Multiple-Instance Learning of Real-Valued Geometric Patterns

Sally A. Goldman (`sg@cs.wustl.edu`)
Department of Computer Science, Washington University, St. Louis, MO 63130-4899

Stephen D. Scott (`sscott@cse.unl.edu`)
Department of Computer Science and Engineering, University of Nebraska, Lincoln, NE 68588-0115

November 3, 2000
Technical Report UNL-CSE-99-006

Abstract

Recently, there has been a significant amount of research studying the multiple-instance learning model, yet all of this work has only considered this model when there are boolean labels. However, in many of the application areas for which the multiple-instance model fits, real-valued labels are more appropriate than boolean labels. In this paper we define and study a real-valued multiple-instance model in which each multiple-instance example is given a real-valued classification in $[0, 1]$. The real-valued classification indicates the degree to which the example satisfies the target concept. To provide additional structure to the resulting learning problem, we associate a real-valued label with each point in the multiple-instance example. These values are then combined using a real-valued aggregation operator to obtain the classification for the example. Motivated by the possible application of learning geometric patterns to problems in pattern recognition and scene classification (with applications to content-based image retrieval), we provide on-line agnostic algorithms for learning real-valued multiple-instance geometric concepts defined by axis-aligned boxes in constant dimensional space. We obtain our learning algorithm by reducing the problem to one in which the exponentiated gradient (or gradient descent) algorithm can be used.

We also give a novel application of the virtual weights technique. In typical applications of the virtual weights technique, all of the concepts in a group have the same weight and prediction which allows a single "representative" concept from each group to be tracked. However, in our application there are an exponential number of different weights (and possible predictions). Hence, boxes in each group have different weights and predictions making the computation of the contribution of a group significantly more involved. However, we are able to both keep the number of groups polynomial in the number of trials and efficiently compute the overall prediction.

Keywords: Exponentiated Gradient algorithm, multiplicative weight updates, virtual weights, geometric patterns, multiple-instance learning, real-valued labels, scene classification, content-based image retrieval, landmark matching

1. Introduction

Recently, Dieterich et al. (1997) introduced the notion of learning from multiple-instance examples where the target concept is a boolean function, each example is a collection (or *bag*) of instances, and the bag is classified as positive if and only if at least one of its elements is classified as positive by the target concept. To help motivate our work, we briefly review the work of Maron and Ratan (1998) on applying the multiple-instance learning model to the domain of scene classification, which can be used in a query by example approach to content-based image retrieval. Suppose you are given a set of images labeled as to whether or not they contain some particular feature, for example, a waterfall. The goal here is to create a rule that can be used to describe which images contain a waterfall. In their work, Maron and Ratan create a set of subimages, which are 2×2 sets of pixels (which they call *blobs*) along with the four neighboring blobs. They represent each of the five blobs in the subimage by a triple with the average red, green and blue intensities. Hence they obtain a 15-dimensional data point for each subimage. Note that if, instead of color images, black-and-white images were being used then each subimage would map into a 5-dimensional data point. A bag for a given image will contain all of the subimages contained within the image. A bag is a positive example of a waterfall (or whatever concept is being learned) exactly when at least one of the subimages corresponds to a waterfall. Suppose instead of classifying an image as to whether or not it contained a waterfall, you want to classify an image as to whether or not it represents a "dangerous" scene. Here, instead of giving a boolean label to each image, it may be much more appropriate to associate a real-valued label indicating the degree of danger indicated by the image. In fact, in the drug discovery application for which the multiple-instance model was first introduced, Dieterich et al. (1997) used a boolean classification as to whether or not a molecule is a musk molecule, yet as they indicated themselves, this setting is atypical in that for most drug discovery applications a real-valued affinity value would be used as the label. Recently, there has a been a significant amount of work in studying the multiple-instance model (Wang and Zucker, 2000; Long and Tan, 1998; Auer et al., 1998; Auer, 1997; Goldman et al., 2000a; Maron and Lozano-Pérez, 1998; Maron, 1998; Maron and Ratan, 1998; Blum and Kalai, 1998), yet *all* of this work has only considered boolean classification. One of the contributions of our work is initiating the study of a real-valued multiple-instance model.

In the standard multiple-instance model, an example contains m points from some domain $\mathcal{X}$. Most typically, $\mathcal{X} = \Re^d$. In both the drug

discovery and natural scene classification examples described above, one can think of there being an ideal point $\mathbf{p}$ in $\Re^d$ for which binding will occur (in the drug discovery application) or which captures the essence of a waterfall (in the scene classification example). In the boolean multiple-instance model the target concept can be defined as a d-dimensional axis-aligned box, and a bag $\{\mathbf{x}_1, \ldots, \mathbf{x}_m\}$ is classified as positive if and only if at least one of the $\mathbf{x}_i$ in the bag is within the target box. In our work, we want to associate a real-valued label with each bag. To provide additional structure to our model, we associate a real-valued label with each point in the bag and then these labels are combined using an *aggregation operator* to obtain the real-valued label for the bag. See Lin and Lee (1996) or Klir and Yuan (1995) for a discussion on fuzzy aggregation operators for the case in which fuzzy labels are used. However, one can easily define aggregation operators that are appropriate to the particular application area when probabilistic or other real-valued labels are used.

As we describe more in the next section, the particular problem motivating our study yields a more complex multiple-instance concept class than the scene classification setting discussed above. Also, instead of a batch learning model as described above in which the learner is provided with a set of labeled training data from which the hypothesis must be built, here we are interesting in applying a on-line model in which the learner must make predictions as it is refining its hypothesis.

In this paper we define and study an on-line, agnostic (defined in the next section), real-valued multiple-instance model in which each multiple-instance example is given a real-valued classification in $[0, 1]$. The real-valued classification indicates the degree to which the example satisfies the target concept. Motivated by the possible application of learning geometric patterns to problems in pattern recognition (Goldman and Scott, 1999; Goldman et al., 2000a) and of scene classification/content-based image retrieval, we provide algorithms for learning real-valued multiple-instance geometric classes defined by a set of axis-aligned boxes. In our work, we let $\mathcal{X} = S^d$, for $S = \{1, 2, \ldots, s\}$ and d a constant. For data with bounded values, (e.g. visual images) using a discretized d-dimensional space is not a restriction since there is generally a fixed degree of precision available. The parameter s corresponds to the number of different discrete values that are possible. Our loss bound (and time complexity when using the virtual weights technique) depend logarithmically on s. Our current algorithm is only feasible for small dimensional spaces since our algorithm has exponential dependence on d. For the drug discovery applications d is typically in the hundreds and hence our results will not directly by applicable to that application area. However, for the scene classification problem,

if black and white images are used, then $d = 5$ and hence our results could be applied to that setting[1]. In Section 3 we give another possible application area for our results which is the problem which provided the motivation for much of this work.

We obtain our loss bounds by reducing our geometric learning problem to one where we can apply either the exponentiated gradient (EGU) algorithm or gradient descent (GD) algorithm. Both of these algorithms are discussed in Section 5. In general, our reduction involves enumerating all boxes in S^d and associating attributes with them. Then we can apply either EGU or GD to learn the best linear combination of the attributes. The technique we use to obtain our results is quite general and can be applied for any concept class and method for obtaining the label for the multiple-instance examples as long as any multiple-instance example can be evaluated in polynomial time and the aggregation function can be approximated using a linear combination.

One drawback of our application of EGU (or GD) is that the reductions we use create an exponential number of attributes (even for d a constant) and thus the predictions cannot be made in polynomial time. Another key contribution of our work is our development of a novel application of the virtual weights technique of Maass and Warmuth (1998) that enables our predictions to be made in polynomial time. In typical applications of the virtual weights technique, all of the concepts in a group have the same weight and prediction, which allows a single "representative" concept from each group to be tracked. However, in our application there are an exponential number of different weights (and possible predictions). Hence, boxes in each group have different weights and predictions which makes the computation of the contribution of a group is significantly more involved. As described in Section 6.2, we are able to both keep the number of groups polynomial in the number of trials and efficiently compute the overall prediction.

This paper is organized as follows. In the next section, we formalize our real-valued multiple-instance learning model. In Section 3, we describe a pattern recognition application to motivate our study of geometric concepts in the real-valued multiple-instance learning model. We contrast our work with related work in Section 4. Then in Section 5 we review some results on the exponentiated gradient and gradient descent algorithms, which we use in our algorithms. Our algorithms are presented in Section 6. We first present our reduction to EGU and GD in Section 6.1 and then the virtual weights variations are described in

---

[1] In addition, if other features (e.g. elongation, region aspect ratio) are used in lieu of blobs, then d is the number of features extracted per region, and may be less than 5.

Section 6.2. Section 7 looks at alternate aggregation functions. Finally, Section 8 summarizes other results and future directions of this work.

2. A Real-Valued Multiple-Instance On-line Model

We apply the on-line agnostic learning model (Haussler, 1992; Kearns et al., 1994) to the multiple-instance setting. Let $\mathcal{S} = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t) \rangle$ be the sequence of trials. Each multiple-instance example $\mathbf{x}_i$ is a set of m_i elements from domain $\mathcal{X}$. While our model is well defined for any domain $\mathcal{X}$, throughout the rest of this paper we will assume that $\mathcal{X} = S^d$ for $S = \{1, 2, \ldots, s\}$. Hence each bag consists of a set of points from a discretized d-dimensional space. As in the standard on-line learning model, during trial t, multiple-instance example $\mathbf{x}_t$ is presented to the learner. In polynomial time the learner must produce a prediction $\hat{y}_t \in [0, 1]$ as to the classification of $\mathbf{x}_t$. Then the learner receives the desired output $y_t \in [0, 1]$ and incurs a *loss* $L(y_t, \hat{y}_t)$ for some loss function L. In this paper we use the square loss function (i.e. $L(\hat{y}_t, y_t) = (\hat{y}_t - y_t)^2$). However, we could instead use other loss functions. The total loss of the learner is given by $\sum_{t=1}^{T} L(\hat{y}_t, y_t)$ where T is the total number of trials. We consider the *agnostic learning model* in which the performance of the learning algorithm is compared 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 has the minimum total loss.

In our model, along with having each bag b defined by a set of points from $\mathcal{X}$, the target concept C will be defined by a set of concepts from some *base class* $\mathcal{C}$. That is, $C = \{c_1, \ldots, c_k\} \subseteq \mathcal{C}$. We must define a set of functions to determine how the final label in $[0, 1]$ is given to the bag b according to the target concept C. Our goal is that the this label will measure the degree to which the bag is positive with respect to the target concept.

For a concept $c \in \mathcal{C}$ and point $\mathbf{p} \in \mathcal{X}$, we define a *membership function*, $\mu_c : \mathcal{X} \to [0, 1]$. That is, much like in the fuzzy sets literature, $\mu_c(\mathbf{p})$ returns an appropriate label in $[0, 1]$ for the point $\mathbf{p}$ that indicates the amount of *membership* a point $\mathbf{p}$ has in c. Throughout this paper, $\mathcal{C}$ will be the class of axis-aligned boxes in S^d. We made this choice since in the applications we are considering, each point $\mathbf{p}$ is d-dimensional point and the label should be related to how close $\mathbf{p}$ is to the center $\mathbf{p_c}$ of the box c. Hence, we define $\mu_c(\mathbf{p})$ to be a function that is 1 at c's center and monotonically decreases as the distance from the center increases, taking the value 0 at c's defining edges. More specifically, we will use the following:

$$\mu_c(\mathbf{p}) = \begin{cases} 0 & \text{if } \mathbf{p} \notin c \\ 1 - \frac{\|\mathbf{p}-\mathbf{p_c}\|_\ell}{\max_{\mathbf{p'} \in c} \|\mathbf{p'}-\mathbf{p_c}\|_\ell} & \text{otherwise} \end{cases}, \qquad (1)$$

where $\mathbf{p_c}$ is the center point of box c and $\|\cdot\|_\ell$ denotes the ℓ-norm. The above equation measures the distance from $\mathbf{p}$ to c's center and normalizes by dividing by the radius of c under $\|\cdot\|_\ell$. Other possibilities include Gaussian-shaped functions and unnormalized linear functions (Lin and Lee, 1996).

In most of the multiple-instance work, the target concept is defined by a single box in $\mathcal{C}$ and hence the membership of point $\mathbf{p}$ in the target concept c is defined by $\mu_c(\mathbf{p})$. However, in our work we consider when the target concept is defined by any number of boxes. Hence, for each $C = \{c_1, \ldots, c_k\} \subseteq \mathcal{C}$ we must define $\mu_C(\mathbf{p})$ as a function of $\mu_{c_1}(\mathbf{p}), \ldots, \mu_{c_k}(\mathbf{p})$. We refer to the function used to combine $\mu_{c_1}(\mathbf{p}), \ldots, \mu_{c_k}(\mathbf{p})$ as the *membership combination function*. For ease of of exposition, we assume that $C = \{a, b\}$ but all of the definitions given below naturally extend to more than two concepts. Given real-valued boxes a and b, their union, intersection, and complement, respectively, are defined as: $\mu_{a \cup b}(\mathbf{p}) = \max\{\mu_a(\mathbf{p}), \mu_b(\mathbf{p})\}$, $\mu_{a \cap b}(\mathbf{p}) = \min\{\mu_a(\mathbf{p}), \mu_b(\mathbf{p})\}$, and $\mu_{\bar{a}}(\mathbf{p}) = 1 - \mu_a(\mathbf{p})$. The union, intersection and complement definitions are standard fuzzy combination operators. (See e.g. Lin and Lee (1996) or Klir and Yuan (1995) for more detail on these operators.)

In addition to the point membership function and the membership combination function, we need an *aggregation function f* to combine the labels for the individual points to obtain a label for the bag. There are three natural choices here. One possibility is that the highest label of any point in the bag should be used as the label for the bag. For example, in the drug discovery application, each point represents a possible conformation (shape) of a molecule and the label would be a measure of the strength of the bond when the molecule is in that shape. However, the molecule will eventually take the shape that has the strongest binding strength, and hence the binding strength for the bag (i.e the molecule) is really defined as the binding strength of the most desirable shape. Similarly, in scene classification (say in learning to classify an image as to whether it contains a waterfall), each point corresponds to one small area of the image and the label corresponds to the likelihood that the image is a waterfall based on that one area. The intuition is that if the image is a waterfall then one of these subimages will contain a feature that indicates it is a waterfall and otherwise, all of the subimages will have a small label. Hence, picking the maximum function for the aggregation function f would be most appropriate for

both of these cases. Another natural choice for f is to use the average function. This choice would be appropriate in applications such as that described in the next section where the label of each point in some way affects the label of the bag. In some settings, a weighted average may even be desired and f can easily be defined that way. Finally, one could use the minimum function for f. This choice is the closest match to the Hausdorff distance (Gruber, 1983) in that you want the portion of the image with the lowest label to determine the overall label.

3. Motivating Example

We now look at the particular problem that motivated this work so that we can see how this new learning model can be applied to it. Developing the ability to recognize a landmark from a visual image of a robot's current location is a fundamental problem in robotics. Consider a robot designed to navigate through a large-scaled environment[2]. 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.*

One approach to 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 should determine how close the robot is to L. Because the visual image may change significantly as small movements around L are made, the pattern matching approach encounters difficulties. Goldberg et al. (1996) first proposed approaching this problem by creating a set of positive examples (i.e. geometric patterns obtained using waveforms from locations in the vicinity of the landmark) and a set of negative examples (i.e. patterns obtained using waveforms from locations not in the vicinity of the landmark). Then they use a learning algorithm to construct a hypothesis to accurately predict if the robot is near the given landmark.

For a standard (boolean) geometric pattern C (Goldman et al., 2000a), a bag P is positive iff (1) each of the boxes $c \in C$ contains a point from P, *and* (2) each point in P lies in some box $c \in C$. This definition in inspired by the Hausdorff metric for measuring visual resemblance (Gruber, 1983) between two patterns/images. For this application, one can view each box as an area in which one expects the

---

[2] By a large-scaled environment we mean that not all landmarks are visible from all locations in the environment.

target image to have certain behavior. In the real-valued generalization, the label[3] of bag P measures the degree to which criteria (1) and (2) are satisfied.

Ideally, in a positive example, there would be a point in the center of each box. However, part of the motivation of using a learning approach for this pattern matching problem is the flexibility provided by the geometric concepts for handling variations between images of the same object obtained from slightly different locations and/or conditions. (One can think of these concept classes as being generalizations of the Hausdorff metric where weighted norms are used, and the weights can vary for each point.) Under the standard (boolean) formulation, a binary classification is made for each point (based on whether it is inside the box or not) and then these are combined to give a classification for the bag. A problem with this formulation is that one could have an example X_1 where all m points are very near the centers of the boxes and another example X_2 in which all m points are along the borders of the boxes. One would like a classification scheme that reflects that X_1 is more similar to the target than X_2. Our work resolves this important problem by using a real-valued model of membership for a point inside a box and then using real-valued aggregation operators to combine the points in the bag.

Similar to the approach first proposed by Goldberg et al. (1996) and later studied by Goldman et al. (2000a), we propose converting a d-dimensional visual image into a $(d+1)$-dimensional geometric pattern. The key difference here is that we would like to model this problem in a way that associates a real-valued label to each pattern versus just a boolean label. This is very important since if the navigation algorithm would like to know if it is at a desired landmark, it is more useful to receive a real value indicating how close it is to the landmark versus just a boolean output.

As discussed earlier, in most applications studied for the multiple-instance model, such as the scene classification problem or drug discovery problem, the target concept is defined by an axis-aligned box in d dimensional space. That is, a bag is classified as positive if and only if at least one point in the bag is in the box. In terms of the model described in the past section, in these settings $k = 1$ (the target concept is defined by a single box) and the natural aggregation function to use would be the maximum since it is the point with the highest label that should define the label for the bag.

---

[3] The labels could come from a human expert or signal processing system that we are trying to approximate.
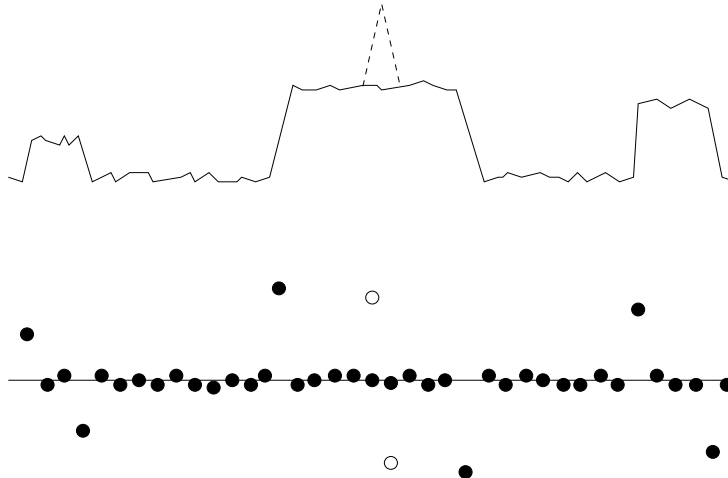
Figure 1. An example of mapping a one-dimensional waveform to a two-dimensional pattern. In the waveform W on the top the y-axis plots the intensity level and the x-axis is the index into the one-dimensional array representing the waveform. The figure on the bottom shows the two-dimensional geometric pattern created from W.

In the landmark recognition problem, a more complex multiple-instance classification method has been used. Consider a d-dimensional waveform W in which the data is representable as a constant-dimensional array of values, e.g. light intensities, sonar data, or temporal difference information. So, for example, in a 2-dimensional image for each (x, y)-value would be an intensity level. Or if you have a color image you could model this as a three-dimensional image where you have an intensity level for each x, y and $\{red, green, blue\}$ value. We then create a $(d+1)$-dimensional geometric pattern as follows. By normalizing the values of W and taking its derivative at different sampling points, we can map W to a set of points P in $(d + 1)$-dimensional space by computing the rate of change in W and then placing a point whenever the absolute value of the derivative of W is larger than some given threshold. The value used for the $(d + 1)$st dimension would be the value of the derivative. Figure 1 shows an example of mapping a one-dimensional waveform to a two-dimensional geometric pattern. We selected a one-dimensional waveform simply because it is easiest to illustrate. The waveform shown at the top of Figure 1 represents the target waveform (with the y-axis used to show the intensity level). The dashed boxes around the target waveform mark portions of the waveform that indicate some behavior that would be expected in any waveform taken near the landmark. In other words, the dashed boxes

around portions of the waveform W indicate components that must be translated, stretched, compressed, and scaled when relating another waveform W' to W. The geometric pattern P below W indicates the two-dimensional geometric pattern yielded by taking the derivative of W and using its value for the second (or in general, $(d+1)$st) dimension, which is the y-coordinate in the figure. The boxes around the points represent the target concept C. The boxes vary in size in each dimension to allow for certain components to vary more than others. Note that there are two types of solid boxes. First there are the boxes that overlap with the x-axis that directly correlate with the dashed boxes. They represent ranges of x for which the waveform has relatively small fluctuations. The other boxes represent areas in which a significant increase or decrease in the rate of change of the waveform is expected. So, one can view each box as an area in which one expects the target image to have certain behavior.

In the boolean domain, one would classify a waveform as positive if the geometric pattern obtained from it has a point in each of the solid boxes and each solid box has at least one point. This definition in inspired by the Hausdorff metric for measuring visual resemblance (Gruber, 1983) between two patterns/images. More formally, in the boolean domain the target concept is defined by a set C of k axis-aligned boxes. Here, a bag $\mathbf{x}$ is classified as positive if and only if (1) each of the $\leq k$ boxes $c \in C$ contains a point in $\mathbf{x}$, *and* (2) each point in $\mathbf{x}$ lies in some box $c \in C$. Our goal here is to expand this work to define a real-valued variation where the label captures the degree of resemblance.

In the most natural extension of the Hausdorff metric, the label for a point would be the maximum label given by any box in the target concept and the aggregation function to combine the points in a bag would be the minimum since the Hausdorff metric is defined by the point which has the lowest membership value. We now argue that this straightforward generalization of the Hausdorff metric is not the best option for this particular application. We return to Figure 1. Let W' be the waveform one would get in the top figure with the dashed spike and P' the pattern shown below that includes the open circles. Using the definition described above would yield a label of 0 because the labels for the two open circles (which are outliers) are 0 and hence the minimum aggregation function would yield a label of 0. If the spike is an important feature of the target waveform then a label of 0 for a waveform without the spike would be appropriate. However, that spike could also just be noise in the image. Since we expect the waveforms to have a high level of noise, we feel that using the minimum aggregation function attaches to much weight to outliers to be appropriate. Instead

we propose using an average aggregation function. Note that taking the average label of the points in P' yields a label that has been reduced by the two outliers but is still greater than 0.

For the remainder of this paper we focus on the following situation, which we feel is most appropriate for this particular application. (Other possibilities are considered in Section 7.) The label for bag $b = \{\mathbf{p}_1, \ldots, \mathbf{p}_m\}$ is the average value over the points, where for each point the maximum taken over the boxes in the target concept is used. That is, we use:

$$(1/m) \sum_{i=1}^{m} \max_{c \in C} \left\{ \mu_c(\mathbf{p}_i) \right\}. \tag{2}$$

At times, the derivations required for the virtual weights application (Section 6.2) are complicated by the maximum function in the above. Furthermore, for the pattern matching application we would expect that the target boxes are disjoint and hence each point lies in at most one target box. Under this restriction, the above equation is equivalent to

$$(1/m) \sum_{c \in C} \sum_{\mathbf{p} \in c} \mu_c(\mathbf{p}). \tag{3}$$

We use this formulation for presenting our main results. The results in Section 6 apply for any concept class and membership and aggregation functions as long as a multi-instance example can be evaluated in polynomial time. The particular choice for the membership and aggregation functions only affects the application of the virtual weights technique, which becomes significantly more complicated when the maximum function is used.

4. Related Learning Models

Recently, there has a been a significant amount of work in studying the multiple-instance model (Wang and Zucker, 2000; Long and Tan, 1998; Auer et al., 1998; Auer, 1997; Maron and Lozano-Pérez, 1998; Maron, 1998; Maron and Ratan, 1998; Blum and Kalai, 1998) in the boolean domain. Most of this work addresses the problem of learning a multiple-instance concept defined by a single axis-aligned box. There also has been some work on learning geometric patterns (Goldman et al., 2000a; Goldman and Scott, 1999; Goldberg et al., 1996) in the boolean domain and our work directly builds upon this work. In particular, the algorithm we propose is inspired by the algorithm given by Goldman et al. (2000a).

The learning model we define is an on-line learning model (Angluin, 1998; Littlestone, 1998) and hence builds upon some of the earlier results in the on-line learning model. In particular, we apply the results of Kivinen and Warmuth (1997a), which are described in more depth in the next section. In addition, we consider an agnostic learning model[4] in the sense that they make no assumptions whatsoever about the target concept to be learned. For a sequence of trials, the *best hypothesis* from a given comparison or "touchstone" class is defined to be the one that makes the minimum number of mistakes (or has the minimum loss). In the agnostic on-line learning algorithm, the learning algorithm's performance is compared with the performance of the best hypothesis from the touchstone class.

On the surface, our model has many similarities with the *p-concepts* model (Kearns and Schapire, 1994). A p-concept c over the domain $\mathcal{X}$ is a mapping $c : \mathcal{X} \to [0,1]$. For each $x \in \mathcal{X}$, $c(x)$ is interpreted as the probability that x is a positive example of the p-concept c. A p-concepts algorithm (to find a good *model of probability*) must infer a hypothesis $h : \mathcal{X} \to [0,1]$ that is a good real-valued approximation to the target concept c from labeled *boolean* examples (x, b) where b is one with probability $c(x)$ and zero with probability $1 - c(x)$. Clearly p-concepts are closely related to real-valued sets. One can view $c(x)$ as the degree to which x satisfies concept c. However, there are several important differences between p-concepts and real-valued concepts. First, consider the following example. The target concept c is "healthful foods". Suppose you are told $c(doughnut) = .2$. This does not mean if you eat a doughnut then there is a 20% chance that it will be healthful and an 80% chance that it will not be healthful. Instead it is giving a rating on a scale of 0 to 1 of the healthfulness of a doughnut.

Second, under this real-valued logic view, one expects the examples to be of the form $(x, c(x))$ and the goal is to predict $c(x')$ given x'. Thus in our real-valued learning model the examples are given real-valued labels in $[0,1]$ (versus the boolean labels of the p-concepts model). To further justify this decision, consider the problem of classifying students in terms of their suitability for admission to a particular university. Many factors (e.g. academic background, leadership potential, extracurricular activities) are used. For each of these factors there are different degrees to which the applicant meets the criteria. For exam-

---

[4] See Haussler (1992) and Kearns et al. (1994) for the definition of agnostic PAC-learning. Auer et al. (1996) first used the term agnostic on-line learning to refer to an on-line learning algorithm in which the loss bound is stated with respect to the best learner from a given touchstone class. Although on-line learning implies PAC learning (Angluin, 1998; Littlestone, 1998), it is not immediately clear whether on-line agnostic learning implies PAC agnostic learning.

ple, one would not want to simply say that an applicant has been a leader or has not been a leader. There is a spectrum between these two extremes. We want the learning algorithm to receive a value between $[0, 1]$ (obtained by aggregating many real-valued attributes) versus a 0 or 1 that was probabilistically selected.

Finally, by studying this type of model in a multiple-instance framework we are able to use the real-valued aggregation operators to provide a structured way to create real-valued concepts that are learnable. For example, Kearns and Schapire (1994) give an algorithm to learn the p-concepts class of all nondecreasing functions $c : \Re \to [0, 1]$. By combining real-valued intervals we can generalize this result to a more general class of functions that are not monotonic.

The *unreliable boundary queries (UBQ)* model (Blum et al., 1998) is designed to study situations in which the boundary between positive and negative examples is ill-defined. However, an important distinction is that in their model the classification of examples near the boundary is not important. Two other models that have somewhat similar motivations are the *unspecified attribute values (UAV)* model (Goldman et al., 1997; Birkendorf et al., 1998b; Bshouty and Wilson, 1999) and the *restricted focus of attention (RFA)* model (Ben-David and Dichterman, 1993; Birkendorf et al., 1998a). In both of these models the learner only sees some of the attributes (with differences in what determines this set). In the RFA model the goal is still to obtain the proper binary classification. However, in the UAV model the learner produces a ternary-valued hypothesis in which it predicts that an example x is positive, negative, or can't be determined. One can think of this as a very coarse real-valued scale (e.g. $[0, \epsilon], (\epsilon, 1 - \epsilon), [1 - \epsilon, 1]$).

5. Exponentiated Gradient and Gradient Descent

Our algorithms convert the geometric problem into a learning problem for which we can then apply the exponentiated gradient (EGU) algorithm [5] or the gradient descent (GD) algorithms. Thus, we briefly describe both algorithms and give some key results known about their performance. Both algorithms maintain a weight vector $\mathbf{w}_t$ that corresponds to the weights of the attributes at trial t. Given the instance $\mathbf{x}_t$ that corresponds to trial t, both algorithms predict[6] $\hat{y} = \mathbf{w}_t \cdot \mathbf{x}_t$. After receiving $\mathbf{x}_t$'s actual label y_t, GD updates its weights as follows:

---

[5] We speculate that our results are applicable to other EG variants, but we use EGU for brevity.

[6] Often the prediction is $\hat{y} = f(\mathbf{w}_t \cdot \mathbf{x}_t)$, where $f(\cdot)$ is a *transfer function* such as a sigmoidal function. In our work we let f be the identity function.

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta L'_{y_t}(\hat{y}_t)\mathbf{x}_t, \tag{4}$$

where η is a learning rate and $L'_{y_t} = (\partial L(y_t, z)/\partial z)_{z=\hat{y}_t}$, i.e. the gradient of the loss function L (in this paper, we use square loss, i.e. $L_{y_t}(\hat{y}_t) = (y_t - \hat{y}_t)^2$). EGU's update function is

$$\mathbf{w}_{t+1} = \mathbf{w}_t \cdot \exp\left(-\eta L'_{y_t}(\hat{y}_t)\mathbf{x}_t\right). \tag{5}$$

The results of Kivinen and Warmuth (1997a) are very general, including analyses for several variants of EG and on loss bounds when different transfer functions (e.g. a sigmoidal function) are applied to the neuron's output. Other related work can be found in Kivinen and Warmuth (1997b), Warmuth and Jagota (1997), Cesa-Bianchi et al. (1996), Long (1997), and Helmbold et al. (1996). We now state the results applied here. Loss $(\mathbf{w}, \mathcal{S})$ is the total loss on trial sequence $\mathcal{S}$ by $\mathbf{w}$, which represents an algorithm or a constant weight vector.

THEOREM 1. *(Kivinen and Warmuth, 1997a) Consider a sequence of trials $\mathcal{S} = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t)\rangle$, and values X, Y, and Z such that for all t, $\mathbf{x}_t \in [0, X]^N$, $y_t \in [0, Y]$, and $\|\mathbf{x}_t\|_2 \le Z$. For both GD and EGU, let the size-N initial weight vector $\mathbf{s} = (1/N, \ldots, 1/N)$. Then for any weight vector $\mathbf{u}$, GD's total loss on $\mathcal{S}$ is at most*

$$2\left(\mathrm{Loss}\,(\mathbf{u}, \mathcal{S}) + \|\mathbf{u} - \mathbf{s}\|_2^2 \, Z^2\right),$$

and EGU's total loss on $\mathcal{S}$ is at most

$$3\left(\mathrm{Loss}\,(\mathbf{u}, \mathcal{S}) + XY\left(1 + (\ln N - 1)\sum_{i=1}^{N} u_i + \sum_{i=1}^{N} u_i \ln u_i\right)\right).$$

Furthermore, if $\mathrm{Loss}\,(\mathbf{u}, \mathcal{S})$ is known a priori to be 0 then $\mathrm{Loss}\,(\mathrm{EGU}, \mathcal{S}) \le 2XY\left(1 + (\ln N - 1)\sum_{i=1}^{N} u_i + \sum_{i=1}^{N} u_i \ln u_i\right).$

6. The Algorithm

6.1. REDUCTION TO EGU AND GD

We obtain our loss bounds by reducing our geometric learning problem to one where we can apply either GD or EGU. The technique we use here is very general. In particular, our results apply for any concept class $\mathcal{C}$, membership function μ, membership combination function and aggregation function as long as any multiple-instance example from

the domain can be evaluated in polynomial time. The loss bounds we obtain will depend on $\log |\mathcal{C}|$ and the time complexity bounds (prior to the application of the virtual weights technique) will depend on $|\mathcal{C}|$. For ease of exposition, we have chosen to present our result for the particular geometric domain we are studying.

In general, our reduction involves enumerating all concepts from $\mathcal{C}$ (which in the geometric setting are all boxes in S^d) and associating attributes with them. Then we can apply either GD or EGU to learn the best linear combination of the attributes. In this section, we describe our reduction and give the loss bounds obtained by applying Theorem 1. Our reduction creates an exponential number of attributes and thus the predictions cannot be made in polynomial time. We then describe a novel application of the virtual weights technique to get a polynomial-time algorithm for some settings.

For all of our results (in this section and in Section 7), when we apply Theorem 1, N is the number of attributes, and since all labels and membership function outputs (which are the attributes) are in $[0, 1]$, we get $X = Y = 1$ and $Z = \sqrt{N}$. We use K to denote the number of *relevant* attributes. (Each non-relevant attribute will have a weight of 0 in the optimal weight vector.)

Under the assumption that $K \ll N$, the bounds we obtain using EGU are substantially better than those of GD, which is consistent with more general theoretical and empirical analyses (Kivinen and Warmuth, 1997a). Since we are only stating our bounds for this specific case, we only state EGU's bounds.

We now present our reduction and resulting loss bounds for learning real-valued geometric patterns labeled by sets of $\leq k$ axis-parallel boxes under the max combination function and the average aggregation operator. We define attributes that capture the degree to which the example is negative. Notice that here (in the noise-free case) there exists a weight function that gives perfect predictions.

THEOREM 2.   *On any sequence of trials $\mathcal{S} = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t)\rangle$, our algorithm for learning real-valued geometric patterns (using average aggregation) on multiple-instance examples of m points from $\{1, \ldots, s\}^d$ has loss on $\mathcal{S}$ of at most*

$$3\left(\mathrm{Loss}\left(\mathbf{u}, \mathcal{S}\right) + 2dk\ln s - k + 1\right).$$

Furthermore, if $\mathrm{Loss}\left(\mathbf{u}, \mathcal{S}\right)$ is known a priori to be 0 then we obtain the stronger loss bound of $4dk\ln s - 2k + 2$.

Proof: Enumerate all boxes in S^d and create one attribute A_b per box b, setting it to $(1/m)\sum_{\mathbf{p} \in b} \mu_b(\mathbf{p})$. So $N \leq s^{2d}$. Since the target concept

is defined by k boxes, there are only k relevant attributes and the target concept is defined by taking the average of these attributes. Without loss of generality, we can assume that $\sum_{i=1}^{N} u_i = 1$. Hence, the worst-case loss bound occurs when the optimal weight vector $\mathbf{u}$ has k entries of $1/k$ and the remaining entries are 0. Applying Theorem 1 yields the stated result.   □

6.2. Efficient Implementation with Virtual Weights

The problem that remains with the direct implementation of EGU or GD is that the number of attributes (and thus the computation time) is exponential in $\log s$, the number of bits required to specify part of the input. We now adapt the virtual weight technique of Maass and Warmuth (1998) to implicitly maintain the weights. The main problem is, given a group of boxes related to each other by some criterion, find a closed-form solution for the total contribution of those boxes' attributes to the total weighted sum. Finding an exact closed-form solution for the unweighted sum of $\mu_b(\mathbf{p})$ values can be difficult enough: e.g. the normalization factor in the denominator of Equation (1) implies that the closed-form solution might be at best approximated by a harmonic, depending on the norm used. But we must also include the boxes' weights, which are no longer equal within a group, though they are related.

We now demonstrate a setting when the virtual weights technique can be applied to this multiple-instance real-valued learning problem when Equation (3) is used to label the bags. Namely, we apply EGU with fixed boxes using the average combining formulation. We have selected this problem since it demonstrates the techniques without getting too complex. We start with the assumption that all boxes in the target concept are the same (with known radius r)[7] and the 1-norm is used. Thus if $\mathbf{p} \in b$, Equation (1) becomes:

$$\mu_b(\mathbf{p}) = 1 - \frac{\sum_{i=1}^{d} |p_i - c_{b,i}|}{r}, \tag{6}$$

where $c_{b,i}$ is the ith coordinate of the center of box b. We then set the value of A_b to be

$$A_b = \frac{1}{m} \sum_{\mathbf{p} \in b} \left(1 - \frac{\sum_{i=1}^{d} |p_i - c_{b,i}|}{r} \right).$$

---

[7] Note that, however, we can apply this procedure to the class of concepts in which each box can come from some finite set $\mathcal{B}$ by simply repeating the virtual weights procedure for each box type in $\mathcal{B}$.

For the remainder of this section we will focus on efficiently comput-
ing the contributions of the A_b attributes to the weighted sum by using
the virtual weight technique of Maass and Warmuth (1998). The basic
idea is to group all boxes (or more generally, concepts) that "behave
alike" (with respect to the points seen so far) into groups. Typically
when using this technique, all boxes in a group have the same weight
and hence a single "representative" box can be kept for each group
then one can multiply its weight by its attribute value and multiply
by the size of the group to compute the contribution of the group. As
we will explain in more depth, in our application each box in a group
will likely have a different weight and different attribute value and
hence the computation of the contribution of a group is more involved.
However, we are able to both keep the number of groups polynomial in
the number of trials and efficiently compute the contribution of each
group.

We now briefly review the procedure used by Goldman et al. (2000a)
to partition all the boxes of the space into groups (which is itself
an adaptation of Maass and Warmuth's (1998) algorithm for learning
unions of boxes in fixed dimension). Suppose we want to predict the
classification of an example P_t. Let $\mathcal{P} = \{\mathbf{p}_1, \ldots, \mathbf{p}_n\}$ be the set of
distinct points that have appeared in all the examples. Note that n is
at most m times the number of trials so far, where m is the maximum
number of points per example. Each of the d axis-parallel $(d - 1)$-
dimensional hyperplanes is passed through each point in $\mathcal{P}$, defining at
most $(n+1)^d$ regions (each of the form $(\ell_{1,i}, \ell_{1,i+1}] \times \cdots \times (\ell_{d,j}, \ell_{d,j+1}]$).
Let $\mathcal{R}$ be the set of these regions. They create a group for each region
containing all boxes completely within the region, and a group for each
pair of regions (R_1, R_2) containing all boxes with one corner in each of
R_1 and R_2 (see Figure 2).

The above groupings have the nice property that every box in a
group G contains the same set of points from $\mathcal{P}$ and hence has the same
weight in the boolean case. However, since we use real-valued boxes,
each point has a different membership value and so the predictions (and
hence the weights) differ. So we need to modify the way of defining the
groups. First, instead of having a group be a set of boxes, a group is a
set of *quadrants*, which are obtained as follows. Divide each box b into
2^d quadrants by passing a hyperplane parallel to each of the d axes
through the center of box b. Each resulting quadrant is given a tag of
the form $(\gamma_1, \ldots \gamma_d)$, where γ_i is 1 iff the ith dimensional coordinate of a
point in that quadrant is greater than that of the center. By maintaining
the property that the quadrants in each group contain the same set of
points from $\mathcal{P}$, we can compute the weight of each quadrant in the
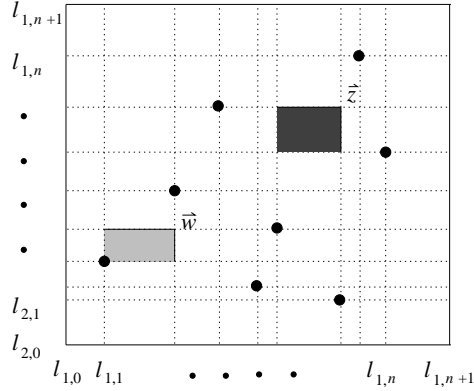group by using the weight of a single representative quadrant and then

Figure 2. This demonstrates (for the case of $d = 2$) how Goldman et al. (2000a) do the groupings for their virtual weight applications. The lightly shaded box (defined by **w** in its top right corner) is $R_{\mathbf{w}}$, and the more heavily shaded box (defined by **z** in its top right corner) is $R_{\mathbf{z}}$. The groups defined by $(R_{\mathbf{w}}, R_{\mathbf{z}})$ contain all boxes with the bottom left corner in the lightly shaded box and the top right corner in the heavily shaded box.

adjusting it by a simple formula that depends on the centers of the two quadrants. Although each group has an exponential number of quadrants whose predictions must be combined, we are able to compute a closed-form expression for the total contribution.

Let $\mathcal{P}_t$ be the set of all points seen up to and including the current trial t. We define a group corresponding to each triple (R_1, R_2, γ) where this group contains all quadrants with their center in R_1, their corner in R_2, and a tag of γ. (Note that for many of these triples, there will be no quadrants in the group and hence they are not needed.) Let $\mathcal{P}_{t,G} \subseteq \mathcal{P}_t$ be the set of all such points that are contained in group G. Without loss of generality, we will on focus on groups with a tag of $(0, \dots, 0)$. Since each quadrant is simply a box in S^d, we will use the term box interchangeably with quadrant. We call the box in G with the smallest possible coordinates for its center as the *defining box* $\check{b}_G$ of G. Let $\check{\mathbf{c}}_G$ be the center point for the box $\check{b}_G$.

We now compute $w_{t,\check{\mathbf{c}}}$, which is the weight of the attribute $A_{\check{\mathbf{c}}}$ associated with the representative box of a group. For ease of exposition, since we are looking at an arbitrary group G, we drop the G in the subscript. Let P_t be the set of points from trial t. Notice that $P_t \cap \check{b}$ is the set of points from P_t that lie in box $\check{b}$. If we use the square loss function $L = (\hat{y}_t - y_t)^2$, then the update function from Equation (5) yields

$$w_{t,\check{c}} = w_{1,\check{c}} \prod_{j=1}^{t-1} \exp\left(\frac{-2\eta}{n}\left(\hat{y}_j - y_j\right) \sum_{\mathbf{p} \in P_j \cap b_{\check{c}}} \left(1 - \sum_{i=1}^{d} \frac{\check{c}_i - p_i}{r}\right)\right).$$

Now suppose that box b (with center $\mathbf{c}$) is in the same group as $\check{b}$. Furthermore, let $q_i = c_i - \check{c}_i$ and recall that all initial weights are equal. Then

$$
\begin{aligned}
w_{t,\mathbf{c}} &= w_{1,\mathbf{c}} \prod_{j=1}^{t-1} \exp\left(\frac{-2\eta}{m}(\hat{y}_j - y_j) \sum_{\mathbf{p} \in P_j \cap b} \left(1 - \sum_{i=1}^{d} \frac{c_i - p_i}{r}\right)\right) \\
&= w_{1,\check{c}} \prod_{j=1}^{t-1} \exp\left(\frac{-2\eta}{m}(\hat{y}_j - y_j) \sum_{\mathbf{p} \in P_j \cap \check{b}} \left[1 - \left(\sum_{i=1}^{d}\left(\frac{\check{c}_i - p_i}{r}\right) + \sum_{i=1}^{d} \frac{q_i}{r}\right)\right]\right) \\
&= w_{t,\check{c}} \prod_{j=1}^{t-1} \exp\left(\frac{2\eta}{m}(\hat{y}_j - y_j)\left|P_j \cap \check{b}\right| \sum_{i=1}^{d} \frac{q_i}{r}\right).
\end{aligned}
\tag{7}
$$

Let

$$Y_{G,t-1} = \exp\left(\frac{2\eta}{m} \sum_{j=1}^{t-1} (\hat{y}_j - y_j)\left|P_j \cap \check{b}\right|\right).$$

Note that Y is easy to update since $Y_{G,t} = Y_{G,t-1}\cdot\exp\left(\frac{2\eta}{n}(\hat{y}_t - y_t)\left|P_t \cap \check{b}\right|\right)$. For $\mathbf{c}$ the center of a box in the group G (whose representative box has center $\check{\mathbf{c}}$), let

$$Z_{G,\mathbf{c},t-1} = (Y_{G,t-1})^{\frac{1}{r}\sum_{i=1}^{d} c_i}.$$

Using this notation, we can rewrite Equation (7) as

$$w_{t,\mathbf{c}} = w_{t,\check{c}} \frac{Z_{G,\mathbf{c},t-1}}{Z_{G,\check{c},t-1}}.$$

When a new example P_t arrives for trial t, we recompute Z for each group. Then the contribution to the prediction by group G is

$$\frac{1}{m} \sum_{b \in G} \sum_{\mathbf{p} \in P_t \cap b} \mu_b(\mathbf{p})\, w_{t,\mathbf{c}_b},$$

where $\mathbf{c}_b$ is the center for box b. Let $c_i^{\min}$ and $c_i^{\max}$ be, respectively, tight lower and upper values for the ith coordinate of the boxes in group G.

Then expanding the above equation gives the following for group G's contribution:

$$\frac{1}{m} \sum_{c_1=c_1^{min}}^{c_1^{max}} \cdots \sum_{c_d=c_d^{min}}^{c_d^{max}} \sum_{\mathbf{p}\in P_t\cap \check{b}} \left[\left(1 - \frac{1}{r}\sum_{i=1}^{d}(p_i - c_i)\right) \cdot w_{t,\check{\mathbf{c}}} \frac{Z_{G,\mathbf{c},t-1}}{Z_{G,\check{\mathbf{c}},t-1}} \right]$$

$$= \frac{\left|P_t\cap \check{b}\right| \cdot w_{t,\check{\mathbf{c}}}}{m\cdot Z_{G,\check{\mathbf{c}},t-1}} \sum_{c_1=c_1^{min}}^{c_1^{max}} \cdots \sum_{c_d=c_d^{min}}^{c_d^{max}} Z_{G,\mathbf{c},t-1}$$

$$-\frac{w_{t,\check{\mathbf{c}}}}{mr} \sum_{c_1=c_1^{min}}^{c_1^{max}} \cdots \sum_{c_d=c_d^{min}}^{c_d^{max}} \sum_{\mathbf{p}\in P_t\cap \check{b}} \left[\sum_{i=1}^{d}(p_i - c_i)\cdot \frac{Z_{G,\mathbf{c},t-1}}{Z_{G,\check{\mathbf{c}},t-1}} \right].$$

Let $Q_{G,t} = w_{t,\check{\mathbf{c}}}/\left(m\cdot Z_{G,\check{\mathbf{c}},t-1}\right)$. Then the contribution from group G is

$$\left|P_t\cap \check{b}\right| \quad \cdot \quad Q_{G,t} \sum_{c_1=c_1^{min}}^{c_1^{max}} \cdots \sum_{c_d=c_d^{min}}^{c_d^{max}} Z_{G,\mathbf{c},t-1}$$

$$- \quad \frac{Q_{G,t}}{r} \left(\sum_{\mathbf{p}\in P_t\cap \check{b}} \sum_{i=1}^{d} p_i \right) \sum_{c_1=c_1^{min}}^{c_1^{max}} \cdots \sum_{c_d=c_d^{min}}^{c_d^{max}} Z_{G,\mathbf{c},t-1}$$

$$+ \quad \frac{Q_{G,t}\cdot \left|P_t\cap \check{b}\right|}{r} \sum_{c_1=c_1^{min}}^{c_1^{max}} \cdots \sum_{c_d=c_d^{min}}^{c_d^{max}} \left(Z_{G,\mathbf{c},t-1}\sum_{i=1}^{d} c_i \right).$$

Now all that remains to make our algorithm efficient is to find an efficient way to compute each of the above three terms without running through the chain of sums $\sum_{c_1=c_1^{\min}}^{c_1^{\max}} \cdots \sum_{c_d=c_d^{\min}}^{c_d^{\max}}$. Let $Y_{G,t-1}' = (Y_{G,t-1})^{1/r}$. Hence we must compute

$$\sum_{c_1=c_1^{\min}}^{c_1^{\max}} \cdots \sum_{c_d=c_d^{\min}}^{c_d^{\max}} Z_{G,\mathbf{c},t-1} = \sum_{c_1=c_1^{\min}}^{c_1^{\max}} \cdots \sum_{c_d=c_d^{\min}}^{c_d^{\max}} \left(Y_{G,t-1}'\right)^{\sum_{i=1}^{d} c_i} \tag{8}$$

and

$$\sum_{c_1=c_1^{\min}}^{c_1^{\max}} \cdots \sum_{c_d=c_d^{\min}}^{c_d^{\max}} \left(Z_{G,\mathbf{c},t-1}\sum_{i=1}^{d} c_i \right)$$

$$= \sum_{c_1=c_1^{\min}}^{c_1^{\max}} \cdots \sum_{c_d=c_d^{\min}}^{c_d^{\max}} \left[\left(Y_{G,t-1}'\right)^{\sum_{i=1}^{d} c_i} \cdot \sum_{i=1}^{d} c_i \right]. \tag{9}$$

In solving for Equation (8), we get

$$\left(Y'_{G,t-1}\right)^{\sum_{i=1}^{d} c_i^{\min}} \left[\sum_{j_1=0}^{c_1^{\max}-c_1^{\min}} \cdots \sum_{j_d=0}^{c_d^{\max}-c_d^{\min}} \left(Y'_{G,t-1}\right)^{\sum_{i=1}^{d} j_i}\right]$$

$$= \left(Y'_{G,t-1}\right)^{\sum_{i=1}^{d} c_i^{\min}} \left[\left(\sum_{j_1=0}^{c_1^{\max}-c_1^{\min}} \left(Y'_{G,t-1}\right)^{j_1}\right) \cdots \left(\sum_{j_d=0}^{c_d^{\max}-c_d^{\min}} \left(Y'_{G,t-1}\right)^{j_d}\right)\right]$$

$$= \left(Y'_{G,t-1}\right)^{\sum_{i=1}^{d} c_i^{\min}} \prod_{j=1}^{d} \left(\frac{\left(Y'_{G,t-1}\right)^{c_j^{\max}-c_j^{\min}+1} - 1}{Y'_{G,t-1} - 1}\right)$$

$$= \prod_{j=1}^{d} \left(\frac{\left(Y'_{G,t-1}\right)^{c_j^{\max}+1} - \left(Y'_{G,t-1}\right)^{c_j^{\min}}}{Y'_{G,t-1} - 1}\right).$$

Since Equation (9) is the product of $Y'_{G,t-1}$ and the derivative of Equation (8) with respect to $Y'_{G,t-1}$, we get for Equation (9)

$$Y'_{G,t-1} \frac{d}{dY'_{G,t-1}} \prod_{j=1}^{d} \left(\frac{\left(Y'_{G,t-1}\right)^{c_j^{\max}+1} - \left(Y'_{G,t-1}\right)^{c_j^{\min}}}{Y'_{G,t-1} - 1}\right)$$

$$= Y'_{G,t-1} \sum_{i=1}^{d} \left(\frac{d}{dY'_{G,t-1}} \left(\frac{\left(Y'_{G,t-1}\right)^{c_i^{\max}+1} - \left(Y'_{G,t-1}\right)^{c_i^{\min}}}{Y'_{G,t-1} - 1}\right)\right)$$

$$\cdot \prod_{j \neq i} \left(\frac{\left(Y'_{G,t-1}\right)^{c_j^{\max}+1} - \left(Y'_{G,t-1}\right)^{c_j^{\min}}}{Y'_{G,t-1} - 1}\right)$$

$$= Y'_{G,t-1} \sum_{i=1}^{d} \left(\frac{\left(Y'_{G,t-1}\right)^{c_i^{\max}} \left(c_i^{\max} \left(Y'_{G,t-1} - 1\right) - 1\right)}{\left(Y'_{G,t-1} - 1\right)^2}\right.$$

$$\left. - \frac{\left(Y'_{G,t-1}\right)^{c_i^{\min}-1} \left(c_i^{\min} \left(Y'_{G,t-1} - 1\right) + Y'_{G,t-1}\right)}{\left(Y'_{G,t-1} - 1\right)^2}\right)$$

$$\cdot \prod_{j \neq i} \left(\frac{\left(Y'_{G,t-1}\right)^{c_j^{\max}+1} - \left(Y'_{G,t-1}\right)^{c_j^{\min}}}{Y'_{G,t-1} - 1}\right).$$

7. Other Aggregation Operators

We now demonstrate how to apply our results to other choices for the membership combination function and other aggregation options. We first consider the situation in which the target concept is defined by a union of the boxes. Hence $\mu_{c_1 \cup \cdots \cup c_k}(\mathbf{p}) = \max_{\mu_{c_1}(\mathbf{p}),\ldots,\mu_{c_k}(\mathbf{p})}$. Here, the maximum function is used for the membership combination and the maximum function is also used as the aggregation operation. Since EGU uses a weighted sum over the attributes, we are approximating the maximum of the relevant attributes by a weighted average.

THEOREM 3. *On any sequence of trials $\mathcal{S} = \langle (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t) \rangle$, our algorithm for learning the union of at most k axis-parallel real-valued boxes with multiple-instance examples of m points from $\{1, \ldots, s\}^d$ has loss on $\mathcal{S}$ of at most*

$$3 \left(\mathrm{Loss}\left(\mathbf{u}, \mathcal{S} \right) + 2d \ln s - \ln k \right).$$

Furthermore, if $\mathrm{Loss}\left(\mathbf{u}, \mathcal{S} \right)$ is known a priori to be 0 then we obtain the stronger loss bound of $2d \ln s - \ln k$.

Proof: Enumerate all possible boxes in S^d and create one attribute A_b per box b, setting $A_b = \max_{\mathbf{p} \in P} \{ \mu_b(\mathbf{p}) \}$. Thus $N \leq s^{2d}$. Since the target concept is defined by k boxes, there are only k relevant attributes and the target concept is defined by taking the maximum value from among these attributes. Without loss of generality, we can assume that $\sum_{i=1}^N u_i = 1$. Hence, the worst-case loss bound occurs when the optimal weight vector $\mathbf{u}$ has k entries of $1/k$ and the remaining entries are 0. Applying Theorem 1 yields the stated result.   □

We now demonstrate how to apply the virtual weights technique to this membership combination function and aggregation option. We assume that all boxes in the target concept are squares with the same known radius r and the 1-norm is used. Thus assuming that $\mathbf{p} \in b$, Equation (1) becomes:

$$\mu_b(\mathbf{p}) = 1 - \frac{\sum_{i=1}^d |p_i - c_{b,i}|}{r},$$

where $c_{b,i}$ is the ith coordinate of the center of box b. To remove the absolute values from the sum, split the above into 2 sums:

$$\mu_b(\mathbf{p}) = 1 - \frac{1}{r} \left[\sum_{i=1, p_i \geq c_i}^d (p_i - c_i) + \sum_{i=1, p_i < c_i}^d (c_i - p_i) \right].$$
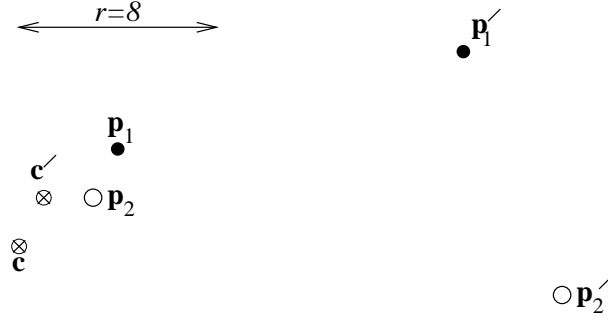
Figure 3. Two examples (sets of points) and each point's region of influence.

For box b, we then set the value of A'_b to be the maximum possible value of the above equation over all $\mathbf{p} \in P$, where P is the set of points in the current example.

Note that each box b in the space either has an attribute value $A'_b = 0$ (if it contains no point from the current example P) or its value is equal to $\max_{\mathbf{p} \in b} \{\mu_b(\mathbf{p})\}$. Thus the assignment of values to the attributes is completely determined by the points in P. For a point $\mathbf{p} \in P$ and a radius $r > 0$, we define $\mathbf{p}$'s *region of influence* as

$$R_r(\mathbf{p}) = \left\{ \mathbf{c} \in S^d : \|\mathbf{p} - \mathbf{c}\|_\ell \leq r \text{ and } \|\mathbf{p}' - \mathbf{c}\|_\ell \geq \|\mathbf{p} - \mathbf{c}\|_\ell \ \forall \mathbf{p}' \in P \right\}.$$

In other words, $R_r(\mathbf{p})$ is the set of centers of boxes whose attributes A' are determined by $\mathbf{p}$. $R_r(\mathbf{p})$ can be thought of as the intersection between the box with radius r centered at $\mathbf{p}$ and $\mathbf{p}$'s share of a Voronoi tesselation of S^d under the ℓ-norm. Any box b in S^d whose center is not in $R_r(\mathbf{p})$ for any $\mathbf{p} \in P$ has $A'_b = 0$.

Now for example, consider the first instance ($P_1 = \{\mathbf{p}_1, \mathbf{p}'_1\}$) seen, with $n = 2$ points, as in Figure 3. Let $r = 8$, which is also the length of each box edge since the 1-norm is used. The points' regions of influence are (solid) boxes of radius 8 centered at those two points. Since these boxes do not intersect, we need not concern ourselves with the Voronoi tesselation. We start EG with $\mathbf{w}_1 = \mathbf{1}$ and use the identity function as our transfer function.

Since we can write

$$w_{t,\mathbf{c}} = w_{1,\mathbf{c}} \ \exp \left(-2\eta \sum_{j=1}^{t-1} \left[(\hat{y}_j - y_j) \left(\sum_{i=1}^{d} \frac{p_{j,i} - c_i}{r} \right) \right] \right),$$
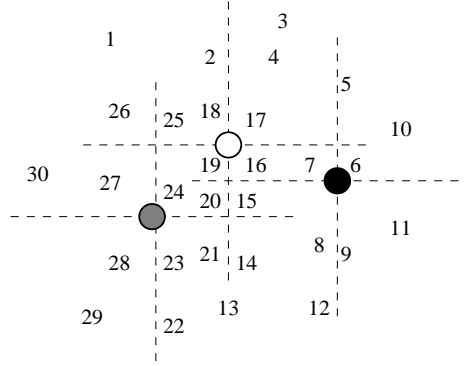
Figure 4. Three points from different trials and the groups they define.

then in general, if boxes $\mathbf{c}$ and $\mathbf{c}'$ have always been influenced by the same point $\mathbf{p}_t$ for each trial t and $p_{t,i} \geq c_{t,i} \, \forall \, i$, then

$$
\begin{aligned}
w_{t,\mathbf{c}'} &= w_{1,\mathbf{c}'} \exp \left(-2\eta \sum_{j=1}^{t-1} \left[(\hat{y}_j - y_j) \left(\sum_{i=1}^{d} \frac{p_{j,i} - c_i'}{r} \right) \right] \right) \\
&= w_{1,\mathbf{c}} \exp \left(-2\eta \sum_{j=1}^{t-1} \left[(\hat{y}_j - y_j) \left(\sum_{i=1}^{d} \frac{p_{j,i} - c_i}{r} - \sum_{i=1}^{d} \frac{q_i}{r} \right) \right] \right) \\
&= w_{t,\mathbf{c}} \exp \left(2\eta \left(\sum_{i=1}^{d} \frac{q_i}{r} \right) \left(\sum_{j=1}^{t-1} (\hat{y}_j - y_j) \right) \right).
\end{aligned}
\tag{10}
$$

Thus assuming no two regions of influence intersect per trial (e.g. Figure 3), then we can partition each square into 2^d quadrants (to account for the absolute values). The intersection of any subset of the quadrants comprises a group (e.g. Figure 4). We choose one representative $\mathbf{c}$ per group and can use Equation (10) (or a variant, to account for absolute value) to compute the weight of any other attribute in the group.

When a new example P_t arrives for trial t, we compute $\mathbf{p}$'s region of influence for each $\mathbf{p} \in P_t$, and (assuming no two ROIs intersect for this trial) partition the space into groups using the previous $t - 1$ ROIs. Now consider all boxes $\mathbf{c}$ in a group $G \in R_r(\mathbf{p})$ (we assume w.l.o.g. that all these boxes lie in a "lower left quadrant" w.r.t. $\mathbf{p}$). Group G's contribution to the prediction is

$$
\sum_{\mathbf{c} \in G} \mu_{\mathbf{c}}(\mathbf{p}) \, w_{\mathbf{c},t}.
$$

Let $c_i^{\min}$ and $c_i^{\max}$ be, respectively, the lower and upper values for the ith coordinate of any box in G. Also, let $\mathbf{c}_G$ be the representative box for group G and let $c_{G,i}$ be its ith coordinate. Expanding the above equation gives the following for group G's contribution, where we let $Y_{G,t-1} = \exp\left((2\eta/r) \left(\sum_{j=1}^{t-1} (\hat{y}_j - y_j) \right) \right)$:

$$
\sum_{c_1=c_1^{\min}}^{c_1^{\max}} \cdots \sum_{c_d=c_d^{\min}}^{c_d^{\max}} \left[\left(1 - \sum_{i=1}^{d} \frac{p_i - c_i}{r} \right) \left(w_{t,\mathbf{c}_G} \left(Y_{G,t-1} \right)^{\left(\sum_{i=1}^{d} q_i \right)} \right) \right]
$$

$$
= \frac{w_{t,\mathbf{c}_G}}{r} \sum_{c_1=c_1^{\min}}^{c_1^{\max}} \cdots \sum_{c_d=c_d^{\min}}^{c_d^{\max}} \left[\left(r - \sum_{i=1}^{d} p_i \right) (Y_{G,t-1})^{\left(\sum_{i=1}^{d} q_i \right)} - \left(\sum_{i=1}^{d} c_i \right) (Y_{G,t-1})^{\left(\sum_{i=1}^{d} q_i \right)} \right]
$$

$$
= \left(\frac{w_{t,\mathbf{c}_G}}{r \left(Y_{G,t-1} \right)^{\left(\sum_{i=1}^{d} c_{G,i} \right)}} \right) \left(r - \sum_{i=1}^{d} p_i \right) \sum_{c_1=c_1^{\min}}^{c_1^{\max}} \cdots \sum_{c_d=c_d^{\min}}^{c_d^{\max}} (Y_{G,t-1})^{\left(\sum_{i=1}^{d} c_i \right)}
$$

$$
- \left(\frac{w_{t,\mathbf{c}_G}}{r \left(Y_{G,t-1} \right)^{\left(\sum_{i=1}^{d} c_{G,i} \right)}} \right) \sum_{c_1=c_1^{\min}}^{c_1^{\max}} \cdots \sum_{c_d=c_d^{\min}}^{c_d^{\max}} \left(\sum_{i=1}^{d} c_i \right) (Y_{G,t-1})^{\left(\sum_{i=1}^{d} c_i \right)} .
$$

The last equality holds since $q_i = c_i - c_{G,i}$. This can be solved very similarly to Equations (8) and (9).

Another interesting option is when the maximum function is used for the membership combination and the minimum is used as the aggregation operation. That is, given a set of boxes, a point receives the maximum label given to it by any of the boxes, and the label for a bag is the minimum label given to any point in the bag: $y = \min_{\mathbf{p} \in P}\{\max_{c \in C}\{\mu_c(\mathbf{p})\}\}$. To learn this class, instead of combining attributes that capture the degree to which the example is positive, we instead define attributes that capture the degree to which the example is negative. These attributes are then combined to obtain a prediction of the degree to which the example is negative. By then returning 1 minus this quantity, we obtain our prediction of the degree to which the example is positive.

Let C be the set of boxes defining the target concept. To approximate the degree to which the "max-min" condition is not satisfied, we assume there is some set of boxes $\bar{C}$ (such that $|\bar{C}| = poly(|C|)$) whose union comprises the complement of the union of boxes from C and such that the degree to which (2) is not satisfied is $\max_{\mathbf{p} \in P}\{\max_{c \in \bar{C}}\{\mu_c(\mathbf{p})\}\} = \max_{c \in \bar{C}}\{\max_{\mathbf{p} \in P}\{\mu_c(\mathbf{p})\}\}$. We assume that $\bar{C}$ is defined as part of the target concept along with C.

THEOREM 4. *On any sequence of trials $\mathcal{S} = \langle(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_t, y_t)\rangle$, our algorithm for learning real-valued geometric patterns (using max combination and min aggregation) on multiple-instance examples of m points from $\{1, \ldots, s\}^d$ has loss on $\mathcal{S}$ of at most*

$$3\left(\text{Loss}\left(\mathbf{u}, \mathcal{S}\right) + 2d\ln(2s) - \ln\left(|C| + |\bar{C}|\right)\right),$$

where C and $\bar{C}$ define the target concept. Furthermore, if $\text{Loss}\left(\mathbf{u}, \mathcal{S}\right)$ is known a priori to be 0 then we obtain the stronger loss bound of

$$2\left(2d\ln(2s) - \ln\left(|C| + |\bar{C}|\right)\right).$$

Proof: Enumerate all possible boxes in S^d and for each box b create an attribute A_b, setting it to $\max_{\mathbf{p}\in P}\{\mu_b(\mathbf{p})\}$. Here $N \leq (2s)^{2d}$. Since the target concept is defined by $|C| + |\bar{C}|$ boxes, there are only $K = |C| + |\bar{C}|$ relevant attributes and the target concept is defined by taking the maximum value from among these attributes. Without loss of generality, we can assume that $\sum_{i=1}^{N} u_i = 1$. Hence, the worst-case loss bound occurs when the optimal weight vector $\mathbf{u}$ has K entries of $1/K$ and the remaining entries are 0. Applying Theorem 1 yields the stated result. $\square$

The computation for A_b for max-max is the same as that for A_b for max-min, though the interpretation is different.

8. Other Results and Future Directions

We believe that we can easily substitute other variants of EG for EGU in our work, and use different loss and transfer functions (e.g. sigmoidal functions) at the node's output. Also, rather than have a single box type with fixed radius, we can allow boxes to come from a known finite set $\mathcal{B}$. For computing virtual weights with the average approach (and others), we can do this as follows: for each group G, run through all boxes $b \in \mathcal{B}$ that fit the points contained in G. This will have time polynomial in $|\mathcal{B}|$ and $|\mathcal{P}|$. A natural extension to this work is to allow the target boxes to be arbitrary within the discrete, bounded space. Obviously the above approach will not work efficiently, so a new technique is required. The main issue is that arbitrary boxes implies widely varying radii, so the radius in the denominator of Equation (1) varies in the sum, and hence our current virtual weights technique does not apply. It is likely that the closed form of our summations would have harmonics, which we can at best approximate with logarithms. If this occurs, perhaps the loss bounds for EGU and GD can be made to work with this approximate simulation, which would accommodate the harmonic.

Of course, other membership functions are possible as well. For example, we could use the function of Equation (1) using the 2- or ∞-norm. We believe that the summations could be solved for these cases, but they become more complicated than for the 1-norm. Other membership functions that we can try include Gaussian-shaped functions and the following *unnormalized linear functions* (Lin and Lee, 1996):

$$\mu_b(\mathbf{p}) = \frac{1}{d} \sum_{i=1}^{d} \left(1 - f\left(p_i - M_{b,i}, \alpha\right) - f\left(m_{b,i} - p_i, \alpha\right) \right),$$

where

$$f(z, \alpha) = \begin{cases} 1 & \text{if } z\alpha > 1 \\ z\alpha & \text{if } 0 \leq z\alpha \leq 1, \\ 0 & \text{if } z\alpha < 0 \end{cases}$$

$m_{b,i}$ is the minimum value of box b in dimension i, $M_{b,i}$ is the maximum value of box b in dimension i, and α is a "ramp" parameter that determines how quickly the membership function decreases as distance from the box's edges increases. So $\mu_b(\mathbf{p}) = 1$ if $\mathbf{p}$ lies entirely inside b and $= 0$ if $\mathbf{p}$ lies entirely outside b. This function has the advantage that no normalization is used, so virtual weights should be applicable even when $\mathcal{B}$ is the set of all possible boxes in S^d. We could even allow the target concept to use one of several possible values for α for each box. This should be easily accommodated if the set of possible values is polynomially sized.

The previous paragraph implied a method of learning the correct membership function for each box while simultaneously learning the boxes. Of course, in these cases the size of the set of possible functions was finite. An interesting question is whether we can learn the appropriate functions from an infinite set while simultaneously learning the boxes.

It is also possible for this algorithm to work in constant-dimensional real space. First draw an unlabeled sample, placing all points into a set $\mathcal{P}$. Then enumerate a set of boxes[8] B such that each box $b \in B$ contains a distinct subset of the points of $\mathcal{P}$. Then attach attributes to each box as in Section 6 and run GD or EG. Here the virtual weights technique is unnecessary since the number of attributes is polynomial in the input size. If all examples are drawn i.i.d. from a fixed distribution, then the result is an algorithm with an agnostic expected on-line mistake bound that can be converted to an agnostic PAC algorithm. To determine an

---

[8] This technique should also work for other components with bounded complexity, e.g. hyperellipsoids or irregular cross-polytopes.

appropriate size of the unlabeled sample, we would first bound the *fat-shattering dimension* (Kearns and Schapire, 1994; Bartlett et al., 1996) of the components and then bound the FSD of k such components combined by our aggregation operator.

Finally, we plan to empirically evaluate our results on the problems in image classification, content-based image retrieval, and pattern matching used to motivate our work. The examples could be real-valued geometric patterns that come from data labeled by a human expert as to how closely it resembles an ideal piece of data. We would also like to apply this algorithm to multiple-instance examples labeled by a real-valued method such as data from a real-valued image database.

Acknowledgements

References

Angluin, D: 1988, 'Queries and concept learning'. *Machine Learning* **2**(4), 319–342.

Auer, P.: 1997, 'On learning from multi-instance examples: Empirical evaluation of a theoretical approach'. In: *Proc. 14th International Conference on Machine Learning.* pp. 21–29.

Auer, P., S. Kwek, W. Maass, and M.K. Warmuth: 1996, 'Learning of depth two neural networks with constant fan-in at the hidden nodes'. In: *Proceedings of the Ninth Annual Conference on Computational Learning Theory*, pp. 333–343.

Auer, P., P. M. Long, and A. Srinivasan: 1998, 'Approximating hyper-rectangles: Learning and pseudo-random sets'. *Journal of Computer and System Sciences* **57**(3), 376–388.

Bartlett, P. L., P. M. Long, and R. C. Williamson: 1996, 'Fat-shattering and the learnability of real-valued functions'. *Journal of Computer and Systems Sciences* **52**(3), 434–452.

Ben-David, S. and E. Dichterman: 1993, 'Learning with restricted focus of attention'. In: *Proc. 6th Annu. Workshop on Comput. Learning Theory.* pp. 287–296.

Birkendorf, A., E. Dichterman, J. Jackson, N. Klasner, and H. U. Simon: 1998a, 'On restricted-focus-of-attention learnability of Boolean functions'. *Machine Learning* **30**, 89–123.

Birkendorf, A., E. Dichterman, N. Klasner, and H. U. Simon: 1998b, 'Structural results about exact learning with unspecified attribute values'. In: *Proceedings of the Eleventh Annual Conference on Computational Learning Theory.* pp. 144–153.

Blum, A., P. Chalasani, S. Goldman, and D. Slonim: 1998, 'Learning with unreliable boundary queries'. *Journal of Computer and System Sciences* **56**(2), 209–222.

Blum, A. and A. Kalai: 1998, 'A note on learning from multiple-instance examples'. *Machine Learning* **30**, 23–29.

Bshouty, N. H. and D. K. Wilson: 1999, 'On learning in the presence of unspecified attribute values'. In: *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*. pp. 81–87.

Cesa-Bianchi, N., P. Long, and M. K. Warmuth: 1996, 'Worst-case quadratic loss bounds for prediction using linear functions and gradient descent'. *IEEE Transactions on Neural Networks* **7**, 604–619.

Dietterich, T. G., R. H. Lathrop, and T. Lozano-Perez: 1997, 'Solving the Multiple-Instance Problem with Axis-Parallel Rectangles'. *Artificial Intelligence* **89**(1–2), 31–71.

Goldberg, P. W., S. A. Goldman, and S. D. Scott: 1996, 'PAC learning of one-dimensional patterns'. *Machine Learning*, **25**(1), 51–70.

Goldman, S. A., S. K. Kwek, and S. D. Scott: 2000a, 'Agnostic learning of geometric patterns'. *Journal of Computer and System Sciences*. To appear. Early version in COLT '97.

Goldman, S. A., S. K. Kwek, and S. D. Scott: 1997, 'Learning from examples with unspecified attribute values'. In: *Proceedings of the Tenth Annual Conference on Computational Learning Theory*, pp. 231–242. Also as technical report WUCS-98-28, Washington University in St. Louis, 1998.

Goldman, S. A. and S. D. Scott: 1999, 'A Theoretical and Empirical Study of a Noise-Tolerant Algorithm to Learn Geometric Patterns'. *Machine Learning* **37**(1), 5–49.

Gruber, P. M.: 1983, 'Approximation of Convex Bodies'. In: P. M. Gruber and J. M. Willis (eds.): *Convexity and its Applications*. Brikhäuser Verlag.

Haussler, D.: 1992, 'Decision Theoretic Generalizations of the PAC Model for Neural Net and Other Learning Applications'. *Inform. Comput.* **100**(1), 78–150.

Helmbold, D. P., J. Kivinen, and M. K. Warmuth: 1996, 'Worst-case Loss Bounds for Single Neurons'. Technical Report UCSC-CRL-96-2, Univ. of Calif. Computer Research Lab, Santa Cruz, CA. Early version in NIPS 8, 1996.

Kearns, M. J. and R. E. Schapire: 1994, *Efficient distribution-free learning of probabilistic concepts*, Vol. I: Constraints and Prospects, Chapt. 10, pp. 289–329. MIT Press. Earlier version appeared in FOCS90.

Kearns, M. J., R. E. Schapire, and L. M. Sellie: 1994, 'Toward efficient agnostic learning'. *Machine Learning* **17**(2/3), 115–142.

Kivinen, J. and M. K. Warmuth: 1997a, 'Exponentiated gradient versus gradient descent for linear predictors'. *Information and Computation* **132**(1), 1–63.

Kivinen, J. and M. K. Warmuth: 1997b, 'Relative loss bounds for multidimensional regression problems'. In: *Proc. 1997 Neural Information Processing Conference*. pp. 287–293.

Klir, G. J. and B. Yuan: 1995, *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall.

Lin, C.-T. and C. S. G. Lee: 1996, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Prentice Hall.

Littlestone, N.: 1988, 'Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm'. *Machine Learning*, **2**, 285–318.

Long, P. M.: 1997, 'On-line evaluation and prediction using linear functions'. In: *Proc. 10th Annu. Conf. on Comput. Learning Theory*. pp. 21–31.

Long, P. M. and L. Tan: 1998, 'PAC learning axis-aligned rectangles with respect to product distributions from multiple-instance examples'. *Machine Learning* **30**, 7–21.

Maass, W. and M. K. Warmuth: 1998, 'Efficient learning with virtual threshold gates'. *Information and Computation* **141**(1), 66–83.

Maron, O.: 1998, 'Learning from Ambiguity'. Ph.D. thesis, Dept. of Electrical Engineering and Computer Science, M.I.T.

Maron, O. and T. Lozano-Pérez: 1998, 'A framework for multiple-instance learning'. In: *Advances in Neural Information Processing Systsms 10*.

Maron, O. and A. L. Ratan: 1998, 'Multiple-instance learning for natural scene classification'. In: *Proc. 15th International Conf. on Machine Learning*. pp. 341–349.

Wang, J. and J. D. Zucker: 2000, 'Solving the Multiple-Instance Problem: A Lazy Learning Approach'. In: *Proc. 17th International Conf. on Machine Learning*. 1119–1125.

Warmuth, M. K. and A. K. Jagota: 1997, 'Continuous and discrete-time nonlinear gradient descent: relative loss bounds and convergence'. In: *Proc. Fifth International Symposium on Artificial Intelligence and Mathematics*.