Structural Results about Exact Learning with Unspecified Attribute Values

Andreas Birkendorf* **Norbert Klasner** **Christian Kuhlmann†** **Hans U. Simon‡**
Lehrstuhl Mathematik und Informatik
Fakultät für Mathematik
Ruhr-Universität Bochum
D-44780 Bochum
birkendo,klasner,kuhlmann,simon@lmi.ruhr-uni-bochum.de

Abstract

This paper deals with the UAV learning model of Goldman, Kwek and Scott [7], where "UAV" is the acronym for "Unspecified Attribute Values". As in [7], we consider exact learning within the UAV framework. A smooth transition between exact learning in the UAV setting and standard exact learning is obtained by putting a fixed bound r on the number of unspecified attribute values per instance. For $r = 0$, we obtain the standard model. For $r = n$ (the total number of attributes), we obtain the (unrestricted) UAV model. Between these extremes, we find the hierarchies $(\text{UAV-MQ}_r)_{0\le r\le n}$, $(\text{UAV-EQ}_r)_{0\le r\le n}$, and $(\text{UAV-ARB-EQ}_r)_{0\le r\le n}$.

Our main results are as follows. We present various lower bounds on the number of ARB-EQs and UAV-MQs in terms of the Vapnik Chervonenkis dimension of the concept class. We show furthermore that a natural extension of Angluin's Sunflower Lemma [1] is still applicable in the exact UAV learning model. Our *UAV Sunflower Lemma* allows to establish exponentially large lower bounds on the necessary number of UAV-MQs for several popular concept classes. On the other hand, we can show that slight simplifications of these classes are efficiently learnable using only few UAV-MQs. Finally, we investigate the inherent structure of the aforementioned three hierarchies and the relations between them. It turns out that query type UAV-EQ_{r-1} is strictly stronger than UAV-EQ_r (for each constant r). The analogous result for UAV-ARB-EQ is valid. Furthermore, $\text{UAV-MQ}_{r+\omega(\log n)}$ is strictly stronger than UAV-MQ_r. We also determine the relation between query types chosen from different hierarchies.

1 INTRODUCTION

Recently, Goldman, Kwek, and Scott introduced the so-called *UAV*[1] *learning model* [7]. 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 "unknown" or "ambiguous").

A learning model capturing unspecified attribute values is appropriate in situations, where the determination of all attribute values is either impossible or prohibitively expensive. Situations of this kind arise, e.g., in the data security sector, when some states may prohibit the release of a certain credit information (as discussed in [7]). Similarly, in the area of medical diagnosis, it is sometimes too expensive (or dangerous) to perform all measurements that are related to a certain disease.

The UAV learning model has several connections to other models dealing with "incomplete" data. Goldman et al. gave a detailed survey on relations to models like *learning with restricted focus of attention* [3, 4], *learning probabilistic concepts* [8], *learning with a consistently ignorant teacher* [6] or *learning with a constrained instance* or *projective equivalence oracle* [2].

In the standard *exact learning model*, introduced by Angluin [1], an unknown target concept f drawn from

*The author gratefully acknowledge the support of Deutsche Forschungsgemeinschaft grant Si 498/3-2.

†The author gratefully acknowledge the support of the German-Israeli Foundation for Scientific Research and Development grant I-403-001.06/95.

‡The author gratefully acknowledge 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.

[1] UAV=Unspecified Attribute Values

a (known) *concept class* $\mathcal{C}$ must be exactly identified by means of membership and equivalence queries. A membership query (MQ) for an instance x is answered $f(x)$. The answer to an equivalence query for a hypothesis h is "YES", if $h \equiv f$ (correct hypothesis), and is of the form $(x, f(x))$, where $f(x) \neq h(x)$, otherwise. Equivalence queries with hypotheses chosen from the concept class are denoted by EQ and equivalence queries with arbitrary hypotheses are denoted by ARB-EQ.

The investigations of Goldman et al. focus on Boolean concept classes. In the sequel, $\mathcal{B}_n$ denotes the set of all functions of the form $f : \{0,1\}^n \to \{0,1\}$. Note that on the Boolean domain, each UAV instance $a \in \{0,1,*\}^n$ represents a subcube of the n-dimensional cube $\{0,1\}^n$. This subcube is denoted as $X(a)$. According to the general rule (described above), the classification of a with respect to a function $f \in \mathcal{B}_n$ is

$$f(a) := \begin{cases} b\ , & \text{if } f(x) = b \text{ for all } x \in X(a); \\ *\ , & \text{otherwise.} \end{cases}$$

As usual, a is called a *positive* (or *negative*, respectively) *UAV example* for concept f, if $f(a) = 1$ (or $f(a) = 0$, respectively).

Goldman et al. adapted the exact learning model to the UAV framework. Let $\mathcal{C}_n \subseteq \mathcal{B}_n$ denote the concept class. The learner gathers information about the unknown target concept $f \in \mathcal{C}_n$ via *UAV membership queries* (UAV-MQs) or *UAV equivalence queries* (UAV-EQs or UAV-ARB-EQs). When the learner performs an UAV-MQ for UAV instance $a \in \{0,1,*\}^n$, which is formally written as UAV-MQ(a), it will receive the correct UAV classification $f(a) \in \{0,1,*\}$. An UAV-EQ for hypothesis $h \in \mathcal{C}_n$, formally written as UAV-EQ(h), leads to answer "YES", if $h \equiv f$ (correct hypothesis), and to an answer of the form $(a, f(a))$, where $f(a) \neq h(a)$, otherwise. In the latter case, a is called *counterexample* to hypothesis h. Query UAV-ARB-EQ(h) with a hypothesis $h \in \mathcal{B}_n$ is answered in the same way.

For each $r \geq 0$, query types UAV-MQ$_r$, UAV-EQ$_r$, and UAV-ARB-EQ$_r$ refer to a modified UAV learning model, where the UAV instances are restricted to have no more than r unspecified attribute values. This model coincides with the exact UAV learning model for $r = n$ and with the ordinary exact learning model for $r = 0$.

In order to (roughly) distinguish learnable from unlearnable classes, we use the following definition. Let $\mathcal{Q}$ be a collection of UAV query types and $(\mathcal{C}_n)_{n\geq 1}$ a (parameterized) concept class. We say that $(\mathcal{C}_n)_{n\geq 1}$ is *UAV learnable using* $\mathcal{Q}$ (or simply $\mathcal{Q}$ *learnable*) if there is an algorithm A which achieves exact identification for each $n \geq 1$ and each target concept $f \in \mathcal{C}_n$ using only poly(n) many queries of a type in $\mathcal{Q}$. (Note that "unlearnable" Boolean classes can still be "learned" using superpolynomially many queries.)

Note that this definition is purely information-theoretic and does not address computational issues. It will turn out, however, that all learning algorithms, which are designed in the course of this paper, can be implemented quite efficiently.

The following relation partially orders collections $\mathcal{Q}$, $\mathcal{Q}'$ of query types according to their "learning power". We say that $\mathcal{Q}'$ is *stronger* than $\mathcal{Q}$ ($\mathcal{Q} \preceq \mathcal{Q}'$), if $\mathcal{Q}$ learnability implies $\mathcal{Q}'$ learnability for all concept classes $(\mathcal{C}_n)_{n\geq 1}$. If $\mathcal{Q} \preceq \mathcal{Q}'$ and there exists a class $(\mathcal{C}_n)_{n\geq 1}$ which is $\mathcal{Q}'$ learnable but not $\mathcal{Q}$ learnable, we say that $\mathcal{Q}'$ is *strictly stronger* than $\mathcal{Q}$ ($\mathcal{Q} \prec \mathcal{Q}'$). Furthermore, we write $\mathcal{Q} \sim \mathcal{Q}'$ if $\mathcal{Q} \preceq \mathcal{Q}'$ and $\mathcal{Q}' \preceq \mathcal{Q}$. If neither $\mathcal{Q} \preceq \mathcal{Q}'$ nor $\mathcal{Q}' \preceq \mathcal{Q}$ holds, we say that $\mathcal{Q}$ and $\mathcal{Q}'$ are *incomparable*, and we write $\mathcal{Q} \not\preceq \mathcal{Q}'$ if $\mathcal{Q}$ learnability does not imply $\mathcal{Q}'$ learnability.

Trivially, the following relations are valid for all $0 \leq r \leq n-1$:

$$\begin{aligned} \text{MQ} &= \text{UAV-MQ}_0 \preceq \text{UAV-MQ}_r \\ &\preceq \text{UAV-MQ}_{r+1} \\ &\preceq \text{UAV-MQ}_n = \text{UAV-MQ}, \\ \text{UAV-EQ} &= \text{UAV-EQ}_n \preceq \text{UAV-EQ}_{r+1} \\ &\preceq \text{UAV-EQ}_r \preceq \text{UAV-EQ}_0 = \text{EQ}, \\ \text{UAV-ARB-EQ} &= \text{UAV-ARB-EQ}_n \\ &\preceq \text{UAV-ARB-EQ}_{r+1} \\ &\preceq \text{UAV-ARB-EQ}_r \\ &\preceq \text{UAV-ARB-EQ}_0 = \text{ARB-EQ}, \\ \text{UAV-EQ}_r &\preceq \text{UAV-ARB-EQ}_r. \end{aligned}$$

In their work, Goldman et al. observed some other relations. For instance, UAV-MQ$_{O(\log n)} \sim$ MQ and (MQ, UAV-EQ$_{O(\log n)}$) $\sim$ (MQ, EQ) $\preceq$ (UAV-MQ, UAV-EQ).

Before we outline the structure of our paper we need some definitions.

Let $\mathcal{C}$ be a class of concepts $f\colon X \to \{0,1\}$. A sequence $S = (x_1, \ldots, x_d) \in X^d$ is called *shattered by* $\mathcal{C}$ if

$$\forall b \in \{0,1\}^d \, \exists f = f_b \in \mathcal{C} \, \forall i = 1, \ldots, d\colon f(x_i) = b_i.$$

The Vapnik Chervonenkis dimension of $\mathcal{C}$ ($VCdim(\mathcal{C})$ for short) is the maximal number d (possibly ∞) such that there exists a shattered sequence of length d.

A *monomial* is a conjunction of literals, where a *literal* is a Boolean variable or its negation. A monomial m is called an *implicant* of f if for each assignment a to the Boolean variables $m(a) = 1$ implies $f(a) = 1$. An implicant m is called *prime implicant* if no literal can be removed from m without destoying the property of an implicant.

A formula in *disjunctive normal form* is a disjunction of monomials. DNF$_n$ is the class of all formulae in disjunctive normal form over n variables. We consider the following restrictions of DNF$_n$.

k-DNF$_n$ is the class of all formulae of DNF$_n$ where each monomial consists of at most k literals. The class of all concepts of DNF$_n$ with at most k monomials is denoted by k-term DNF$_n$. Monotone DNF$_n$ is the class of all formulae from DNF$_n$ whose literals are unnegated variables. The class of formulae of DNF$_n$ where each variable appears at most once (twice, respectively) is called read-once DNF$_n$ (read-twice DNF$_n$, respectively).

A *decision list* is an ordered list of pairs $[(m_1, b_1), \ldots, (m_r, b_r)]$, where each m_j is a monomial and each b_j is a Boolean value called label. A pair (m_j, b_j) is satisfied

by an assignment $a \in \{0,1\}^n$ if $m_j(a) = 1$. The value of a decision list on the assignment a is determined by the label of the first item in the list which is satisfied by a. To ensure that at least one item is always satisfied, we assume that the last item is of the form $(1, b)$, where 1 is the monomial which is always satisfied. A k-decision list is a decision list where all monomials m_j consist of at most k literals. We denote by k-DL$_n$ the class of all k-decision lists over n variables.

We omit the subscript n, if the number of variables is clear from context.

Our paper extends the work of Goldman et al. in several ways. In Section 2, we present lower bounds on the number of ARB-EQs and UAV-MQs in terms of the Vapnik Chervonenkis dimension of the concept class. We show furthermore that a natural extension of Angluin's Sunflower Lemma [1], a well known technique for proving lower bounds on the number of queries in the ordinary exact learning model, is still applicable in the exact UAV learning model. Our *UAV Sunflower Lemma* allows to establish exponentially large lower bounds on the number of UAV-MQs needed to achieve exact identification of Boolean concepts from the following classes: monotone DNF, read-twice DNF, $O(n)$-term DNF and 2-DL. This is contrasted by positive learning results in Section 3, where efficient UAV-MQ learning algorithms for read-once DNF, constant-term DNF and 1-DL are presented (in addition to some other positive learning results). In Section 4, we establish three proper hierarchies: the UAV-MQ, UAV-EQ and UAV-ARB-EQ hierarchy. In particular, we show that the following relations are valid: UAV-EQ$_r \prec$ UAV-EQ$_{r-1}$, UAV-ARB-EQ$_r \prec$ UAV-ARB-EQ$_{r-1}$, UAV-MQ$_{r(n)} \prec$ UAV-MQ$_{r(n)+\omega(\log n)}$. In Section 5, we completely determine all relations between these hierarchies: for each two query types chosen from two distinct (out of the three) hierarchies, we have determined whether they are equivalent or independent, or whether one is strictly stronger than the other. In addition, we show that UAV-MQ $\not\preceq$ (EQ, MQ) holds. This allows to strengthen (MQ, EQ) $\preceq$ (UAV-MQ, UAV-EQ) to (EQ, MQ) $\prec$ (UAV-EQ, UAV-MQ).

2 GENERAL LOWER BOUNDS AND APPLICATIONS

In this section, we identify a concept c of a concept class $\mathcal{C}$ with the set of its positive examples.

Observation 2.1 *The number of UAV-MQs, needed to learn $\mathcal{C}$, is at least* $\log|\mathcal{C}|/\log 3 \geq VCdim(\mathcal{C})/\log 3 > 0.63 \cdot VCdim(\mathcal{C})$.

Proof Each UAV-MQ can be answered by an adversary, such that at least a fraction of $1/3$ of the concepts remain in the version space[2]. This yields a lower bound of $\log|\mathcal{C}|/\log 3$ queries. Furthermore, $\log|\mathcal{C}| \geq VCdim(\mathcal{C})$. •

[2] i.e., the set of concepts, which are conform to all answers received so far

The next lemma shows that the lower bound within Observation 2.1 cannot be (substantially) improved in general, because there exists a class for which it is (approximately) tight.

Lemma 2.2 *For each $k \geq 1$, there is a class $\mathcal{C}$ of size 3^k and $VCdim(\mathcal{C}) = \lfloor k \log 3 \rfloor$ which can be learned using $k = \log|\mathcal{C}|/\log 3 < (VCdim(\mathcal{C}) + 1)/\log 3$ UAV-MQs.*

Proof First, we show that, for each $k \geq 1$, there is a class $\mathcal{C}' = \mathcal{C}_{\lfloor \log k \rfloor + 1}$ which can be learned using $k = \log|\mathcal{C}'|/\log 3$ UAV-MQs.

Fix an $k \geq 1$. We enumerate the concepts of $\mathcal{C}'$ by ternary strings of length k, i.e., strings from $\{0,1,2\}^k$. Let

$$\begin{aligned} d_0^i &= \emptyset \\ d_1^i &= \{bin(i)0\} \\ d_2^i &= \{bin(i)0, bin(i)1\} \end{aligned}$$

where $bin(i)$ is the binary representation of i using $\lfloor \log k \rfloor$ bits. Now, let

$$c_{a_{k-1}\ldots a_0} = \bigcup_{i=0}^{k-1} d_{a_i}^i$$

be the concepts of $\mathcal{C}'$. Figure 1 shows a simple learning algorithm for $\mathcal{C}'$ using k UAV-MQs. In each iteration of the algorithm, one of the k ternary digits, specifying the target concept, is learned.

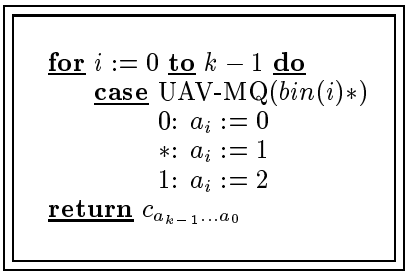

Figure 1: Algorithm which learns $\mathcal{C}'$ using k UAV-MQs.

Since $VCdim(\mathcal{C}') = k < \lfloor k \log 3 \rfloor = VCdim(\mathcal{C})$, we cannot use this class immediately for the second claim. Hence, we combine $\mathcal{C}'$ with a class $\mathcal{C}''$ of VC-dimension $\lfloor k \log 3 \rfloor$. $\mathcal{C}''$ will have the same domain

$$X = \{0,1\}^{1+\lfloor \log k \rfloor}$$

as $\mathcal{C}'$ and is obtained as follows. Pick a subset Y of size $\lfloor k \log 3 \rfloor$ from X and choose $\mathcal{C}''$ as powerset of Y augmented with some extra-subsets of X in order to achieve $|\mathcal{C}''| = 3^k$. Clearly, $VCdim(\mathcal{C}'') = \lfloor k \log 3 \rfloor$. (Note that the VC-dimension of a class with 3^k concepts cannot exceed $\lfloor k \log 3 \rfloor$.)

Now $\mathcal{C}'$ and $\mathcal{C}''$ are combined as follows. Enumerate the concepts of both classes from 1 to 3^k, respectively. Now, let

$$\mathcal{C} = \left\{ 0c_i' \cup 1c_i'' \;\middle|\; \begin{array}{l} c_i' \text{ is concept number } i \text{ of } \mathcal{C}', \\ c_i'' \text{ is concept number } i \text{ of } \mathcal{C}'' \end{array} \right\},$$

where $ac = \{ax \mid x \in c\}$ for $a \in \{0,1\}$ and $c \subseteq X$. With a slight modification (ask UAV-MQ($0bin(i)*$) instead of UAV-MQ($bin(i)*$)), the algorithm given above learns $\mathcal{C}$ using k UAV-MQs. Since $VCdim(\mathcal{C}) = VCdim(\mathcal{C}'')$, the result follows. •

Simon and Klasner [12] proved the following lower bound.

Theorem 2.3 ([12]) *The total number of ARB-EQs and UAV-MQs, needed to learn a class $\mathcal{C}$, is at least $(1 - \log_3 2)VCdim(\mathcal{C})$.*

The following observation shows that the general lower bound within Theorem 2.3 cannot be (substantially) improved in general, because there exists a class for which it is (approximately) tight.

Observation 2.4 *There is a class $(\mathcal{C}_n)_{n\geq 1}$ with $VCdim(\mathcal{C}_n) \geq n$ which can be learned using at most $(\alpha + o(1))VCdim(\mathcal{C}_n)$ EQs and UAV-MQs, where $\alpha = 1 - \log_3 2 < 0.3691$.*

The observation follows from a slight adaptation of the proof of [11, Proposition 6.3].

Angluin [1] provided a tool for showing non-learnability results in the query-learning model: the so-called "Sunflower Lemma". The lower bound, stated in this lemma, does not hold in general when we substitute the stronger UAV-MQs for the classical MQs. But, assuming a simple "Subcube Property", we are able to formulate an "UAV Sunflower Lemma" providing the same lower bound as the classical lemma.

Lemma 2.5 (UAV Sunflower Lemma) *Let $\mathcal{C}$ be a class containing distinct concepts $c_1, \ldots, c_N \in \mathcal{B}_n$ with the following properties.*

Common Intersection *There exists $c_\cap \in \mathcal{B}_n$ such that for every distinct pair $i, j \in \{1 \ldots N\}$: $c_\cap = c_i \cap c_j$*

Subcube Property *Let $c_i' = c_i \setminus c_\cap$. Every subcube X of $\{0,1\}^n$, which contains elements of c_i' and c_j' for two distinct indices i and j, also contains at least one element of $c_\cap$.*

Query Constraints *The learner may ask UAV-MQs, EQs and subset queries[3], where the EQs refer to a hypothesis class $\mathcal{H}$ not containing $c_\cap$ (which holds, in particular, if $\mathcal{H} = \mathcal{C}$ and $c_\cap \notin \mathcal{C}$).*

Then the total number of queries, needed to learn $\mathcal{C}$, is at least $N - 1$. The same lower bound applies (regardless of how $\mathcal{H}$ is chosen), when the learner may ask UAV-MQs only.

Proof We describe an adversary which answers to each query in such a way that at most one concept of $c_1, \ldots, c_N$ can be eliminated from the version space. Hence, at least $N - 1$ queries are necessary in general to achieve exact identification. The answers of the adversary to EQs and subset queries are already described in [1]. Hence, we focus on answers to UAV-MQs.

Let V be the subset of concepts from $\{c_1, \ldots, c_N\}$ which are still in the version space. Initially, $V = \{c_1, \ldots, c_N\}$. Let X be the subcube corresponding to the UAV instance of the learners actual UAV-MQ. Now, one of the following three cases arises.

Case 1: X is disjoint to c_i' for all $c_i \in V$.
Then, the adversary (arbitrarily) picks a concept c from V and answers according to c. Since all concepts in V coincide on X, no concept is removed from V.

Case 2: X is not disjoint to c_i' for exactly one $c_i \in V$.
Then, the adversary (arbitrarily) picks a concept c from $V \setminus \{c_i\}$ and answers according to c. At most c_i is removed from V.

Case 3: X contains elements $x_i \in c_i', x_j \in c_j'$ for two distinct concepts $c_i', c_j' \in V$.
According to the Subcube Property, X contains an element $x_\cap \in c_\cap$. Note that each concept in V contains $x_\cap$, but not both x_i and x_j. Thus, the adversary answers $*$, and no concept is removed from V. •

In the rest of this section, the UAV Sunflower Lemma is applied to some concrete classes.

Let $par_n = \{x \in \{0,1\}^n \mid x_1 \oplus \ldots \oplus x_n = 1\}$ denote the set of all positive examples for the parity function over n Boolean variables x_i. A simple application of the UAV Sunflower Lemma shows that the class *dirty-par*$_n := \{par_n \cup \{y\} \mid y \in \{0,1\}^n \setminus par_n\}$ is hard to learn using only UAV-MQs.

Lemma 2.6 *1. Consider the same types of queries as in the UAV Sunflower Lemma. Let $\mathcal{H}_n$ be a hypothesis class not containing par_n (e.g.,* dirty-par$_n$*). Then the total number of queries, necessary and sufficient to learn* dirty-par$_n$ *with hypotheses from $\mathcal{H}_n$, is $2^{n-1} - 1$.*

2. The number of UAV-MQs, necessary and sufficient to learn dirty-par$_n$ *is $2^{n-1} - 1$.*

Proof Let $Y = \{0,1\}^n \setminus par_n$. Class *dirty-par*$_n$ contains the 2^{n-1} concepts of the form $c_y = par_n \cup \{y\}, y \in Y$. If $x, y \in Y, x \neq y$, then $c_x \cap c_y = par_n$. Thus, the first assumption, needed to apply the UAV Sunflower Lemma, is satisfied. In order to check the Subcube Property, observe that $c_y' := c_y \setminus \{par_n\} = \{y\}$. Let $x, y \in Y, x \neq y$. Each subcube X containing x and y necessarily contains an element of par_n. Thus, the UAV Sunflower Lemma can be applied, and lower bound $2^{n-1} - 1$ follows. On the other hand, $2^{n-1} - 1$ UAV-MQs are sufficient because they allow for an exhaustive search through *dirty-par*$_n$. •

Class *dirty-par*$_n$ is not interesting in its own right. We will however see in Section 5, that it can be used to show that UAV-EQ $\not\preceq$ UAV-MQ.

Another application of the UAV Sunflower Lemma extends a result of [1] to UAV-MQs.

[3] Subset queries are not of general interest in this paper. We mention them just for sake of completeness. See [1] for a definition.

Theorem 2.7 *The total number of UAV-MQs, needed to learn one of the following classes (or even the intersection of these classes) over domain* $\{0,1\}^{2n}$*, is at least* $2^n - 1$*:*

- *monotone* DNF_{2n}
- *read-twice* DNF_{2n}
- $(n+1)$*-term* DNF_{2n}
- 2-DL_{2n}

Proof Let class $\mathcal{C}_{2n}$ consist of the 2^n formulas $c_t := c_\cap \vee t$, where $c_\cap = x_1x_{1+n} \vee x_2x_{2+n} \vee \ldots \vee x_nx_{2n}$ and t is a term of the form $p_1p_2 \ldots p_n$, where $p_i \in \{x_i, x_{i+n}\}$. Note, that each $c_t \in \mathcal{C}_{2n}$ is a monotone read-twice $(n+1)$-term DNF, which can also be written as 2-DL:

$$[(x_1x_{1+n}, 1), \ldots, (x_nx_{2n}, 1), (\overline{p}_1, 0), \ldots, (\overline{p}_n, 0), (1,1)]$$

The following observations show that $\mathcal{C}_{2n}$ satisfies the assumptions of the UAV Sunflower Lemma, implying lower bound $2^n - 1$ on the number of UAV-MQs.

Obviously, the intersection of any two distinct concepts yields $c_\cap$. Furthermore, any set $c_t \setminus c_\cap$ contains only the instance $x(t) = (q_1, \ldots, q_n, 1-q_1, \ldots, 1-q_n)$, where

$$q_i = \begin{cases} 1, & \text{if } p_i = x_i \\ 0, & \text{if } p_i = x_{i+n} \end{cases}.$$

Let us consider two concepts c_t and $c_{t'}$, where $t = p_1 \ldots p_n \neq t' = p'_1 \ldots p'_n$. Then, there is an $i \in \{1, \ldots, n\}$ such that $p_i \neq p'_i$, i.e., $p_i = x_i$ and $p'_i = x_{i+n}$ or vice versa. It follows that any subcube X containing $x(t)$ and $x(t')$ is covered by the monomial x_ix_{i+n}. Hence, X contains an element of $c_\cap$. •

3 POSITIVE LEARNING RESULTS

This section demonstrates the high "learning power" of UAV-MQs. The main results are that classes read-once DNF_n and 1-DL_n both are learnable using $n(n+1)$ UAV-MQs, respectively.

Remember that $X(a)$ denotes the subcube induced by UAV instance a. For sake of brevity, we write $a \subseteq b$ when $X(a) \subseteq X(b)$. A powerful tool in the design of UAV-MQ learning algorithms is provided by two simple procedures, called *Contract* and *Expand*, which achieve the following:

Contract Let f be the target concept and a an UAV instance with $m \geq 0$ unspecified attribute values (UAVs). Assume that $f(a) \neq 0$. When Procedure *Contract* is applied to a, it stops after $O(m)$ steps and returns an UAV example, denoted as $C_f(a)$ or simply $C(a)$, which satisfies $C(a) \subseteq a$ and $f(C(a)) = 1$. *Contract* uses at most m UAV-MQs.

Expand Let f be the target concept and a an UAV instance with m specified attribute values (SAVs). Assume that $f(a) = 1$. When Procedure *Expand* is applied to a, it stops after $O(m)$ steps performing at most m UAV-MQs. The returned UAV instance, denoted as $E_f(a)$ or simply $E(a)$, satisfies $E(a) \supseteq a$, $f(E(a)) = 1$ and $f(b) \neq 1$ for each $b \supset E(a)$.

Clearly, $C(a)$ represents an implicant of f and $E(a)$ a prime implicant.

The implementation of these procedures is fairly straightforward. *Contract* iteratively modifies input a. In each iteration, one UAV of a is converted to either 0 or 1, thereby maintaining property $f(a) \neq 0$. After at most m iterations, the resulting UAV instance is a positive UAV example of f. Each iteration takes constant time and uses one UAV-MQ.
Expand iteratively modifies input a as follows. In each iteration, one SAV of a is tested for possible conversion to $*$ without violating property $f(a) = 1$. Since this is done as long as possible, it ends with a maximal positive UAV example of f. Each iteration takes constant time and uses one UAV-MQ.

We present two simple applications of *Contract* and *Expand*. Let $singleton_n = \{l_1 \wedge \ldots \wedge l_n | l_i \in \{x_i, \overline{x}_i\}\}$.

Observation 3.1 *The class* $singleton_n$ *is learnable using* n *UAV-MQs.*

Proof For each $f \in singleton_n$, $C_f(*\ldots*)$ is the unique singleton in $\{0,1\}^n$ that is set to 1 by f. •

Let $monomial_n$ be the class of all monomials over n Boolean variables.

Observation 3.2 *The class* $monomial_n$ *is learnable using* $2n$ *UAV-MQs.*

Proof First check, whether target monomial f is constant using UAV-MQ$(*\ldots*)$. If so, we are done. Otherwise, $E_f(C_f(*\ldots*))$ identifies f. A carefully implemented computation of $E_f(C_f(*\ldots*))$ consumes $2n-1$ instead of $2n$ UAV-MQs. (The last attribute value that was changed by *Contract* need not be rechanged by *Expand*.) Thus, $2n$ UAV-MQs are used altogether. •

For the sake of simplicity, we omit subscript f in the sequel. The following more general observations concerning *Contract* and *Expand* are valid. Let f be an arbitrary (but fixed) Boolean target concept and a an UAV instance. First observe that, if $f(a) \neq 0$, then $E(C(a))$ represents a prime implicant of f. Assume that h is a hypothesis for f, which is a disjunction of some (but not all) of the prime implicants of f. The second observation is that, if $h(a) = 0$ and $f(a) \neq 0$, then $E(C(a))$ represents a prime implicant of f which is not yet covered by the prime implicants contained in h. Thus, iterative applications of $E \circ C$ would build a DNF formula for f, if we were able to provide an UAV instance a satisfying $h(a) = 0$ and $f(a) \neq 0$ in each stage of the iteration. These observations form the base of the following two results.

Lemma 3.3 *The class read-once* DNF_n *is learnable using at most* $n(n+1)$ *UAV-MQs.*

Proof Each read-once DNF formula f is uniquely represented as the disjunction of all its prime implicants. Let h denote a hypothesis which is a disjunction of some (but not necessarily all) of these prime implicants. (Initially, $h \equiv 0$.) Note that h is itself a read-once DNF. Let V_0 (or V_1, V_*, respectively) denote the set of Boolean

variables which occur negated (or unnegated, or neither negated nor unnegated, respectively) in h. Let a be the UAV instance which assigns 1 to variables in V_0, 0 to variables in V_1, and $*$ to variables in V_*. Clearly, $h(a) = 0$. Since f is read-once, each prime implicant still missing in h uses only variables from V_*. Thus, either $h \equiv f$ or $f(a) \neq 0$. If UAV-MQ(a) yields 0, we know that $h \equiv f$. Otherwise, $E(C(a))$ yields a new prime implicant. A careful implementation of the iterative computation of all prime implicants uses only $n(n+1)$ UAV-MQs. We omit the details of the implementation in this brief abstract. •

Lemma 3.3 stands in contrast to the non-learnability of read-twice DNF using UAV-MQs (Theorem 2.7) and to the non-learnability of read-once DNF using MQs ($singleton_n \subset$ read-once DNF).

Lemma 3.4 *Each Boolean concept $f \in \mathcal{B}_n$ whose total number of prime implicants is bounded by m is learnable using at most $m+1$ UAV-EQs and at most $2nm$ UAV-MQs.*

Proof Let h denote the current hypothesis (which is always a disjunction of some of the prime implicants of f, initially $h \equiv 0$), and a the counterexample returned as answer to UAV-EQ(h). Either $h(a) = 0, f(a) \neq 0$ or $h(a) = *, f(a) = 1$. In the former case, $E(C(a))$ leads to a prime implicant of f not already covered by h. In the latter case, $E(a)$ can be used for the same purpose. The lemma now easily follows. •

We remark without proof that the same kind of reasoning can be used to show that class DNF can be learned in the so-called RUAV-model (see [7] for a definition) using $2nm$ RUAV-MQs and $m+1$ RUAV-EQs, where m denotes the number of prime implicants within the DNF representation of f (as opposed to the total number of prime implicants of f). This improves on a result in [7], where $\Theta(nm^2)$ RUAV-MQs were needed.

We close this section with positive results for k-DNF and 1-DL. The proofs are fairly straightforward and therefore omitted in this brief abstract.

Lemma 3.5 *For each k, class $(k\text{-}DNF_n)_{n\geq 1}$ is learnable using $O(n^k)$ UAV-MQs.*

Lemma 3.6 *The class $1\text{-}DL_{n,q}$ is learnable using $2nq - q^2 + q \leq n(n+1)$ UAV-MQs. Here, q denotes an upper bound on the number of label-alternations needed to represent the target function as 1-DL.*

Lemma 3.6 stands in contrast to the non-learnability of 2-DL using UAV-MQs (Theorem 2.7) and to the non-learnability of 1-DL using MQs ($singleton_n \subset$ 1-DL).

4 HIERARCHY RESULTS

We already mentioned that relation UAV-EQ$_r \preceq$ UAV-EQ$_{r-1}$ is valid, i.e., equivalence queries cannot become less powerful when we decrease the bound on the number of UAVs. We aim to show that this relation is strict. We will even show the following somewhat stronger result:

Theorem 4.1 *For each $r \geq 1$, there exists a class $(\mathcal{C}_n^r)_{n\geq 1}$ of Boolean concepts with the following properties:*

1. *$\mathcal{C}_n^r$ is efficiently learnable using n queries of type UAV-EQ$_{r-1}$.*
2. *$\mathcal{C}_n^r$ is not learnable using less than $2^n - 1$ queries of type UAV-ARB-EQ$_r$ even if the learner has unlimited computational power.*

Proof The proof partitions into two parts. We first show the result for $r = 1$. Afterwards, we use a "padding argument" and obtain the result for $r \geq 2$.

Let's start with $r = 1$. We will present a class $\mathcal{C}_n = \mathcal{C}_n^1$, defined over domain $X_n = \{0,1\}^n$, which is efficiently learnable using n EQs, but not learnable using less than $2^n - 1$ queries of type UAV-ARB-EQ$_1$. Note that the latter type of query leads to a counterexample with 1 UAV. A counterexample of this form may be viewed as edge of the Boolean cube X_n. Let $N = 2^n$. The construction of $\mathcal{C}_n$ makes use of an enumeration $x_0, \ldots, x_{N-1}$ of all Boolean instances, which forms a Hamiltonian circuit in X_n. This allows the adversary to pick counterexamples of the form $[(x_i, x_{i+1 \bmod N}), l]$. $\mathcal{C}_n$ is defined as the class of all concepts of the form

$$f_i(x_j) = \begin{cases} j \bmod 2, & \text{if } j < i; \\ j+1 \bmod 2, & \text{if } j \geq i \end{cases}$$

for $i = 0, \ldots, N-1$. Note that f_0 produces label sequence $1, 0, 1, 0, \ldots, 1, 0$ on sequence $x_0, \ldots, x_{N-1}$. The label sequence of each concept f_i with $i \geq 1$ exhibits a somewhat different pattern. First, it starts with label 0 (instead of 1). Second, the perfect alternation between labels 0 and 1 is disturbed at positions $i-1, i$ where two identical labels are produced. As a side effect, the last label of the sequence must be identical to the first-one, i.e., $f_i(x_0) = f_i(x_{N-1}) = 0$ for all $i \geq 1$.

Target concept f_k can be exactly identified from n EQs by applying the obvious binary search strategy: The first hypothesis is $f_{N/2}$. If the resulting counterexample x belongs to the first half of sequence x_i then $k < N/2$. Otherwise, $k \geq N/2$. In the former case, the binary search for k proceeds in $\{0, \ldots, N/2-1\}$. In the latter case, it proceeds in $\{N/2, \ldots, N-1\}$.

In order to show that less than $2^n - 1$ queries of type UAV-ARB-EQ$_1$ are insufficient, we choose a counterexample to the current hypothesis h according to the following adversary strategy:

Case 1 $h(x_0) \neq h(x_{N-1})$.
Then counterexample $[(x_0, x_{N-1}), 0]$ is presented. Only f_0 can be eliminated from the version space.

Case 2 $h(x_0) = h(x_{N-1})$.
It follows that there exists a position $i \in \{1, \ldots, N-1\}$ such that $h(x_{i-1}) = h(x_i)$. Now, counterexample $[(x_{i-1}, x_i), *]$ is presented. Only f_i can be eliminated from the version space.

Obviously, exact identification requires at least $2^n - 1$ queries.

Let's now proceed with $r \geq 2$. We will construct a concept class $\mathcal{C} = \mathcal{C}_n^r$ which separates UAV-EQ$_{r-1}$ from UAV-ARB-EQ$_r$ (in the sense of our theorem). $\mathcal{C}$ will be a "padded version" of $\mathcal{C}_n^1$. Concepts from $\mathcal{C}$ will be defined over domain $X = X_{rn+1} = \{0,1\}^{rn+1}$. Instances in X are denoted in the form (z,b), where $z \in \{0,1\}^{rn}$ and $b \in \{0,1\}$. z and (z,b) are called *important* if $z = x^r$ for some $x \in X_n = \{0,1\}^n$, i.e., if z is the r-fold replication of a vector from X_n. The remaining instances are called *unimportant*. Note that the Hamming-distance between two different important instances z is at least r. With each $x \in X_n$, we associate the two important instances $(x^r,0),(x^r,1)$ in X. Note that any important instance can be written in this form. With each Boolean function f defined on X_n, we associate the following Boolean function f^r defined on X:

$$f^r(z,b) = \begin{cases} f(x) \oplus b, & \text{if } z = x^r; \\ 0, & \text{if } z \text{ is unimportant} \end{cases}$$

In the sequel, we call a Boolean function *well-formed* if it has the form f^r for some Boolean function defined on X_n. Focusing on important instances, we see that f^r represents two copies of f (one in subcube $X_{b=0}$, the other-one with negated labels in subcube $X_{b=1}$). We are now in the position to define $\mathcal{C} = \{f^r \mid f \in \mathcal{C}_n\}$.

We claim that $\mathcal{C}$ is efficiently learnable using n queries of type UAV-EQ$_{r-1}$. The learner A for $\mathcal{C}$ simply simulates the efficient and proper learner A_n for $\mathcal{C}_n$. If A_n issues hypothesis h, A issues hypothesis h^r. The resulting counterexample represents a subcube X' of X of dimension at most $r-1$. Since h^r is well-formed, a mistake is made on at least one important instance (x^r,b) of X'. On the other hand, this instance is unique because X' cannot contain two instances of Hamming distance exceeding $r-1$. A feeds x—which can be computed efficiently from the counterexample—into the simulation. In this way, the simulation proceeds until the target is identified exactly.

We finally claim that $\mathcal{C}$ cannot be learned from less than 2^n-1 queries of type UAV-ARB-EQ$_r$. To this end, we design an adversary strategy ADV which chooses counterexamples to hypotheses that lead to the elimination of at most one concept from the current version space. ADV will simply simulate ADV$_n$. We may assume w.l.o.g. that the current hypothesis is well-formed, and can therefore be written as h^r for some hypothesis h defined on X_n. Otherwise, the learner may receive a completely uninformative counterexample, e.g., $[(z,b),0]$, if the hypothesis labels an unimportant instance (z,b) positively, or edge $[(z,*),*]$, if z is important and $h(z,0) = h(z,1)$.

Let $[(x,y),l]$ denote the counterexample that was chosen by ADV$_n$, where (x,y) represents an edge in the Boolean cube X_n. Let C_r denote the r-dimensional subcube of $\{0,1\}^{rn}$ spanned by x^r and y^r, and $c_r \in \{0,1,*\}^{rn}$ the corresponding UAV instance with r UAVs.

We are now prepared to describe ADV. If $l \in \{0,1\}$, counterexample $[(c^r,l),0]$ is presented. Since $f(x) = f(y) = l \Leftrightarrow f^r(c^r,l) = 0$, ADV reveals the same amount of information on f as ADV$_n$. If l coincides with $*$, then $f(x) \neq f(y)$ and $h(x) = h(y) = b$ for some $b \in \{0,1\}$. The latter condition is equivalent to $h^r(c^r,b) = 0$. Now counterexample $[(c^r,b),*]$ is presented. It reveals the same amount of information on f as $[(x,y),*]$ (namely $f(x) \neq f(y)$). •

Corollary 4.2 *For all $r \geq 1$, the following holds:*

$$\begin{aligned} \textit{UAV-EQ}_{r-1} &\not\preceq \textit{UAV-ARB-EQ}_r \\ \textit{UAV-EQ}_r &\prec \textit{UAV-EQ}_{r-1} \\ \textit{UAV-ARB-EQ}_r &\prec \textit{UAV-ARB-EQ}_{r-1} \end{aligned} \quad (1)$$

Proof (1) follows immediately from Theorem 4.1. Obviously, UAV-EQ$_r \preceq$ UAV-EQ$_{r-1}$ and UAV-ARB-EQ$_r \preceq$ UAV-ARB-EQ$_{r-1}$. The strictness of these relations immediately follows from UAV-EQ$_{r-1} \not\preceq$ UAV-ARB-EQ$_r$. •

Note that Theorem 4.1 and Corollary 4.2 are proven for each constant $r \geq 1$. It is by no means clear that these results remain valid when $r = r(n)$ is a function in n. As a warning, note that the hierarchy collapses when we step from $n-1$ to n:

Lemma 4.3 *UAV-EQ$_n \sim$ UAV-EQ$_{n-1}$ and UAV-ARB-EQ$_n \sim$ UAV-ARB-EQ$_{n-1}$*

We omit the proof in this brief abstract.

We made no serious attempt to discuss the UAV-EQ or UAV-ARB-EQ hierarchy for general functions $r(n)$.

We close this section with the main result concerning the UAV-MQ-hierarchy.

Theorem 4.4 *For all functions $r = r(n)$ and $d = d(n)$ satisfying $1 \leq r(n) \leq r(n) + d(n) \leq n$, the following holds:*

1. *If $d(n) = O(\log n)$, then UAV-MQ$_r \sim$ UAV-MQ$_{r+d}$.*
2. *If $d(n) = \omega(\log n)$, then UAV-MQ$_r \prec$ UAV-MQ$_{r+d}$.*

Proof Assume that $d = O(\log n)$. Clearly, a query of type UAV-MQ$_{r+d}$ can be simulated by 2^d queries of type UAV-MQ$_r$. Since $d = O(\log n)$, factor 2^d is polynomial in n.

Assume that $d = \omega(\log n)$. Clearly, UAV-MQ$_r \preceq$ UAV-MQ$_{r+d}$. We have to present a concept class which separates the two query types.

We first consider the special case $r+d = n$. Remember that *singleton$_n$* is learnable using n queries of type UAV-MQ$_n$. However, each query of type UAV-MQ$_{n-d}$ asks for the label of an at most $(n-d)$-dimensional subcube of $\{0,1\}^n$. The subcubes of 2^d-1 such queries cover at most $2^n - 2^{n-d}$ instances in $\{0,1\}^n$. If label 0 is returned to all these queries, the at least 2^{n-d} uncovered instances still belong to the version space. Thus, the target is not yet exactly identified. Thus, at least 2^d queries of type UAV-MQ$_{n-d}$ are required. For $d = \omega(\log n)$, this is superpolynomial in n.

We now pass to the general case $0 \leq r \leq r+d \leq n$ (still assuming $d = \omega(\log n)$). Consider the class *singleton$_{r+d}$*. We may interpret this class as a Boolean function from $\mathcal{B}_n$ by adding $n-(r+d)$ dummy variables. The resulting class separates UAV-MQ$_{r+d}$ from UAV-MQ$_r$. •

5 RELATIONS BETWEEN THE HIERARCHIES

Figure 2 shows three hierarchies and the relations between them. A solid edge from $\mathcal{Q}$ to $\mathcal{Q}'$ should be interpreted as $\mathcal{Q} \prec \mathcal{Q}'$. This relation is transitive. A dashed edge from $\mathcal{Q}$ to $\mathcal{Q}'$ should be interpreted as $\mathcal{Q}' \not\preceq \mathcal{Q}$, i.e., there exists at least one concept class which is learnable using queries of type $\mathcal{Q}'$, but not using queries of type $\mathcal{Q}$. This relation is not transitive in general. The left column of Figure 2 shows the UAV-MQ hierarchy, the middle column the UAV-EQ hierarchy, and the right column the UAV-ARB-EQ hierarchy. According to Corollary 4.2, all three hierarchies are proper (which justifies the solid edges within each of these hierarchies running from the bottom to the top), and UAV-EQ$_{r-1} \not\preceq$ UAV-ARB-EQ$_r$ (which justifies the dashed edge from UAV-ARB-EQ$_r$ to UAV-EQ$_{r-1}$).

We aim to justify the remaining edges in Figure 2. The following lemma, which is based on a construction in [11], provides a useful tool for this purpose.

Lemma 5.1 *Let $p(n)$ be a polynomial. For each Boolean concept class $(\mathcal{C}_n)_{n\geq 1}$ with $|\mathcal{C}_n| \leq 2^{p(n)}$, there is a class $(\mathcal{C}_n^A)_{n\geq 1}$, called the* address extension *of $\mathcal{C}$, with the following properties:*

1. *$(\mathcal{C}_n^A)_{n\geq 1}$ is learnable in polynomial time using $p(n)$ MQs.*
2. *Let $\mathcal{Q}$ be an arbitrary query type chosen from the UAV-EQ or UAV-ARB-EQ hierarchy. Then each polynomially time bounded learner for $(\mathcal{C}_n^A)_{n\geq 1}$ using at most $q(n)$ queries of type $\mathcal{Q}$ can be converted into a polynomially time bounded learner for $(\mathcal{C}_n)_{n\geq 1}$ using at most $q(n)$ queries of type $\mathcal{Q}$. In particular, if $(\mathcal{C}_n)_{n\geq 1}$ is not $\mathcal{Q}$ learnable, then $(\mathcal{C}_n^A)_{n\geq 1}$ is not $\mathcal{Q}$ learnable.*

We omit the proof in this short abstract. The basic idea is contained in [11] (although our construction is slightly more general).

We are now prepared to argue that all edges in Figure 2 are justified:
Relation UAV-EQ $\not\preceq$ UAV-MQ (①) is witnessed by $(\textit{dirty-par}_n \cup \{\textit{par}_n\})_{n\geq 1}$. Lemma 2.6 implies that this class is not learnable using UAV-MQs. It is, however, learnable using only UAV-EQ($\textit{par}_n$).
Relation MQ $\not\preceq$ EQ [11] (②) is witnessed by the address extension of $\textit{singleton}_n$ (which is basically the construction in [11]).
Relation UAV-ARB-EQ $\not\preceq$ EQ (③) is witnessed by $(\textit{dirty-par}_n)_{n\geq 1}$. According to Lemma 2.6, this class is not learnable using EQs. However, one UAV-ARB-EQ with $\textit{par}_n$ as hypothesis suffices.
Relation MQ $\not\preceq$ UAV-ARB-EQ$_1$ (④) is witnessed by the address extension of the class $(\mathcal{C}_n^1)_{n\geq 1}$ from Theorem 4.1, which is not learnable from queries of type UAV-ARB-EQ$_1$.
Let $r \geq 0$ be constant. Obviously, UAV-EQ$_r \preceq$ UAV-ARB-EQ$_r$. The corresponding strict relation UAV-EQ$_r \prec$ UAV-ARB-EQ$_r$ (⑤) immediately follows from UAV-ARB-EQ $\not\preceq$ EQ.
We claim that ARB-EQ is strictly stronger than any other query type within the three hierarchies (⑥). This can be seen as follows. Clearly, ARB-EQ is stronger than any other query type in the UAV-EQ and UAV-ARB-EQ hierarchy. Let's compare ARB-EQ with UAV-MQ, which is the strongest query type in the UAV-MQ hierarchy. According to Lemma 2.1, a class learnable using polynomially many UAV-MQs has a polynomial upper bound on $\log|\mathcal{C}_n|$. On the other hand, the "Majority Vote" strategy succeeds learning $\mathcal{C}_n$ using only $\log|\mathcal{C}_n|$ ARB-EQs. Thus, UAV-MQ $\preceq$ ARB-EQ implying UAV-MQ$_r \preceq$ ARB-EQ for all $1 \leq r \leq n$. As yet, it follows that UAV-ARB-EQ is stronger than any other query type in Figure 2. However, since query types UAV-MQ, EQ and UAV-ARB-EQ$_1$ are obviously pairwise non-equivalent, ARB-EQ is even strictly stronger. This completes the verification of all edges in Figure 2.

Figure 2 implies that the hierarchies for UAV-MQ and UAV-EQ are completely incomparable: each query type of one hierarchy is incomparable to each query type of the other. This holds because the strongest query type in one hierarchy is not stronger than the weakest query type in the other one. The same remark applies to the hierarchies for UAV-MQ and UAV-ARB-EQ, except that ARB-EQ is strictly stronger than every other query type. The situation is different for the UAV-EQ and UAV-ARB-EQ hierarchy. But still query types UAV-EQ$_i$ and UAV-ARB-EQ$_j$ are incomparable iff $j > i$. For $j \leq i$, arbitrary hypotheses become strictly stronger.

So far, we compared two single query types with each other and got a fairly complete picture of their relations. We did not systematically try to compare two collections of query types. Instead, we restrict ourselves to the following three simple observations:

1. UAV-MQ $\not\preceq$ (MQ, EQ). Witness: $(\textit{singleton}_n)_{n\geq 1}$
2. UAV-EQ$_{O(\log n)} \not\preceq$ (MQ, UAV-EQ). Witness: $(\textit{singleton}_n \cup \{F_n\})_{n\geq 1}$
3. (MQ, EQ) $\prec$ (UAV-MQ, UAV-EQ).

In Observation 2, $F_n \in \mathcal{B}_n$ denotes the constant zero function. Observation 3 is a direct consequence of Observation 1 and the fact from [7] that (MQ, EQ) $\preceq$ (UAV-MQ, UAV-EQ).

6 CONCLUSIONS AND OPEN PROBLEMS

Our results give answers to some questions that were raised in [7]. For instance, Goldman et al. asked how the different query types in the UAV learning model are mutually related (including computational issues). From the perspective of information-theory (i.e., excluding computational issues), Figure 2 gives a complete answer to this question. Note however that most edges in Figure 2 allow for a stronger interpretation which also pays attention to computational complexity:

1. A dashed edge in Figure 2 from $\mathcal{Q}$ to $\mathcal{Q}'$ may be interpreted as follows. There exists at least one

concept class which is learnable in polynomial time using $\mathcal{Q}'$, but not learnable using $\mathcal{Q}$ (even with unlimited computational resources).

2. A solid edge in Figure 2 from $\mathcal{Q}$ to $\mathcal{Q}'$ with $(\mathcal{Q}, \mathcal{Q}') \neq$ (UAV-MQ, ARB-EQ) may be interpreted as follows. Each class that can be learned using $\mathcal{Q}$ can also be learned using $\mathcal{Q}'$. The same implication holds for learnability in polynomial time. Additionally, $\mathcal{Q}$ and $\mathcal{Q}'$ satisfy the condition for dashed edges.

Note that, except for the mutual relations between ARB-EQ and the query types in the UAV-MQ hierarchy, this is the strongest possible answer to the question raised by Goldman et al. The problem with the relation between UAV-MQ and ARB-EQ is that the "Majority Vote" strategy is not time-efficient in general.

In fact, it is very likely that ARB-EQ and the UAV-MQ hierarchy are incomparable as far as learnability in polynomial time is concerned. The fact that UAV-MQ is not more powerful than ARB-EQ is witnessed by $(\textit{dirty-par}_n)_{n\geq 1}$. According to Lemma 5.1, the fact that ARB-EQ is not more powerful than MQ can be witnessed by the address extension of any class $(\mathcal{C}_n)_{n\geq 1}$ which satisfies the following conditions:

- $(\mathcal{C}_n)_{n\geq 1}$ is not ARB-EQ learnable in polynomial time.

- $(\mathcal{C}_n)_{n\geq 1}$ has a *polynomial representation*, i.e., $\log |\mathcal{C}_n|$ is polynomially bounded in n.

Under some standard cryptographic assumptions, it can be shown that such classes $(\mathcal{C}_n)_{n\geq 1}$ exist. For instance, poly-size Boolean formulae are not PAC-learnable in polynomial time [9] (given some standard cryptographic assumptions). This result is representation independent, i.e., it holds even when the PAC-learner may use an arbitrary Boolean function as its final hypothesis. It is well known [1, 10] that each class which is ARB-EQ learnable in polynomial time can be PAC learned (using an arbitrary final hypothesis) in polynomial time. Hence, poly-size Boolean formulae are not ARB-EQ learnable in polynomial time (given some standard cryptographic assumptions). Clearly, they have a polynomial representation. Thus the address extension of poly-size Boolean formulae witnesses that ARB-EQ is not more powerful than MQ (given some standard cryptographic assumptions).

The following challenging problems are still open:

- Goldman et al. proved that decision trees are learnable in polynomial time using UAV-MQs, UAV-ARB-EQs and an additional query type EV, where $\text{EV}(h, a)$ yields the evaluation of hypothesis h on UAV instance a. Are decision trees learnable in polynomial time even without EVs?

- It is known that decision trees are learnable in polynomial time using MQs and ARB-EQs [5]. Are they properly learnable in polynomial time using MQs and EQs?[4]

- Is DNF learnable using UAV-MQs, UAV-EQs and EVs (maybe even in polynomial time or even without EVs)?

References

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

[2] Dana Angluin, Lisa Hellerstein, and Marek Karpinski. Learning read-once formulas with queries. *Journal of the ACM*, 40(1):185–210, January 1993.

[3] Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. In *Proceedings of the 6th Annual Conference on Computational Learning Theory*, pages 287–296. ACM Press, New York, NY, 1993.

[4] Andreas Birkendorf, Eli Dichterman, Jeffrey Jackson, Norbert Klasner, and Hans Ulrich Simon. On restricted-focus-of-attention learnability of boolean functions. In *Proceedings of the 9th Annual Conference on Computational Learning Theory*, pages 205–216. ACM Press, New York, NY, June 1996. To appear in *Machine Learning*.

[5] Nader H. Bshouty. Exact learning boolean functions via the monotone theory. *Information and Computation*, 123:146–153, 1995.

[6] Michael Frazier, Sally Goldman, N. Mishra, and Leonard Pitt. Learning from a consistently ignorant teacher. In *Proceedings of the 7th Annual Conference on Computational Learning Theory*, pages 328–338. ACM Press, New York, NY, 1994.

[7] Sally A. Goldman, Stephen Kwek, and Stephen D. Scott. Learning from examples with unspecified attribute values. In *Proceedings of the 10th Annual Conference on Computational Learning Theory*, pages 231–242. ACM Press, New York, NY, 1997.

[8] Michael Kearns and Robert E. Schapire. Efficient distribution-free learning of probabilistic concepts. *Journal of Computer and System Sciences*, 48:464–497, 1994.

[9] Michael Kearns and Leslie G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, January 1994.

[10] Nick Littlestone. From on-line to batch learning. In *Proceedings of the 2nd Annual Workshop on Computational Learning Theory*, pages 269–284. Morgan Kaufmann, San Mateo, CA, 1989.

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

[12] Hans Ulrich Simon and Norbert Klasner. Tight bounds on the query complexity for exact learning. Unpublished manuscript, 1998.

[4] An affirmative answer to this open problem implies an affirmative answer to the preceding one [7].

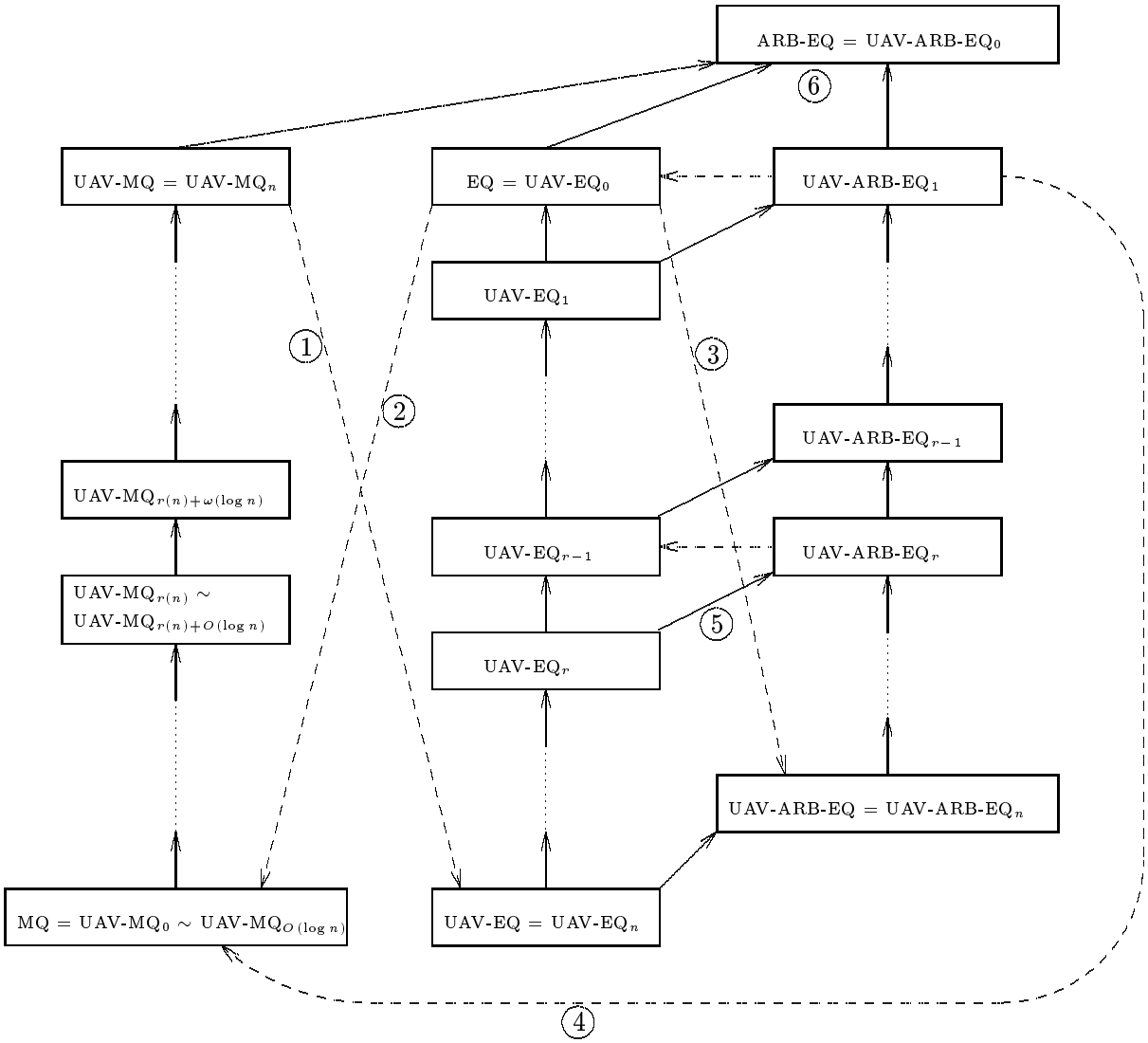

Figure 2: The UAV-MQ, UAV-EQ and UAV-ARB-EQ hierarchy and the relations between different hierarchies. The circled numbers correspond to the ones given in the text.