Structural Results about Exact Learning with Unspecified Attribute Values

Andreas Birkendorf, Norbert Klasner, Christian Kuhlmann, and Hans U. Simon

Lehrstuhl Mathematik und Informatik
Fakultät für Mathematik
Ruhr-Universität Bochum
D-44780 Bochum

February 3, 1998

Abstract

This paper deals with the UAV learning model of Goldman, Kwek and Scott [7]. ("UAV" is the acronym for "Unspecified Attribute Values".) As in [7], we consider exact learning within the UAV framework, where the learner has to exactly identify an unknown target concept by means of UAV membership (UAV-MQs) and/or UAV equivalence queries (UAV-EQs or UAV-ARB-EQs, respectively). 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 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 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-decision lists. This is contrasted by our positive learning results, where efficient UAV-MQ learning algorithms for read-once DNF, constant-term DNF and 1-decision lists are presented (in addition to some other positive learning results). Finally, we determine 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. As for the relations between two query types from different hierarchies, we can always tell whether they are equivalent, or independent, or whether one is strictly stronger than the other. (For instance, each query type from the UAV-MQ hierarchy is incomparable to each query type from the UAV-EQ hierarchy.) We also compare two collections of query types (though in a less systematic fashion).

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* [9], *learning with a consistently ignorant teacher* [6] or *learning with a constrained instance* or *projective equivalence oracle* [2].

In the sequel, we assume that the reader is familiar with the standard *exact learning model*, introduced by Angluin [1], where an unknown target concept drawn from a (known) *concept class* $\mathcal{C}$ must be exactly identified by means of membership queries (MQs) and equivalence queries (EQs).

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). 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.

In this paper, we additionally consider the adaptation of arbitrary equivalence queries to the UAV framework. This type of query is denoted as UAV-ARB-EQ. The only difference

[1]UAV=Unspecified Attribute Values

to UAV-EQ is that query UAV-ARB-EQ can be instantiated with an arbitrary hypothesis $h \in \mathcal{B}_n$ (as opposed to hypotheses from $\mathcal{C}_n$). A standard equivalence query using an arbitrary hypothesis is abbreviated ARB-EQ.

Furthermore, 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 $r = 0$.

The goal of a learner (learning algorithm) is the exact identification of the unknown target concept from $\mathcal{C}_n$ using as few queries as possible in the worst case. The worst case is taken over all possible choices of target concepts from $\mathcal{C}_n$ and over all answers to queries which are consistent with the chosen target concept. (For fixed target f, an answer of the form $f(a)$ is unique. But there are typically several choices for a counterexample to the current hypothesis.)

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. Information-theoretic lower bounds on the number of queries hold even when the learner has unlimited computational resources. Information-theoretic upper bounds may rely on learning strategies which are inherently inefficient. 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 (which are still easy to obtain). For instance, UAV-MQ$_{O(\log n)} \sim$ MQ and $(\text{MQ}, \text{EQ}) \sim (\text{MQ}, \text{UAV-EQ}_{O(\log n)})$.

In general, the power of membership queries increases with the number of unspecified attribute values per UAV instance, whereas the power of (arbitrary) equivalence queries decreases. The increased power of UAV-MQ may help to turn unlearnability results in the ordinary exact learning model to positive results in the UAV framework. Goldman et al., for instance, designed an UAV-MQ algorithm for the class of ordered decision trees, which is not learnable using ordinary MQs. Clearly, the decreased power of UAV-EQ works in the opposite direction. It is a straightforward question to ask how $(\mathrm{EQ}, \mathrm{MQ})$ and $(\text{UAV-EQ}, \text{UAV-MQ})$ are related to each other. Goldman et al. have shown that $(\mathrm{EQ}, \mathrm{MQ}) \preceq (\text{UAV-EQ}, \text{UAV-MQ})$ by transforming a standard exact learner into an exact UAV learner.[2]

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-decision lists. 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-decision lists 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:

$$\begin{aligned} \text{UAV-EQ}_r &\prec \text{UAV-EQ}_{r-1}, \\ \text{UAV-ARB-EQ}_r &\prec \text{UAV-ARB-EQ}_{r-1}, \\ \text{UAV-MQ}_{r(n)} &\prec \text{UAV-MQ}_{r(n)+\omega(\log n)}. \end{aligned}$$

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 $\text{UAV-MQ} \not\preceq (\mathrm{EQ}, \mathrm{MQ})$ holds which (in combination with the aforementioned result $(\mathrm{EQ}, \mathrm{MQ}) \preceq (\text{UAV-EQ}, \text{UAV-MQ})$ from [7]) implies that $(\mathrm{EQ}, \mathrm{MQ}) \prec (\text{UAV-EQ}, \text{UAV-MQ})$.

Our paper gives answers to several questions that were raised in [7]. This is explained in more detail in Section 6, which also mentions some (still) open problems.

[2]Remember that our partial ordering ignores computational issues. The transformation of a standard exact learner into an exact UAV learner is query-efficient, but, in general, not time-efficient. However, if the standard exact learner uses only hypotheses which can be evaluated on UAV instances from $\{0, 1, *\}^n$ in $\mathrm{poly}(n)$ steps, the transformation is even time-efficient. See [7] for more details.

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. Furthermore, let $VCdim(\mathcal{C})$ be the Vapnik Chervonenkis dimension of $\mathcal{C}$.

2.1 General lower bounds

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[3]. This yields a lower bound of $\log|\mathcal{C}|/\log 3$ queries. Furthermore, $\log|\mathcal{C}| \geq VCdim(\mathcal{C})$. •

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.

Since $VCdim(\mathcal{C}') = k < \lfloor k \log 3 \rfloor$, we cannot use this class directly 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

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

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

```
for i := 0 to k − 1 do
    case UAV-MQ(bin(i)∗)
        0: a_i := 0
        ∗: a_i := 1
        1: a_i := 2
return c_{a_{k−1}...a_0}
```

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

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

$$\mathcal{C} = \{0c_i' \cup 1c_i'' \,|\, c_i' \text{ is concept number } i \text{ of } \mathcal{C}', c_i'' \text{ is concept number } i \text{ of } \mathcal{C}''\},$$

where $ac = \{ax \,|\, x \in c\}$ 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. •

Observation 2.3 *The total number of ARB-EQs and UAV-MQs, needed to learn a class $\mathcal{C}$, is at least $\frac{1}{7} \cdot VCdim(\mathcal{C})$.*

See [12, Theorem 6.5 and Proposition 6.10 a)]. The proofs given for these results imply the observation.

We do not know whether the general lower bound within Observation 2.3 can be improved. The following upper bound (for a special class $\mathcal{C}_n$) leaves a gap.

Observation 2.4 *For each $n \geq 1$, there is a class $\mathcal{C}_n$ 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 [12, 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[4], 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. •

2.2 Applications of the UAV Sunflower Lemma

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.

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

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$_n$ *using 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 $X = \{0,1\}^n$, and $Y = \{0,1\}^n \setminus par_n$. 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 $\not\leq$ UAV-MQ.

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

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 This proof uses Angluin's construction [1] showing non-learnability using MQs. 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 $c_t \in \mathcal{C}$ 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 $c_\cap$. •

Input: $a \in \{0,1,*\}^n$, where $f(a) \neq 0$

for each $a_i = *$ **do**

  $a_i := 0$

  case UAV-MQ(a)

    1: **return** a

    0: $a_i := 1$

return a

Figure 2: Algorithm *Contract*.

3 Positive learning results

First, we present two simple algorithms called *Contract* and *Expand*. Later in this section, we will use them as subroutines. Remember that, for each UAV instance a, $X(a) \subseteq \{0,1\}^n$ denotes the corresponding subcube. For sake of brevity, we write $a \subseteq b$ if $X(a) \subseteq X(b)$.

Contract starts with an UAV instance a, where $X(a)$ contains at least one positive example for the target concept f, and outputs a positive UAV example $b \subseteq a$. Clearly, b represents an implicant of f.

Algorithm *Expand* starts with a positive UAV example a (i.e., a represents an implicant of f) and outputs a "maximal" positive UAV example $b \supseteq a$, i.e., $X(b) \supseteq X(a)$, but each proper subcube extension of $X(b)$ contains at least one negative example for f. Clearly, b represents a prime implicant of f. These algorithms are shown in Figures 2 and 3.

Lemma 3.1 *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 Algorithm* Contract *is applied to a, it stops after $O(m)$ steps and returns an UAV example $b \subseteq a$ which satisfies $f(b) = 1$.* Contract *uses at most m UAV-MQs.*

Proof Algorithm *Contract* in Figure 2 performs the obvious search for a suitable b. 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 $b \subseteq a$ must satisfy $f(b) = 1$. Each iteration takes constant time and uses one UAV-MQ. •

Before we present Algorithm *Expand*, we give a simple application of Algorithm *Contract*. Let $singleton_n = \{l_1 \wedge \ldots \wedge l_n | l_i \in \{x_i, \overline{x}_i\}\}$.

Lemma 3.2 *The class $singleton_n$ is learnable using n UAV-MQs.*

Proof Apply Algorithm *Contract* to $a = (* \ldots *)$. The returned instance identifies the target concept. •

Input: $a \in \{0,1,*\}^n$, where $f(a) = 1$

<u>for each</u> $a_i \neq *$ **<u>do</u>**

  <u>if</u> UAV-MQ$(a_1 \ldots a_{i-1} * a_{i+1} \ldots a_n) = 1$

  <u>then</u> $a_i := *$

<u>return</u> a

Figure 3: Algorithm *Expand*.

Lemma 3.3 *Let f be the target concept and a an UAV instance with m specified attribute values (SAVs). Assume that $f(a) = 1$. When Algorithm* Expand *is applied to a, it stops after $O(m)$ steps performing at most m UAV-MQs. The returned UAV instance $b \supseteq a$ satisfies:*

1. $f(b) = 1$
2. $f(b') \neq 1$ *for each* $b' \supset b$.

Proof Algorithm *Expand* in Figure 3 performs the obvious search for a suitable b. 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 $b \supseteq a$. Each iteration takes constant time and uses one UAV-MQ. •

Let $monomial_n$ be the class of all monomials over n Boolean variables. We now apply algorithms *Contract* and *Expand* as subroutines of learning algorithms for $monomial_n$ and read-once DNF.

Lemma 3.4 *The class $monomial_n$ is learnable using $2n$ UAV-MQs.*

Proof First, test whether the target concept is a constant function, or not, using one UAV-MQ. If so, we are done. Otherwise, apply Algorithm *Contract* to $a = (* \ldots *)$ to find a positive UAV example b (at most n queries). Now, apply Algorithm *Expand* to b yielding a maximal positive UAV example c. The monomial corresponding to c is the target concept.

The procedure just described uses $2n + 1$ UAV-MQs. A more clever implementation can save 1 UAV-MQ during the expansion phase by keeping the last attribute value that was changed by *Contract* fixed. (Note that this value cannot be reset to $*$ without "catching" at least one negative example of the target concept.) •

Let $Contract_f(a)$ and $Expand_f(a)$ denote the instances returned by *Contract* and *Expand*, respectively, when the procedure is called on input instance a and $f \in \mathcal{B}_n$ is the target concept. We define the *EC-operator (associated with f)* as follows:

$$\mathrm{EC}_f(a) := Expand_f(Contract_f(a))$$

(We will omit subscript f when it is clear from context.) The above learning algorithm for monomials, for instance, computed the target monomial by applying the EC-operator to $(*\ldots*)$, unless the target concept is constant. As explained in the proof of Lemma 3.4, a clever implementation of operator EC can save 1 UAV-MQ in the expansion phase, given the assumption that *Contract* has changed at least one attribute value. Thus, one application of EC does never cost more than $2n-1$ UAV-MQs.

The following more general observations about the EC-operator concern an arbitrary (but fixed) Boolean target concept f and an UAV instance a. First observe that, if $f(a) \neq 0$, then $\mathrm{EC}_f(a)$ 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 $\mathrm{EC}(a)$ represents a prime implicant of f which is not yet covered by the prime implicants contained in h. Thus, iterative applications of the EC-operator 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 result.

Lemma 3.5 *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, we may apply the EC-operator successfully.

Let $i = |V_*|$. Clearly, *Contract* consumes only i UAV-MQs. Without further modification *Expand* consumes $n-1$ UAV-MQs. However, a more clever implementation of the expansion phase sets initially all variables outside V_* to $*$ without issuing UAV-MQs. Thus, the expansion phase focuses on variables in V_* (which does not harm because f is a read-once DNF). Therefore, *Expand* consumes at most $i-1$ UAV-MQs. Taking into account the extra UAV-MQ for checking whether $h \equiv f$, the total number of UAV-MQs for the construction of one new prime implicant is at most $2i$. Target f can have no more than n prime implicants. The total number of UAV-MQs is therefore bounded by $\sum_{i=1}^{n} 2i = n(n+1)$. •

Lemma 3.5 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 3.6 *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, EC(a) leads to a prime implicant of f not already covered by h. In the latter case, *Expand*(a) can be used for the same purpose. The lemma now easily follows. $\bullet$

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.

Lemma 3.7 *For each k, class $(k\text{-}DNF_n)_{n\geq 1}$ is learnable using $O(n^k)$ UAV-MQs.*

Proof Let t be a monomial and a the corresponding UAV instance. Obviously, t is an implicant of f iff the UAV-MQ for a is answered 1. A k-DNF$_n$ formula f is equivalent to the disjunction of all k-monomials (monomials consisting of at most k literals), which are implicants of f. Since the number of k-monomials is $O(n^k)$, it takes only $O(n^k)$ UAV-MQs to find the implicants of f among them. $\bullet$

Lemma 3.8 *The class 1-$DL_{n,q}$ 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.*

Proof Let f be the target function from 1-DL$_{n,q}$. Let

$$L = [(l_1, \beta_1), \ldots, (l_i, \beta_i)]$$

be a list of items that is *consistent* with f, i.e., if an instance x satisfies one of the literals within L, then the first item with a satisfied literal has label $f(x)$. We say that (l, β) (or $(1, \beta)$, respectively) is a *legal extension* of L with respect to f if list L' coinciding with L followed by item (l, β) (or $(1, \beta)$, respectively) is still consistent with f.

For the time being, let us assume that there is a procedure P which can efficiently compute all legal extensions of a given list L. It is well known that 1-DL$_{n,q}$ is learnable using q calls of P. (See [13] or [8], for instance). The proof is now completed as follows. Observe that (l, β) is a legal extension of L iff $f \equiv \beta$ on the subcube given by $\bar{l}_1\bar{l}_2\ldots\bar{l}_i l$, which can be tested by 1 UAV-MQ. Thus, $2(n-i)$ UAV-MQs are enough to exhaustively test all candidate literals l. (Note that $(1, \beta)$ is a legal extension iff all $2(n-i)$ UAV-MQs lead to the same answer β. We need therefore no extra query for this extension.) The total number of UAV-MQs performed during the q calls of P is bounded by

$$\sum_{i=0}^{q-1} 2(n-i) = 2nq - q^2 + q \leq n(n+1).$$

$\bullet$

Lemma 3.8 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 $X_{b=1}$).

This is a subtle feature of the padding construction which will be understood more clearly later when we discuss the adversary's strategy of presenting counterexamples in X. We are now in the position to define $\mathcal{C}$:

$$\mathcal{C} = \{f^r \mid f \in \mathcal{C}_n\}.$$

Obviously, all concepts from $\mathcal{C}$ are well-formed.

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 the adversary strategy, say ADV$_n$, which was described in the first part of the proof. 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:*

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

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

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, we show now that the hierarchy collapses when we step from $n-1$ to n.

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

Proof We restrict ourselves to query type UAV-EQ. The proof for UAV-ARB-EQ is similar. Since $\text{UAV-EQ}_n \preceq \text{UAV-EQ}_{n-1}$ trivially holds, we have to show that $\text{UAV-EQ}_{n-1} \preceq \text{UAV-EQ}_n$.

Let $\mathcal{C}_n$ denote the concept class which is learnable with an algorithm A performing q queries of type UAV-EQ_{n-1}. We assume w.l.o.g., that A is conservative, i.e., no hypothesis of A is disproven by an already received counterexample. (Otherwise, A gains no information about the target concept if this counterexample is presented again.)

Then the following algorithm A' learns $\mathcal{C}_n$ with at most $4q$ queries of type UAV-EQ_n. A' runs A until a counterexample $c=[(*\ldots*),l]$ with n UAVs is returned. If $l\in\{0,1\}$ then the target concept, a constant function, is identified and A' stops. If $l=*$ then the last hypothesis h of A' was a constant function. It is easy to see that at least one of the UAV instances $c_1=[(0*\ldots*),*]$, $c_2=[(1*\ldots*),*]$, $c_3=[(*\ldots*0),*]$, $c_4=[(*\ldots*1),*]$ is a counterexample for h with $n-1$ UAVs. Now, A' runs four copies A_1,A_2,A_3,A_4 of A in parallel, where copy A_i is provided with counterexample c_i. Since A is conservative and the target concept is not a constant function, all copies A_i receive only counterexamples with at most $n-1$ UAVs. Hence, one copy succeeds and identifies the target function. •

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

We now turn our attention to the UAV-MQ-hierarchy. We start with the following lemma (which is a slight generalization of Observation 1 in [7]).

Lemma 4.4 *For all functions $k = k(n), k' = k'(n)$ satisfying $1 \leq k, k' \leq n$ and $|k - k'| = O(\log n)$, the following holds:*

$$\textit{UAV-MQ}_k \sim \textit{UAV-MQ}_{k'}.$$

Proof Let $d = O(\log n)$. Clearly, a query of type UAV-MQ$_k$ can be simulated by 2^d queries of type UAV-MQ$_{k-d}$. Since $d = O(\log n)$, factor 2^d is polynomial in n. •

Hence, two types of UAV-MQs are equally strong as long as the corresponding numbers of UAVs per query differ at most by a logarithmic term. In the remainder of this section, we show that this result is tight: if the difference between the numbers of UAVs is super-logarithmic in n, there exists a concept class which separates the two query types. We first consider the special case $k(n) = n$.

Lemma 4.5 *For all $1 \leq k \leq n$, $singleton_n$ is not learnable using less than 2^{n-k} queries of type UAV-MQ$_k$.*

Proof Each query of type UAV-MQ$_k$ asks for the label of an at most k-dimensional subcube of $\{0,1\}^n$. The subcubes of $2^{n-k} - 1$ such queries cover at most $2^n - 2^k$ instances in $\{0,1\}^n$. If label 0 is returned to all these queries, the at least 2^k uncovered instances still belong to the version space. Thus, the target is not yet exactly identified. •

Corollary 4.6 *For all functions $k = k(n)$ and $d = d(n)$ satisfying $\omega(\log n) \leq d \leq k \leq n$ the following holds:*

$$\textit{UAV-MQ}_{k-d} \prec \textit{UAV-MQ}_k.$$

Proof Clearly, UAV-MQ$_{k-d} \preceq$ UAV-MQ$_k$. We have to present a concept class which separates the two query types. For $k(n) = n$, the separation follows from Lemma 4.5. For general $k(n) \leq n$, we may first apply Lemma 4.5 to $n' = k(n)$, and then use the obvious padding construction that adds $n - n'$ Boolean dummy variables (i.e., the concepts in $singleton_{n'}$ are considered as concepts over domain $\{0,1\}^n$, but do not essentially depend on the $n - n'$ dummy variables). •

5 Relations between the hierarchies

Figure 4 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 4 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 $\text{UAV-EQ}_{r-1} \not\preceq \text{UAV-ARB-EQ}_r$ (which justifies the dashed edge from UAV-ARB-EQ_r to UAV-EQ_{r-1}).

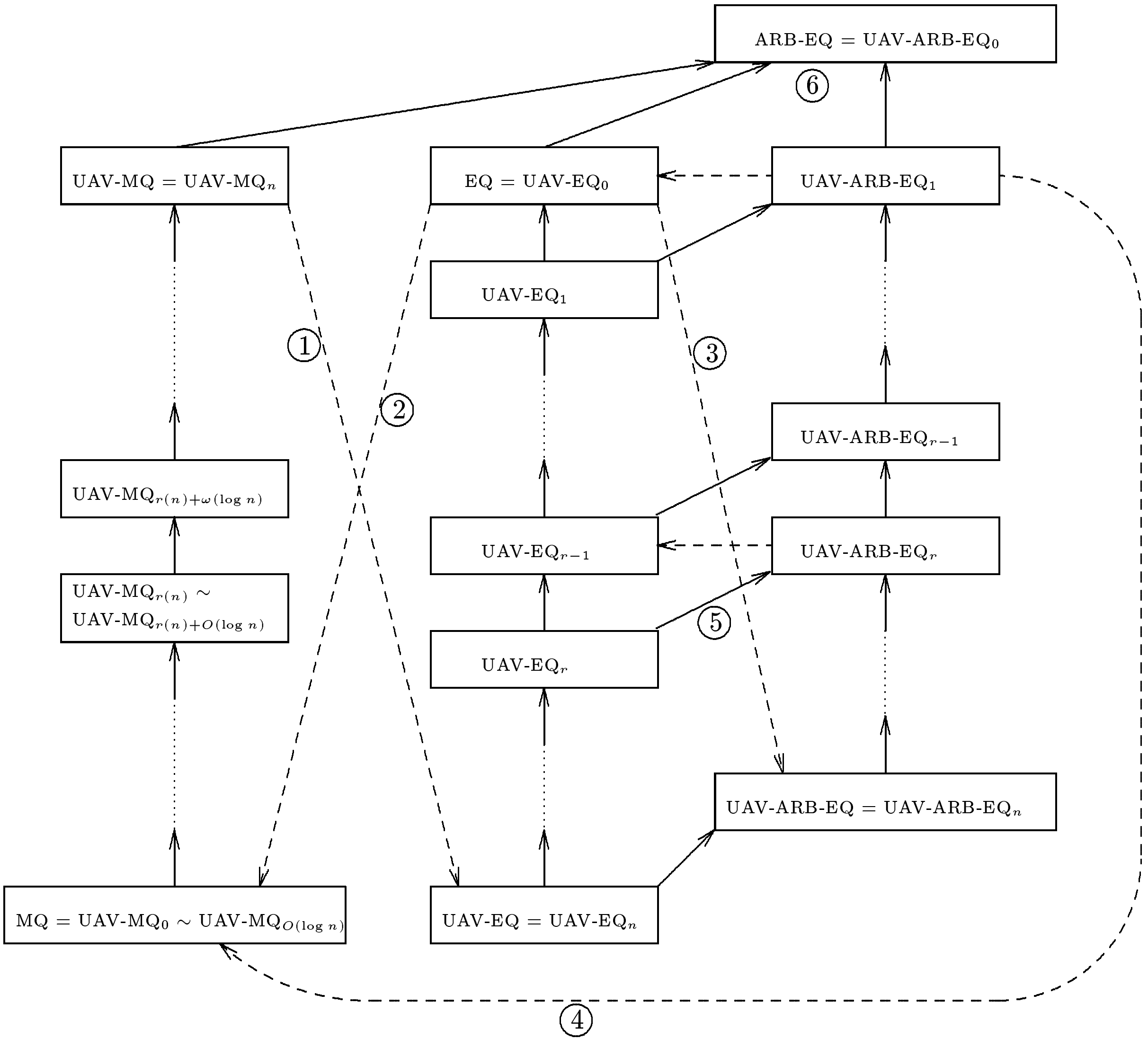

Figure 4: The UAV-MQ, UAV-EQ and UAV-ARB-EQ hierarchy and the relations between different hierarchies. The circled numbers refer to the corresponding observations.

We aim to justify the remaining edges in Figure 4. The following lemma, which is based on a construction in [12], provides a useful tool for this purpose.

Lemma 5.1 *Let $p(n)$ be a 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.*

Proof For ease of exposition, we assume that $|\mathcal{C}_n| = 2^{p(n)}$ and $p(n)$ is a power of 2, say 2^d, although these assumptions are not essential. Let $f_0, \ldots, f_{2^{p(n)}-1}$ denote the $2^{p(n)}$ concepts in $\mathcal{C}_n$. For each $0 \leq i \leq 2^{p(n)} - 1$, let $(b_{i,p(n)-1}, \ldots, b_{i,0})$ denote the binary representation of i. For each $0 \leq l \leq p(n) - 1$, let $bin(l) \in \{0,1\}^d$ denote the binary representation of l. Then $\mathcal{C}_n^A$ is defined as the class of the following functions f_i^A (for $i = 0, \ldots, 2^{p(n)} - 1$):

$$\begin{aligned} f_i^A(x, bin(l), 0) &= f_i(x) \\ f_i^A(x, bin(l), 1) &= b_{i,l} \end{aligned}$$

Let f_i^A be the target concept. Since MQ$(x, bin(l), 1)$ yields bit $b_{i,l}$, the corresponding $p(n) = 2^d$ MQs for all possible choices of l reveal the binary representation of i, which exactly identifies the target. The algorithm obviously runs in polynomial time.
Now assume that A is a learner for $(\mathcal{C}_n^A)_{n\geq 1}$ using queries of type $\mathcal{Q}$. We obtain a corresponding learner A' for $(\mathcal{C}_n)_{n\geq 1}$ as follows. A' simulates A. When A issues hypothesis h, then A' issues the hypothesis h' given by $h'(x) = h(x, bin(0), 0)$. When counterexample a is returned, A' feeds counterexample $(a, bin(0), 0)$ into the simulation. The simulation proceeds in this way until the target concept from $\mathcal{C}_n$ is exactly identified. •

We are now prepared to argue that all edges in Figure 4 are justified.

Observation 1 UAV-EQ $\not\leq$ UAV-MQ.
Lemma 2.6 implies that the class $(\textit{dirty-par}_n \cup \{par_n\})_{n\geq 1}$ is not learnable using UAV-MQs. It is however learnable using only UAV-EQ(par_n).

Observation 2 MQ $\not\leq$ EQ [12].
This relation is witnessed by the address extension of $(singleton_n)_{n\geq 1}$ (which is basically the construction used in [12]).

Observation 3 UAV-ARB-EQ $\not\leq$ EQ.
According to Lemma 2.6, class $(\textit{dirty-par}_n)_{n\geq 1}$ is not learnable using EQs. It is however learnable using only UAV-ARB-EQ(par_n).

Observation 4 MQ $\not\preceq$ UAV-ARB-EQ$_1$.

This relation is witnessed by the address extension of the class $(\mathcal{C}_n^1)_{n\geq 1}$ from Theorem 4.1, which is not learnable using queries of type UAV-ARB-EQ$_1$.

Observation 5 For each constant $r \geq 0$: UAV-EQ$_r$ $\prec$ UAV-ARB-EQ$_r$.

Relation "stronger" is trivial. The strictness immediately follows from Observation 3. Observation 5 justifies all solid edges in Figure 4 with tail in the UAV-EQ hierarchy and head in the UAV-ARB-EQ hierarchy.

Observation 6 ARB-EQs are strictly stronger than any other query type within the three hierarchies.

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. As yet, it follows that UAV-ARB-EQ is stronger than any other query type in Figure 4. However, since query types UAV-MQ, EQ and UAV-ARB-EQ$_1$ are obviously pairwise non-equivalent, ARB-EQ is even strictly stronger.

Note that Figure 4 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.

Observation 7 UAV-MQ $\not\preceq$ $(\mathrm{MQ}, \mathrm{EQ})$.

Lemma 3.2 shows that $singleton_n$ is learnable using n UAV-MQs. On the other hand, Angluin's Sunflower Lemma [1] implies that the total number of MQs and and EQs, needed to learn $singleton_n$, is $2^n - 1$.

Observation 8 $(\mathrm{MQ}, \mathrm{EQ}) \prec (\text{UAV-MQ}, \text{UAV-EQ})$.

As already mentioned in the introduction, this is a consequence of Observation 7 and the fact from [7] that $(\mathrm{MQ}, \mathrm{EQ}) \preceq (\text{UAV-MQ}, \text{UAV-EQ})$.

Observation 9 UAV-EQ$_{O(\log n)}$ $\not\preceq$ $(\mathrm{MQ}, \text{UAV-EQ})$.

Let $F_n \in \mathcal{B}_n$ denote the Boolean function which is zero on $\{0,1\}^n$ and consider the class $(\mathcal{C}_n)_{n\geq 1}$ with $\mathcal{C}_n = (singleton_n \cup \{F_n\})$. (This class was used in [7] as a witness for EQ $\not\preceq$ UAV-EQ.) A straightforward adversary strategy shows that $\mathcal{C}_n$ cannot be learned using less than 2^n queries of type MQ or UAV-EQ. However, $\mathcal{C}_n$ is learnable in polynomial time, using queries of type UAV-EQ$_{O(\log n)}$, as follows. If the first query issues hypothesis F_n, the returned counterexample with at most $O(\log n)$ UAVs allows to eliminate all but poly(n) many concepts from the version space. The target concept can then be found in an exhaustive search manner.

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 4 gives a complete answer to this question. Note however that most edges in Figure 4 allow for a stronger interpretation which also pays attention to computational complexity:

1. A dashed edge in Figure 4 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 4 from $\mathcal{Q}$ to $\mathcal{Q}'$ with $(\mathcal{Q}, \mathcal{Q}') \neq (\text{UAV-MQ}, \text{ARB-EQ})$ 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 [10] (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, 11] 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 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?[5]

- 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] David Helmbold, Robert Sloan, and Manfred K. Warmuth. Learning nested differences of intersection-closed concept classes. *Machine Learning*, 5:165–196, 1990.

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

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

[10] 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.

[11] 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.

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

[13] Hans Ulrich Simon. Learning decision lists and trees with equivalence-queries. In *Proceedings of the 2nd European Conference on Computational Learning Theory*, pages 322–336. Springer, Berlin, Heidelberg, 1995.