

Some Lower Bounds for the Computational Complexity of Inductive Logic Programming

Jörg-Uwe Kietz

German National Research Center for Computer Science, I3-KI,
P.O.Box 1316, D-5205 St.Augustin, Germany

Abstract. The field of Inductive Logic Programming (ILP), which is concerned with the induction of Hornclauses from examples and background knowledge, has received increased attention over the last time. Recently, some positive results concerning the learnability of restricted logic programs have been published. In this paper we review these restrictions and prove some lower-bounds of the computational complexity of learning. In particular, we show that a learning algorithm for i -determinate Hornclauses (with variable i) could be used to decide the PSPACE-complete problem of Finite State Automata Intersection, and that a learning algorithm for 12-nondeterminate Hornclauses could be used to decide the NP-complete problem of Boolean Clause Satisfiability (SAT). This also shows, that these Hornclauses are not PAC-learnable, unless $RP = NP = PSPACE$.

Keywords: Inductive Logic Programming, PAC-Learning.

1 Introduction

Most success within the field of Machine Learning has been achieved with systems learning in a propositional logic. Also the theory of learnability, e.g. PAC-learnability, was mostly concerned with propositional logic. But, despite their successes, propositional learning approaches suffer from the limited expressiveness of their hypothesis languages and the lack of background knowledge. Therefore, the field of Inductive Logic Programming (ILP), which is concerned with the induction of first-order Hornclauses from examples and background knowledge, has received increased attention recently [13, 2, 6, 7, 8, 9, 12, 14]. The problem tackled by these approaches can be described formally as follows.

Definition 1 Learning Problem.

Given:

- background knowledge B in a language LB
- positive and negative examples $E = E^+ \cup E^-$ in a language LE consistent with B ($B, E \not\models \square$) and not a consequence of B ($B \not\models E$).
- a hypothesis language LH .

Find a hypothesis $h \in LH$ such that:

- (1) ($B, h, E \not\models \square$), h is consistent with B and E .

- (II) $(B, h \models E^+)$, h and B explain E^+ .
 (III) $(B, h \not\models E^-)$, h and B do not explain E^- .

The triple (LB, LE, LH) is called the *learning problem*. Deciding whether there exists such an $h \in LH$, is called the *consistency problem*. An algorithm which accepts any $B \in LB$ and $E \in LE$ as input and computes such an $h \in LH$ if it exists, or "no" if it does not exist is called a *learning algorithm* for (LB, LE, LH) . If LB , LE and LH are subsets of first-order clauses, (LB, LE, LH) is called the ILP-problem and a learning algorithm for (LB, LE, LH) is called an ILP-algorithm.

As the ILP problem has been proven to be undecidable in the general case, where LE , LB and LH are first-order clauses [10], current research focuses on the identification of effective subclasses of first-order logic that are efficiently learnable. Recently, some positive results on the PAC-learnability of so called constrained Hornclauses [9] and a superset of them called *ij-determinate* Hornclauses (with fixed i and j) [3] have been obtained.

In this paper, we show that these hypothesis languages cannot be further extended without the loss of the PAC-learnability property. To prove non-PAC-learnability, we will use an adaptation of a theorem of Pitt and Valiant, also proven as theorem 6.2.1 in [1].

Corollary 2. *If a learning problem (LB, LE, LH) is PAC-learnable, then the consistency problem for (LB, LE, LH) is in RP, or turning it the other way around, if the consistency problem is not in RP, then (LB, LE, LH) is not PAC-learnable.*

In the following we will prove that the ILP problem for determinate Hornclauses is PSPACE-hard, and therefore *i2-determinate* Hornclauses are not PAC-learnable in general (where i is variable), unless $RP = PSPACE$. We will also show that the consistency problem for *12-nondeterminate* Hornclauses is NP-hard, and therefore *12-nondeterminate* Hornclauses are not PAC-learnable, unless $RP = NP$. But, first let us review the proposed restrictions of the ILP problem more formally.

2 Common Restrictions of the ILP Problem

The background knowledge used in ILP programs must always be restricted, otherwise the ILP problem inherits the undecidability of the deduction process. A common restriction in ILP programs (e.g.[7, 12]) is the restriction to ground background knowledge and ground unit clauses as examples.

Definition 3 Ground background knowledge. The background knowledge B is ground if it consists of ground unit clauses only. A clause is ground if it does not contain any variables.

One way to avoid such a restriction is the use of generative Hornclauses and a depth-bounded inference process to generate such finite ground background knowledge prior to learning (e.g. [6],[7]).

Definition 4 Generative Hornclause. A Hornclause is generative if every variable in the head also occurs in the body of the clause.

Another common restriction applied not only to B , but also to LH is the restriction to function-free clauses (e.g. [6, 12]).

Definition 5 Function-free clause. A clause is called function-free if all its arguments are either variables, or constants (function symbols of arity 0).

The restriction to function-free ground unit clauses as examples and function-free ground background knowledge enables us to use θ -subsumption as a complete inference procedure in the following way: The learning problem (ground B , ground unit E, LH) as defined in definition 1, is equivalent to the learning problem $(\{\}, E_{new}, LH)$, where E_{new} is defined as $E_{new}^+ := \{e \leftarrow B \mid e \in E^+\}$, and $E_{new}^- := \{e \leftarrow B \mid e \in E^-\}$. Clearly, the examples are now ground Hornclauses. Between function-free Hornclauses θ -subsumption is a correct and complete inference procedure ($h \models h'$, iff $h\theta \subseteq h'$). This means the theorems 10 and 16 which use ground function-free Hornclauses as examples and θ -subsumption as inference relation also hold for ground unit clause examples and ground background knowledge together with implication as inference.

Function-free knowledge has another positive effect on deduction. It has been proven that inferring ground background knowledge can be done completely in time polynomial to the size of B if B consists only of function-free generative Hornclauses ([15]) and there is a fixed maximum arity of predicates in the background knowledge.

Definition 6 Bounded-arity background knowledge.

The language for background knowledge LB is of bounded arity if there exist an integer j , which is greater than the maximum arity of any of the predicates in B .

This bounded-arity restriction is also used by Page and Frisch [9] to prove the PAC-learnability of a special kind of hypothesis language LH called *constrained clauses*.

Definition 7 Constrained clause. A clause is constrained if all variables in the body also occur in the head.

Muggleton and Feng [7], have proposed a restriction of LH to ij -determinate Hornclauses.

Definition 8 Determinate Clauses and determinate depth of terms.

A Hornclause h is determinate with respect to the examples E and the background knowledge B if every term t in h is linked by a determinate linking-chain. A term occurring in the head is linked by a determinate linking-chain of

length 0. Let $h = \{A, \neg B_1, \dots, \neg B_m, \neg B_{m+1}, \dots, \neg B_n\}$ be ordered. The term t found in $\neg B_{m+1}$ is linked by a determinate linking-chain of length $d + 1$, if and only if all terms in $\{A, \neg B_1, \dots, \neg B_m\}$ are linked by a determinate linking-chain of length at most d and for every substitution θ such that $A\theta \in E^+$ and $\{\{B_1\}, \dots, \{B_m\}\}\theta \subseteq B$ there is a unique substitution δ whose domain is the variables in t such that $\{B_{m+1}\}\theta\delta \in B$. The determinate depth of a term is the minimum length of its determinate linking chains.

A clause with maximum determinate depth of terms i , and maximum arity of literals j , is called ij -determinate.

Muggleton and Feng have used this ij -determinate restriction to prove that the ILP-problem (ground background knowledge, ground unit clause examples, one ij -determinate Hornclause) is solvable in polynomial time [7].

Recently, it has been proven [3] that a k -disjunction of function-free non-recursive ij -determinate Hornclauses is PAC-learnable under simple distributions. This was proven by showing that this ILP-learning problem can be reduced to an equivalent only polynomially larger propositional learning problem, i.e., learning function-free ij -determinate Hornclauses from ground background knowledge and ground examples is no more powerful than learning in propositional logic. The remaining advantage is that this kind of representation is more compact and therefore potentially more user-friendly in the preparation of the input.

So far, there is no answer to the question whether the restriction to ij -determinate Hornclauses can be relaxed without the loss of polynomial computability of the ILP-problem. This paper gives a negative answer to this question by proving that the consistency problem for non depth-bounded determinate Hornclauses is PSPACE-hard and that the consistency problem for depth bounded non-determinate Hornclauses is NP-hard.

Definition 9 Linked Hornclause and depth of terms.

A Hornclause is linked if all its literals are linked. A literal is linked if at least one of its terms is linked. A term is linked with linking-chain of length 0 if it occurs in the head of the clause. A term in a literal is linked with a linking-chain of length $d + 1$, if another term in the same literal is linked with a linking-chain of length d . The depth of a term is the minimum length of its linking-chains.

A not determinate clause with maximum depth of terms i , and maximum arity of literals j , is called ij -nondeterminate.

Clearly, terms in nondeterminate clauses need not have a determinate linking-chain. If a term has a *determinate linking-chain*, this is also a (nondeterminate) *linking-chain*, but not vice versa. Therefore, the *determinate depth* of a term is always greater than or equal to the (nondeterminate) *depth* of a term.

3 Non-Learnability of Determinate Hornclauses

Theorem 10. *The consistency-problem for non depth bounded 2-determinate Hornclauses is PSPACE-Hard.*

We will prove this theorem, by reducing the following PSPACE-complete problem [4] to the consistency-problem (cf. Def. 1).

Definition 11 Finite State Automata Intersection. Given an alphabet Σ and a sequence $A_1 \dots, A_n$ of deterministic finite state automata with input alphabet Σ , does there exist a word $w \in \Sigma^*$ accepted by each of the $A_i, 1 \leq i \leq n$?

First let us recall some basics of deterministic finite state automata (DFA). A DFA A is formally described as a 5-tuple $(Q, \Sigma, \delta, s, F)$, where Q is the set of states, Σ is the input alphabet, $s \in Q$ is the initial state, $F \subseteq Q$ is the set of final states, and δ is the transition function mapping $Q \times \Sigma$ to Q . The transition function δ is extended to words on Σ^* as follows: $\hat{\delta}(q, \epsilon) = q$, $\hat{\delta}(q, w\bar{a}) = \delta(\hat{\delta}(q, w), a)$ where ϵ is the empty word, $a \in \Sigma, w \in \Sigma^*$. A DFA is said to accept a word $x \in \Sigma^*$, iff $\hat{\delta}(s, x) \in F$.

Now, we are able to give a useful encoding of DFAs and words as 2-determinate Hornclauses.

Definition 12 Encoding of DFA's and words as Hornclauses. Let Γ be a function from $\Sigma^* \cup (Q, \Sigma, \delta, s, F)$ to Hornclauses defined as follows:
 $\Gamma_{word}(x_1 x_2 \dots x_n) = h(Q_0) \leftarrow x_1(Q_0, Q_1), x_2(Q_1, Q_2), \dots, x_n(Q_{n-1}, Q_n), f(Q_n)$
 $\Gamma_{DFA}(A) = \{h(s)\} \cup \{\neg x(q, p) | x \in \Sigma, p, q \in Q, \delta(q, x) = p\} \cup \{\neg f(q) | q \in F\}$

As is easily seen, DFAs are encoded as ground 2-determinate Hornclauses, i.e. they are suitable as examples. Words are encoded as non-ground 2-determinate Hornclauses, i.e. they are suitable as hypotheses. The usefulness of this encoding is shown by the following lemma.

Lemma 13. *A word is recognized by a DFA if and only if the encoding of the word θ -subsumes the encoding of the DFA.*

Proof. $\hat{\delta}(s, x) \in F \Rightarrow \Gamma(x) \leq_{\theta} \Gamma(A)$

Let $x = x_1 \dots x_m$, then there exist a sequence of states (the computation of A on x) q_0, \dots, q_m , such that $q_0 = s, q_m \in F$, and $\delta(q_{i-1}, x_i) = q_i$. Let θ be $\{Q_i/q_i\}$, for $0 \leq i \leq m$, from the definition of Γ it follows that $\Gamma(x)\theta \subseteq \Gamma(A)$.

$\Gamma(x) \leq_{\theta} \Gamma(A) \Rightarrow \hat{\delta}(s, x) \in F$

Let θ be $\{Q_i/q_i\}$, for $0 \leq i \leq m$, from the definition of Γ it follows that $q_0 = s, q_m \in F$, and $\delta(q_{i-1}, x_i) = q_i$, i.e. the sequence $q_0 \dots, q_m$ is an accepting computation of x by A . \square

Now we are able to prove that Γ is indeed an encoding of the DFA intersection problem in terms of the learning problem of determinate Hornclauses.

Lemma 14. *Let $A_1 \dots, A_n$ be a sequence of deterministic finite state automata. There exists a nonempty word $x \in \Sigma^*$ accepted by each of the $A_1 \dots, A_n$ if and only if there exists a linked determinate Hornclause consistent with the following set of positive and negative examples: $E^+ = \{\Gamma(A_i) | 1 \leq i \leq n\}$
 $E^- = \{h(s) \leftarrow f(s), x_1(s, q), \dots, x_m(s, q), x_1(q, q), \dots, x_m(q, q),$
 $h(s) \leftarrow x_1(s, s), \dots, x_m(s, s)\}$, where $\Sigma = \{x_1, \dots, x_m\}$*

Proof. “the if direction”

Let h be a linked determinate Hornclause consistent with the examples. As h is a generalization of all positive examples each of its literals are either of the form $h(X)$, $\neg f(X)$, or $\neg x(S1, S2)$, where $x \in \Sigma$. As h is a Hornclause $h(Q_0)$ is in h . As h is consistent with the second negative example it contains a literal $f(Q_n)$, and as h is consistent with the first negative example, the variable $f(Q_n)$ is only unifiable with the head variable Q_0 if this unification also unifies a second argument of an $\neg x(S1, S2)$ literal with the head variable. As h is linked the literal $f(Q_n)$ must be linked by a chain of $\neg x(S1, S2)$ literals. The Hornclause C_w consisting only of the head literal $h(Q_0)$, the final literal $f(Q_n)$, and the linking chain of $\neg x(S1, S2)$ literals is a generalization of h and therefore a generalization of all the positive examples. Using the argument above it is not a generalization of the negative examples. C_w is by construction the Γ encoding of a word. By Lemma 13 this word is accepted by each of the automata $A_1 \dots, A_n$.

“the only if direction”

Let x be the non empty word accepted by all the A_1, \dots, A_n . By Lemma 13 $\Gamma(x)$ is a generalization of E^+ . $\Gamma(x)$ is not a generalization of the first negative example, as x is not empty. $\Gamma(x)$ is not a generalization of the second negative example as it contains a literal $\neg f(Q)$. Therefore $\Gamma(x)$ is a consistent generalization of the examples. Clearly, $\Gamma(x)$ is linked and determinate. \square

Corollary 15 Non PAC-learnability of determinate Hornclauses. *Using Corollary 2 and Theorem 10 we can conclude that determinate linked Hornclauses are not PAC-learnable, as long as the widely assumed $RP \neq PSPACE$ conjecture is true.*

4 Non Learnability of nondeterminate Hornclauses

Theorem 16. *Consistency of 12-nondeterminate Hornclauses is NP-hard.*

We will prove this theorem, by reducing the well known NP-complete SAT problem [4] to the consistency-problem as stated above. The reduction is inspired by a similar proof of Haussler [5] for *existentially quantified conjunctive expressions*.

Definition 17 SAT. Given a set $V = \{v_1, \dots, v_n\}$ of Boolean variables and a set of clauses $C = \{C_1, \dots, C_m\}$ over V , the question is whether there exists a truth assignment of V that satisfies the clauses in C .

Now, let us prove that a SAT instance is satisfiable iff a special learning problem is solvable. The idea is to construct positive examples, such that every generalisation consists of truth assignments to Boolean variables of the SAT problem and to construct the negative examples so that only hypotheses whose truth assignments satisfy the SAT problem are consistent with them.

Lemma 18. *The SAT instance (V, C) is satisfiable if and only if there exists a Hornclause consistent with $E = E^+ \cup E^-$ defined as follows:*

Define a set of $2n$ 1-place predicates a_1, \dots, a_{2n} , each of which are mapped via Ψ to one of the $2n$ literals $v_1, \dots, v_n, \bar{v}_1, \dots, \bar{v}_n$. The positive examples are then coded as following: $E^+ = \{p(X) \leftarrow p(X, X_1), a_1(X_1), \dots, a_{i-1}(X_1), a_{i+1}(X_1), \dots, a_{2n}(X_1), p(X, X_2), a_1(X_2), \dots, a_{n+i-1}(X_2), a_{n+i+1}(X_2), \dots, a_{2n}(X_2)\}$, for all $i : 1 \leq i \leq n$

The negative example is coded as follows. $E^- = \{\{p(X)\} \cup \{\neg p(X, X_j), \neg a_i(X_j) \mid \Psi(a_i) \notin C_j, 1 \leq i \leq 2n, 1 \leq j \leq m\}\}$

Proof. “the if direction”

Let h be a Hornclause that is consistent with the positive and negative examples. As h is a generalization of all positive examples, it is of the form

$$p(X) \leftarrow p(X, Y_k), a_1(Y_k), \dots, a_{2n}(Y_k), k : 1 \leq k \leq 2^n$$

and for all $i : 1 \leq i \leq n$, at most one of $a_i(Y_k)$ or $a_{n+i}(Y_k)$ is in h . As h does not θ -subsume the negative example, there exists at least one k such that

$$h' = p(X) \leftarrow p(X, Y_k), a_1(Y_k), \dots, a_{2n}(Y_k) h' \subseteq h$$

and there is no substitution $\theta = X/X, Y_k/X_j$ such that

$$h'\theta \subseteq \{\{p(X)\} \cup \{\neg p(X, X_j), \neg a_i(X_j) \mid \Psi(a_i) \notin C_j, 1 \leq i \leq 2n\}\}$$

This means for every $C_j \in C$, h' contains at least one literal $a_i(Y_k)$, such that $\Psi(a_i) \in C_j$. This means $\{\Psi(a_i) \mid a_i(Y_k) \in h'\}$ is a partial truth assignment for V that satisfies all clauses in C .

“the only if direction”

Assume, that (V, C) is satisfiable. Let L be the set of literals true in an assignment that satisfies (V, C) such that either v_i or $\neg v_i$ is in L , but not both. Let h be a Horn clause defined by

$$h = \{\{p(X)\} \cup \{\neg p(X, Y), \neg a_i(Y) \mid \Psi(a_i) \in L, 1 \leq i \leq 2n\}\}$$

h θ -subsumes all positive examples, as for every $i : 1 \leq i \leq n$ it contains either $a_i(X_1)$ or $a_{n+i}(X_1)$, but not both. We now prove by contradiction that h does not θ -subsume the negative example. If h θ -subsumes the negative example, there is an $\theta = \{X/X, Y/X_j\}$, such that

$$h\theta \subseteq \{\{p(X)\} \cup \{\neg p(X, X_j), \neg a_i(X_j) \mid \Psi(a_i) \notin C_j, 1 \leq i \leq 2n\}\}$$

But, this means that h contains only a_i , such that $\Psi(a_i)$ is not in C_j , which contradicts the assumption that L satisfies all clauses of C . \square

Corollary 19. *Using Corollary 2 and Theorem 16 we can conclude that 12-nondeterminate linked Hornclauses are not PAC-learnable, as long as the widely assumed $RP \neq NP$ conjecture is true.*

5 Conclusion

In this paper, we have proven some lower bounds of the computational complexity of learning within the ILP framework. These results show that large classes of Hornclauses cannot be learned by a complete and efficient learning algorithm. This includes e.g. CLINT's [2] languages greater equal L2, and the hypothesis space searched by the inverse resolution approaches e.g. [8, 14]. It also shows that fast heuristic search algorithms like FOIL [11] are necessarily incomplete, i.e they may not find a correct definition, even if one exists. Theorem 10 also shows that the extension of FOIL2 [12], which adds determinate literals if no good information gain can be achieved by a literal, could lead to an explosion of the size of the hypothesis exponential in the number of times this happens.

Acknowledgements: I would like to thank Katharina Morik and Bernhard Nebel for the discussions and suggestions they have provided to this work. I would also like to thank the DFKI for the research visit, they have supported. This work is partially supported by the European Community ESPRIT program under contract number P6020 "Inductive Logic Programming".

References

1. M. Anthony and N. Biggs. *Computational Learning Theory*. Cambridge University Press, 1992.
2. L. de Raedt and M. Bruynooghe. An overview of the interactive concept-learner and theory revisor clint. In S. Muggleton, editor, *Inductive Logic Programming*. Academic Press, 1992.
3. S. Džeroski, S. Muggleton, and S. Russell. Pac-learnability of determinate logic programs. In *Proc. of the 5th ACM Workshop on Computational Learning Theory (COLT)*, 1992.
4. Michael R. Garey and David S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. Freeman, San Francisco, Cal., 1979.
5. David Haussler. Learning conjunctive concepts in structural domains. *Machine Learning*, 4(1):7-40, 1989.
6. Jörg-Uwe Kietz and Stefan Wrobel. Controlling the complexity of learning through syntactic and task-oriented models. In S. Muggleton, editor, *Inductive Logic Programming*, pages 107-126. Academic Press, 1992.
7. Stephan Muggleton and Cao Feng. Efficient induction of logic programs. In S. Muggleton, editor, *Inductive Logic Programming*. Academic Press, 1992.
8. Stephen Muggleton and Wray Buntine. Machine invention of first-order predicates by inverting resolution. In *Proc. Fifth Intern. Conf. on Machine Learning*, Los Altos, CA, 1988. Morgan Kaufman.
9. C. D. Page and A. M. Frisch. Generalisation and learnability: a study in constrained atoms. In S. Muggleton, editor, *Inductive Logic Programming*. Academic Press, 1992.
10. Gordon D. Plotkin. A further note on inductive generalization. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 6, chapter 8, pages 101-124. American Elsevier, 1971.

11. J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5(3):239 – 266, 1990.
12. J. R. Quinlan. Determinate literals in inductive logic programming. In *Proc. of the 12th IJCAI*, 1991.
13. Celine Rouveirol. Completeness for inductive procedures. In *Proc. Eighth Intern. Workshop on Machine Learning*, pages 452–456, 1991.
14. Celine Rouveirol. Semantic model for induction of first order theories. In *Proc. 12th International Joint Conference on Artificial Intelligence*, 1991.
15. Stefan Wrobel. Higher-order concepts in a tractable knowledge representation. In K. Morik, editor, *GWAI-87 11th German Workshop on Artificial Intelligence*, Informatik-Fachberichte Nr. 152, pages 129 – 138, Berlin, New York, Tokyo, October 1987. Springer.