On the Complexity of Learning From Counterexamples
(extended abstract)

Wolfgang Maass*,** and György Turán*,**

ABSTRACT

The complexity of learning concepts $C \in \mathcal{C}$ from various concrete concept classes $\mathcal{C} \subseteq 2^X$ over a finite domain X is analyzed in terms of the number of counterexamples that are needed in the worst case (we consider the deterministic learning model of Angluin, where the learning algorithm produces a series of "equivalence queries"). It turns out that for many interesting concept classes $\mathcal{C}$ there exist exponential differences between the number of counterexamples that are required by a "naive" learning algorithm for $\mathcal{C}$ (e.g. one that always outputs the minimal consistent hypothesis) and a "smart" learning algorithm for $\mathcal{C}$ that attempts to make a more sophisticated prediction (this is in contrast to the situation for pac-learning, where every consistent learning algorithm requires about the same number of examples).

We give $\theta(\log n)$ bounds for the number of counterexamples that are required for learning boxes, balls, and halfspaces in a d-dimensional discrete space $X = \{1, \ldots, n\}^d$ (for every finite dimension d). We also give an upper bound of $O(d^3)$ and a lower bound of $\Omega(d^2)$ for the complexity of learning a threshold function with d input bits (i.e. $X = \{0,1\}^d$). For each of these concept classes one can give learning algorithms that are both optimal (resp. close to optimal in the case of threshold functions) with regard to the number of counterexamples which they require <u>and</u> computationally feasible (in the case of balls, halfspaces and threshold functions our learning algorithms use the method of the ellipsoid algorithm).

Finally, we determine the complexity of learning of the considered concept classes (as well as linear orders, perfect matchings, and some other concept classes that turn out to be useful for the separation of learning models) on several variations of the considered learning model (such as learning with arbitrary hypotheses, partial hypotheses, membership queries). We also clarify the relationship between these learning models and some related combinatorial invariants.

1. INTRODUCTION

We analyze the complexity of learning "concepts" from various "concept classes" $\mathcal{C} \subseteq 2^X$ over a finite domain X. We consider a simple mathematical learning model where the learner produces a sequence of hypotheses $H_1, \ldots, H_i, \ldots$ from some hypothesis space $\mathcal{H}$ with $\mathcal{C} \subseteq \mathcal{H} \subseteq 2^X$ in order to identify (i.e. "learn") some "target concept" $C \in \mathcal{C}$ that has been fixed in the beginning by the environment (without knowledge of the learner). For each hypothesis H_i with $H_i \neq C$ the environment replies with some "counterexample" $x_i \in H_i \Delta C$ (where $H_i \Delta C := (C - H_i) \cup (H_i - C)$; x_i is called a "positive counterexample" if $x_i \in C - H_i$, x_i is called a "negative counterexample" if $x_i \in H_i - C$). We analyze the complexity of "learning algorithms" A for $\mathcal{C}$ that produce new hypotheses

$$H_{i+1}^A := A(x_1, \ldots, x_i;\ H_1^A, \ldots, H_i^A)$$

in dependence of the preceding hypotheses H_j^A and the received counterexamples $x_j \in H_j^A \Delta C$ (actually, the only information that A can gain from the preceding hypotheses H_j^A is the partition of $x_1, \ldots, x_i$ into positive and negative counterexamples).

We are interested in the number of counterexamples that a learning algorithm A needs for the learning of an arbitrary concept $C \in \mathcal{C}$, independently of the particular choice of counterexamples $x_j \in H_j^A \Delta C$ ("worst case analysis"):

$$\mathrm{LC}(A) := \max\{i \in \mathbf{N} \mid \text{there is some } C \in \mathcal{C} \text{ and some choice of counterexamples } x_j \in H_j^A \Delta C \text{ for } j = 1, \ldots, i-1 \text{ such that } H_i^A \neq C\}.$$

One also refers to $\mathrm{LC}(A)$ as the "mistake bound of A" [Li], [Ra]. One defines the learning complexity of a concept class $\mathcal{C}$ given a hypothesis space $\mathcal{H}$ as $\mathrm{LC}^{\mathcal{H}}(\mathcal{C}) := \min\{\mathrm{LC}(A) \mid A \text{ is a learning algorithm for } \mathcal{C} \text{ with hypothesis space } \mathcal{H}\}$.

We will focus primarily on the investigation of $\mathrm{LC}(\mathcal{C}) := \mathrm{LC}^{\mathcal{C}}(\mathcal{C})$. The case $\mathcal{H} = \mathcal{C}$ is of particular interest because it does not require the learner to know when the learning process is over (this appears to be a more realistic model). In this case one can view each H_i^A as a "temporary solution" to the learning problem, which may work well for a while until it is later refuted by some counterexample $x_i \in H_i^A \Delta C$.

One calls a learning algorithm A "consistent" if it only produces hypotheses $H_{i+1}^A = A(x_1, \ldots, x_i; H_1^A, \ldots, H_i^A)$ that are consistent with the received counterexamples $x_1, \ldots, x_i$. For the analysis of $\mathrm{LC}(\mathcal{C})$ one can assume that every learning algorithm A is consistent. The existence of a consistent learning algorithm for every $\mathcal{C}$ implies that always $\mathrm{LC}(\mathcal{C}) \leq |X|$.

Special cases of the model considered (where the hypotheses H_{i+1}^A have to be computed on a specific machine model) have already been considered for quite a while ([Ro], [N], [MP], [Li], [Ra]). The perceptron algorithms for learning a halfspace over $\{0,1\}^d$ give upper bounds containing $\frac{1}{\delta^2}$ for $\mathrm{LC}(\mathrm{HALFSPACE}_2^d)$ where δ is a measure of separation for the considered partition of the points ([Ro], [N], [MP], [Li]); note that δ can be exponentially small in d. Some first results about the machine-independent learning complexity $\mathrm{LC}(\mathcal{C})$ for concrete concept classes $\mathcal{C}$ are due to D. Angluin [A1]) (she refers to $\mathrm{LC}(\mathcal{C})$ as the required number of equivalence queries for $\mathcal{C}$; in the course of comparing

*Department of Mathematics, Statistics, and Computer Science, University of Illinois at Chicago, Chicago, IL. 60680. E-mail: U45381 @ UICVM.BITNET (W. Maass); U11557 @ UICVM.BITNET (G. Turan).

**Written under partial support by NSF-Grant CCR 8703889.

***Automata Theory Research Group of the Hungarian Academy of Sciences, Szeged, Hungary. Partially supported by OTKA-433.

various modes of learning she observed that LC(kDNF) $= O(n^k)$ and LC(SINGLETONS) $\geq |X| - 1$). Raghavan [Ra] gives a $k \log n$ upper bound on LC for learning "k-tonic" symmetric functions.

One appealing aspect of the model considered here is that it provides a simple mathematical framework for the analysis and comparison of the number of examples that are required by various learning algorithms. It has turned out that the related probabilistic learning model of Valiant [V] is less suited for this purpose. In Valiant's model the number of samples that are required by an (ϵ, δ)-learning algorithm A is almost the same for all consistent learning algorithms A for the same concept class $\mathcal{C}$ (according to [EHKV] they differ in the general case at most by a factor of $\log \frac{1}{\epsilon}$, and at most by a constant factor if $\log |\mathcal{C}| = \theta(VC\text{-dim}(\mathcal{C}))$). In contrast, it follows from the results of this paper that for many interesting concept classes $\mathcal{C}$ there exist exponential differences between LC(A) for a "naive" consistent learning algorithm A (e.g. one that always outputs the minimal consistent hypothesis) and LC(A) for a "smart" consistent learning algorithm A that attempts to make a more sophisticated prediction. For many concept classes $\mathcal{C}$ the use of such "smart" consistent learning algorithm A appears to be advantageous. Such algorithm is guaranteed to learn fast not only in a situation where the assumptions of Valiant's probabilistic model are met, but also in a less favorable situation (for example if the probability distribution of the sample points changes with the time). Furthermore we show in this paper that for many concrete concept classes $\mathcal{C}$ there exist "smart" learning algorithms that are computationally feasible.

The framework considered here also allows us to make quantitative comparisons between different "modes of learning" for the same concept class $\mathcal{C}$. The table and the figure at the end of this paper specify for various concept classes $\mathcal{C}$ whether one can speed up the learning process for $\mathcal{C}$ if one allows the learner to make "experiments" (i.e. he can ask "is x in the target concept?"), or if one allows the learner to consider more general types of hypotheses. Earlier results of this kind were obtained by Angluin [A1] and Littlestone [Li] (see the remarks following Figure 1 in Section 3).

Finally, the machine independent learning model considered here provides an interesting "yard stick" for the evaluation of the performance of various more restricted "learning machines" (in particular perceptrons and computational brain models, where the hypothesis is generated with severely limited computational resources and without a global control [Ro], [N], [MP], [Li], [Ra], [RM]). For example it is shown in Corollary 4 that LC(HALFSPACE$_2^d$) $= O(d^3)$ for the concept class HALFSPACE$_2^d$ of all subsets of $X := \{0,1\}^d$ that can be computed by a single threshold gate of fan - in d with arbitrary numbers as weights (we also show that LC(HALFSPACE$_2^d$) $= \Omega(d^2)$). This shows that a threshold gate which uses the familiar Δ-rule to generate new hypotheses is a relatively inefficient learner: it follows from the results of [MP] that it needs exponentially in d many counterexamples to learn certain $C \in$ HALFSPACE$_2^d$ (we will show in a subsequent paper that the same is true for Littlestone's variation of the Δ-rule [Li]).

In Section 2 upper and lower bounds on the learning complexity are proven for some concrete geometric learning problems such as boxes, halfspaces and balls in d dimensions. Different learning modes and some related combinatorial parameters are considered in Section 3.

2. BOUNDS ON THE LEARNING COMPLEXITY FOR GEOMETRIC PROBLEMS

In the first theorem we consider the concept class

$$\text{BOX}_n^d := \{\{i_1, i_1+1, \dots, j_1\} \times \dots \times \{i_d, i_d+1, \dots, j_d\} \mid 1 \leq i_k, j_k \leq n \text{ for } k = 1, \dots, d\},$$

which consists of all rectangular axis-parallel "boxes" that are contained in the discrete d-dimensional space

$$X_n^d := \{1, \dots, n\}^d$$

(similar discrete geometrical objects have been considered in [MP] in the context of computations by perceptrons). Note that a "naive" learning algorithm which always outputs the minimal consistent box as hypothesis needs $\Omega(n)$ counterexamples to learn arbitrary $C \in \text{BOX}_n^d$.

Theorem 1. Consider any fixed dimension $d \in \mathbf{N}$. Then LC(BOX$_n^d$) $= \theta(\log n)$. Furthermore there exists a learning algorithm A for BOX$_n^d$ with LC(A) $= O(\log n)$ that uses altogether at most $O(\text{poly}(\log n))$ computation steps.

Sketch of the Proof. In order to design a learning algorithm for BOX$_n^d$ that learns substantially faster than the naive algorithm (which always outputs the minimal consistent hypothesis) one has to generate hypotheses that interpolate between the minimal consistent hypothesis and some maximal consistent hypothesis. However for $d > 1$ there is in general no unique maximal box that is consistent with the previously received counterexamples. Some maximal consistent box may run *to the right* of some negative counterexample x, while another one avoids x by running *below* x to the left. This ambiguity corresponds to conflicting "theories" *why* x is not in the target box (more precisely: which of the defining conditions for points in the target box are not met by x). For BOX$_n^d$, as well for most other concrete concept classes that are discussed below, the interesting point in the design of an efficient learning algorithm lies in the construction of a next hypothesis H_{i+1}^A that guarantees substantial progress (from any counterexample to H_{i+1}^A) no matter which of the conflicting "theories" about the explanation of the previously received counterexamples are true. Technically, this amounts to giving the right definition of "progress" for learning in the considered concept class.

For BOX$_n^d$ it is useful to measure the learning progress in terms of the number of points in X_n^d that could be a corner-point of the target box (on the basis of all counterexamples received so far). For $d = 2$ (the general case is analogous) let NW $\subseteq X_n^2$ be the set of currently still possible locations of the "north-west corner" $\langle i_1, j_2 \rangle$ of the (unknown) target box $C = \{i_1, \dots, j_1\} \times \{i_2, \dots, j_2\}$. The sets SW, NE, SE of still possible locations for the other three corner points of the target box are defined analogously.

Our learning algorithm starts with the hypothesis $H_1^A = \emptyset$. After step i it construct a hypothesis $H := H_{i+1}^A$ s.t. any counterexample to H reduces the size of at least one of the sets NW, SW, NE, SE by a constant fraction. We fix a vertical line L_v^{NW} that partitions NW in such a way that on each side of L_v^{NW} there are at least one third of the points of NW (the case where there is no line with this property can be handled by a separate argument). Then we fix a horizontal line L_h^{NW} that partitions the points of NW to the left of L_v^{NW} into two sets s.t. each set has at least $\frac{1}{9}$ of the size of NW. For the region SW of all still possible locations of $\langle i_1, j_1 \rangle$ one constructs two lines L_v^{SW}, L_h^{SW} in the same fashion. The lines L_v^{NE}, L_h^{NE}, L_v^{SE}, L_h^{SE} which partition the regions NE and SE are constructed analogously, but here one chooses $L_h^{\text{NE}}(L_h^{\text{SE}})$ so that it partitions the points of NE(SE) to the *right* of $L_v^{\text{NE}}(L_v^{\text{SE}})$ into sets of size $\geq \frac{1}{9}$ of the size of NE(SE).

The next hypothesis H is then constructed as follows: Its left borderline is determined by the rightmost one of the two lines L_v^{NW}, L_v^{SW}, its right borderline by the leftmost one of the lines L_v^{NE}, L_v^{SE}. Analogously the upper borderline of H is determined by the lower one of the two lines L_h^{NW}, L_h^{NE}, and its lower borderline by the higher one of L_h^{SW}, L_h^{SE}.

One can then verify that if H does not agree with the target box, then no matter which counterexample the learner gets he can eliminate at least $\frac{1}{9}$ of the points of one of the regions NW, SW, NE, SE.

This concludes the sketch of the proof that $\mathrm{LC}(\mathrm{BOX}_n^2) = O(\log n)$ (the proof for an arbitrary dimension d is analogous). It is very easy to prove directly that $\mathrm{LC}(\mathrm{BOX}_n^d) = \Omega(\log n)$. Alternatively one can use the fact that $\mathrm{chain}(\mathrm{BOX}_n^d) = \Omega(n)$ (see Figure 1 in Section 3), or apply Lemma 7.

A straightforward argument shows that the learning algorithm for BOX_n^d that we have sketched requires altogether at most $O(\mathrm{poly}(\log n))$ computation steps (observe that at any stage of the learning process one can represent each of the regions that consists of the possible locations of one of the corners of the target box as a disjoint union of $\mathrm{poly}(\log n)$ boxes). $\square$

Remark. We will show in a subsequent paper that boxes are substantially more difficult to learn if they are not required to be axis-parallel.

Now we turn to the problem of learning halfspaces. For $X \subseteq \mathbf{R}^d$ let

$$\mathrm{HALFSPACE}_X^d = \{S \subseteq X \mid \exists \text{ halfspace } F \text{ in } \mathbf{R}^d \text{ s.t. } X \cap F = S\}.$$

Instead of $\mathrm{HALFSPACE}_X^d$ we write $\mathrm{HALFSPACE}_n^d$ if $X = X_n^d = \{1, \dots, n\}^d$, and $\mathrm{HALFSPACE}_2^d$ if $X = \{0,1\}^d$. First we present a learning algorithm for this concept class.

Theorem 2. There is a learning algorithm A for $\mathrm{HALFSPACE}_2^d$ with $\mathrm{LC}(A) = O(d^3 \log d)$ such that the total computation time required by A is polynomial in d.

Idea of the Proof. The algorithm is an application of the ellipsoid method ([K], [GLS], [Sch]).

A point $\underline{x} = (x_1, \dots, x_d) \neq \underline{0}$ and a hyperplane E in $\mathbf{R}^d$ with $\underline{0} \notin E$ are dual if E is given by $x_1 u_1 + \dots + x_d u_d = 1$; we write $E = \mathrm{dual}(\underline{x})$, $\underline{x} = \mathrm{dual}(E)$. If E is a hyperplane with $\underline{0} \notin E$, E^+ is the closed halfspace containing $\underline{0}$.

We may assume w.l.o.g. that $\underline{0}$ is in the target concept (use the empty set as first hypothesis; if a counterexample is obtained, this must be a positive one, let this point be the origin).

For a target concept S let $\mathrm{SOL}_S := \{\mathrm{dual}(E) \mid E^+ \cap \{0,1\}^d = S\}$. It follows by standard arguments that there is a ball B of radius $2^{O(d \log d)}$ around $(1, 0, \dots, 0)$ such that for every S, the volume of $\mathrm{SOL}_S \cap B$ is $2^{-O(d^2 \log d)}$.

Assume we have an ellipsoid L with center $\underline{y}$ containing $\mathrm{SOL}_S \cap B$ (initially $L = B$). Consider as next hypothesis dual $(\underline{y})^+ \cap \{0,1\}^d$, i.e. the concept corresponding to $\underline{y}$.

Any counterexample $\underline{x}$ to this hypothesis defines a halfspace H (defined by dual $(\underline{x})$) s.t. $\mathrm{SOL}_S \subseteq H$ and $\underline{y} \notin H$. We replace L by a smaller ellipsoid L' with $L' \supseteq L \cap H$ and repeat the procedure. The progress we make with one hypothesis is the decrease in the volume of the ellipsoids, i.e. a factor $\leq e^{-1/2(d+1)}$. The lower bound on the volume of $\mathrm{SOL}_S \cap B$ implies that at most $O(d^3 \log d)$ iterations are necessary.

A technical detail here is that if the center of the ellipsoid L happens to be $\underline{0}$, it has to be replaced by another ellipsoid with center $\neq \underline{0}$ which contains L and is only "slightly" larger. This can be achieved using standard methods without changing the bounds above.

The computation necessary for implementing this algorithm consists of the updating of the ellipsoids. An updating requires $O(d^3)$ arithmetic operations. The arithmetic operations have to be performed with large precision, but polynomially many digits are sufficient (the theoretical upper bound is $O(d^4 \log d)$ digits). Hence the total computation required is polynomial. $\square$

The argument above can be generalized for $\mathrm{HALFSPACE}_n^d$.

Corollary 3. There is a learning algorithm A for $\mathrm{HALFSPACE}_n^d$ with $\mathrm{LC}(A) = O(d^3(\log d + \log n))$ such that the total computation time required is polynomial in d and $\log n$. $\square$

The problem with generalizing this approach to $\mathrm{HALFSPACE}_X^d$ is that in general there is no lower bound on the volume of the solution set in terms of d and n, thus we do not get any upper bound on the number of iterations necessary. Nevertheless, one can give a similar algorithm for the general case as well, using the notion of centerpoints.

Theorem 4. For every $X = \{\underline{x}_1, \dots, \underline{x}_n\} \subseteq \mathbf{R}^d$,

$$\mathrm{LC}(\mathrm{HALFSPACE}_X^d) = O(d^2 \log n).$$

Idea of the Proof. A point $\underline{y} \in \mathbf{R}^d$ is called a centerpoint of a finite set $Y \subseteq \mathbf{R}^d$ if every open halfspace not containing $\underline{y}$ contains $\leq \frac{d}{d+1}|Y|$ points from Y.

Lemma 5. ([YB], see [Ed].) For every finite set there exists a centerpoint. $\square$

We may assume w.l.o.g. that $\underline{0} \notin X$ and $\underline{0}$ is contained in the target halfspace. By continuity, we may also assume that the hyperplane defining the target halfspace is disjoint from $X \cup \{\underline{0}\}$.

Now let $E_i := \mathrm{dual}(\underline{x}_i)$ and consider the open regions of $\mathbf{R}^d$ formed by these hyperplanes. The positive halfspaces corresponding to hyperplanes (in the original space) represented by a region form an equivalence class, giving rise to the same target concept. Select points $Q_1, \dots, Q_m$, one from each region s.t. each subset of these points has a centerpoint different from $\underline{0}$ and not contained in any of the E_i's.

During the learning algorithm we maintain a set CAND of those Q_i's which have not been excluded. Initially $\mathrm{CAND} = \{Q_1, \dots, Q_m\}$. The next hypothesis is $\mathrm{dual}(Q)^+ \cap X$, where Q is a centerpoint of CAND with the properties above. If a counterexample is obtained, we update CAND.

Initially $|\mathrm{CAND}| = m = 0(n^d)$ (see e.g. [Ed]). Arguing as in Theorem 2, it follows from the definition of a centerpoint that each counterexample reduces CAND by a factor $\leq \frac{d}{d+1}$. Thus the number of iterations needed is $O(\log_{(d+1)/d} n^d) = O(d^2 \log n)$. $\square$

Corollary 6. $\mathrm{LC}(\mathrm{HALFSPACE}_n^d) = O(d^3 \log n)$, in particular

$$\mathrm{LC}(\mathrm{HALFSPACE}_2^d) = O(d^3). \qquad \square$$

Now we give some lower bounds which differ by a factor d from the upper bounds.

A decision tree T for $(X, \mathcal{C})$ is a binary tree with inner nodes labelled by elements of X, edges labelled by 0 or 1, and leaves labelled by concepts $C \in \mathcal{C}$ s.t.

- every inner node has two outgoing edges labelled 0 or 1,
- for every $C \in \mathcal{C}$ there is exactly one leaf labelled C,
- the labels along the path leading from the root to a leaf labelled C correspond to membership of the queried elements in C.

Let $d_{\min}(T)$ be the minimal depth of a leaf in T and let $\mathrm{ADV}(\mathcal{C}) = \max\{d_{\min}(T) \mid T$ is a decision tree for $\mathcal{C}\}$ be the "adversary complexity" of $\mathcal{C}$. One may view $\mathrm{ADV}(\mathcal{C})$ as the "dual decision tree complexity" of $\mathcal{C}$.

We write LC - ARB($\mathcal{C}$) for $\mathrm{LC}^{2^X}(\mathcal{C})$ (in this model one may use arbitrary subsets of X as hypotheses). Obviously LC - ARB($\mathcal{C}$) $\leq$ LC($\mathcal{C}$) for every $\mathcal{C}$.

The following lemma is a reformulation of a result of Littlestone [Li, Th. 3] (in his notation: $K(\mathcal{C}) = \mathrm{opt}(\mathcal{C})$).

Lemma 7. [Li] ADV($\mathcal{C}$) = LC - ARB($\mathcal{C}$) for every concept class $\mathcal{C}$. □

Using Lemma 7 one can prove a lower bound for LC($\mathcal{C}$) (and for LC - ARB($\mathcal{C}$)) by constructing a decision tree with a "bad best-case behavior" (i.e. a tree which has only "long" paths).

Theorem 8. $\mathrm{LC}(\mathrm{HALFSPACE}_2^d) = \Omega(d^2)$.

Idea of the Proof. A decision tree can be constructed using the argument proving a lower bound to the number of threshold functions (see [M]). □

This argument can also be generalized to $\mathrm{HALFSPACE}_n^d$.

Corollary 9. $\mathrm{LC}(\mathrm{HALFSPACE}_n^d) = \Omega(d^2 \log n)$. □

Remark. The preceding results show that the complexity of learning a Boolean threshold function of d variables (with weights of arbitrary size) is $\Omega(d^2)$ and $O(d^3)$. We are not aware of any previously published nonlinear lower bounds or polynomial upper bounds for this problem.

For $X \subseteq \mathbf{R}^d$ let

$$\mathrm{BALL}_X^d = \{S \subseteq X \mid \exists \text{ ball } B \subseteq \mathbf{R}^d \text{ s.t. } X \cap B = S\}.$$

Theorem 10. For every $X = \{\underline{x}_1, \ldots, \underline{x}_n\} \subseteq \mathbf{R}^d$ $\mathrm{LC}(\mathrm{BALL}_X^d) = O(d^2 \log n)$. For the case $X = \{1, \ldots, n\}^d$ one can give a learning algorithm A for BALL_X^d with $\mathrm{LC}(A) = O(d^3(\log d + \log n))$ such that the total computation time of A is polynomial in d and $\log n$.

Idea of the proof. One can reduce this to the problem of learning halfspaces by projecting X to the paraboloid $P \subseteq \mathbf{R}^{d+1}$ defined by $u_{d+1} = \sum_{i=1}^{d} u_i^2$. □

Using this theorem and the argument of Corollary 9 one can prove

Proposition 11. For $X = \{1, \ldots, n\}^d$

$$\mathrm{LC}(\mathrm{BALL}_X^d) = O(d^3 \log n) \text{ and } \mathrm{LC}(\mathrm{BALL}_X^d) = \Omega(d^2 \log n). \qquad \square$$

3. DIFFERENT MODES OF LEARNING

In this section we consider variations on the learning model and some concept classes which are either of some interest on their own or are useful for proving relations between the various learning modes.

In Table 1, we consider the concept classes $\mathrm{SINGLETONS}_n$, $\mathrm{SINGLETONS}_n \cup \{\emptyset\}$, LINEAR ORDER_n, PERFECT $\mathrm{MATCHING}_n$ and $\mathrm{ADDRESSING}_n$.

One defines

$$\mathrm{SINGLETONS}_n := \{\{i\} \mid i = 1, \ldots, n\}, \text{ with domain } \{1, \ldots, n\}.$$

Let $X_n := \{(i,j) \mid 1 \leq i < j \leq n\}$. We define

$$\text{LINEAR ORDER}_n := \{C \subseteq X_n \mid \exists \text{ linear order } \prec \text{ on } \{1, \ldots, n\} \text{ s.t. for every } (i,j) \in X_n((i,j) \in C \Leftrightarrow i \prec j)\};$$

$$\text{PERFECT MATCHING}_n := \{C \subseteq X_n \mid C \text{ is a perfect matching on } \{1, \ldots, n\}\}.$$

For $\mathrm{ADDRESSING}_n$ we set $X_n := Y_n \cup Z_n$, $Y_n := \{y_0, \ldots, y_{\lfloor \log n \rfloor - 1}\}$, $Z_n := \{z_0, \ldots, z_{n-1}\}$, and we define

$$\mathrm{ADDRESSING}_n := \{Y \cup \{z_i\} \mid Y \subseteq Y_n \text{ and } i \text{ is the number denoted by the characteristic vector of } Y\}.$$

In Table 1 we consider, besides LC($\mathcal{C}$) and LC - ARB($\mathcal{C}$) (defined preceding Lemma 7) the following learning modes and corresponding complexity measures.

- MEMB($\mathcal{C}$) (this is the "decision tree complexity" of $\mathcal{C}$, where the "learner" can only ask membership queries $x \in C$?, for $x \in X$).
- LC - MEMB($\mathcal{C}$) (here the learner can ask membership queries and he can present hypotheses from $\mathcal{C}$; this is a combination of the learning capabilities of the models for LC($\mathcal{C}$) and MEMB($\mathcal{C}$)).
- LC-HALVING($\mathcal{C}$) (here the learner uses a specific algorithm for the LC-ARB model: the next hypothesis H_{i+1} is always the set of those points in the domain that lie in at least 50% of all $C \in \mathcal{C}$ that are consistent with the first i counterexamples; note that LC-HALVING($\mathcal{C}$) is the same as $\mathrm{M_{HALVING}}(\mathcal{C})$ in [Li]).
- LC - PARTIAL($\mathcal{C}$) (here the learner can present arbitrary "partial" hypotheses $H \in \{0, 1, *\}^X$, he gets as response either some $x \in X$ s.t. $H(x) \in \{0, 1\}$ and $H(x) = 1 - C(x)$, where $C \in \mathcal{C}$ is the target concept, or he gets the message "H is correct" if no such $x \in X$ exists; note that all the other previously discussed learning models are special cases of this model).

All of these models, except for the last one, have previously been studied ([A1], [Li]). The last model appears to be of some interest because partial hypotheses are common in human learning (note that a typical hypothesis does not assign a truth value simultaneously to every possible yes/no decision in the world). With regard to the speed of learning a partial hypothesis may be more advantageous than an arbitrary hypothesis $H \in \{0, 1\}^X$ in a situation where one has many $x \in X$ that are "unbalanced" (i.e. the number of remaining consistent concepts that contain x is much larger or much smaller than the number of remaining concepts that do not contain x). One can then present a partial hypothesis that only specifies the "probable" behaviour of the unbalanced elements (and one can leave the behaviour of the other elements open). In this way one may arrange that each possible response brings more progress than just halving the number of remaining concepts.

A nice illustration of the power of this learning mode is the proof that LC-PARTIAL($\mathrm{ADDRESSING}_n$) ≤ 1. In this case the learner produces a partial hypothesis that assigns 0 to all $z \in Z_n$ and * to all $y \in Y_n$. Any counterexample to this has to be a positive counterexample from Z_n.

We define chain $(\mathcal{C}) := \max\{\ell \in \mathbf{N} \mid \exists C_i \in \mathcal{C},\ i = 1, \ldots, \ell \text{ with } C_1 \subsetneq C_2 \subsetneq \ldots \subsetneq C_\ell\}$.

Remarks to Table 1.

a) The upper bound for LC(LINEAR ORDER_n) is obtained by considering the following learning algorithm. Let P be the partial order formed by the previous counterexamples (initially P is an antichain). Let $h_P(i)$ be the average position of i taken over all linear extensions of P. The next hypothesis is obtained by ordering the elements according

TABLE 1	1 SINGLETONS$_n$	2 SINGLETONS$_n$ $\cup \{\phi\}$	3 ADDRESSING$_n$	4 BOX^d_n, BALL^d_n HALFSPACE^d_n for fixed d	LINEAR ORDER$_n$	PERFECT MATCHING$_n$
LC($\mathcal{C}$) (hypotheses from $\mathcal{C}$)	n	1	n	log n	n log n	n^2
LC-ARB($\mathcal{C}$) (arbitrary hyp. $\subseteq$ X)	1	1	log n	log n	n log n	n
LC-PARTIAL($\mathcal{C}$) (partial hypotheses)	1	1	1	log n	n	n
LC-MEMB($\mathcal{C}$) (hyp. $\in \mathcal{C}$ and memb.qu.)	n	1	log n	log n	n log n	n^2
MEMB($\mathcal{C}$) (membership queries)	n	n	log n	n^d (*)	n log n	n^2
VC-dim($\mathcal{C}$)	1	1	log n	1	n	n
log(chain($\mathcal{C}$))	1	1	1	log n	1	1

All bounds in this table are Θ-bounds.

(*)exception: MEMB(HALFSPACE^d_n) = Θ(log n)

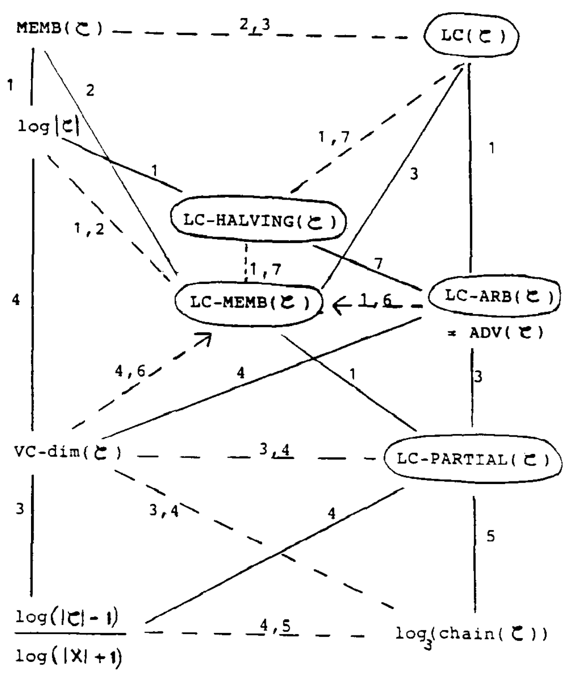

FIGURE 1

to their h_P values (ties are resolved arbitrarily). It follows from the theorem of Kahn and Saks [KS] that every counterexample to this hypothesis reduces the number of candidates by a constant factor. The lower bound follows from Lemma 7 using the decision tree of mergesort as adversary tree.

b) The lower bound for LC(PERFECT MATCHING$_n$) follows by considering an adversary which always gives negative counterexamples (the selection of the negative counterexample is arbitrary as long as there is such a pair). When this is no longer possible, the complement of the previous counterexamples contains a unique perfect matching. A result of Hetyei ([H], see [Lo], Problem 7.24 c) implies that $\Omega(n^2)$ hypotheses must have been asked (see [FT] for a similar argument). The lower bound can be generalized to an $\Omega(n^k)$ lower bound for the generalization of this problem to k-uniform hypergraphs using a generalization of Hetyei's result due to Erdös [Er].

c) ADDRESSING$_n$ turns out to be of particular interest because it separates the three learning modes LC, LC - ARB, LC - PARTIAL.

In the remaining part of the section we summarize the known relationship between these complexity measures and some combinatorial parameters.

Remarks to Figure 1.

a) If A is above B and both are connected by a solid line then this indicates that $A(\mathcal{C}) \geq B(\mathcal{C})$ for every concept class $\mathcal{C}$, and that for some $\mathcal{C}$ $A(\mathcal{C})$ is exponentially larger than $B(\mathcal{C})$ ($\theta(\log n)$ versus $\theta(1)$, or $\theta(n)$ versus $\theta(\log n)$). The number next to the line specifies a class $\mathcal{C}$ with the latter property. Note that the relation between A and B that is defined by a solid line is a transitive relation.

b) If A and B are connected by a broken line without an arrow then A and B are incomparable in the strong sense that for some $\mathcal{C}$ $A(\mathcal{C})$ is exponentially larger than $B(\mathcal{C})$, and for some other $\mathcal{C}$ $B(\mathcal{C})$ is exponentially larger than $A(\mathcal{C})$ (such classes $\mathcal{C}$ are specified by the two numbers next to the broken line). A broken line with an arrow from B to A indicates that there is some $\mathcal{C}$ for which $A(\mathcal{C})$ is exponentially larger than $B(\mathcal{C})$, and that there is some other $\mathcal{C}$ for which $A(\mathcal{C}) \leq \frac{1}{2} \cdot B(\mathcal{C})$ (but it is open whether there exists some $\mathcal{C}$ for which $B(\mathcal{C})$ is exponentially larger than $A(\mathcal{C})$).

c) It is obvious from the definition that a broken line between A and B (with or without an arrow) rules out that A and B are in the relation that is expressed by the solid line (in either direction). Therefore the combined results that are indicated in Figure 1 settle for *every* pair A, B of the occuring learning modes and combinatorial parameters the question whether A and B are in the relation that is expressed by the solid line.

d) The numbers 1 to 4 in Figure 1 refer to the first four concept classes (with the same numbers) in Table 1. Class 5 is

$$\mathrm{HALFSIZE}_n := \left\{ S \subseteq \{1, \ldots, n\} \mid |S| = \left\lfloor \frac{n}{2} \right\rfloor \right\}$$

and class 6 is

$$\mathrm{MAJORITY}_n := \left\{ S \subseteq \{0, 1, 2, \ldots, n\} \mid 0 \in S \Leftrightarrow |S - \{0\}| > \frac{n}{2} \right\}.$$

For class 7 we set $X_n := \{1, \ldots, n + \lceil \log n \rceil\}$, and we define class 7 as

$$\text{TAGGED SINGLETONS}_n := \{\emptyset\} \cup \{\{n+\ell\} \mid \ell \in \{1, \ldots, \lceil \log n \rceil\}\} \cup \Big\{ \{i\} \cup \{n+\ell\} \mid i \in \{1, \ldots, n\},\ \ell \in \{1, \ldots, \lceil \log n \rceil\}, \text{ and } \ell \text{ is minimal such that } i \leq \sum_{j=1}^{\ell} \left(\frac{n}{2^j} + 1 + \lceil \log n \rceil \right) \Big\}.$$

It is easy to see that LC(TAGGED SINGLETONS$_n$) ≤ 2, whereas

$$\text{LC - HALVING(TAGGED SINGLETONS}_n) = \Omega(\log n).$$

e) On the first sight it is somewhat surprising that there are at all concept classes $\mathcal{C}$ for which LC - MEMB($\mathcal{C}$) $<$ VC $-$ dim($\mathcal{C}$) $\leq$ min(LC($\mathcal{C}$), MEMB($\mathcal{C}$)), or LC - MEMB($\mathcal{C}$) $<$ LC - ARB($\mathcal{C}$). However it is easy to see that

$$\begin{aligned} \text{LC - MEMB}(\mathrm{MAJORITY}_n) &\leq \frac{n}{2} + 1 < n = \text{VC - dim}(\mathrm{MAJORITY}_n) \\ &= \text{LC - ARB}(\mathrm{MAJORITY}_n). \end{aligned}$$

MAJORITY$_n$ is obtained by decomposing $\{0,1\}^n$ into two balls of radius $\frac{n}{2}$ with centers $(0, \ldots, 0)$ and $(1, \ldots, 1)$. More generally, by decomposing $\{0,1\}^n$ into ℓ balls of radius k and using $\lceil \log \ell \rceil$ new variables for addressing, one can obtain a concept class with VC-dimension n that can be learned with $\lceil \log \ell \rceil$ membership queries followed by k hypotheses from the concept class. It can be shown that for some concept class of this type $\lceil \log \ell \rceil + k = (2 - \log 3)n + o(n) < \frac{n}{2}$, improving the separation between LC - MEMB and the VC-dimension.

f) It was already previously known that for every $\mathcal{C}$

$$\log(|\mathcal{C}| - 1)/\lceil \log |X| \rceil \leq \text{VC - dim}(\mathcal{C}) \leq \text{LC - ARB}(\mathcal{C}) \leq \text{LC - HALVING}(\mathcal{C}) \leq \log |\mathcal{C}| \leq \text{MEMB}(\mathcal{C}) \quad \text{([A1], [EKHV], [Li])}.$$

Furthermore it is obvious from the definitions that LC - PARTIAL($\mathcal{C}$) $\leq$ LC - ARB($\mathcal{C}$) $\leq$ LC($\mathcal{C}$) and LC - MEMB($\mathcal{C}$) $\leq$ min(LC($\mathcal{C}$), MEMB($\mathcal{C}$)).

Littlestone [Li] had already observed that there are asymptotic gaps between VC - dim($\mathcal{C}$) and LC - ARB($\mathcal{C}$), and between LC - HALVING($\mathcal{C}$) and $\log |\mathcal{C}|$. He also constructed an example of a concept class $\mathcal{C}$ over an 8-element domain where LC - HALVING($\mathcal{C}$) $= 3 > 2 =$ LC - ARB($\mathcal{C}$).

g) Lower bounds for LC - ARB($\mathcal{C}$) (or LC($\mathcal{C}$)) can often be shown by proving a lower bound for VC - dim($\mathcal{C}$) (using that VC - dim($\mathcal{C}$) $\leq$ LC - ARB($\mathcal{C}$) for all concept classes $\mathcal{C}$). However LC - PARTIAL($\mathcal{C}$) is incomparable with VC - dim($\mathcal{C}$), and one has to prove lower bounds for LC - PARTIAL($\mathcal{C}$) in a different way. In many cases one can do this by giving a lower bound for $\log_3(\text{chain}(\mathcal{C}))$ or $\log(|\mathcal{C}| - 1)/\log(|X| + 1)$.

Acknowledgement

We are grateful to Paul Erdös, David Haussler, and Manfred Warmuth for their valuable comments.

References

[A1] D. Angluin. Types of queries for concept learning, Yale Univ. Dept. Comp. Sci., TR-479, 1986, 29pp.

[A2] D. Angluin. Learning regular sets from queries and counterexamples, Inf. and Contr., 75, (1987), 87-106.

[Ed] H. Edelsbrunner. Algorithms in Combinatorial Geometry, EATCS Monographs on Theor. Comp. Sci., Vol. 10, Springer, 1987.

[EKHV] A. Ehrenfeucht, D. Haussler, M. Kearns, L. Valiant. A general lower bound on the number of examples needed for learning, Proc. of the 1988 Workshop on Computational Learning Theory (COLT '88), ed. D. Haussler, L. Pitt; Morgan Kaufman Publ. [San Mateo, 1988], 139-154.

[Er] P. Erdös. Personal communication, 1989.

[FT] U. Faigle, Gy. Turán. Sorting and recognition problems for ordered sets, SIAM J. Comp., 17, (1988), 100-113.

[GLS] M. Grötschel, L. Lovász, A. Schrijver. The Ellipsoid Method and Combinatorial Optimization, Springer, 1986.

[H] G. Hetyei. Pécsi Tan. Föisk. Közl., 1964, 151-168.

[IJ] O. H. Ibarra, T. Jiang. Learning regular languages from counterexamples. Proc. of the 1988 Workshop on Computational Learning Theory (COLT '88), ed. D. Haussler, L. Pitt; Morgan Kaufman Publ. [San Mateo, 1988], 371-385.

[KS] J. Kahn, M. Saks. Balancing poset extensions, Order 1, (1984), 113-126.

[K] L. G. Khachiyan. A polynomial algorithm in linear programming, Dokl. Akad. Nauk SSSR, 244, (1979), 1093-1096. (English translation: Soviet Mathematics Doklady 20, 1979, 191-194.)

[Li] N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm, *Machine Learning*, 2 (4), (1987), 285-318.

[Lo] L. Lovász. Combinatorial Problems and Exercises. Akadémiai Kiadó (1979).

[MP] M. Minsky, S. Papert. Perceptrons: An Introduction to Computational Geometry, Expanded edition, MIT Press, 1988.

[M] S. Muroga. Threshold Logic and its Applications, Wiley (New York), 1971.

[N] N. Nilsson. Learning Machines, McGraw-Hill (New York, 1965).

[Ra] P. Raghavan. Learning in threshold networks, Proc. of the 1988 Workshop on Computational Learning Theory (COLT '88), ed. D. Haussler, L. Pitt; Morgan Kaufmann Publ., [San Mateo, 1988], 19-27.

[Ro] F. Rosenblatt. Principles of Neurodynamics, Spartan Books (New York, 1962).

[RM] D.E. Rumelhart, J. L. McClelland. Parallel Distributed Processing. MIT Press (Cambridge, 1986).

[Sch] A. Schrijver. Theory of Linear and Integer Programming, Wiley (New York), 1986.

[V] L. G. Valiant. A theory of the learnable, Proc. of the 1984 STOC, 436-445.

[YB] M. Yaglom, V. G. Boltyanskii. Convex Figures, English translation. Holt, Rinehart and Winston, 1961.