learning of General Geometric Patterns and Multi-Instance Learning in $\mathcal{R}^d$

Stephen D. Scott

Dept. of Computer Science  
University of Nebraska  
Lincoln, NE 68588-0115  
sscott@cse.unl.edu

### Abstract

*The concept class of geometric patterns has been heavily studied and has applications in pattern recognition. Previous work on this concept class has been restricted to one or two dimensions or to finite and discretized domains. We present an algorithm to learn a very flexible generalization of previously studied geometric patterns in any constant-dimensional real space, making its potential applicability to pattern matching very high since it can operate on any data representable as a constant-dimensional array of values. To our knowledge, these classes of patterns are more complex than any class of geometric patterns previously studied. We also give variations of our algorithms to learn the union of constant-dimensional geometric objects from multiple-instance examples.*

### 1. Introduction

Consider a robot designed to navigate through a large-scaled environment<sup>1</sup>. Suppose a set of key “landmarks” have already been selected (by another component of the navigation system). It is crucial that the robot be able to recognize whether or not it is in the vicinity of a given landmark from data taken at the robot’s current location. We refer to this problem as the *landmark matching problem*. The landmark matching algorithm should be noise-tolerant.

Much work on designing landmark matching algorithms uses a pattern matching approach to match the visual image (or whatever form of data is available) to the data taken at landmark position  $L$ . The matching algorithm must determine if the robot is near  $L$  (i.e. in a small circle centered around  $L$ ). Because the visual image may change significantly as small movements around  $L$  are made, the pattern matching approach encounters difficulties. Goldberg et

al. [9] and Goldman and Scott [12] proposed using a learning algorithm to construct an accurate hypothesis for performing landmark matching. They obtained their training data by converting the visual data into one-dimensional geometric patterns. Then by applying their algorithm, giving it a set of positive examples (patterns obtained from locations in the vicinity of the landmark) and a set of negative examples, their algorithm constructs a hypothesis to accurately predict if the robot is near the given landmark.

While the basic approaches suggested by Goldman and Scott of using learning versus pattern matching for the landmark matching problem can be applied to a wide range of data, the rest of their work was specific to the data from an imaging system that generates a one-dimensional array of light intensities (called a *signature*) taken at eye-level [16, 18, 26, 29]. The motivation for using one-dimensional data is to reduce the processing time. For some settings, such as an office environment, it seems feasible that the signature taken at eye-level is sufficient. On the other hand, if one wants to design a landmark matching data for a Mars rover, such an approach would not work. Specifically, some experimental work performed by Goldman and Scott [12] revealed that moving from the processed one-dimensional visual image to a one-dimensional pattern lost some key information. So in later work, Goldman et al. [11] defined a class of two-dimensional geometric patterns for which the important features from the visual image are incorporated in the two-dimensional pattern. They then developed an algorithm to learn geometric patterns in any constant dimension  $d$ , allowing for different types and dimensionality of data to be used (e.g. two-dimensional data from an arbitrary pattern recognition problem could be mapped to a three-dimensional pattern). Their algorithm, while online with an absolute mistake bound, has the drawback that it only works in a discretized and bounded space, which can limit its applicability to general pattern recognition problems, especially if no a priori information is available about the range of the data values or the precision required in the

<sup>1</sup> By a large-scaled environment we mean that not all landmarks are visible from all locations in the environment.

data to discriminate between the classes. In contrast, our algorithm works in  $\mathfrak{R}^d$ . The main drawback of our algorithm is that it yields only an *expected* mistake bound (that is polynomial in  $k$  and  $n$ ), whereas in Goldman et al. [11] an absolute bound is given. Of course, in  $\mathfrak{R}^d$ , no absolute mistake bound is possible since an adversary can always find an example to force an on-line algorithm to err.

So that we can relate our paper to previous work we briefly describe the class of one-dimensional geometric patterns. In a one-dimensional geometric pattern, the “target” pattern is a configuration (collection) of up to  $k$  points from  $\mathfrak{R}$ . Each example (instance) is a configuration of up to  $n$  points from  $\mathfrak{R}$ , where it is labeled according to whether or not it visually resembles the target pattern based on the *Hausdorff metric* (for example, see Gruber [14]). Goldberg et al. [9] gave an Occam-based PAC algorithm for learning the class of one-dimensional geometric patterns from the continuous domain, yielding a PAC algorithm. Following that work, Goldman and Scott [12] gave a statistical query algorithm (and hence a noise-tolerant PAC algorithm) for the class of geometric patterns from the real line. Later, Goldman et al. [11] created an algorithm that works in any constant dimension  $d$ , so long as the space is discretized and finite in each dimension. This was further extended by Goldman and Scott [13] to the case of real-valued labels. The work of Goldman et al. has also been evaluated empirically in the context of multiple-instance learning with significant success [28, 30, 31], including an algorithm that scales polynomially in  $d$  at the expense of losing strong learning-theoretic learning bounds [31]. However, all of these algorithms require the input space to be discretized and bounded and require the hypothesis class to be restricted to combinations of axis-parallel boxes (i.e. the Hausdorff metric under the  $L_\infty$  norm only).

One contribution of this paper is an on-line agnostic learning algorithm (that tolerates classification noise) for learning the class of geometric patterns in  $\mathfrak{R}^d$ , which generalizes the classes studied previously. Our algorithm can learn classes that involve very flexible generalizations of the Hausdorff metric, in which the  $L_1$ ,  $L_2$ , or  $L_\infty$  norms may be used to measure distances between points and the distances may be arbitrarily re-scaled in each dimension for each point in the target pattern. In fact, our algorithm can be used to learn classes based on any convex shapes of bounded complexity. To our knowledge, these classes of patterns are more complex than any class of geometric patterns previously studied.

We obtain our algorithm by first sampling the probability distribution  $\mathcal{D}$  over the instance space and then reducing the learning problem to that of learning a disjunction of a set of attributes defined with respect to the sample. (What sets this concept class apart from simpler ones, e.g. unions of boxes, is that our attributes must represent both the geo-

metric objects *and* the complement of their union.) We then apply Winnow [19] to obtain our learning algorithm with a bound on the expected number of prediction mistakes. Using results from *agnostic learning models*, we can bound the expected number of mistakes of our algorithm even if we make no assumptions about the true target concept (the bounds are in terms of the best we can do with the hypothesis class we use). For a variation of our algorithm, another expected mistake bound holds if the target concept shifts (changes) in time but the distribution over the instance space remains fixed.

As with prior work in learning geometric patterns, our work strictly generalizes the multiple-instance learning model (e.g. [8, 22, 4, 24, 25, 1, 27, 33]). In this model, the target concept is a boolean function and each example is a collection of instances and the example (collection) is classified as positive iff at least one of its elements is mapped to positive by the target concept. Long and Tan [22] and Auer et al. [4] have described PAC algorithms for learning a single axis-parallel box from multiple-instance examples where the dimension need not be constant. In their papers, each example is classified as positive if at least one of its points is inside the target box. Our algorithm for learning constant-dimensional patterns can be viewed as learning a union of various geometric objects (including axis-parallel boxes) from a constant-dimensional space where a more complex rule is used for specifying when an example is classified as positive. Specifically, one can define a concept by a set of  $k$  “attraction” points  $C = \{c_1, \dots, c_k\}$  and a set of  $k'$  “repulsion” points  $\bar{C} = \{\bar{c}_1, \dots, \bar{c}_{k'}\}$ . Then the label for a bag  $P = \{p_1, \dots, p_n\}$  is positive if and only if there is a subset of  $r$  points  $C' \subseteq C \cup \bar{C}$  such that each attraction point  $c_i \in C'$  is near some point in  $P$  (where “near” is defined as within a certain distance under some weighted norm) and each repulsion point  $\bar{c}_j \in \bar{C}$  is not near any point in  $P$ . In other words, if one defines a boolean attribute  $a_i$  for each attraction point  $c_i \in C$  that is 1 if there exists a point  $p \in P$  near it and 0 otherwise and another boolean attribute  $\bar{a}_i$  for each repulsion point  $\bar{c}_i \in \bar{C}$  that is 1 if there is no point from  $P$  near it, then  $P$ 's label is an  $r$ -of- $(k + k')$  threshold function over the attributes.

## 2. Agnostic Learning Model and Winnow

We consider the on-line (or mistake-bound) learning model [2, 19] as applied to concept learning (i.e. each example's label is 1 or 0). The learning proceeds in trials, where in trial  $t$  an example  $X_t$  is presented to the learner, and in polynomial time the learner must produce a prediction  $\rho_t$  as to the classification of  $X_t$ . Then the learner receives the desired output  $y_t$  and incurs a loss  $\ell(y_t, \rho_t)$  for some loss function. Since we are studying concept learning, we use the discrete loss function:  $\ell(y_t, \rho_t)$  is 1 if  $y_t \neq \rho_t$ , and 0 otherwise.

The performance of the on-line learner is measured by the total loss over all trials, which in our case is equivalent to the total number of prediction mistakes made. Our on-line learning algorithms are *agnostic* [15, 17] in the sense that they make no assumptions whatsoever about the target concept to be learned. Instead, we compare their performance with the performance of the best hypothesis selected from a comparison or “touchstone” class. For a sequence of trials, the *best hypothesis* from the touchstone class is the one that makes the minimum number of mistakes. We say that an algorithm has polynomial complexity (for either the mistake bound or the time complexity) if it is polynomial in the number of bits required to specify an example and the number of bits needed to encode the best hypothesis.

An important result in this model is Littlestone’s on-line noise-tolerant algorithm Winnow for learning  $K$ -disjunctions of boolean attributes when there is a large number  $N$  of total attributes [19]. Winnow makes predictions based on a linear threshold function  $\sum_{i=1}^N w_i x_i \geq \theta$ , where  $w_i$  is the weight associated with the boolean attribute  $x_i$ . If the prediction is wrong then the weights are updated as follows. On a false negative prediction, for each attribute  $x_i$  that is 1, Winnow *promotes* the weight  $w_i$  by multiplying  $w_i$  by some constant update factor  $\alpha > 1$  (typically  $\alpha = 2$ ). On a false positive prediction, for each literal  $x_i$  that is 1, Winnow *demotes* the weight  $w_i$  by dividing it by  $\alpha$ .

Recently Auer and Warmuth [5], in generalizing the work of Littlestone [21], showed that Winnow makes at most  $O(A + K \log N)$  mistakes on any sequence of trials where the target  $K$ -disjunction makes at most  $A$  attribute errors. The number of attribute errors of a labeled example  $\langle X_t, y_t \rangle$  with respect to the target disjunction is the minimum number of attributes (bits) of  $X_t$  that have to be changed so that the classification of the resulting example by the target is consistent with  $y_t$ . In the agnostic model, whenever the best hypothesis makes a prediction mistake, we only need to change at most  $K$  attributes of the example so that the classification is consistent. Thus we have the following interpretation of the mistake bound in the presence of attribute errors.

**Theorem 1** [5] *Suppose in a sequence of trials for on-line learning an unknown boolean concept defined by  $\leq K$  of  $N$  possible attributes, the best  $K$ -disjunction makes  $M_{opt}$  mistakes (classification errors). Then Winnow, running with  $\alpha = 1.75$ , each initial weight  $= 1/N$ , and  $\theta = (\alpha \ln \alpha)/(\alpha^2 - 1)$ , makes at most the following number of mistakes:*

$$2.75KM_{opt} + 4.92K(\ln N - 1) + 4.92.$$

Auer and Warmuth [5] also offer a version of Winnow that tolerates concept shift (i.e. the target disjunction may change completely in time). When a weight is sufficiently

small, they do not demote it any further. Specifically, no weight is allowed to fall below  $\beta/N$  for some  $\beta \geq 0$ . In this version of Winnow, the mistake bound includes the total number of shifts over all trials.

### 3. The Class of General Geometric Patterns

The instance space  $\mathcal{X}_n$  we will work in consists of all configurations of at most  $n$  points from  $\mathbb{R}^d$  for some constant  $d$ . Before describing the concept class that is the subject of this paper, we first describe a well-studied special case. In this special case, a concept is the set of all configurations from  $\mathcal{X}_n$  within unit distance<sup>2</sup> under the *Hausdorff metric* of some “ideal” configuration of at most  $k$  points. The Hausdorff distance between configurations  $P$  and  $Q$ , denoted  $\text{HD}(P, Q)$ , is

$$\max \left\{ \max_{\mathbf{p} \in P} \left\{ \min_{\mathbf{q} \in Q} \{ \text{dist}(\mathbf{p}, \mathbf{q}) \} \right\}, \max_{\mathbf{q} \in Q} \left\{ \min_{\mathbf{p} \in P} \{ \text{dist}(\mathbf{p}, \mathbf{q}) \} \right\} \right\},$$

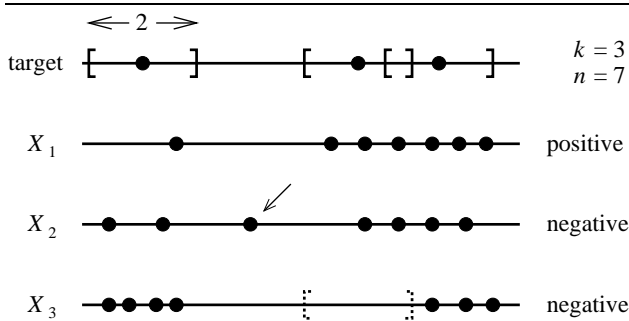
where  $\text{dist}(\mathbf{p}, \mathbf{q})$  is the distance between  $\mathbf{p}$  and  $\mathbf{q}$  under some norm. Thus  $\text{HD}(P, Q) \leq 1$  if every point in  $P$  is within unit distance of some point in  $Q$  and every point in  $Q$  is within unit distance of some point in  $P$ . For  $P \in \mathcal{X}_k$ , we define the concept  $C_P$  that corresponds to  $P$  by  $C_P = \{X \in \mathcal{X}_n : \text{HD}(P, X) \leq 1\}$ . Figure 1 illustrates an example of such a concept for  $d = 1$ . The concept class of  $d$ -dimensional patterns is defined as

$$\mathcal{F}_{n,k} = \{C_P : P \text{ is a config. of } \leq k \text{ pts. from } \mathbb{R}^d\}.$$

We now describe our generalization of the above concept class. First, we allow the Hausdorff metric to use either the  $L_\infty$ ,  $L_1$ , or  $L_2$  norm for its distance function. Further, we allow a target concept to apply its distance function after *re-scaling each dimension independently and separately for each point in the target concept*. To this end, we define a target concept  $f_{n,C_k} : \mathcal{X}_n \rightarrow \{0, 1\}$  with respect to a set of  $k$  components  $C_k = \{c_1, \dots, c_k\}$ . Each component comes from some set  $\mathcal{C}$  of components. Each component  $c_i$  can be represented by a *center of mass*  $\mathbf{y}_i$  (i.e. the target point) and a *scale vector*  $\mathbf{a}_i$ . For example, each component from  $C_k$  might come from one of the following.

1.  $C_{box}$  = the set of axis-parallel boxes in  $\mathbb{R}^d$  (when the  $L_\infty$  norm is used in the Hausdorff metric).  $c_i \in C_{box}$  is represented by the equation  $\max_{1 \leq j \leq d} \left\{ \frac{|p_j - y_{ij}|}{a_{ij}} \right\} \leq 1$ .
2.  $C_{cross-poly}$  = the set of *irregular cross-polytopes* with axis-parallel diagonals (when the  $L_1$  norm is used).  $c_i \in C_{cross-poly}$  is represented by the equation  $\sum_{j=1}^d \frac{|p_j - y_{ij}|}{a_{ij}} \leq 1$ .

<sup>2</sup> Without loss of generality, unit distance can be replaced by any fixed distance since we can just rescale.



**Figure 1.** An example concept from  $\mathcal{F}_{7,3}$  for  $d = 1$ . The top line shows the target pattern with an interval centered at each target point that covers all points within unit distance. Every positive example must have every point within one of the intervals *and* no interval can be empty (e.g.  $X_1$ ). For an example to be negative, there must be a point in it that is not within unit distance of any target point (e.g.  $X_2$ ) and/or there are no points in the example near some target point (e.g.  $X_3$ ).

3.  $\mathcal{C}_{circ}$  = the set of hyperellipsoids with axis-parallel axes (when the  $L_2$  norm is used).  $c_i \in \mathcal{C}_{circ}$  is represented by the equation  $\sum_{j=1}^d \frac{(p_i - y_{ij})^2}{a_{ij}^2} \leq 1$ .

Let  $P$  be a set of  $n$  points from  $\mathbb{R}^d$ . The function  $f_{n, C_k}(P) = 1$  iff the following two criteria are met.

1. Every point  $\mathbf{p} \in P$  lies inside some  $c_j \in C$ .
2. Every component  $c_j \in C$  contains some point  $\mathbf{p} \in P$ .

We define  $\mathcal{F}_{n, k, \ell}$  as the set of all functions  $f_{n, C_k}$  such that the components of  $C_k$  use the  $L_\ell$  norm.

As an example to motivate our definition, we review findings from the experimental work done by Goldman and Scott [12]. A potential problem with their approach was that information was lost when mapping from one-dimensional signatures to one-dimensional patterns. We now briefly describe their mapping. The signatures (which can be viewed as the raw data), consisted of  $s + 1$  distinct light intensity values. (In the data from Pinette [26] that they used,  $s = 359$ .) Each signature was pre-processed by computing its first derivative and then normalizing it by dividing each of the  $s$  derivative values by the difference between the signature's maximum and minimum values. Let  $r$  denote the number of discrete values (precision) for these normalized derivative values. As  $r$  was increased, more information was retained, but the complexity of the learning process (and thus the number of prediction mistakes and the time needed to make a prediction) increased. Next they obtained the training data for the class of one-dimensional patterns by converting the arrays of derivatives into one-dimen-

sional geometric patterns by placing points where there are significant changes. Then they applied their algorithm, giving it positive examples (patterns from locations near the landmark) and negative examples.

Although their experimental results were promising, they discovered that some important information was lost in moving from the signature to the one-dimensional pattern. For example, the points did not reflect the magnitude of light intensity change, or even the direction of change (i.e. was the intensity increasing or decreasing). Thus Goldman et al. [11] developed a way to map the signatures to two-dimensional patterns in such a way that (1) all important information from the signatures is maintained and (2) they can successfully learn the resulting class of two-dimensional patterns. Since our algorithm generalizes theirs, our algorithm can also make use of their mapping, as well as other mappings from  $d$ -dimensional data to  $(d + 1)$ - (or higher-) dimensional patterns in real space. While the above mapping to a discrete and finite space seems acceptable for the landmark matching problem, in other pattern recognition problems, bounds on the feature values and the required precision for successful classification may not be known a priori and might not be well estimated from the training data. Thus mapping data to  $\mathbb{R}^{d+1}$  rather than to a discrete and bounded space helps alleviate problems of re-scaling data and insufficient precision.

## 4. Our Algorithm for Learning General Geometric Patterns

We now present a framework for developing our algorithm. We first give a new interpretation of our concept class, expressing each concept as a disjunction of attributes. We then describe how to generate appropriate attributes to approximate the target concept.

### 4.1. Another Interpretation of $\mathcal{F}_{n, k, \ell}$

Recall that  $\mathcal{F}_{n, k, \ell}$  is the set of all functions  $f_{n, C_k}$  such that the components of  $C_k$  are defined using the  $L_\ell$  norm. Note that we can represent any function from  $\mathcal{F}_{n, k, \ell}$  as  $1 - f'_{n, C_k, \bar{C}_{k_{comp}}}$ , where

$$f'_{n, C_k, \bar{C}_{k_{comp}}}(P) = \left( \bigvee_{c_i \in C_k} g_i(P) \right) \vee \left( \bigvee_{\bar{c}_j \in \bar{C}_{k_{comp}}} \bar{g}_j(P) \right),$$

$$g_i(P) = \begin{cases} 1 & \text{if } \nexists \mathbf{p} \in P \text{ s.t. } \mathbf{p} \in c_i, \\ 0 & \text{otherwise} \end{cases},$$

$$\bar{g}_j(P) = \begin{cases} 1 & \text{if } \exists \mathbf{p} \in P \text{ s.t. } \mathbf{p} \in \bar{c}_j, \\ 0 & \text{otherwise} \end{cases},$$

and  $\bar{C}_{k_{comp}} = \{\bar{c}_1, \dots, \bar{c}_{k_{comp}}\}$  is a set of components (not necessarily of the same type as those of  $C_k$ ) such that  $c_i \cap$

$\bar{c}_j = \emptyset \forall i, j$  and

$$\left( \bigcup_{c_i \in C_k} c_i \right) \cup \left( \bigcup_{\bar{c}_i \in \bar{C}_{k_{comp}}} \bar{c}_i \right) = \mathfrak{R}^d.$$

In other words,  $\bar{C}_{k_{comp}}$  is a set of  $k_{comp}$  components whose union is exactly the complement of the union of the components of  $C_k$ . By defining appropriate attributes for these components (corresponding to  $g_i$  and  $\bar{g}_j$ ), we can re-define any target function as 1 – the disjunction of those attributes and then apply Winnow to learn the appropriate disjunction. Of course, since we are working in  $\mathfrak{R}^d$ , we cannot necessarily find a perfect set of attributes, but by applying VC-dimension theory, we can find a set of attributes that (with high probability) performs arbitrarily well. We will then apply the agnostic results of Theorem 1 to achieve an expected mistake bound for our learning algorithm.

To find a good set of attributes for Winnow, we will draw a sufficiently large unlabeled sample  $S$  according to an arbitrary but fixed probability distribution  $\mathcal{D}$ .  $S$  will be such that any set of  $k$  components consistent with it (if we knew the labels) would have error at most  $\epsilon$  with probability at least  $1 - \delta$ . Let  $S$  be the points of all examples from  $S$ . We will then generate a set of components  $C_S$  that contains every possible subset of the points in  $S$ . We will then partition the space into regions where each region consists of the intersection of a distinct subset of the components of  $C_S$  whenever this intersection is nonempty. For  $d$  dimensions, there are at most  $O(|C_S|^d)$  such regions. We then assign two attributes  $A_r$  and  $A'_r$  per region  $r$  that correspond to the functions  $g_i(\cdot)$  and  $\bar{g}_j(\cdot)$ . This set of attributes will include the ones relevant to representing the target concept. Since the “best” subset of attributes have error at most  $\epsilon$  (with high probability), we can use this in the agnostic portion of Theorem 1 to get a bound on the expected number of on-line prediction mistakes, assuming all the examples are drawn according to  $\mathcal{D}$ .

## 4.2. Creating the Attributes

The set of attributes we create for Winnow is based on a random sample drawn according to  $\mathcal{D}$ . We use this sample to partition  $\mathfrak{R}^d$  into regions and assign two attributes per region. The size of the sample is derived from a result of Blumer et al. [6]. To apply their results, we first bound the VC-dimension of our concept class, which is stated in the following theorem (proof in the appendix).

### Theorem 2

$$\text{VCD}(\mathcal{F}_{n,k,\ell}) \leq \begin{cases} 2kd \ln(8ekn) & \text{for } \ell = 1 \\ 2kd \ln(16ekn) & \text{for } \ell = 2 \\ 2kd \ln(8eknd) & \text{for } \ell = \infty \end{cases}.$$

Combining Theorem 2 with a result of Blumer et al. [6] allows us to determine the size of a sample  $S$  such that any concept  $f_{n,C_k} \in \mathcal{F}_{n,k,\ell}$  consistent with it will have error at most  $\epsilon$  with probability at least  $1 - \delta$ . Note that if the true target function cannot be perfectly represented by a function from  $\mathcal{F}_{n,k,\ell}$ , we can still prove error bounds. The results of Kearns et al. [17] say that any function from  $\mathcal{F}_{n,k,\ell}$  that *minimizes disagreements* with a sufficiently large sample  $S'$  will, with probability at least  $1 - \delta$ , have error at most  $(\inf_{f \in \mathcal{F}_{n,k,\ell}} \{E[L_f]\} + \epsilon)$ , where  $E[L_f]$  is the expected error of function  $f$  on examples drawn according to  $\mathcal{D}$ . The size of  $S'$  (which depends on  $\epsilon$ ,  $\delta$ , and the VC-dimension of  $\mathcal{F}_{n,k,\ell}$ ) can be determined via uniform convergence results [32]. Throughout this paper we will assume that the sample is labeled according to a function from  $\mathcal{F}_{n,k,\ell}$ , noting that we can substitute  $(\inf_{f \in \mathcal{F}_{n,k,\ell}} \{E[L_f]\} + \epsilon)$  for  $\epsilon$  in our discussion.

Let the size of our sample be  $m$ . By ignoring the labels and ignoring which points come from which examples, we get a set of points  $S$  of size  $mn$ . From this we build  $C_S$ , which is a set of components that contains all possible subsets of the points in  $S$ .<sup>3</sup> By our earlier arguments, with high probability there is some set of  $k$  components from  $C_S$  that has error at most  $\epsilon$  or is within  $\epsilon$  of the best possible given our hypothesis class.

We build  $C_S$  given  $S$  by applying the results of Blumer et al. [6]. Define the concept classes of single elements from  $C_{box}$ ,  $C_{cross-poly}$ , and  $C_{circ}$  as  $\mathcal{F}_{1,1,\infty}$ ,  $\mathcal{F}_{1,1,1}$ , and  $\mathcal{F}_{1,1,2}$ , respectively. We first note that  $\text{VCD}(\mathcal{F}_{1,1,\ell}) \leq 2d$  for  $\ell \in \{1, 2, \infty\}$ . Also note that the *consistent hypothesis problem* for each of these classes can be solved in polynomial time. (For example, given any set of points labeled by a single axis-parallel box, we can efficiently find an axis-parallel box consistent with it.) Under these two conditions, we can apply a result of Blumer et al. that says we can enumerate all possible behaviors of a finite set  $S$  of points with respect to  $\mathcal{F}_{1,1,\ell}$  in time polynomial in  $|S| = mn$ . Since these behaviors correspond to components in  $C_S$ , we can easily create  $C_S$  in polynomial time. (Note that for special cases such as  $C_{box}$ , faster algorithms might exist.) Finally, because the VC-dimension of the single-component classes are all finite, we can bound the size of  $C_S$  as

$$|C_S| \leq (mn)^{\text{VCD}(\mathcal{F}_{1,1,\ell})} \leq (mn)^{2d}.$$

Once  $C_S$  has been created, we proceed to the next step, which is to generate the regions to which we will attach the attributes for Winnow. The set of regions is

$$R = C_S \cup I \cup \{r_{comp}\},$$

<sup>3</sup> We may restrict the components placed in  $C_S$  so as to avoid overfitting. We may do this by restricting the sizes and/or aspect ratios of the components used.

where

$$I = \bigcup_{C \in 2^{C_S}} \left\{ \bigcap_{c \in C} c : \text{the intersection is non-empty} \right\},$$

$$r_{comp} = \mathbb{R}^d \setminus \left( \bigcup_{c \in C_S} c \right),$$

and  $2^{C_S}$  is the power set of  $C_S$ . So the set of regions is the set  $C_S$  of components plus each intersection of a distinct subset of the components of  $C_S$  whenever this intersection is nonempty, plus the space not covered by any of the components. For  $d$  dimensions, it can be shown that  $|I| = O(|C_S|^d) = O((mn)^{2d^2})$ . Thus there are  $O((mn)^{2d} + (mn)^{2d^2})$  total regions. We then assign two attributes  $A_r$  and  $A'_r$  per region  $r \in R$  that are assigned as follows when given an example:  $A_r$  is 1 iff  $r$  contains a point, and  $A'_r$  is 1 iff  $r$  is empty (region  $r_{comp}$  only receives one attribute  $A_{r_{comp}}$ , which is 1 when it contains a point). We call the entire set of attributes  $\mathcal{A}$  and note that the total number of attributes  $N = |\mathcal{A}| = O((mn)^{2d^2})$ . Finally, it is worth mentioning that for  $\mathcal{C}_{box}$ ,  $I \subseteq C_S$ , so  $|R|$  and  $N = |\mathcal{A}|$  are both  $O((mn)^{2d})$ .

A consequence of the VC-dimension results is that there exists a size- $k$  subset  $C^* \subseteq C_S$  that (with high probability) has error at most  $\epsilon$  on examples drawn according to  $\mathcal{D}$ . Let  $\mathcal{A}'^*$  be the set of attributes  $A'_r$  associated with the regions of  $C^*$  and let  $\mathcal{A}^*$  be the set of attributes  $A_r$  associated with  $r_{comp}$  and the regions of  $I$  that do not intersect any region of  $C^*$ . By definition, any region from  $I$  intersecting a region from  $C^*$  lies entirely inside  $C^*$ , so the regions associated with the attributes of  $\mathcal{A}^*$  form the complement of the regions of  $C^*$ . Then based on the discussion of Section 4.1, our target disjunction is

$$\left( \bigvee_{A'_r \in \mathcal{A}'^*} A'_r \right) \vee \left( \bigvee_{A_r \in \mathcal{A}^*} A_r \right).$$

The number of relevant attributes is, from Section 4.1,  $K = k + k_{comp}$ , where  $k = |C^*|$  is the number of attributes in the first term, and the number of attributes in the second term ( $k_{comp}$ ) is at most the number of regions of  $R$  that do not intersect any region of  $C^*$ . So for example [11], for  $\mathcal{C}_{box}$  we have  $k_{comp} \leq (2k + 1)^d$  and  $K \leq k + (2k + 1)^d$ . In general,  $k_{comp} \leq (|C_S| - k) + 1 + (|I| - |I^*|)$ , where  $I^* \subseteq I$  is the set of regions from  $I$  that intersect some region from  $C^*$  and the +1 accounts for  $A_{r_{comp}}$ . Let  $S_{c_i}$  be the set of points from  $S$  that lie in component  $c_i \in C_S^*$ . Let  $\Pi_{\mathcal{F}_{1,1,\ell}}(S_{c_i})$  be the number of behaviors (dichotomies) on  $S_{c_i}$  that are realized by functions from  $\mathcal{F}_{1,1,\ell}$ . This is at most twice the number of regions from  $I$  that lie in  $c_i$ . Thus  $k_{comp} \leq (|C_S| - k) + 1 + |I| - \frac{1}{2} \left( \sum_{c_i \in C^*} \Pi_{\mathcal{F}_{1,1,\ell}}(S_{c_i}) - 1 \right)$ , yield-

ing

$$K \leq |C_S| + 1 + |I| - \frac{1}{2} \left( \sum_{c_i \in C^*} \Pi_{\mathcal{F}_{1,1,\ell}}(S_{c_i}) - 1 \right). \quad (1)$$

(The extra  $-1$  in the final term accounts for the component  $c_i$  itself.)

Of course, the actual value of the final term of Equation 1 depends on the original sample, especially the number of positive examples<sup>4</sup> and the arrangement of the points. Recall that the sample also influences  $|C_S|$  and  $|I|$ , so when finding an upper bound on  $K$ , we want to consider  $|C_S|$ ,  $|I|$  and  $|I^*|$  together.

From Section 4.1, we know that there is a set of attributes from  $\mathcal{A}$  whose disjunction has prediction error at most  $(\inf_{f \in \mathcal{F}_{n,k,\ell}} \{E[L_f]\} + \epsilon)$  (w.h.p.) on examples drawn according to  $\mathcal{D}$ . Thus Theorem 1 can be applied to yield the following expected mistake bound.

**Theorem 3** *If all examples are drawn independently from distribution  $\mathcal{D}$ , then the expected number of prediction mistakes for our algorithm on  $t$  trials is*

$$2.75Kt \left( \inf_{f \in \mathcal{F}_{n,k,\ell}} \{E[L_f]\} + \epsilon \right) + 4.92K(\ln N - 1) + 4.92.$$

Note that we may also apply Winnow's shift-tolerant bounds [5] to achieve an expected mistake bound in the presence of a shifting concept (i.e. if the target points change) so long as the distribution  $\mathcal{D}$  remains fixed. This is because we generated our attributes for Winnow based on an *unlabeled* sample, so the only information we got from  $S$  is the nature of  $\mathcal{D}$ . Note that this also implicitly gives us tolerance of classification noise since our set of attributes is independent of the labels. Naturally, one would like to select  $\epsilon$  to minimize the bound of Theorem 3, which is challenging given that decreasing  $\epsilon$  generally increases  $N$  and  $K$ , and  $N$  and  $K$  also depend on the specific unlabeled sample drawn.

## 5. Concluding Remarks

Presented was an algorithm to agnostically learn the concept classes of very general geometric patterns in  $\mathbb{R}^d$ , where  $d$  is a constant, which is an improvement over previous algorithms that either required a discretized and bounded space or were limited to  $d = 1$  or 2. Our algorithm combines VC-dimension theory with Winnow to obtain an on-line learning algorithm with an expected mistake bound. The mistake bounds can be improved if we were to tune the parameters, including Winnow's parameters and  $\epsilon$ .

<sup>4</sup> If the components of  $C^*$  are consistent with the original sample, then we have for all  $c_i \in C_S^*$ ,  $|S_{c_i}| \geq$  the number of positive examples in the original sample.

We note that the technique described in this paper works for any other convex components of bounded complexity. Naturally, increased complexity of the components increases the total number of attributes and the expected mistake bound.

Removing the exponential dependence on  $d$  from our time bounds would be very difficult. It is well known (e.g. Maass and Warmuth [23], Bshouty et al. [7]) that learning unions of at most  $k$   $d$ -dimensional boxes (with single-instance examples) in time polynomial in  $d$  with an instance space of  $\{0, 1\}^d$  yields an algorithm for learning  $k$ -term DNF formulas over  $d$  variables in time polynomial in  $k$  and  $d$ , which is a major open problem in learning theory. Because our algorithm generalizes learning unions of axis-parallel boxes, removing the exponential dependence on  $d$  would solve the  $k$ -term DNF problem.

It is well-known that learning algorithms in the mistake-bound model can be mapped to PAC algorithms [2, 20]. We can convert our mistake-bound algorithm to a PAC algorithm with expected sample complexity as follows [2, 19]. For each trial, we draw  $q_i = \lceil (1/\epsilon)(\ln(1/\delta) + i \ln 2) \rceil$  examples randomly according to  $\mathcal{D}$  and check if our current hypothesis (the setting of the weights in Winnow) is consistent with the sample. If it is, then we halt. Otherwise we take one of the examples our hypothesis misclassifies and use it to update the weights. We then move on to the next trial. For a mistake bound of  $M$ , the total sample complexity is  $\sum_{i=1}^M q_i = O\left(\frac{1}{\epsilon} \left(M \ln \frac{1}{\delta} + M^2\right)\right)$ . Note that the mistake bound  $M$  need not be known in this mapping. Thus we can substitute our expected mistake bound from Theorem 3 into the above equation and get an expected bound on the sample complexity for a PAC algorithm for general geometric patterns in constant-dimensional space. (Note that the  $\epsilon$  and  $\delta$  of the above equation are different from those used to draw our original sample  $S$ .) A better bound on sample complexity can be attained if an upper bound  $M$  is known in advance by applying the results of Littlestone [20]. Future work is to adapt his results to prove an absolute sample complexity for a PAC version of our algorithm.

Another interesting direction is to determine if we can prune some regions from  $R$  before mapping to attributes. One possible way of doing this is through the use of *statistical queries* [3], where we would test each region  $r \in R$  to determine the probability that it contains a point from a positive (or negative) example (other statistical tests are also possible). Perhaps such information can provably reduce the size of  $\mathcal{A}$  without significantly increasing the error bound. A benefit of this is that we still maintain tolerance of classification noise by using the SQ model.

## Appendix

To prove Theorem 2, we make use of a result of Goldberg and Jerrum [10].

**Theorem 4** [10] *Let  $\{\mathcal{F}_{k',n'} : k', n' \in \mathbb{N}\}$  be a family of concept classes where concepts in  $\mathcal{F}_{k',n'}$  and instances are represented by  $k'$  and  $n'$  real values, respectively. Suppose that the membership test for any instance and any concept  $f$  of  $\mathcal{F}_{k',n'}$  can be expressed as a boolean formula  $\Phi_{k',n'}$  containing  $\sigma = \sigma(k', n')$  distinct atomic predicates, each predicate being a polynomial inequality over  $k' + n'$  variables (representing  $f$  and  $x$ ) of degree at most  $q = q(k', n')$ . Then  $\text{VCD}(\mathcal{F}_{k',n'}) \leq 2k' \ln(8eq\sigma)$ .*

**Proof of Theorem 2:** Each component  $c_i \in C_k$  can be represented by its scaled distance (under the appropriate norm) to  $c_i$ 's center of mass, which we denote  $\mathbf{y}_i$ . In other words, if a point  $\mathbf{p} = (p_1, \dots, p_d)$  satisfies

$$\sum_{j=1}^d \frac{|p_j - y_{ij}|}{a_{ij}} \leq 1,$$

$$\sum_{j=1}^d \frac{(p_j - y_{ij})^2}{a_{ij}^2} \leq 1,$$

or

$$\max_{1 \leq j \leq d} \left\{ \frac{|p_j - y_{ij}|}{a_{ij}} \right\} \leq 1,$$

for the  $L_1$  norm, the  $L_2$  norm, and the  $L_\infty$  norm, respectively, then we know that point  $\mathbf{p}$  lies in component  $c_i$ . The positive scalars  $a_{ij}$  represent the amount that the norm is scaled in dimension  $j$  for component  $c_i$ , and since these values are positive, we may include them in the absolute values of the above expression. Thus we can represent the target concept by the values  $y_{ij}/a_{ij}$ , and there are  $k' = kd$  such values. Similarly, each point  $\mathbf{p}_\ell$  from an example can be represented by the values  $p_{\ell j}/a_{ij}$ , and there are  $kd$  such values. Multiplying by the  $n$  points per example yields  $n' = nkd$ . For the  $L_1$  norm, at most  $kn$  distinct degree-1 inequalities are required. For the  $L_2$  norm, at most  $kn$  distinct degree-2 inequalities are required. For the  $L_\infty$  norm, at most  $dkn$  distinct degree-1 inequalities are required (the extra factor of  $d$  is due to the fact that in max, we must check each dimension individually).  $\square$

## Acknowledgments

The author thanks Subhash Suri and Sally Goldman for their helpful discussions. This research was funded in part by NSF grant CCR-0092761. It was also supported in part by NIH Grant Number RR-P20 RR17675 from the IDeA program of the National Center for Research Resources.

## References

- [1] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. *Advances in Neural Information Processing Systems 15*, 2002.
- [2] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, April 1988.

- [3] J. A. Aslam and S. E. Decatur. Specification and simulation of statistical query algorithms for efficiency and noise tolerance. *J. of Comp. and Sys. Sciences*, 56(2):191–208, 1998.
- [4] P. Auer, P. M. Long, and A. Srinivasan. Approximating hyper-rectangles: Learning and pseudo-random sets. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 314–323. ACM, 1997.
- [5] P. Auer and M. Warmuth. Tracking the best disjunction. *Machine Learning*, 32(2):127–150, 1998.
- [6] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *JACM*, 36(4):929–965, 1989.
- [7] N. Bshouty, P. W. Goldberg, S. A. Goldman, and H. D. Mathias. Exact learning of discretized geometric concepts. *SIAM Journal of Computing*, 28(2):675–700, 1999.
- [8] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Perez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1–2):31–71, 1997.
- [9] P. Goldberg, S. A. Goldman, and S. D. Scott. PAC learning of one-dimensional patterns. *Machine Learning*, 25(1):51–70, October 1996.
- [10] P. W. Goldberg and M. R. Jerrum. Bounding the Vapnik-Chervonenkis dimension of concept classes parameterized by real numbers. *Machine Learning*, 18(2/3):131–148, 1995.
- [11] S. Goldman, S. Kwek, and S. Scott. Agnostic learning of geometric patterns. *J. of Computer and System Sciences*, 62(1):123–51, 1998.
- [12] S. A. Goldman and S. D. Scott. A theoretical and empirical study of a noise-tolerant algorithm to learn geometric patterns. *Machine Learning*, 37(1):5–49, 1999.
- [13] S. A. Goldman and S. D. Scott. Multiple-instance learning of real-valued geometric patterns. *Annals of Mathematics and Artificial Intelligence*, 39(3):259–290, 2003.
- [14] P. M. Gruber. Approximation of convex bodies. In P. M. Gruber and J. M. Willis, editors, *Convexity and its Applications*. Birkhäuser Verlag, 1983.
- [15] D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inform. Comput.*, 100(1):78–150, September 1992.
- [16] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. Riseman. Image-based homing. *IEEE Control Systems Magazine*, 12(1):38–45, 1992.
- [17] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2/3):115–142, 1994.
- [18] T. Levitt and D. Lawton. Qualitative navigation for mobile robots. *Artificial Intelligence*, 44(3):305–360, 1990.
- [19] N. Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1988.
- [20] N. Littlestone. From on-line to batch learning. In *Proc. 2nd Ann. Workshop on Comput. Learning Theory*, pages 269–284, San Mateo, CA, 1989. Morgan Kaufmann.
- [21] N. Littlestone. Redundant noisy attributes, attribute errors, and linear threshold learning using Winnow. In *Proc. 4th Annu. Workshop on Comput. Learning Theory*, pages 147–156, San Mateo, CA, 1991. Morgan Kaufmann.
- [22] P. M. Long and L. Tan. Pac learning axis-aligned rectangles with respect to product distributions from multiple-instance examples. In *Proceedings of the Ninth Annual ACM Conference on Computational Learning Theory*, pages 228–234. ACM Press, New York, NY, 1996.
- [23] W. Maass and M. K. Warmuth. Efficient learning with virtual threshold gates. *Inf. and Comp.*, 141(1):66–83, 1998.
- [24] O. Maron and T. Lozano-Pérez. A framework for multiple-instance learning. In *Advances in Neural Information Processing Systems 10*, 1998.
- [25] O. Maron and A. L. Ratan. Multiple-instance learning for natural scene classification. In *Proc. 15th International Conf. on Machine Learning*, pages 341–349. Morgan Kaufmann, San Francisco, CA, 1998.
- [26] B. Pinette. *Image-Based Navigation Through Large-Scaled Environments*. PhD thesis, University of Massachusetts, Amherst, 1993.
- [27] S. Ray, and D. Page. Multiple-instance regression. In *Proceedings of the Eighteenth International Conference on Machine Learning* (pp. 425–432), 2001.
- [28] S. D. Scott, J. Zhang, and J. Brown. On generalized multiple-instance learning. Technical report UNL-CSE-2003-5, University of Nebraska, 2003.
- [29] H. Suzuki and S. Arimoto. Visual control of autonomous mobile robot based on self-organizing model for pattern learning. *Journal of Robotic Systems*, 5(5):453–470, 1988.
- [30] Q. Tao and S. Scott. A faster algorithm for generalized multiple-instance learning. In *Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 550–555, May 2004.
- [31] Q. Tao, S. Scott, N. V. V., and T. Osugi. SVM-based generalized multiple-instance learning via approximate box counting. In *Proceedings of the Twenty-First International Conference on Machine Learning*, pages 799–806, 2004.
- [32] V. Vapnik. *Statistical Learning Theory*. Adaptive and Learning Systems for Signal Processing, Communications, and Control. John Wiley and Sons, New York, 1998.
- [33] Q. Zhang and S. Goldman. EM-DD: An improved multiple-instance learning technique. *Neural Information Processing Systems 14* (pp. 1073–1080), 2001.