On the Complexity of Learning From Counterexamples
(extended abstract)

Wolfgang Maass* and György Turán**

ABSTRACT

The complexity of learning concepts \( C \subseteq \mathcal{C} \) 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 \( \Theta(n) \) 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 \( O(n^d) \) and a lower bound of \( \Omega(n^d) \) 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 and 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 \( H_1, \ldots, H_i \) 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" \( C \) 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" \( x \in \mathcal{H} \setminus \Delta C \) (where \( H_i \Delta C := (C \setminus H_i) \cup (H_i \setminus C) \) : \( x \) is called a "positive counterexample" if \( x \in C \setminus H_i \) : \( x \) is called a "negative counterexample" if \( x \in H_i \setminus C \)). We analyze the complexity of "learning algorithms" \( A \) for \( \mathcal{C} \) that produce new hypotheses

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

in dependence of the preceding hypotheses \( H_A \) and the received counterexamples \( x_j \in H_A \setminus C \) (actually, the only information that \( A \) can gain from the preceding hypotheses \( H_A \) 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 \( x_j \in H_A \setminus C \) ("worst case analysis")

\[ \text{LC}(A) := \max \{ i \in \mathbb{N} \mid \text{there is some } C \in \mathcal{C} \text{ and some choice of counterexamples } x_j \in H_A \setminus C \text{ such that } H_A^i \neq C \} \]

One also refers to \( \text{LC}(A) \) 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 \( \text{LC}(\mathcal{C}) := \min \{ \text{LC}(A) \mid A \text{ is a learning algorithm for } \mathcal{C} \text{ with hypothesis space } \mathcal{H} \} \).

We will focus primarily on the investigation of \( \text{LC}(\mathcal{C}) := \text{LC}^\mathcal{C}(\mathcal{C}) \). The case \( \mathcal{H} = C \) 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 \( H_A \) as a "temporary solution" to the learning problem, which may work well for a while until it is later refuted by some counterexample \( x_l \in H_A \setminus C \).

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

Special cases of the model considered (where the hypotheses \( H_A^i \) 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 \( \frac{1}{\delta} \) for \( \text{LC}(\text{HALFSPACE}) \) where \( \delta \) is a measure of separation for the considered partition of the points ([Ro], [N], [MP], [Li]); note that \( \delta \) can be exponentially small in \( d \). Some first results about the machine-independent learning complexity \( \text{LC}(\mathcal{C}) \) for concrete concept classes \( \mathcal{C} \) are due to D. Angluin [A1] (she refers to \( \text{LC}(\mathcal{C}) \) as the required number of equivalence queries for \( C \); in the course of comparing...
various modes of learning she observed that \( LC(KDFN) = O(n^2) \) and \( LC(SINGLETONS) \geq |X| - 1 \). Raghavan [Ra] gives a \( kn \) 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 \((c,b)\)-learning algorithm \( A \) is almost always the same for all consistent learning algorithms \( A \) for the same concept class \( C \) (according to [EHKV] they differ in the general case at most by a factor of \( \log n \), and at most by a constant factor if \( \log |C| = \Theta(VC - dim(C)) \)). In contrast, it follows from the results of this paper that for many interesting concept classes \( C \) there exist exponential differences between \( LC(A) \) for a "naive" consistent learning algorithm \( A \) and \( LC(A) \) for a "smart" consistent learning algorithm \( A \) that attempts to make a more sophisticated prediction. For many concept classes \( C \) the use of such "smart" consistent learning algorithm \( A \) appears to be advantageous. Such a guarantee is learned 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 \( C \) 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 \( C \). The table and the figure at the end of this paper specify for various concept classes \( C \) whether one can speed up the learning process for \( C \) 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 Singh [A] and Littlestone [Li] (see the remarks following Figure 1 in Section 4).

Finally, the machine independent learning model considered here provides an interesting "yardstick" for the evaluation of the performance of various more restricted "learning machines" (in particular perceptron 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 \( LC(HALFSPACE) = O(d^2) \) for the concept class \( HALFSPACE \) 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 \( LC(HALFSPACE) = O(d^2) \)). This shows that a threshold gate which uses the familiar \( A \)-rule to generate new hypotheses is a relatively inefficient learner: it follows from the results of [MP] that it needs exponentially \( n \) many counterexamples to learn certain \( C \in HALFSPACE \) (we will show in a subsequent paper that the same is true for Littlestone's variation of the \( A \)-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 \( BOX := \{ \{i_1,i_1+1,\ldots,i_2\} \times \cdots \times \{i_k,i_k+1,\ldots,i_{k+1}\} \mid 1 \leq i_v,j_v \leq n \text{ for } v = 1,\ldots,d \} \), which consists of all rectangular axis-parallel "boxes" that are contained in the discrete \( d \)-dimensional space \( X := \{1,\ldots,n\}^d \), (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 \( C \in BOX \).

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

Sketch of the Proof. In order to design a learning algorithm for \( BOX \) which 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 \( BOX \) it is 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 \( H_{n+1} \), that guarantees substantial progress (from any counterexample to \( H_{n+1} \)) 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 \( BOX \) it is useful to measure the learning progress in terms of the number of points in \( X \) 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 \( NW \subset X \) be the set of currently still possible locations of the "north-west corner" \( (i_1,j_1) \) of the (unknown) target box \( C = \{i_1,\ldots,i_d\} \times \{j_1,\ldots,j_d\} \). The sets \( SW, NW, NE, SE \) of still possible locations for the other three corner points of the target box are defined analogously.

Our learning algorithm starts with the hypothesis \( H_0 = \emptyset \). After step \( i \) it constructs a hypothesis \( H := \bigcup_{j=1}^i H_j \), 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 \( L^SW \) that partitions \( NW \) in such a way that on each side of \( L^SW \) there are at least one third of the points of \( NW \). For \( d = 2 \) (the general case is analogous) let \( LC \subset X \) be the set of currently still possible locations of the "north-west corner" \( (i_1,j_1) \) of the (unknown) target box \( C = \{i_1,\ldots,i_d\} \times \{j_1,\ldots,j_d\} \). The sets \( SW, NW, NE, SE \) of still possible locations for the other three corner points of the target box are defined analogously.
The next hypothesis $H$ is then constructed as follows: its left borderline is determined by the rightmost one of the two lines $L^{NW}$, $L^{SE}$, its right borderline by the leftmost one of the lines $L^{SW}$, $L^{NE}$. Analogously the upper borderline of $H$ is determined by the lower one of the two lines $L^{NW}$, $L^{SW}$, and its lower borderline by the higher one of $L^{SW}$, $L^{NE}$.

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 $\frac{1}{3}$ of the points of one of the regions NW, SW, NE, SE.

This concludes the sketch of the proof that $LC(BOX_2)$ is an upper bound to the number of iterations necessary. It is very easy to prove directly that $LC(BOX_2) = O(\log n)$. Alternatively one can use the fact that $\text{chain}(BOX_2) = \Omega(n)$ (see Figure 1 in Section 3), or apply Lemma 7.

A straightforward argument shows that the learning algorithm for $BOX_2$ that we have sketched requires altogether at most $O(\text{poly}(\log n))$ 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 $\text{poly}(\log n)$ boxes).

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 $X \subseteq \mathbb{R}^d$ let

$$HALFSPACE^+_d = \{S \subseteq X \mid S \text{ halfspace in } \mathbb{R}^d \text{ s.t. } X \cap F = S\}.$$!

Instead of $HALFSPACE^+_d$ we write $HALFSPACE^+_d$ if $X = X^+_d = \{1, \ldots, n\}$, and $HALFSPACE^d$ if $X = \{0,1\}^d$. First we present a learning algorithm for this concept class.

**Theorem 2.** There is a learning algorithm $A$ for $HALFSPACE^d$ with $LC(A) = O(d^3 \log d + \log n)$ 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 $z = (z_1, \ldots, z_d) \neq 0$ and a hyperplane $E \subseteq \mathbb{R}^d$ with $\emptyset \neq E$ are dual if $E$ is given by $z_1 x_1 + \ldots + z_d x_d = 1$; we write $E = \text{dual}(z)$, $z = \text{dual}(E)$. If $E$ is a hyperplane with $\emptyset \neq E$, $E^\perp$ is the closed halfspace containing $\emptyset$.

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

Now let $E_i := \text{dual}(z_i)$ and consider the open regions of $\mathbb{R}^d$ 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 $Q_1, \ldots, Q_m$, one from each region s.t. each subset of these points has a centerpoint different from $\emptyset$ and not contained in any of the $E_i$’s.

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

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

**Corollary 6.** $LC(HALFSPACE^d) = O(d^2 \log n)$, in particular $LC(HALFSPACE^d) = O(d^2)$.

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

A technical detail here is that if the center of the ellipsoid $L$ happens to be $\emptyset$, it has to be replaced by another ellipsoid with center $\emptyset \neq \emptyset$ 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 $O(d^2)$ arithmetic operations. The arithmetic operations have to be performed with large precision, but polynomially many digits are sufficient (the theoretical upper bound is $O(d^2 \log d)$ digits). Hence the total computation required is polynomial.

The argument above can be generalized for $HALFSPACE^d$.

**Corollary 3.** There is a learning algorithm $A$ for $HALFSPACE^d$ with $LC(A) = O(d^2 \log d + \log n)$ such that the total computation time required is polynomial in $d$ and $\log n$.

The problem with generalizing this approach to $HALFSPACE^d$ 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 $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$.

$$LC(HALFSPACE^d) = O(d^2 \log n).$$

**Idea of the Proof.** A point $y \in \mathbb{R}^d$ is called a centerpoint of a finite set $Y \subseteq \mathbb{R}^d$ if every open halfspace not containing $y$ contains $\leq \frac{1}{2d^2}|Y|$ points from $Y$.

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

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

Now let $E_i := \text{dual}(z_i)$ and consider the open regions of $\mathbb{R}^d$ 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 $Q_1, \ldots, Q_m$, one from each region s.t. each subset of these points has a centerpoint different from $\emptyset$ and not contained in any of the $E_i$’s.

A decision tree $T$ for $(X, C)$ is a binary tree with inner nodes labelled by elements of $X$, edges labelled by 0 or 1, and leaves labelled by concepts $C \subseteq \mathbb{U}$ s.t.
Let $d_{\text{mm}}(T)$ be the minimal depth of a leaf in $T$ and let $\text{ADV}(C) = \max(d_{\text{mm}}(T) \mid T$ is a decision tree for $C$) be the "adversary complexity" of $C$. One may view $\text{ADV}(C)$ as the "dual decision tree complexity" of $C$.

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

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

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

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

**Theorem 8.** $\text{LC}(\text{HALFSPACE})_s = \Omega(d^2)$. \qed

**Remark.** The preceding results show that the complexity of learning a Boolean threshold function of $d$ variables (with weights of arbitrary sign) 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 $X \subseteq \mathbb{R}^d$ let

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

**Theorem 10.** For every $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$ $\text{LC}(\text{BALL}_X) = O(d^2 \log n)$. For the case $X = \{1, \ldots, n\}^d$ one can give a learning algorithm $A$ for $\text{BALL}_X$ with $\text{LC}(A) = O(d^2 \log d + \log n)$ such that the total computation time of $A$ is polynomial in $d$ and $\log n$.

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

This argument can also be generalized to $\text{HALFSPACE}_s$.

**Corollary 9.** $\text{LC}(\text{HALFSPACE})_s = \Omega(d^2 \log n)$. \qed

**Remark.** The preceding results show that the complexity of learning a Boolean threshold function of $d$ variables (with weights of arbitrary sign) 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 $X \subseteq \mathbb{R}^d$ let

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

**Theorem 10.** For every $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$ $\text{LC}(\text{BALL}_X) = O(d^2 \log n)$. For the case $X = \{1, \ldots, n\}^d$ one can give a learning algorithm $A$ for $\text{BALL}_X$ with $\text{LC}(A) = O(d^2 \log d + \log n)$ such that the total computation time of $A$ is polynomial in $d$ and $\log n$.

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

This argument can also be generalized to $\text{HALFSPACE}_s$.

**Corollary 9.** $\text{LC}(\text{HALFSPACE})_s = \Omega(d^2 \log n)$. \qed

**Remark.** The preceding results show that the complexity of learning a Boolean threshold function of $d$ variables (with weights of arbitrary sign) 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 $X \subseteq \mathbb{R}^d$ let

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

**Theorem 10.** For every $X = \{x_1, \ldots, x_n\} \subseteq \mathbb{R}^d$ $\text{LC}(\text{BALL}_X) = O(d^2 \log n)$. For the case $X = \{1, \ldots, n\}^d$ one can give a learning algorithm $A$ for $\text{BALL}_X$ with $\text{LC}(A) = O(d^2 \log d + \log n)$ such that the total computation time of $A$ is polynomial in $d$ and $\log n$.

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

This argument can also be generalized to $\text{HALFSPACE}_s$.

**Corollary 9.** $\text{LC}(\text{HALFSPACE})_s = \Omega(d^2 \log n)$. \qed

**Remark.** The preceding results show that the complexity of learning a Boolean threshold function of $d$ variables (with weights of arbitrary sign) 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.

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

- $\text{MEMB}(C)$ (this is the "decision tree complexity" of $C$, where the "learner" can only ask membership queries $x \in C$; for $x \in X$).
- $\text{LC} - \text{MEMB}(C)$ (here the learner can ask membership queries and he can present hypotheses from $C$; this is a combination of the learning capabilities of the models for $\text{LC}(C)$ and $\text{MEMB}(C)$).
- $\text{HALVARING}(C)$ (the learner uses a specific algorithm for the $\text{LC} - \text{ARB}$ model: the next hypothesis $H_{z+1}$ is always the set of those points in the domain that lie in at least 50% of all $C \subseteq C$ that are consistent with the first $i$ counterexamples; note that $\text{HALVARING}(C)$ is the same as $M_{\text{HALVARING}}(C)$ in [Li]).
- $\text{LC} - \text{PARTIAL}(C)$ (here the learner can present arbitrary "partial" hypotheses $H \subseteq \{0, 1, \ldots, 4\}^X$; he gets as response either some $x \in X$ or $H(x) \subseteq \{0, 1\}$ and $H(x) \subseteq 1 - C(x)$, where $C \subseteq C$ 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 ([All], [Li]). The last model appears to be of some interest because some 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 $H \subseteq \{0, 1\}^X$ 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 $\text{LC} - \text{PARTIAL}(\text{ADDRESSING}) \leq 1$. 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 $(C) := \max(i \in N \mid \exists G_i \subseteq C, i = 1, \ldots, t$ with $G_i \subseteq G_{i+1})$.

**Remarks to Table 1.**

a) The upper bound for $\text{LC}(\text{LINEAR ORDER})$ 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

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>n</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>LC(C)</strong></td>
<td>n</td>
<td>1</td>
<td>n</td>
<td>log n</td>
<td>n log n</td>
<td>n</td>
<td>n log n</td>
<td>n</td>
</tr>
<tr>
<td>hypotheses</td>
<td>from</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LC-ARB(C)</strong></td>
<td>1</td>
<td>1</td>
<td>log n</td>
<td>log n</td>
<td>log n</td>
<td>n log n</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>(arbitrary hyp.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LC-PARTIAL(C)</strong></td>
<td></td>
<td>1</td>
<td></td>
<td>log n</td>
<td>n</td>
<td>n</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(partial hypotheses)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>LC-MEMBER(C)</strong></td>
<td>n</td>
<td>1</td>
<td>log n</td>
<td>log n</td>
<td>log n</td>
<td>n log n</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>(hyp. ∈ C and</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>MEMBER(C)</strong></td>
<td>n</td>
<td>n</td>
<td>log n</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(membership queries)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>VC-dim(C)</strong></td>
<td>1</td>
<td>1</td>
<td>log n</td>
<td>1</td>
<td>1</td>
<td>n</td>
<td>n</td>
<td></td>
</tr>
<tr>
<td>log(chain(C))</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

All bounds in this table are Θ-bounds. (*exception: MEMBERSHALFSPACE,\textsuperscript{n}D = Θ(n log n)

FIGURE 1

To their hp 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,) follows by considering an adversary which always gives negative counterexamples (the selection of the negative counterexamples 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 n(n^2) hypotheses must have been asked (see [FT] for a similar argument). The lower bound can be generalized to an O(n^k) lower bound for the generalization of this problem to k-uniform hypergraphs using a generalization of Hetyei's result due to Erdős [Er].

c) ADDRESSING, 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(C) ≥ B(C) for every concept class C, and that for some C A(C) is exponentially larger than B(C) (θ(log n) versus θ(1), or θ(n) versus θ(log n)). The number next to the line specifies a class 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 C A(C) is exponentially larger than B(C), and for some other C B(C) is exponentially larger than A(C) (such classes 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 C for which A(C) is exponentially larger.
than \(B(C)\), and that there is some other \(C\) for which \(A(C) \leq \frac{1}{2} B(C)\) (but it is open whether there exists some \(C\) for which \(B(C)\) is exponentially larger than \(A(C)\)).

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 occurring 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

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

and class 6 is

\[
\text{MAJORITY}_n := \left\{ S \subseteq \{0,1,2,\ldots,n\} \mid 0 \in S \Leftrightarrow |S - \{0\}| \geq \left\lfloor \frac{n}{2} \right\rfloor \right\}.
\]

For class 7 we set \(X_n := \{1, \ldots, n + \lceil \log_2 n \rceil\}\), and we define class 7 as

\[
\text{TAGGED SINGLETONS}_n := \{\emptyset\} \cup \{\{i\} \cup (n + \ell) \mid \ell \in \{0, \ldots, \lceil \log_2 n \rceil\}, i \in \{1, \ldots, n\}\}.
\]

It is easy to see that \(LC(\text{TAGGED SINGLETONS}_n) \leq 2\), whereas

\[
LC(\text{HALVING}(\text{TAGGED SINGLETONS}_n)) = 0(\log n).
\]

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

\[
\text{MEMB}(\text{MAJORITY}_n) := \text{MEMB}(\text{MAJORITY}_n) = LC - \text{ARB}(\text{MAJORITY}_n).
\]

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

f) It was already previously known that for every \(C\)

\[
\text{log}(|C| - 1)/\text{log}(|X|) \leq \text{VC - dim}(C) \leq \text{LC} - \text{ARB}(C) \leq \text{LC} - \text{HALVING}(C) \leq \text{log}(|C|) \leq \text{MEMB}(C)\]

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

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

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

Acknowledgement

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

References


