Function-Free Horn Clauses
Are Hard to Approximate

Richard Nock and Pascal Jappy

Laboratoire d’Informatique, de Robotique et de Microélectronique de Montpellier,
161, rue Ada,
34392 Montpellier, France
{nock,jappy}@lirmm.fr

Abstract. In this paper, we show two hardness results for approximat-
ing the best function-free Horn clause by an element of the same class. Our
first result shows that for some constant $k > 0$, the error rate of the
best $k$-Horn clause cannot be approximated in polynomial time to within
any constant factor by an element of the same class. Our second result is
much stronger. Under some frequently encountered complexity hypoth-
esis, we show that if we replace the constant number of Horn clauses
by a small, poly-logarithmic number, the constant factor blows up expo-
ponentially to a quasi-polynomial factor $n^{\log^k n}$, where $n$ is the number of
predicates of the problem, a measure of its complexity. Our main result
links the difficulty of error approximation with the number of clauses al-
lowed. We finally give an outline of the incidence of our result on systems
that learn using ILP (Inductive Logic Programming) formalism.

1 Introduction and motivation

ILP is an active research branch at the crossroads of of Machine Learning and
Logics. It aims at learning concepts expressed as (variously) restricted Horn
Clause Programs from examples, and in the presence of background knowledge.
Many experimental applications are available, that have been applied to domains
such as biology, chess playing and natural language analysis. Theoretical work has
allowed to establish learnability results for some subclasses of first order Horn
clauses. Early studies were undertaken in the Identification in the limit model
[7], but most work has focused on Approximately Correct (PAC) learnability
[15], [10] which is thought to better quantify the complexity of learning in terms
of computational effort and number of examples required. In ILP, this latter
problem is intractable for very general classes such as unconstrained Horn clauses
(see [11] for a detailed presentation of computational hardness results). So, in
order to achieve positive results, several restrictions of Horn Clause programs
have been considered [13], [4], [5], and [6].

However, conflicts between PAC results and practical ones have led researchers
to look for other learnability models [12]. In a previous paper, we highlighted
divergences between PAC and robust learning [8] results for some of the main
ILP classes. Whereas PAC learning makes the strong assumption that any target concept can be represented in the hypothesis class \( \mathcal{H} \), (which is very rarely acceptable in practice), robust learning studies the degradation in prediction performance of a hypothesis class \( \mathcal{H} \) when it is not known a priori whether it contains the target concept's class. This makes this model a stricter one but it is closer to practical requirements. The commonpoint to both PAC and robust learning models is the sufficiency of worst case analyses to obtain negative results. Our result in [9] states that, even when considering a simple subclass of ILP formalism and even when looking for a single Horn clause, no polynomial-time algorithm can produce a formula whose error comes close to the error of the optimal single Horn clause. In this paper, we go further in worst-case analyses. We show that the condition on the error can be replaced by a much weaker one without losing negative results. We show that no polynomial-time algorithm can produce a formula approximating the error of the optimal one to within very large factors. The rest of this paper is organised as follows: in section 2, we present the ILP background we need for our results, and the link between ILP and structural complexity. In section 3 and 4 we prove that approximating function-free Horn clause is hard. Finally, in section 5, we highlight some relevant subclasses of ILP formalism for which our results are valid.

2 An ILP approximation problem

For a complete formalization of the ILP background needed for this article, we refer the reader to [9]. Given a Horn clause language \( \mathcal{L} \) and a correct inference relation on \( \mathcal{L} \), an ILP learning problem can be formalized as follows. Assume a background knowledge \( \mathcal{B} \) expressed in a language \( \mathcal{L}_B \subseteq \mathcal{L} \), and a set of examples \( \mathcal{E} \) in a language \( \mathcal{L}_E \subseteq \mathcal{L} \). The goal is to produce a hypothesis \( h \) in a hypothesis class \( \mathcal{H} \subseteq \mathcal{L} \) consistent with \( \mathcal{B} \) and \( \mathcal{E} \) such that \( h \) and the background knowledge cover all positive examples and none of the negative ones. The choice of the representation languages for the background knowledge and the examples, and the inference relation greatly influence the complexity (or decidability) of the learning problem. A common restriction for both \( \mathcal{B} \) and \( \mathcal{E} \) is to use ground facts. As in [11], we use \( \theta \)-subsumption as inference relation. Its main drawback being that it does not allow the use of background knowledge, other subsumption relations have been defined to do so, in particular generalized subsumption [2], and are thus preferred in ILP. We now state a useful lemma

**Lemma 1** Learning a Horn clause program from a set of ground background knowledge \( \mathcal{B} \) and ground examples \( \mathcal{E} \), the inference relation being generalized subsumption, is equivalent to learning the same program with \( \theta \)-subsumption, and empty background knowledge and examples defined as ground Horn clauses of the form \( e \leftarrow b \), where \( e \in \mathcal{E} \) and \( b \in \mathcal{B} \).

This lemma allows us to incorporate the background knowledge in the new examples (and is thus empty). Examples and clauses are defined by predicates. To the variables of these predicates that are in the clauses built correspond constant