TUFTS-CS Technical Report 2004-8

September 2004

Polynomial Certificates for Propositional Classes

by

Marta Arias
Dept. of Computer Science
Tufts University
Medford, Massachusetts 02155

Aaron Feigelson
Dept. of
Leydig, Voit and Mayer Ltd.
Chicago, Illinois 60601

Roni Khardon
Dept. of Computer Science
Tufts University
Medford, Massachusetts 02155

Rocco A. Servedio
Dept. of Computer Science
Columbia University
New York, NewYork 10027

Polynomial Certificates for Propositional Classes*

Marta Arias
Department of Computer Science
Tufts University
Medford, MA 02155, USA
marias@cs.tufts.edu

Aaron Feigelson
Leydig, Voit & Mayer, Ltd.
Chicago, IL 60601
arf@alumni.northwestern.edu

Roni Khardon
Department of Computer Science
Tufts University
Medford, MA 02155, USA
roni@cs.tufts.edu

Rocco A. Servedio
Department of Computer Science
Columbia University
New York, NY 10027, USA
rocco@cs.columbia.edu

August 17, 2004

Abstract

This paper studies the complexity of learning classes of expressions in propositional logic from equivalence queries and membership queries. In particular, we focus on bounding the number of queries that are required to learn the class ignoring computational complexity. This quantity is known to be captured by a combinatorial measure of concept classes known as the certificate complexity. The paper gives new constructions of polynomial size certificates for monotone expressions in conjunctive normal form (CNF), for unate CNF functions where each variable affects the function either positively or negatively but not both ways, and for Horn CNF functions. Lower bounds on certificate size for these classes are derived showing that for some parameter settings the new certificates constructions are optimal. Finally, the paper gives an exponential lower bound on the certificate size for a natural generalization of these classes known as renamable Horn CNF functions, thus implying that the class is not learnable from a polynomial number of queries.

*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 A.F. was at the Department of Electrical and Computer Engineering, Northwestern University, and while R.S. was at the Division of Engineering and Applied Sciences, Harvard University.

1 Introduction

This paper is concerned with the model of exact learning from equivalence and membership queries (Angluin, 1988). Since its introduction 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 (Valiant, 1984; Angluin, 1988), unate DNF expressions (Bshouty, 1995), and Horn CNF expressions (Angluin, Frazier, and Pitt, 1992; Frazier and Pitt, 1993). 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" (Reddy and Tadepalli, 1997) the query complexity is either exponential in one of the crucial parameters (e.g. universally quantified variables) (Khardon, 1999; Arias and Khardon, 2002) or the algorithms use additional syntax based oracles (Arimura, 1997; Reddy and Tadepalli, 1998; Rao and Sattar, 1998). 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* (Hellerstein et al., 1996; Hegedűs, 1995); see also (Balcázar, Castro, and Guijarro, 1999; Angluin, 2001). In particular, Hellerstein et al. (1996) and Hegedűs (1995) show that 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 (Feigelson and Hellerstein, 1998). A recent result of Hellerstein and Raghavan (2002) shows that DNF expressions require a super-polynomial number of queries even when the hypotheses are larger than the target function by some factor, albeit the factor is small. The question whether DNF can be learned with hypotheses that are larger than the target by a polynomial factor is a major open question in learning theory.

This paper establishes 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 (Frazier and Pitt, 1993) that is studied extensively in Inductive Logic Programming (see e.g. (De Raedt, 1997)). 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 (Valiant, 1984; Angluin, 1988; Bshouty, 1995; Angluin, Frazier, and Pitt, 1992) 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 (del Val, 2000). While unate CNF and Horn CNF each have polynomial certificates, we give an exponential lower bound on certificate size for renamable Horn CNF. This proves that renamable Horn CNF is not learnable in polynomial time from membership and equivalence queries, and answers an open question posed in (Feigelson, 1998).

2 Preliminaries

2.1 Representation Classes

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 *DNF* expression is a disjunction of terms. A *clause* is a disjunction of literals. A *CNF* expression is a conjunction of clauses. 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 $|f|_{\mathcal{R}} = \infty$.

Next, we present some classes of boolean formulas and their properties. In what follows we use the notation $f(x) = 1$ or $x \models f$ interchangeably, where f is a boolean function and x is an assignment. Both stand for classical formula satisfiability.

A *monotone CNF (DNF)* expression is a CNF (DNF) with no negated variables. Semantically, a function is monotone iff:

$$\forall x, y \in \{0,1\}^n: \text{ if } x \leq y \text{ then } f(x) \leq f(y), \tag{1}$$

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

An *anti-monotone CNF (DNF)* expression is a CNF (DNF) where all variables appear negated. Semantically, a function is anti-monotone iff:

$$\forall x, y \in \{0,1\}^n: \text{ if } x \leq y \text{ then } f(x) \geq f(y). \tag{2}$$

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. Intuitively if a_i, the i'th bit of a is 0 then we get the normal order on this bit. But if $a_i = 1$ we used $1 < 0$ for the corresponding variable. We denote $x <_a y$ iff $x \leq_a y$ but $y \not\leq_a x$.

A boolean CNF function f (of arity n) is *unate* iff there exists some assignment a (called an *orientation* for f) such that

$$\forall x, y \in \{0,1\}^n: \text{ if } x \leq_a y \text{ then } f(x) \leq f(y). \tag{3}$$

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

A term t is a *minterm* for a boolean function f if $t \models f$ but $t' \not\models f$ for every other term $t' \subset t$. It is well known that a unate DNF expression has a unique minimal representation given by the disjunction of its minterms.

A *Horn clause* is a clause in which there is at most one positive literal, and a Horn expression is a conjunction of Horn classes. A Horn clause $(\overline{x}_{i_1} \vee \cdots \vee \overline{x}_{i_k} \vee x_{i_{k+1}})$ is easily seen to be equivalent to the implication $x_{i_1} \cdots x_{i_k} \rightarrow x_{i_{k+1}}$; we refer to $x_{i_1} \cdots x_{i_k}$ as the *antecedent* and to $x_{i_{k+1}}$ as the *consequent* of such a clause. Notice that an anti-monotone CNF expression can be seen as a Horn CNF whose clauses have empty consequents. For example, the anti-monotone CNF $(\bar{a} \vee \bar{b}) \wedge (\bar{b} \vee \bar{c})$ is equivalent to the Horn CNF $(ab \rightarrow \texttt{false}) \wedge (bc \rightarrow \texttt{false})$.

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. It is well known that a function is Horn iff

$$\forall x, y \in \{0,1\}^n : \text{ if } x \models f \text{ and } y \models f, \text{ then } x \cap y \models f \tag{4}$$

The original characterization is due to McKinsey (1943), although it was stated in a different context and in more general terms. It was further explored by Horn (1956). Finally, a proof adapted to our setting can be found e.g. in (Khardon and Roth, 1996).

Let $a, x, y \in \{0,1\}^n$ be three assignments. Let $a[i]$ be the i-th bit of assignment a. The unate intersection $x \cap_a y$ is defined as:

$$(x \cap_a y)[i] = \begin{cases} x[i] \wedge y[i] & \text{if } a[i] = 0 \\ x[i] \vee y[i] & \text{otherwise} \end{cases}$$

It is easy to see that this definition is equivalent to $(x \cap_a y)[i] = ((x[i] \oplus a[i]) \cap (y[i] \oplus a[i])) \oplus a[i]$ and that $(x \cap_a y) \leq_a x$ and $(x \cap_a y) \leq_a y$ so that $\leq_a$ and $\cap_a$ behave like their normal counterparts.

We say that 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. Equivalently, an expression is renamable Horn iff there exists an assignment c such that

$$\forall x, y \in \{0,1\}^n : \text{ if } x \models f \text{ and } y \models f, \text{ then } x \cap_c y \models f. \tag{5}$$

Let $\mathcal{B}$ be any of the classes of propositional expressions defined above; $\mathcal{B}_m$ denotes the subclass of $\mathcal{B}$ whose concepts have size at most m.

2.2 Derivation Graphs

We next introduce the notion of derivation graphs. Derivation graphs are useful in capturing logical consequence of a Horn clause from a Horn expression.

Definition 2.1 *A* derivation *of a clause $C = A \rightarrow a$ from a Horn expression T is a finite directed acyclic graph G with the following properties. Nodes in G are propositional variables.*

The node a is the unique node of out-degree zero. For each node b in G, let $Pred(b)$ be the set of nodes b' in G with edges from b' to b. Then, for every node b in G, either $b \in A$ or $Pred(b) \rightarrow b$ is a clause in T. A derivation G of C from T is minimal *if no proper subgraph of G is also a derivation of C from T.*

It follows from the *Subsumption Theorem for SLD-resolution* (Theorem 7.10 in (Nienhuys-Cheng and De Wolf, 1997)) that if $T \models C$, where T is a Horn expression and C a non-tautological Horn clause, then there exists a derivation of C from T.

2.3 Learning with Queries and Certificates

We briefly review the model of exact learning with equivalence queries and membership queries (Angluin, 1988). Before the learning process starts, a concept $c \in \mathcal{B}$ is fixed. We refer to this concept as the *target* concept. The learning algorithm has access to an equivalence oracle and a membership oracle that provide information about the target concept. In an equivalence query, the learner presents a hypothesis (a formula in $\mathcal{B}$) and the oracle answers `Yes` if it is a representation of the target concept. Otherwise, it answers `No` and provides a counterexample, that is, an example $x \in \{0,1\}^n$ where the target and hypothesis disagree. In a membership query, the learner presents an example and the oracle answers `Yes` or `No` depending on whether the example presented is a member of the target concept. For any target expression in the concept class the learning algorithm is required to identify the target expression and get a `Yes` answer to an equivalence query. When concept classes are parametrized by size we allow the learning algorithm to learn concepts in $\mathcal{B}_m$ using hypotheses in $\mathcal{B}_{p(m,n)}$ for some polynomial $p()$.

Definition 2.2 *The* query complexity *of a concept class $\mathcal{B}$, $QC(\mathcal{B})$, is the minimum number of queries required by any algorithm that learns $\mathcal{B}$ with equivalence and membership queries.*

Finally we define the notion of certificates:

Definition 2.3 *Let $\mathcal{R}$ be a class of representations defining a boolean concept class $\mathcal{B}$. 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 $g \in \mathcal{B}_m$ there is some $x \in Q$ s.t. $g(x) \neq f(x)$. In other words, (2) states that no function in $\mathcal{B}_m$ is consistent with f over Q.*

The certificate size $CS(\mathcal{B})$ is the smallest function $q(m,n)$ for which the above holds.

As mentioned above certificates roughly characterize the number of queries needed to learn a concept class:

Theorem 2.1 (Hegedűs, 1995; Hellerstein et al., 1996)

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

3 Certificates for monotone and unate CNFs

In this section we give constructions of certificates for monotone and unate classes. We present the basic result for the class of anti-monotone CNF so as to make the relation to the certificate for Horn expressions as clear as possible.

Theorem 3.1 *The class of anti-monotone CNF has polynomial size certificates with* $p(m, n) = m$ *and* $q(m, n) = \min\{(m + 1)n, \binom{m+1}{2} + m + 1\}$.

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, $|Q| = 2 \leq q(m, n)$.

Case 2. f is anti-monotone. Let $c_1 \wedge c_2 \wedge .. \wedge c_m \wedge .. \wedge c_k$ be a minimal representation for f. Notice that $k \geq m + 1$ since $|f|_{anti\text{-}monCNF} > p(m, n) = m$.

We give two different constructions for certificates in this case that achieve the two parts in the bound. 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 $c_i = v_3 v_5 \rightarrow \texttt{false}$ then $x^{[c_i]} = 00101$.

Remark 3.1 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.

Let 0_i be the assignment with 0 in position i and 1 elsewhere. For the first construction let $Q = Q^+ \cup Q^-$ where

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

$$Q^+ = \left\{x^{[c_i]} \cap 0_j \,\middle|\, 1 \leq i \leq m + 1 \text{ and } x^{[c_i]}[j] = 1.\right\}$$

Notice that $|Q| \leq (m + 1)n$. By the remark assignments in Q^- are negative, and since these are the maxterms it is also clear that assignments in Q^+ are positive. Any anti-monotone CNF g with at most m clauses will cover two examples $x^{[c_i]}, x^{[c_j]}$ in Q^- with the same term. As a result one of the assignments directly below $x^{[c_i]}$ which is in Q^+ is also falsified by this clause. So g is not consistent with Q.

For the second construction let $Q = Q^+ \cup Q^-$ 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 $|Q| \leq \binom{m+1}{2} + m + 1$. As before the assignments in Q^- are negative for 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. Then 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 $x^{[c_j]}$ and 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. $|g|_{anti\text{-}monCNF} \leq m$ 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 $x^{[c_i]}, x^{[c_j]} \in Q^-$ since we have $m+1$ assignments in Q^- but strictly fewer 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 and f cannot be consistent. ■

By duality of the boolean operators and DNF vs. CNF representations we get

Corollary 3.2 *The classes monotone DNF, anti-monotone DNF, monotone CNF, anti-monotone CNF have certificates of size* $\min\{(m+1)n, \binom{m+1}{2} + m + 1\}$.

Constructing certificates for unate expressions appears harder at first consideration since that are many more g functions that may be consistent with Q. Nonetheless essentially the same construction works here as well. Since for unate classes we define an orientation to transform the function to be monotone rather than anti-monotone, one would need the dual of the previous construction. To make the notation similar to the previous case we present the result for DNF which means taking the dual again so that we can use intersection as before.

Theorem 3.3 *Unate DNFs have polynomial size certificates with* $p(m,n) = m$ *and* $q(m,n) = \min\{(m+1)n, \binom{m+1}{2} + m + 1\}$.

Proof: Fix $m, n > 0$. Fix any $f \subseteq \{0,1\}^n$ s.t. $|f|_{unateDNF} > p(m,n) = m$. Now we proceed by cases.

Case 1. f is not unate. In this case, there must exist four assignments $x, y, z, w \in \{0,1\}^n$ and a position i $(1 \leq i \leq n)$ such that:

- $x[j] = y[j]$ for all $1 \leq j \leq n, j \neq i$ and $x[i] < y[i]$
- $z[j] = w[j]$ for all $1 \leq j \leq n, j \neq i$ and $z[i] > w[i]$
- $f(x) > f(y)$ and $f(z) > f(w)$

Let $Q = \{x, y, z, w\}$. Notice that $|Q| \leq q(m,n)$. To see that no unate DNF can be consistent with f over Q, take any unate DNF g and suppose it is consistent. Let a be an orientation for g. If $a[i] = 0$ (i-th value given by a) then we have that $x \leq_a y$ but $g(x) > g(y)$. If $a[i] = 1$ then $z \leq_a w$ but $g(z) > g(w)$. Therefore there cannot be any unate function consistent with f over Q.

Case 2. f is unate. Let a be the assignment witnessing f being unate. Suppose w.l.o.g. (just rename variables accordingly) that $a = 0^r 1^{n-r}$ where r is the number of monotone variables in f. Suppose that the variables in f are $\{v_1, ..., v_n\}$. Therefore, variables $\{v_1, ..., v_r\}$

appear always positive in f and variables $\{v_{r+1}, ..., v_n\}$ appear always negative. Let $t_1 \vee t_2 \vee ... \vee t_m \vee ... \vee t_k$ be a minimal DNF representation of f. Notice that $k \geq m+1$ since $|f|_{unateDNF} > p(m,n) = m$. Define j-th value of assignment $x^{[t_i]}$ as (for $1 \leq j \leq n$):

$$x^{[t_i]}[j] = \begin{cases} 1 & \text{if } j \leq r \text{ and } v_j \text{ appears in } t_i \\ 0 & \text{if } j \leq r \text{ and } v_j \text{ does not appear in } t_i \\ 0 & \text{if } j > r \text{ and } \bar{v}_j \text{ appears in } t_i \\ 1 & \text{if } j > r \text{ and } \bar{v}_j \text{ does not appear in } t_i \end{cases}$$

Notice that if f does not depend on a variable v_j, so that it does not appear in any of the terms, then its value is fixed to the same value in all the assignments.

Let 0_j be defined as above. For the first construction let $Q = Q^+ \cup Q^-$ where

$$Q^+ = \left\{ x^{[t_i]} \,\middle|\, 1 \leq i \leq m+1 \right\} \text{ and}$$

$$Q^- = \left\{ x^{[t_i]} \cap_a (a \oplus 0_j) \,\middle|\, 1 \leq i \leq m+1 \text{ and } x^{[c_i]}[j] = 1 - a[j]. \right\}$$

Notice that $a \oplus 0_j$ has all bits except the jth at their maximal value so $x^{[t_i]} \cap_a (a \oplus 0_j)$ flips the jth bit in $x^{[t_i]}$ to its minimum value. Each relevant variable has at least one pair of assignments in Q^+, Q^- with Hamming distance 1 showing the direction of its influence. Therefore any unate g consistent with Q must have all variable polarities set correctly. As a result, the argument for the monotone case shows that any unate g with at most m terms over the relevant variables cannot be consistent with Q. Since irrelevant variables have a constant value in Q they cannot affect consistency of any potential g.

For the second construction let

$$Q^+ = \left\{ x^{[t_i]} \,\middle|\, 1 \leq i \leq m+1 \right\} \text{ and}$$

$$Q^- = \left\{ x^{[t_i]} \cap_a x^{[t_j]} \,\middle|\, 1 \leq i < j \leq m+1 \right\}.$$

As before it is easy to see that the assignments in Q^+ are positive and assignments in Q^- are negative for f.

It is left to show that no unate DNF g s.t. $|g|_{unateDNF} \leq m$ is consistent with f over Q. If g is consistent with Q^+, then there is a $t' \in g$ satisfied by two assignments $x^{[t_i]}, x^{[t_j]} \in Q^+$. Since $x^{[t_i]} \models t'$ and $x^{[t_j]} \models t'$ all variables appearing in t' have the same value in $x^{[t_i]}$ and $x^{[t_j]}$ and therefore so does their their intersection $\cap_c$ with respect to any orientation c and in particular for $\cap_a$. Since $x^{[t_i]} \cap_a x^{[t_j]} \in Q^-$, g is not consistent with Q^-. ∎

Corollary 3.4 *The class of Unate CNF has certificates of size* $\min\{(m+1)n, \binom{m+1}{2} + m + 1\}$.

4 Saturated Horn CNFs

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

Definition 4.1 *Let f be a Horn CNF. We define Saturation(f) as the Horn expression returned by the following procedure:*

Saturation(f)

```
Sat ← f
repeat
        if there exist s_i → b_i, s_j → b_j in Sat s.t. b_i ≠ b_j, s_j ⊆ s_i, b_j ∉ s_i
            then s'_i ← s_i ∪ {b_j}
                replace s_i → b_i with s'_i → b_i in Sat.
    until no changes are made to Sat
return Sat
```

By a *saturation of f* we mean any of the possible outcomes of the procedure Saturation(f). Note that we are not proposing to run this procedure but we are simply using it in the analysis that follows. In any case it is clear that the procedure must terminate within $O(mn)$ iterations, where m is the number of clauses in the initial expression, and n is the number of variables.

Example 4.1 Notice that an expression can have many possible saturations. As an example, take $f = \{a \rightarrow b, a \rightarrow c\}$; this expression has two possible saturations: $Sat_1 = \{ac \rightarrow b, a \rightarrow c\}$ and $Sat_2 = \{a \rightarrow b, ab \rightarrow c\}$. Clearly, the result depends on the order in which we saturate clauses.

Lemma 4.1 *Every Horn expression is logically equivalent to its saturation.*

Proof: We show inductively that after every iteration of the main loop in the procedure above the logical value of the expression being computed does not change. Let Sat be the expression before the update and Sat' after. Let $s_i \rightarrow b_i \in Sat$ be the clause updated to $s'_i \rightarrow b_i \in Sat'$. We have to show that $Sat \equiv Sat'$. Since $s_i \rightarrow b_i \models s'_i \rightarrow b_i$ it follows that $Sat \models Sat'$. For the other direction $Sat' \models Sat$ fix an arbitrary x such that $x \models Sat'$. We show that $x \models Sat$. It holds that $x \models C'$ for all clauses $C' \in Sat'$ so we only need to show that $x \models s_i \rightarrow b_i$. The two following cases arise: (1) the extra variable in s'_i (w.r.t. s_i) is set to 1 by x, or (2) it is set to 0. If (1) holds, then it is easy to see that $x \models s_i \rightarrow b_i$ iff $x \models s'_i \rightarrow b_i$ and we conclude $x \models s_i \rightarrow b_i$. If (2) holds, then let $s_j \rightarrow b_j$ be the clause that was used to add the extra variable (b_j) to s'_i. We have seen that $x \models s_j \rightarrow b_j$ and that b_j is set to 0, therefore s_j must be falsified by x (that is, some variable in s_j is set to 0 by x). Notice, too, that $s_j \subseteq s_i$. Hence, some variable in s_i must be set to 0 by x. Thus $x \models s_i \rightarrow b_i$ as required. ■

Example 4.2 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 4.1 does not hold for simultaneous saturation. An easy example illustrates this. Let $f = \{a \rightarrow b, a \rightarrow c\}$. Clearly, $SimSat(f) = \{ac \rightarrow b, ab \rightarrow c\}$ is not logically equivalent to f (notice $f \models a \rightarrow b$ but $SimSat(f) \not\models a \rightarrow b$).

Definition 4.2 *An expression f is saturated iff $f = Saturation(f)$.*

Definition 4.3 *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.*

Lemma 4.2 *Let f be a non-redundant Horn expression. Let $s_i \rightarrow b$ and $s_j \rightarrow b$ 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 $s_i \rightarrow b$ subsumes $s_j \rightarrow b$ and f is redundant. ■

Lemma 4.3 *If a Horn expression f is non-redundant, then all of its saturations are non-redundant.*

Proof: We show that if any fixed but arbitrary saturation of f is redundant (call it Sat'), then f has to be redundant as well. Assume that Sat' is redundant. We argue inductively on the number of changes made to the expression f during the saturation process.

Base case: f is saturated (i.e. $f = Sat'$). Clearly f is redundant if Sat' is.

Step case: f is not saturated (i.e. $f \neq Sat'$). Consider the last change made by the saturation procedure before obtaining Sat'. Let Sat be the expression just before obtaining Sat'; let $s_i \rightarrow b_i \in Sat$ be the clause replaced by $s_i' \rightarrow b_i \in Sat'$ using $s_j \rightarrow b_j \in Sat$. Notice that $s_i' = s_i \cup \{b_j\}$ and that Sat and Sat' coincide in clauses other than $s_i \rightarrow b_i$ and $s_i' \rightarrow b_i$. Since Sat' is redundant, there is a clause $C' \in Sat'$ that can be deduced from the other clauses of Sat'. Therefore, there is a minimal derivation graph G' of $Sat' \setminus C' \vdash C'$. Denote $C \in Sat$ the clause corresponding to C' in Sat'. Now we proceed by cases. In every case we transform G' proving redundancy of the clause C in Sat.

Case 1a. If $s_i' \rightarrow b_i$ does not appear in the derivation graph and $C' \neq s_i' \rightarrow b_i$, then no modification is needed to show that $C = C'$ is redundant in Sat.

Case 1b(i). If $C' = s_i' \rightarrow b_i$ and the added b_j does not appear in G', then no modification is needed and G' shows that $C = s_i \rightarrow b_i$ is redundant in Sat.

Case 1b(ii). If $C' = s_i' \rightarrow b_i$ and the added b_j appears in G', then we just add edges $b \rightarrow b_j$ for every $b \in s_j$ (first add nodes $b \in s_j$ not in G' already). Notice that this is a valid derivation graph for the redundant $C = s_i \rightarrow b_i$ from Sat.

Case 2. Now suppose that the updated clause appears in the proof. Notice that the variable b_j has to be different from the consequent of the redundant clause. If this were not so, we would have a smaller derivation graph, contradicting the fact that we assume a minimal one. Therefore, the clause $s_j \rightarrow b_j$ used to saturate cannot be C' itself. We modify G' in the following way. If the variable b_j has only one edge going to b_i, we simply remove b_j, the edge $b_j \rightarrow b_i$, all edges $* \rightarrow b_j$ reaching b_j and any unconnected parts remaining in the derivation graph. If b_j has more edges pointing at variables other than b_i, we remove the edge $b_j \rightarrow b_i$ but add edges $b \rightarrow b_j$ for every $b \in s_j$ (first adding any $b \in s_j$ not in G' already).

In either case, we obtain that $C \in Sat$ is redundant. Applying the induction hypothesis, we conclude that (the possibly unsaturated version of) C is redundant in the initial f. ■

Example 4.3 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 $Saturation'(f) = \{ab \to c, c \to d, \underline{abc \to d}\}$.

Finally, we observe that saturated expressions share a useful property with monotone expressions:

Lemma 4.4 *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 $x^{[c]} \models c'$. If $s' \subseteq s$ then Lemma 4.2 guarantees that $b \neq b'$ since otherwise there would be a redundant clause in f. Furthermore, $b' \in s$ or f would not be saturated. Thus, $x^{[c]} \models b'$ and $x^{[c]} \models c'$. ■

5 Certificates for Horn CNF

We can now use saturated expressions with the certificate construction developed for the anti-monotone case.

Theorem 5.1 *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. By Eq. (4), there must exist two assignments $x, y \in \{0, 1\}^n$ s.t. $x \models f$ and $y \models f$ but $x \cap y \not\models f$. Let $Q = \{x, y, x \cap y\}$. Again by Eq. (4) no Horn CNF can be consistent with Q. Moreover, $|Q| = 3 \leq q(m, n)$.

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 by Lemma 4.3 saturation does not produce redundant clauses when starting from a non-redundant representation. 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, including the constant `false`). Let these clauses be $c_1 = s_1 \to b, .., c_k = s_k \to b$, with $k \geq m + 1$. As before 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 $x^{[c_i]} = 00101$. Let $Q = Q^+ \cup Q^-$ where

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

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

Notice that $|Q| = |Q^+| + |Q^-| \le \binom{m+1}{2} + m + 1 = q(m,n)$. 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 4.4 shows that $x^{[c_i]} \models c$ and $x^{[c_j]} \models c$. Since c is Horn, $x^{[c_i]} \cap x^{[c_j]} \models c$. For clauses with the same consequent as c_i (and c_j), we have two cases. Either $c \neq c_i$ or $c \neq c_j$. If $c \neq c_i$, then Lemma 4.2 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, $x^{[c_i]} \cap x^{[c_j]} \models c$. The other case is analogous. Hence, all assignments in Q^+ are positive for f. The argument that no Horn CNF g s.t. $|g|_{hornCNF} \le m$ is consistent with f over Q is identical with the anti-monotone case. ■

Remark 5.1 The construction above relies on the fact that we can find many clauses with the same consequent. This fact does not hold in first order logic since the number of possible consequents is not bounded and therefore this hinders generalization. It is thus worth noting that a related construction with slightly worse bounds does not rely on this fact. In this construction we use $p(m,n) = m(n+1)$ and assignments from $m(n+1)+1$ clauses in Q^-. The set Q also includes all their pairwise intersections. Note that assignments in the latter may be either positive or negative since we have not restricted the consequent. However, now we get that a clause of g captures at least $n+2$ assignments. On the other hand since subsumption chains for antecedents (given by the subset relation over variables) are of length at most $n+1$, any set of of clauses of this size must have a pair of clause whose antecedents do not subsume one another. As a result there is at least one pair of clauses with incomparable antecedents, so that the intersection of assignments is positive for f but negative for g and g is not consistent. Unfortunately, subsumption chains for antecedents in first order logic can be long (Arias and Khardon, 2004) so there are still obstacles in lifting the construction.

6 Learning from entailment

Work in inductive logic programming addresses learning formulas in first order logic and several setups for representing examples have been studied. The setup studied above where an example is an assignment in prepositional logic generalizes to using first order structures (also known as interpretations) as examples. The model is therefore known as learning from interpretations (De Raedt and Dzeroski, 1994). In the model of learning from entailment an example is a clause. An clause example is positive if it is implied by the target and negative otherwise. Therefore a certificate in this context is a set of clauses. In particular, as in previous case, for any expression f whose size is more than $p(m,n)$, a set Q of at most $q(m,n)$ clasues must satisfy that for any $g \in \mathcal{B}_m$ at least one element c of Q separates f and g, that is $f \models c$ and $g \not\models c$ or vice versa. We present a general transformation that allows us to obtain an entailment certificate from an interpretation certificate. Similar observations have been made before in different contexts, e.g. (Khardon and Roth, 1999; De Raedt, 1997), where one transforms efficient algorithms instead of just certificates.

Definition 6.1 *Let x be an assignment. Then $ones(x)$ is the set of variables that are set to 1 in x. We slightly abuse notation and write $ones(x)$ to denote also the conjunction of the variables in the set $ones(x)$.*

Lemma 6.1 *Let f be a boolean expression and x an assignment. Then,*

$$x \models f \text{ if and only if } f \not\models (ones(x) \rightarrow \bigvee_{b \notin ones(x)} b).$$

Proof: Suppose $x \models f$. 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 $f \models (ones(x) \rightarrow \bigvee_{b \notin ones(x)} b)$. ■

Theorem 6.2 *Let S be an interpretation certificate for an expression f w.r.t. a class $\mathcal{B}$ of boolean expressions. Then, the set $\{ones(x) \rightarrow \bigvee_{b \notin ones(x)} b \mid x \in S\}$ is an entailment certificate for f w.r.t. $\mathcal{B}$.*

Proof: If S is an interpretation certificate for f w.r.t. some class $\mathcal{B}$ of propositional expressions, then for all $g \in \mathcal{B}$ there is some assignment $x \in S$ such that $x \models f$ and $x \not\models g$ or vice versa. Therefore, by Lemma 6.1, 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. ■

Remark 6.1 In the theorem above we include non-Horn clauses in the certificate. This is necessary since otherwise one cannot distinguish a function f from its Horn least upper bound (Khardon and Roth, 1996; Selman and Kautz, 1996), the function that is equivalent to the conjucntion of all Horn clauses implied by f. For example, one cannot distinguish $f = \{a \rightarrow b, b \rightarrow c \vee d\}$ from $g = \{a \rightarrow b\}$ with Horn clauses only. It is worth noting, however, that the learning algorithm using these certificates can use them while making queries on Horn clauses only. The algorithm in (Hegedűs, 1995; Hellerstein et al., 1996) simulates the Halving Algorithm and asks membership queries on members of the certificates for some functions in the process. For a Horn expression T it holds that $T \models s \rightarrow b_1 \vee ... \vee b_k$ if and only if $T \models s \rightarrow b_i$ for some i. Thus, instead of asking a membership query on $s \rightarrow b_1 \vee ... \vee b_k$, the algorithm can ask k membership queries on $s \rightarrow b_i$ and reconstruct the answer. So while the certificate must include non-Horn clauses, the queries can avoid those.

7 Certificate size lower bounds

The certificate results above imply that unate and Horn CNF are learnable with a polynomial number of queries but as mentioned above this was already known. It is therefore useful to review the relationship between the certificate size of a class and its query complexity. Recall from Theorem 2.1 that we have $CS(\mathcal{B}) \leq QC(\mathcal{B}) \leq CS(\mathcal{B}) \log(|\mathcal{B}|)$. For the class of monotone DNF there is an algorithm that achieves query complexity $O(mn)$ (Valiant, 1984; Angluin, 1988). 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 $O(m^2 n)$ (Angluin, Frazier, and Pitt, 1992). 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. We do this by giving lower bounds on certificate size. Naturally, these also imply lower bounds for the learning complexity.

In particular, for every m, n with $m < n$ we construct an n-variable monotone DNF f of size $\leq n$ and show that any certificate that f has more than m terms must have cardinality at least $q(m,n) = m + 1 + \binom{m+1}{2}$. Recall that in *strongly* proper learning (Hellerstein and Raghavan, 2002; Pillaipakkamnatt and Raghavan, 1996) the hypothesis must have size smaller than or equal to the target. In this case our bound is tight for strongly proper learning and also for algorithms using hypotheses of size up to $n - 1$. For $m > n$ we show that there is a monotone DNF of size $m + 1$ that requires a certificate of size $\Omega(mn)$. Again the bound is tight for strongly proper learning of monotone expressions. The lower bounds apply for Horn expressions as well where for $m > n$ we have a gap between $O(m^2)$ upper bound and $\Omega(mn)$ lower bound. The result for $m < n$ is given in the next two theorems:

Theorem 7.1 *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 fewer 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 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 7.1 implies a lower bound on the query complexity of any strongly proper learning algorithm (Hellerstein and Raghavan, 2002; Pillaipakkamnatt and Raghavan, 1996). 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 7.2 *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 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 argue 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). This follows since 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 which dominate the original assignments to get a set Q'. Now any monotone function g consistent with Q' is also consistent with Q. As a result if Q' is not a certificate then neither is Q.

We next show that if $|Q| < q(m,n)$ then there exists a function g consistent with Q. Now since assignments in Q have either 1 or 2 zeros we can 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 c we find a term t_c that (1) is satisfied by the positive assignments in Q that have been assigned 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 since otherwise there would be at least $vm/2 \geq \frac{(m+2)m}{2} = \frac{(m+1)m}{2} + \frac{m}{2} > \binom{m+1}{2} + m - v$ edges in the graph. 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, we can color G 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 7.1 this result gives a lower bound on query complexity for any strongly proper learning algorithm.

Theorem 7.3 *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.

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. Note that $|P| = \binom{n}{k}$ and $|N| = (m + 1)\frac{n-k}{k+1} = \binom{n}{k+1} = \Omega(mn)$ for constant k. Moreover, any assignment in N is the intersection of two assignments in P.

We next show that any certificate for f must have size at least $|P| + |N|$. As in the previous proof, we may assume w.l.o.g. that any certificate Q contains assignments in $P \cup N$ only. Let $Q \subset P \cup N$. 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 by using 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 $\Omega(nm)$. ■

Finally, we observe that all the lower bounds above apply to unate and Horn expressions as well. This follows from the fact that the function f used in the construction is outside the class (has size more than m in all cases) and that the function g constructed is in the class (since monotone DNF is a special case of unate DNF and Horn DNF). We therefore have:

Corollary 7.4 *Any certificate construction for unate CNF (DNF) and for Horn CNF (DNF) must satisfy the bounds given in Theorems 7.1, 7.2 and 7.3.*

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 of (Feigelson, 1998) and implies that the class of renamable Horn CNF is not exactly learnable using a polynomial number of membership and equivalence queries. In the following let $\mathcal{B}$ be the class of renamable Horn expressions.

To show non-existence of certificates, we need to prove the negation of the property in Definition 2.3, namely: for all two-variable polynomials $p(\cdot,\cdot)$ and $q(\cdot,\cdot)$ there exist $n, m > 0$ and a boolean function $\hat{f} \subseteq \{0,1\}^n$ with $\left|\hat{f}\right|_{\mathcal{B}} > p(m,n)$ such that for every $Q \subseteq \{0,1\}^n$ it holds (1) $|Q| > q(m,n)$ or (2) some $g \in \mathcal{B}_m$ is consistent with f over Q.

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

Hence, we need to show: for every polynomial $q(\cdot,\cdot)$, there exist $n, m > 0$ and a non-renamable Horn $\hat{f} \subseteq \{0,1\}^n$ s.t. if no $g \in \mathcal{B}_m$ is consistent with $\hat{f}$ over some set of assignments Q, then $|Q| > q(m,n)$.

What we actually show is: for each n which is a multiple of 3 we can pick $m = n^6$ and there exists a non-renamable Horn $\hat{f} \subseteq \{0,1\}^n$ s.t. if no $g \in \mathcal{B}_{n^6}$ 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.

We say that a set Q such that no $g \in \mathcal{B}_{n^6}$ is consistent with $\hat{f}$ over Q is a *certificate that $\hat{f}$ is not small renamable Horn.* The following lemma is useful:

Lemma 8.1 *Let f be a renamable Horn function. Then there is an orientation c for f such that $c \models f$.*

Proof: Let c' an orientation of f such that $c' \not\models f$. Let c be the positive assignment of f which is minimal with respect to the partial order $<_{c'}$. Such an assignment is unique. This can be seen via Eq. (5) since if a and b are both positive assignments unrelated in the partial order, then $c'' = a \cap_{c'} b$ is positive.

We claim that c is an orientation for f. It suffices to show $a \cap_{c'} b = a \cap_c b$ for all positive assignments a and b. We show that $(a \cap_{c'} b)[i] = (a \cap_c b)[i]$ for all $1 \leq i \leq n$. If i is such that $c[i] = c'[i]$ then clearly $(a \cap_{c'} b)[i] = (a \cap_c b)[i]$. Let i be such that $c[i] \neq c'[i]$. Then every positive assignment sets the bit i like $c[i]$: if $a[i] \neq c[i]$ then $(a \cap_{c'} c)[i] = c'[i]$ and thus $(a \cap_{c'} c) <_{c'} c$ (strictly), contradicting the minimality of c. Thus $a[i] = b[i] = c[i]$ and $(a \wedge b)[i] = (a \vee b)[i]$, and therefore $(a \cap_c b)[i] = (a \cap_{c'} b)[i]$. ∎

Definition 8.1 *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 8.2 *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 8.1 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 $c = 1^k1^k0^k$. ∎

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 8.3 *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: 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 $q \leq_c p_1 \cap_c p_2$. We define 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.

First we show that the function g is renamable Horn with orientation c. Consider any assignments p_1, p_2 that are positive for g, i.e., $p_1 \models g$ and $p_2 \models g$, and let $t = p_1 \cap_c p_2$. If p_1, p_2 are included in Q, then clearly $t \models g$ by the definition of g. If $p_1 \notin Q$ then $p_1 \leq_c (s_1 \cap_c s_2)$ for some positive $s_1, s_2 \in Q$ (second condition in the definition of g). Since $t \leq_c p_1 \leq_c (s_1 \cap_c s_2)$, then by the definition of g, $t \models g$ as well. The same reasoning applies for the remaining case $p_2 \notin Q$. Hence, g is renamable Horn with orientation c.

Now, we show that g 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). For example, if $k = 2$ and $x_1 = 001111$, $x_2 = 110011$ and $c = 101001$ then $t_1 = \overline{v_1}\,\overline{v_2} v_3 v_4 v_5 v_6$, $x_1 \cap_c x_2 = 101011$, and the only variable at its maximal value is v_5 so $t_{1,2} = v_1 \overline{v_2} v_3 \overline{v_4} v_6$.

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. (McKinsey, 1943) or Claim 6.3 of Khardon and Roth (1996). 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 8.4 *For all $n = 3k$, there is a function $\hat{f} : \{0,1\}^n \rightarrow \{0,1\}$ 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 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), and hence any negative example in Q contributes to at most $3 \cdot 2^{n/3}$ orientations. The theorem follows since we need to reject all orientations. ∎

Corollary 8.5 *Renamable Horn CNFs do not have polynomial size certificates.*

9 Conclusion

The paper provides a study of the certificate complexity of several well known representation classes for propositional expressions. Since certificates are known to characterize the query complexity of exact learning with queries our results have direct implications for learnability. In particular the paper provides certificates constructions and hence upper bounds on their size for monotone, unate and Horn expressions. Lower bounds for these classes are also derived and these are tight in some cases. A lower bound for the class of renamable Horn expressions establishes that the class is not learnable with a polynomial number of queries. The following table summarizes bounds obtained:

Class	*LowerBound*		*UpperBound*	
unate DNF/CNF $m < n$	$\binom{m+1}{2} + m + 1^*$	(Th. 7.2)	$\binom{m+1}{2} + m + 1$	(Th. 3.3)
unate DNF/CNF $m \geq n$	$\Omega(mn)^{**}$	(Th. 7.3)	$O(mn)$	(Th. 3.3)
Horn CNF $m < n$	$\binom{m+1}{2} + m + 1^*$	(Th. 7.2)	$\binom{m+1}{2} + m + 1$	(Th. 5.1)
Horn CNF $m \geq n$	$\Omega(mn)^{**}$	(Th. 7.3)	$\binom{m+1}{2} + m + 1$	(Th. 5.1)
renamable Horn CNF	$\frac{1}{3}2^{2n/3}$	(Th. 8.4)		

* For $p(m, n) < n$.

** Strong certificate size only, i.e. $p(m, n) = m$.

Several interesting questions remain unsolved. For Horn expressions with $m > n$ clauses there is a gap between the bounds of $\Omega(mn)$ and $O(m^2)$. Also except for renamable Horn the lower bounds are for strongly proper learnability or a small expansion in hypothesis size $p(m, n) < n$. It would be interesting to clarify the status of these cases. Identifying the certificate complexity of general DNF would be a big step toward resolving the complexity of learning this class. As mentioned in the introduction known lower and upper bounds on certificate complexity for first order Horn expressions still have an exponential gap and certificates may provide a tool to resolve this question.

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

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

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

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

Arias, M. and R. Khardon. 2002. Learning closed Horn expressions. *Information and Computation*, 178:214–240.

Arias, M. and R. Khardon. 2004. The subsumption lattice and query learning. In *Proceedings of the International Conference on Algorithmic Learning Theory*.

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

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

Bshouty, Nader H. 1995. Simple learning algorithms using divide and conquer. In *Proceedings of the Conference on Computational Learning Theory*.

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

De Raedt, L. and S. Dzeroski. 1994. First order jk-clausal theories are PAC-learnable. *Artificial Intelligence*, 70:375–392.

del Val, A. 2000. On 2-SAT and renamable Horn. In *Proceedings of the National Conference on Artificial Intelligence*.

Feigelson, Aaron. 1998. *On Boolean Functions and their Orientations: Learning, monotone dimension and certificates*. Ph.D. thesis, Northwestern University, Evanston, IL, USA, June.

Feigelson, Aaron and Lisa Hellerstein. 1998. Conjunctions of unate DNF formulas: Learning and structure. *Information and Computation*, 140(2):203–228.

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

Hegedűs, T. 1995. On generalized teaching dimensions and the query complexity of learning. In *Proceedings of the Conference on Computational Learning Theory*, pages 108–117, New York, NY, USA, July. ACM Press.

Hellerstein, L., K. Pillaipakkamnatt, V. Raghavan, and D. Wilkins. 1996. How many queries are needed to learn? *Journal of the ACM*, 43(5):840–862, September.

Hellerstein, Lisa and Vijay Raghavan. 2002. 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. ACM Press.

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

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

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

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

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

Nienhuys-Cheng, S. and R. De Wolf. 1997. *Foundations of Inductive Logic Programming*. Springer-verlag. LNAI 1228.

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

Rao, K. and A. Sattar. 1998. Learning from entailment of logic programs with local variables. In *Proceedings of the International Conference on Algorithmic Learning Theory*, Otzenhausen, Germany. Springer-verlag. LNAI 1501.

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

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

Selman, Bart and Henry Kautz. 1996. Knowledge compilation and theory approximation. *J. of the ACM*, 43(2):193–224.

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