LEARNABILITY OF MIN-MAX PATTERN CLASSIFIERS

Ping-Fai Yang and Petros Maragos

Division of Applied Sciences, Harvard University, Cambridge, MA 02138, USA

Abstract

This paper introduces the class of *thresholded min-max functions* and studies their learning under the probably approximately correct (PAC) model introduced by Valiant. These functions can be used as pattern classifiers of both real-valued and binary-valued feature vectors. They are a lattice-theoretic generalization of Boolean functions and are also related to three-layer perceptrons and morphological signal operators. Several subclasses of the thresholded min-max functions are shown to be learnable under the PAC model.

1 Introduction

In the field of machine learning there have been many theoretical advancements on distribution-free learning of Boolean functions. This learning framework is also known as the *probably approximately correct (PAC)* model, pioneered by Valiant (1984) and further developed by him and other researchers. There is already a wealth of literature about the PAC learning model; examples include Valiant (1984,1985), Blumer et al. (1987,1989), Haussler (1990), Kearns, Li, Pitt & Valiant (1987), Kearns (1990), Rivest (1990), and Schapire (1991). Most of the results in PAC learning deal with Boolean functions. If such functions are used as (Boolean) pattern classifiers, then the input features must be binary-valued. Although this may be sufficient for classifying high-level predicate-like features, most of the pattern recognition applications, for example in computer speech and object recognition, involve real-valued feature vectors. For example, Figure 1 shows morphological size histograms of binary character images, which we have experimentally found to be promising real-valued features for character recognition; the values of the normalized size histograms are real numbers in $[0, 1]$.

In this paper, we present a class of classifiers, called *thresholded min-max functions*, which can accept as inputs both real-valued and binary-valued feature vectors. Each input variable to these functions is in the range $[0, 1]$, in contrast to $\{0, 1\}$ for the Boolean classifiers. Moreover, these thresholded min-max functions are natural generalizations of the Boolean functions, because they are based on MIN/MAX operations which are the lattice-theoretic counterparts of Boolean AND/OR operations on real numbers. Although there exist many types of classifiers for real-valued data, the class of thresholded min-max functions has the appealing property that many of its large subclasses are PAC learnable (as will be shown later).

Another motivation for working with the thresholded min-max functions is their close relation to a large class of nonlinear signal/image operators known as morphological filters, which are defined via min-max operations on their inputs. As discussed in Serra (1982) and Maragos & Schafer (1990), these min-max morphological operators can be applied to a broad variety of feature extraction and shape analysis/detection tasks in images or arbitrary geometrical objects. Hence, learning of the thresholded min-max functions provides an ability for automated training of the above feature extraction and shape analysis/detection signal operators.

This paper is organized as follows: In Section 2, we define the min-max functions and their thresholded counterparts. A discussion of their relations with Boolean classifiers and three-layer perceptrons is included. We also investigate aspects of their representation power using techniques from mathematical morphology. Section 3 provides a brief summary of prior results from PAC learning used in later sections for proving learnability of subclasses of thresholded min-max functions. This is followed by three sections that contain the major learnability results of this paper. Section 4 discusses the learnability of a subclass

 0-8194-0743-7/91/$4.00

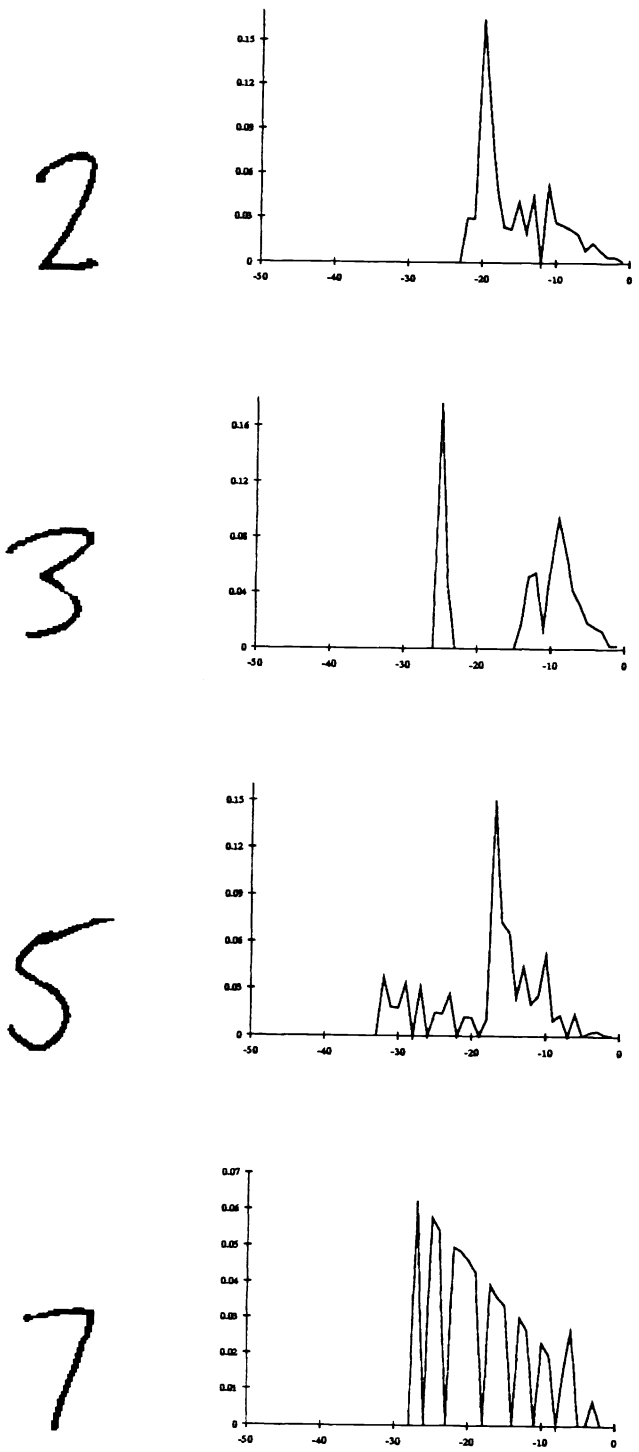

Figure 1: Morphological size histograms of hand-written characters using multiscale closings by a triangle ($\triangleright$) structuring element. (The horizontal axis shows the sizes of the structuring element. The vertical axis shows the normalized areas of differences among consecutive closings.)

called thresholded monotone minimum functions. Section 5 presents the results for the thresholded monotone maximum functions. Finally, Section 6 contains learnability results about the class of thresholded k-min-max functions, which are generalizations of the Boolean k-DNF functions.

2 Min-Max Functions

Here we start from general definitions about min-max functions and then use tools from mathematical morphology to explore some of their properties and relations to Boolean functions and perceptrons.

2.1 Definitions

A Boolean function $B(\vec{b})$, $\vec{b} = (b_1, ..., b_d) \in \{0,1\}^d$, in disjunctive normal form (DNF) is a finite disjunction (i.e, Boolean OR) of *terms*. A term is a conjunction (i.e., Boolean AND) of *literals*. A literal is either a Boolean variable $b_i \in \{0,1\}$ or its complement $\overline{b_i}$. To generate a min-max function from a DNF Boolean function, we replace the uncomplemented Boolean inputs b_i with real-valued variables $x_i \in [0,1]$, complemented variables $\overline{b_i}$ with *real complements* $x_i' \triangleq 1 - x_i$, and the Boolean AND/OR with MIN/MAX (denoted by $\wedge/\vee$).[1]

Formally, let $\vec{x} = (x_1, x_2, ..., x_d)$ be a real-valued vector in the d-dimensional unit cube $[0,1]^d$. We define a *min-max* function $f : [0,1]^d \to [0,1]$ with input $\vec{x}$ as the function

$$f(x_1, x_2, ..., x_d) = \bigvee_n \bigwedge_{j \in I_n} \ell_j \quad , \qquad \ell_j \in \{x_j, 1 - x_j\} \tag{1}$$

where an argument ℓ_j is called a *literal*, equal either to a variable x_j or its complement x_j'. Each minimum function $\bigwedge_{j \in I_n} \ell_j$ is a called a *min term*. Each I_n denotes the set of coordinates of the input vector $\vec{x}$ that appear in the argument of the n-th min term. The *size of a min term* is the number of literals in the minimum function. The maximum $\bigvee_n$ has a finite number of terms. Thus, a min-max function is a finite maximum of min terms. Note that the restriction of a min-max function on the finite discrete space $\{0,1\}^d$ is a Boolean function.

A Boolean function B is called *monotone* (or positive) if $B(\vec{a}) \leq B(\vec{b})$ whenever $\vec{a} \leq \vec{b}$, where $\vec{a} \leq \vec{b}$ means $a_i \leq b_i$ for all i. Gilbert (1954) showed that B is monotone if and only if all its variables appear uncomplemented. Similarly we call a function $f : [0,1]^d \to [0,1]$ *monotone* if

$$\vec{x} \leq \vec{y} \Longrightarrow f(\vec{x}) \leq f(\vec{y}) \quad , \quad \forall \vec{x}, \vec{y} \tag{2}$$

It can be shown that a min-max function is monotone if and only if it admits an expression that does not contain any complemented variables.

To use a min-max function f as a classifier performing binary decisions we need to threshold f at some arbitrary value $\theta \in [0,1]$. This creates a *thresholded min-max function* $f_\theta : [0,1]^d \to \{0,1\}$ defined by

$$f_\theta(\vec{x}) = P[f(\vec{x}) \geq \theta] \tag{3}$$

where $P(\cdot)$ is the predicate function, equal to 1 if the inequality inside is true and equal to 0 otherwise. An example of a thresholded min-max function is $P\{((x_1 \wedge x_4) \vee (x_2 \wedge (1 - x_3) \wedge x_5)) \geq 0.6\}$. It is generalized from the Boolean function $(b_1 \cdot b_4 + b_2 \cdot \overline{b_3} \cdot b_5)$. The min term $(x_1 \wedge x_4)$ is size two while $(x_2 \wedge (1 - x_3) \wedge x_5)$ is size three. In the second min term, the variable x_3 is complemented. Note that there are an infinite number

[1] In this paper, Boolean AND is denoted by the product symbol '$\cdot$', which may be left out occasionally. The Boolean OR is denoted by '+'. The symbols $\vee$ and $\wedge$ are defined as $\bigvee_{n \in I} x_n = \max_n\{x_n\}$ and $\bigwedge_{n \in I} x_n = \min_n\{x_n\}$ if the index set I is finite; if I is infinite, then the max and min should be replaced by supremum and infimum, respectively.

of thresholded min-max functions corresponding to a Boolean function. This is due to the freedom in the choice of the threshold value θ, which in our work will be a free parameter to be learned. A thresholded min-max function is monotone if its corresponding min-max function is monotone.

Exchanging the roles of AND and OR in a DNF Boolean function transforms the latter into a conjunctive normal form (CNF). Similarly, exchanging the roles of MIN and MAX in a (thresholded) min-max function will yield a (thresholded) *max-min function*, i.e., a (thresholded) minimum of maxima. Due to the straightforward duality relationships between these two latter function classes, in this paper we focus on functions in the min-max form.

An input vector is classified as a *positive* or *negative* instance of a thresholded min-max function $c(\vec{x})$ accordingly to whether the output of $c(\vec{x})$ is 1 or 0 respectively. We shall also refer to this classification process as the *labeling* of the input vector $\vec{x}$ by $c(\vec{x})$, and call general 0-1-valued functions *classifier* functions to emphasize the possibility of their use as pattern classifiers. In the setting of our learning model, classifier functions are also referred to as *concept functions*, or simply *concepts*. We shall use the later name more often in the rest of the paper. A collection of concepts is called a *concept class*, which is usually denoted by $\mathcal{C}$. The set of all thresholded min-max functions with d variables is denoted by $\mathcal{C}^d_{min-max}$.

In this paper, we shall demonstrate the learning (in the PAC model) of the following three subclasses of $\mathcal{C}^d_{min-max}$:

Thresholded Monotone Minimum Functions: A thresholded monotone minimum function has the general form: $P\left(\bigwedge_{i\in I} x_i \geq \theta\right)$, where I is the set of coordinate indices of the input vector. This class of functions is denoted by the symbol $\mathcal{C}^d_{min}$.

Thresholded Monotone Maximum Functions: These functions are dual forms of the thresholded minimum functions. The general form is $P\left(\bigvee_{i\in I} x_i \geq \theta\right)$, where I is again the set of coordinate indices of the input vector. The collection of all thresholded monotone maximum functions is denoted by the symbol $\mathcal{C}^d_{max}$.

Thresholded k-Min-Max Functions: They are thresholded min-max functions with the restriction on the size of each min term to be $\leq k$. The class is denoted by the symbol $\mathcal{C}^d_{k-min-max}$.

These three classes of functions are generalizations of the Boolean positive term, positive clause, and k-DNF functions respectively.

2.2 Morphological Representations and Relations to Boolean Functions

Here we establish some relationships between (thresholded) min-max functions and Boolean functions using concepts from morphological filtering as discussed in Maragos & Schafer (1987). First note the following three useful properties of the predicate function $P()$, which can be easily proven. The minimum and maximum functions obey a *threshold homomorphism* property:

$$P(x \wedge y \geq \theta) = P(x \geq \theta) \wedge P(y \geq \theta) = P(x \geq \theta) \cdot P(y \geq \theta), \tag{4}$$

$$P(x \vee y \geq \theta) = P(x \geq \theta) \vee P(y \geq \theta) = P(x \geq \theta) + P(y \geq \theta). \tag{5}$$

In addition, we have the *threshold reconstruction* property:

$$x = \bigvee_{\theta\in[0,1]} P(x \geq \theta) \quad , \quad \forall x \in [0,1] \tag{6}$$

From (1), (3) and the above properties it follows that

$$f_\theta(x_1, x_2, ..., x_d) = \bigvee_n \bigwedge_{j\in I_n} P(\ell_j \geq \theta) \quad , \qquad \ell_j \in \{x_j, 1 - x_j\} \tag{7}$$

Thus, a thresholded min-max function is equal to the disjunction of terms containing Boolean variables formed by thresholding the input coordinates x_i or their complements. Turning to the thresholding of a complemented variable,

$$P(x' \geq \theta) = P(1 - x \geq \theta) = P(x \leq 1 - \theta). \tag{8}$$

The thresholding of $x' = 1 - x$ is not equal to the Boolean complement $\overline{P(x \geq \theta)} = P(x < \theta)$ in general. However, this particular definition of x' remains a reasonable choice because it is identical to the Boolean complement if x takes on only 0,1 values. It also preserves the range of the variable; i.e., $x \in [0,1] \Longrightarrow x' \in [0,1]$.

Next we present a result that indicates the representation power of monotone min-max functions. Some definitions are needed first: Consider arbitrary functions $f : [0,1]^d \rightarrow [0,1]$ that are "consistent" generalizations of Boolean functions, i.e., their value is binary whenever the input vector is binary; formally

$$f(\vec{x}_\theta) \in \{0,1\} \quad , \qquad \forall \theta \in [0,1] \tag{9}$$

where $\vec{x}_\theta \in \{0,1\}^d$ is a thresholded (and hence binary) input vector:

$$\vec{x}_\theta = (x_{1,\theta}, ..., x_{d,\theta}) \stackrel{\triangle}{=} (P(x_1 \geq \theta), ..., P(x_d \geq \theta)) \tag{10}$$

We also say that f *commutes with thresholding* if

$$f_\theta(\vec{x}) = f(\vec{x}_\theta) \quad , \qquad \forall \vec{x}, \theta \tag{11}$$

Commuting with thresholding is an important property since it implies that the Boolean function $f(\vec{x}_\theta)$ obtained from f by thresholding the input vector at any θ (and hence restricting f on the finite discrete space $\{0,1\}^d$) gives identical values with thresholding the output of f at θ. It will be shown in the following theorem that, if f commutes with thresholding, then it is monotone. Further, this theorem establishes that functions that commute with thresholding can be represented by monotone min-max functions.

Theorem 1 *Let $f : [0,1]^d \rightarrow [0,1]$ be a function that obeys property (9). Then f commutes with thresholding if and only if it is monotone min-max function, or equivalently if and only if it is a min-max function without any complemented variables.*

Proof

Let f commute with thresholding. Consider binary vectors $\vec{a} \leq \vec{b} \in \{0,1\}^d$. We can always find some real vector $\vec{x}$ such that $\vec{a} = \vec{x}_{\theta_1}$ and $\vec{b} = \vec{x}_{\theta_2}$ with $\theta_1 \geq \theta_2$. Then, if B is the Boolean function corresponding to f, since $f(\vec{x}_\theta) = B(\vec{x}_\theta)$ for each θ, we have

$$B(\vec{a}) = f(\vec{x}_{\theta_1}) = f_{\theta_1}(\vec{x}) \leq f_{\theta_2}(\vec{x}) = f(\vec{x}_{\theta_2}) = B(\vec{b}) \tag{12}$$

Hence, B is monotone. Since the monotone B admits a DNF expression as a unique irreducible OR of AND terms, it follows from (6) that:

$$\begin{aligned} f(\vec{x}) &= \bigvee_\theta f_\theta(\vec{x}) = \bigvee_\theta f(\vec{x}_\theta) \\ &= \bigvee_\theta \bigvee_n \bigwedge_{j \in I_n} P(x_j \geq \theta) \\ &= \bigvee_n \bigvee_\theta P\left(\bigwedge_{j \in I_n} x_j \geq \theta \right) = \bigvee_n \bigwedge_{j \in I_n} x_j \end{aligned}$$

Hence, f is equal to a min-max function. Further f is monotone because its min-max representation contains no complements. Conversely, let f be a monotone min-max function. Then

$$f(\vec{x}) = \bigvee_n \bigwedge_{j \in I_n} x_j \Longrightarrow f_\theta(\vec{x}) = \bigvee_n \bigwedge_{j \in I_n} P(x_j \geq \theta) = f(\vec{x}_\theta) \tag{13}$$

Hence f commutes with thresholding, which completes the proof. □

The essence of the above theorem is that any monotone real-input real-output function that yields a binary output whenever the input vector is binary and commutes with thresholding can be represented as a min-max function (with no complements). Conversely, the class of thresholded monotone min-max functions is almost isomorphic to Boolean functions, except for the generally unknown parameter θ which is to be learned.

2.3 Relations to Perceptrons

Another class of classifiers that is related to thresholded min-max functions is the three-layer perceptrons. The link is provided by the thresholded homomorphism properties (4) and (5). We demonstrate this using an example. Consider the thresholded min-max function

$$P(x_1 \vee (x_2' \wedge x_4) \geq \theta). \tag{14}$$

Applying first (5) and then (4), we derive an equivalent function

$$P(x_1 \geq \theta) + (P(x_2' \geq \theta) \cdot P(x_3 \geq \theta)).$$

Finally, we use (8) to arrive at the desired form

$$P(x_1 \geq \theta) + (P(x_2 \leq 1 - \theta) \cdot P(x_3 \geq \theta)). \tag{15}$$

Observe that each predicate function in the above expression is the thresholding of a single variable, which can be implemented using a single layer perceptron. The Boolean conjunctions and disjunctions can also be implemented using single layer perceptrons. Therefore the original thresholded min-max function in (14) can be implemented using a three-layer perceptron. It is easy to see that any thresholded min-max function can be implemented using a three layer perceptron because the thresholded min-max function is formed by the composition of several minima and a maximum. It is not true that any three layer perceptron can be expressed as a thresholded min-max function. To see the reason, one simply has to look at (15). The first layer of perceptrons have the form $P(\ell_i \geq \theta)$. Their decision regions are parallel to the coordinate axes. Since the second and third layers are Boolean AND and OR respectively, the positive region of the cascaded structure should have boundaries parallel to the axes. For a general three-layer perceptron, this condition is not necessarily true. Therefore, $\mathcal{C}^d_{min-max}$ is a *subclass* of the class of general three layer perceptrons.

3 Background on PAC Learning

In this section, we shall present some well known results from PAC learning theory which we employ in the rest of the paper. For the definition of the PAC learning model, see, for example, Valiant (1984), Kearns (1990), and Rivest (1990).

In the PAC model, we want to construct a learning algorithm that returns concept functions with small error rates. One possibility is to estimate the error rate from the sequence of training data. The estimate is called the *empirical error rate*. Since the *target concept* $t(\vec{x})$ belongs to the concept class $\mathcal{C}$, the empirical

error rate of the target concept is equal to zero. Accordingly, one strategy for the learning algorithm is to return any concept (the *hypothesis* $h(\vec{x})$) that has a zero empirical error rate. This type of algorithm is called *consistent* algorithm. In case the algorithm runs in polynomial time (in the number of training data), it has a special name: *poly-hy-fi* (for polynomial hypothesis finder). The following proposition gives the minimum number of training data a consistent algorithm requires.

Proposition 1 (Blumer et al. (1989)) *Let $\mathcal{C}$ be a nontrivial, well-behaved*[2] *concept class. If the Vapnik-Chervonenkis dimension of $\mathcal{C}$ is $VC(\mathcal{C}) < \infty$, then for $0 < \epsilon, \delta < 1$, and training sample size at least*

$$\max(\frac{4}{\epsilon}\log_2\frac{2}{\delta}, \frac{8\,VC(\mathcal{C})}{\epsilon}\log_2\frac{13}{\epsilon}), \tag{16}$$

then with probability $\geq 1-\delta$, any consistent algorithm will return a hypothesis $h(\vec{x})$ with true error rate $\xi(h) \leq \epsilon$.

The Vapnik-Chervonenkis (VC) dimension of a concept class $\mathcal{C}$ is the size of the largest finite subset of the domain X which is labeled in all possible ways using concepts in $\mathcal{C}$. A formal definition of this parameter is discussed in the same paper by Blumer et.al. From (16), the number of training data required is polynomial in d, $\frac{1}{\epsilon}$, and $\frac{1}{\delta}$ if the VC dimension is polynomial in the dimension d. Hence, using this technique, the proof of learnability of $\mathcal{C}$ can be divided into two steps:

1. show that the VC dimension of $\mathcal{C}$ is polynomial in d;
2. find a poly-hy-fi for $\mathcal{C}$.

We follow this approach in this paper to show the learnability of the classes $\mathcal{C}^d_{min}$, $\mathcal{C}^d_{max}$, and $\mathcal{C}^d_{k-min-max}$.

4 Learnability of Thresholded Monotone Minimum Functions

The general form of a thresholded monotone minimum function is:

$$P\left(\bigwedge_{i\in I} x_i \geq \theta\right) \tag{17}$$

where I denotes the set of coordinate indices, $I \subseteq \{1,\ldots,d\}$. Using the threshold homomorphism property (4), this equation can be transformed to a Boolean product

$$\prod_{i\in I} P(x_i \geq \theta)\,. \tag{18}$$

Each of the expression $P(x_i \geq \theta)$ on the right side of Equation (18) is equal to 1 only in the "positive" half space defined by the axis parallel hyperplane $x_i = \theta$. Since (18) is a conjunction of the predicates, it is equal to 1 only if the input vector $\vec{x}$ satisfies all the inequalities that are present. In other words, the positive region of each predicate $P(x_i \geq \theta)$ is the part of the d-dimensional hypercube whose i-th coordinate is $\geq \theta$. Therefore, the positive region of (17) is the intersection of the positive regions of its constituent predicates $P(x_i \geq \theta), i \in I$.

To illustrate the above observation, we present an example with $d = 2$. Figure 2 is provided for graphical illustration. In two dimensions, thresholded monotone minimum functions take only three general forms:

[2] A concept class is trivial if it has only one concept or it has two disjoint concepts such that $c_1 \cup c_2 = X$. The well-behavedness condition is some measurablity conditions on the functions, it is detailed in Appendix A of Blumer et al. (1989).

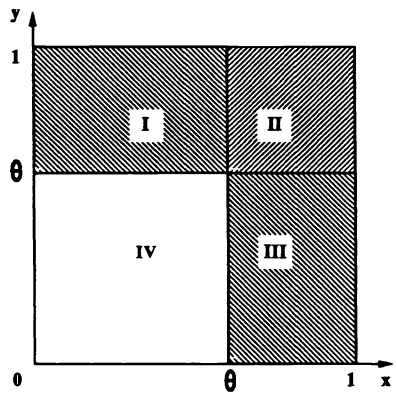

Figure 2: Decision regions of various two dimensional thresholded monotone minimum functions. (The meaning of the labelings are given in Section 4.)

1. $P(x \geq \theta)$
2. $P(y \geq \theta)$
3. $P(x \wedge y \geq \theta)$

The first two forms are functions of only one coordinate. Referring to Figure 2, the positive region of the first function is the union of regions II and III, which is an axes-parallel rectangle with one vertex at (1,1). As for the second one, the positive region is the union of regions I and II, another axes-parallel rectangle. Finally, the positive region of the last one is the intersection of that of the first two, which is region II in the figure. Its positive region is an axes-parallel square. It is easy to see that in higher dimensions, the positive regions become axes-parallel hyper-rectangles, with one vertex at $(1, \ldots, 1)$.

4.1 VC-dimension of the Thresholded Monotone Minimum Functions

In this section, we show that $\mathrm{VC}(\mathcal{C}^d_{min}) = d$ by providing upper and lower bound of $VC(\mathcal{C}^d_{min})$.

We start the first part of the proof with an example. Figure 3 shows three points I, J, and K inside the unit square. Any attempt to shatter these three points using two dimensional thresholded monotone minimum functions will fail. For example, we cannot label points I and K positive while J negative (this is the labeling shown in the figure). This is so because the positive region of two dimensional thresholded monotone minimum functions are axes-parallel squares or rectangles with one vertex at (1,1). This property can be generalized to d dimensions and is formalized in Lemma 1. The proof shows that for any $d+1$ points inside the d-dimensional unit hypercube, one of them must be inside the bounding hyper-rectangle formed by the others and the vertex $(1, \ldots, 1)$. Theorem 2 uses this fact to show that any set of $m \geq d+1$ is not shattered.

We now proceed with the lemma. Throughout this paper, we use the symbol x_k to denote the k-th coordinate of a general vector $\vec{x}$.

Lemma 1 *Let S be a set of points in $[0,1]^d$.*

$$\forall \vec{y} \in S, \exists c \in \mathcal{C}^d_{min} \text{ such that } \{\vec{y}\} \text{ is labeled negative and } S \setminus \{\vec{y}\} \text{ is labeled positive}$$

if and only if

$$\forall \vec{y} \in S, \exists \text{ a coordinate index } k \text{ such that } y_k < \bigwedge_{\vec{w} \in S \setminus \{\vec{y}\}} w_k.$$

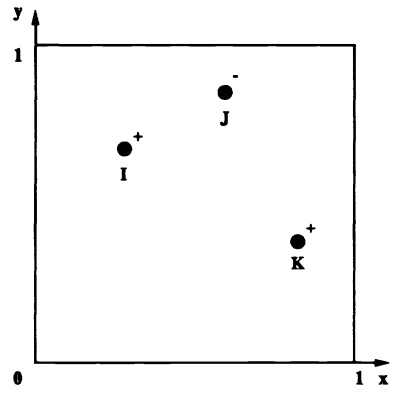

Figure 3: The labeling shown cannot be achieved using thresholded monotone minimum functions. Hence the set $\{I, J, K\}$ cannot be shattered.

Proof
$\Leftarrow$: Assuming the second condition holds, we only have to demonstrate $|S|$ thresholded minimum functions that perform the $|S|$ partitionings as specified. For each point $\vec{y} \in S$, one possibility is:

$$P\left(\bigwedge_{i\in\{k\}} x_i \geq \theta\right)$$

where $\theta = \frac{1}{2}\left\{\bigwedge_{\vec{w}\in S\setminus\{\vec{y}\}}(w_k) + y_k\right\}$.

$\Rightarrow$: Consider the point $\vec{y}$. If $\vec{y}$ is labeled negative by some monotone thresholded minimum function

$$P\left(\bigwedge_{i\in I} x_i \geq \theta\right) \tag{19}$$

while $S \setminus \{\vec{y}\}$ is labeled positive, we must have the following inequalities:

$$\bigwedge_{i\in I} y_i < \theta \leq \bigwedge_{\vec{w}\in S\setminus\{\vec{y}\}, i\in I} w_i$$

The last one holds due to the fact that the function defined in (19) labels all the points $\vec{w} \in S \setminus \{\vec{y}\}$ positive. Suppose $y_k = \bigwedge_{i\in I} y_i$ then,

$$y_k = \bigwedge_{i\in I} y_i < \theta \leq \bigwedge_{\vec{w}\in S\setminus\{\vec{x}\}, i\in I} w_i \leq \bigwedge_{\vec{w}\in S\setminus\{\vec{x}\}} w_k$$

which proves the required condition. □

Next, we present the proposition which gives the upper bound on $\mathrm{VC}(\mathcal{C}^d_{min})$.

Theorem 2 *No set of $m \geq d+1$ points in $[0,1]^d$ can be shattered by $\mathcal{C}^d_{min}$, i.e. $VC(\mathcal{C}^d_{min}) \leq d$.*

Proof
Let S be a set of $m \geq d+1$ points in the d dimensional unit hypercube. Suppose the set can be shattered, there must be m concept functions, each labeling only one of the m points negative.

Since $m \geq d+1$, and each of the point in S must satisfy the condition stated in Lemma 1, there must be two distinct vectors $\vec{x}, \vec{y} \in S$ and a coordinate axis k such that

$$x_k < \bigwedge_{\vec{w} \in S \setminus \{\vec{x}\}} w_k \tag{20}$$

$$\leq y_k \tag{21}$$

Equation (20) follows from Lemma 1 while (21) is a consequence of $\vec{y} \in S \setminus \{\vec{x}\}$. By exchanging the role of $\vec{x}$ and $\vec{y}$ in the above derivation, we get both $x_k < y_k$ and $y_k < x_k$, a contradiction. □

Theorem 2 provides the upper bound for $VC(\mathcal{C}^d_{min})$. To conclude the proof, the lower bound is provided by the following theorem.

Theorem 3 *There exists a set of d points in* $[0,1]^d$ *that is shattered by* $\mathcal{C}^d_{min}$, *i.e.* $VC(\mathcal{C}^d_{min}) \geq d$.

Proof
This theorem is proved using an explicit construction of a set of d points in $[0,1]^d$ which is shattered by $\mathcal{C}^d_{min}$. One possibility is:

$$S = \{\vec{x}^1, \ldots, \vec{x}^d\}$$

$$x^i_k = \begin{cases} \frac{3}{4}, & \text{if } i \neq k, \\ \frac{1}{4}, & \text{if } i = k, \end{cases}$$

where x^i_k denotes the k-th coordinate of the i-th vector in S. It is easy to see that this set is shattered by $\mathcal{C}^d_{min}$. See Yang & Maragos (1991) for details of the proof.

□

It follows immediately from Theorems 2 and 3 that the VC dimension of $\mathcal{C}^d_{min}$ is d.

We conclude this section with a numerical example. Suppose we want to learn a thresholded monotone minimum function with maximum error rate $\epsilon = 0.1$ and the parameter $\delta = 0.01$. Take $d = 50$. Using the expression for the training data bound (16), we get

$$p(\frac{1}{\epsilon}, \frac{1}{\delta}, d) \geq 28,090$$

This amounts to about 560 training data per input dimension.

4.2 Poly-Hy-Fi for $\mathcal{C}^d_{min}$

The notation symbols used in this section are listed below:

$t^j = (\vec{x}^j, l^j) =$ a training sample, with $\vec{x}^j$ being the input vector and l^j the label,
$t^+ = (\vec{x}^+, 1) =$ a general positive training data,
$t^- = (\vec{x}^-, 0) =$ a general negative training data,
$x^j_c =$ the c-th coordinate of the vector $\vec{x}^j$,
$x^+_c =$ the c-th coordinate of the vector $\vec{x}^+$,
$x^-_c =$ the c-th coordinate of the vector $\vec{x}^-$,
$n =$ total number of training data,
$n_+ =$ total number of positive training data,
$n_- =$ total number of negative training data.

Under the PAC learning model, the labels are assumed to be generated by a target function in the concept class, i.e. $l^j = t(\vec{x}^j)$, $t() \in \mathcal{C}^d_{min}$.

In the first step of the algorithm, d threshold estimates $\theta_k, 1 \leq k \leq d$ are produced. The k-th estimate is calculated by taking the minimum of the k-th coordinate of all the $\vec{x}^+$. The intuition behind this comes from observing the shape of the positive region of a thresholded monotone minimum function, which is a hyper-rectangle with one vertex at the point $(1, \ldots, 1)$. Under the assumption of the PAC model, the training data is consistent with a thresholded monotone minimum function. If we form an axes-parallel hyper-rectangle that bounds the positive training data and $(1, \ldots, 1)$, the coordinates of the faces should give good estimates of the threshold. These are found by the minimum operation.

In the second step, the algorithm uses these d estimates to generate d thresholded monotone minimum functions $h_k(\vec{x}) = P\left(\bigwedge_{i \in I_k} x_i \geq \theta_k\right)$. The coordinate list I_k is initialized to be $\{1, \ldots, d\}$ for all values of k. Denote $\mathcal{I} = \{I_k : 1 \leq k \leq d\}$. Then, the algorithm eliminates from I_k all coordinates c such that $\exists \vec{x}^+, x^+_c < \theta_k$. The rationale for this step comes from the threshold homomorphism property (Equation (4))

$$P\left(\bigwedge_{i \in I_k} x_i \geq \theta_k\right) = \prod_{i \in I_k} P(x_i \geq \theta_k).$$

If there is an index $c \in I_k$ such that $x^+_c < \theta_k$, the positive data will be labeled negative. Therefore, the variable c should be removed from I_k. If an I_k becomes empty after this step, the corresponding concept h_k is removed from further consideration.

In the final step, $h_k()$ is eliminated if it is inconsistent with any negative training data. Therefore, the remaining thresholded monotone minimum functions are consistent with all the training data. One of them is returned as the hypothesis.

Before we present the correctness proof, we shall show that the algorithm is indeed polynomial time. We assume that comparison requires unit time. The first step takes at most nd comparisons (there are at most n positive training data, each one requires d comparisons). The next one requires at most n_+d^2 comparisons (there are at most d indices lists I_k, and each generates at most d comparisons for each positive training data.) The final one also takes n_-d^2 comparisons with the same reasoning. Total number of comparisons is

$$n_+d^2 + n_-d^2 + nd = nd^2 + nd = O(nd^2),$$

which is polynomial in the number of training data and the dimension d. Since n is polynomial in d, $\frac{1}{\epsilon}$, and $\frac{1}{\delta}$, this consistent hypothesis runs in time polynomial in these variables too. Therefore, the PAC condition is met.

Theorem 4 asserts the correctness of the algorithm.

Theorem 4 *Assume that $t() \in \mathcal{C}^d_{min}$. The algorithm presented in this section always returns a thresholded monotone minimum function that is consistent with all the training data.*

Proof

It is obvious that the hypothesis returned by this algorithm is consistent with all the training data. We only have to show that after the last step, $\mathcal{I}$ is not empty.

Suppose the target function has the functional form

$$t(\vec{x}) = P\left(\bigwedge_{i \in I} x_i \geq \theta\right).$$

In the first step, one of the estimates θ_k must be equal to

$$\hat{\theta} = \bigwedge_{\vec{x}^+} \bigwedge_{i \in I} x^+_i = \bigwedge_{i \in I} \left(\bigwedge_{\vec{x}^+} x^+_i\right) = \bigwedge_{i \in I} \theta_i$$

since θ_k is calculated for *all* coordinate indices. Using this estimate, the index list generated after the second step ($\hat{I}$) is not empty, and will not be eliminated from $\mathcal{I}$. Moreover, the variables present in I are not removed, i.e. $I \subseteq \hat{I}$. This fact is easy to show using the definition of $\hat{\theta}$. Finally, $\hat{I}$ is not removed from $\mathcal{I}$ in the last step. To show this, observe that

$$\hat{\theta} \geq \theta$$

and hence for any negative training data $\vec{x}^-$ there is a coordinate $c \in I$ such that

$$x_c^- < \theta \leq \hat{\theta}$$

because the $\vec{x}^-$ must be labeled negative by the target concept. Using the fact that $I \subseteq \hat{I}$, it is obvious that the hypothesis corresponding to $\hat{\theta}, \hat{I}$ must be labeled negative for any negative training data. □

It is important to note that we have not assumed the independence of the input variables $\{x_1, \ldots, x_d\}$. They can in fact be functions of each other. The only assumption made is that the target function takes the form of a thresholded monotone minimum function in these variables. This observation is important especially in Section 6, where the input variables for the poly-hy-fi are functionally related.

5 Learnability of Thresholded Monotone Maximum Functions

All the results in the previous section can be transcribed to apply to $\mathcal{C}_{max}^d$ by using the duality relation between thresholded monotone maximum and thresholded monotone minimun functions:

$$\overline{P\left(\bigwedge_{i\in I} x_i < \theta\right)} = \prod_{i\in I} \overline{P(x_i < \theta)} = \prod_{i\in I} \overline{\overline{P(x_i \geq \theta)}} = \left(\bigvee_{i\in I} x_i \geq \theta\right). \tag{22}$$

The function on the left hand side resembles a complemented thresholded monotone minimum function, with the exception of the definition of the thresholding ($< \theta$ instead of $\geq \theta$). From (22), this function is equal to a thresholded monotone maximum function. To convert the results for $\mathcal{C}_{min}^d$ to apply to $\mathcal{C}_{max}^d$, we use the following substitutions:

- Replace minimum by maximum and vice versa;
- Replace $x_i \geq (>)\theta$ by $x_i < (\leq)\theta$ and vice versa;
- Replace "positive" by "negative" and vice versa.

These substitutions should be applied carefully because some of the statements concern a *set* of thresholded monotone minimum functions. In the following, we provide a summary of the theoretical results for $\mathcal{C}_{max}^d$. The proofs are omitted and refer the reader to Yang & Maragos (1991) for details.

Lemma 2 *Let S be a set of points in $[0,1]^d$.*

$$\forall \vec{y} \in S, \exists c \in \mathcal{C}_{max}^d \text{ such that } \{\vec{y}\} \text{ is labeled positive and } S \setminus \{\vec{y}\} \text{ is labeled negative}$$

if and only if

$$\forall \vec{y} \in S, \exists \text{ a coordinate index } k \text{ such that } y_k \geq \bigvee_{\vec{w}\in S\setminus\{\vec{y}\}} w_k.$$

Theorem 5 *No set of $n \geq d+1$ points in $[0,1]^d$ can be shattered by $\mathcal{C}_{max}^d$, i.e. $VC(\mathcal{C}_{max}^d) \leq d$.*

Theorem 6 *There exists a set of d elements which is shattered by $\mathcal{C}_{max}^d$, i.e. $VC(\mathcal{C}_{max}^d) \geq d$.*

Combining the two theorems, we immediately find that $VC(\mathcal{C}^d_{max}) = d$.

Turning to the poly-hy-fi, we can use the duality result to transform the poly-hy-fi for $\mathcal{C}^d_{min}$ to one for $\mathcal{C}^d_{max}$. In the general discussion in Section 4.2, swap the words "positive"/"negative" and "minimum"/"maximum", and replace "conjunction" by "disjunction" and "$x^+_c < \theta_k$" with "$x^+_c \geq \theta_k$". The basic operations in this algorithm are the same as its $\mathcal{C}^d_{min}$ counterpart, so the computational complexity remains polynomial ($O(nd^2)$). The correctness proof in Section 4.2 can also be adapted using duality.

6 Learnability of Thresholded k-Min-Max Functions

A general thresholded k-min-max function has the form $P(\bigvee_n T_n \geq \theta)$, with T_n denoting a min term of size at most k (i.e. a minimum function with at most k literals in its argument). This form is very suggestive of the connection between thresholded k-min-max function and thresholded monotone maximum function: the k-min-max function ($\bigvee_n T_n$) is a maximum of the uncomplemented min terms T_n. Using this observation, the evaluation of a thresholded k-min-max function can be broken up into two parts — the first step calculates the values of *all* min terms with size $\leq k$. This can be considered a *remapping* of the input variables $\vec{x}$ into the set of min terms $R = \{r_n = \bigwedge_{i \in I, |I| \leq k} \ell_i\}$. The min terms T_n will be elements of R. These are the dependent variables of the thresholded monotone maximum function that is evaluated in the second step: $P\left(\bigvee_{\{n|r_n=T_n\}} r_n \geq \theta\right)$. Therefore, any thresholded k-min-max function is equivalent to a thresholded monotone maximum function in a higher dimensional space. In other words, we establish a mapping between the class $\mathcal{C}^d_{k-min-max}$ and a class of thresholded maximum functions with a larger number of input variables ($\mathcal{C}^{|R|}_{max}$).

To illustrate the remapping idea, we shall use the following thresholded min-max function

$$P(x_1 \vee (x'_2 \wedge x_3) \geq \theta). \tag{23}$$

This function belongs to $\mathcal{C}^3_{2-min-max}$, i.e. the class of thresholded min-max functions with 3 input variables and at most 2 literals in each of the in term. Following the remapping scheme, we introduce a set of new variables r_n which are min terms of x_1, x_2, x_3 with at most 2 literals. The new variables are listed below:

$$\begin{array}{ccc}
r_1 = x_1, & r_2 = x_2, & r_3 = x_3, \\
r_4 = x'_1, & r_5 = x'_2, & r_6 = x'_3, \\
r_7 = x_1 \wedge x_2, & r_8 = x_1 \wedge x_3, & r_9 = x_2 \wedge x_3, \\
r_{10} = x'_1 \wedge x_2, & r_{11} = x_1 \wedge x'_2, & r_{12} = x'_1 \wedge x'_2, \\
r_{13} = x'_1 \wedge x_3, & r_{14} = x_1 \wedge x'_3, & r_{15} = x'_1 \wedge x'_3, \\
r_{16} = x'_2 \wedge x_3, & r_{17} = x_2 \wedge x'_3, & r_{18} = x'_2 \wedge x'_3, \\
r_{19} = x_1 \wedge x'_1, & r_{20} = x_2 \wedge x'_2, & r_{21} = x_3 \wedge x'_3.
\end{array}$$

Exact numbering of the new variables is irrelevant as long as all the possible min terms of size ≤ 2 are present. Note that the variables r_{19} through r_{21} are formed by taking the minimum of a variable with its complement. This is because the expression $P(x_i \wedge x'_i \geq \theta)$ is not always equal to 0. The second step in the process entails the introduction of a thresholded monotone maximum function that uses the r_n's as input. For the thresholded function in (23), the new function is $P\left(\bigvee_{i \in \{1,16\}} r_i \geq \theta\right)$. Other functions in $\mathcal{C}^3_{2-min-max}$ can be expressed as a thresholded monotone maximum function of $\vec{r}$. For example, $P(((x'_1 \wedge x_2) \vee (x'_1 \wedge x'_3)) \geq \theta)$ can be expressed as $P(r_{10} \vee r_{15} \geq \theta)$.

Denote the number of variables in the remapped vector $\vec{r}$ by $d' = |R|$. (In the example the dimension of the new vector is $d' = 18$.) The parameter d' is a function of d and k. By a simple combinatorial argument, one can easily show that the functional form is:

$$d' = \binom{2d}{1} + \ldots + \binom{2d}{k} \leq k(2d)^k \leq (2d)^{(k+1)} \tag{24}$$

where $\binom{p}{q} = \frac{p!}{q!(p-q)!}$ is the combination. The upper bound on d' is polynomial in the parameter d when k is fixed.

Using the remapping idea, the domain of the k-min-max functions X can be mapped to a subset of a d' dimensional space X'. Also, any set of points $S \subset X$ can be mapped to $S' \subset X'$. From this observation, we can easily prove the following theorem.

Theorem 7 $VC(\mathcal{C}^d_{k-min-max}) \leq VC(\mathcal{C}^{d'}_{max})$.

Proof
Consider a set $S \subset X$ that is shattered by $\mathcal{C}^d_{k-min-max}$. The size of S is VC($\mathcal{C}^d_{k-min-max}$). Using the remapping procedure, this set can be mapped to $S' \subset X'$ which has the same number of elements. Moreover, the output value of thresholded k-min-max function $c(\vec{x})$ is the same as that of the thresholded monotone maximum function $c'(\vec{r})$ after the remapping. Therefore, if the set S is shattered by a collection of concepts $\{c(\vec{x}) \in \mathcal{C}^d_{k-min-max}\}$, the set S' will also be shattered by the remapped functions $\{c'(\vec{r}) \in \mathcal{C}^{d'}_{max}\}$. Since the set S' is shattered by $\mathcal{C}^{d'}_{max}$, the VC dimension of this concept class is bounded by the inequality

$$VC(\mathcal{C}^{d'}_{max}) \geq |S'| = VC(\mathcal{C}^d_{k-min-max})$$

□

From Section 5, the VC dimension of $\mathcal{C}^d_{max}$ is d. Also, using the upper bound on d' in Equation (24), we get the following bound on VC($\mathcal{C}^d_{k-min-max}$).

Corollary 1 $VC(\mathcal{C}^d_{k-min-max}) \leq (2d)^{(k+1)}$.

Turning to the learning algorithm for $\mathcal{C}^d_{k-min-max}$, we found that any thresholded k-min-max function becomes a thresholded monotone maximum function in the new variables r_i. Moreover, there are only a polynomial number of remapped variables r_i. Therefore, the sequence of training data $(\vec{x}^j, l^j)$ can first be mapped into the new coordinates $(\vec{r}^j, l^j)$. The result is fed to the poly-hy-fi for $\mathcal{C}^d_{max}$. It will return a hypothesis in the remapped variable, which can be converted back to a thresholded k-min-max function easily by replacing the coordinates r_i by the corresponding minimum function on $\vec{x}$. Assuming unit time for computing a comparison, the amount of time required by the remapping step is at most $ndd' \leq nd(2d)^{(k+1)}$. Therefore the total time required by the algorithm is

$$ndd' + (nd' + nd'^2) = (d+1)nd' + nd'^2 \leq n\left\{(d+1)(2d)^{k+1} + (2d)^{2(k+1)}\right\} = O(n(2d)^{2(k+1)}).$$

7 Conclusion

In this paper, we have shown the learnability of three subclasses of thresholded min-max functions under the PAC model. In addition to finding polynomial bounds on the number of required training samples, we have also devised learning algorithms that run in polynomial time.

One of our long-term goals in this research work is to apply the results in this paper to practical applications such as recognition of hand-written characters. For example, Figure 1 shows some features (i.e., the size histogram vectors), which, as our preliminary experiments indicate, appear to be promising for character recognition and can be used as inputs to the min-max classifiers.

Despite its intellectual clarity, a rather restricting assumption of the standard PAC model is the presence of a target concept that belongs to the concept class. Therefore, it would be interesting to investigate the learnability of thresholded min-max functions under less restrictive models of learning, such as the "probabilistic concept" model proposed by Schapire (1991).

Acknowledgement

This research was supported by the NSF under Grant MIPS-86-58150 with matching funds from Xerox and DEC, and in part by the ARO under Grant DAALO3-86-K-0171 to the Brown-Harvard-MIT Center for Intelligent Control Systems.

8 References

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M.K. (1987). Occam's Razor. *Information Processing Letters,* 24, 377–380.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M.K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM,* 36, 4:929–965.

Gilbert, E. N. (1954). Lattice-theoretic properties of frontal switching functions. *Journal of Mathematical Physics*, 33, 57-67.

Haussler, D. (1990). Probably Approximate Correct Learning. *Technical Report UCSC-CRL-90-16.* Univ. of California at Santa Cruz.

Kearns, M. (1990). *The Computational Complexity of Machine Learning.* Cambridge, MA: MIT Press.

Kearns, M., Li, M., Pitt, L., and Valiant, L.G. (1987). On the Learnability of Boolean Formulae. *Proceedings of 19th Annual ACM Symposium on Theory of Computing* (pp. 285–295). New York, NY.

Maragos, P., & Schafer, R. W. (1987). Morphological Filters (Parts I and II). *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35, 8:1153-1184; *ibid.*, 37, 597, 1989.

Maragos P., & Schafer, R.W. (1990). Morphological Systems for Multidimensional Signal Processing. *Proceedings of IEEE*, 78, 4:690-710.

Rivest, R. L. (1990). *Lecture Notes in Machine Learning.* Massachusetts Institute of Technology.

Schapire, R. E. (1991). The Design and Analysis of Efficient Learning Algorithms, *Technical Report CICS-TH-273.* Brown-Harvard-MIT Center for Intelligent Control Systems.

Serra, J. (1982). *Image Analysis and Mathematical Morphology*, Acad. Press.

Valiant, L. G. (1984). A Theory of the Learnable. *Communications of the ACM,* 27, 11:1134–1142.

Valiant, L. G. (1985). Learning Disjunctions of Conjunctions. *Proceedings of 9th Int'l Joint Conference on Artificial Intelligence* (pp. 560-566). Los Angeles, CA.

Yang, P. F., & Maragos, P. (1991). Learnability of Thresholded Min-Max Functions. *Technical Report 91-8.* Harvard Robotics Laboratory.