Machine Learning, 9, 107–145 (1992)

Lower Bound Methods and Separation Results for On-Line Learning Models

WOLFGANG MAASS MAASS@IGI.TU-GRAZ.AC.AT
Institute for Theoretical Computer Science, Technische Universität Graz, Klosterwiesgasse 32, A-8010 Graz, Austria, and the University of Illinois at Chicago

GYÖRGY TURÁN U11557@UICVM.BITNET
Department of Mathematics, Statistics and Computer Science, University of Illinois at Chicago, IL, and Automata Theory Research Group of the Hungarian Academy of Sciences, Szeged, Hungary.

Abstract. We consider the complexity of concept learning in various common models for on-line learning, focusing on methods for proving lower bounds to the learning complexity of a concept class. Among others, we consider the model for learning with equivalence and membership queries. For this model we give lower bounds on the number of queries that are needed to learn a concept class $\mathcal{C}$ in terms of the Vapnik-Chervonenkis dimension of $\mathcal{C}$, and in terms of the complexity of learning $\mathcal{C}$ with arbitrary equivalence queries. Furthermore, we survey other known lower bound methods and we exhibit all known relationships between learning complexities in the models considered and some relevant combinatorial parameters. As it turns out, the picture is almost complete. This paper has been written so that it can be read without previous knowledge of Computational Learning Theory.

Keywords. Formal models for learning, learning algorithms, lower bound arguments, VC-dimension, machine learning

1. Introduction

We investigate the complexity of learning in the most common formal models for on-line, or "incremental" learning (as opposed to batch-learning). One may also describe the models considered here as models for learning with queries. We focus on methods for proving lower bounds to the "learning complexity" of a concept class $\mathcal{C}$. In other words, we are interested in proving lower bounds to the number of steps that are needed by *any* learning algorithm in order to learn an arbitrary target concept C_T from the concept class $\mathcal{C}$. In particular, we clarify the relationship between the learning complexities in different learning models and some relevant combinatorial parameters.

In the models considered here, a learning process is viewed as a game between two agents: the *learner* (or *learning algorithm*) and the *environment*. At the beginning both agents agree on a *domain* X (typically a finite set) and a collection $\mathcal{C}$ of subsets of X. In Computational Learning Theory $\mathcal{C}$ is usually referred to as the *concept class*. This terminology is motivated by examples from logic, where $\mathcal{C}$ consists of all subsets of X that are definable by a logical formula of a specified type, e.g., monomials in Boolean logic.

The learning process starts by the environment fixing a *target concept* $C_T \in \mathcal{C}$. The goal of the learner is to *learn* (i.e. to identify) the target concept in as few steps as possible.

The definition of each learning model specifies what is meant by a step. The models considered in this paper are defined in Section 2. Here we only give an outline of the basic model, which is due to Barzdin and Freiwalds (1972), Angluin (1988) and Littlestone (1988). It may be viewed as a machine-independent version of the classical models for learning on perceptrons (Rosenblatt, 1962; Minsky & Papert, 1988; Nilsson, 1965) and neural networks (Rumelhart & McClelland, 1986). In this model the learner probes the environment with queries of the form "$H = C_T$?" for some *hypothesis* $H \in C$ (Angluin, 1988) refers to these queries as *equivalence queries*). If $H = C_T$ then the environment responds "yes." Otherwise the response is a *counterexample* $x \in H \Delta C_T$. The algorithms are *on-line*, i.e. each hypothesis may depend on the previous counterexamples.

The *learning complexity* (or *mistake bound* in Littlestone (1988)) of a learning algorithm is the maximal number of counterexamples it may receive before identifying the target concept C_T, considering all possible responses to the hypotheses and all possible target concepts. The learning complexity of the concept class C is the learning complexity of the best learning algorithm for C.

We would like to point out that this definition of the learning complexity of C is analogous to the common definition of the *computational* complexity of a computational problem. The latter is defined as the least computational complexity of any algorithm solving the problem (where the complexity of an algorithm is determined by a *worst-case* analysis).

As noted above, this notion of a learning process contains the usual notions of a learning process for perceptrons and neural networks as a special case. In this case X is the set of all assignments to the input variables that may occur, and C is the class of all subsets of X that can be computed by the computational device considered, for some setting of its internal parameters (such as weights of edges, thresholds, etc.). In the simplest case of a single threshold gate, i.e. a feedforward neural net without hidden units, with d input bits and with one output bit, we have $C = \text{HALFSPACE}_2^d$, where

$$\text{HALFSPACE}_2^d := \{C \subset \{0, 1\}^d \mid \text{there exist } w_1, \ldots, w_d, t \in \mathbf{R} \text{ such that for every } (x_1, \ldots, x_d) \in \{0, 1\}^d \text{ it holds that } (x_1, \ldots, x_d) \in C \Leftrightarrow \sum_{i=1}^{d} w_i x_i \geq t\}.$$

A hypothesis H from C is in this case the subset of $\{0, 1\}^d$ accepted by the considered threshold gate with its current values $w_1, \ldots, w_d, t$ of weights and threshold. The values remain unchanged until the threshold gate encounters an input x which it processes incorrectly (i.e. $x \in H \Delta C_T$, where C_T is the set accepted by the target threshold gate). For any occurrence of such a counterexample x, the current weights and threshold are changed according to some learning algorithm. The performance of the algorithm is measured by the maximal number of counterexamples, also called mistakes (Littlesone, 1988), that may occur before it converges to C_T. Obviously this coincides with the above definition of learning complexity. We refer to Maass and Turán (1990c) for an account of the known learning algorithms for HALFSPACE_2^d.

One purpose of formal models of learning is to provide a suitable framework for the design and analysis of learning algorithms. The techniques developed in order to design efficient learning algorithms for such formal models may provide a useful contribution of Computational Learning Theory to more application oriented areas such as Machine Learning in Artificial Intelligence.

The task of a concept learning algorithm is to provide a "smart" hypothesis on the basis of the information available. Different formal models of learning give different answers to the question: What distinguishes a "smart" hypothesis from a less intelligent one?

In Valiant's intensively studied model (Valiant, 1984) for *probably approximately correct* learning ("PAC" learning) remarkable results show that a "smart" hypothesis, i.e. a hypothesis used by an optimal learning algorithm in this model, is essentially *any* hypothesis $H \in C$ that is consistent with all preceding examples (such hypotheses are called *consistent hypotheses*), see Blumer, Ehrenfeucht, Haussler and Warmuth (1989). Thus the PAC learning model provides no suitable basis for distinctions among different consistent hypotheses from C (except for issues of *computational complexity*). This observation points to a structural difference between the PAC model and various "natural" learning processes where it is frequently expected that an "intelligent" learner presents more than just *any* consistent hypothesis.

An attempt for defining a "smart" hypothesis is implicitly contained in the on-line learning models considered here. An optimal learning algorithm in these models will issue hypotheses that are not only consistent, but which have the additional property that any counterexample to them eliminates a large number of possible candidates for the target concept (amortized over several learning steps).

The essence of this additional property becomes clear if one examines on-line learning algorithms for learning a subinterval $\{1, \ldots, i\}$ of a fixed discrete domain $\{1, \ldots, n\}$. An optimal on-line learning algorithm for this concept class HALF-INTERVAL$_n$ (see Section 4) outputs hypotheses which are not only consistent, but whose boundary lies halfway between the largest known positive example and the smallest known negative example. In this way the learning algorithm can carry out a binary search for the "boundary" i of the target concept in at most log n steps. On the other hand a learning algorithm that always outputs the "simplest" consistent hypothesis, e.g. the minimal consistent hypothesis, needs up to $n - 1$ steps. Intuitively the first algorithm is "smarter" than the second one, and this can be expressed quantitatively in terms of their different learning complexities (log n for the first algorithm versus $n - 1$ for the second one).

This example illustrates that the models which are considered in this paper provide a framework for making meaningful quantitative distinctions between different consistent learning algorithms which are equivalent from the point of view of PAC-learning. Of course one may turn this argument around noting that it is simpler to design an efficient learning algorithm in the PAC model.

Because of structural differences between the PAC model and the models considered here, it is not possible to compare directly the efficiency of optimal learning algorithms in the two types of models. (In the PAC model the learner receives examples rather than counterexamples, and he is only required to output an ϵ-approximation of C_T with confidence $\geq 1 - \delta$). If one ignores these essential differences and nevertheless compares the number of examples required by an (ϵ, δ) PAC-learning algorithm with the number of counterexamples required for 100% correct learning, then it turns out that for some interesting concept classes the two numbers are in the same range already for not too small values of ϵ and δ. We refer to Angluin (1988) for further discussion of the relation between the two types of models.

A possible advantage of fast learning algorithms in the on-line learning models considered here is the fact that they are also guaranteed to perform well in a situation where the environment cannot be adequately modeled by a time-invariant probability distribution (as it is required for PAC learning). For example, this is the case if the environment consists of sensory inputs received by a moving robot; if an optical character recognition machine is trained on the handwritings of different persons successively; or if a speech recognition machine is trained by different speakers successively.

Finally we would like to point out that the on-line learning models provide a useful "yardstick" for evaluating the performance of various concrete learning algorithms for specific "learning machines" such as perceptrons or neural networks. In these models one usually considers only learning algorithms which generate the next hypothesis with severely limited resources and without a global control (such as the Δ-rule or backwards propagation). It is obviously of interest to find out how seriously various machine-dependent restrictions affect the efficiency of learning in comparison with the theoretically fastest possible on-line learning algorithm for the same concept class. We refer to Maass and Turán (1990a; 1990c) for some comparisons of this type for the case of a perceptron.

The remainder of this paper is organized as follows. The learning models considered are introduced in Section 2. Section 3 defines the combinatorial parameters needed later on. In Section 4 we discuss some basic methods for proving lower bounds to the complexity of learning with equivalence queries and arbitrary equivalence queries. Section 5 surveys relationships between learning with arbitrary equivalence queries, decision trees and the halving algorithm. In Section 6, which contains the main results of this paper, we discuss the relationship between learning with equivalence and membership queries, learning with arbitrary equivalence queries and the Vapnik-Chervonenkis dimension. Section 7 gives a further discussion of learning models allowing membership queries. In Section 8 we investigate the power of learning with partial hypotheses. Throughout these sections we introduce and discuss a few concrete concept classes which turn out to be useful "benchmarks" for the evaluation of different learning models. The results for these concept classes are summarized in Table 1 of Section 9. In this section we also display in Figure 1 the known relationships between learning complexities and combinatorial parameters considered in the previous sections. Finally in Section 10 we mention some open problems.

This paper contains detailed proofs for several results that were previously announced in Maass and Turán (1989; 1990a). Proofs for the other results announced in these extended abstracts will appear in Maass and Turán (1990b; 1990c). For results concerning randomized learning algorithms in on-line learning models we refer to Maass (1991).

2. Learning models

A learning problem is specified by a *domain* X and a *concept class* $C \subset 2^X$. In this paper X is always a finite set, with the exception of the domain for learning DFA mentioned in

Section 10. Later we will usually consider sequences $(X_n)_{n \in \mathbf{N}}$ of domains and associated sequences of concept classes with $\mathcal{C}_n \subset 2^{X_n}$. The parameter n serves here as a measure for the size of the domain X_n (and implicitly also for the size of the concept class $\mathcal{C}_n$).

First we describe the model of *learning with equivalence queries* outlined in the introduction. A learning process starts with the *environment* fixing a *target concept* $C_T \in \mathcal{C}$.

The *learner* (or *learning algorithm*) proposes *hypotheses* H from $\mathcal{C}$ (or *equivalence queries* "$H = C_T$?"; we use the two terms interchangeably). If $H = C_T$, the environment responds "yes." Otherwise it responds with a *counterexample* x from the symmetric difference $H \Delta C_T := (C_T \backslash H) \cup (H \backslash C_T)$. Viewing H and C_T as functions from X to $\{0, 1\}$, we also use the notation $H(x) \neq C_T(x)$. If $x \in C_T \backslash H$ then it is called a *positive* counterexample, if $x \in H \backslash C_T$ then it is called a *negative* counterexample.

Thus a learner (or learning algorithm) for $\mathcal{C}$ is any algorithm A which produces new hypotheses

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

in dependence on the previous hypotheses H_j^A and the counterexamples $x_j \in H_j^A \Delta C_T$ received. Since in this paper we only consider *deterministic* algorithms, we may suppress the hypotheses H_j^A as arguments of A.

The *learning complexity* $\mathrm{LC}(A)$ of such a learning algorithm A is

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

Note that in the definition of $\mathrm{LC}(A)$ the amount of computation performed by A to determine the next hypothesis is not taken into account; attention is focused on the *amount of interaction* between the learner and the environment.

The *learning complexity* $\mathrm{LC}(\mathcal{C})$ of the concept class $\mathcal{C}$ is

$$\mathrm{LC}(\mathcal{C}) := \min\{\mathrm{LC}(A) | A \text{ is a learning algorithm for } \mathcal{C}\}.$$

In the preceding definition of a learning process we assumed that the *hypotheses space* $\mathcal{H}$, i.e. the space from which the hypotheses H used in the equivalence queries "$H = C_T$?" are drawn, coincides with the concept class $\mathcal{C}$ of all possible target concepts. This of course need not be the case in general. For any class $\mathcal{H}$ with $\mathcal{C} \subseteq \mathcal{H} \subseteq 2^X$ one can define

$$\mathrm{LC}^{\mathcal{H}}(\mathcal{C}) := \min\{\mathrm{LC}(A) | \ A \text{ is a learning algorithm for } \mathcal{C} \text{ using equivalence queries with hypotheses from } \mathcal{H}\}.$$

Of particular interest is the case $\mathcal{H} = 2^X$, where *arbitrary* subsets of X may be used as hypotheses. We set

$$\mathrm{LC} - \mathrm{ARB}(\mathcal{C}) := \mathrm{LC}^{2^X}(\mathcal{C}).$$

This learning complexity measure is discussed in Sections 4 and 5.

Littlestone (1988) has provided an alternative interpretation of this learning model which does not refer to queries. He assumes that the environment provides an arbitrary sequence $x_1, x_2, \ldots$ of elements of X as *examples* for the target concept $C_T \in \mathcal{C}$. For each example x_i the learner has to *predict* whether $x_i \in C_T$. At each step one may view the set of all $x \in X$ for which the learner would currently predict that $x \in C_T$ as the current *hypothesis* H of the learner. After each prediction for an element x_i the learner is told whether $x_i \in C_T$. If his prediction was incorrect (i.e. $x_i \in \mathcal{H} \Delta C_T$) one says that the learner has made a *mistake*. After each mistake the learner may change his hypothesis $\mathcal{H}$. The goal of the learner is to make as few mistakes as possible. It is easy to see (Littleston, 1988) that the associated optimal mistake bound opt($\mathcal{C}$) for a concept class $\mathcal{C}$ agrees with LC $-$ ARB($\mathcal{C}$) (respectively LC($\mathcal{C}$), if one demands that each hypothesis $\mathcal{H}$ of the prediction algorithm belongs to $\mathcal{C}$). This results from the worst case analysis in both models.

A general issue in computational learning theory concerns the power of *carrying out experiments*. For which concept classes is it possible to learn substantially faster if the learner can also probe the environment with queries of the form "$x \in C_T$?" for $x \in X$, in addition to his other queries? We assume that the environment provides the correct answer to every such *membership query*. For a learning algorithm A that may use both equivalence queries and membership queries we write LC(A) for the maximal number of counterexamples and membership queries needed until the target concept C_T is identified (for any choice of $C_T \in \mathcal{C}$ and any choice of the counterexamples to the equivalence queries of A). We set

$$\text{LC} - \text{MEMB}(\mathcal{C}) := \min\{\text{LC}(A) \mid A \text{ is a learning algorithm for } \mathcal{C} \text{ that uses equivalence queries with hypotheses from } \mathcal{C} \text{ and membership queries}\}.$$

We also consider the restricted model where the learner can ask membership queries *only*. The learning complexity of an algorithm A in this model is defined analogously to the previous definitions and

$$\text{MEMB}(\mathcal{C})$$

denotes the complexity of the concept class $\mathcal{C}$ when only membership queries may be used.

Concerning our notation we would like to point out that we always write LC(A) for the maximal number of learning steps needed by a learning algorithm A, no matter which type of queries are used by A. However when we talk about the learning complexity of a *concept class* $\mathcal{C}$, we make explicit in the notation (LC($\mathcal{C}$), LC-ARB($\mathcal{C}$), LC-MEMB($\mathcal{C}$), etc.) which types of learning algorithms are considered.

It has turned out that there are several important concept classes for which one can design efficient learning algorithms using equivalence and membership queries. For several of these classes it is also known that equivalence, resp. membership queries alone are not sufficient for efficient learning. The list of these concept classes includes DFA (Angluin, 1987a; 1990), one-counter languages (Berman & Roos, 1987), simple deterministic languages (Ishizaka, 1990), this algorithm uses extended equivalence queries), k-term DNF (Angluin, 1987a; Pitt & Valiant, 1988), read-once formulas (Angluin, Hellerstein & Karpinski, 1989), conjunctions of Horn clauses (Angluin, Frazier & Pitt, 1990) and intersections of halfspaces (Baum, 1990; Bultman & Maass, 1990). We note that in the first three examples the models considered also take into account the lengths of the counterexamples received, resp. the amount of computation performed by the learning algorithm (see Section 10 for further details).

The exact power of the LC-MEMB model remained somewhat elusive because previously there has been no method available to prove lower bounds. In Section 6 we show that LC-MEMB(C) $= \Omega$(VC-dim(C)) for every concept class C, where C is the Vapnik-Chervonenkis dimension of C (see the definition in the next section). As an application of this lower bound we determine LC-MEMB($C_{k,n}$) for the class of conjunctions of k literals from n variables.

In Section 6 we also establish a somewhat unexpected relationship between LC-MEMB(C) and LC-ARB(C). It turns out that this relationship can also be used to prove lower bounds to LC-MEMB(C) for certain concept classes C, e.g. for C = HALFSPACE$^d_2$.

The lower bounds derived for LC-MEMB(C) remain in fact valid if the learner is allowed to use equivalence queries "$H = C_T$?" with *arbitrary* subsets $H \subseteq X$ as hypotheses, in addition to membership queries. We write

LC-ARB-MEMB(C)

for the learning complexity of C in this learning model.

In Section 8 we discuss a new model for on-line learning, where the learner is more powerful than in any of the preceding models. In this case the learner probes the environment with hypotheses $H \in \{0, 1, *\}^X$ called *partial hypotheses*. Unlike in the models above (where $H \in \{0, 1\}^X$), here the learner may also assign the "don't care" symbol $*$ to some elements $x \in X$, meaning that currently he is not interested in their membership in C_T. The environment is obliged to respond to such a query either with a counterexample, i.e. with an element $x \in X$ such that $H(x) \in \{0, 1\}$ and $H(x) \neq C_T(x)$, or with the reply "correct" if there is no such counterexample. Note that a partial hypothesis assigning 1 to a single $x \in X$ and $*$ to all other elements is equivalent to a membership query, thus the learner is indeed at least as powerful in this model than in all those introduced above. Analogously as before, if A is a learning algorithm using partial hypotheses then LC(A) denotes the worst case number of queries required before the target concept is identified and

$$\text{LC-PARTIAL}(C) := \min\{\text{LC}(A) \mid A \text{ is a learning algorithm for } C \text{ using partial hypotheses}\}.$$

Learning with partial hypotheses can be of interest because it allows the learner to focus attention on a specific subset of the domain X. One may argue that this ability plays a significant role in human learning, where a typical "hypothesis" does not assign a truth value to every possible yes/no decision in the world. In Section 8 we will show that the ability to use partial hypotheses makes the learner substantially more powerful in our formal model of on-line learning.

3. Combinatorial parameters

In this paper we will also compare the different learning complexities of a concept class C with various combinatorial parameters of C, mainly for the purpose of finding lower bounds.

If C is a concept class over the domain X and Y is a subset of X, then the concept class $C \cap Y$ *induced by C on Y* is

$$C \cap Y := \{C \cap Y \mid C \in C\}. \tag{1}$$

Y is *shattered* by C if $C \cap Y = 2^Y$, where 2^Y is defined as the class of all subsets of Y.

The *Vapnik-Chervonenkis dimension* of C (denoted by VC-dim(C)) is

$$\text{VC-dim}(C) := \max\{|Y| \mid Y \subset X \text{ is shattered by } C\}.$$

It has turned out that VC-dim(C) is the key parameter determining the number of samples needed for learning C in Valiant's model (Valiant, 1984) for PAC learning (Blumer, Ehrenfeucht, Haussler & Warmuth, 1989). The basic result concerning the Vapnik-Chervonenkis dimension is the following.

Lemma 3.1. (Sauer (1972), Perles and Shelah, see Shelah (1972), Vapnik and Chervonenkis (1971)). If C is a concept class over the domain X with VC-dim(C) $= d$ then

$$|C| \leq \sum_{i=0}^{d} \binom{|X|}{i}. \qquad \square$$

We also consider another combinatorial parameter of a concept class C, the maximal size of a chain in C under inclusion:

$$\text{chain}(C) := \max\{\ell \in \mathrm{N} \mid \text{there are concepts } C_1, \ldots, C_\ell \in C \text{ with } C_1 \subsetneq C_2 \subsetneq \ldots \subsetneq C_\ell\}.$$

This parameter is useful as $\lfloor \log_2(\text{chain}(C)) \rfloor$ provides a lower bound to the learning complexity of C in all models introduced in Section 2 except the model allowing partial hypotheses. It provides *optimal* lower bounds up to a constant factor for several interesting concept classes with VC-dim(C) $\ll \log_2(\text{chain}(C))$, such as classes of geometrical objects (boxes, balls, etc.) in a d-dimensional discrete space. (In the sequel $\log_2 x$ will be written as $\log x$). Furthermore, as it will be shown in Section 8, $\log_3(\text{chain}(C))$ is a lower bound to LC-PARTIAL(C) as well, thus it can be used as a lower bound for *all* models considered in this paper.

In addition to VC-dim(C) and $\log(\text{chain}(C))$, we also discuss the role of $\log(|C|)$ and $\log(|C| - 1)/\log(|X| + 1)$. The latter is used to unify the slightly larger values $\log(|C| - 1)/\log|X|$ in Proposition 3.2 and $\log|C|/\log(|X| + 1)$ in Proposition 8.2. We note the following relationship between these parameters.

Proposition 3.2. If C is a concept class over the domain X with $|C| > 1$, $|X| > 1$ then

$$\log |C| \geq \text{VC-dim}(C) \geq \frac{\log(|C| - 1)}{\log |X|}.$$

Proof. The first inequality follows from the definition of the Vapnik-Chervonenkis dimension: if $Y \subseteq X$ is shattered by C then $|C| \geq 2^{|Y|}$. As noted in Blumer, Ehrenfeucht, Haussler and Warmuth (1989), Lemma 3.1 implies $|C| \leq |X|^{\text{VC-dim}(C)} + 1$, which in turn implies the second inequality. □

4. Some elementary methods for proving lower bounds for learning with equivalence queries and learning with arbitrary equivalence queries

In this section we describe some simple examples illustrating learning with equivalence queries and learning with arbitrary equivalence queries, and discuss adversary strategies for proving lower bounds to learning complexity in these models.

First we introduce the following concept classes over the domain $X_n = \{1, \ldots, n\}$:

$$\begin{aligned} \text{SINGLETON}_n &:= \{\{i\} \mid i \in \{1, \ldots, n\}\}, \\ \text{HALF-INTERVAL}_n &:= \{\{1, 2, \ldots, i\} \mid i \in \{1, \ldots, n\}\}, \\ \text{POWER-SET}_n &:= \{C \mid C \subset \{1, \ldots, n\}\}. \end{aligned}$$

These three concept classes are studied in the context of learning not so much for their intrinsic interest, but rather because they play an important role in the analysis of the learning complexity of *other*, more important concept classes. Almost any known proof of a lower bound for LC(C) (or LC-ARB(C), etc.) for a concrete concept class C proceeds by showing that for a sufficiently large n one of these three classes C_n is *embedded* into C and that this implies that C is at least as difficult to learn as C_n. Hence these three classes may be viewed as prototypes for three main sources of "learning difficulty" for on-line learning.

For each of these three classes one proves an optimal lower bound for LC(C_n) and LC-ARB(C_n) by a different, but very simple adversary strategy, outlined below for completeness.

Proposition 4.1. (Angluin, 1988).

a) LC(SINGLETON$_n$) $= n - 1$,
b) LC-ARB(SINGLETON$_n$) $= 1$.

Proof. a) The upper bound follows from the general fact that LC(C) $\leq |C| - 1$ for every concept class C (use any learning algorithm that carries out an *exhaustive search* through C, i.e. which uses all $C \in C$ as hypotheses in some order).

The lower bound is proved by the following adversary strategy: for each hypothesis $H = \{j\}$ give j as a negative counterexample. After this adversary strategy has been played for $n - 2$ steps, there are still at least two different concepts in SINGLETON$_n$ that are consistent with all preceding counterexamples. Hence the learner needs at least one further counterexample before he can identify the target concept.

(b) In order to prove the upper bound one uses $H_1 := \emptyset$ as the first hypothesis. It is obvious that 1 is also a lower bound. □

We note that formally the environment is required to choose a target concept at the beginning of the learning process. However it is equivalent (and more useful for the design of adversary strategies) to assume that at any step of the learning process the environment has not determined yet which among those $C \in \mathcal{C}$ that are consistent with all preceding counterexamples, is the one that will serve as its target concept. Hence the learning process is not completed as long as more than one concept $C \in \mathcal{C}$ is consistent with all preceding counterexamples.

It is interesting to compare SINGLETON$_n$ with the concept class SINGLETON$_n \cup \{\emptyset\}$. For this concept class clearly LC(SINGLETON$_n$ $\cup$ $\{\emptyset\}$) = 1 and hence LC$^{\text{SINGLETON}_n \cup \{\emptyset\}}$ (SINGLETON$_n$) = 1. Thus by allowing a single further subset of the domain (the empty set) as hypothesis, the learning complexity of SINGLETON$_n$ drops from $n - 1$ to 1. This curious "instability" of the learning complexity of SINGLETON$_n$ distinguishes it from the other two concept classes HALF-INTERVAL$_n$ and POWER-SET$_n$ for which we have LC($\mathcal{C}_n$) = LC-ARB($\mathcal{C}_n$) (see Proposition 4.2 and 4.3 below), and hence LC$^{\mathcal{H}_n}$($\mathcal{C}_n$) = LC($\mathcal{C}_n$) for every hypothesis class $\mathcal{H}_n$ with $\mathcal{C}_n \subset \mathcal{H}_n \subset 2^{\{1,\ldots,n\}}$.

Although SINGLETON$_n$ becomes easy to learn if the empty set may be used as a hypothesis, there are several interesting concept classes $\mathcal{C}$ with $\emptyset \in \mathcal{C}$ for which one can prove a lower bound for LC($\mathcal{C}$) by identifying an embedded version of SINGLETON$_n$ (without $\emptyset$) in $\mathcal{C}$, as noted by Angluin (1990). As examples we refer to Angluin's lower bounds for DFA, NFA and classes of Boolean formulas (Angluin, 1990); she calls this approach the method of *approximate fingerprints*), and the lower bounds for boxes in general position and for intersections of two halfspaces over the domain $\{1, \ldots, n\}^2$ (Maass & Turán, 1990a; 1990b; 1990c).

All known proofs of a lower bound for LC($\mathcal{C}$) which is larger than LC-ARB($\mathcal{C}$), for some concrete concept class $\mathcal{C}$, proceed essentially by identifying an embedding of SINGLETON$_n$ in $\mathcal{C}$. This is understandable in view of the previous remark that among the three concept classes considered, this is the only one which separates LC and LC-ARB.

Proposition 4.2. LC(HALF-INTERVAL$_n$) = LC-ARB(HALF-INTERVAL$_n$) = $\lfloor \log n \rfloor$.

Proof. The upper bound is proved by constructing a learning algorithm A using equivalence queries from HALF-INTERVAL$_n$ such that LC(A) $\leq$ $\lfloor \log n \rfloor$. Assume that after i learning steps the largest number known to be in C_T is u, and the largest number not excluded yet from C_T is v. Set $H_{i+1}^A := \{1, \ldots, \lfloor (u+v)/2 \rfloor\}$. Before the i + 1'st hypothesis there are $v - u + 1$ candidates for the target concept. If a positive counterexample is obtained then the number of remaining candidates is at most $v - \lfloor (u+v)/2 \rfloor \leq (v - u + 1)/2$.

If a negative counterexample is obtained then the number of remaining candidates is at most $\lfloor (u + v)/2 \rfloor - u \leq (v - u + 1)/2$. Hence after $\lfloor \log n \rfloor$ hypotheses there is at most one candidate left.

LC-ARB(HALF-INTERVAL$_n$) $\geq \lfloor \log n \rfloor$ is proved by giving an adversary strategy. The adversary gives the counterexamples in such a way that after i counterexamples there is an interval of at least $\lfloor n/2^i \rfloor$ elements such that each element can occur as the right endpoint of the target concept (initially this interval is $\{1, \ldots, n\}$). Let the interval after the i-th hypothesis be $\{u, \ldots, v\}$ and consider the $i + 1$'st hypothesis H^A_{i+1}. Let the right endpoint of H_{i+1} be x, we may assume w.l.o.g. $u \leq x \leq v$. If x is given as a negative counterexample then the $x - u$ elements in the interval $\{u,, \ldots, x - 1\}$ can still be the right endpoints of the target concept. If $x + 1$ is given as a positive counterexample then the $v - x$ elements in the interval $\{x + 1, \ldots, v\}$ can still be the right endpoints of the target concept. As $(v - x) + (x - u) = v - u$, the adversary can select the counterexample so that there remains an interval of at least $\lceil (v - u)/2 \rceil$ candidates for being the right endpoints of the target concept. As

$$\left\lceil \frac{v - u}{2} \right\rceil = \left\lfloor \frac{v - u + 1}{2} \right\rfloor \geq \left\lfloor \frac{1}{2} \left\lfloor \frac{n}{2^i} \right\rfloor \right\rfloor = \left\lfloor \frac{n}{2^{i+1}} \right\rfloor ,$$

the claim is proved. This implies that after less than $\lfloor \log n \rfloor$ hypotheses there are at least two candidates left for being the target concept and the learning process cannot be concluded yet. □

We note that the same argument shows that LC(HALF-INTERVAL$_n$ $\cup$ $\{\emptyset\}$) $= \lfloor \log(n + 1) \rfloor$. It can also be remarked here that log(chain(HALF-INTERVAL$_n$)) $=$ log n, while VC-dim(HALF-INTERVAL$_n$) $= 1$ and log($|$HALF-INTERVAL$_n|$$-1$)/log($n + 1$) < 1. Thus for this concept class the first combinatorial parameter is much larger than the other two.

Proposition 4.3. (Folklore). LC(POWER-SET$_n$) $=$ LC-ARB(POWER-SET$_n$) $= n$.

Proof. It is obvious that LC and LC-ARB coincide for this concept class. The upper bound follows from the trivial fact that LC($\mathcal{C}$) $\leq |X|$ for any concept class $\mathcal{C}$ over the domain X (use any consistent learning algorithm). For the lower bound consider the adversary which gives element i as a counterexample to the i-th hypothesis H_i. Then after $n - 1$ learning steps there are still two candidates left for being the target concept and one more hypothesis is needed. □

If $\mathcal{C}_1$, $\mathcal{C}_2$ are concept classes over the same domain X with $\mathcal{C}_1 \subset \mathcal{C}_2$ then clearly LC-ARB($\mathcal{C}_1$) $\leq$ LC-ARB($\mathcal{C}_2$). However the examples of SINGLETON$_n$ and SINGLETON$_n$ $\cup$ $\{\emptyset\}$ show that it may be the case that LC($\mathcal{C}_1$) $\gg$ LC($\mathcal{C}_2$). Thus from set-theoretic relationships between two concept classes one cannot infer a relationship between their learning complexity in general. The following lemma exhibits conditions which imply monotonicity for both LC and LC-ARB. This lemma is useful in order to prove a lower bound for a concept class $\mathcal{C}_2$ in which one can identify an embedded copy of another concept class $\mathcal{C}_1$ which is known to have high learning complexity.

Lemma 4.4. (Monotonicity Lemma.)

a) Assume that $X_1 \subseteq X_2$, $\mathcal{C}_1 \subseteq 2^{X_1}$, $\mathcal{C}_2 \subseteq 2^{X_2}$ and $\mathcal{C}_1 \subseteq \mathcal{C}_2 \cap X_1$. Then LC-ARB($\mathcal{C}_1$) $\leq$ LC-ARB($\mathcal{C}_2$).

b) If in addition $\mathcal{C}_1 = \mathcal{C}_2 \cap X_1$ then LC($\mathcal{C}_1$) $\leq$ LC($\mathcal{C}_2$).

Proof. We prove b) first. Let A_2 be an optimal learning algorithm for $\mathcal{C}_2$. Thus it holds that LC(A_2) = LC($\mathcal{C}_2$) and by definition A_2 uses hypotheses from $\mathcal{C}_2$. We construct a learning algorithm A_1 for $\mathcal{C}_1$. The first hypothesis of A_1 is $H_1^{A_1} := H_1^{A_2} \cap X_1$; note that $H_1^{A_1} \in \mathcal{C}_1$ as we assumed $\mathcal{C}_1 = \mathcal{C}_2 \cap X_1$. If A_1 receives a counterexample $x_1 \in X_1$ such that $x_1 \in H_1^{A_1} \Delta C_T$, then as $C_T = C \cap X_1$ for some $C \in \mathcal{C}_2$, x_1 may also be viewed as a counterexample to the hypothesis $H_1^{A_2}$, having C as the target concept. Continuing similarly assume that we defined $H_j^{A_1} := H_j^{A_2} \cap X_1$ for $j = 1, \ldots, i-1$. Then the counterexamples $x_j \in X_1$, $x_j \in H_j^{A_1} \Delta C_T$ ($j = 1, \ldots, i-1$) may be viewed as counterexamples for the hypotheses of A_2, having C as the target concept. Then set $H_i^{A_1} := H_i^{A_2} \cap X_1$, where $H_i^{A_2} = A_2(x_1, \ldots, x_{i-1})$. We claim that A_1 is a learning algorithm for $\mathcal{C}_1$ with LC(A_1) $\leq$ LC(A_2), implying LC($\mathcal{C}_1$) $\leq$ LC($\mathcal{C}_2$). Assume that A_1 receives counterexamples $x_1, \ldots, x_\ell$ and $\ell >$ LC(A_2) for some target concept C_T. Then as noted, these counterexamples will also occur in a learning process of A_2 for some target concept C with $C_T = C \cap X_1$, which is a contradiction.

The proof of part a) is similar except that in this case the weaker assumption $\mathcal{C}_1 \subseteq \mathcal{C}_2 \cap X_1$ suffices as we do not need $H_i^{A_1} \in \mathcal{C}_1$. □

Using the Monotonicity Lemma one can draw the following conclusions from Propositions 4.2 and 4.3.

Proposition 4.5. For every concept class $\mathcal{C}$

$$\text{LC-ARB}(\mathcal{C}) \geq \lfloor \log(\text{chain}(\mathcal{C})) \rfloor .$$

Proof. Let $C_1 \subset \ldots \subset C_\ell$ be a longest chain in $\mathcal{C}$. Assume first that $C_1 \neq \emptyset$. Fix some $y_1 \in C_1$ and $y_i \in C_i \setminus C_{i-1}$ for $i = 2, \ldots, \ell$. Set $X_1 := \{y_1, \ldots, y_\ell\}$ and $\mathcal{C}_1 := \{\{y_1, \ldots, y_i\} \mid i = 1, \ldots, \ell\}$. Then $\mathcal{C}_1 \subset \mathcal{C} \cap X_1$ and hence from Proposition 4.2 and the Monotonicity Lemma LC-ARB($\mathcal{C}$) $\geq$ LC-ARB($\mathcal{C}_1$) $\geq \lfloor \log \ell \rfloor = \lfloor \log(\text{chain}(\mathcal{C})) \rfloor$. If $C_1 = \emptyset$ then one can argue similarly by letting $\mathcal{C}_1 := \{\{y_2, \ldots, y_i\} \mid i = 2, \ldots, \ell\} \cup \{\emptyset\}$ and referring to the remark following Proposition 4.2 on HALF-INTERVAL$_{\ell-1}$ $\cup \{\emptyset\}$. □

For the concept class POWER-SET$_n$ it holds that LC-ARB(POWER-SET$_n$) = n and log(chain(POWER-SET$_n$)) = log($n + 1$), thus the lower bound of Proposition 4.5 is far from being sharp.

Proposition 4.6. (Littlestone, 1988). For every concept class $\mathcal{C}$

$$\text{LC-ARB}(\mathcal{C}) \geq \text{VC-dim}(\mathcal{C}).$$

Proof. Let Y be a shattered subset of the domain of $\mathcal{C}$ with $|Y| =$ VC-dim($\mathcal{C}$). Set $\mathcal{C}_1 := 2^Y$. Then from Proposition 4.3 and the Monotonicity Lemma

$$\text{LC-ARB}(\mathcal{C}) \geq \text{LC-ARB}(\mathcal{C}_1) = |Y| = \text{VC-dim}(\mathcal{C}). \qquad \square$$

Here one can use HALF-INTERVAL$_n$ as an example to show that the lower bound of Proposition 4.6 is not sharp in general. Indeed, we have seen that LC-ARB(HALF-INTERVAL$_n$) = $\lfloor \log n \rfloor$ and VC-dim(HALF-INTERVAL$_n$) = 1.

We note that both propositions can be proved directly by applying the simple adversary arguments of Propositions 4.2 and 4.3. The presentation above is intended to emphasize the approach of proving a lower bound to learning complexity by finding an embedded "difficult" concept class.

5. Arbitrary equivalence queries, adversary trees and the halving algorithm

In this section we review a method due to Littlestone (1988) for proving lower bounds to LC-ARB($\mathcal{C}$). This method proceeds in a different fashion than the previously discussed ones. The problem of proving a lower bound for LC-ARB($\mathcal{C}$) is reduced to the construction of a decision tree for $\mathcal{C}$ in which every leaf has large depth. In the second part of this section we discuss the relationship between LC-ARB($\mathcal{C}$) and the speed of the halving algorithm for $\mathcal{C}$.

A rooted binary tree T is called a *decision tree* for a concept class $\mathcal{C}$ over a domain X if

- each inner node is labelled by an element x of X (this label represents a query "$x \in C_T$?"),
- the two edges leaving any inner node are labeled "yes" and "no" (these labels correspond to the possible answers to the membership query asked at the node),
- each leaf is labeled by a concept $C \in \mathcal{C}$ in such a way that each $C \in \mathcal{C}$ occurs as the label of exactly one leaf, and the label C of any leaf is consistent with all labels along the path leading from the root to this leaf.

Decision trees form a convenient representation of learning algorithms using membership queries. In fact, MEMB($\mathcal{C}$) could have been defined as the smallest possible depth of a decision tree for $\mathcal{C}$. Here we consider another, related measure of complexity.

For any concept class $\mathcal{C}$ over a domain X we set

$$\text{ADV}(\mathcal{C}) := \max\{\ell \in \mathbf{N} \mid \text{there is a decision tree } T \text{ for } \mathcal{C} \text{ such that every leaf of } T \text{ has depth} \geq \ell\}.$$

The abbreviation ADV is motivated by the fact that these trees, introduced by Littlestone (1988) using a somewhat different notation, can be used to construct an *adversary* for a learning algorithm.

Let us introduce the following notation. If $\mathcal{C}$ is a concept class over a domain X and x is an element of X then the subclass of $\mathcal{C}$ *containing* x is

$$\mathcal{C}_x := \{C \in \mathcal{C} \mid x \in C\} \tag{2}$$

and the subclass of C *not containing* x is

$$C_{-x} := \{C \in C \mid x \notin C\}. \tag{3}$$

Proposition 5.1. (Littlestone, 1988). For every concept class C

$$\text{LC-ARB}(C) = \text{ADV}(C).$$

Proof. In order to show that LC-ARB(C) $\geq$ ADV(C), it is sufficient to observe that every decision tree T for C can be used to force every learning algorithm using arbitrary hypotheses to ask at least d hypotheses before identifying the target concept, where d is the minimum of the depths of the leaves of T.

The adversary starts at the root and moves down the tree, always giving the label x of the current node as a counterexample. If x was a positive (resp. negative) counterexample, he moves along the edge labelled "yes" (resp. "no"). As all concepts occurring as labels of leaves below the current node are always consistent with the previous counterexamples, the learning process has to continue until a leaf is reached.

The proof of the other direction is based on the observation that for every element x of the domain X it holds that

$$\min(\text{ADV}(C_x), \text{ADV}(C_{-x})) < \text{ADV}(C).$$

Indeed, if T_x (resp. T_{-x}) is a decision tree for C_x (resp. C_{-x}), then the tree T formed by putting x into the root and adding T_x (resp. T_{-x}) as left (resp. right) subtree is a decision tree for C, and the depth of each leaf in T is larger than the minimum of the depths of the leaves in T_x and T_{-x}.

Now LC-ARB(C) $\leq$ *ADV(C)* is proved by constructing a learning algorithm, by induction on ADV(C). The case ADV(C) $= 0$ is trivial. The first hypothesis of the learning algorithm in the case ADV(C) > 0 is

$$H_1 := \{x \in X \mid \text{ADV}(C_x) \geq \text{ADV}(C_{-x})\}.$$

If x_1 is a negative counterexample to H_1, then $x_1 \in H_1 \backslash C_T$, thus $C_T \in C_{-x_1}$. From the above observation ADV(C_{-x_1}) $<$ ADV(C), thus by induction we can continue with a learning algorithm using at most ADV(C_{-x_1}) $\leq$ ADV(C) $- 1$ hypotheses before identifying the target concept. Hence in this case the algorithm needs at most ADV(C) counterexamples altogether. If x_1 is a positive counterexample then the analogous argument works by considering C_{x_1}. □

Proposition 5.1 shows an interesting *dual* relationship between MEMB(C) and LC-ARB(C). MEMB(C) is the *minimum* of the *maximal* depth of the leaves of T, with the minimum taken over all decision trees T for C. LC-ARB(C) is the *maximum* of the *minimal* depth of the leaves of T, with the maximum taken over all decision trees T for C.

Proposition 5.1 can be used to prove an optimal $\Omega(d^2)$ lower bound for LC-ARB (HALFSPACE$_2^d$), where HALFSPACE$_2^d$ is the class of concepts over $\{0, 1\}^d$ computable by a Boolean threshold gate with d input bits, formally defined in Section 1 (see (Maass & Turán, 1989; 1990c)). We are not aware of any other methods that would give any lower bound for LC(HALFSPACE$_2^d$) which is superlinear in d.

When discussing learning algorithms using arbitrary equivalence queries, there is one algorithm which deserves particular attention: the *halving algorithm,* or *majority vote strategy* (see (Angluin, 1988; Littlestone, 1988)).

The halving algorithm HALVING$_C$ for a concept class C over a domain X works as follows. At any step $i + 1$ $(i \geq 0)$ let C_i be the class of all concepts $C \in C$ which are consistent with the first i counterexamples. Then the next hypothesis consists of those elements which are contained in at least half of the concepts from C_i, i.e. using the notation introduced preceding Proposition 5.1

$$H_{i+1} := \{x \in X \mid |(C_i)_x| \geq |(C_i)_{-x}|\}.$$

It is obvious that for any counterexample x_{i+1} to H_{i+1} one gets $|C_{i+1}| \leq |C_i|/2$. This implies that

$$\text{LC}(\text{HALVING}_C) \leq \lfloor \log |C| \rfloor \tag{4}$$

and in particular

$$\text{LC-ARB}(C) \leq \lfloor \log |C| \rfloor \tag{5}$$

for every concept class C.

The example of SINGLETON$_n$ shows that these bounds can also be far from being sharp as LC-ARB(SINGLETON$_n$) = LC(HALVING$_{\text{SINGLETON}_n}$) = 1 and log |SINGLETON$_n$| = log n.

For every concept class C that has been considered it turned out that in fact LC-ARB(C) = Θ(LC(HALVING$_C$)). On the other hand Littlestone (1988) presented a concept class C on 8 elements for which LC-ARB(C) < LC(HALVING$_C$). This gave rise to the question whether for every concept class C the halving algorithm is optimal at least up to a constant factor among all learning algorithms using arbitrary hypotheses for C. The following example shows that this is not the case.

We consider the domain $X_n := \{1, \ldots, n, n+1, \ldots, n + \lceil \log n \rceil\}$ and the concept class

$$\text{TAGGED-SINGLETON}_n := \{\emptyset\} \cup \{\{n+\ell\} \mid \ell = 1, \ldots, \lceil \log n \rceil\} \cup$$
$$\cup \left\{ \{i, n+\ell\} \mid i = 1, \ldots, n, \text{and } \ell \in \{1, \ldots, \lceil \log n \rceil \mid\} \right.$$

is the least ℓ with the property that

$$i \leq \sum_{j=1}^{\ell} \left(\frac{n}{2^j} + \lceil \log n \rceil + 1 \right) \Big) \Big\}$$

A concept of the form $\{i, n + \ell\}$ may be viewed as a singleton $\{i\}$ together with a "tag" $\{n + \ell\}$. The other concepts in TAGGED-SINGLETON$_n$ have only been added in order to ensure that LC(TAGGED-SINGLETON$_n$) ≤ 2 (see part a) of the proof of Proposition 5.2). These tags have been distributed in such a manner that the learner gains only constantly many bits of information about C_T when he learns that a certain tag $n + \ell$ is *not* the tag of C_T.

Proposition 5.2.

a) LC-ARB(TAGGED-SINGLETON$_n$) $\leq$ LC(TAGGED-SINGLETON$_n$) ≤ 2
b) LC(HALVING$_{\text{TAGGED-SINGLETON}_n}$) $= \Omega(\log n)$.

Proof. a) For the upper bound we describe a learning algorithm A using equivalence queries. Let $H_1^A := \emptyset$. If a counterexample x_1 is received, this must be a positive one. If $x_1 \leq n$, no further hypotheses are needed as every such element belongs to a single concept. If $x_1 > n$, let $H_2^A := \{x_1\}$. If a counterexample x_2 is received, this must also be a positive one and clearly $C_T = \{x_1, x_2\}$.

b) In order to prove the lower bound for the halving algorithm, we exploit the fact that the majority of all concepts in the class contain the element $n + 1$, therefore this element is contained in the first hypothesis H_1. In the adversary strategy that we construct, $n + 1$ is given as a negative counterexample to H_1. The majority of concepts $C \in$ TAGGED-SINGLETON$_n$ with $n + 1 \notin C$ contain the element $n + 2$, therefore this element is contained in the second hypothesis H_2 of the halving algorithm. The adversary gives $n + 2$ as a negative counterexample to H_2.

Continuing in this way we can guarantee that for any $k \in \{1, \ldots, \log n\}$ with $\Sigma_{j=1}^{k} (n/2^j + \lceil \log n \rceil + 1) < n$ it holds that $n + k \in H_k$, where H_k is the k-th hypothesis of HALVING$_{\text{TAGGED-SINGLETON}_n}$, and there are at least 2 different concepts which are consistent with the first k (negative) counterexamples $n + 1, \ldots, n + k$. The preceding condition on k is satisfied if $k = \lfloor (\log n)/2 \rfloor$ and n is sufficiently large. Hence for n large enough LC(HALVING$_{\text{TAGGED-SINGLETON}_n}$) $\geq \lfloor (\log n)/2 \rfloor$. □

6. Lower bounds for learning with membership and equivalence queries

In this section we prove two general bounds for learning with membership and equivalence queries. In order to motivate these results we start with lower bounds for the simpler models of learning with equivalence queries only, and learning with memberhsip queries only.

Proposition 6.1. For every concept class C

$$\min(\text{LC}(C), \text{MEMB}(C)) \geq \text{LC-ARB}(C) \geq \text{VC-dim}(C).$$

Proof. We only have to show MEMB(C) $\geq$ LC-ARB(C). As noted in the previous section, every learning algorithm A for C using membership queries may be viewed as a decision tree T for C. The learning complexity of A is the depth of the tree T. As T has $|C|$ leaves, its depth is at least $\log |C|$. Therefore

$$\text{MEMB}(C) \geq \log |C|. \tag{6}$$

On the other hand the consideration of the halving algorithm (see (5)) implies $\log |C| \geq$ LC-ARB(C). Putting these inequalities together we get MEMB(C) $\geq$ LC-ARB(C). □

Thus one can raise the question whether LC-ARB(C) or at least VC-dim(C) provide a lower bound to learning complexity in the more powerful model allowing membership queries *and* equivalence queries. We observe first that the answer is negative. Consider the concept class

$$\text{MAJORITY}_n := \left\{ S \mid n + 1 \in S \Leftrightarrow |S \backslash \{n + 1\}| > \frac{n}{2} \right\}$$

over the domain $\{1, \ldots, n + 1\}$.

Proposition 6.2. LC-MEMB(MAJORITY$_n$) $\leq n/2 + 1$ and

$$\text{VC-dim}(\text{MAJORITY}_n) = n.$$

Proof. The second claim is trivial, to prove the first claim we describe a learning algorithm. Start with the membership query "$n + 1 \in C_T$?". If the answer is "no," we know that $|C_T| \leq n/2$. Continue with the hypothesis $\emptyset$. If a counterexample is received, this must be a positive one, providing an element x_1 of the target concept. Let the second hypothesis be $\{x_1\}$. Again, if a counterexample is received, this must be a positive one, providing an element x_2 of C_T, etc. Proceeding in this manner C_T is identified in at most $n/2 + 1$ learning steps. If the answer to the first membership query is "yes" then we know that $|C_T \backslash \{n + 1\}| > n/2$. Then continue with the hypothesis $\{1, \ldots, n + 1\}$. Arguing as above, C_T will be identified in at most $n/2 + 1$ learning steps (this time only negative counterexamples can be obtained). □

In view of the relation

$$\text{LC-MEMB}(\text{MAJORITY}_n) \leq \left(\frac{1}{2} + o(1)\right) \text{VC-dim}(\text{MAJORITY}_n)$$

one can ask if LC-MEMB(C_n) can be *much* smaller than VC-dim(C_n), i.e. if for every $\epsilon > 0$ one can find a family of concept classes C_n such that LC-MEMB(C_n) $\leq (\epsilon + o(1))$VC-dim(C_n). We show that this is not the case, hence in an approximate sense the Vapnik-Chervonenkis dimension is a lower bound to the complexity of learning with membership and equivalence queries. Before that we present an improvement of the construction of the concept class MAJORITY$_n$, showing that the constant 1/2 in Proposition 6.2 can sometimes be replaced by a smaller constant. The proof is a standard application of the probabilistic method in combinatorics (see(Erdös & Spencer, 1974)).

Proposition 6.3. There is a family $(\mathcal{C}_n)_{n \in \mathbf{N}}$ of concept classes with VC-dim$(\mathcal{C}_n) \geq n$ and

$$\text{LC-MEMB}(\mathcal{C}_n) \leq (c_0 + o(1))\text{VC-dim}(\mathcal{C}_n),$$

where $c_0 = 2 - \log 3 = 0.41$.

Proof. Let X be a finite domain and $C, C' \subset X$. Then the *distance* of C and C' is $d(C, C') := |C \Delta C'|$. The ball of radius r around C is $\mathcal{B}(C, r) := \{C' \mid d(C, C') \leq r\}$.

Lemma 6.4. $\text{LC}(\mathcal{B}(C, r)) \leq r$.

Proof. Consider the following learning algorithm. The first hypothesis is $H_1 := C$. If x_i is the counterexample to the i-th hypothesis H_i then $H_{i+1} := H_i \Delta \{x_i\}$. Clearly it holds that $d(C, H_i) = i - 1$. Also, $d(C, H_i) \leq d(C, C_T)$ as $C \Delta H_i \subset C \Delta C_T$. However, $d(C, C_T) \leq r$, thus after at most r counterexamples C_T is identified. □

The construction of Proposition 6.2 is based on the fact that POWER-SET$_n$ is the union of two balls of radius $\lfloor n/2 \rfloor$ with center $\emptyset$, resp. $\{1, \ldots, n\}$. Deciding membership of the additional element $n + 1$ in the target concept at the beginning of the learning algorithm is used as an indicator bit telling which ball to consider. This idea can be generalized as follows.

Assume that POWER-SET$_n$ is the union of ℓ balls $\mathcal{B}_1, \ldots, \mathcal{B}_\ell$ of radius k, where the values of k and ℓ will be determined later on. Consider the domain $X_n := \{1, \ldots, n, n + 1, \ldots, n + \lceil \log \ell \rceil\}$ and the concept class

$$\mathcal{C}_n := \{C \mid C = C_1 \cup C_2, C_1 \subset \{1, \ldots, n\}, C_2 \subset \{n + 1, \ldots, n + \lceil \log \ell \rceil\}, C_1 \in \mathcal{B}_i, \text{ where } i - 1 \text{ is the number denoted by the characteristic vector of } C_2\}.$$

Then VC-dim$(\mathcal{C}_n) \geq n$ as $\{1, \ldots, n\}$ is shattered by $\mathcal{C}_n$. On the other hand LC-MEMB$(\mathcal{C}_n) \leq \lceil \log \ell \rceil + k$. Indeed, after $\lceil \log \ell \rceil$ queries "$j \in C_T$?" for $j = n + 1, \ldots, n + \lceil \log \ell \rceil$ we know the characteristic vector of $C_T \cap \{n + 1, \ldots, n + \lceil \log \ell \rceil\}$. If the number denoted by this vector is $i - 1$ then

$$\{C \cap \{1, \ldots, n\} \mid C \text{ is a consistent hypothesis}\} = \mathcal{B}_i.$$

Therefore we may apply Lemma 6.4 to identify C_T with at most k additional hypotheses.

Hence it remains to choose appropriate values for k and ℓ.

Select a ball $\mathcal{B}$ of radius k randomly by choosing each center $Z \subset \{1, \ldots, n\}$ with the same probability $1/2^n$. Then for a fixed $C_1 \subset \{1, \ldots, n\}$

$$Pr(C_1 \notin \mathcal{B}) = 1 - \frac{m}{2^n},$$

where

$$m = \sum_{j=0}^{k} \binom{\mathrm{n}}{j}$$

is the cardinality of any ball with radius k. If this experiment is repeated ℓ times independently to get the balls $\mathcal{B}_1, \ldots, \mathcal{B}_\ell$, then

$$Pr\left[C_1 \notin \bigcup_{j=1}^{\ell} \mathcal{B}_j\right] = \left[1 - \frac{m}{2^n}\right]^{\ell}.$$

If this probability is less than $1/2^n$ then

$$Pr\left[\text{for some } C_1 \subset \{1, \ldots, n\} \text{ it holds that } C_1 \notin \sum_{j=1}^{\ell} \mathcal{B}_j\right]$$

$$\leq \sum_{C_1} Pr\left(C_1 \notin \bigcup_{j=1}^{\ell} \mathcal{B}_j\right) < 1,$$

therefore

$$Pr\left[\bigcup_{j=1}^{\ell} \mathcal{B}_j \text{ covers POWER-SET}_n\right] > 0.$$

This implies that POWER-SET$_n$ is the union of ℓ balls of radius k.

Hence for a fixed k we need an ℓ satisfying

$$\left(1 - \frac{m}{2^n}\right)^{\ell} < \frac{1}{2^n}. \tag{7}$$

Let $k := \alpha n$. From the Stirling approximation $n! \sim (n/e)^n\sqrt{2\pi n}$ and the fact that $1 - x < e^{-x}$ one gets

$$\binom{n}{k} \sim \frac{1}{\sqrt{\alpha(1 - \alpha)2\pi n}} 2^{h(\alpha)\cdot n}$$

where $h(\alpha) := -\alpha \log \alpha - (1 - \alpha) \log(1 - \alpha)$, and

$$\left(1 - \frac{m}{2^n}\right)^{\ell} < e^{-\frac{m\ell}{2^n}} < e^{\frac{-\binom{n}{k}\ell}{2^n}}.$$

Choosing $\ell := n^2 \cdot 2^{(1-h(\alpha))n}$ the inequality (7) will be satisfied. Hence for the concept class $\mathcal{C}_n$ defined using these values of k and ℓ it holds that

$$\text{LC-MEMB}(\mathcal{C}_n) \leq \alpha n + (1 - h(\alpha))n + o(n).$$

This bound is minimal if $\alpha = 1/3$ and in this case

$$\text{LC-MEMB}(C_n) \leq (2 - \log 3)n + o(n). \qquad \square$$

Now we turn to proving the general bound for LC-MEMB(C) in terms of the Vapnik-Chervonenkis dimension of C.

Theorem 6.5. For every concept class C

$$\text{LC-MEMB}(C) \geq \frac{1}{7}\,\text{VC-dim}(C).$$

Proof. Similarly to the previous lower bounds, the proof proceeds by constructing an adversary. Let us fix a shattered subset Y of the domain X of maximal size (thus $|Y| = $ VC-dim(C)). Let C_i denote the class of concepts which are still candidates for being the target concept after responses were given to the first i queries of the learner. The adversary concentrates on the concept class $C_i \cap Y$ induced by C_i on Y and tries to keep it large, in order to "slow down" the learner.

Thus for each query the adversary gives the response which keeps $|C_{i+1} \cap Y|$ as large as possible. In more detail this means the following (we use the notation (1), (2) and (3)).

I. For a *membership query* "$x \in C_T$?" reply "yes" iff

$$|(C_i \cap Y)_x| \geq |(C_i \cap Y)_{-x}|.$$

Hence the adversary chooses the reply which keeps more subsets of Y as the possible "trace" of the target concept on Y. We note that x is not necessarily an element of Y.

II. For a *hypothesis* $H \in C$ choose an element $y \in Y$ as counterexample for which

$$M_y := |\{C \cap Y \mid C \in C_i \text{ and } C(y) \neq H(y)\}|$$

is as large as possible.

Thus again, the adversary chooses a counterexample from Y which keeps as many subsets of Y as possible, as candidates for being the "trace" of the target concept on Y.

It is obvious that in case I. $|C_{i+1} \cap Y| \geq 1/2|C_i \cap Y|$. However an estimate of the type $|C_{i+1} \cap Y| = \Omega(|C_i \cap Y|)$ is false for case II in general. For example, if $C_i \cap Y$ consists of all singletons and $\emptyset$, and the hypothesis is $\emptyset$, then for every counterexample from Y it holds that $1 = |C_{i+1} \cap Y| = |C_i \cap Y|/(|Y| + 1)$.

Nevertheless, in the next lemma we show that as long as $|C_i \cap Y|$ is "relatively large," it is possible to find a counterexample in such a way that $|C_{i+1} \cap Y| = \Omega(|C_i \cap Y|)$ holds.

In what follows we use some basic notions and results of information theory (see (Csiszár & Körner, 1981) for background and (Kleitman, Shearer & Sturtevant, 1981) for a similar combinatorial application).

The binary entropy function $h : (0, 1) \to \mathbf{R}$ is $h(z) := -z \log z - (1 - z) \log(1 - z)$. It holds that $h(1/2) = 1$, h is strictly monotone increasing on $(0, 1/2]$ and it is symmetric around 1/2. (This function was already used in the proof of Proposition 6.3.)

Lemma 6.6. Let $\alpha \in (0, 1]$ be arbitrary and consider the unique $\beta \in (0, 1/2]$ with $h(\beta) = \alpha$. Let $Y \neq \emptyset$ and $\mathcal{C} \subset 2^Y$ with $|\mathcal{C}| \geq 2^{\alpha|Y|}$. Then for some $y \in Y$ it holds that

$$\beta \leq \frac{|\mathcal{C}_y|}{|\mathcal{C}|} \leq 1 - \beta.$$

Proof. If ξ is a discrete random variable which takes on different values with probabilities $p_1, \ldots, p_n$ then the *entropy* of ξ is defined as

$$H(\xi) := \sum_{i=1}^{n} -p_i \log p_i.$$

We use the fact that if $\xi = (\xi_1, \ldots, \xi_k)$ is a discrete random vector variable then

$$H(\xi) \leq \sum_{i=1}^{k} H(\xi_i). \tag{8}$$

Let $\mathcal{R}$ be a random variable taking values in $\mathcal{C}$ with uniform distribution. (Thus $\mathcal{R}$ is a randomly selected set from $\mathcal{C}$.) Then $H(\mathcal{R}) = \log |\mathcal{C}| \geq \alpha \cdot |Y|$. For every $y \in Y$ let $\mathcal{R}_y$ be the induced random variable with

$$\mathcal{R}_y = \begin{cases} 1 & \text{if } y \in \mathcal{R} \\ 0 & \text{if } y \notin \mathcal{R}. \end{cases}$$

Then as one can identify $\mathcal{R}$ with the vector $\langle \mathcal{R}_y \rangle_{y \in Y}$, (8) implies

$$\alpha|Y| \leq H(\mathcal{R}) \leq \sum_{y \in Y} H(\mathcal{R}_y).$$

Therefore for some $y_0 \in Y$ it holds that $H(\mathcal{R}_{y_0}) \geq \alpha$. But $Pr(\mathcal{R}_{y_0} = 1) = |\mathcal{C}_y|/|\mathcal{C}|$ and $H(\mathcal{R}_{y_0}) = h(Pr(\mathcal{R}_{y_0} = 1))$. Hence the properties of h mentioned above imply that $\beta \leq Pr(\mathcal{R}_{y_0} = 1) \leq 1 - \beta$. □

Now in order to finish the proof of Theorem 6.5 let $\alpha := 1/3$. For $\beta \in (0, 1/2]$ with $h(\beta) = 1/3$ it holds that $\beta \geq 0.0615$. Lemma 6.6 guarantees that as long as $|C_i \cap Y| \geq 2^{|Y|/3}$, after the response of the adversary it holds that $|C_{i+1} \cap Y| \geq \beta |C_i \cap Y|$. Hence if for some i it holds that

$$\beta^i \geq 2^{-2|Y|/3} \tag{9}$$

then after i responses of the adversary

$$|C_i \cap Y| \geq 2^{|Y|} \cdot \beta^i \geq 2^{|Y|/3} > 1$$

and the learning process cannot be concluded yet. The proof is completed by noting that in (9) i can be chosen to be $|Y|/7$. □

Theorem 6.5 in combination with Theorem 2.1 of Blumer, Ehrenfeucht, Haussler and Warmuth (1989) implies that for any finite C

$$O\left(\max\left(\frac{1}{\epsilon}\log\frac{2}{\delta}, \frac{\text{LC-MEMB}(C)}{\epsilon}\log\frac{13}{\epsilon}\right)\right)$$

samples are sufficient for PAC-learning. This appears to be the first result which indicates that learning with membership and equivalence queries cannot be substantially faster than PAC-learning (if one ignores possible differences in *computational complexity*).

Now we present an example where the lower bound provided by Theorem 6.5 is optimal up to a constant factor. Consider the concept class $C_{k,n}$ over the domain $X_n = \{0, 1\}^n$ defined by

$$C_{k,n} := \{C \mid C \text{ is definable by a conjunction of at most } k \text{ literals from } \{x_1, \ldots, x_n, \bar{x}_1, \ldots, \bar{x}_n\}\}.$$

This concept class is of interest not only in the case $k = n$ but also in the case $k \ll n$, as many practically occurring concepts are defined as a conjunction of very few literals from a very large reservoir of potentially relevant attributes (see (Littlestone, 1988)). It can be shown that $\text{LC}(C_{k,n}) \geq \binom{n}{k}$, *hence learning with equivalence* queries from $C_{k,n}$ is not feasible if the number of potentially relevant attributes is large. Thus it is of interest to examine whether $C_{k,n}$ can be learned substantially faster if the learner can make equivalence queries from $C_{k,n}$ and membership queries.

Corollary 6.7. $\text{LC-MEMB}(C_{k,n}) = \Theta(k(1 + \log(n/k)))$.

Proof. We describe a learning algorithm proving the upper bound. Let the first hypothesis be $H_1 := \emptyset$ (defined by $x_1 \wedge \bar{x}_1$). If a counterexample g_1 is received then this must be a positive one. The subsequent queries of the algorithm will all be membership queries. Divide the variables into k groups of approximately equal size and let $g_1^{(i)}$ $(i = 1, \ldots, k)$ be the vector obtained from g_1 by switching the components in the i-th group. Ask the k membership queries "$g_1^{(i)} \in C_T$?". The "no" answers identify those groups which contain a relevant literal, i.e. a literal occurring in the definition of the target concept. If a group contains j relevant literals then these can be found with $O(j \log(n/k))$ membership queries using depth-first search in the search tree describing binary search within that group. After these queries it will also be known that the group contains no further relevant literals. Hence processing all groups containing relevant literals one after the other, all relevant literals can be identified with $O(k \log (n/k))$ additional queries.

The lower bound follows from the result of Littlestone that VC-dim$(C_{n,k}) = \Omega(k(1 + \log (n/k)))$ (Littlestone, 1988) and Theorem 6.5.

One may ask whether Theorem 6.5 can be strengthened to LC-MEMB$(C) = \Omega($LC-ARB$(C))$ for every concept class C. This question remains open. We can prove a slightly weaker lower bound which nevertheless leads to lower bounds for some concrete concept classes which are not implied by Theorem 6.5.

Theorem 6.8. For every concept class C with $|C| > 1$

$$\text{LC-MEMB}(C) \geq \frac{\text{LC-ARB}(C)}{\log(1 + \text{LC-ARB}(C))} \geq \frac{\text{LC-ARB}(C)}{\log(1 + \log |C|)}.$$

Proof. Consider a decision tree T for C such that every leaf of T has depth at least $d :=$ LC-ARB(C). Such a tree exists by Proposition 5.1. We use T to construct an adversary somewhat similar to the adversary of Theorem 6.5.

We focus attention on the 2^d nodes of T at level d. After a certain number of queries a node v *at level d* is called *alive* if the subtree rooted at v contains at least one concept $C \in C$ which is consistent with all the previous responses.

The adversary tries to maintain the following assertion as long as $2^d/2^i(d + 1)^j > 1$: after i membership queries and j equivalence queries there are still at least $2^d/2^i(d + 1)^j$ alive nodes at level d.

The assertion implies the theorem: if $i + j < d/\log(d + 1)$ then there are at least two concepts which are still candidates for being the target concept. We will now prove the assertion by induction on $i + j$. As the assertion clearly holds if $i = j = 0$, we only have to consider the induction step. The strategy of the adversary is to keep as many alive nodes at level d as possible. In somewhat more detail this means the following.

Assume that i membership queries and j equivalence queries have been asked already and the number of alive nodes is $s > 1$.

I. For a *membership query* "$x \in C_T$?" reply "yes" iff this keeps at least $s/2$ nodes alive.
II. For a *hypothesis* $H \in C$ give a counterexample which keeps as many alive nodes as possible.

The response given in case I clearly maintains the truth of the assertion.

In case II one can argue as follows. There is a unique leaf of T labelled H. This leaf determines a unique path leading from the root of T to some node v_H on level d. Assume for contradiction that for every node v on this path the immediate subtree of v which does not contain v_H contains fewer that $s/(d + 1)$ alive nodes. The assumption $s > 1$ implies that in fact $s > d + 1$, as otherwise v_H would be the only alive node. Therefore

$$s < 1 + d \cdot \frac{d}{d + 1} < \frac{s}{d + 1} + d \cdot \frac{s}{d + 1} = s,$$

a contradiction. Thus for some node $\tilde{v}$ on the path it must be the case that the immediate subtree of $\tilde{v}$ not containing v_H contains at least $s/(d + 1)$ alive nodes. Giving the label of this node as a counterexample will keep at least $s/(d + 1)$ alive nodes. Hence the choice of the adversary in case II also maintains the truth of the assertion. □

As an application of Theorem 6.8 we consider the concept class HALFSPACE$_n^d$ of d-dimensional halfspaces over the discrete d-dimensional domain $X_n^d := \{1, \ldots, n\}^d$ defined as

$$\text{HALFSPACE}_n^d := \{C \mid C = X_n^d \cap H \text{ for some halfspace } H \subset \mathbf{R}^d\}.$$

(A halfspace in $\mathbf{R}^d$ is a set $\{(x_1, \ldots, x_d) \mid \Sigma_{i=1}^d w_i x_i \geq t\}$ for some $w_1, \ldots, w_d, t \in \mathbf{R}$.)

These concept classes generalize HALFSPACE$_2^d$ defined in the Section 1.

Corollary 6.9. LC-MEMB(HALFSPACE$_n^d$) $= \Omega(d^2 \log n/(\log d + \log \log n))$.

Proof. This follows from the fact that LC-ARB(HALFSPACE$_n^d$) $= O(d^2 \log n)$ (Maass & Turán, 1989) and Theorem 6.8. □

This lower bound is quite sharp as the best known upper bound is

$$\text{LC-MEMB(HALFSPACE}_n^d) \leq \text{LC(HALFSPACE}_n^d) = O(d^2(\log d + \log n)),$$

(see(Maass and Turán, 1990c)).

Concerning the sharpness of the lower bounds of Theorems 6.5 and 6.8 in general we note that for the concept class SINGLETON$_n$ it holds that LC-ARB(SINGLETON$_n$) = VC-dim(SINGLETON$_n$) = 1. On the other hand LC-MEMB(SINGLETON$_n$) = $n - 1$ as shown by the adversary giving only "no" replies and negative counterexamples. Thus the lower bounds can be very far from being sharp.

We close this section by observing that the lower bounds of Theorems 6.5 and 6.8 remain valid in the stronger model allowing membership and *arbitrary* equivalence queries.

Proposition 6.10. For every concept class $\mathcal{C}$ with $|\mathcal{C}| > 1$

a) LC-ARB-MEMB($\mathcal{C}$) $\geq$ VC-dim($\mathcal{C}$)/7,

b) $\text{LC-ARB-MEMB}(\mathcal{C}) \geq \dfrac{\text{LC-ARB}(\mathcal{C})}{\log(1 + \text{LC-ARB}(\mathcal{C}))} \geq \dfrac{\text{LC-ARB}(\mathcal{C})}{\log(1 + \log |\mathcal{C}|)}$.

Proof. The proof of Theorem 6.5 implies a) without any change. As to b), the only modification needed in the proof of Theorem 6.8 is in the discussion of case II of the adversary strategy. Here H does not necessarily occur as a label of a leaf of T, but nevertheless it defines a unique path from the root to a node v_H on level d. Therefore the same argument applies to this case as well. □

7. Additional remarks on learning with membership queries and equivalence queries

We continue the investigation of learning models allowing membership queries and equivalence queries by establishing some further relationships involving these models and the combinatorial parameters.

First we compare the models allowing equivalence queries only, resp. membership queries only. The learning power of these models is incomparable in the sense that there are concept classes for which learning is much more efficient in the first, resp. the second model. It also follows that the model allowing both membership and equivalence queries is strictly more powerful than any of these models.

We define the concept class ADDRESSING$_n$ over the domain $X_n = \{1, \ldots, n, n + 1, \ldots, n + \lfloor \log n \rfloor\}$ as

ADDRESSING$_n$ $:= \{C \mid C = C_1 \cup C_2, C_1 = \{i\}, 1 \leq i \leq n, C_2 \subset \{n + 1, \ldots, n + \lfloor \log n \rfloor\}$ *and* $i - 1$ is the number denoted in binary notation by the $\lfloor \log n \rfloor$ bits of the characteristic vector of $C_2\}$.

Concept classes related to ADDRESSING$_n$ are used as test cases for neural nets (Rumelhart & McClelland, 1986).

Proposition 7.1.

a) LC-MEMB(SINGLETON$_n$ $\cup$ $\{\emptyset\}$) = LC(SINGLETON$_n$ $\cup$ $\{\emptyset\}$) = 1 and MEMB(SINGLETON$_n$ $\cup$ $\{\emptyset\}$) = n,
b) LC-MEMB(ADDRESSING$_n$) $\leq$ MEMB(ADDRESSING$_n$) $\leq$ $\lfloor \log n \rfloor$ and LC(ADDRESSING$_n$) $\geq n - 1$.

Proof. a) See (Angluin, 1988). The first equality was noted in Section 4. The second equality follows from considering the adversary who responds "no" to the first $n - 1$ membership queries. After these responses there are still two consistent concepts remaining, therefore one more query is required.

b) Let the learning algorithm ask the $\lfloor \log n \rfloor$ membership queries "$i \in C_T$?" for $i = n + 1, \ldots, n + \lfloor \log n \rfloor$. The responses to these queries identify the target concept, thus MEMB(ADDRESSING$_n$) $\leq$ $\lfloor \log n \rfloor$. The lower bound is proved using the adversary who gives $H \cap \{1, \ldots, n\}$ as a negative counterexample to every hypothesis H. After $n - 2$ counterexamples the target concept is not identified yet, hence the learning process cannot be concluded. □

We note that using the concept classes considered above it is possible to construct concept classes C_n for which LC-MEMB(C_n) $\ll$ min(LC(C_n), MEMB(C_n)).

It can also be noted here that clearly VC-dim(ADDRESSING$_n$) = $\lfloor \log n \rfloor$, log(|ADDRESSING$_n$| $-$ 1)/log($|X_n|$ + 1) $<$ 1 and log(chain(ADDRESSING$_n$)) = 0. Thus for this concept class the Vapnik-Chervonenkis dimension is much larger than the other two combinatorial parameters.

Concerning lower bounds to MEMB(C) it is already noted in the proof of Proposition 6.1 that for every concept class C it holds that MEMB(C) $\geq \log|C|$. The example of SINGLETON$_n$ (with or without $\{\emptyset\}$) shows that there are concept classes for which this lower bound is far from being sharp.

Lower bounds to LC-MEMB(C) were discussed in detail in the previous section. Here we observe that LC-MEMB(C), and in fact LC(C) are incomparable to both log $|C|$ and LC(HALVING(C)) in general. The proofs repeat previous arguments and are therefore omitted.

Proposition 7.2.

a) LC-MEMB(SINGLETON$_n$) = LC(SINGLETON$_n$) = $n - 1$ and log |SINGLETON$_n$| = log n,
b) LC-MEMB(SINGLETON$_n$ ∪ $\{\emptyset\}$) = LC(SINGLETON$_n$ ∪ $\{\emptyset\}$) = 1 and log |SINGLETON$_n$ ∪ $\{\emptyset\}$| = log(n + 1). □

Proposition 7.3.

a) LC-MEMB(SINGLETON$_n$) = LC(SINGLETON$_n$) = $n - 1$ and LC(HALVING$_{\text{SINGLETON}_n}$) = 1,
b) LC-MEMB(TAGGED-SINGLETON$_n$) ≤ LC(TAGGED-SINGLETON$_n$) ≤ 2 and LC(HALVING$_{\text{TAGGED-SINGLETON}_n}$) = $\Omega(\log n)$. □

Finally concerning the model allowing membership queries and *arbitrary* equivalence queries, it is clear that this model is *at least as powerful* as learning with membership queries and equivalence queries, and the example of SINGLETON$_n$ again shows that for some concept classes it is *much more powerful*. It is also obvious that this is *at least as powerful* as learning with arbitrary equivalence queries. Propositions 6.2 and 6.3 show that learning with membership and arbitrary equivalence queries can give a speedup by a *constant factor* as compared to learning with arbitrary equivalence queries, and Proposition 6.10 b) shows that it *cannot give a much larger speedup*. The question whether it can give more than a constant speedup remains open. The example of HALF-INTERVAL$_n$ shows that learning complexity in this model can be much larger than the Vapnik-Chervonenkis dimension. The relationship in the other direction is described by Proposition 6.10 a).

8. Learning with partial equivalence queries

In order to illustrate the power of learning with partial equivalence queries we consider the concept class ADDRESSING$_n$ introduced in the previous section.

Proposition 8.1.

a) LC-PARTIAL(ADDRESSING$_n$) = 1;
b) LC-ARB-MEMB(ADDRESSING$_n$) = $\lfloor \log n \rfloor$.

Proof. a) Let the first partial hypothesis of the learner assign 0 to 1, . . . , n and * to $n + 1, \ldots, n + \lfloor \log n \rfloor$. As this partial hypothesis cannot be correct, the response must be a positive counterexample $i \leq n$ and this already identifies the target concept. Thus LC-PARTIAL(ADDRESSING$_n$) ≤ 1 (the $\geq$ part is trivial).

b) As $\lfloor \log n \rfloor$ membership queries are sufficient to learn the target concept we only have to prove the lower bound. Consider the following adversary strategy for the first $\lfloor \log n \rfloor - 1$ queries:

— for an equivalence query give an arbitrary counterexample from $\{n + 1, \ldots, n + \lfloor \log n \rfloor\}$,

— for a membership query give the answer "no."

Assume that after $k \leq \lfloor \log n \rfloor - 1$ queries $\ell \leq k$ elements have been specified from $\{n + 1, \ldots, n + \lfloor \log n \rfloor\}$ and $k - \ell$ elements have been specified from $\{1, \ldots, n\}$. We claim that the target concept is not identified yet.

There are at least $2^{\lfloor \log n \rfloor - \ell} - (k - \ell)$ consistent concepts left, as each unspecified element in $\{n + 1, \ldots, n + \lfloor \log n \rfloor\}$ can be specified in two ways to get a concept, and at most $k - \ell$ of these are eliminated by the elements specified in $\{1, \ldots, n\}$. But as $2^m > m$,

$$2^{\lfloor \log n \rfloor - \ell} - (k - \ell) \geq 2^{\lfloor \log n \rfloor - \ell} - ((\lfloor \log n \rfloor - 1) - \ell) \geq 2,$$

proving the claim. □

The simple learning algorithm of a) above is based on the fact that each $i \in \{1, \ldots, n\}$ is contained in at most one concept. Let us give an informal argument motivating the use of partial hypotheses in general. There may be situations when there are *many* elements $x \in X$ which are *unbalanced*, i.e. for which the number of remaining consistent concepts containing x is either much smaller or much larger than the number of remaining consistent concepts not containing x. If the learner presents a partial hypothesis which only specifies the "probable" behavior of the unbalanced elements and assigns * to the other elements, then it can be expected that each possible response will significantly reduce the number of remaining consistent concepts. The reason is that a *counterexample* will show the occurrence of an unlikely event, and a "*correct*" *reply* will settle the behavior of many elements at the same time.

With the possible exception of VC-dim($\mathcal{C}$), $\log(|\mathcal{C}| - 1)/\log(|X| + 1)$ and $\log(\text{chain}(\mathcal{C}))$, LC-PARTIAL($\mathcal{C}$) can *never* be larger than *any* of the quantities considered in this paper, and in the case of ADDRESSING$_n$ it is *much* smaller than *all* of them. As VC-dim(ADDRESSING$_n$) $= \lfloor \log n \rfloor$, VC-dim($\mathcal{C}$) is not a lower bound to LC-PARTIAL($\mathcal{C}$) in general. On the other hand Proposition 8.3 below shows that the two remaining parameters provide lower bounds for LC-PARTIAL($\mathcal{C}$), with the slight modification that instead of $\log(\text{chain}(\mathcal{C}))$ we have to use $\log_3(\text{chain}(\mathcal{C}))$.

We note that LC-PARTIAL(POWER-SET$_n$) $= n$. This follows from the following adversary strategy: if H is a hypothesis, and there is an i such that $H(i) \in \{0, 1\}$ and i was not given as a counterexample before, then give any such i as a counterexample; if there is no such i then reply "correct." As ADDRESSING$_n$ $\cap$ $\{n + 1, \ldots, n + \lfloor \log n \rfloor\}$ = POWER-SET$_{\lfloor \log n \rfloor}$, this shows that the analogue of the Monotonicity Lemma (Lemma 4.4) is *false* for learning with partial hypotheses.

Proposition 8.2. For every concept class C over a domain X

a) $\text{LC-PARTIAL}(C) \geq \dfrac{\log(|C|)}{\log(|X| + 1)}$,

b) $\text{LC-PARTIAL}(C) \geq \log_3(\text{chain}(C))$.

Proof. a) In Sections 5 and 7 we made use of the fact that learning algorithms with membership queries can be viewed as decision trees. A learning algorithm A with partial hypotheses can be represented as a decision tree of a somewhat different kind. One difference is that an inner node now corresponds to a partial hypothesis. Therefore it has several (at most $|X| + 1$) outgoing edges, one for each possible reply to the hypothesis. Another difference is that a concept may occur as a label of more than one leaf, as different counterexamples may lead to the same target concept. On the other hand every concept occurs as a label of a leaf at least once. The learning complexity LC(A) of A is equal to the depth d of the tree. From these observations we get $(|X| + 1)^d \geq |C|$, implying the claimed bound.

b) The proof essentially requires showing that LC-PARTIAL(HALF-INTERVAL$_n$) $\geq$ $\log_3 n$. However as the Monotonicity Lemma does not hold in this case, one has to argue directly. Again we construct an adversary. Let $C_1 \subsetneq C_3 \subsetneq \ldots \subsetneq C_\ell$ be a longest chain in C, i.e. $\ell = \text{chain}(C)$.

Assume that after a certain number of replies of the adversary there is an interval $\{u, u + 1, \ldots, v\}$ such that all concepts C_j with $u \leq j \leq v$ are still consistent. It is sufficient to show that after the next reply there still remains an interval of $\lceil (v - u + 1)/3 \rceil$ consistent concepts.

Divide the interval $\{u, \ldots, v\}$ into three successive subintervals I_1, I_2 and I_3 such that $|I_1| = \lceil (v - u + 1)/3 \rceil$ and $|I_3| = \lfloor (v - u + 1)/3 \rfloor$. In the adversary strategy we distinguish 3 cases.

1. If $H(x) = 1$ for some $x \notin C_{u + \lceil (v-u+1)/3 \rceil - 1}$ then give x as a negative counterexample.
2. If 1. does not hold but $H(x) = 0$ for some $x \in C_j$ with $j \in I_1 \cup I_2$ then give x as a positive counterexample.
3. If neither 1 nor 2 do hold then reply "correct."

In case 1 all concepts C_j with $j \in I_1$ remain consistent and $|I_1| = \lceil (v - u + 1)/3 \rceil$.

In case 2 all concepts C_j with $j \in I_3$ *and* the concept $C_{v - \lfloor (v-u+1)/3 \rfloor}$ (the largest concept with subscript from I_2) remain consistent, and the number of these concepts is $\lfloor (v - u + 1)/3 \rfloor + 1 \geq \lceil (v - u + 1)/3 \rceil$.

In case 3 all concepts C_j with $j \in I_2$ *and* the concept $C_{u + \lceil (v-u+1)/3 \rceil - 1}$ (the largest concept with subscript from I_1) remain consistent, hence the number of these concepts is at least $\lfloor (v - u + 1)/3 \rfloor + 1 \geq \lceil (v - u + 1)/3 \rceil$. □

Finally we consider the sharpness of the lower bounds of Proposition 8.2. An example where a) is not sharp is provided by HALF-INTERVAL$_n$, where $\log(|C|)/\log(|X| + 1) < 1$. This example also shows that LC-PARTIAL(C) can be much larger than VC-dim(C). To get an example where b) is not sharp we define

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

Clearly $\log_3(\text{chain}(\text{HALFSIZE}_n)) = 0$. On the other hand we observe that LC-PARTIAL(HALFSIZE$_n$) is large.

Proposition 8.3. LC-PARTIAL(HALFSIZE$_n$) $= \lfloor n/2 \rfloor$.

Proof. One can use the following adversary strategy to prove the lower bound: give any counterexample if this is possible, otherwise reply "correct." After the responses are given to $k < \lfloor n/2 \rfloor$ hypotheses, the target concept is not identified yet. The upper bound is achieved by the learning algorithm which always asks a consistent hypothesis with as many 0's as possible. □

One can also note here that

$$\frac{\log(|\text{HALFSIZE}_n| - 1)}{\log(|X_n| + 1)} = \frac{n - o(n)}{\log(n + 1)} \gg \log_3(\text{chain}(\text{HALFSIZE}_n)) = 0.$$

9. Summary of the bounds and relationships

In this section we collect the bounds and relationships presented in the previous sections.

Table 1 contains the bounds for the different learning complexities and combinatorial parameters for the concrete classes used as examples in the paper. In addition, it contains bounds for the concept classes BOX$_n^d$, BALL$_n^d$, HALFSPACE$_n^d$, LINEAR ORDER$_n$ and PERFECT MATCHING$_n$.

The domain for BOX$_n^d$ and BALL$_n^d$ is the discrete d-dimensional space $X_n^d := \{1, \ldots, n\}^d$.

$$\text{BOX}_n^d := \{\{i_1, i_1 + 1, \ldots, j_1\} \times \ldots \times \{i_d, i_d + 1, \ldots, j_d\} \mid 1 \leq i_k, j_k \leq n \text{ for } k = 1, \ldots, d\},$$
$$\text{BALL}_n^d := \{C \subset X_n^d \mid C = X_n^d \cap B \text{ for some ball } B \subset \mathbf{R}^d\}$$

are the classes of d-dimensional boxes, resp. balls over X_n^d. The class HALFSPACE$_n^d$ of d-dimensional halfspaces over X_n^d is defined in Section 6, preceding Corollary 6.9.

The domain for LINEAR ORDER$_n$ and PERFECT MATCHING$_n$ is $X_n := \{(i, j) \mid 1 \leq i \leq j \leq n\}$, and

Table 1. Asymptotic bounds for concrete concept classes.

	SINGLETON$_n$	SINGLETON$_n$ ∪ {∅}	HALF-INTERVAL$_n$ and BALL$_n^d$, BOX$_n^d$, HALFSPACE$_n^d$ for fixed d	ADDRESSING$_n$	LINEAR ORDER$_n$	PERFECT MATCHING$_n$
LC(C) equivalence queries from C	n	1	log n	n	n log n	n^2
MEMB(C) membership queries	n	n	log n (exception: n^d for BALL$_n^d$ and BOX$_n^d$)	log n	n log n	n^2
LC-MEMB(C) equivalence queries from C and membership queries	n	1	log n	log n	n log n	n^2
LC-ARB(C) arbitrary equivalence queries	1	1	log n	log n	n log n	n
LC-ARB-MEMB(C) arbitrary equivalence queries and membership queries	1	1	log n	log n	n log n	n
LC-PARTIAL(C) partial equivalence queries	1	1	log n	1	n	n
VC-dim(C) Vapnik-Chervonenkis dimension	1	1	1	log n	n	n
log(chain(C)) length of longest chain	1	1	log n	1	log n	1

LINEAR ORDER$_n$:= $\{C \subset X_n \mid$ there exists a linear order $\prec$ on $\{1, \ldots, n\}$ such that for every (i, j), $1 \leq i \leq j \leq n$, it holds that $(i, j) \in C \Rightarrow i \prec j\}$,

PERFECT MATCHING$_n$:= $\{C \subset X_n \mid C$ is a pairing of $\{1, \ldots, n\{$ i.e. each $i = 1, \ldots, n$ occurs in exactly one element of $C\}$.

Thus LINEAR ORDER$_n$ contains an encoding of each linear ordering of $\{1, \ldots, n\}$. Each such linear ordering P is encoded by the set of pairs which are ordered in the same way by P and the natural ordering. PERFECT MATCHING$_n$ consists of pairings, or in graph-theoretical terms, perfect matchings of the domain. When we refer to PERFECT MATCHING$_n$, it is always assumed that n is even.

These concept classes appear to be of interest in themselves, and furthermore they provide examples of concept classes for which efficient learning algorithms or lower bounds can be given using results from Combinatorial Optimization or Combinatorics.

All bounds in Table 1 are Θ-bounds, i.e. they represent matching lower and upper bounds up to constant factors. With the exception of the five classes mentioned above, all bounds are shown in the previous sections or can be proved by similar arguments. The results for balls, boxes and halfspaces are presented in Maass and Turán (1989; 1990a; 1990b; 1990c) and Bultman and Maass (1990). The bounds for LINEAR ORDER$_n$ and PERFECT MATCHING$_n$ are given below.

We note that for LINEAR ORDER$_n$ and PERFECT MATCHING$_n$ it holds that

$$|X_n| = \binom{n}{2},$$

$$|\text{LINEAR ORDER}_n| = n!,$$

$$|\text{PERFECT MATCHING}| = \Theta\left(\left(\frac{n}{2e}\right)^{n/2}\right).$$

Proposition 9.1.

a) LC(LINEAR ORDER$_n$) = $O(n \log n)$,
b) MEMB(LINEAR ORDER$_n$) = $O(n \log n)$,
c) LC-ARB-MEMB(LINEAR ORDER$_n$) = $\Omega(n \log n)$,
d) LC-PARTIAL(LINEAR ORDER$_n$) = $\Theta(n)$,
e) VC-dim(LINEAR ORDER$_n$) = $\Theta(n)$.

Proof. If P is a partial order on $\{1, \ldots, n\}$ then a linear extension of P can be viewed as an order-preserving bijection σ between P and the natural ordering on $\{1, \ldots, n\}$. Let the average height $h(x)$ of an element x be the average of $\sigma(x)$ taken over all linear extensions of P. Let $Pr(x < y)$ be the fraction of linear extensions of P for which x is smaller than y.

We use the following results of Kahn and Saks (1984).

Lemma 9.2. (Kahn & Saks, 1984). If $h(x) - h(y) < 1$ then $Pr(y < x) < 8/11$. □

Lemma 9.3. (Kahn & Saks, 1984). For every P there are elements x and y such that $3/11 < Pr(x < y) < 8/11$. □

a) We describe a learning algorithm. Each counterexample determines the relation of two elements in the target linear ordering. Thus after a certain number of hypotheses the information obtained is a partial order on $\{1, \ldots, n\}$. In order to determine the i-th hypothesis H_i consider the partial order P_{i-1} determined by the first $i - 1$ counterexamples and compute $h_{i-1}(x)$ for every x. Let H_i be a linear extension obtained by ordering the elements in the order of increasing h_i value (ties are broken arbitrarily). This will be a linear extension of P_{i-1} as if x is smaller than y in P_{i-1} then $h_{i-1}(x) < h_{i-1}(y)$. From Lemma 9.2 we get that $h(x) \leq h(y)$ implies $Pr(y < x) < 8/11$. Therefore if (x, y) is a counterexample to H_i, where e.g. x precedes y in H_i, then $h_{i-1}(x) \leq h_{i-1}(y)$, therefore the number of linear orders remaining as possible target concepts is reduced by a factor $< 8/11$. Hence the learning algorithm will identify the target linear order after $O(n \log n)$ hypotheses.

b) This is just a reformulation of the standard sorting problem.

c) The lower bound is implied by the following adversary strategy. For a membership query that response is given which keeps more consistent concepts. For an equivalence query we again consider the partial order formed by the previous responses. The counterexample is the pair provided by Lemma 9.3 in this partial order. The response to a membership query clearly keeps at least half of the consistent concepts. Lemma 9.3 implies that a counterexample given to a hypothesis reduces the number of consistent concepts by a constant factor. The adversary can play this strategy for $\Omega(n \log n)$ steps, before the number of consistent concepts is reduced to 1.

d) The learning algorithm to be described will update the following kind of information during the learning process: there are disjoint lists $L_1 = (a_1, \ldots, a_j)$ and $L_2 = (b_1, \ldots, b_k)$ of elements from $\{1, \ldots, n\}$ such that $a_1 \succ \ldots \succ a_j$ are the j largest elements and it holds that $b_1 \prec \ldots \prec b_k$ in the target linear ordering.

The first hypothesis is $H_1 := X_n$, i.e., the encoded version of the natural linear ordering on $\{1, \ldots, n\}$. If a counterexample (c, d) is received then $d \prec c$ in the target linear ordering and we set $L_1 := \emptyset$, $L_2 := (d, c)$.

In general there are two cases.

If $L_2 \neq \emptyset$ then the next partial hypothesis is the encoded version of the relations $\{b_k \succ u : u \notin L_1 \cup L_2\}$. If a counterexample $b_k \prec \tilde{u}$ is obtained then L_2 can be augmented by adding $b_{k+1} := \tilde{u}$ (L_1 is unchanged). If the reply "correct" is obtained then b_k is the largest element not in L_1, therefore L_1 can be augmented by adding $a_{j+1} := b_k$; at the same time b_k is removed from L_2.

If $L_2 = \emptyset$ then the next partial hypothesis is the encoded version of the relations $\{v \succ u : u \notin L_1 \cup \{v\}\}$, where v is an arbitrary fixed element such that $v \notin L_1$. If a counterexample $v \prec \tilde{u}$ is obtained then we set $L_2 := (v, \tilde{u})$. If the "correct" reply is obtained then L_1 can be augmented by adding $a_{j+1} := v$.

After each learning step $|L_1| + |L_1 \cup L_2|$ increases. As this can happen at most $2n$ times, the target linear order is identified in at most $2n$ steps.

The lower bound can be proved by the following adversary strategy. Consider the undirected graph formed by the pairs given as counterexamples to previous partial hypotheses. If the next partial hypothesis assigns 0 or 1 to some pair connecting two connected components of this graph then give this pair as a counterexample, otherwise give the reply "correct." Clearly at least $n - 1$ counterexamples are needed before the target linear concept can be identified.

e) We claim VC-dim(LINEAR ORDER$_n$) $= n - 1$. This follows from the observation that a set of pairs is shattered iff the corresponding undirected graph is a forest. □

Related results on learning linear orders by randomized, polynomial time computable prediction algorithms are obtained by Goldman, Rivest and Schapire (1989).

Proposition 9.2.

a) LC-MEMB(PERFECT MATCHING$_n$) $= \Omega(n^2)$,
b) LC-ARB(PERFECT MATCHING$_n$) $= O(n)$,
c) $\dfrac{\log |\text{PERFECT MATCHING}_n|}{\log |X_n|} = \Omega(n)$.

Proof. a) The adversary for this problem gives negative answers and counterexamples as long as possible. In particular, it responds "no" to a membership query whenever after this response there still remains a consistent concept. Assume that the learner asks an equivalence query H. H cannot be the only consistent concept (otherwise the learner does not need the hypothesis). Let C be another consistent concept. As *all concepts have the same size*, $H \backslash C \neq \emptyset$, therefore the adversary can give a negative counterexample from H and C still remains consistent.

We claim that no learning algorithm can identify the target concept in less than $n(n - 2)/4$ steps. If C is the only consistent concept then it must be the case that C is the only perfect matching in the complement of the graph G formed by the pairs obtained from "no" answers to membership queries and as negative counterexamples to equivalence queries. To see this, note that any other perfect matching in this graph would also be consistent as the "yes" answers of the adversary were *forced* in the sense that every consistent concept must contain them. Now we can use the following result of Hetyei (1964).

Lemma 9.3. (Hetyei, 1964), see (Lovász, 1979). If a graph on n vertices contains a unique perfect matching then it has at most $n^2/4$ edges. □

This implies that G must contain at least

$$\binom{n}{2} - \frac{n^2}{4} = \frac{n(n - 2)}{4}$$

pairs, and thus the claim follows.

Corollary 6.9 gives an almost quadratic lower bound to the complexity of learning d-dimensional halfspaces using membership queries and arbitrary hypotheses. It would be interesting to consider the complexity of this problem in the more powerful model of learning with partial hypotheses, and close the gap between the quadratic upper bound provided by the halving algorithm, and the linear lower bound implied by Proposition 8.2.

Acknowledgments

This paper was written under partial support by NSF-Grant CCR 8903398, OTKA-433 and OTKA-501.

References

Angluin, D. (1981). A note on the number of queries needed to identify regular languages. *Information and Control, 51*, 76–87.

Angluin, D. (1987a). Learning regular sets from queries and counterexamples. *Information and Computation, 75*, 87–106.

Angluin, D. (1987b). *Learning k-term DNF formulas using queries and counterexamples.* (Technical Report YALEU/DCS/RR-557). New Haven, CT: Yale University, Department of Computer Science.

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

Angluin, D. (1990). Negative results for equivalence queries. *Machine Learning, 5*, 121–150.

Angluin, D., Frazier, M. & Pitt, L. (1990). Learning conjunctions of Horn clauses. *Proceedings of the Thirty-First Annual Symposium on Foundations of Computer Science* (pp. 186–192). Washington, DC: IEEE Computer Society.

Angluin, D., Hellerstein, L. & Karpinski, M. (1989). *Learning read-once formulas with queries.* (Technical Report UCB/CSD 89/527). University of California at Berkeley, Computer Science Division. (Also, Technical Report TR-89-050, International Computer Science Institute, Berkeley, California.)

Barzdin, T.M. & Freiwalds, R.V. (1972). On the prediction of general recursive functions, *Sov. Math. Dokl., 13*, 1224–1228.

Baum, E.B. (1990). Polynomial time algorithms for learning neural nets. *Proceedings of the Third Annual Workshop on Computational Learning Theory* (pp. 258–272). San Mateo, CA: Morgan Kaufmann.

Berman, P. & Roos, R. (1987). Learning one-counter languages in polynomial time. *Proceedings of the Twenty-Eighth Annual Symposium on Foundations of Computer Science* (pp. 61–67). Washington, DC: IEEE Computer Society.

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

Bultman, W. & Maass, W. (1991). On-line learning of geometrical concepts with membership queries. *Proceedings of the Fourth Annual Workshop on Computational Learning Theory* (pp. 337–353). San Mateo, CA: Morgan Kaufmann.

Csiszár, I. & Körner, J. (1981). *Information theory.* New York: Academic Press.

Erdős, P. & Spencer, J. (1974). *Probabilistic methods in combinatorics.* New York-Budapest: Academic Press-Akadémiai Kiadó.

Faigle, U. & Turán, Gy. (1988). Sorting and recognition problems for ordered sets. *SIAM Journal on Computing, 17*, 100–113.

Gaizer, T. (1990). The Vapnik-Chervonenkis dimension of finite automata. Unpublished manuscript.

Goldman, S.A., Rivest, R.L. & Schapire, R.E. (1989). Learning binary relations and total orders. *Proceedings of the Thirtieth Annual Symposium on Foundations of Computer Science* (pp. 46–51), Washington, DC: IEEE Computer Society.

Hetyei, G. (1964). *Pécsi Tanárképzö Föiskola Közleményei*, 151–168.

Ishizaka, H. (1990). Polynomial time learnability of simple deterministic langauges. *Machine Learning, 5,* 151–164.

Kahn, J. & Saks, M. (1984). Balancing poset extensions. *Order, 1,* 113–126.

Kleitman, D.J., Shearer, J.B., & Sturtevant, D. (1981). Intersections of *k*-element sets. *Combinatorica, 1,* 381–384.

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

Lovász, L. (1979). *Combinatorial problems and exercises.* Budapest: Akadémiai Kiadó.

Maass, W. (1991). On-line learning with an oblivious environment and the power of randomization. *Proceedings of the Fourth Annual Workshop on Computational Learning Theory* (pp. 167–175). San Mateo, CA: Morgan Kaufmann.

Maass, W. & Turán, Gy. (1989). On the complexity of learning from counterexamples. *Proceedings of the Thirtieth Annual Symposium on Foundations of Computer Science* (pp. 262–267). Washington, DC: IEEE Computer Society Press.

Maass, W. & Turán, Gy. (1990a). On the complexity of learning from counterexamples and membership queries. *Proceedings of the Thirty-First Annual Symposium on Foundations of Computer Science* (pp. 203–210). Washington, DC: IEEE Computer Society Press.

Maass, W. & Turán, Gy. (1990b). Algorithms and lower bounds for on-line learning of geometrical concepts. To appear in *Machine Learning*.

Maass, W. & Turán, Gy. (1990c). How fast can a threshold gate learn? In S. Hanson, G. Drastal, R. Rivest, (Eds.), *Computational learning theory and natural learning systems: Constraints and prospects*, Cambridge, MA: MIT Press, to appear.

Minsky, M. & Papert, S. (1988). *Perceptrons: an introduction to computational geometry, Expanded edition.* Cambridge, MA: MIT Press.

Nilsson, N.J. (1965). *Learning machines.* New York: McGraw-Hill.

Pitt, L. & Valiant, L.G. (1988). Computational limitations on learning from examples. *Journal of the ACM, 35,* 965–984.

Rosenblatt, F. (1962). *Principles of neurodynamics.* New York: Spartan Books.

Rumelhart, D.E. & McClelland, J.L. (1986). *Parallel distributed processing: Explorations in the microstructure of cognition.* Cambridge, MA: MIT Press.

Sauer, N. (1972). On the density of families of sets. *Journal of Combinatorial Theory* (A), *13,* 154–147.

Shelah, S. (1972). A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics, 41,* 247–261.

Valiant, L.G. (1984). A theory of the learnable. *Communications of the ACM, 27,* 1134–1142.

Vapnik, V.N. & Chervonenkis, A. Ya. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability and Its Applications, 16,* 264–280.

Vitter, J.S. & Lin, J.H. (1988). Learning in parallel. *Proceedings of the 1988 Workshop on Computational Learning Theory* (pp. 106–124). San Mateo, CA: Morgan Kaufmann.