Polynomial Certificates for Propositional Classes[⋆]

Marta Arias[1], Roni Khardon[1], and Rocco A. Servedio[2]

[1] Department of Computer Science, Tufts University
Medford, MA 02155, USA
{marias,roni}@cs.tufts.edu

[2] Department of Computer Science, Columbia University
New York, NY 10027, USA
rocco@cs.columbia.edu

Abstract. This paper studies the query complexity of learning classes of expressions in propositional logic from equivalence and membership queries. We give new constructions of polynomial size certificates of non-membership for monotone, unate and Horn CNF functions. Our constructions yield quantitatively different bounds from previous constructions of certificates for these classes. We prove lower bounds on certificate size which show that for some parameter settings the certificates we construct for these classes are exactly optimal. Finally, we also prove that a natural generalization of these classes, the class of renamable Horn CNF functions, does *not* have polynomial size certificates of non-membership, thus answering an open question of Feigelson.

1 Introduction

This paper is concerned with the model of exact learning from equivalence and membership queries [1]. Since its introduction [1] this model has been extensively studied and many classes have been shown to be efficiently learnable. Of particular relevance for the current paper are learning algorithms for monotone DNF expressions [26, 1], unate DNF expressions [7], and Horn CNF expressions [2, 11]. Some results in this model have also been obtained for sub-classes of Horn expressions in first order logic but the picture there is less clear. Except for a "monotone-like case" [24] the query complexity is either exponential in one of the crucial parameters (e.g. universally quantified variables) [17, 4] or the algorithms use additional syntax based oracles [5, 25]. It is thus interesting to investigate whether this gap is necessary. The current paper takes a first step in this direction by studying the query complexity in the propositional case.

Query complexity can be characterized using the combinatorial notion of *polynomial certificates* [14, 12] (see also [6, 3]). In particular, [14, 12] show that

[⋆] This work has been partly supported by NSF Grant IIS-0099446 (M.A. and R.K.) and by an NSF Mathematical Sciences Postdoctoral Research Fellowship (R.S). Work done while R.S. was at the Division of Engineering and Applied Sciences, Harvard University.

a class $\mathcal{C}$ is learnable from a polynomial number of proper equivalence queries (using hypotheses in $\mathcal{C}$) and membership queries if and only if the class $\mathcal{C}$ has polynomial certificates. This characterization is information theoretic and ignores run time. Certificates have already proved to be a useful tool for studying learnability. For example, conjunctions of unate formulas are learnable with a polynomial number of queries but not learnable in polynomial time unless P=NP [10]. A recent result [15] shows that DNF expressions require a super-polynomial number of queries even when the hypotheses are larger than the target function by some (relatively small) factor.

Our own results establish lower and upper bounds on certificates for several classes. We give constructions of polynomial certificates for (1) monotone CNF where no variables are negated, (2) unate CNF where by renaming some variables as their negations we get a monotone formula, and (3) Horn CNF where each clause has at most one positive literal. We give certificates in the standard learning model as well as the model of learning from entailment [11] which is studied extensively in Inductive Logic Programming (see e.g. [8]). The construction of certificates for the Horn case is based on an analysis of *saturation* forming a "standardized representation" for Horn expressions that has useful properties.

The learnability results that follow from these certificate results are weaker than the results in [26, 1, 7, 2] since we obtain query complexity results and the results cited are for time complexity. However, the certificate constructions which we give are different from those implied by these earlier algorithms, so our results may be useful in suggesting new learning algorithms. We also give new lower bounds on certificate size for each of these concept classes. For some parameter settings, our lower bounds imply that our new certificate constructions are exactly optimal.

Finally, we also consider the class of renamable Horn CNF expressions. Note that unate CNF and Horn CNF generalize monotone expressions in two different ways. Renamable Horn expressions combine the two allowing to get a Horn formula after renaming variables. Renamable Horn formulas can be identified in polynomial time and have efficient satisfiability algorithms and are therefore interesting as a knowledge representation. While unate CNF and Horn CNF each have polynomial certificates, we give an exponential lower bound on certificate size for renamable Horn CNF. This answers an open question of Feigelson [9] and proves that renamable Horn CNF is not learnable in polynomial time from membership and equivalence queries.

2 Preliminaries

We assume that the reader is familiar with basic propositional logic. For completeness we give some of the definitions we use repeatedly in the paper.

We consider families of expressions built from $n \geq 1$ propositional variables. We assume some fixed ordering so that an element of $\{0,1\}^n$ specifies an assignment of a truth value to these variables. A *literal* is a variable or its negation. A *term* is a conjunction of literals. A *clause* is a disjunction of literals. A *Horn* *clause* is a clause in which there is at most one positive literal. A *DNF* expression is a disjunction of terms. A *CNF* expression is a conjunction of clauses; it is Horn if all its clauses are Horn.

Let $x, y \in \{0,1\}^n$ be two assignments. Their intersection $x \cap y$ is the assignment that sets to 1 only those variables that are 1 in both x and y.

The *DNF size* of a boolean function $f \subseteq \{0,1\}^n$, denoted $|f|_{DNF}$, is the minimum number of terms in a DNF representation of f. The *CNF size* of f, $|f|_{CNF}$, is defined analogously. In general, let $\mathcal{R}$ be a representation class for boolean formulas. Then $|f|_{\mathcal{R}}$ is the size of a minimal representation for f in $\mathcal{R}$. If $f \notin \mathcal{R}$, we assign $|f|_{\mathcal{R}} = \infty$.

Let $\mathcal{C}$ be a boolean class, i.e. $\mathcal{C} \subseteq 2^{\{0,1\}^n}$. Then $\mathcal{C}_m$ denotes the subclass of $\mathcal{C}$ whose concepts have size at most m.

Definition 1 (Polynomial Certificates, from [14]). *Let $\mathcal{R}$ be a class of representations defining a boolean concept class $\mathcal{C}$. The class $\mathcal{R}$ has* polynomial certificates *if there exist two-variable polynomials $p(\cdot,\cdot)$ and $q(\cdot,\cdot)$ such that for every $n, m > 0$ and for every boolean function $f \subseteq \{0,1\}^n$ s.t. $|f|_{\mathcal{R}} > p(m,n)$, there is a set $Q \subseteq \{0,1\}^n$ satisfying the following: (1) $|Q| \leq q(m,n)$ and (2) for every $g \in \mathcal{C}_m$ there is some $x \in Q$ s.t. $g(x) \neq f(x)$. In other words, (2) states that no function in $\mathcal{C}_m$ is consistent with f over Q.*

We also need the notion of redundant expressions:

Definition 2. *A clause C in a Horn expression f is* redundant *if $f \setminus \{C\} \equiv f$. An expression f is redundant if it contains a redundant clause.*

3 Certificates for Monotone and Unate CNFs

In this section we construct polynomial certificates for anti-monotone CNFs (generalizable to unate CNF/DNF). This is to facilitate the presentation of certificates for Horn CNF. A certificate for unate DNF was given in [9]:

Theorem 1 (Lemma 5 from [9], page 26). *The classes of monotone and unate functions under DNF have polynomial size certificates with $p(m,n) = m$ and $q(m,n) = O(mn)$.*

Feigelson's construction is based on the fact that to show that a unate DNF function has more than m terms, it is sufficient to prove that it has $m+1$ minterms, which can be done by including in the certificate $m+1$ positive assignments corresponding to the minterms and $O(mn)$ negative assignments corresponding to the assignments one level below the positive ones.

We next show a construction that achieves a certificate of size $O(m^2)$ which improves Feigelson's construction when $m < n$.

An anti-monotone CNF expression is a CNF where all variables appear negated. In this case we have that anti-monotone CNFs satisfy:

$$\forall x, y \in \{0,1\}^n : (x < y \Longrightarrow f(x) \geq f(y)),$$

where $<$ between assignments denotes the standard bit-wise relational operator.

Notice that an anti-monotone CNF expression can be seen as a Horn CNF whose clauses have empty consequents. As an example, the anti-monotone CNF $(\bar{a} \vee \bar{b}) \wedge (\bar{b} \vee \bar{c})$ is equivalent to the Horn CNF $(ab \to \mathtt{false}) \wedge\ (bc \to \mathtt{false})$.

Theorem 2. *The class of anti-monotone CNF has polynomial size certificates with* $p(m,n) = m$ *and* $q(m,n) = \binom{m+1}{2} + m + 1$.

Proof. Fix $m, n > 0$. Fix any $f \subseteq \{0,1\}^n$ s.t. $|f|_{anti-monCNF} > p(m,n) = m$. We proceed by cases.

Case 1. f is not anti-monotone. In this case, there must exist two assignments $x, y \in \{0,1\}^n$ s.t. $x < y$ but $f(x) < f(y)$ (otherwise f would be anti-monotone). Let $Q = \{x, y\}$. Notice that by definition no anti-monotone CNF can be consistent with Q. Moreover, $|Q| \leq q(m,n)$.

Case 2. f is anti-monotone. Let $c_1\ \wedge\ c_2\ \wedge\ \ldots\ \wedge\ c_m\ \wedge\ \ldots\ \wedge\ c_k$ be a minimal representation for f. Notice that $k \geq m+1$ since $|f|_{anti-monCNF} > p(m,n) = m$. Define assignment $x^{[c_i]}$ as the assignment that sets to 1 exactly those variables that appear in c_i's antecedent. For example, if $n = 5$ and $c_i = v_3 v_5 \to \mathtt{false}$ then $x^{[c_i]} = 00101$.

Remark 1. Notice that every $x^{[c_i]}$ falsifies c_i (antecedent is satisfied but consequent is `false`) but satisfies every other clause in f. If this were not so, then we would have that some other clause c_j in f is falsified by $x^{[c_i]}$, that is, the antecedent of c_j is true and therefore all variables in c_j appear in c_i as well (i.e. $c_j \subseteq c_i$). This is a contradiction since c_i would be redundant and we are looking at a minimal representation of f.

Now, define the set $Q = Q^+ \cup Q^-$ where

$$Q^- = \left\{ x^{[c_i]} \;\middle|\; 1 \leq i \leq m+1 \right\} \text{ and } Q^+ = \left\{ x^{[c_i]} \cap x^{[c_j]} \;\middle|\; 1 \leq i < j \leq m+1 \right\}.$$

Notice that $|Q| \leq \binom{m+1}{2} + m + 1 = q(m,n)$. The assignments in Q^- are negative for f, since $x^{[c_i]}$ clearly falsifies clause c_i (and hence it falsifies f). The assignments in Q^+ are positive for f. To see this, suppose some $x^{[c_i]} \cap x^{[c_j]} \in Q^+$ is negative. This implies that there is some clause c in f that is falsified by $x^{[c_i]} \cap x^{[c_j]} \in Q^+$. That is, all variables in c are set to 1 by $x^{[c_i]} \cap x^{[c_j]} \in Q^+$. Therefore, all variables in c are set to 1 by $x^{[c_i]}$ and $x^{[c_j]}$. Hence, they falsify the same clause which is a contradiction by the remark above. Hence, all assignments in Q^+ are positive for f.

It is left to show that no anti-monotone CNF g s.t. $|g|_{anti-monCNF} \leq m$ is consistent with f over Q. Fix any $g = c'_1\ \wedge\ \ldots\ \wedge\ c'_l$ with $l \leq m$. If g is consistent with Q^-, then there is a $c' \in g$ falsified by two different $x^{[c_i]}, x^{[c_j]} \in Q^-$ (because we have $m+1$ assignments in Q^- but less than $m+1$ clauses in g). Since they falsify c', all variables in c' are set to 1 in both $x^{[c_i]}$ and $x^{[c_j]}$. Therefore, all variables in c' are set to 1 in their intersection $x^{[c_i]} \cap x^{[c_j]}$. Hence, clause c' (and therefore g) is falsified by $x^{[c_i]} \cap x^{[c_j]}$. Thus, $x^{[c_i]} \cap x^{[c_j]} \in Q^+$ is negative for g and g is not consistent with f over Q. □

By duality of the boolean operators and DNF/CNF representations we get that Monotone CNF, monotone DNF and anti-monotone DNF have polynomial certificates of size $O(min(mn, m^2))$.

3.1 Unate CNF

Definition 3. *Let $a, x, y \in \{0,1\}^n$ be three assignments. The inequality between assignments $x \leq_a y$ is defined as $x \oplus a \leq y \oplus a$, where $\leq$ is the bit-wise standard relational operator and $\oplus$ is the bit-wise exclusive OR.*

Definition 4. *A boolean CNF function f (of arity n) is* unate *if there exists some assignment a such that:*

$$\forall x, y \in \{0,1\}^n : (x <_a y \Longrightarrow f(x) \leq f(y))$$

Equivalently, a variable cannot appear both negated and unnegated in any minimal CNF representation of f. Variables are either monotone or anti-monotone.

The construction above can be used to give certificates for unate CNF. Case 2 in the proof follows along the same lines but reversing $\cap$ with $\cup$ for variables on which the function is anti-monotone. Case 1 can be dealt with using 4 assignments showing that f is neither monotone nor anti-monotone in one of the variables ([13]). We therefore get:

Theorem 3. *Unate CNFs have polynomial certificates of size $O(min(mn, m^2))$.*

4 Saturated Horn CNFs

This section develops a "standardized" representation for Horn expression which can be obtained by an operation we call saturation. We establish properties on saturated expressions that makes it possible to construct a set of certificates in a similar way to the case of anti-monotone CNF.

Definition 5. *Let f be a Horn expression. We define Saturation(f) as the Horn expression returned by the following function:*

```
– Sat := f
– repeat until no changes are made to Sat
    • if there are two clauses $s_i \to b_i$ and $s_j \to b_j$ in Sat s.t.
      (i) $b_i \neq b_j$, (ii) $s_j \subseteq s_i$ and (iii) $b_j \notin s_i$ then
        ∗ $s'_i := s_i \cup \{b_j\}$
        ∗ replace $s_i \to b_i$ with $s'_i \to b_i$ in Sat.
– return Sat
```

Example 1. Notice that an expression can have many possible saturations. As an example, take $f = \{a \to b, a \to c\}$; this expression has two possible saturations: $Sat_1 = \{ac \to b, a \to c\}$ and $Sat_2 = \{a \to b, ab \to c\}$. Clearly, the result depends on the order in which we saturate clauses.

The proofs of the next two lemmas are omitted due to space limit; both can be done by induction on the number of modifications in the saturation process.

Lemma 1. *Every Horn expression is logically equivalent to its saturation.*

Notice that we use the notion of a "sequential" saturation in the sense that we use the updated expression to continue the process of saturation. There is a notion of "simultaneous" saturation that uses the original expression to saturate all the clauses. Lemma 1 does not hold for simultaneous saturation. An easy example illustrates this. Let $f = \{a \to b, a \to c\}$. Clearly, $SimSat(f) = \{ac \to b, ab \to c\}$ is not logically equivalent to f (notice $f \models a \to b$ but $SimSat(f) \not\models a \to b$).

Definition 6. *An expression f is saturated iff $f = Saturation(f)$.*

Lemma 2. *If a Horn expression f is non-redundant, then all of its saturations are non-redundant, too.*

The converse of the previous lemma does not hold. That is, there are redundant expressions f with non-redundant saturations. As an example: $f = \{ab \to c, c \to d, ab \to d\}$ is clearly redundant since the third clause $ab \to d$ can be deduced from the first two. If we saturate the first clause with the third, we obtain: $Saturation(f) = \{abd \to c, c \to d, ab \to d\}$ which is not redundant! However, if we saturate the third clause with the first, we obtain a redundant saturation $Saturation'(f) = \{ab \to c, c \to d, \underline{abc \to d}\}$

Lemma 3. *Let f be a non-redundant Horn expression. Let $s_i \to b$ and $s_j \to b$ be any two distinct clauses in f with the same consequent. Then, $s_i \not\subseteq s_j$.*

Proof. If $s_i \subseteq s_j$, then $s_i \to b$ subsumes $s_j \to b$ and f is redundant. □

Lemma 4. *Let f be a non-redundant, saturated Horn expression. Let c be any clause in f. Let $x^{[c]}$ be the assignment that sets to one exactly those variables in the antecedent of c. Then, $x^{[c]}$ falsifies c but satisfies every other clause c' in f.*

Proof. Let $c = s \to b$. Clearly, $x^{[c]}$ falsifies c: its antecedent is satisfied but its consequent is not. It also satisfies every other clause $c' = s' \to b'$ in f. To see this, we look at the following two cases: if $s' \not\subseteq s$, there is a variable in s' not in s. Hence $x^{[c]} \not\models s'$ and $x^{[c]} \models c'$. Else, $s' \subseteq s$ and Lemma 3 guarantees that $b \neq b'$ (otherwise there would be a redundant clause in f). Furthermore, $b' \in s$ (otherwise f would not be saturated). Thus, $x^{[c]} \models b'$ and $x^{[c]} \models c'$. □

5 Certificates for Horn CNF

We proceed with the construction of the certificates for Horn CNFs. The following characterization is due to [22], although it was stated in a different context and in more general terms. It was further explored by [16]. Finally, a proof adapted to our setting can be found e.g. in [19]. Horn CNF expressions are characterized by

$$\forall x, y \in \{0,1\}^n : (x \models f) \wedge (y \models f) \Longrightarrow (x \cap y \models f) \tag{1}$$

Theorem 4. *Horn CNFs have polynomial size certificates with* $p(m,n) = m(n+1)$ *and* $q(m,n) = \binom{m+1}{2} + m + 1$.

Proof. Fix $m, n > 0$. Fix any $f \subseteq \{0,1\}^n$ s.t. $|f|_{hornCNF} > p(m,n) = m(n+1)$. Again, we proceed by cases.
Case 1. f is not Horn. In this case, there must exist two assignments $x, y \in \{0,1\}^n$ s.t. $x \models f$ and $y \models f$ but $x \cap y \not\models f$ (otherwise f would be Horn). Let $Q = \{x, y, x \cap y\}$. Notice that by (1) no Horn CNF can be consistent with Q. Moreover, $|Q| \leq q(m,n)$.
Case 2. f is Horn. Let $c_1 \wedge c_2 \wedge ... \wedge c_{k'}$ be a minimal, saturated representation of f. Notice that $k' \geq m(n+1)+1$ since $|f|_{hornCNF} > p(m,n) = m(n+1)$ and saturation does not produce redundant clauses when starting from a non-redundant representation (see Lemma 2). Since there are more than $m(n+1)$ clauses, there must be at least $m+1$ clauses sharing a single consequent in f (there are at most $n+1$ different consequents among the clauses in f – we must count the constant `false`, too). Let these clauses be $c_1 = s_1 \to b, ..., c_k = s_k \to b$, with $k \geq m+1$. Define assignment $x^{[c_i]}$ as the assignment that sets to 1 exactly those variables that appear in c_i's antecedent. For example, if $n = 5$ and $c_i = v_3 v_5 \to v_2$ then $x^{[c_i]} = 00101$. Define the set $Q = Q^+ \cup Q^-$ where

$$Q^- = \left\{ x^{[c_i]} \,\middle|\, 1 \leq i \leq m+1 \right\} \text{ and } Q^+ = \left\{ x^{[c_i]} \cap x^{[c_j]} \,\middle|\, 1 \leq i < j \leq m+1 \right\}.$$

Notice that $|Q| = |Q^+| + |Q^-| \leq \binom{m+1}{2} + m + 1 = q(m,n)$. The assignments in Q^- are negative for f, since $x^{[c_i]}$ clearly falsifies clause c_i (and hence it falsifies f). The assignments in Q^+ are positive for f. To see this, we show that every assignment in Q^+ satisfies every clause in f. Take any assignment $x^{[c_i]} \cap x^{[c_j]} \in Q^+$. For clauses c with a different consequent than c_i (thus $c \neq c_i, c \neq c_j$), Lemma 4 shows that $x^{[c_i]} \models c$ and $x^{[c_j]} \models c$. Since c is a Horn clause, we obtain that $x^{[c_i]} \cap x^{[c_j]} \models c$. For clauses with the same consequent as c_i (and c_j), we have two cases. Either (1) $c \neq c_i$ or (2) $c \neq c_j$. If (1) holds, then Lemma 3 guarantees that $s \not\subseteq s_i$, where s is c's antecedent. Therefore some variable in s is set to 0 by $x^{[c_i]}$ and hence by $x^{[c_i]} \cap x^{[c_j]}$, too. Thus, $x^{[c_i]} \cap x^{[c_j]} \models c$. Case (2) is analogous. Hence, all assignments in Q^+ are positive for f.

It is left to show that no Horn CNF g s.t. $|g|_{hornCNF} \leq m$ is consistent with f over Q. Fix any $g = c'_1 \wedge ... \wedge c'_l$ with $l \leq m$. If g is consistent with Q^-, then there is a $c' \in g$ falsified by two different $x^{[c_i]}, x^{[c_j]} \in Q^-$ (because we have $m+1$ assignments in Q^- but less than $m+1$ clauses in g). Since they falsify c', all variables in the antecedent of c' are set to 1 in both $x^{[c_i]}$ and $x^{[c_j]}$. Also, in both assignments the consequent of c' is set to 0. Therefore, the assignment $x^{[c_i]} \cap x^{[c_j]}$ sets all variables in the antecedent of c' to 1 and the consequent to 0, too. Hence, clause c' (and therefore g) is falsified by $x^{[c_i]} \cap x^{[c_j]}$. Thus, $x^{[c_i]} \cap x^{[c_j]} \in Q^+$ is negative for g and g is not consistent with f over Q. □

6 Learning from Entailment

The learning model we have been using, where an example is an assignment to propositional variables is natural in the propositional setting. Models for learnability of first order logic have generalize this in two ways [8]. *Learning from interpretations* is a direct lifting of the above. In *learning from entailment*, formalized in [11] examples are *clauses*. An example is positive iff it is implied by the target expression. This model has been widely used in inductive logic programming both in theoretical studies and in practice. We can adapt the query model to treat such examples in a natural manner. Membership queries accept clauses and give their classification and equivalence queries return clauses as counterexamples.

We present a general transformation that allows us to obtain an entailment certificate from an interpretation certificate for propositional logic. Similar observations have been made before in different context (e.g. [18, 8]) where one transforms efficient algorithms not just certificates. Note however, that for efficiency we must be able to solve the implication problem for the language of hypotheses used by the algorithm.

Definition 7. *Let x be an interpretation. Then $ones(x)$ is the set of variables that are set in x.*

Lemma 5. *Let f be a boolean expression and x an interpretation. Then,*

$$x \models f \iff f \not\models ones(x) \rightarrow \bigvee_{b \notin ones(x)} b.$$

Proof. Suppose $x \models f$. By construction, $x \not\models ones(x) \rightarrow \bigvee_{b \notin ones(x)} b$. Suppose by way of contradiction that $f \models ones(x) \rightarrow \bigvee_{b \notin ones(x)} b$. But since $x \not\models ones(x) \rightarrow \bigvee_{b \notin ones(x)} b$ we conclude that $x \not\models f$, which contradicts our initial assumption. Now, suppose $x \not\models f$. Hence, there is a clause $s \rightarrow \bigvee_i b_i$ in f falsified by x. This can happen only if $s \subseteq ones(x)$ and $b_i \notin ones(x)$ for all i. Clearly, $s \rightarrow \bigvee_i b_i \models ones(x) \rightarrow \bigvee_{b \notin ones(x)} b$. Therefore $f \models ones(x) \rightarrow \bigvee_{b \notin ones(x)} b$ and the result follows. □

Theorem 5. *Let S be an interpretation certificate for a boolean expression f w.r.t. a class $\mathcal{C}$ of boolean expressions. Then, the set of clauses $\{ones(x) \rightarrow \bigvee_{b \notin ones(x)} b \mid x \in S\}$ is an entailment certificate for f w.r.t. $\mathcal{C}$.*

Proof. If S is an interpretation certificate for f w.r.t. some class $\mathcal{C}$ of propositional expressions, then for all $g \in \mathcal{C}$ there is some assignment $x \in S$ such that $x \models f$ and $x \not\models g$ or vice versa. Therefore, by Lemma 5, it follows that $f \not\models ones(x) \rightarrow \bigvee_{b \notin ones(x)} b$ and $g \models ones(x) \rightarrow \bigvee_{b \notin ones(x)} b$ or vice versa. Given the arbitrary nature of g the theorem follows. Moreover, both sets have the same cardinality. □

7 Certificate Size Lower Bounds

The certificate results above imply that Monotone and Horn CNF are learnable with queries but as mentioned in the introduction this was already known. It is therefore useful to review the relationship between the certificate size of a class and its query complexity. From [12, 14] we know that if $CS(\mathcal{C})$ is the certificate size of a certain class $\mathcal{C}$, then its query complexity (denoted $QC(\mathcal{C})$) satisfies:

$$CS(\mathcal{C}) \leq QC(\mathcal{C}) \leq CS(\mathcal{C}) \log(|\mathcal{C}|)$$

For the class of monotone DNF there is an algorithm that achieves query complexity $O(mn)$ [26, 1]. Since $\log(|MonotoneDNF_m|) = \Theta(mn)$, a certificate result is not likely to improve the known learning complexity. In the case of Horn CNF, there is an algorithm that achieves query complexity $O(m^2n)$ [2]. Since again $\log(|HornCNF_m|) = \Theta(mn)$ improving on known complexity would require a certificate for Horn of size $o(m)$. The results in this section show that this is not possible and in fact that our certificate constructions are optimal.

In particular, for every m, n with $m < n$ we construct an n-variable monotone DNF f of size greater than m and show that any certificate that f has more than m terms must have cardinality at least $q(m, n) = m + 1 + \binom{m+1}{2}$. We also show that if $m > n$ then there is a monotone DNF of size greater than m that requires a certificate of size $\Omega(mn)$. These results also apply to both unate and Horn CNF/DNF as described below. We first give the result for $m < n$:

Theorem 6. *Any certificate construction for monotone DNF for $m < n$ with $p(m, n) = m$ has size $q(m, n) \geq m + 1 + \binom{m+1}{2}$.*

Proof. Let $X_n = \{x_1, .., x_n\}$ be the set of n variables and let $m < n$. Let $f = t_1 \vee \cdots \vee t_{m+1}$ where t_i is the term containing all variables (unnegated) except x_i. Such a representation is minimal and hence f has size exactly $m + 1$. We show that any set with less than $m + 1 + \binom{m+1}{2}$ assignments cannot certify that f has more than m terms. That is, for any set Q of size less than $m + 1 + \binom{m+1}{2}$ assignments we will show that there is a monotone DNF with at most m terms consistent with f over Q.

If Q contains at most m positive assignments of weight $n - 1$ then it easy to see that the function with minterms corresponding to these positive assignments is consistent with f over Q. Hence we may assume that Q contains at least $m+1$ positive assignments of weight $n-1$. Since f only has $m+2$ positive assignments, one of which is 1^n, Q must include all $m + 1$ positive assignments corresponding to the minterms of f. Thus if $|Q| < m + 1 + \binom{m+1}{2}$ then Q must contain strictly less than $\binom{m+1}{2}$ negative assignments. Notice that all the intersections between pairs of positive assignments of weight $n - 1$ are different and there are $\binom{m+1}{2}$ such intersections. It follows that Q must be missing some intersection between some pair of positive assignments in Q. But then there is an m-term monotone DNF consistent with Q which uses one term for the missing intersection and $m - 1$ terms for the other $m - 1$ positive assignments. □

We can strengthen the previous theorem so that for every n a fixed function f serves for all $m < n$. The motivation behind this is that the lower bound in Theorem 6 implies a lower bound on the query complexity of any strongly proper learning algorithm [15, 23]. Such algorithms are only allowed to output hypotheses that are of size at most that of the target expression; this is in contrast with the usual scenario in which learning algorithms are allowed to present hypotheses of size polynomial in the size of the target. In the following certificate lower bound we use a function f of DNF size n, so the resulting lower bound for learning algorithms applies to algorithms which may use hypotheses of size at most $n-1$ (even if the target function is much smaller).

Theorem 7. *Any certificate construction for monotone DNF for $m < n$ with $p(m,n) < n$ has size $q(m,n) \geq m+1+\binom{m+1}{2}$.*

Proof. Let $q(m,n) = m+1+\binom{m+1}{2}$ and let f be defined as $f = \bigvee_{i \in \{1,..,n\}} t_i$ where t_i is the term containing all variables (unnegated) except x_i. Clearly, all t_i are minterms, f has size exactly n and f is monotone. We will show that for any $m < n$ and any set of assignments Q of cardinality strictly less than $q(m,n)$, there is a monotone function g of at most m terms consistent with f over Q.

We first claim that w.l.o.g. we can assume that all the assignments in the potential certificate Q have exactly one bit set to zero (positive assignments) or two bits set to zero (negative assignments). We prove that if Q contains the positive assignment 1^n, or a negative assignment with more than 2 bits set to zero, then we can replace these by appropriate assignments with exactly 1 or 2 zeros so that any monotone function g consistent with the latter set of assignments (call it Q') is also consistent with Q. Suppose first that we have a function g consistent with f over Q' where the positive assignment $b \in Q$ with all its bits set to 1 has been changed to b' with just one bit set to 0 (choose it arbitrarily). Since g is monotone, g is consistent with f over Q', $b' \leq b$, and $g(b') = 1$, it follows that $g(b) = 1$ and hence g is also consistent with f over the initial Q. Now suppose that we have a function g consistent with the set Q where one negative assignment a with more than two bits set to zero has been (arbitrarily) changed so that some of the extra zero bits are set to one (call the new assignment a'). Since g is consistent with Q', $g(a') = 0$, and since g is also monotone and $a \leq a'$ it follows that $g(a) = 0$, too. Hence, g is consistent with Q in this second case. By induction, our assumption results in no loss of generality.

We may assume, then, that Q is a set of fewer than $q(m,n)$ assignments each of which has either 1 or 2 zeros. We model the problem of finding a suitable monotone function as a graph coloring problem. We map Q into a graph $G_Q = (V,E)$ where $V = \{p \in Q \mid f(p) = 1\}$ and $E = \{(p_1,p_2) \mid \{p_1,p_2,p_1 \cap p_2\} \subseteq Q\}$. Let $|V| = v$ and $|E| = e$.

First we show that if G_Q is m-colorable then there is a monotone function g of DNF size at most m that is consistent with f over Q. It is sufficient that for each color we find a single term t_c that (1) is satisfied by the positive assignments in Q that have been assigned some color c, with the additional condition that (2) t_c is not satisfied by any of the negative assignments in Q. We define t_c as the minterm corresponding to the intersection of all the assignments colored c by the m-coloring. Property (1) is clearly satisfied, since no variable set to zero in any of the assignments is present in t_c. To see that (2) holds it suffices to notice that the assignments colored c form an independent set in G_Q and therefore none of their pair-wise intersections is in Q. By the assumption no negative point below the intersections is in Q either. The resulting consistent function g contains all minterms t_c. Since the graph is m-colorable, g has at most m terms.

It remains to show that G_Q is m-colorable. Note that the condition $|Q| < q(m,n)$ translates into $v + e < q(m,n)$ in G_Q. If $v \leq m$ then there is a trivial m-coloring. For $v \geq m+1$, it suffices to prove the following: any v-node graph with $v \geq m+1$ with at most $\binom{m+1}{2} + m - v$ edges is m colorable. We prove this by induction on v.

The base case is $v = m+1$; in this case since the graph has at most $\binom{m+1}{2} - 1$ edges it can be colored with only m colors (reuse one color for the missing edge). For the inductive step, note that any v-node graph which has at most $\binom{m+1}{2} + m - v$ edges must have some node with fewer than m neighbors (otherwise there would be at least $vm/2$ nodes in the graph, and this is more than $\binom{m+1}{2} + m - v$ since v is at least $m+2$ in the inductive step). By the induction hypothesis there is an m-coloring of the $(v-1)$-node graph obtained by removing this node of minimum degree and its incident edges. But since the degree of this node was less than m in G_Q, we can color G_Q using at most m colors. □

Finally, we give an $\Omega(mn)$ lower bound on certificate size for monotone DNF for the case $m > n$. Like Theorem 6 this result gives a lower bound on query complexity for any strongly proper learning algorithm.

Theorem 8. *Any certificate construction for monotone DNF for $m > n$ with $p(m,n) = m$ has size $q(m,n) = \Omega(mn)$.*

Proof. Fix any constant k. We show that for all n and for all $m = \binom{n}{k} - 1$, there is a function f of monotone DNF size $m+1$ such that any certificate showing that f has more than m terms must contain $\Omega(nm)$ assignments.

Fix n, fix k. We define f as the function whose satisfying assignments have at least $n-k$ bits set to 1. Notice that the size of f is exactly $\binom{n}{k} = m+1$. Let P be the set of assignments corresponding to the minterms of f, i.e. P consists of all assignments that have exactly $n-k$ bits set to 1. Let N be the set of assignments that have exactly $n-(k+1)$ bits set to 1. Notice that f is positive for the assignments in P but negative for those in N. Clearly, assignments in P are minimal weight positive assignments and assignments in N are maximal weight negative assignments. As in the previous proof, we may assume w.l.o.g. that any certificate Q contains assignments in $P \cup N$ only. Notice, too, that $|P| = \binom{n}{k}$ and $|N| = \frac{(m+1)(n-k)}{k+1} = \binom{n}{k+1} = \Omega(mn)$ for constant k. Moreover, any assignment in N is the intersection of two assignments in P.

Let $Q \subseteq P \cup N$. If Q has at most m positive assignments then it is easy to construct a function consistent with Q regardless of how negative examples are placed. Otherwise, Q contains all the $m+1$ positive assignments in P and the rest are assignments in N. If Q misses any assignment in N then we build a consistent function as follows: use the minterm corresponding to the missing intersection to "cover" two of the positive assignments with just one term. The remaining $m-1$ positive assignments in P are covered by one minterm each. Hence, any certificate Q must contain $P \cup N$ and thus is of size $\Omega(nm)$. □

We note that all of the lower bounds above apply to unate or Horn CNF/DNF as well. This follows from the fact that monotone CNF/DNF is a special case of unate or Horn CNF/DNF and that the function f is outside the class (has size more than m in all cases).

It is known [20] that the VC-dimension of m-term monotone DNF is $\Omega(mn)$, so a result in [21] implies a $\Omega(mn)$ lower bound on the number of queries to learn this class. Our result gives an alternative proof of this fact. For the Horn case we have a gap between the $\Omega(mn)$ and $O(m^2)$ bounds on certificate size, and the $O(m^2n)$ query complexity of the algorithm from [2]. Closing this gap is an interesting problem for future work.

8 An Exponential Lower Bound for Renamable Horn

In this section we show that Renamable Horn CNF expressions do not have polynomial certificates. This answers an open question posed in [9] and implies that the class of Renamable Horn CNF is not exactly learnable using a polynomial number of membership and equivalence queries [14, 12].

Definition 8. *A boolean CNF function f (of arity n) is* renamable Horn *if there exists some assignment c such that f_c is Horn, where $f_c(x) = f(x \oplus c)$ for all $x \in \{0,1\}^n$. In other words, the function obtained by renaming the variables according to c is Horn. We call such an assignment c an orientation for f.*

To show non-existence of certificates, we need to prove the negation of the property in Definition 1, namely: for all two-variable polynomials $p(\cdot,\cdot)$ and $q(\cdot,\cdot)$ there exist $n, m > 0$ and a boolean function $\hat{f} \subseteq \{0,1\}^n$ s.t. $\left|\hat{f}\right|_{ren\mathcal{H}} > p(m,n)$ such that for every $Q \subseteq \{0,1\}^n$ it holds (1) $|Q| > q(m,n)$ or (2) some $g \in \mathcal{C}_m$ is consistent with f over Q.

In particular, we define an $\hat{f}$ that is not renamable Horn, so that $\left|\hat{f}\right|_{ren\mathcal{H}} = \infty > p(m,n)$ holds for any function $p(m,n)$.

Hence, we need to show: there exist $n, m > 0$ and a non-renamable Horn $\hat{f} \subseteq \{0,1\}^n$ s.t. if no $g \in \mathcal{C}_m$ is consistent with $\hat{f}$ over some set of assignments Q, then $|Q| > q(m,n)$ for every polynomial $q(m,n)$. We say that a set Q such that no $g \in \mathcal{C}_m$ is consistent with $\hat{f}$ over Q is a *certificate that $\hat{f}$ is not small renamable Horn.*

What we actually show is: for each n which is a multiple of 3, there exists a non-renamable Horn $\hat{f} \subseteq \{0,1\}^n$ s.t. if no $g \in \mathcal{C}_{n^6}$ is consistent with $\hat{f}$ over some set of assignments Q, then $|Q| \geq \frac{1}{3}2^{2n/3}$. Equivalently, for every such n every certificate Q that $\hat{f}$ is not a renamable Horn CNF function of size n^6 has to be of super-polynomial (in fact exponential) size. This is clearly sufficient to prove the non-existence of polynomial certificates for renamable Horn boolean functions. The following lemma due to Feigelson will be useful:

Lemma 6 (Lemma 44 from [9], page 86). *Let f be a renamable Horn function. Then there is an orientation c for f such that $c \models f$.*

Definition 9. *The function $\hat{f}$ which we use is as follows: Let $n = 3k$ for some $k \geq 1$. We define $\hat{f} : \{0,1\}^n \to \{0,1\}$ to be the function whose only satisfying assignments are $0^k1^k1^k, 1^k0^k1^k$, and $1^k1^k0^k$.*

Lemma 7. *The function $\hat{f}$ defined above is not renamable Horn.*

Proof. To see that a function f is not renamable Horn with orientation c it suffices to find a triple (p_1, p_2, q) such that $p_1 \models f$, $p_2 \models f$ but $q \not\models f$ where $q = p_1 \cap_c p_2$. By Lemma 6 it is sufficient to check that the three positive assignments are not valid orientations for f:
The triple $(1^k1^k0^k, 1^k0^k1^k, 1^k1^k1^k)$ rejects $c = 0^k1^k1^k$.
The triple $(0^k1^k1^k, 1^k1^k0^k, 1^k1^k1^k)$ rejects $c = 1^k0^k1^k$.
The triple $(0^k1^k1^k, 1^k0^k1^k, 1^k1^k1^k)$ rejects $c = 1^k1^k0^k$. □

The following lemma is an extension of Lemma 57 from [9]. We say that a triple (p_1, p_2, q) such that $p_1 \models f$, $p_2 \models f$ but $q \not\models f$ is *suitable* for c if $q \leq_c p_1 \cap_c p_2$.

Lemma 8. *If Q is a certificate that $\hat{f}$ is not small renamable Horn with orientation c, then Q includes a suitable triple (p_1, p_2, q) for c.*

Proof. Following the same strategy as in [9], suppose that a certificate Q that $\hat{f}$ is not small renamable Horn with orientation c does not include a suitable triple (p_1, p_2, q) for c. That is, $p_1 \models \hat{f}$, $p_2 \models \hat{f}$ but $q \not\models \hat{f}$ where $q \leq_c p_1 \cap_c p_2$. Feigelson [9] defines a function g that is consistent with $\hat{f}$ on Q as follows:

$$g(x) = \begin{cases} 1 \text{ if } x \in Q \text{ and } x \models \hat{f} \\ 1 \text{ if } x \leq_c (s_1 \cap_c s_2) \text{ for any } s_1, s_2 \in Q \text{ s.t. } s_1 \models \hat{f} \text{ and } s_2 \models \hat{f} \\ 0 \text{ otherwise.} \end{cases}$$

The function g is consistent with Q since by assumption no negative example is covered by the second condition. Feigelson [9] shows that g is renamable Horn with orientation c; here, we show that it is also *small.* We use the fact that our particular $\hat{f}$ is designed to have very few positive assignments. First notice that g only depends on the positive assignments in Q. Moreover, these must be positive assignments for $\hat{f}$. Suppose that Q contains any $l \leq 3$ of these positive assignments. Let these be $x_1, .., x_l$. A DNF representation for g is:

$$g = \bigvee_{1 \leq i \leq l} t_i \vee \bigvee_{1 \leq i < j \leq l} t_{i,j}$$

where t_i is the term that is true for the assignment x_i only and $t_{i,j}$ is the term that is true for the assignment $x_i \cap_c x_j$ and all assignments below it (w.r.t. c). Notice that we can represent this with just one term by removing literals that correspond to maximal values (w.r.t. c).

Since $l \leq 3$, g has at most $3 + \binom{3}{2} = 6$ terms. Hence, g has CNF size at most n^6 (multiply out all terms to get the clauses). Now we use the fact that if there is a CNF formula representing g of size at most n^6, then there must be a (syntactically) renamable Horn representation $\tilde{g}$ for g which is also of size at most n^6. (It is well known that if a function h is Horn and g is a non-Horn CNF representation for h, then every clause in g can be replaced with a Horn clause which uses a subset of its literals; see e.g. [22] or Claim 6.3 in [19].) We arrive at a contradiction: Q is not a certificate that $\hat{f}$ is not small renamable Horn with orientation c since $\tilde{g}$ is not rejected. □

Theorem 9. *For all $n = 3k$, there is a function $\hat{f} : \{0,1\}^n \to \{0,1\}$ which is not renamable Horn such that any certificate Q showing that the renamable Horn size of $\hat{f}$ is more than n^6 must have $|Q| \geq \frac{1}{3}2^{2n/3}$.*

Proof. The Hamming distance between any two positive assignments for $\hat{f}$ is $2n/3$. Since, as observed in [9, 10], the intersection of two different bits equals the minimum of the two bits, any triple can be suitable for at most $2^{n/3}$ orientations. A negative example in Q can appear in at most 3 triples (only 3 choices for p_1, p_2). Hence any negative example in Q contributes to at most $3 \cdot 2^{n/3}$ orientations. The theorem follows since we must reject all orientations. □

Corollary 1. *Renamable Horn CNFs do not have polynomial size certificates.*

Acknowledgments

We thank José Luis Balcázar for comments on earlier drafts, and Lisa Hellerstein for raising the question of certificate lower bound for strong learnability.

References

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

[2] D. Angluin, M. Frazier, and L. Pitt. Learning conjunctions of Horn clauses. *Machine Learning*, 9:147–164, 1992.

[3] Dana Angluin. Queries revisited. In *Proceedings of the 12th International Conference on ALT*, volume 2225 of *Lecture Notes in Computer Science*, pages 12–31, Washington, DC, USA, November 25-28 2001. Springer.

[4] Marta Arias and Roni Khardon. Learning closed horn expressions. *Information and Computation*, pages 214–240, 2002.

[5] Hiroki Arimura. Learning acyclic first-order Horn sentences from entailment. In *Proceedings of the International Conference on ALT*, Sendai, Japan, 1997. Springer-Verlag. LNAI 1316.

[6] José L. Balcázar, Jorge Castro, and David Guijarro. The consistency dimension and distribution-dependent learning from queries. In *Proceedings of the 10th International Conference on ALT*, Tokyo, Japan, December 6-8 1999. Springer. LNAI 1702.

[7] Nader H. Bshouty. Simple learning algorithms using divide and conquer. In *Proceedings of the Conference on COLT*, 1995.

[8] L. De Raedt. Logical settings for concept learning. *Artificial Intelligence*, 95(1):187–201, 1997. See also relevant Errata (forthcoming).

[9] Aaron Feigelson. On boolean functions and their orientations: Learning, monotone dimension and certificates. Ph.D. Thesis, Northwestern University, Department of Electrical and Computer Engineering. 1998.

[10] Aaron Feigelson and Lisa Hellerstein. Conjunctions of unate DNF formulas: Learning and structure. *Information and Computation*, 140(2):203–228, 1988.

[11] M. Frazier and L. Pitt. Learning from entailment: An application to propositional Horn sentences. In *Proceedings of the International Conference on Machine Learning*, pages 120–127, Amherst, MA, 1993. Morgan Kaufmann.

[12] T. Hegedus. On generalized teaching dimensions and the query complexity of learning. In *Proceedings of the 8th Annual Conference on Computational Learning Theory (COLT'95)*, pages 108–117, New York, NY, USA, July 1995. ACM Press.

[13] Lisa Hellerstein. On generalized constraints and certificates. *Discrete Mathematics*, 226:211–232, 2001.

[14] Lisa Hellerstein, Krishnan Pillaipakkamnatt, Vijay Raghavan, and Dawn Wilkins. How many queries are needed to learn? *Journal of the ACM*, 43(5):840–862, September 1996.

[15] Lisa Hellerstein and Vijay Raghavan. Exact learning of DNF formulas using DNF hypotheses. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing (STOC-02)*, pages 465–473, New York, May 19–21 2002. ACM Press.

[16] A. Horn. On sentences which are true of direct unions of algebras. *Journal of Symbolic Logic*, 16:14–21, 1956.

[17] R. Khardon. Learning function free Horn expressions. *Machine Learning*, 37:241–275, 1999.

[18] R. Khardon and D. Roth. Learning to reason with a restricted view. *Machine Learning*, 35(2):95–117, 1999.

[19] Roni Khardon and Dan Roth. Reasoning with models. *Artificial Intelligence*, 87(1–2):187–213, 1996.

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

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

[22] J. C. C. McKinsey. The decision problem for some classes of sentences without quantifiers. *J. Symbolic Logic*, 8:61–76, 1943.

[23] Krishnan Pillaipakkamnatt and Vijay Raghavan. On the limits of proper learnability of subclasses of DNF formulas. *Machine Learning*, 25:237, 1996.

[24] C. Reddy and P. Tadepalli. Learning Horn definitions with equivalence and membership queries. In *International Workshop on Inductive Logic Programming*, pages 243–255, Prague, Czech Republic, 1997. Springer. LNAI 1297.

[25] C. Reddy and P. Tadepalli. Learning first order acyclic Horn programs from entailment. In *International Conference on Inductive Logic Programming*, pages 23–37, Madison, WI, 1998. Springer. LNAI 1446.

[26] Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, November 1984.