# General Lower Bounds on the Query Complexity within the Exact Learning Model

Norbert Klasner and Hans Ulrich Simon

Fakultät für Mathematik
LS Mathematik & Informatik
Ruhr-Universität Bochum
44780 Bochum
Germany

email: {klasner,simon}@lmi.ruhr-uni-bochum.de

September 17, 1998

**Abstract**

In the exact learning model, a learner has to exactly identify a hidden target concept by means of queries. The query complexity of a concept class $\mathcal{C}$ is the smallest number of queries which suffices (in the worst case) to exactly identify an unknown target concept $C \in \mathcal{C}$. Popular query types are equivalence and membership queries, for instance. In this paper, we prove general lower bounds on the query complexity of a concept class $\mathcal{C}$ in terms of combinatorial complexity measures like the capacity function or the VC-dimension of $\mathcal{C}$. These bounds are parameterized by an additional parameter $k$, and are valid for all collections of query types which include equivalence queries and (arbitrarily many) other types of queries with at most $k$ possible answers. They are tight (up to an asymptotically negligible additive term). Special cases of the most general lower bound lead to improvements of previous bounds. The methods, that we apply to derive the various lower bounds, allow to compare the power of equivalence queries with the power of queries with at most $k$ answers. It turns out that (for concept classes whose query complexity asymptotically matches the lower bounds) in a first phase queries with at most $k$ answers provide more information. In a second phase equivalence queries become more effective. The last two sections of the paper are devoted to a discussion of the "breakpoint", where the learner should switch from Phase 1 to Phase 2. Besides this intuitive insight into the power of different query types, our proof method has a strong combinatorial flavor and the auxiliary results may be interesting in their own right.

**Key words:** Lower bounds; Learning; Query learning; Capacity function; Vapnik-Chervonenkis dimension.

# 1 Introduction

## 1.1 Exact Learning, Version Spaces and Query Complexity

The exact learning model was introduced by Angluin in [1]. In this subsection, we briefly describe this model and recall some basic notions like version space and query complexity.

Let $X$ be a finite set and $\mathcal{C} \subseteq 2^X$ a collection of subsets of $X$. In the sequel, $\mathcal{C}$ is called *concept class over domain $X$*. Each set $C \in \mathcal{C}$ is called *concept*. The elements of $X$ are called *instances*. The goal of the learner is to exactly identify a *(hidden) target concept* $\mathcal{C} \in \mathcal{C}$ by means of queries. It is assumed that the queries are answered correctly by an oracle.

In order to formalize the communication protocol between the learner and the oracle, one has to specify the allowed types of queries. Among the most popular query types are the following ones:

**Equivalence Query (EQ)** The learner supplies a set $M \subseteq X$, called its current *hypothesis*[1], and the reply is either "YES", signifying that $M = C$, or a *counterexample*, consisting of an instance $x \in X$ drawn from the symmetric difference $M \Delta C = (M \setminus C) \cup (C \setminus M)$ between $M$ and $C$. (Note that we do not assume that $M \in \mathcal{C}$.)

**Membership Query (MQ)** The learner supplies an instance $x \in X$ and receives one bit of information, signifying whether $x$ belongs to $C$ or not.

In her original paper [1], Angluin considered a couple of additional query types, e.g., subset, superset or disjointness queries.

Some of the papers about exact learning deal with query types allowing for three possible answers, e.g., "YES", "NO" and "AMBIGUOUS". For instance, Goldman, Kwek, and Scott introduced the model of exact learning with unspecified attribute values (UAV model). This model deals with instances viewed as vectors of attributes where not all attributes are necessarily specified. The classification of such an UAV instance $a$ is positive (negative, respectively) if all possible assignments for the unspecified attributes lead to a positive (negative, respectively) classification. Otherwise, $a$ is classified "$*$" (for "AMBIGUOUS").

Queries with even more possible answers arise in learning scenarios, where the learner (e.g., a robot) can perform measurements in order to explore the target concept (e.g., to classify an unknown object).

Given a collection of allowed query types, we say that $\mathcal{C}$ is *exactly learnable* from $q$ queries, if there exists a learner (learning algorithm) $L$ such that each target concept $C \in \mathcal{C}$ is exactly identified by $L$ after at most $q$ queries (and a potentially unlimited amount of computation). The bound $q$ on the number of queries must hold for each learning scenario in which all answers are given correctly. (We can think of an adversary selecting the correct answers.) Note that we ignore computational issues in this paper. All lower bounds that we present are purely information-theoretic and therefore valid even if the learner has unlimited computational power.

In order to keep track of the progress made by the learner, the following notion is quite useful. At each stage of the learning process, the current *version space*[2] is the collection

[1] Intuitively, the hypothesis is the learners current guess for the target concept.

[2] a notion that was introduced by Mitchell [5]

$\mathcal{V}$ of all concepts from $\mathcal{C}$ which are consistent with all answers received so far. Initially, $\mathcal{V} = \mathcal{C}$. If $\mathcal{V}$ is the current version space, the learner issues query $Q$ and receives answer $A$, the subsequent version space is denoted by $\mathcal{V}[Q, A]$. The formal definition of $\mathcal{V}[Q, A]$ clearly depends on the type of $Q$. For instance, if $Q$ is an equivalence query with hypothesis $M$ and answer $A$ consists of a counterexample $x$, then $\mathcal{V}[Q, A] = \{C \in \mathcal{V} | \ x \in M \Delta C\}$. A successful learning scenario using $q$ queries leads to a sequence $\mathcal{V}_0 \supseteq \cdots \supseteq \mathcal{V}_q$ of version spaces with $\mathcal{V}_0 = \mathcal{C}$ and $\mathcal{V}_q = \{C\}$, where $C$ denotes the target concept. In this paper, we consider only queries whose answers $A$ can be (information-theoretically) identified with $\mathcal{V}[Q, A]$.[3] The smallest number of queries needed to learn $\mathcal{C}$ exactly (*query complexity of* $\mathcal{C}$) is denoted as $R_{\mathcal{Q}}(\mathcal{C})$, where $\mathcal{Q}$ denotes the collection of query types. Equivalently, $R_{\mathcal{Q}}(\mathcal{C})$ is the smallest number of queries needed to shrink the initial version space $\mathcal{V}_0 = \mathcal{C}$ to a singleton set. More generally, let $\mathcal{V} \subseteq \mathcal{C}$. Then $R_{\mathcal{Q}}(\mathcal{V})$ denotes the smallest number of queries needed to shrink $\mathcal{V}$ to a singleton set. Clearly, $R_{\mathcal{Q}}(\mathcal{V}) = 0$ if $|\mathcal{V}| = 1$. It is well known that the following recursion (with $Q$ ranging over all queries of a type in $\mathcal{Q}$ and $A$ ranging over all correct answers to $Q$) is valid for all $\mathcal{V}$ with $|\mathcal{V}| \geq 2$:

$$R_{\mathcal{Q}}(\mathcal{V}) = 1 + \min_{Q} \max_{A} R_{\mathcal{Q}}(\mathcal{V}[Q, A]). \tag{1}$$

Unfortunately, this recursion is hard to solve in general.

## 1.2 Combinatorial Measures of Complexity

For each $Y \subseteq X$ and each $\mathcal{V} \subseteq \mathcal{C}$, the *trace* of $\mathcal{V}$ on $Y$ is defined as

$$\mathcal{V} \sqcap Y = \{C \cap Y | \ C \in \mathcal{V}\}.$$

In other words, $\mathcal{V} \sqcap Y$ contains the concepts in $\mathcal{V}$ restricted to subdomain $Y$. It is intuitively clear that it is impossible to shrink the version space to a singleton set using "few" queries if there exists a set $Y$ such that $\mathcal{C} \sqcap Y$ is "large". The following notions prepare more formal statements in this direction. The *capacity function* of $\mathcal{C}$ is given by

$$\Phi_{\mathcal{C}}(m) = \max_{Y \subseteq X : |Y| = m} |\mathcal{C} \sqcap Y|.$$

The *Vapnik Chervonenkis dimension (VC-dimension)* of $\mathcal{C}$ is defined as follows:

$$d(\mathcal{C}) = \max\{m | \ \Phi_{\mathcal{C}}(m) = 2^m\}.$$

Let $n = |X|$ and $d = d(\mathcal{C})$. The following inequality, which bounds the capacity of $\mathcal{C}$ from above in terms of the VC-dimension $d$ of $\mathcal{C}$, is known as *Sauer's Lemma* [6, 7]:

$$\Phi_{\mathcal{C}}(m) \leq \Phi_d(m) := \sum_{0 \leq i \leq d} \binom{m}{i},$$

where $0 \leq m \leq n$.

[3] We are not aware of any natural query type violating this condition.

## 1.3 New Results and Related Work

In this paper, we provide tight general lower bounds on $R_{\mathcal{Q}}(\mathcal{C})$ in terms of the capacity function or in terms of the VC-dimension of $\mathcal{C}$. Before we state our main results, we briefly discuss related work in [4] and [2].

It is easy to see that $R_{EQ}(\mathcal{C}) \geq d(\mathcal{C})$. This general lower bound is tight in the sense that, for each $d$, there exists a specific class $\mathcal{C}_d$ (e.g., the powerset over a domain of size $d$) with $R_{EQ}(\mathcal{C}_d) = d = d(\mathcal{C}_d)$.[4] The same remark holds for $R_{MQ}$. Maass and Turan observed in [4] that the barrier $d(\mathcal{C})$ can be broken when EQs and MQs are combined. They presented a specific family $\mathcal{C}_d$ of classes such that $d(\mathcal{C}_d) \geq d$, but $R_{EQ,MQ}(\mathcal{C}_d) \leq \log(4/3)d + o(d)$. (Note that $\log(4/3) = 0.41\cdots < 1$.) Auer and Long [2] derived the general lower bound $R_{EQ,MQ}(\mathcal{C}) \geq \log(4/3)d(\mathcal{C})$, matching the upper bound of Maass and Turan up to additive term $o(d)$. The lower bound of Auer and Long remains valid if MQ is replaced by any collection of query types with two possible answers per query.[5]

We extend these results in different directions. We consider queries with $k$ possible answers for arbitrary choices of parameter $k \geq 2$. Our upper and lower bounds are stated in terms of $k$ and the capacity function (resp., VC-dimension) of $\mathcal{C}$. These bounds coincide up to an additive term which is smaller than $1 + \log_k d(\mathcal{C})$, where $\log_k$ denotes the logarithm to base $k$. In the special case $k = 2$, our upper (resp., general lower) bound improves on the previously best upper (resp., previously best general lower) bound. The improvement on the upper bound of Maass and Turan is easy to obtain by a slightly more accurate analysis (of basically the same special family of concept classes). The improvement on the general lower bound $\log(4/3)d(\mathcal{C})$ of Auer and Long is harder to obtain. For some classes (e.g., classes for which the general lower bound of Auer and Long is almost tight), there is only little improvement. For others (e.g., for all so-called maximum classes), the improvement is considerable. (The definition of maximum classes and the corresponding lower bounds are given below.)

We close this section with a formal description of the main results in this paper. We say that $\mathcal{Q}$ is a *k-ary* collection of query types if each query $Q$ of a type in $\mathcal{Q} \setminus \{\mathrm{EQ}\}$ has at most $k$ possible answers.[6] Since there is one correct answer for each concept from the current version space $\mathcal{V}$, it follows that the $l \leq k$ possible answers lead to version spaces of the form $\mathcal{V}[Q, A_1], \ldots, \mathcal{V}[Q, A_l]$ with $\mathcal{V} = \cup_{i=1}^{l} \mathcal{V}[Q, A_i]$. The main usage of this definition is that, by the pigeon-hole principle, there exists an answer leading to a version space $\mathcal{V}'$ with $|\mathcal{V}'| \geq |\mathcal{V}|/k$ (except perhaps for queries of type EQ). We define $R_k(\mathcal{C}) = \min_{\mathcal{Q}} R_{\mathcal{Q}}(\mathcal{C})$, where

[4] Of course, there exist also classes for which the lower bound is poor. For instance, halfintervals over domain $\{1, \ldots, n\}$ have VC-dimension 1, but require $\log n$ EQs.

[5] It was shown in [2] that such binary queries can be eliminated at the expense of additional EQs such as to increase the total number of queries at most by factor $1/\log(4/3)$. Since $R_{EQ}(\mathcal{C}) \geq d(\mathcal{C})$, the lower bound follows.

[6] Note that the number of possible answers to a query of type EQ can, in general, not be bounded by a small constant, because many instances can play the role of a counterexample. In contrast, a membership query has only two possible answers. The number of answers to a subset query depends on the precise convention that is used. As defined in [1], an answer "NO" is augmented with an instance $x$ witnessing that the set specified by the learner is not a subset of the target concept. This leads to many possible answers in general. However, a weaker version of a subset query with answer either "YES" or "NO" clearly can join a 2-ary (=binary) collection of query types. A similar remark applies to superset or disjointness queries.

$\mathcal{Q}$ ranges over all $k$-ary collections of query types. Thus $R_k(\mathcal{C})$ is a general lower bound on the query complexity of $\mathcal{C}$ which is valid for all $k$-ary collections of query types. Remember that $\log_k$ denotes the logarithm to base $k$ (where $k = 2$ by default if index $k$ is missing). Let $|X| = n$ and $\Phi_t$ the function defined above within Sauer's Lemma. The main result of our paper reads as follows:

$$R_k(\mathcal{C}) \geq \max_{0 \leq m \leq n} \left( \log_k \Phi_{\mathcal{C}}(m) + \min_{0 \leq t \leq m} (t - \log_k \Phi_t(m)) \right). \tag{2}$$

This result is shown in Section 2. The paper contains a variety of other results which are either shown on the way to the main result or derived as corollaries. Among these results are the following ones:

1. We prove a combinatorial lemma (Lemma 2.1 in Section 2), which implies that a version space of size $\Phi_t(m)$ cannot shrink after an EQ to a size smaller than $\Phi_{t-1}(m-1)$. (Lemma 2.1 is somewhat more general.) This lemma improves on a lemma of Maass and Turan in [4], where bounds on the shrink factor are expressed in terms of the binary entropy function. Our result is optimal in the sense that there exist classes $\mathcal{C}$ which are shrunk to a singleton set in exactly the speed that is suggested by Lemma 2.1.

2. A class $\mathcal{C}$ with $\Phi_{\mathcal{C}}(m) = \Phi_d(m)$ for all $0 \leq m \leq n$ (i.e., the capacity function attains the largest possible values) is called a *maximum class.*[7] As corollary to Lemma 2.1, we prove that each maximum class $\mathcal{C}$ satisfies $R_{EQ,MQ}(\mathcal{C}) \geq d(\mathcal{C})$. In other words, the VC-dimension is a barrier for exact learners of maximum classes, even when EQs and MQs are combined.

3. Section 3 is devoted to an analysis of the maxmin-expression in the main result (see Inequality (2) above). We formulate so-called breakpoint conditions which are valid for the best choices of $t = t(m)$ (minimizing the min-expression) and $m$ (maximizing the max-expression). These conditions enable us to derive general lower bounds (see Theorem 3.1 and Corollary 3.2) implying

   $$R_k(\mathcal{C}) \geq log_k \frac{2k}{k+1} \cdot d(\mathcal{C}).$$

   For $k = 2$, this collapses to the general lower bound

   $$R_{EQ,MQ}(\mathcal{C}) \geq R_2(\mathcal{C}) \geq \log(4/3) \cdot d(\mathcal{C}),$$

   of Auer and Long. Furthermore, the optimal choices of $t$ and $m$ in the maxmin-expression are determined quite accurately (exact determination of $t$ in the case $k = 2$). These estimates of the breakpoints lead to a second lower bound on the query complexity of maximum classes. It can be shown that $R_k(\mathcal{C}) \geq d(\mathcal{C})$ provided that $\mathcal{C}$ is a maximum class over a sufficiently large domain. (See Corollary 3.4 for a precise statement.)

[7] For instance, the class of all subsets of $\{1, \ldots, n\}$ of size at most $d$ is a maximum class. There exists a wide variety of non-isomorphic maximum classes.

# 2 The General Lower Bound

Since Recursion (1) is hard to solve, we follow an idea from [4] and apply a suboptimal (but easier to analyze) adversary strategy for the selection of answers. Clearly, lower bounds on the number of queries that can be derived for a suboptimal adversary strategy are a-fortiori lower bounds on the number of queries in the worst case.

A basic part of the analysis concerns two subsequent version spaces $\mathcal{V}$ and $\mathcal{V}'$. From the perspective of the learner, it is desirable that $\mathcal{V}'$ is much smaller than $\mathcal{V}$. If a high "shrinking speed" of the version space could be achieved all the time, the required number of queries would be small. From the perspective of the adversary, who wants to slow down the learning process, it is desirable that $\mathcal{V}'$ is not much smaller than $\mathcal{V}$.

In Subsection 2.1, we define the so-called greedy adversary strategy. The definition basically follows the intuitive idea of locally slowing down the learning process as much as possible, except that the considerations are restricted to a carefully chosen subdomain $Y$. In Subsection 2.2, we analyze the shrink factor which measures how much the adversary can slow down the learning process if it applies the greedy strategy. As a corollary to the central lemma of this subsection (Lemma 2.1), we show that $R_{EQ,MQ}(\mathcal{C}) \geq d(\mathcal{C})$ for each maximum class $\mathcal{C}$. After the presentation of a rather technical auxiliary result in Subsection 2.3, Subsection 2.4 completes the derivation of the general lower bound, whose tightness is discussed afterwards in Subsection 2.5.

## 2.1 Notational Conventions and the Greedy Adversary Strategy

As long as our considerations focus on two subsequent version spaces, we use the following notations:

- $\mathcal{V}$ denotes the current version space.
- $Q$ denotes the next query of the learner.
- $A$ denotes the answer of the adversary.
- $\mathcal{V}' = \mathcal{V}[Q, A]$ denotes the subsequent version space.
- $Y \subseteq X$ denotes a subdomain of $X$ and $m = |Y|$ denotes its size. We assume that $|\mathcal{C} \sqcap Y| = \Phi_{\mathcal{C}}(m)$. (According to the definition of the capacity function, this can always be achieved.)
- $\mathcal{W} = \mathcal{V} \sqcap Y$ and $\mathcal{W}' = \mathcal{V}' \sqcap Y$ denote the traces of $\mathcal{V}$ and $\mathcal{V}'$ on $Y$, respectively.
- $f = |\mathcal{W}'|/|\mathcal{W}|$ denotes the *shrink factor* associated with $\mathcal{W}$ and $\mathcal{W}'$.

We say that the adversary applies a *greedy strategy* (w.r.t. $Y$), if, given $\mathcal{V}$ and $Q$, the answer $A$ is selected according to the following rules:

**Rule 1** If $Q$ is a query of type EQ, then the counterexample is always chosen from $Y$ (if possible).

**Rule 2** Subject to Rule 1, answer $A$ is chosen such as to maximize shrink factor $f$.

We will later show that the number of queries needed to learn the target concept $C$ on subdomain $Y$ is already sufficiently large. Conceptually, we can think of the learning process as stopped when the learner has identified $C \cap Y$, and assume that counterexamples are always chosen from $Y$.

## 2.2 Analysis of the Shrink Factor

Since $\mathcal{Q}$ is a $k$-ary collection of query types, $f \geq 1/k$ if $Q$ is not an equivalence query.

The following notations and definitions ease a discussion of equivalence queries. Let $w = |\mathcal{W}|$. For each $y \in Y$, let $w_+(y)$ denote the number of sets in $\mathcal{W}$ containing $y$, $w_-(y)$ the number of sets in $\mathcal{W}$ not containing $y$, $w_{min}(y) = \min\{w_+(y), w_-(y)\}$, and $w_{max} = \max_{y \in Y} w_{min}(y)$. A hypothesis $M \subseteq X$ is called a *majority vote* w.r.t. $\mathcal{W}$ if, for each $y \in Y$, $w_+(y) > w_-(y)$ implies that $y \in M$ and $w_-(y) > w_+(y)$ implies that $y \notin M$. The following observations concerning an equivalence query with hypothesis $M$ are obvious:

1. If $M$ is not a majority vote w.r.t. $\mathcal{W}$, then there exists a counterexample $y \in Y$ leading to a shrink factor of at least $1/2$.
2. If $M$ is a majority vote w.r.t. $\mathcal{W}$, then the greedy strategy picks a counterexample $y \in Y$ with $w_{min}(y) = w_{max}$, which leads to a subsequent version space $\mathcal{V}'$ with $|\mathcal{W}'| = |\mathcal{V}' \sqcap Y| = w_{max}$ and thus to shrink factor $w_{max}/w \leq 1/2$.

These observations motivate the search for good lower bounds on $w_{max}$. The following distinction between essential and inessential elements in $Y$ prepares the derivation of a suitable bound. We say that $y \in Y$ is *inessential* for $\mathcal{W}$ (or for $\mathcal{V}$) if it either belongs to all sets in $\mathcal{W}$ or to none. The remaining elements are called *essential* for $\mathcal{W}$ (or for $\mathcal{V}$). Let $m'$ be an upper bound on the number of elements in $Y$ being essential for $\mathcal{W}$. Note that these definitions and the binomial theorem imply that

$$w \leq 2^{m'} = \sum_{i=0}^{m'} \binom{m'}{i}. \tag{3}$$

We are interested in lower bounds on $w_{max}$ in terms of $w$ and $m'$. To this end, we consider the following decomposition of $w$:

$$w = \Phi_t(m') + r = \sum_{i=0}^{t} \binom{m'}{i} + r, \tag{4}$$

where $0 \leq t \leq m'$ and $0 \leq r < \binom{m'}{t+1}$. An inspection of (3) shows that $t$ and $r$ are uniquely determined by $w$ and $m'$. With these notations, the following holds.

**Lemma 2.1**

$$w_{max} \geq \Phi_{t-1}(m'-1) + \frac{r(t+1)}{m'}$$

**Proof** Choose a sequence $y_1, \ldots, y_{m'} \in Y$ containing all elements of $Y$ being essential for $\mathcal{W}$. We associate with $\mathcal{W}$ the matrix $W \in \{0,1\}^{m' \times w}$, where $W_{i,j} = 1$ if and only if $y_i$ belongs to the $j$-th set in $\mathcal{W}$. Note that the columns of $W$ are pairwise different. Furthermore, $w_+(y_i)$ is the number of ones in row $i$ of $W$, $w_-(y_i)$ is the number of zeroes in this row, and $w_{min}(y_i)$ is the minimum of these two quantities, which we call the *balance term* of row $i$, denoted as $w_{min}(i)$ in the course of this proof. The quantity of interest, $w_{max}$, is the maximal balance term.

In the sequel, we use a modification of $W$ (implicitly a modification of $\mathcal{W}$), which leaves the balance terms of all rows (and thus $w_{max}$) invariant. Also the fact that all columns are pairwise different will still be true. An elementary operation, which clearly has this property, is to flip all bits in a row of $W$. If we perform this operation within all rows having more ones than zeroes, we get a new matrix which has no row with more ones than zeroes (and the same maximal balance term as the original matrix and still pairwise different columns). The balance term of each row $i$ is now obtained by counting the number of ones:

$$w_{min}(i) = \sum_{1 \le j \le w} W_{i,j}$$

We may assume w.l.o.g. that $W$ has these properties.

Remember the decomposition of $w$ according to Equation (4). We claim that the total number of ones in $W$ satisfies the following inequality:

$$\sum_{1 \le i \le m', 1 \le j \le w} W_{i,j} \ge \left( \sum_{i=1}^{t} \binom{m'}{i} i \right) + r(t+1).$$

This follows because the columns of $W$ represent a collection of $w$ different subsets of $\{1, \ldots, m'\}$. The total number of ones in $W$ is minimized if this collection is at the bottom of the lattice $2^{\{1,\ldots,m'\}}$, that is, it contains $w$ pairwise different subsets of $\{1, \ldots, m'\}$ of the smallest possible sizes.

By the pigeon-hole principle, there exists an $i \in \{1, \ldots, m'\}$ such that the following holds:

$$\begin{aligned}
w_{min}(i) &= \sum_{1 \le j \le w} W_{i,j} \\
&\ge \frac{1}{m'} \left( \left( \sum_{i=1}^{t} \binom{m'}{i} i \right) + r(t+1) \right) \\
&= \left( \sum_{i=1}^{t} \binom{m'}{i} \frac{i}{m'} \right) + \frac{r(t+1)}{m'} \\
&= \left( \sum_{i=1}^{t} \binom{m'-1}{i-1} \right) + \frac{r(t+1)}{m'} \\
&= \Phi_{t-1}(m'-1) + \frac{r(t+1)}{m'}
\end{aligned}$$

This completes the proof of the lemma. •

**Corollary 2.2** *Let $\mathcal{C}$ be a maximum class. Then $R_{EQ,MQ}(\mathcal{C}) \geq d(\mathcal{C})$.*

**Proof** We begin with the following preliminary consideration. Let $X$ be the domain of $\mathcal{C}$, $d = d(\mathcal{C})$, and $n = |X|$. Note that $|\mathcal{C}| = \Phi_{\mathcal{C}}(n)$ for each concept class. If $\mathcal{C}$ is a maximum class, then $|\mathcal{C}| = \Phi_{\mathcal{C}}(n) = \Phi_d(n)$.

The adversary applies the greedy strategy w.r.t. $Y = X$ for the selection of answers. We may assume w.l.o.g. that the learner does never issue a redundant query. If $Y = X$ and only irredundant queries of type either EQ or MQ are issued, then after $i$ queries at most $n - i$ elements of $X$ are essential for the current version space $\mathcal{V}$. Initially, the version space $\mathcal{V}_0 = \mathcal{C}$ contains $\Phi_d(n)$ concepts, and at most $n$ (all) elements of $X$ are essential for $\mathcal{V}_0$. According to Lemma 2.1 (and according to the fact that MQs lead to a shrink factor of 1/2 or more), the following holds for the version space $\mathcal{V}_1$ obtained after the first query. $\mathcal{V}_1$ contains at least $\Phi_{d-1}(n-1)$ concepts and at most $n-1$ elements of $X$ are essential for $\mathcal{V}_1$. An iterative application of this argument leads to the lower bound $d$ on the total number of queries. •

**Example 2.3** *Let $\mathcal{C}$ be the concept class over domain $\{1, \ldots, n\}$ consisting of all sets of size at most $d$ (for some $d \leq n$). Clearly, $\mathcal{C}$ is a maximum class of VC-dimension $d$. The straightforward learning strategy which issues only EQs, starts with hypothesis $\emptyset$ and iteratively expands the current hypothesis $M$ by the counterexample to $M$ (until $M$ coincides with the target concept), exactly learns $\mathcal{C}$ from $d$ EQs. According to Corollary 2.2, exact learning cannot be achieved with fewer queries even when MQs are allowed in addition to EQs. Note also that the version spaces in the learning scenario have sizes $\Phi_d(n), \Phi_{d-1}(n-1), \ldots, \Phi_0(n-d)$, respectively. This shows that Lemma 2.1 cannot be improved.*

**Example 2.4** *Let $\mathcal{C}$ be the powerset over a domain of size $d$. Clearly, $\mathcal{C}$ is maximum class of VC-dimension $d$. Using an adaptation of a method in [4], it can be shown that $R_2(\mathcal{C}) < \log(4/3)d + O(\log d)$. This shows that Corollary 2.2 does not readily generalize from $R_{EQ,MQ}$ to $R_2$. We will however present a generalization (even to $R_k$) in Section 3 that holds if $n/d$ is sufficiently large.*

Given the decomposition (4) of $w$, function $F_{m'}$ is defined as follows:

$$F_{m'}(w) = \frac{r(t+1)/m' + \Phi_{t-1}(m'-1)}{r + \Phi_t(m')}.$$

Lemma 2.1 immediately leads to

**Corollary 2.5** $w_{max}/w \geq F_{m'}(w)$.

Remember that $w_{max}/w \leq 1/2$ represents the shrink factor after an equivalence query with a majority vote as hypothesis. If the hypothesis is not a majority vote, the shrink factor is at least 1/2. Thus, $F_{m'}(w)$ is a lower bound on the shrink factor after an arbitrary equivalence query.

## 2.3 A Technical Lemma

The final auxiliary result needed to prove our main theorem is as follows.

**Lemma 2.6** *For each $m \geq 1$, $F_m(w)$ is increasing in $w$.*

**Proof** The proof of the lemma makes use of three auxiliary results.

**Claim 2.7** *$\binom{m}{t}/\Phi_t(m)$ and $\binom{m}{t+1}/\Phi_t(m)$ are both decreasing in $t$.*

**Proof of the Claim** It is easy to see that the following holds:

$$\begin{aligned}
\frac{\binom{m}{t+1}}{\binom{m}{t}}\Phi_t(m) &= \frac{m-t}{t+1}\Phi_t(m) \\
&= \sum_{i=0}^{t} \frac{m-t}{t+1}\binom{m}{i} \\
&\leq \sum_{i=0}^{t} \frac{m-i}{i+1}\binom{m}{i} \\
&= \sum_{i=0}^{t} \binom{m}{i+1} \\
&= \Phi_{t+1}(m) - 1 \\
&< \Phi_{t+1}(m)
\end{aligned}$$

It follows that $\binom{m}{t}/\Phi_t(m)$ is decreasing in $t$. Because of

$$\binom{m}{t+1} = \frac{m-t}{t+1}\binom{m}{t},$$

also $\binom{m}{t+1}/\Phi_t(m)$ is decreasing in $t$. •

**Claim 2.8** *$\Phi_t(m)/\Phi_{t+1}(m+1)$ is increasing in $t$.*

**Proof of the Claim** An application of Claim 2.7 shows that

$$\frac{\Phi_{t+1}(m+1)}{\Phi_t(m)} = \frac{\Phi_t(m) + \Phi_{t+1}(m)}{\Phi_t(m)} = 2 + \frac{\binom{m}{t+1}}{\Phi_t(m)}$$

is decreasing in $t$. This implies Claim 2.8. •

The following result is obvious.

**Claim 2.9** *If $f(x)$ is a function of the form*

$$f(x) = \frac{a+bx}{c+dx},$$

*then*

$$f'(x) = \frac{bc-ad}{(c+dx)^2}.$$

*It follows that $f$ is either constant, or decreasing everywhere, or increasing everywhere.*

The proof of the lemma is now obtained as follows. According to Claim 2.8 and the definition of $F_m$,

$$F_m(\Phi_t(m)) = \frac{\Phi_{t-1}(m-1)}{\Phi_t(m)} < \frac{\Phi_t(m-1)}{\Phi_{t+1}(m)} = F_m(\Phi_{t+1}(m))$$

for all $t = 0, \ldots, m-1$. We have still to show that, for each (arbitrarily fixed) $t \in \{0, \ldots, m-1\}$ and all $r = 0, \ldots, \binom{m}{t+1} - 1$, $F_m(\Phi_t(m)+r) < F_m(\Phi_t(m)+r+1)$. Note that $F_m(\Phi_t(m)+r)$ considered as a function in $r$ has the form of function $f$ in Claim 2.9. Claim 2.9 yields that the function is increasing in $r$. This completes the proof of the lemma. •

## 2.4 Derivation of the General Lower Bound

We are now prepared to prove the main theorem of this section along with an important corollary.

**Theorem 2.10**

$$R_k(\mathcal{C}) \geq \max_{0 \leq m \leq n} \left( \log_k \Phi_{\mathcal{C}}(m) + \min_{0 \leq t \leq m} (t - \log_k \Phi_t(m)) \right). \tag{5}$$

**Proof** Let $m$ be an integer between 0 and $n$ and $Y \subseteq X$ a set of size $m$ satisfying $|\mathcal{C} \sqcap Y| = \Phi_{\mathcal{C}}(m)$. Let $\mathcal{V}_0, \mathcal{V}_1, \ldots, \mathcal{V}_q$ with $\mathcal{V}_0 = \mathcal{C}$ and $|\mathcal{V}_q \sqcap Y| = 1$ denote an arbitrary but fixed sequence of version spaces which can evolve when the adversary applies a greedy strategy w.r.t. $Y$. We have to show that $q \geq \max_{0 \leq m \leq n} (\log_k \Phi_{\mathcal{C}}(m) + \min_{0 \leq t \leq m} (t - \log_k \Phi_t(m)))$.

For $i = 0, 1 \ldots, q$, let $\mathcal{W}_i$ denote the trace of $\mathcal{V}_i$ on set $Y$and $w_i = |\mathcal{W}_i|$. If the $i$-th query of the learner is of type EQ and the reply consists of counterexample $y \in Y$, then $y$ is inessential for $\mathcal{W}_i$ (and for all subsequent version spaces). We assume w.l.o.g. that the learner does never use a hypothesis which allows to return a counterexample from $Y$ twice. It follows that after $j$ equivalence queries $m - j$ is an upper bound on the number of essential elements in $Y$.

For $i = 1, \ldots, q$, let $f_i = w_i / w_{i-1}$ denote the shrink factor associated with the $i$-th query of the learner. Thus, for all $i = 1, \ldots, q$, the following equation is valid:

$$w_i = w_0 \cdot f_1 \cdots f_i = \Phi_{\mathcal{C}}(m) \cdot f_1 \cdots f_i. \tag{6}$$

In particular for $i = q$, we obtain:

$$1 = w_q = \Phi_{\mathcal{C}}(m) \cdot f_1 \cdots f_q. \tag{7}$$

It is intuitively clear that Equation (7) can be used to derive a lower bound on $q$: if the shrink factors $f_i$ are much larger than $1/\Phi_{\mathcal{C}}(m)$, there must be many of them, i.e., $q$ is "large". In order to turn this informal reasoning into a formal proof, we must find a computation scheme which allows to replace the shrink factors $f_i$ (which we do not know explicitly) by reasonably good lower bounds which can be constructed explicitly. To this end, we define upper bounds $m'_i$ on the number of elements in $Y$ being essential for $\mathcal{W}_i$ and lower bounds $f'_i$ on $f_i$ and $w'_i$ on $w_i$ inductively as follows:

1. $m'_0 = m$ and $w'_0 = w_0 = \Phi_{\mathcal{C}}(m)$.
2. Assume inductively that $m'_0, \ldots, m'_{i-1}$, $w'_0, \ldots, w'_{i-1}$ and $f'_1, \ldots, f'_{i-1}$ are given already. If the $i$-th query is of type EQ, we define

$$m'_i = m'_{i-1} - 1 \text{ and } f'_i = F_{m'_{i-1}}(w_{i-1}). \tag{8}$$

Otherwise, we define

$$m'_i = m'_{i-1} \text{ and } f'_i = \frac{1}{k}.$$

In both cases, we set

$$w'_i = w'_0 \cdot f'_1 \cdots f'_i.$$

Applying Corollary 2.5 and Lemma 2.6, an easy induction shows that indeed $f'_i \leq f_i$ and $w'_i \leq w_i$ for $i = 1, \ldots, q$.

Let $q = s+t$, where $t$ is the number of EQs among the $q$ queries and $s$ is the number of the other queries. We claim that the smallest possible value of product $f'_1 \cdots f'_q$ is attained, when we assume the last $t$ factors correspond to EQs, and the first $s$ factors correspond to the other queries. This follows easily from Lemma 2.6: the subproduct of the $t$ factors associated with EQs becomes smaller if these factors occur at the end of the product, whereas the remaining $s$ factors have all value $1/k$ no matter where they occur. Thus $f_1 \cdots f_q \geq f'_1 \cdots f'_q$ given the assumption that $f'_1 = \cdots = f'_s = 1/k$ and the remaining $t$ shrink factors are determined according to Equation (8).

We are now able to complete the proof of the theorem. We know that

$$1 = \Phi_{\mathcal{C}}(m) \cdot f_1 \cdots f_q \geq \Phi_{\mathcal{C}}(m) \cdot f'_1 \cdots f'_q = \frac{\Phi_{\mathcal{C}}(m)}{k^s} \cdot f'_{s+1} \cdots f'_{s+t}.$$

We claim that

$$w'_s = \Phi_{\mathcal{C}}(m)/k^s \leq \Phi_t(m). \tag{9}$$

Assume for the sake of contradiction that $w'_s > \Phi_t(m)$. Iterative application of Lemma 2.1 leads to $w'_{s+i} > \Phi_{t-i}(m-i)$ for $i = 1, \ldots, t$. In particular, $w'_q = w'_{s+t} > \Phi_0(m-t) = 1$.

This contradicts $w'_q \leq w_q = 1$. Thus, Inequality (9) is correct. Therefore, $s \geq \log_k \Phi_{\mathcal{C}}(m) - \log_k \Phi_t(m)$ and $q = s + t \geq \log_k \Phi_{\mathcal{C}}(m) + t - \log_k \Phi_t(m)$. Note that $m$ is determined by the adversary and $t = t(m)$ by the learner. We have therefore to maximize over all possible choices of $m$ and to minimize over all possible choices of $t$ for a given $m$. This completes the proof of the theorem. •

Setting $m = d(\mathcal{C})$, we get the following

**Corollary 2.11**

$$R_k(\mathcal{C}) \geq \log_k 2 \cdot d(\mathcal{C}) + \min_{0 \leq t \leq d(\mathcal{C})} \left(t - \log_k \Phi_t(d(\mathcal{C}))\right). \tag{10}$$

## 2.5 Tightness of the General Lower Bound

The following Theorem 2.12 presents an upper bound for a special family of concept classes matching the general lower bound in Corollary 2.11 up to an additive term that is smaller than $1 + \log_k d(\mathcal{C})$. The proof of this theorem is an easy adaptation of a similar proof in [4], where it was shown that there exists a family $(\mathcal{C}_d)_{d \geq 1}$ of concept classes with $d(\mathcal{C}_d) \geq d$ and $R_{EQ,MQ}(\mathcal{C}_d) \leq \log(4/3)d + o(d)$.

**Theorem 2.12** *Let $\mathcal{C}_d$ be the powerset over domain $X_d = \{1, \ldots, d\}$. Then $d(\mathcal{C}_d) = d$ and*

$$R_k(\mathcal{C}_d) < 1 + \log_k d + \log_k 2 \cdot d + \min_{0 \leq t \leq d} \left(t - \log_k \Phi_t(d)\right).$$

**Proof** Let $C \subseteq X_d$. The *ball* of radius $t$ around $C$ consists of all sets $M$ satisfying $|M \Delta C| \leq t$. Maass and Turan have shown in [4] that $\mathcal{C}_d$ can be covered by $N$ balls of radius $t$ provided that $N, t$ satisfy

$$(1 - \Phi_t(d)2^{-d})^N < 2^{-d}.$$

Formula $1 + x \leq e^x$ with equality for $x = 0$ only implies that

$$(1 - \Phi_t(d)2^{-d})^{d2^d/\Phi_t(d)} < e^{-d} < 2^{-d}.$$

Thus, $\mathcal{C}_d$ can be covered by $N(d) := \lceil d \cdot 2^d/\Phi_t(d) \rceil$ balls of radius $t$. Let $s(d) := \lceil \log_k N(d) \rceil$. We arbitrarily fix a $k$-ary enumeration of these $N$ balls such that each ball has a unique address of the form $(\alpha_{s(d)-1}, \ldots, \alpha_0) \in \{0, \ldots, k-1\}^{s(d)}$. With each concept $C \in \mathcal{C}_d$ we associate the address $\alpha(C)$ of a ball containing $C$. Clearly, $s(d)$ $k$-ary questions suffice to learn $\alpha(C)$ for target concept $C$. As mentioned in [4], $t$ EQs suffice to identify the target concept $C$ within a ball of radius $t$ containing $C$. The total number of queries is

$$\begin{aligned} s(d) + t &= \lceil \log_k d + \log_k 2 \cdot d + t - \log_k \Phi_t(d) \rceil \\ &< 1 + \log_k d + \log_k 2 \cdot d + t - \log_k \Phi_t(d). \end{aligned}$$

Since $t$ can be chosen such as to minimize this expression, the theorem follows. •

The general lower bound (10) resulted from (5) by setting $m = d(\mathcal{C})$. If $\mathcal{C}_d$ is the powerset over a domain of size $d$, $m = d = d(\mathcal{C}_d)$ is a perfect choice because it maximizes the max-expression within bound (5). In other words, bounds (5) and (10) coincide in this case. Note, however, that bound (5) can be considerably better than bound (10) for other classes. For instance, for maximum classes of VC-dimension $d$ over a domain of size $n \geq (k+1)d$, it can be shown that the best choice for $m$ is (roughly) $(k+1)d$. With this choice, bound (5) leads to a value close to $d$, whereas bound (10) leads to a value close to $\log_k \frac{2k}{k+1} d$.[8] In general, the following holds. If $\Phi_{\mathcal{C}}(n)$ is close to $2^{d(\mathcal{C})}$ (the smallest possible value for the capacity function), bounds (10) and (5) do not differ much. However, the more $\Phi_{\mathcal{C}}(n)$ exceeds $2^{d(\mathcal{C})}$, the more bound (5) becomes superior.

# 3 Breakpoint Analysis and Implications

This section is devoted to a thorough analysis of the lower bound that was established in Theorem 2.10. We begin with a discussion of the term

$$\min_{0 \leq t \leq m} t - \log_k \Phi_t(m).$$

Obviously,

$$\begin{aligned} (t+1) - \log_k \Phi_{t+1}(m) \leq t - \log_k \Phi_t(m) &\Leftrightarrow \Phi_{t+1}(m) \geq k\Phi_t(m) \\ &\Leftrightarrow \binom{m}{t+1} \geq (k-1)\Phi_t(m). \end{aligned} \tag{11}$$

We say that it is *profitable to increment* $t$ in this case. According to Claim 2.7 in Subsection 2.3, $\binom{m}{t+1}/\Phi_t(m)$ is decreasing in $t$. It follows that there exists a unique *breakpoint* $t_k(m)$ such that for all $t$ it is profitable to increment $t$ if $t < t_k(m)$, but not profitable to increment $t$ if $t \geq t_k(m)$. The following obvious *breakpoint conditions* on $t$ are only satisfied for $t = t_k(m)$:

$$\binom{m}{t} \geq (k-1)\Phi_{t-1}(m) \text{ and } \binom{m}{t+1} < (k-1)\Phi_t(m).$$

The breakpoint conditions imply that $t_k(m) \leq m/2$. We assume in the sequel that $m \geq 2k-1 \geq 1$ which implies that $t_k(m) \geq 2$. The following theorem establishes a general bound in terms of the binary entropy function

$$h(\alpha) = -\alpha \log \alpha - (1-\alpha)\log(1-\alpha).$$

**Theorem 3.1**

$$\begin{aligned} R_k(\mathcal{C}) &\geq \max_{2k-1 \leq m \leq n} \left(\log_k \Phi_{\mathcal{C}}(m) + t_k(m) - \log_k \Phi_{t_k(m)}(m)\right) \\ &\geq \max_{2k-1 \leq m \leq n} \left(\log_k \Phi_{\mathcal{C}}(m) + (1 - \log_k(k+1))m\right). \end{aligned}$$

[8] In Section 3, we rigorously prove that $R_k(\mathcal{C}) \geq d(\mathcal{C})$ is valid for all maximum classes over a sufficiently large domain. Here we only want to stress that bound (5) is strong enough to yield approximately the same result.

**Proof** The first inequality rewrites the lower bound in Theorem 2.10 in terms of the breakpoint $t_k(m)$. We have to show the second inequality. Stirling's formula

$$1 \leq \frac{n!}{\sqrt{2\pi n} n^n e^{-n}} \leq e^{\frac{1}{12n}},$$

combined with the fact that $2 \leq t_k(m) \leq m/2$ and with the breakpoint conditions implies that for $t = t_k(m)$ the following holds:

$$\begin{aligned}
\Phi_t(m) &= \binom{m}{t} + \Phi_{t-1}(m) \\
&\leq \left(1 + \frac{1}{k-1}\right)\binom{m}{t} \\
&\leq \frac{(1 + 1/(k-1))e^{\frac{1}{12m}} 2^{h(t/m)m}}{\sqrt{2\pi t(1-t/m)}} \\
&\leq \frac{2e^{\frac{1}{12m}}}{\sqrt{\pi t}} 2^{h(t/m)m} \\
&\leq 2^{h(t/m)m}
\end{aligned}$$

The observations that $t - \log_k 2^{h(t/m)m}$ is minimized for $t = m/(k+1)$ and

$$h\left(\frac{1}{k+1}\right) = \log(k+1) - \frac{k \log k}{k+1}$$

lead, after some straightforward algebraic manipulations, to the second inequality. •

The special case $m = d(\mathcal{C})$ leads to the following

**Corollary 3.2** *If $d(\mathcal{C}) \geq 2k-1$, then*

$$R_k(\mathcal{C}) \geq \log_k \frac{2k}{k+1} \cdot d(\mathcal{C}).$$

*In particular:*

$$R_2(\mathcal{C}) \geq \log \frac{4}{3} \cdot d(\mathcal{C}) \text{ and } R_3(\mathcal{C}) \geq \log_3 \frac{3}{2} \cdot d(\mathcal{C})$$

The lower bound on $R_2(\mathcal{C})$ is known from [2]. It can be shown that the bounds in Corollary 3.2 are tight up to additive term $O(\log_k d)$ for the powerset over a domain of size $d$. However, for each concept class $\mathcal{C}$ with $\Phi_{\mathcal{C}}(n)$ being much larger than $2^{d(\mathcal{C})}$, the bounds are poor. (Compare with the discussion at the end of Subsection 2.5.)

The following result presents quite accurate estimates for the breakpoint. It holds for all $k \geq 2$.

**Lemma 3.3** *1. For all $m$: $\left\lfloor \frac{m+1}{k+1} \right\rfloor \leq t_k(m) \leq \left\lfloor \frac{m+1}{k} \right\rfloor$.*
*If $m$ is sufficiently large, the upper bound on $t_k(m)$ improves to $\left\lceil \frac{m}{k+1} \right\rceil + 1$.*

2. *For all $m \le 12$: $t_2(m) = \lceil m/3 \rceil$.*

3. *For all $m > 12$: $t_2(m) = \lfloor m/3 \rfloor + 1$.*

4. *For all $m$: $t_k(m+1) \in \{t_k(m), t_k(m)+1\}$.*

For the sake of readability the proof is given in the Appendix A.

**Corollary 3.4** *Let $\mathcal{C}$ be a maximum class over a domain of size $n$ and $n \ge (k+1) \cdot d(\mathcal{C})$. Then, $R_k(\mathcal{C}) \ge d(\mathcal{C})$.*

**Proof** The proof is similar to the proof for Corollary 2.2. Initially the version space $\mathcal{V}_0 = \mathcal{C}$ contains $\Phi_d(n)$ concepts and $n$ is an upper bound on the number of the elements in $X$ being essential. We claim that, for each $0 \le i \le d$, there exists an $n_i$ with $n - i \le n_i \le n$ such that after $i$ queries the current version space $\mathcal{V}_i$ contains at least $\Phi_{d-i}(n_i)$ concepts and at most $n_i$ elements of $X$ are essential for $\mathcal{V}_i$. Note that this claim implies the corollary. For $i = 0$, this assertion of the claim holds with $n_0 = n$. We assume inductively that the assertion of the claim holds for an arbitrary but fixed $i \ge 0$. If the next query is of type EQ, Lemma 2.1 applies and the assertion of the claim holds for $i+1$ with $n_{i+1} = n_i - 1$. Otherwise, we set $n_{i+1} = n_i$ and observe that $\mathcal{V}_{i+1}$ contains at least $\Phi_{d-i}(n_i)/k$ concepts. We have to show that $\Phi_{d-i}(n_i) \ge k \cdot \Phi_{d-i-1}(n_i)$, which is equivalent to $d - i < t_k(n_i)$. According to Lemma 3.3, $t_k(n_i) \ge \frac{n_i+1}{k+1}$. Thus, it suffices to show that $d - i < \frac{n_i+1}{k+1}$. This easily follows from $n \ge (k+1) \cdot d$ after some algebraic manipulations. •

We now turn our attention to the breakpoint conditions for parameter $m$, i.e., we look for conditions which yield information about the optimal choice $m_{max}$ of $0 \le m \le n$ when we want to maximize the *profit*

$$P(m) = \log_k \Phi_{\mathcal{C}}(m) + t_k(m) - \log_k \Phi_{t_k(m)}(m). \tag{12}$$

We say that it is *profitable to increment* $m$ if $m < n$ and $P(m+1) \ge P(m)$.

**Lemma 3.5** *Let $m \in \{0, \dots, n-1\}$. If*

$$\Phi_{\mathcal{C}}(m+1) \ge \frac{k+1}{k} \cdot \Phi_{\mathcal{C}}(m),$$

*then it is profitable to increment $m$.*

**Proof** We recall from Equation (12) that profit function $P(m)$ is the sum of three profit terms: $\log_k \Phi_{\mathcal{C}}(m)$ (first term), $t_k(m)$ (second term), and $-\log_k \Phi_{t_k(m)}(m)$ (third term). Since $t_k(m+1) \in \{t_k(m), t_k(m)+1\}$ (see Lemma 3.3), we distinguish two cases:

**Case 1** $t_k(m+1) = t_k(m)$.

In this case, the second profit term does not change when we increment $m$. From the breakpoint conditions for $t = t_k(m)$ and from (11), we get $\Phi_{t-1}(m) \le \Phi_t(m)/k$. Thus:

$$\Phi_t(m+1) = \Phi_t(m) + \Phi_{t-1}(m) \le \frac{k+1}{k} \Phi_t(m).$$

It follows that the third profit term decreases at most by $\log_k \frac{k+1}{k}$ when $m$ is incremented. This is however compensated by the first profit term if $\Phi_{\mathcal{C}}(m+1) \ge \frac{k+1}{k} \Phi_{\mathcal{C}}(m)$.

**Case 2** $t_k(m+1) = t_k(m) + 1$.

In this case, the second profit term is incremented by 1. From the breakpoint conditions for $t = t_k(m)$ and from (11), we get $\Phi_{t+1}(m) < k\Phi_t(m)$. Thus:

$$\Phi_{t+1}(m+1) = \Phi_{t+1}(m) + \Phi_t(m) < (k+1)\Phi_t(m).$$

The third profit term is therefore decreased by less than $\log_k(k+1)$. This is however compensated by the first two profit terms if $\Phi_{\mathcal{C}}(m+1) \geq \frac{k+1}{k}\Phi_{\mathcal{C}}(m)$.

•

Let $d$ denote the VC-dimension of $\mathcal{C}$. As long as $m < d$, $\Phi_{\mathcal{C}}(m+1) = 2\Phi_{\mathcal{C}}(m)$. Lemma 3.5 leads therefore immediately to the following result:

**Corollary 3.6** $d \leq m_{max} \leq n$.

We discuss two extreme examples. If $X = \{1, \ldots, n\}$, $d \leq n$, and $\mathcal{C}$ over domain $X$ contains exactly the subsets of $\{1, \ldots, d\}$, the capacity function $\Phi_{\mathcal{C}}$ satisfies $\Phi_{\mathcal{C}}(m) = 2^{\min\{d,m\}}$. Clearly, $m_{max} = d$. On the other side, if $\mathcal{C}$ is a maximum class of VC-dimension $d$ over a domain of size $n$, then $\Phi_{\mathcal{C}}(m) = \Phi_d(m)$ for all $0 \leq m \leq n$. In this case, the breakpoint condition in Lemma 3.5 can be reformulated as follows:

$$\begin{aligned} \Phi_d(m+1) = \Phi_d(m) + \Phi_{d-1}(m) \geq \frac{k+1}{k}\Phi_d(m) \Leftrightarrow \Phi_d(m) &\leq k\Phi_{d-1}(m) \\ \Leftrightarrow \binom{m}{d} &\leq (k-1)\Phi_{d-1}(m). \end{aligned}$$

This is basically (the dual of) the breakpoint condition on $t_k(m)$. We can initially set $m = d$ and then increment $m$ while it is profitable. This leads to the maximal value of $m$ which satisfies $\binom{m}{d} \leq (k-1)\Phi_{d-1}(m)$. Applying Lemma 3.3, it can be shown that $m_{max}$ is roughly $\min\{n, (k+1)d\}$ in this case.

For classes $\mathcal{C}$ between these extremes, one can roughly expect the following. While $m < d(\mathcal{C})$, the value of the capacity function doubles, when we increment $m$. If we increment $m$ beyond the threshold $d(\mathcal{C})$, the exponential growth (by factor 2) of the capacity function will eventually slow down to a polynomial growth (according to Sauer's Lemma). Intermediately, $m$ will pass through values where an incrementation of $m$ increases the value of the capacity function roughly by factor $(k+1)/k$. Among these values, we will find $m_{max}$.

Note however that not every capacity function smoothly slows down from exponential to polynomial growth. In fact, one can construct artificial classes, where the growth of the capacity function is fairly small in an interval $m, \ldots, m+c$, but for values exceeding $m+c$, it "recovers" and grows with exponential speed again. One can therefore not expect for general $\mathcal{C}$ to get the *global* optimum $m_{max}$ by a *local* breakpoint condition like the one in Lemma 3.5.

# Acknowledgements

Hans Ulrich Simon gratefully acknowledges the support of Deutsche Forschungsgemeinschaft grant Si 498/3-2 and the support of the German-Israeli Foundation for Scientific Research and Development grant I-403-001.06/95.

# A Proof of Lemma 3.3

First let us repeat the statement of Lemma 3.3.

**Lemma 3.3** *1. For all m:* $\left\lfloor\frac{m+1}{k+1}\right\rfloor \leq t_k(m) \leq \left\lfloor\frac{m+1}{k}\right\rfloor$.
*If m is sufficiently large, the upper bound on* $t_k(m)$ *improves to* $\left\lceil\frac{m}{k+1}\right\rceil + 1$.

2. *For all* $m \leq 12$: $t_2(m) = \lceil m/3 \rceil$.

3. *For all* $m > 12$: $t_2(m) = \lfloor m/3 \rfloor + 1$.

4. *For all m:* $t_k(m+1) \in \{t_k(m), t_k(m)+1\}$.

Recall the breakpoint conditions

$$\binom{m}{t} \geq (k-1) \cdot \Phi_{t-1}(m) \tag{BP1}$$

and

$$\binom{m}{t+1} < (k-1) \cdot \Phi_t(m). \tag{BP2}$$

The following simple observation is helpful in the proof of the lemma.

**Observation A.1**

$$(\text{BP1}) \Rightarrow t_k(m) \geq t \tag{13}$$
$$(\text{BP2}) \Rightarrow t_k(m) < t+1 \tag{14}$$

For the sake of notational simplicity we introduce the following notations (see [3]). We define the $n$-th *falling factorial power* of $x$ by $x^{\underline{n}} = x!/(x-n)! = x \cdot (x-1) \cdots (x-n+1)$ and the $n$-th *rising factorial power* of $x$ by $x^{\overline{n}} = (x+n-1)!/(x-1)! = x \cdot (x+1) \cdots (x+n-1)$.

**Observation A.2** $\binom{m}{k+1}/\Phi_k(m)$ *is increasing in m.*

**Proof**

$$\frac{\Phi_k(m)}{\binom{m}{k+1}} = \sum_{i=0}^{k} \frac{\binom{m}{k-i}}{\binom{m}{k+1}} = \sum_{i=0}^{k} \frac{(k+1)! \cdot (m-k-1)!}{(k-i)! \cdot (m-k+i)!} = \sum_{i=0}^{k} \frac{(k+1)^{\underline{i+1}}}{(m-k)^{\overline{i+1}}}.$$

$(m-k)^{\overline{i+1}}$ is increasing in $m$ for each $0 \leq i \leq k$. •

**Proof of 1.**

$\boldsymbol{t_k(m) \geq \left\lfloor \frac{m+1}{k+1} \right\rfloor}$**:**

Because of (13), it is sufficient to prove

$$\forall t \leq \frac{m+1}{k+1} \colon \binom{m}{t} \geq (k-1) \cdot \Phi_{t-1}(m).$$

Assume $t \leq \frac{m+1}{k+1}$.

$$\frac{\binom{m}{t-1}}{\binom{m}{t}} = \frac{t}{m-t+1}$$

is increasing in $t$. $\frac{t}{m-t+1} \leq \frac{1}{k}$ for $t \leq \frac{m+1}{k+1}$. Hence, for each $i = 0, \ldots, t-1$:

$$\frac{\binom{m}{t-i-1}}{\binom{m}{t-i}} \leq \frac{1}{k}.$$

This yields

$$\frac{\binom{m}{t-i-1}}{\binom{m}{t}} = \frac{\binom{m}{t-i-1}}{\binom{m}{t-i}} \cdot \frac{\binom{m}{t-i}}{\binom{m}{t-i+1}} \cdot \ldots \cdot \frac{\binom{m}{t-1}}{\binom{m}{t}} \leq \frac{1}{k^{i+1}}$$

and therefore

$$\frac{\Phi_{t-1}(m)}{\binom{m}{t}} = \sum_{i=0}^{t-1} \frac{\binom{m}{t-i-1}}{\binom{m}{t}} \leq \frac{1}{k} \cdot \sum_{i=0}^{t-1} \frac{1}{k^i} < \frac{1}{k} \cdot \frac{1}{1-\frac{1}{k}} = \frac{1}{k-1}.$$

$\boldsymbol{t_k(m) \leq \left\lfloor \frac{m+1}{k} \right\rfloor}$**:**

Since $t$ is an integer, $t > \left\lfloor \frac{m+1}{k} \right\rfloor$ is equivalent to $t > \frac{m+1}{k}$. The latter inequality implies

$$\binom{m}{t} < (k-1) \cdot \binom{m}{t-1} \leq (k-1) \cdot \Phi_{t-1}(m).$$

By (14), $t_k(m) < t$ for each $t > \left\lfloor \frac{m+1}{k} \right\rfloor$. Hence, $t_k(m) \leq \left\lfloor \frac{m+1}{k} \right\rfloor$.

$\boldsymbol{t_k(m) \leq \left\lceil \frac{m}{k+1} \right\rceil + 1}$ **for sufficiently large** $\boldsymbol{m}$**:**

Because of (14), it is sufficient to prove

$$\binom{m}{u+3} < (k-1) \cdot \Phi_{u+2}(m)$$

for $u = \left\lceil \frac{m}{k+1} \right\rceil - 1$ and sufficiently large $m$.

According to Observation A.2, it suffices to consider the case $m \equiv 0 \mod k+1$, i.e., $m = (k+1)(u+1)$. Hence, we prove

$$\binom{(k+1)(u+1)}{u+3} < (k-1) \cdot \Phi_{u+2}((k+1)(u+1))$$

for sufficiently large $m$. In the sequel, we assume that $m$ (and therefore $u$) is sufficiently large. For notational convenience, we introduce the following notion. For a polynomial $p(x) = \sum_{i=0}^{n} a_i x^i$ and $m \leq n$, let $[p(x)]_m$ be the restriction of $p(x)$ to degree $m$, i.e., $\sum_{i=0}^{m} a_i x^i$.

We show

$$
\begin{aligned}
\sum_{i=0}^{u+2} \frac{\binom{(k+1)(u+1)}{i}}{\binom{(k+1)(u+1)}{u+3}} &= \sum_{i=0}^{u+2} \frac{\binom{(k+1)(u+1)}{u+2-i}}{\binom{(k+1)(u+1)}{u+3}} \\
&= \sum_{i=0}^{u+2} \frac{(u+3)! \cdot (ku+k-2)!}{(u+2-i)! \cdot (ku+k-1+i)!} \\
&= \sum_{i=0}^{u+2} \frac{(u+3)^{\underline{i+1}}}{(ku+k-1)^{\overline{i+1}}} \\
&\overset{!}{>} \sum_{i=0}^{\infty} k^{-(i+1)} = \frac{1}{k-1}.
\end{aligned}
$$

For

$$
\Delta_i = \begin{cases} \frac{(u+3)^{\underline{i+1}}}{(ku+k-1)^{\overline{i+1}}} - k^{-(i+1)}, & \text{if } 0 \leq i \leq u+2 \\ -k^{-(i+1)}, & \text{if } u+3 \leq i \end{cases}
$$

it is sufficient to show $\sum_{i=0}^{\infty} \Delta_i > 0$. We obtain

$$
\sum_{i=0}^{4} \Delta_i \geq \frac{2k^5 + 4k^4 + 4k^3 + 2k^2 - 2k - 5}{k^6 u} - \frac{2(k^6 - k^4 + k^3 + k^3 + 3k^2 - 5k - 10)}{k^7 u^2}.
$$

In the sequel, let $5 \leq i \leq u+2$ and

$$
p = \left(u + \frac{k-1}{k}\right)\left(u + \frac{k+1}{k}\right)\left(u + \frac{k+3}{k}\right) \prod_{j=k+5}^{k+i-1} \left(u + \frac{j}{k}\right).
$$

Since

$$
\frac{(u+3)(u+2)(u+1)}{(ku+k+4)(ku+k+2)(ku+k)} \geq \frac{1}{k^3},
$$

we get

$$
\begin{aligned}
\Delta_i &\geq \frac{u^{\underline{i-2}}}{k^3(ku+k-1)(ku+k+1)(ku+k+3)(ku+k+5)^{\overline{i-5}}} - k^{-(i+1)} \\
&= \frac{u^{\underline{i-2}} - p}{k^{i+1} p} \\
&= \frac{u^{i-2} - \left(\frac{i^2}{2} - \frac{5i}{2} + 3\right) u^{i-3} + [u^{\underline{i-2}}]_{i-4}}{k^{i+1} p} \\
&\quad - \frac{u^{i-2} + \frac{1}{2k}\left(i^2 + (2k-1)i - (4k+14)\right) u^{i-3} + [p]_{i-4}}{k^{i+1} p}
\end{aligned}
$$

$$= \underbrace{\frac{\left(-(k+1)i^2 + (3k+1)i - (2k-14)\right)u^{i-3}}{2k^{i+2}p}}_{D_1} + \underbrace{\frac{\left[u^{i-2} - p\right]_{i-4}}{k^{i+1}p}}_{D_2}$$

Since $D_1 < 0$ and $p > u^{i-2}$, we can bound $D_1$ from below as follows:

$$\begin{aligned} D_1 &= \frac{\left(-(k+1)i^2 + (3k+1)i - (2k-14)\right)u^{i-3}}{2k^{i+2}p} \\ &> \frac{-(k+1)i^2 + (3k+1)i - (2k-14)}{2k^{i+2}u}. \end{aligned}$$

To determine a lower bound on $D_2$, we give a lower bound on $\left[u^{i-2}\right]_{i-4}$ and an upper bound on $[p]_{i-4}$. It is easy to see that

$$\left[u^{i-2}\right]_{i-4} \geq 0.$$

Now, we show that

$$[p]_{i-4} \leq (i-3)(i-2)^2(k+i-1)^2k^{-}2u^{i-4},$$

if $i^2/k + i \leq u$:

$$\begin{aligned} [p]_{i-4} &\leq \left[\left(u + \frac{k+i-1}{k}\right)^{i-2}\right]_{i-4} \\ &= \sum_{j=0}^{i-4} \binom{i-2}{j} \left(\frac{k+i-1}{k}\right)^{i-2-j} u^j \\ &\leq (i-3) \max_{0\leq j\leq i-4} \binom{i-2}{j} \left(\frac{k+i-1}{k}\right)^{i-2-j} u^j \\ &\leq (i-3) \max_{0\leq j\leq i-4} (i-2)^{\min\{j,i-2-j\}} \left(\frac{k+i-1}{k}\right)^{i-2-j} u^j. \end{aligned}$$

It is easy to verify that, if $i^2/k + i \leq u$, the maximum is achieved for the maximal value of $j$. This yields

$$\begin{aligned} [p]_{i-4} &\leq (i-3)(i-2)^{\min\{i-4,2\}} \left(\frac{k+i-1}{k}\right)^2 u^{i-4} \\ &\leq (i-3)(i-2)^2(k+i-1)^2k^{-2}u^{i-4}. \end{aligned}$$

We will see later that the condition $i^2/k + i \leq u$ is fulfilled for sufficiently large $m$.

Hence, we get

$$
\begin{aligned}
D_2 &= \frac{[u^{i-2}]_{i-4} - [p]_{i-4}}{k^{i+1}p} \\
&\geq -\frac{[p]_{i-4}}{k^{i+1}p} \\
&\geq -\frac{(i-3)(i-2)^2(k+i-1)^2u^{i-4}}{k^{i+3}p} \\
&\geq -\frac{(i-3)(i-2)^2(k+i-1)^2}{k^{i+3}u^2}.
\end{aligned}
$$

Combining the derived bounds, we get

$$
\begin{aligned}
\Delta_i &\geq D_1 + D_2 \\
&> \frac{-(k+1)i^2 + (3k+1)i - (2k-14)}{2k^{i+2}u} - \frac{(i-3)(i-2)^2(k+i-1)^2}{k^{i+3}u^2}.
\end{aligned}
$$

Let $5 \leq l \leq u+2$. (A suitable value for $l$ is determined later on.) Then

$$
\sum_{i=l+1}^{\infty} \Delta_i \geq \sum_{i=l+1}^{\infty} -k^{-(i+1)} = -\frac{1}{(k-1)k^{l+1}}
$$

and

$$
\begin{aligned}
\sum_{i=5}^{l} \Delta_i &\geq -\sum_{i=5}^{l} \frac{(k+1)i^2 - (3k+1)i + (2k-14)}{2k^{i+2}u} \\
&\quad - \sum_{i=5}^{l} \frac{(i-3)(i-2)^2(k+i-1)^2}{k^{i+3}u^2} \\
&\geq -\sum_{i=5}^{l} \frac{(k+1)i^2 - (3k+1)i + (2k-14)}{2k^{i+2}u} \\
&\quad - \sum_{i=5}^{\infty} \frac{(i-3)(i-2)^2(k+i-1)^2}{k^{i+3}u^2} \\
&= -\frac{6k^3 - 5k^2 + 2k - 1}{k^6(k-1)^3u} \\
&\quad + \frac{(k^3 - k^2 - k + 1)l^2 - (k^3 - 5k^2 + 3k + 1)l - (14k^2 - 32k + 14)}{2k^{l+2}(k-1)^3u} \\
&\quad - \frac{18k^7 + 84k^6 - 14k^5 - 108k^4 + 384k^3 - 400k^2 + 192k - 36}{k^7(k-1)^6u^2}.
\end{aligned}
$$

Finally, we set $l = 2\log_k(u)$ and get

$$\begin{aligned}
\sum_{i=0}^{\infty} \Delta_i &= \sum_{i=0}^{4} \Delta_i + \sum_{i=5}^{2\log_k(u)} \Delta_i + \sum_{i=2\log_k(u)+1}^{\infty} \Delta_i \\
&> \frac{2k^5 + 4k^4 + 4k^3 + 2k^2 - 2k - 5}{k^6 u} - \frac{6k^3 - 5k^2 + 2k - 1}{k^6(k-1)^3 u} - O\left(\frac{1}{u^2}\right) \\
&= \frac{2k^8 - 2k^7 - 2k^6 - 3k^3 + 12k^2 - 15k + 6}{k^6(k-1)^3 u} - O\left(\frac{1}{u^2}\right) \\
&\geq \frac{2k^8 - 2k^7 - 2k^6 - 3k^3}{k^6(k-1)^3 u} - O\left(\frac{1}{u^2}\right) \\
&\geq \frac{2k^2 - 2k - \frac{19}{8}}{(k-1)^3 u} - O\left(\frac{1}{u^2}\right) \\
&\geq \frac{13}{8(k-1)^3 u} - O\left(\frac{1}{u^2}\right) \\
&> 0,
\end{aligned}$$

if $m$ (and therefore $u$) is sufficiently large.

Note that the chosen value for $l$ implies that the condition $i^2/k + i \leq u$ is fulfilled for sufficiently large values of $m$.

•

**Proof of 2.** The proof is obtained from an explicit calculation of $t_2(m)$ for $m = 1, \ldots, 12$.

•

**Proof of 3.**

$\boldsymbol{t_2(m) \geq \lfloor m/3 \rfloor + 1}$**:**

Because of (13), it is sufficient to show

$$\binom{m}{u+1} \geq \Phi_u(m)$$

for $u = \lfloor m/3 \rfloor$ and $m \geq 13$. For $m = 13, \ldots, 17$, this can be verified explicitly.

According to Observation A.2, it is sufficient to consider the case $m \equiv 0 \mod 3$. Hence, it suffices to prove

$$\binom{3u}{u+1} \geq \Phi_u(3u)$$

or equivalently

$$\begin{aligned}
\frac{\Phi_u(3u)}{\binom{3u}{u+1}} = \sum_{i=0}^{u} \frac{\binom{3u}{u-i}}{\binom{3u}{u+1}} = \sum_{i=0}^{u} \frac{(u+1)! \cdot (2u-1)!}{(u-i)! \cdot (2u+i)!} &= \sum_{i=0}^{u} \frac{(u+1)^{\underline{i+1}}}{(2u)^{\overline{i+1}}} \\
&\stackrel{!}{\leq} \sum_{i=0}^{u} \frac{1}{2^{i+1}} < \sum_{i=0}^{\infty} \frac{1}{2^{i+1}} = 1
\end{aligned}$$

for $u \geq 6$.

$$\begin{aligned}
\sum_{i=0}^{5} \frac{(u+1)^{\underline{i+1}}}{(2u)^{\overline{i+1}}} &= \sum_{i=0}^{5} \frac{1}{2^{i+1}} - \frac{3(76u^4 - 218u^3 - 801u^2 - 1082u - 320)}{64u(2u+1)(2u+3)(u+2)(2u+5)} \\
&\leq \sum_{i=0}^{5} \frac{1}{2^{i+1}} - \frac{3u\left(76u^2 - 218u - \frac{8912}{9}\right)}{64(2u+1)(2u+3)(u+2)(2u+5)} \\
&< \sum_{i=0}^{5} \frac{1}{2^{i+1}}
\end{aligned}$$

for $u \geq 6$. Since

$$\frac{(u+1)^{\underline{i+2}}}{(2u)^{\overline{i+2}}} \leq \frac{u}{2u+1} \cdot \frac{(u+1)^{\underline{i+1}}}{(2u)^{\overline{i+1}}} < \frac{1}{2} \cdot \frac{(u+1)^{\underline{i+1}}}{(2u)^{\overline{i+1}}},$$

the terms of the sum

$$\sum_{i=0}^{u} \frac{(u+1)^{\underline{i+1}}}{(2u)^{\overline{i+1}}}$$

decrease faster than the terms of the geometric row.

**$t_2(m) \leq \lfloor m/3 \rfloor + 1$:**

By (14) it is sufficient to show

$$\binom{m}{t+1} < \Phi_t(m)$$

for $t = \lfloor m/3 \rfloor + 1$ and $m \geq 13$. Because of Observation A.2, it suffices to consider the case $m \equiv 2 \mod 3$. Hence, we have to prove

$$\binom{3t-1}{t+1} < \Phi_t(3t-1)$$

for $t \geq 5$.

For $t = 5, \ldots, 718$, one can verify the condition explicitly. For $t \geq 719$ we show

$$\begin{aligned}
\sum_{i=0}^{t} \frac{\binom{3t-1}{i}}{\binom{3t-1}{t+1}} &= \sum_{i=0}^{t} \frac{\binom{3t-1}{t-i}}{\binom{3t-1}{t+1}} \\
&= \sum_{i=0}^{t} \frac{(t+1)! \cdot (2t-2)!}{(t-i)! \cdot (2t-1+i)!} \\
&= \sum_{i=0}^{t} \frac{(t+1)^{\underline{i+1}}}{(2t-1)^{\overline{i+1}}} \\
&\overset{!}{>} \sum_{i=0}^{\infty} 2^{-(i+1)} = 1.
\end{aligned}$$

For

$$\Delta_i = \begin{cases} \frac{(t+1)^{\underline{i+1}}}{(2t-1)^{\overline{i+1}}} - 2^{-(i+1)}, & \text{if } 0 \leq i \leq t \\ -2^{-(i+1)}, & \text{if } t+1 \leq i \end{cases}$$

it is sufficient to show $\sum_{i=0}^{\infty} \Delta_i > 0$. Let $u = t-1$. We get

$$\sum_{i=0}^{3} \Delta_i = \frac{15}{16u} - \frac{21}{64u^2} - \frac{3}{64u^3} + \frac{3(29u+3)}{64u^3(2u+1)(2u+3)} > \frac{15}{16u} - \frac{21}{64u^2} - \frac{3}{64u^3}.$$

In the sequel, let $4 \leq i \leq t = u+1$ and

$$p = \left(u + \frac{1}{2}\right)\left(u + \frac{3}{2}\right)\prod_{j=5}^{i+1}\left(u + \frac{j}{2}\right).$$

We get

$$\begin{aligned}
\Delta_i &= \frac{u^{\underline{i-1}} - \left(u+\frac{1}{2}\right)\left(u+\frac{3}{2}\right)\prod_{j=5}^{i+1}\left(u+\frac{j}{2}\right)}{4(2u+1)(2u+3)(2u+5)^{\overline{i-3}}} \\
&= \frac{u^{\underline{i-1}} - p}{2^{i+1}p} \\
&= \frac{u^{i-1} - \left(\frac{i^2}{2} - \frac{3i}{2} + 1\right)u^{i-2} + \left(\frac{i^4}{8} - \frac{11i^3}{12} + \frac{19i^2}{8} - \frac{31i}{12} + 1\right)u^{i-3} + [u^{\underline{i-1}}]_{i-4}}{2^{i+1}p} \\
&\quad - \frac{u^{i-1} + \left(\frac{i^2}{4} + \frac{3i}{4} - \frac{5}{2}\right)u^{i-2} + \left(\frac{i^4}{32} + \frac{7i^3}{48} - \frac{17i^2}{32} - \frac{103i}{48} + \frac{11}{2}\right)u^{i-3} + [p]_{i-4}}{2^{i+1}p} \\
&= \underbrace{\frac{\left(-\frac{3i^2}{4} + \frac{3i}{4} + \frac{3}{2}\right)u^{i-2}}{2^{i+1}p}}_{D_1} + \underbrace{\frac{\left(\frac{3i^4}{32} - \frac{17i^3}{16} + \frac{93i^2}{32} - \frac{7i}{16} - \frac{9}{2}\right)u^{i-3}}{2^{i+1}p}}_{D_2} + \underbrace{\frac{[u^{\underline{i-1}} - p]_{i-4}}{2^{i+1}p}}_{D_3}.
\end{aligned}$$

First, we determine a lower bound on $D_1$:

$$\begin{aligned}
D_1 &= \frac{-3(i^2-i-2)u^{i-2}}{2^{i+3}p} \\
&= \frac{-3(i^2-i-2)}{2^{i+3}u} + \frac{3(i^2-i-2)}{2^{i+3}}\left(\frac{1}{u} - \frac{u^{i-2}}{p}\right) \\
&= \frac{-3(i^2-i-2)}{2^{i+3}u} + \frac{3(i^2-i-2)(p-u^{i-1})}{2^{i+3}pu} \\
&= \frac{-3(i^2-i-2)}{2^{i+3}u} + \frac{3(i^2-i-2)\left(\frac{1}{2}+\frac{3}{2}+\sum_{j=5}^{i+1}\frac{j}{2}\right)u^{i-2}}{2^{i+3}pu} + \underbrace{\frac{3(i^2-i-2)[p]_{i-3}}{2^{i+3}pu}}_{\geq 0}
\end{aligned}$$

$$
\begin{aligned}
&\geq \frac{-3(i^2-i-2)}{2^{i+3}u} + \frac{3(i^4+2i^3-15i^2+4i+20)u^{i-2}}{2^{i+5}pu} \\
&= \frac{-3(i^2-i-2)}{2^{i+3}u} + \frac{3(i^4+2i^3-15i^2+4i+20)}{2^{i+5}u^2} \\
&\quad - \frac{3(i^4+2i^3-15i^2+4i+20)[p]_{i-2}}{2^{i+5}pu^2}.
\end{aligned}
$$

$D_2$ can be rewritten as

$$
\begin{aligned}
D_2 &= \frac{(3i^4-34i^3+93i^2-14i-144)u^{i-3}}{2^{i+6}p} \\
&= \frac{3i^4-34i^3+93i^2-14i-144}{2^{i+6}u^2} - \frac{(3i^4-34i^3+93i^2-14i-144)[p]_{i-2}}{2^{i+6}pu^2}.
\end{aligned}
$$

Now, we show that

$$
[p]_{i-b} \leq (i-b+1)(i-1)^{b-1}(i+1)^{b-1}2^{-(b-1)}u^{i-b}
$$

for $b \geq 1$, if $i^2 \leq 2u-1$ (and $i \geq 3$):

$$
\begin{aligned}
[p]_{i-b} &\leq \left[\left(u + \frac{i+1}{2}\right)^{i-1}\right]_{i-b} \\
&= \sum_{j=0}^{i-b} \binom{i-1}{j} \left(\frac{i+1}{2}\right)^{i-1-j} u^j \\
&\leq (i-b+1) \max_{0\leq j\leq i-b} \binom{i-1}{j} \left(\frac{i+1}{2}\right)^{i-1-j} u^j \\
&\leq (i-b+1) \max_{0\leq j\leq i-b} (i-1)^{\min\{j,i-1-j\}} \left(\frac{i+1}{2}\right)^{i-1-j} u^j.
\end{aligned}
$$

It is easy to verify that, if $i^2 \leq 2u-1$ (and $i \geq 3$), the maximum is achieved for the maximal value of $j$. This yields

$$
\begin{aligned}
[p]_{i-b} &\leq (i-b+1)(i-1)^{\min\{i-b,b-1\}} \left(\frac{i+1}{2}\right)^{b-1} u^{i-b} \\
&\leq (i-b+1)(i-1)^{b-1}(i+1)^{b-1}2^{-(b-1)}u^{i-b}.
\end{aligned}
$$

We will show later that the condition $i^2 \leq 2u-1$ is fulfilled. This implies that

$$
\frac{[p]_{i-2}}{p} \leq \frac{(i-1)^2(i+1)u^{i-2}}{2p} \leq \frac{(i-1)^2(i+1)}{2u}.
$$

Hence, we get

$$\begin{aligned}
D_1 + D_2 &\geq \frac{-3(i^2 - i - 2)}{2^{i+3}u} + \frac{9i^4 - 22i^3 + 3i^2 + 10i - 24}{2^{i+6}u^2} \\
&\quad - \frac{(9i^4 - 22i^3 + 3i^2 + 10i - 24)[p]_{i-2}}{2^{i+6}pu^2} \\
&\geq \frac{-3(i^2 - i - 2)}{2^{i+3}u} + \frac{9i^4 - 22i^3 + 3i^2 + 10i - 24}{2^{i+6}u^2} \\
&\quad - \frac{(9i^4 - 22i^3 + 3i^2 + 10i - 24)(i-1)^2(i+1)}{2^{i+7}u^3}.
\end{aligned}$$

To bound $D_3$ from below, we derive a lower bound on $[u^{\underline{i-1}}]_{i-4}$ and an upper bound on $[p]_{i-4}$:

$$\begin{aligned}
[u^{\underline{i-1}}]_{i-4} &\geq -u^{i-4} \sum_{l=1}^{i-2} \sum_{k=1}^{l-1} \sum_{j=1}^{k-1} jkl \\
&= -\frac{u^{i-4}}{48}(i^6 - 13i^5 + 67i^4 - 175i^3 + 244i^2 - 172i - 48)
\end{aligned}$$

and (assuming $i^2 \leq 2t - 1$):

$$[p]_{i-4} \leq \frac{(i-3)(i-1)^3(i+1)^3 u^{i-4}}{8} = \frac{u^{i-4}}{48} 6(i-3)(i-1)^3(i+1)^3.$$

This yields the following lower bound on $D_3$:

$$\begin{aligned}
D_3 &\geq -\frac{(6i^7 - 17i^6 - 31i^5 + 121i^4 - 157i^3 + 190i^2 - 178i - 30)u^{i-4}}{3 \cdot 2^{i+5}p} \\
&\geq -\frac{6i^7 - 17i^6 - 31i^5 + 121i^4 - 157i^3 + 190i^2 - 178i - 30}{3 \cdot 2^{i+5}u^3}.
\end{aligned}$$

Combining the derived bounds, we get

$$\begin{aligned}
\Delta_i &= D_1 + D_2 + D_3 \\
&\geq \frac{-3(i^2 - i - 2)}{2^{i+3}u} + \frac{9i^4 - 22i^3 + 3i^2 + 10i - 24}{2^{i+6}u^2} \\
&\quad - \frac{51i^7 - 161i^6 - 76i^5 + 598i^4 - 805i^3 + 811i^2 - 610i - 192)}{3 \cdot 2^{i+7}u^3}.
\end{aligned}$$

Let $4 \leq l \leq u + 1$. (A suitable value for $l$ is determined later on.) Then

$$\sum_{i=l+1}^{\infty} \Delta_i \geq \sum_{i=l+1}^{\infty} -2^{-(i+1)} = -\frac{1}{2^{l+1}}$$

and

$$\sum_{i=4}^{l} \Delta_i \geq -\sum_{i=4}^{l} \frac{3(i^2-i-2)}{2^{i+3}u} + \sum_{i=4}^{l} \frac{9i^4-22i^3+3i^2+10i-24}{2^{i+6}u^2}$$
$$-\sum_{i=4}^{l} \frac{51i^7-161i^6-76i^5+598i^4-805i^3+811i^2-610i-192)}{3\cdot 2^{i+7}u^3}$$
$$= -\frac{15}{16u} + \underbrace{\frac{3(l+1)+3(l+1)^2}{2^{l+3}u}}_{D_4}$$
$$+ \frac{789}{64u^2} - \underbrace{\frac{12+\frac{143}{16}(l+1)+\frac{99}{32}(l+1)^2+\frac{7}{16}(l+1)^3+\frac{9}{32}(l+1)^4}{2^{l+1}u^2}}_{D_5}$$
$$- \frac{137729}{16u^3} + \underbrace{\frac{\frac{34435}{4}+\frac{286397}{48}(l+1)+\frac{66261}{32}(l+1)^2+\frac{45661}{96}(l+1)^3}{2^{l+1}u^3}}_{\geq 0}$$
$$+ \underbrace{\frac{\frac{8089}{96}(l+1)^4+\frac{2171}{192}(l+1)^5+\frac{49}{48}(l+1)^6+\frac{17}{64}(l+1)^7}{2^{l+1}u^3}}_{\geq 0}$$
$$\geq -\frac{15}{16u} + \frac{789}{64u^2} - \frac{137729}{16u^3},$$

where the last inequality is true, because $D_4 - D_5 \geq 0$ for each $u \geq 718$.

Finally, we get

$$\sum_{i=0}^{\infty} \Delta_i \geq \frac{12}{u^2} - \frac{550919}{64u^3} - \frac{1}{2^{l+1}} = \frac{12}{u^2} - \frac{550951}{64u^3} > 0$$

for $l = 3\log u$ and $u \geq 718$.

The condition $i^2 \leq 2u-1$ immeadeately follows from $i \leq 3\log u$ and $u \geq 718$.

•

**Proof of 4.**

$\boldsymbol{t_k(m+1) \geq t_k(m)}$**:**

Let $t = t_k(m)$. By (BP1), $\binom{m}{t} \geq (k-1)\cdot\Phi_{t-1}(m)$. According to Observation A.2, this implies that $\binom{m+1}{t} \geq (k-1)\cdot\Phi_{t-1}(m+1)$. From (13), we get $t_k(m+1) \geq t = t_k(m)$.

$\boldsymbol{t_k(m+1) \leq t_k(m)+1}$**:**

Because of (14), it is sufficient to show

$$\binom{m+1}{t} < (k-1)\cdot\Phi_{t-1}(m+1)$$

for $t = t_k(m) + 2$. Let $t = t_k(m) + 2$. The following calculation, which applies condition (BP2) to $t_k(m)$, proves the stated inequality:

$$
\begin{aligned}
\binom{m+1}{t} &= \frac{m+1}{t}\binom{m}{t-1} = \frac{m+1}{t}\binom{m}{t_k(m)+1} \\
&< \frac{m+1}{t} \cdot (k-1) \cdot \Phi_{t_k(m)}(m) = \frac{m+1}{t} \cdot (k-1) \cdot \Phi_{t-2}(m) \\
&= (k-1) \cdot \sum_{i=0}^{t-2} \frac{m+1}{t}\binom{m}{i} \\
&\le (k-1) \cdot \sum_{i=0}^{t-2} \frac{m+1}{i+1}\binom{m}{i} \\
&= (k-1) \cdot \sum_{i=0}^{t-2} \binom{m+1}{i+1} \\
&= (k-1) \cdot \sum_{i=1}^{t-1} \binom{m+1}{i} \\
&< (k-1) \cdot \Phi_{t-1}(m+1).
\end{aligned}
$$

•

# References


[1] Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[2] Peter Auer and Philip M. Long. Simulating access to hidden information while learning. In *Proceedings of the 26th Annual Symposium on Theory of Computing*, pages 263–272, 1994. Full paper will appear in Machine Learning.

[3] Ronald Lewis Graham, Donald E. Knuth, and Oren Patashnik. *Concrete Mathematics*. Addison-Wesley, 2nd edition, 1994.

[4] Wolfgang Maass and György Turán. Lower bound methods and separation results for on-line learning models. *Machine Learning*, 9:107–145, 1992.

[5] T. M. Mitchell. Generalization as search. *Artificial Intelligence*, 18:203–226, 1982.

[6] N. Sauer. On the density of families of sets. *Journal of Combinatorial Theory (A)*, 13:145–147, 1972.

[7] V. N. Vapnik and A. Ya. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theor. Probability and Appl.*, 16(2):264–280, 1971.