Learning Acyclic First-Order Horn Sentences from Entailment *

Hiroki Arimura

Department of Informatics
Kyushu University
Kasuga, 816 Japan
Email: arim@i.kyushu-u.ac.jp

Abstract. This paper considers the problem of learning an unknown first-order Horn sentence H_* from examples of Horn clauses that H_* either implies or does not imply. Particularly, we deal with a subclass of first-order Horn sentences $ACH(k)$, called *acyclic constrained Horn programs of constant arity k*. $ACH(k)$ allows recursions, disjunctive definitions, and the use of function symbols. We present an algorithm that exactly identifies every target Horn program H_* in $ACH(k)$ in polynomial time in p, m and n using $O(pmn^{k+1})$ entailment equivalence queries and $O(pm^2n^{2k+1})$ request for hint queries, where p is the number of predicates, m is the number of clauses contained in H_* and n is the size of the longest counterexample. This algorithm combines *saturation* and *least general generalization* operators to invert resolution steps. Next, using the technique of replacing request for hint queries with entailment membership queries, we have a polynomial time learning algorithm using entailment equivalence and entailment membership queries for a subclass of $ACH(k)$. Finally, we show that any algorithm which learns $ACH(k)$ using entailment equivalence and entailment membership queries makes $\Omega(mn^k)$ queries, and that the use of entailment cannot be eliminated to learn $ACH(k)$ even with both equivalence and membership queries for ground atoms are allowed.

1 Introduction

This paper considers the problem of learning an unknown Horn sentence H_* from examples of Horn clauses that H_* either implies or does not imply. This type of learning framework is called *learning from entailment* and has been introduced in the studies of learning propositional Horn sentences [2, 11]; Frazier and Pitt [11] showed that propositional Horn sentences are polynomial time learnable using equivalence and membership queries in the sense of entailment. The notion of

* A part of this research was done while the author was staying at Department of Computer Science, University of Helsinki in the summer of 1996. This research is partly supported from the Ministry of Education, Science and Culture, Japan by Grant-in-Aid for Scientific Research on Priority Area "Advanced Databases", and from Academy of Finland.

learning from entailment has been successfully applied to first-order logic and has demonstrated that various interesting fragments of first-order logic are shown to be efficiently learnable from entailment, such as constrained atoms with a background theory [19] and description logic CLASSIC [12, 10].

In this paper, we consider the learnability of a subclass of first-order Horn sentences $ACH(k)$, called *acyclic constrained Horn programs of constant arity* k. A Horn program is a conjunction H of implications $a_1, \ldots, a_n \rightarrow a$, where $a_1, \ldots, a_n, a$ are atoms. A Horn program is *constrained* if all clauses in the program satisfy the constraint that all terms occurring in the body $a_1, \ldots, a_n$ are subterms of some arguments of the head a. Particularly, we restrict programs to be *acyclic*, that is, every computation for entailment will finitely terminate.

The class $ACH(k)$ is a simple but interesting fragment of first-order Horn sentences. $ACH(k)$ allows most basic computational mechanisms of logic programming such as recursions, disjunctive definitions, and the use of complex objects with function symbols, while related computational problems, such as subsumption and entailment, are still efficiently solvable for every fixed k. This is a contrast to that recent studies of inductive logic programming mainly deal with single nonrecursive function-free clauses.

Page and Frisch [19] showed that single nonrecursive clauses in $ACH(k)$ are polynomial time learnable from examples in the sense of entailment. However, recent studies of inductive logic programming showed that most generalizations of single nonrecursive clauses are hard to learn from examples alone [8, 10]. For the class of Horn programs consisting of arbitrary number of clauses, it is an open question whether there exists a polynomial time learning algorithm even if both equivalence and membership queries are available.

In this paper, we consider *active learning*, where a learning algorithm can actively make entailment membership queries rather than only passively receive entailed examples. A *request for hint query* is an augmented entailment membership query introduced by Angluin [2], which tells whether a given Horn clause is subsumed, implied, or not implied by the unknown Horn sentence. In Section 3, we present a polynomial time learning algorithm for $ACH(k)$ using entailment equivalence queries and request for hint queries. Our algorithm $LEARN\_ACH$ exactly identifies every target Horn program H_* in $ACH(k)$ in polynomial time in p, m and n using $O(pmn^{k+1})$ entailment equivalence queries and $O(pm^2n^{2k+1})$ request for hint queries, where p is the number of predicates, m is the number of clauses contained in H_* and n is the size of the longest counterexample.

Next in Section 4, we present that request for hint queries can be replaced by entailment membership queries for a proper subclass of $ACH(k)$, *uniformly acyclic* constrained Horn programs. Using this method, we give a polynomial time learning algorithm for learning this class using entailment equivalence and entailment membership queries. Since this algorithm is presented in the paradigm of exact learning, we can transform it into either a *polynomial time PAC-learning algorithm using membership queries*, or a *polynomial time on-line learning algorithm using membership queries* in terms of entailment semantics.