Negative Robust Learning results for Horn Clause Programs

Pascal Jappy
LIRMM, 161 rue ADA,
34392 Montpellier
CEDEX 5, FRANCE.
jappylirmm.fr

Richard Nock
LIRMM, 161 rue ADA,
34392 Montpellier
CEDEX 5, FRANCE.
nocklirmm.fr

Olivier Gascuel
LIRMM, 161 rue ADA,
34392 Montpellier
CEDEX 5, FRANCE.
gascuellirmm.fr

Abstract

We study the learnability of Inductive Logic Programming (ILP) concept classes with respect to robust-learning. We first investigate the class of k-Horn clauses, and show that it is not learnable in that model. We prove this using a reduction on which we impose as few constraints as possible. From this proof, we then show how we can also derive negative results for some PAC-learnable classes. Finally, we end by discussing the applicational consequences of our work and its links with other learnability studies regarding new learnability models for ILP.

1 INTRODUCTION AND MOTIVATION

Inductive Logic Programming (ILP) is a branch of Machine Learning which aims at learning concepts, expressed as (variously) restricted Horn Clause Programs, from examples and in the presence of background knowledge. In recent years, ILP has produced both experimental applications and theoretical learnability results. Among the former are systems such as MIS (Schapiro, 1983), FOIL (Quinlan, 1990), LINUS (Lavrac et al., 1991), GOLEM (Muggleton and Feng, 1992), CLINT (Raedt and Bruynooghe, 1992), ITOU (Rouveirol, 1992), FORCE2 (Cohen, 1993b), and others, which have been applied to domains such as biology, chess playing and natural language analysis. A common feature of those systems is their use of search space size reductions, achieved by restricting the expressivity of the concept classes they learn. For instance, determinacy is frequently used to ensure tractability. Algorithmically, these programs follow two different approaches to hypothesis production. MIS and CLINT, for instance, identify the target at the limit, whereas most others use polynomial heuristics for concept induction. Consequently, these systems are generally efficient learners, but, to our knowledge, none can be formally shown to find the target concept in polynomial time.

Simultaneously, 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 (Gold, 1967), which describes learning as converging towards the target concept, in finite time but given an unbounded amount of examples. Schapiro (Schapiro, 1983) identified à most general class learnable in this model by a consistent algorithm (MIS) and other studies have since been carried out in this framework (Banerji, 1987), (Raedt, 1992). But most work focuses on Probably Approximately Correct (PAC) learnability (Valiant, 1984), (Kearns et al., 1987) which is thought to better quantify the complexity of learning in terms of computational effort and number of examples required. PAC learning relaxes the convergence requirement and aims at obtaining a hypothesis which is a good approximation of this target, from a reasonable computation and from a polynomial number of positive and negative examples drawn according to some fixed but unknown probability distribution. In ILP, this is intractable for very general classes such as unconstrained Horn clauses (see (Kietz and Dzeroski, 1994) for a detailed presentation of computational hardness results). So, in order to achieve positive results, several restrictions of Horn Clause programs have been considered. Again, determinacy has played a central role, and the class of single nonrecursive ij-determinate clauses was shown to be PAC learnable in (Muggleton and Feng, 1992). Several studies have since considered extensions of this basic class, thereby leading to both positive and negative results, first by relaxing some of its syntactical constraints (Cohen, 1993b), (Cohen, 1994), (Cohen, 1995a) and (Cohen, 1995b), then by allowing multiple clauses and recursivity. Dzeroski, Muggleton and Russell (Dzerovski et al., 1992) have shown that k (constant) non recursive constant depth determinate clauses are learnable under simple distributions, and extended this result to otherwise similar recursive clauses of constant maximum arity provided membership queries (Angluin et al., 1992) are allowed. Cohen (Cohen, 1993a) then showed that the class of 2 ij-determinate linear and closed recursive clauses are PAC learnable with special basecase queries.

In this paper, we have chosen a different model to examine the practical learnability of ILP concept classes. Indeed, 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. So, we consider the (more realistic in that respect) model of ***robust*** learnability studied in (Hoffgen and Simon, 1992). Robust learning abandons the above assumption and 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, the situation in which it doesn't being described as overstraining $\mathcal{H}$. Since the target concept may not be expressible in the hypothesis class, robust learning contrasts with PAC by comparing the performance of a proposed hypothesis not with a precision parameter ϵ, but with that of a theoretically 'best' hypothesis in $\mathcal{H}$ plus ϵ. It should be underestood that making the above PAC assumption simplifies the learning task. Consequently, suppressing it makes robust learning a stricter model than PAC, and any robustly learnable (henceforth R-learnable) class (e.g. symmetric functions) will obviously also be PAC-learnable. However, it also makes for a model which is closer to applicative requirements, and is particularly relevant for the highly restricted hypothesis classes often considered. So, our goal here is to highlight divergences with PAC results for some of the main ILP classes. In order to prove these, we first give an intermediate theorem for the class of k function-free Horn clauses, which contains most of the ILP classes studied in the PAC model, and which we show not to allow robust learning. To our knowledge, the PAC equivalent of this problem is open. We then show how, from the very general reduction used to prove the theorem, we can derive properties for the main PAC learnable ILP classes such as determinate or local clauses.

The rest of this paper is organised as follows: in section 2, we first make the ILP problem we deal with clear, then present the robust learning model in which we set our study. In section 3, we prove the non learnability of k-Horn clauses in that model. This intermediate result allows us to derive the corollaries of section 4, in which we expose its consequences on well known ILP classes. Finally, given the apparent negative nature of our study, we discuss its applicational consequences, in the concluding remarks of section 5.

2 FORMAL PRELIMINARIES

In section 2.1, we precisely define the ILP setting in which we place ourselves. In particular, we present the inference procedure we implicitely use, the choice of which directly influences our proofs later on. Then, in section 2.2, we present the robust learnability model which we consider and highlight its close relations with the PAC framework.

2.1 Learning Horn Clauses

Throughout this paper, we follow classical First Order Logic (FOL) notations. We assume the reader is familiar with these, and refer him to (Lloyd, 1992) for background definitions.

Given a Horn clause language $\mathcal{L}$, and a correct inference relation on $\mathcal{L}$, an ILP learning problem can be formalised as follows. Assume a background knowledge $\mathcal{BK}$ expressed in a language $\mathcal{LB} \subseteq \mathcal{L}$ and a set of examples $\mathcal{E}$ in a language $\mathcal{LE} \subseteq \mathcal{L}$. The goal is to produce a hypothesis h in a hypothesis class $\mathcal{H} \subseteq \mathcal{L}$ consistent with $\mathcal{BK}$ and $\mathcal{E}$ such that h and the background knowledge cover all positive examples and none of the negative ones. The choice of representation languages for the background knowledge and the examples, and of inference relation greatly influence the complexity (or decidability) of the learning problem. A common restriction for both $\mathcal{BK}$ and $\mathcal{E}$ is to use ground facts. And in order to ensure tractability, subsumption is preferred to implication as correct inference procedure. As Kietz and Dzeroski (Kietz and Dzeroski, 1994), we use θ-subsumption as inference relation. Its main drawback being that it doesn't allow the use of background knowledge, other subsumption relations have been defined to do so, in particular generalised subsumption (Buntine, 1988), and are thus preferred in ILP. The following lemma establishes a usefull link between the two in our context.

Lemma 1 ((Kietz and Dzeroski, 1994))
Learning a Horn clause program from a set of ground background knowledge facts $\mathcal{BK}$ and ground example

facts $\mathcal{E}$, the inference relation being generalised subsumption, is equivalent to learning the same program with θ-subsumption, an empty background knowledge and examples defined by $e \leftarrow b$, where $e \in \mathcal{E}$ and $b \in \mathcal{BK}$.

In the following, we are interested in learning concepts in the form of k Horn clauses from ground background knowledge and examples. This lemma allows us to consider the background knowledge has been incorporated in the new examples (and is thus empty). Hence, all our proofs make use of θ-subsumption and an empty background knowledge. We will let k-HORN denote the class containing sets of k function free non recursive Horn clauses. An element of k-HORN θ-subsumes an example $e \in \mathcal{E}$ if one of its k clauses θ-subsumes e.

2.2 Robust Learning

We now define the learnability model in which we set our study. The basic definitions are taken from (Hoffgen and Simon, 1992). Let $\mathcal{X}$ be a finite description set on which is defined a probability distribution D, and let $\mathcal{C} \subseteq 2^{\mathcal{X}}$ and $\mathcal{H} \subseteq 2^{\mathcal{X}}$ respectively denote a concept class and a hypothesis class. We let n denote the size of the largest example in $\mathcal{X}$. In the Boolean framework, the size of any example is always the number of description variables, but ILP examples are ground facts of variable size, so the size of the largest is taken as complexity parameter. A hypothesis $h \in \mathcal{H}$ is said to be ϵ-accurate for the target concept $c \in \mathcal{C}$ if its prediction error relative to c is bounded by ϵ, that is $P_D(h \neq c) \leq \epsilon$. Another important notion, particularly relevant if we assume $\mathcal{H} \subset \mathcal{C}$, is ϵ-optimality, which describes closeness to the best possible hypothesis. Let $c \in \mathcal{C}$ denote a concept, $h_{opt}(c)$ an optimal hypothesis for c in $\mathcal{H}$, that is $P_D(h_{opt}(c) \neq c) = inf_{h \in \mathcal{H}} P_D(h \neq c) = opt_{\mathcal{H},D}(c)$, and $n_{opt_{\mathcal{H},D}(c)}$ the size of the smallest such optimal hypothesis (when $c \in \mathcal{H}$, that is the PAC assumption, $h_{opt}(c)$ has a nil error).

Definition 1 (ϵ-optimality) *A hypothesis $h \in \mathcal{H}$ is ϵ-optimal if $P_D(h \neq c) \leq opt_{\mathcal{H},D}(c) + \epsilon$*

We now define robust learnability, a close variant of PAC-learnability used when no assumption is made that the hypothesis space contains the concept being learnt. We adapt Hoffgen and Simon's (Hoffgen and Simon, 1992) definition to ILP requirements by introducing a size notion. Note that our extension is consistent with the original and similar in nature to that of uniform PAC-learnability and predictability defined by Cohen in (Cohen, 1993b).

Definition 2 (Robust Learnability) *A hypothesis class $\mathcal{H}$ is said to allow robust learning if there exists a learning algorithm L and a polynomial $p(.,.,.,.)$, which for any concept c in $2^{\mathcal{X}}$ accuracy and confidence parameters ϵ and δ can, using a sample of $m \leq p(1/\epsilon, 1/\delta, n, n_{opt_{\mathcal{H},D}(c)})$ examples (supplied by an oracle, according to any unknown but fixed distribution D), outputs an ϵ-optimal hypothesis h with probability at least $1 - \delta$ and in time polynomial in m.*

Note that various learnability notions are linked according to the mutual inclusion relationship between $\mathcal{C}$ and $\mathcal{H}$. In particular, robust learning is the special case of *agnostic learning* (Kearns et al., 1992), (Auer et al., 1995) which occurs when $\mathcal{C} = 2^{\mathcal{X}}$. When $\mathcal{C} \subseteq \mathcal{H}$, ϵ-accuracy and ϵ-optimality become equivalent. Therefore, robust learnability becomes synonymous to PAC-learnability since the aim of PAC-learning is to produce an ϵ-accurate hypothesis with probability at least $1 - \delta$. In the special case $\mathcal{C} = \mathcal{H}$, $\mathcal{C}$ is said to be PAC-learnable by itself.

Lemma 2 *Robust learnability is a stricter model than PAC.*

Proof Assume $\mathcal{H}$ is not PAC-learnable by itself. This means the target concept c (which is also the optimum concept) cannot be approached within the PAC requirements. Since these are the same for robust learning, at least some concepts (c, for instance) in $2^{\mathcal{X}}$ exist which cannot be approximated by a hypotesis in $\mathcal{H}$. Therefore, if $\mathcal{H}$ is not PAC-learnable it is not R-learnable either.

3 A NEGATIVE RESULT FOR k FUNCTION-FREE NON RECURSIVE HORN CLAUSES

In this section, we show that the general class of k-function free Horn clauses does not allow proper agnostic learning. This class contains most of the more restricted ones for which PAC learning results have already been obtained. As stated earlier, this intermediate result is trivial for the non PAC learnable ones. However, the reduction used in our proof allows us to derive negative robust learning properties for some of the others. It also solves a problem which, to our knowledge is open in the PAC framework, that is, are k function free Horn clauses learnable or not ?

Theorem 1 *If $RP \neq NP$, for any integer constant $k > 0$, k-HORN does not allow robust learning.*

Proof: Hoffgen and Simon (Hoffgen and Simon, 1992) use an intermediate complexity problem associated to the hypothesis class, which they show to be NP-Hard then show that this implies negative robust learnability properties for the class. We perform both steps at once. This allows us to give a self contained proof, without having to present their simulation technique, even though we use a very similar argument. Our proof is based on a property shown by Arora ((Arora, 1994), pp 102-106), and which we now restate:
(P1.1): Fix some constant $k' > 0$. For any set of clauses instance of 3-SAT (Garey and Johnson, 1979), we can construct in polynomial time a graph $G = (V, E)$ instance of Clique, associated to some integer m (also computable in polynomial time) such that:

- If the instance if 3-SAT is satisfiable, then the clique number of G (denoted $\omega(G)$) is equal to m.
- If the instance of 3-SAT is not satisfiable, then $\omega(G) = m/k$.

With the help of (P1.1), we show that if, for some constant k, k-Horn clauses allow robust learning (name L this algorithm), then we can construct an RP-algorithm to solve 3-SAT, a contradiction if RP $\neq$ NP. The idea is to run L with appropriate examples and parameters, generated from any instance of Clique. From any graph G, define a set of $|V|$ unary predicates $a_1(.), ..., a_{|V|}(.)$, in one to one correspondence with the set V of vertices of G. The sets of examples are then defined as follows:
$S^+ = \{p_i = q(l_i) \leftarrow \wedge_{k\in\{1,...,|V|\}-\{i\}} a_k(l_i), \forall i \leq |V|\}$
and
$S^- = \{n_{ij} = q(l_{ij}) \leftarrow \wedge_{k\in\{1,...,|V|\}-\{i,j\}} a_k(l_{ij}), \forall (i,j) \notin E\}$
where $\{l_i, 1 \leq i \leq |V|\}$ and $\{l_{ij}, (i,j) \notin E\}$ are constant symbols. S^- describes the edge structure of the complementary graph of G. The respective cardinalities are:

$$|S^+| = |V| \text{ and } |S^-| = \binom{|V|}{2} - |E|,$$

where $\binom{n}{k}$ is the binomial coefficient. We fix the example weights to 1 for every positive example and to $|V| + 1$ for every negative one. The probability distribution over the examples used when L is run is generated from these weights in an obvious manner: each example has an associated probability equal to its weight divided by the total sum of the weights $\Sigma = |S^+| + (|V| + 1)|S^-|$. The other learning parameters are:

$$\epsilon = 1/(\Sigma + 1) \text{ and } \delta < 1.$$

The target concept is supposed to be S^+. Note that the parameters $1/\epsilon, 1/\delta, n, n_{opt_{\mathcal{H},D}(c)}$ are all polynomial in the size of G (the latter because the optimal hypothesis needs only one variable and its size is at most $k * |V|$, that is k clauses each containing all the litterals found in the examples). Therefore, when L is run, the learning task is performed in time polynomial in the size of G, and therefore in the size of the instance of 3-SAT. Assume that the reduction of (P1.1) is performed by fixing $k' = k + 1$. The following property proves the theorem by reaching a contradiction if RP $\neq$ NP:
(P1.2) if we answer positively to 3-SAT whenever the formula h obtained from L makes at most $(|V| - m)/\Sigma$ errors on the examples, we obtain an RP algorithm which solves 3-SAT.
To prove (P1.2), we have to prove the three following intermediate results:
(P1.3) If the error of some set of k-Horn clause is strictly lower than $|V| + 1/\Sigma$, then any of the Horn clauses it contains corresponds to a clique in G, containing the vertices corresponding to literals absent from the body of the corresponding clause. Indeed, any negative example corresponds to an edge absent from G, and because of the hypothesis, all of them are well classified. If the set of vertices corresponding to the absent literals did not give rise to a clique, a negative example corresponding to the missing edge would be θ-subsumed by the clause, which is impossible.
(P1.4) The error of any set of k-Horn clause is at least $(|V| - k\omega(G))/\Sigma$. Indeed, if a hypothesis misclassifies some negative example, the property is true. If not, because of (P1.3), to any clause corresponds a clique in G. The number of postive examples θ-subsumed by some clause is exactly the size of the corresponding clique in G: indeed, in order to θ-subsume some positive example, the literal absent in the positive example must be absent from the body of the clause. And conversely, if some literal is absent from the body of the clause, then the positive example that does not contain this literal is θ-subsumed by the clause. Therefore, there are at most $k\omega(G)$ correctly classified positive examples.
(P1.5) There always exists some set of k-Horn clause whose error is $(|V| - \omega(G))/\Sigma$. Indeed, note that a single Horn clause whose body contains exactly the literals corresponding to the vertices absent from a max-clique makes mistakes only on positive examples, and its error is equal to $(|V| - \omega(G))/\Sigma$.
We can now prove (P1.2):

- Case 1: Suppose that the instance of 3-SAT is satisfiable. Then $\omega(G) = m$. Because of (P1.5), there exists some element of k-HORN whose error is no more than $(|V| - m)/\Sigma$. In that case, with probability greater than $1-\delta$, L returns a hypothesis h whose error is no greater than $opt_{\mathcal{H},D}(c) + \epsilon$, *i.e.* no greater than $opt_{\mathcal{H},D}(c)$ (because of the our choice of ϵ), and so smaller than $(|V| - m)/\Sigma$. And in that case we answer positively with a probability greater than $1 - \delta$.

- Case 2: Suppose that the instance of 3-SAT is not satisfiable. Then $\omega(G) = m/(k+1)$. (P1.4) shows that the error of any element of k-HORN clauses is at least $(|V| - mk/(k+1))/\Sigma > (|V| - m)/\Sigma$. In that case, we never answer positively.

Cases 1 and 2 establish that, under the hypothesis that k-HORN allows robust learning, we can construct some RP-algorithm to solve 3-SAT, which is a contradiction if RP $\neq$ NP.□

4 CONSEQUENCES ON PAC-LEARNABLE ILP CLASSES

The proof of theorem 2 imposes very little syntactic/size constraints on the clauses and the examples. In this section, we show that this allows us to derive consequences for some well studied ILP classes, since the previous result is preserved by the usual restrictions found in the literature, and thus covers a large majority of ILP problems that are known to be PAC-learnable. We now review the main restrictions on function free Horn clauses which have been adopted by various authors in order to achieve PAC-learnability, and briefly show that the proof given above applies individually to each of them. Obviously, since robust learning is stricter than PAC-learning, any non PAC-learnable class will not be robustly learnable. So, we concentrate on classes for which positive PAC results exist.

4.1 Determinate clauses

Determinacy implies that given a ground substitution for all the variables in the head of a clause, those for the variables in the body are uniquely determined by a step by step (literal by literal) process.

Definition 3 (Determinate clause) *A Horn clause h is determinate (with respect to the background knowledge and the examples) if every term t in h is determinate.*

A term t in the head of a clause is determinate (that is, linked by a determinate linking chain of length 0). Assume $h = A \leftarrow B_1, ..., B_m, B_{m+1}, ..., B_n$. The term t in the literal B_{m+1} is linked by a determinate linking-chain of length $i + 1$ iff all the terms in B_{m+1} that appear in $A \leftarrow B_1, ..., B_m$ are linked by determinate linking-chains of length at most i and for every substitution θ such that $A\theta \in \mathcal{E}$ and $\mathcal{BK} \vdash \{B_1, ..., B_m\}$, there is a unique substitution σ, on the variables in t such that $\mathcal{BK} \vdash B_{m+1}\theta\sigma$, i.e. if every variable in t which does not appear in preceding terms has only one possible binding, given the bindings of the variables of the previous terms.

Definition 4 *The determinate depth of a term is the minimal lengths of its determinate linking-chains (covering all the terms). The (nondeterminate) depth of a clause is the maximum depth of any of its variables. The depth of a variable is 0 if it appears in the head of a clause and $d + 1$ if it appears in a literal in the body of the clause alongside another variable of depth d.*

A function-free clause of maximum arity j and depth i is said to be ij-determinate. This important restriction has been used by numerous authors to achieve positive results. Muggleton and Feng (Muggleton and Feng, 1992) used ij-determinacy to show the PAC-learnability of a single non recursive ij-determinate clause. Dzeroski *et al.* (Dzerovski et al., 1992) have shown that k non recursive clauses of determinate depth i are learnable under simple distributions for any positive integers i and k (they extended this result to otherwise similar recursive clauses of constant maximum literal arity provided membership queries (Angluin et al., 1992) are allowed). (Cohen, 1993a) has shown that the class of 2 ij-determinate linear and closed recursive clauses are PAC learnable with Basecase queries.

Theorem 2 *If RP $\neq$ NP, for any integers $i \geq 0$ and $j, k > 0$ the class ij-determinate non recusrive k-HORN is not R-learnable.*

Proof The proof is extremely simple and relies entirely on the generality of the proof of theorem 1. In the construction of the examples sets, only literals of arity one are used. If the clauses produced as hypotheses contained predicates of arity greater than one, they would subsume none of the examples. Hence, the optimum hypothesis can be found in the subset of k-HORN containing only literals of arity one (*i.e.* 0,1 determinate hypotheses). Also, since the head predicate of the clauses never appears in the body of our examples, the clauses are non recursive. In other words, the proof of

Theorem 1 would have been the same if, instead of considering k-HORN, we had limited the hypotheses to 01-determinate k-HORN. If the optimum could be found in ij-determinate k-HORN, we could also find it for 01-determinate k-HORN. The converse proves the result. □

Several authors have analysed the relaxation of the ij-determinacy restriction with respect to learnability. However, Cohen (Cohen, 1994) and Kietz and Dzeroski (Kietz and Dzeroski, 1994) have shown that using either non constant (even logarithmically so) depth, or nondeterminate clauses leads to negative results. The following subsection discusses an alternative restriction which leads to a more general yet PAC-learnable language.

4.2 Local clauses

Locality is a completely semantic restriction which attempts to quantify how much of a clause is influenced by the binding of a given variable. It is presented by the author as an alternative (to determinacy) restriction which allows a greater number of practical problems to be tackled.

Definition 5 (Local clause) *Let $h = A \leftarrow B_1, ..., B_n$ be a clause. A variable V in h is said to be free if it appears in the body of h but not in its head A. A free variable V_1 is said to touch another V_2 if both appear in the same literal, and to influence V_2 if it either touches it or touches some other free variables which influence it. In other words, V_1 influences V_2 if V_1's chosen binding affects the choices for V_2. The locale of a free variable is then defined as the set of literals which contain it or another variable influenced by it. And the locality of a clause h is defined to be the size (i.e. number of literals) of the largest locale of all the variables in h, or 0 if h contains no free variables.*

Cohen (Cohen, 1994) compares the expressivity of the class of nonrecursive clauses of constant locality with that of non recursive clauses of constant depth.

Theorem 3 (from (Cohen, 1994)) *For every ij-determinate clause C, there exists a semantically equivalent clause C' of locality $k = a^{d+1}$ and of size no greater than k times that of C.*

This class is also shown to be PAC-learnable in (Cohen, 1994). We now show that it isn't R-learnable.

Theorem 4 *If $RP \neq NP$, for any positive integers $l, k > 0$ the class l-local k-HORN is not R-learnable.*

Proof As for Theorem 2 note that the predicates used in the examples of the main proof are unary. So, an optimum can be found in clauses of arity one . Also, note that all the examples contain the same constant. So, in order to θ-subsume examples, a clause needs only one variable. Therefore, among optimal hypotheses of arity one, there necessarily exists one using only one variable. Since, as is usually the case we are only interested in linked clauses (Kietz and Dzeroski, 1994), this proves that 0-local k-HORN always contains the optimum hypothesis. Again, the proof of Theorem 1 would have been the same if, instead of considering k-HORN, we had limited the hypotheses to 0-local k-HORN. If the optimum could be found in l-local k-HORN, we could also find it for 0-local k-HORN. The converse proves the result. □

Remark We have shown that ij-determinate k-HORN and l-local k-HORN are not R-learnable. Given that the proofs of Theorems 2 and 4 simply boil down to showing that that of Theorem 1 would hold unchanged for these classes, it is quite obvious that the result also holds for their intersection. These results imply that, used in a polynomial-time learning system, those classes (which are all PAC learnable) could not tolerate overstraining.

5 CONCLUDING REMARKS

In this work we have studied the learnability of some common ILP concept classes from a viewpoint not considered before in the domain: instead of the more usual PAC learning model, we have focused on robust learnability, an extension of PAC which relaxes a frequently unacceptable constraint on the hypothesis class. In order to obtain general properties, we have first studied the class of k-Horn clauses, later deriving results for its PAC-learnable subsets. It turns out that none of the classes is learnable in that model, so our study appears to increase the gap between theoretical results and applicative ones. However, we may make two remarks.

First, as stated before, the real implication of our study is that no polynomial time exact algorithm can be hoped to always produce good learning results using these hypothesis languages, and that heuristics have to be used. This reflects reality since all polynomial time practical algorithms are based on heuristics.

Also, it should be noted that even in the more conventional PAC framework, a mismatch between theory and practice has been reported. Muggleton (Muggleton, 1994), for instance, attributes this to distributional assumption differences between theory and practice, then goes on to propose a new learnability model (U-learnability) which replaces the worst case analysis by an average case one. In a way, we come to the same conclusion and our work brings complementary theoretical arguments. Because it formally implies the need for heuristic algorithms, our study shows that new models incorporating limited probability distribution families and average time complexity analysis are required in order to evaluate them. Just as PAC-results often use the link with NP-Completeness theory, it seems likely that such models would rely on links with RNP-Completeness theory (Gurevich, 1991).

Acknowledgements

This work was completed in the *Machine Learning and Classification* team of the LIRMM. The authors wish to thank Colin de la Higuera for reading and correcting earlier drafts of this paper, as well as Sazo Dzeroski for his useful technical advice.

References

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

Arora, S. (1994). Probabilistic checking of proofs and hardness of approximation problems. Technical Report CS-TR-476-94. Princeton University.

Auer, P., Holte, R., and Maass, W. (1995). Theory and applications of agnostic pac-learning with small decision trees. In *Proceedings of the XII International Conference on Machine Learning, ML'95*, pages 21–29.

Banerji, R. (1987). Theory and applications of agnostic pac-learning with small decision trees. In *Proceedings of the Tenth International Joint Conference on Artificial Intelligence, IJCAI-87*, pages 280–282.

Buntine, W. (1988). Generalized subsumption and its applications to induction and redundancy. volume 36, pages 149–176.

Cohen, W. (1993a). Cryptographic limitations on learning one-clause logic programs. In *Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI'93*, pages 80–85.

Cohen, W. (1993b). Pac-learning a restricted class of recursive logic programs. In *Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI'93*, pages 86–92.

Cohen, W. (1994). Pac-learning nondeterminate clauses. In *Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI'94*, pages 676–681.

Cohen, W. (1995a). Pac-learning recursive logic programs: Efficient algorithms. *Journal of Artificial Intelligence Research*, 2:501–539.

Cohen, W. (1995b). Pac-learning recursive logic programs: Negative results. *Journal of Artificial Intelligence Research*, 2:541–571.

Dzerovski, S., Muggleton, S., and Russel, S. (1992). Pac-learnability of determinate logic programs. In *Proceedings of the Fifth Workshop on COmputational Learning Theory, COLT-92*, pages 128–137.

Garey, M. and Johnson, D. (1979). *Computers and Intractability - A guide to the Theory of NP-Completeness*. Freeman, San Francisco.

Gold, E. (1967). Language indentification in the limit. *Information and Control*, 10:447–474.

Gurevich, Y. (1991). Average case completeness. *Journal of Computer and System Sciences*, pages 346–398.

Hoffgen, K. and Simon, H. (1992). Lower bounds on learning decision lists and trees. In *Proceedings of the Fifth Workshop on COmputational Learning Theory, COLT'92*, pages 428–439.

Kearns, M., Li, M., and Valiant, L. (1987). On the learnability of boolean formulae. In *Proceedings of the Nineteenth ACM Symposium on Theory of Computing, STOCS'87*, pages 285–294.

Kearns, M., Schapire, R., and Sellie, L. (1992). Towards efficient agnostic learning. In *Proceedings of the Fifth ACM Workshop on COmputational Learning Theory, COLT'92*, pages 341–352.

Kietz, J. and Dzeroski, S. (1994). Inductive logic programming and learnability. *Sigart Bulletin*, 5:22–32.

Lavrac, N., S.Dzeroski, and Grobelnik, M. (1991). Learning non recursive definitions of relations with linus. In *Proceedings of the Fifth European Working Session on Learning, EWSL'91*.

Lloyd, J. (1992). *Foundations of Logic Programming*. Springer, Berlin, 2nd edition.

Muggleton, S. (1994). Bayesian inductive logic programming. In *Proceedings of the Seventh Workshop on COmputational Learning Theory.*

Muggleton, S. and Feng, C. (1992). *Efficient induction of logic programs.* Inductive Logic Programming. Academic Press, New York.

Quinlan, R. (1990). Learning logical definitions from relations. *Machine Learning*, 5:239–266.

Raedt, L. D. (1992). Interactive concept learning and constructive induction by analogy. *Machine Learning*, 8:107–150.

Raedt, L. D. and Bruynooghe, M. (1992). Belief updating from integrity constraints and queries. *Artificial Intelligence*, 53:291–307.

Rouveirol, C. (1992). *ITOU: induction of First Order Theories.* Inductive Logic Programming. Academic Press, New York.

Schapiro, E. Y. (1983). *Algorithmic Program Debugging.* MIT Press, Cambridge, MA.

Valiant, L. (1984). A theory of the learnable. *Association for Computing Machinery Communications*, 27:1134–1142.