Query Learning and Certificates in Lattices

M. Arias and J.L. Balcázar

LARCA Research Group, Departament LSI
Universitat Politècnica de Catalunya
{marias,balqui}@lsi.upc.edu

Abstract. We provide an abstract version, in terms of lattices, of the Horn query learning algorithm of Angluin, Frazier, and Pitt. To validate it, we develop a proof that is independent of the propositional Horn logic structure. We also construct a certificate set for the class of lattices that generalizes and improves an earlier certificate construction and that relates very clearly with the new proof.

1 Introduction

In the area of Learning via Queries, the most successful protocol so far is via Membership and Equivalence queries. Three major algorithms for this model are the L* algorithm for learning regular sets in terms of deterministic finite automata [1]; the algorithm that learns monotone DNF (closely related to its corresponding PAC version by Valiant [20]); and a sophisticated evolution of the latter which allows for learning Horn formulas [3] (the two variants of the algorithm were named HORN and HORN1 in [3] but most researchers chose to call it "AFP algorithm" after the initials of the authors' names).

Each of these has had a number of interesting applications. In particular, the AFP algorithm turned out to be crucial for developments in other areas such as Knowledge Compilation [19], Intuitionistic Logic Programming [14] or even Databases [18], and similar algorithms exist for learning from entailment [11] and for learning certain description logics [12]. Related research questions, with answers to a couple of them, are proposed in [6] and [17]; one of these contributions is an alternative way of proving the correctness of the AFP algorithm.

Several studies proceeded to analyze combinatorial dimensions of learnability problems (see the survey [4]); initially, they were ways of stating lower bounds aiming at nonlearnability results [2]; it was later found that these dimensions characterized polynomial query learnability, for specific protocols first ([9], [10], [15], [16]) and in more generality later on ([4], [7], [8]). Specifically, for Membership and Equivalence protocols, the certificate size was shown to characterize query complexity [16]. Certificates are, essentially, sets of labeled examples whose correct classification requires a certain size of the representation mechanism under study (precise definitions are given below).

The algorithm for learning deterministic finite automata proceeds by gathering strings that lead to each of the states, plus additional strings showing how each pair of states must be distinguished: in essence, they show how many states

Y. Freund et al. (Eds.): ALT 2008, LNAI 5254, pp. 303–315, 2008.

are required, and can be seen as certificates. The same consideration applies to monotone DNFs, where the minterms, plus the negative examples just below them (or, in other formulations, their pairwise intersections), are both the crucial queries and the certificates lower-bounding the number of necessary terms. Nevertheless, the study of certificates for Horn formulas [5] was somewhat less tightly connected with the learning algorithm, and in fact an interesting research problem is to close the existing gap between the lower bound on the number of queries given by the certificates and the upper bound attained by the currently known algorithms.

Our contributions here are as follows: first, we abstract both the AFP algorithm and its proof into the more abstract problem of learning sublattices; translating the algorithm is an easy task but our correctness proof has to be very different from the original one (and is also much simpler than the proof given in [6]). Second, we show that our proof makes explicit how the AFP algorithm is indeed constructing certificates, and we take advantage of this explanation to prove bounds on certificate size for learning sublattices. Third, we show that the translation of our bounds back into the Horn clause setting gives a bound applicable to the more demanding setting of strong proper learning [16], thus strengthening the result of [5].

2 Preliminaries

We are learning finite lower-sub-semi-lattices. This means that we have a finite lower-semi-lattice $\mathcal{L}$: a partial order relation on a finite carrier set, where the meet (or: greatest lower bound) of two elements is always well-defined. The target to be identified is a subset of $\mathcal{L}$ that is closed under meet: therefore, the subset is itself a lower semi-lattice. However, the connection between lattices and lower semi-lattices is very strong: every lattice is a lower semi-lattice, of course, but the converse direction only has to discuss the presence of a top element. Therefore, for the rest of the paper, we assume, as a minor simplification, that we are working in a lattice, learning a sublattice that includes the top element.

The particular case that guides us is as follows: the carrier lattice is the hypercube $\{0,1\}^n$ with bitwise-comparison as ordering, and bitwise-and as meet. The target, as a set of bit-vectors, is thus a propositional theory; being closed under bitwise-and is equivalent to requiring that it can be axiomatized by a conjunction of Horn clauses. The learning algorithm for this case is given in [3].

For this particular case, the equivalence queries hypothesized by the algorithm are thus Horn formulas, that we assume written in the form of implications: all clauses with the same left hand side α are treated together in a single implication $\alpha \rightarrow \beta$ [1]. Here α and β are terms, and β stands in fact for $\alpha \cup \beta$; note that terms are in bijective correspondence with bit-vectors, and an implication thus

[1] Notice that this differs from an alternative existing interpretation [21] in which $\alpha \rightarrow \beta$ represents the clause $(\overline{x}_1 \vee \ldots \vee \overline{x}_k \vee y_1 \vee \ldots \vee y_{k'})$, where $\alpha = \{x_1, \ldots, x_k\}$ and $\beta = \{y_1, \ldots, y_{k'}\}$. Though identical in syntax, the semantics are different; in particular, ours can only represent a conjunction of definite Horn clauses whereas the other represents a general possibly non-Horn clause.

correspond to a pair of bit-vectors, one above the other. The implication $\alpha \to \beta$ can be seen as a conjunction of clauses with the same antecedent: we propose to call these conjunctions para-clauses. A bit-vector x satisfies the implication, denoted $x \models \alpha \to \beta$, if it either fails the antecedent or satisfies the consequent, that is, $x \not\models \alpha$ or $x \models \beta$ respectively. The target can be seen as a conjunction of implications or, alternatively, as the set of all bit-vectors that satisfy these implications: a propositional theory. We call *cost* of the theory the minimum number of implications needed to axiomatize it. The original Horn learning algorithm is polynomial in the cost and in the number of propositional variables, which bounds how many times one can step down along the boolean hypercube.

In the more general case, we learn sublattices. We describe a sublattice via a *sublattice basis* (abbreviated from now on as *subbasis*): a set of pairs of elements of the lattice, abstracting the notion of implications. Let x, x' be elements of the lattice $\mathcal{L}$, with $x \leq x'$: we say that x and x' are a *comparable pair*, and write always second the larger element. The elements y that *respect* the comparable pair, denoted $y \models (x, x')$, are those for which the implication "if $x \leq y$ then $x' \leq y$" holds, that is, either y is not above x, or is above x'. A subbasis, that is, a set of pairs, then defines the subset of $\mathcal{L}$ consisting of the elements that respect every pair. It turns out that:

Proposition 1. *Given a set of comparable pairs (a subbasis), the set of all elements that respect all of them is closed under meet; and every sublattice has a subbasis, that is, a set of comparable pairs such that the sublattice is exactly the set of elements that respect all of them.*

Proof. To see that if two elements y, y' respect a comparable pair (x, x'), then their meet still respects it, note that if either of y or y' is not above x the meet is not above x either, and that if both are above x, both must be above x' to respect the pair, and so must be their meet, as greatest lower bound. To construct a subbasis for an arbitrary sublattice, take every element that is not in the sublattice and use it to construct a comparable pair, by pairing it to the meet of all the elements above it in the sublattice. □

The *height* of the lattice $\mathcal{L}$ is the length of the longest descending chain. The *cost* of a sublattice of $\mathcal{L}$ is the minimum cardinality of a subbasis.

3 The Algorithm

The lattice $\mathcal{L}$ is assumed known, and a target sublattice $\mathcal{L}^\star$ is to be identified via subbases. We assume throughout that the top of $\mathcal{L}$ belongs to the target, and we denote it $\top$. The learning algorithm is almost exactly the main one in [3], except that it is formulated in lattice-theoretic terms. The correctness proof in the next section is, however, purely lattice-theoretic and is very different from the one in [3] (and is also different from, and much simpler than, the ones in [6], although along similar lines).

For $x \in \mathcal{L}$, we denote $x^\star = \bigwedge\{y \in \mathcal{L}^\star \mid x \leq y\}$; since the target is a sublattice, thus closed under meet, always $x^\star \in \mathcal{L}^\star$, and $x \leq x^\star$, with $x = x^\star$ if and only

if $x \in \mathcal{L}^\star$. It is easy to see that the $\star$ operator is extensive (that is, $x \leq x^\star$), monotonic (if $x \leq y$ then $x^\star \leq y^\star$) and idempotent ($x^{\star\star} = x^\star$); that is, a closure operator.

In the case of propositional theories and Horn formulas, $x^\star$ has true all the variables that can be inferred to be true from the variables true in x, using the target Horn clauses as rules.

Examples for the learning algorithm are elements of $\mathcal{L}$; they are positive or negative according to their membership into the target. Everywhere in this section, the inequality is the comparison according to $\mathcal{L}$'s partial order, and the meet operation is $\mathcal{L}$'s too.

The algorithm maintains a set P of all the positive examples seen so far. The algorithm maintains also a sequence $N = (x_1, \ldots, x_t)$ of negative examples. The argument of an equivalence query is prepared from the list $N = (x_1, \ldots, x_t)$ of negative examples combined with the set P of positive examples. The query corresponds to the following intuitive bias: everything is assumed positive unless some x_i suggests otherwise, and everything that some x_i suggests negative is assumed negative unless some positive example suggests otherwise. This is exactly the intuition in the hypothesis constructed by the AFP algorithm.

More precisely, the equivalence query is represented as a subbasis. For the set of positive examples P, denote $P_x = \{y \in P \mid x \leq y\}$. Observe, for later use, that since $P \subseteq \mathcal{L}^\star$, we have:

$$\bigwedge P_x = \bigwedge\{y \in P \mid x \leq y\} \geq \bigwedge\{y \in \mathcal{L}^\star \mid x \leq y\} = x^\star$$

The hypothesis to be queried, given the set P and the list $N = (x_1, \ldots, x_t)$, will be denoted $H(N, P)$. It is specified by the subbasis constructed as follows: for each x_i in N, add to the subbasis the pair $(x_i, \bigwedge P_{x_i})$. For the initial status of an empty N, this is an empty basis, which is respected by the whole lattice $\mathcal{L}$; that will be the first equivalence query issued.

A positive counterexample is treated just by adding it to P. A negative counterexample y is used to either refine some x_i into a smaller negative example, or to add x_{t+1} to the list. Specifically, let

$$i := \min(\{j \mid (x_j \wedge y) \notin \mathcal{L}^\star \text{and } x_j \wedge y < x_j\} \cup \{t+1\})$$

and then refine x_i into $x_i' = x_i \wedge y$, in case $i \leq t$, or else make $x_{t+1} = y$, subsequently increasing t. The value of i is found through membership queries on all the $x_j \wedge y$.

Putting all together, the algorithm, which is called here AFP-L since it is the **L**attice version of the algorithm by Angluin, Frazier, and Pitt [3], is:

Algorithm AFP-L:
  $N =$ empty list
  $P = \{\top\}$
  $t = 0$
  while EQ$(H(N, P)) =$ ("no", y):
    if $y \not\models H(N, P)$:
      add y to P

else: /* $N = (x_1, \ldots, x_t)$ */
use MQ to find the first i such that:
$x_i \wedge y$ is negative
$x_i \wedge y < x_i$, that is, $x_i \not\leq y$
if found, refine: replace x_i by $x_i \wedge y$ in N
if not found: /* append y to the end of N */
$t = t + 1$
$x_t = y$

4 The Correctness Proof

Like in usual query learning algorithms, in the presence of equivalence queries, the algorithm only stops upon receiving a positive answer to an equivalence query, so that termination implies correctness. We will prove termination of the algorithm by showing that the number t of negative examples x_i in use never exceeds the cost of the target. First, we discuss separately the following easy property:

Lemma 1. *Let P be a set of positive examples, and let x be a negative example. Consider the comparable pair $(x, \bigwedge P_x)$, and assume that $y \models (x, \bigwedge P_x)$ and that $x \leq y$. Then $x^\star \leq y$.*

Proof. Simply notice that, from the definition of $y \models (x, \bigwedge P_x)$, we get that $y \geq x$ implies $y \geq \bigwedge P_x \geq x^\star$. □

4.1 Invariant

We prove that the list N of negative examples maintains always a specific property of existence of certain positive examples. Then we will explain that N, jointly with these positive examples, are precisely certificates for the target lattice, with respect to its cost [16]: this will imply that the size t of the list of negative examples will never exceed the cost (minimal size of a subbasis) of the target.

The main result for the proof is therefore as follows:

Lemma 2. *Along the running of the algorithm in the previous section, at the point of issuing the equivalence query, for every x_i and x_j in N with $i < j$ there exists an element $z \in \mathcal{L}^\star$ such that $x_i \wedge x_j \leq z \leq x_j$.*

Here is where we depart from the proof of the AFP algorithm given in [6], which stated a property similar to this lemma but much weaker, that had to be complemented through complex case analysis and additional facts.

Proof. We need to establish the fact at the time of creating a new element of N, that is, for each $i \leq t$ with respect to x_{t+1}, and we need to argue that the refinements that take place when no such new element is created maintain the

fact stated. Both are argued similarly. If a positive counterexample is received, no element of N changes; thus we only need to consider a negative counterexample y.

First note the easiest case whereby x_i gets refined into $x'_i = x_i \wedge y$. This leaves x_j untouched, and brings down $x_i \wedge x_j$ into $x'_i \wedge x_j$; the same value of z will do: $x'_i \wedge x_j \leq x_i \wedge x_j \leq z \leq x_j$.

Now consider the case in which x_j is refined into $x'_j = x_j \wedge y$. We provide a positive example z' for which $x_i \wedge y \leq z' \leq y$. Then, considering its meet with the already existing z gives

$x_i \wedge x'_j = x_i \wedge x_j \wedge y \leq z \wedge z' \leq x_j \wedge y = x'_j$

Moreover, both z and z' being positive, and the target being closed under meet, ensures that $z \wedge z'$ is positive.

To find z', observe that x_i came before but was not chosen for refinement; either $x_i \wedge y$ is itself positive, and can be chosen for z', or $x_i \leq y$, in which case we use lemma 1: $(x_i, \bigwedge P_{x_i})$ is part of the query, and y was a negative counterexample so it must satisfy the query, including that pair: $y \models (x_i, \bigwedge P_{x_i})$. Lemma 1 tells us $x_i^\star \leq y$, whence $x_i \wedge y \leq x_i \leq x_i^\star \leq y$; and $x_i^\star$ is of course positive.

The case of creation of $x_{t+1} = y$ is handled in the same way: the z' obtained fulfills the condition $x_i \wedge y \leq z' \leq y$ which is what we need. □

4.2 Termination

It remains to show how the lemma just proved guarantees that t is bounded by the dimension of the target. Essentially, the proof says that the elements of N, together with their corresponding values of z considered pairwise as per the lemma, establish a lower bound on the cost of the target sublattice, exactly in the same way as the combinatorial notion of certificates used for nonlearnability proofs [16].

Lemma 3. *Let B be a subbasis of the target. Consider two different negative examples x_i and x_j. Each of them has a comparable pair in B that they do not respect: but it cannot be the same pair.*

Therefore, if the target has cost m, then it has a subbasis with m pairs, and the number of examples x_i will not exceed m.

Proof. Consider a pair $(x, y) \in B$ such that $x_i \not\models (x, y)$ and, likewise, $x_j \not\models (x, y)$, for $i \neq j$. That is, $x \leq x_i$, and $x \leq x_j$, which implies that $x \leq x_i \wedge x_j \leq z$ for the value z given by lemma 2; however, z is a positive example, and must respect the pair (x, y). Then, from $x \leq z$ immediately follows $y \leq z \leq x_j$, as given by the same lemma, and therefore $x_j \models (x, y)$ actually. □

4.3 Final Analysis

It is straightforward to implement in polynomial time the algorithm; the analysis of the number of queries is also easy now: let h be the height of $\mathcal{L}$, and let m

be the cost of the target. Each negative counterexample either increases N, which can happen at most m times, or refines some x_i, which can happen at most h times each, for a total of at most $h \times m$ negative counterexamples. Likewise, each positive counterexample added to P must change at least one of the comparable pairs in the query, by reducing $\bigwedge P_x$, which again can happen at most h times each: we just need to add a constant factor to $h \times m$. Finally, between counterexamples, at most m membership queries are asked.

Therefore, we can state:

Theorem 1. *Let $\mathcal{L}$ be a lattice of height h, and let $\mathcal{L}^\star$ be any sublattice of $\mathcal{L}$ including the top of $\mathcal{L}$, and of cost m. Then the algorithm AFP-L learns $\mathcal{L}^\star$ in polynomial time, with at most $O(h \times m^2)$ queries.*

5 Certificates

The certificate size [16, 5] $q(m, n)$ of a given class of sublattices $\mathcal{L}_n^{\leq m}$ of cost at most m and height n with respect to hypothesis expansions[2] of cost at most $p(m, n)$ is the minimum cardinality of a set of elements such that for any given sublattice f which is not in the class $\mathcal{L}_n^{\leq p(m,n)}$, there is a set Q_f of cardinality at most $q(m, n)$ such that no sublattice in the class $\mathcal{L}_n^{\leq m}$ is consistent with f over Q_f. In other words, f differs with every sublattice in $\mathcal{L}_n^{\leq m}$ on at least one element in Q_f. In a way, Q_f is "rejecting" all the sublattices in $\mathcal{L}_n^{\leq m}$ and "certifying" that f is not in $\mathcal{L}_n^{\leq m}$.

Our construction is very similar to the one for Horn CNFs in [5], but improves on it by constructing certificates for $p(m, n) = m$ thus applying to the more demanding strong proper learning model and generalizing it to the more abstract setting of lattices used in this paper.

5.1 Useful Facts

Assume throughout this section that $(\alpha_1, \beta_1), \ldots, (\alpha_k, \beta_k)$ is a subbasis of minimum cost of a lattice $\mathcal{L}^\star \subseteq \mathcal{L}$. Now, consider the subbasis $(\alpha_1, \beta_1^\star), \ldots, (\alpha_k, \beta_k^\star)$. Clearly, this is a subbasis for $\mathcal{L}^\star$ as well: $\mathcal{L}^\star \models (\alpha_i, \beta_i^\star)$ for each $i = 1, \ldots, k$ and any example x respecting all of $\{(\alpha_i, \beta_i)\}_i$ must respect all $\{(\alpha_i, \beta_i^\star)\}_i$ as well.

Remark 1. Without loss of generality we assume that the subbasis of minimum cost used throughout the proof $\{(\alpha_i, \beta_i)\}_i$ is such that $\beta_i^\star = \beta_i$, that is, $\beta_i \in \mathcal{L}^\star$. By the observations above, such subbasis always exists, represents $\mathcal{L}^\star$ exactly, and is of minimum cost as well. It is also immediate to verify that, in this case, $\beta_i^\star = \beta_i = \alpha_i^\star$.

[2] The fact that hypotheses can have an expansion of cost at most $p(m, n)$ means that algorithms are allowed to present hypotheses of cost at most $p(m, n)$. It is assumed that $m \leq p(m, n)$.

For each $i = 1, \ldots, k$, let $\mathcal{L}_i^\star$ be the sublattice described by the same comparable pairs with the exception of (α_i, β_i). Note that $\mathcal{L}^\star \subseteq \mathcal{L}_i^\star$ and, further, that the inclusion has to be proper since, otherwise, the given subbasis would not be of minimum cost. For each (α_i, β_i), define $x_i \in \mathcal{L}$ as follows:

$$x_i = \bigwedge \{y \in \mathcal{L}_i^\star \mid \alpha_i \leq y\} \tag{1}$$

The following series of lemmas is going to be useful for our main Theorem 2; in several cases the essentials of the proofs are already in the literature.

Lemma 4. *For all i, $x_i \not\models (\alpha_i, \beta_i)$; but $x_i \models (\alpha_j, \beta_j)$ for every $j \neq i$.*

Proof. The proof generalizes from that of Lemma 10 in [5]. Clearly $x_i \geq \alpha_i$. If $x_i \models (\alpha_i, \beta_i)$, then all $y \in \mathcal{L}_i^\star$ with $y \geq \alpha_i$ fulfill $y \geq x_i \geq \beta_i$ so that $\mathcal{L}_i^\star$ satisfies (α_i, β_i), contradicting the strict inequality $\mathcal{L}_i^\star \subset \mathcal{L}^\star$. On the other hand, x_i is the meet of a number of elements in $\mathcal{L}_i^\star$ which is a sublattice, thus closed under meet, and $x_i \in \mathcal{L}_i^\star$ so that $x_i \models (\alpha_j, \beta_j)$ for every $j \neq i$. □

Lemma 5. *For all $i = 1, \ldots, k$ it holds that $x_i^\star = \bigwedge \{y \in \mathcal{L}^\star \mid \alpha_i \leq y\} = \alpha_i^\star$ and $x_i^\star \in \mathcal{L}^\star$.*

Proof. We prove $\alpha_i \leq x_i \leq \alpha^\star$; then, by monotonicity and idempotency, $\alpha^\star \leq x_i^\star \leq \alpha^{\star\star} = \alpha^\star$. The first inequality is from the definition of x_i, whereas the second one holds because $\alpha^\star$ is itself among the y's whose meet is taken to define x_i. Also, $\mathcal{L}^\star$ being closed under meet implies that $x_i^\star$ is positive. □

Lemma 6. *If $x_i \leq x_j$, then $x_i^\star \leq x_j$.*

Proof. Variants of this lemma have been stated in previous work in various guises, e.g. [22]. Suppose that $x_i \leq x_j$. Since $x_j \in \mathcal{L}_j^\star$, it must respect the pair (α_i, β_i). Then, since $\alpha_i \leq x_i \leq x_j$ then $\beta_i \leq x_j$. It only remains to notice that $\beta_i = \beta_i^\star = \alpha_i^\star = x_i^\star$ by Remark 1 and Lemma 5 so that $x_i^\star \leq x_j$. □

Lemma 7. *$x_i \wedge x_j \notin \mathcal{L}^\star$ iff $x_i \leq x_j$ or $x_j \leq x_i$.*

Proof. The direction from right to left is clear: if $x_i \leq x_j$ (or $x_j \leq x_i$), then $x_i \wedge x_j = x_i$ (or $x_i \wedge x_j = x_j$). In either case, we know from Lemma 4 that $x_i \notin \mathcal{L}^\star$ and $x_j \notin \mathcal{L}^\star$.

For the other direction, assume that $x_i \wedge x_j \notin \mathcal{L}^\star$. The comparable pairs (α, β) of $\mathcal{L}^\star$ that are neither (α_i, β_i) nor (α_j, β_j) are satisfied by $x_i \wedge x_j$, since $\mathcal{L}^\star$ is closed under meet and Lemma 4 guarantees that both x_i and x_j respect (α, β). Therefore, it must be that either $x_i \wedge x_j \not\models (\alpha_i, \beta_i)$ or $x_i \wedge x_j \not\models (\alpha_j, \beta_j)$. Assume w.l.o.g. that $x_i \wedge x_j \not\models (\alpha_i, \beta_i)$. This implies that $\alpha_i \leq x_i \wedge x_j$ so that $\alpha_i \leq x_j$. Since x_j respects (α_i, β_i), it must hold that $\beta_i \leq x_j$. Moreover, it always holds that $x_i \leq x_i^\star$. Putting these two together and using the fact that $x_i^\star = \beta_i$ (see Remark 1 and Lemma 5), we get $x_i \leq x_j$ as required. □

5.2 The Certificate Set

Theorem 2. *Sublattices have polynomial size certificates with*

$$q(m,n) = \binom{m+1}{2} + m + 1 \ .$$

Proof. Fix $n, m > 0$. Let $\mathcal{L}^\star$ be a lattice of height n that is not representable by a subbasis of cost m. First, we construct a certificate set Q of cardinality at most $\binom{m+1}{2} + 2m + 2$ for every such $\mathcal{L}^\star$; we will see later that we can remove some elements from Q and apply a finer count to obtain the bound in the statement.

We are given a minimal subbasis $(\alpha_1, \beta_1), \ldots, (\alpha_k, \beta_k)$ where $k \geq m+1$. Let x_i be defined as in Equation (1). Now, define the certificate set $Q = Q^- \cup Q^+ \cup Q^\wedge$, where $Q^- = \{x_i\}_i$,$Q^+ = \{x_i^\star\}_i$, and $Q^\wedge = \{x_i \wedge x_j\}_{i<j}$, where $1 \leq i, j \leq k$.We prove that Q is indeed a certificate set.

Take any sublattice $\mathcal{L}'$ of cost at most m. By way of contradiction, assume that $\mathcal{L}'$ is consistent with $\mathcal{L}^\star$ over all the elements in Q. Lemmas 4 and 5 guarantee that $x_i \not\models \mathcal{L}'$ but $x_i^\star \models \mathcal{L}'$ for all $1 \leq i \leq m+1$. There must be one particular comparable pair (α, β) in $\mathcal{L}'$ violated by both x_i and x_j for some i, j s.t. $1 \leq i < j \leq m+1$. Clearly $x_i \wedge x_j \not\models (\alpha, \beta)$; therefore $x_i \wedge x_j \not\models \mathcal{L}'$. Furthermore, $\alpha \leq x_i \leq x_i^\star$ and thus $\beta \leq x_i^\star$, otherwise we would have that $x_i^\star \not\models (\alpha, \beta)$.

If x_i and x_j are incomparable, then Lemma 7 guarantees that $x_i \wedge x_j \models \mathcal{L}^\star$, thus $\mathcal{L}'$ and $\mathcal{L}^\star$ differ on $x_i \wedge x_j \in Q^\wedge$. If $x_i \leq x_j$, then by Lemma 6 and our observation above we obtain that $\beta \leq x_i^\star \leq x_j$ and thus x_j respects (α, β), which contradicts $x_j \not\models (\alpha, \beta)$. The case where $x_j \leq x_i$ is analogous.

We conclude that Q is a certificate set of cardinality at most $q(m,n) = \binom{m+1}{2} + 2m + 2$. Notice however that the proof above only uses $x_i^\star$ when there is a x_j s.t. $x_i \leq x_j$. Moreover, in this case, it is clear that $x_i \wedge x_j = x_i$ so we do not need to count it in $Q^\wedge$. Therefore it is sufficient to include in Q^+ only those $x_i^\star$ such that $x_i \leq x_j$ for some $i < j \leq k$. The resulting set Q is also a certificate and has the desired cardinality, which concludes our proof. □

To apply this theorem in the Horn formulas, we need to discuss also the case of a set of propositional models that does not allow for Horn axiomatization: but, in such a case, it is not closed under meet; then there must exist 3 examples $x, y, x \wedge y$ such that $x, y \in \mathcal{L}^\star$ but $x \wedge y \notin \mathcal{L}^\star$. Therefore the certificate set Q can be constructed as $Q = \{x, y, x \wedge y\}$ and $|Q| \leq 3$. Clearly, no lattice, closed under meet, can be consistent over these three examples.

The certificates of [5] for Horn CNF use $p(m,n) = m(n+1)$. In contrast, here we only require that $p(m,n) = m$ thus applying to the more stringent model of strong proper learning [16]. On the other hand, in [5] m counts the number of clauses whereas here m is the number of comparable pairs, which translates to using *para-clauses* or implications in the Horn logic setting. Notice that the size of a Horn expression in terms of para-clauses is at most its size in clauses. The authors feel that the right thing to count in the case of Horn expressions is the number of para-clauses, especially in the context of analyzing the AFP algorithm that learns using the para-clause representation.

Notice also that this upper bound matches *exactly* the lower bound from [5] in the case that $m < n$. If $m > n$, there is still a gap between the lower bound of $\Omega(mn)$ in [5] and our upper bound of $O(m^2)$.

6 Certificates and the Learning Algorithm

We are going to illustrate the link between our certificate construction and the learning algorithm with an example where the lattice to be learned is a Horn formula $\mathcal{H}^\star$. The example is taken from [13], where a canonical representation for definite Horn theories is introduced. The propositional variables are going to be a, b, c, d, e, f (and so the height of the lattice is 6). Suppose that our target concept is

$$\mathcal{H}^\star = \{e \to d, bc \to d, bd \to c, cd \to b, ad \to bce, ce \to ab\}.$$

First let us compute the examples $x_1, \ldots, x_6$ from Lemma 4, and their closure $x_1^\star, \ldots, x_6^\star$.

i	$\alpha_i \to \beta_i$	$\alpha_i \to \beta_i^\star$	x_i	$x_i^\star$
1	$e \to d$	$e \to ed$	00001	00011
2	$bc \to d$	$bc \to bcd$	01100	01110
3	$bd \to c$	$bd \to bcd$	01010	01110
4	$cd \to b$	$cd \to bcd$	00110	01110
5	$ad \to bce$	$ad \to abcde$	10010	11111
6	$ce \to ab$	$ce \to abcde$	01111	11111

We can compute x_i and $x_i^\star$ by setting to one the bits corresponding to the antecedent of para-clause 1 and then doing forward chaining exhaustively with all the other clauses to obtain x_i, and with all the clauses to obtain $x_i^\star$. It is worth noting that for $x_1, .., x_5$ the forward chaining is not needed since no antecedents of the other clauses are satisfied. However, when computing x_6 we start with the assignment 00101 but need to set to 1 the bits bd since the clauses 1 and 4 are triggered in the deduction process.

Notice also that the comparable pairs corresponding to a minimal subbasis of this concept are $(x_i, x_i^\star)$ for $i = 1, \ldots, 6$.

In our example simulation, we are going to feed

$$x_1, x_1^\star, x_2, x_2^\star, x_3, x_4, x_5, x_6$$

as counterexamples. From the simulation it will become clear why we do not need to present some of the positive $x_i^\star$.

Let $t = 11111$ be the top assignment. We assume that we are dealing with definite clauses, so t is always positive. The following table captures the internal state of the algorithm and the dynamics of the simulation:

	N	P	$EQ(H(N,P))$	MQs issued
1	$()$	$\{t\}$	$x_1 = 00001$	
2	(x_1)	$\{t\}$	$x_1^\star = 00011$	
3	(x_1)	$\{t, x_1^\star\}$	$x_2 = 01100$	$x_1 \wedge x_2 = 00000, +$
4	(x_1, x_2)	$\{t, x_1^\star\}$	$x_2^\star = 01110$	
5	(x_1, x_2)	$\{t, x_1^\star, x_2^\star\}$	$x_3 = 01010$	$x_1 \wedge x_3 = 00000, +$
				$x_2 \wedge x_3 = 01000, +$
6	(x_1, x_2, x_3)	$\{t, x_1^\star, x_2^\star\}$	$x_4 = 00110$	$x_1 \wedge x_4 = 00000, +$
				$x_2 \wedge x_4 = 00100, +$
				$x_3 \wedge x_4 = 00010, +$
7	(x_1, x_2, x_3, x_4)	$\{t, x_1^\star, x_2^\star\}$	$x_5 = 10010$	$x_1 \wedge x_5 = 00000, +$
				$x_2 \wedge x_5 = 00000, +$
				$x_3 \wedge x_5 = 00010, +$
				$x_4 \wedge x_5 = 00010, +$
8	$(x_1, x_2, x_3, x_4, x_5)$	$\{t, x_1^\star, x_2^\star\}$	$x_6 = 01111$	$x_5 \wedge x_6 = 00010, +$
9	$(x_1, x_2, x_3, x_4, x_5, x_6)$	$\{t, x_1^\star, x_2^\star\}$	Yes	

The table is to be read as follows: first $P = \{11111\}$ and N is empty. The first counterexample received by the algorithm is $x_1 = 00001$, it is negative and so it is added to N. Clearly x_1 was a negative counterexample since initially the hypothesis corresponds to the concept that classifies every example as positive. The next counterexample is $x_1^\star = 00011$, which is indeed a positive counterexample since $H((00001), \{11111\})$ classifies this example as positive. After adding this positive counterexample to P, the algorithm has discovered exactly the first paraclause and will include in its hypothesis the correct comparable pair $(x_1, x_1^\star)$. The next counterexample is x_2 which is indeed a counterexample since it satisfies the comparable pair (Lemma 4) but falsifies the target concept. Since $x_1 \not\leq x_2$, the algorithm needs to check that their intersection $x_1 \wedge x_2 = 00000$ is not negative, which indeed is not. So x_2 is appended to N.

It is worth noticing that all membership queries are responded with an affirmative answer because we only ask when $x_i \wedge x_j \not\leq x_i$, for $i < j$ (Lemma 7), so negative counterexamples are always appended to N. For example, when receiving x_6 as counterexample, we do not need to check the intersections $x_i \wedge x_6$ for $i = 1, \ldots, 4$ because we have that $x_i \leq x_6$; only $x_5 \wedge x_6$ is checked. Also, we do not need to present positive examples $x_3^\star, x_4^\star, x_5^\star, x_6^\star$ because they are already included in P.

In general, one can always do such a simulation with any hidden target sublattice $\mathcal{L}^\star$. Assume that $\mathcal{L}^\star$ is represented by a minimal subbasis of cost k and $x_1, \ldots, x_k$ are the examples satisfying Lemma 4. Without loss of generality, assume that the examples follow a particular order: x_i appears before x_j if $x_i \leq x_j$. Notice that this set of preferences induce a DAG among the x_i (no cycles) and therefore such an ordering is always possible. Notice that $\{(x_i, x_i^\star) \mid 1 \leq i \leq l\}$ is also a minimal basis for $\mathcal{L}^\star$.

The counterexamples are going to be the a subsequence of:

$$x_1, x_1^\star, x_2, x_2^\star, \ldots, x_k, x_k^\star.$$

The positives $x_i^\star$ are presented only if they are not already in P, namely, if $x_i^\star$ is not the *top* element and $x_{i'}^\star \neq x_i^\star$ for all $i' < i$.

So, after presenting counterexamples x_l and $x_l^\star$ (if not in P already), we are guaranteed that $N = (x_1, \ldots, x_l)$ and therefore all comparable pairs $(x_i, x_i^\star)$ up to l are exactly discovered. When the last pair of negative and positive counterexamples is issued (if the positive one is needed), the equivalence query will be done with the exact same hypothesis as the target concept, so that the simulation will terminate in success.

Clearly, the elements in N and P are exactly those of Q^- and Q^+ of our certificate construction, and the membership queries represent the positive ones in $Q^\wedge$.

Notice also that the order in which we try to refine the elements of N is irrelevant, since with this particular sequence of examples we never refine any existing counterexample in N. In general, this is obviously not the case, and keeping the elements in N in order is important. More particularly, the order is important for pairs of elements $n_{i'}, n_{j'}$ in N that are violating pairs of $\mathcal{L}^\star$ with corresponding x_i and x_j s.t. $x_i \leq n_{j'}$.

Certificates and the Invariant of Section 4.1. Lemma 2 states the following: for every x_i and x_j in N with $i < j$ there exists an element $z \in \mathcal{L}^\star$ such that $x_i \wedge x_j \leq z \leq x_j$. Let us analyze what the z witnessing the Lemma are. For those pairs that satisfy $x_i \leq x_j$, the witnessing z is $x_i^\star$. Lemma 6 guarantees that $x_i \wedge x_j = x_i \leq x_i^\star \leq x_j$. If x_i and x_j are incomparable, then by Lemma 7 $x_i \wedge x_j$ is going to be positive and thus z is $x_i \wedge x_j$ itself, and trivially we get: $x_i \wedge x_j \leq x_i \wedge x_j \leq x_j$. Notice that, again, the x_i, $x_i^\star$, and $x_i \wedge x_j$ used in Lemma 2 constitute precisely our certificates.

Acknowledgements. For many related discussions, in person or remote, the authors are particularly indebted to Jaume Baixeries, Montserrat Hermo, and Josefina Sierra.

References

[1] Angluin, D.: Learning Regular Sets from Queries and Counterexamples. Information and Computation 75, 87–106 (1987)
[2] Angluin, D.: Negative Results for Equivalence Queries. Machine Learning 5, 121–150 (1990)
[3] Angluin, D., Frazier, M., Pitt, L.: Learning Conjunctions of Horn Clauses. Machine Learning 9, 147–164 (1992)
[4] Angluin, D.: Queries revisited. Theor. Comput. Sci. 313, 175–194 (2004)
[5] Arias, M., Feigelson, A., Khardon, R., Servedio, R.: Polynomial Certificates for Propositional Classes. Information and Computation 204, 816–834 (2006)
[6] Balcázar, J.L.: Query learning of Horn formulas revisited. In: Computability in Europe Conference, Amsterdam (2005)
[7] Balcázar, J.L., Castro, J., Guijarro, D.: A New Abstract Combinatorial Dimension for Exact Learning via Queries. J. Comput. Syst. Sci. 64, 2–21 (2002)

[8] Balcázar, J.L., Castro, J., Guijarro, D., Köbler, J., Lindner, W.: A general dimension for query learning. J. Comput. Syst. Sci. 73, 924–940 (2007)
[9] Balcázar, J.L., Castro, J., Guijarro, D., Simon, H.-U.: The consistency dimension and distribution-dependent learning from queries. Theor. Comput. Sci. 288, 197–215 (2002)
[10] Bshouty, N.H., Cleve, R., Gavaldà, R., Kannan, S., Tamon, C.: Oracles and Queries That Are Sufficient for Exact Learning. J. Comput. Syst. Sci. 52, 421–433 (1996)
[11] Frazier, M., Pitt, L.: Learning From Entailment: An Application to Propositional Horn Sentences. In: Int. Conference Machine Learning 1993, pp. 120–127 (1993)
[12] Frazier, M., Pitt, L.: CLASSIC Learning. Machine Learning 25, 151–193 (1996)
[13] Guiges, J.L., Duqenne, V.: Familles minimales d'implications informatives resultants d'un tableau de donnees binaires. Math. Sci. Hum. 95, 5–18 (1986)
[14] Gaintzarain, J., Hermo, M., Navarro, M.: On Learning Conjunctions of Horn$^{\supset}$ clauses. In: Computability in Europe Conference, Amsterdam (2005)
[15] Hegedűs, T.: On generalized teaching dimensions and the query complexity of learning. In: Proceedings of the Conference on Computational Learning Theory, pp. 108–117. ACM Press, New York (1995)
[16] Hellerstein, L., Pillaipakkamnatt, K., Raghavan, V., Wilkins, D.: How Many Queries Are Needed to Learn? Journal of the ACM 43, 840–862 (1996)
[17] Hermo, M., Lavín, V.: Negative Results on Learning Dependencies with Queries. In: Proceedings on Seventh International Symposium on Artificial Intelligence and Mathematics (2002)
[18] Kivinen, J., Mannila, H.: Approximate Inference of Functional Dependencies from Relations. Theoretical Computer Science 149, 129–149 (1995)
[19] Selman, B., Kautz, H.: Knowledge Compilation and Theory Approximation. Journal of the ACM 43, 193–224 (1996)
[20] Valiant, L.: A Theory of the Learnable. Communications of the ACM 27, 1134–1142 (1984)
[21] Wang, H.: Toward mechanical mathematics. IBM Journal for Research and Development 4, 2–22 (1960); In: Wang, H.: A survey of mathematical logic, Peking and Amsterdam. Science Press and North Holland, pp. 224-268 (1963)
[22] Wild, M.: A theory of finite closure spaces based on implications. Advances in Mathematics 108, 118–139 (1994)