Constraint-based Learning of Long Relational Concepts

Jacques Ales Bianchetti ALES@LRI.FR
Céline Rouveirol CELINE@LRI.FR
Michèle Sebag SEBAG@LRI.FR
Laboratoire de Recherche en Informatique, CNRS UMR 8623
Université Paris-Sud, 91405 Orsay, France

Abstract

Empirically, some difficulties are met in Inductive Logic Programming (ILP) when learning long concepts from large relational examples. A framework inspired from the Constraint Satisfaction paradigm has been proposed to systematically investigate the learners behavior and scalability; these experiments shed some light on the intrinsic limitations of top-down ILP learners.

This paper presents a constraint-based learner addressing the reported limitations with tractable complexity. A proof of principle of the approach is obtained as this learner efficiently solves artificial problems, relevant from a relational data mining perspective, that were beyond reach of previous relational learners.

1. Introduction

Learning from large and relational data is the challenge offered by Data Mining to Machine Learning, particularly Relational Learning and Inductive Logic Programming (ILP) (Muggleton & De Raedt, 1994; Dzeroski & Lavrac, 2001).

A novel perspective for assessing the strengths, limitations and scalability of ILP has been proposed recently (Giordana et al., 1999; Giordana et al., 2000; Serra et al., 2001). This perspective borrows to the Constraint Satisfaction area (CSP) its current complexity paradigm, rather focusing on *where the hard problems really are* (Cheeseman et al., 1991), than on the worst-case complexity. The reason why this new paradigm, known as Phase Transition phenomenon (Hogg et al., 1996), might be relevant to ILP is the fact that ILP as most learning techniques heavily relies on the covering test, and that the relational covering test most widely used in ILP is θ-subsumption (Plotkin, 1970), which is equivalent to a CSP.

Previous works have demonstrated the relevance of the phase transition to ILP (Giordana et al., 1999). Systematic experiments have been conducted on artificial, large-size, fully relational learning problems with FOIL (Quinlan, 1990). These experiments suggest that for a given range of learning problems close to the phase transition region, greedy search-based learning is bound to miss the true target concept. Practically, there exists a failure region where FOIL builds seemingly random clauses, with no better predictive accuracy than random guessing (Giordana et al., 2000). An explanation for this failure region is that greedy search has to go through large portions of the hypothesis space, where coverage-based criteria (information gain (Quinlan, 1990), Gini criterion (Botta & Giordana, 1993), MDL principle (Muggleton, 1995)) are misleading (section 2).

Several heuristics have been proposed to address the myopic search limitations (beam search (Botta & Giordana, 1993), reduced error pruning (Cohen, 1995), look-ahead (Blockeel & De Raedt, 1998), genetic-algorithms based search (Anglano et al., 1997), among others); however, these heuristics clearly cannot scale up beyond a limited extent.

As an alternative, this paper presents a constraint-based relational learner termed *cBIL*, devised to learn long concepts from large relational examples. *cBIL* exploits a language bias expressed in complexity terms (number of variables and predicate symbols), as opposed to the grammar-based bias used in e.g. DLAB (De Raedt & Dehaspe, 1997). Assuming that the target concept is rare, i.e. would hardly cover a random example with same size as the training examples, *cBIL* efficiently narrows the candidate hypotheses that are complete wrt a few positive examples. The tractable characterization of all such relevant hypotheses is achieved through a constraint-based approach; the approach provably finds the target concept if it exists in the language (section 4).

This paper is organized as follows. Section 2 briefly describes the CSP-based relational setting proposed by (Giordana et al., 1999) and the ILP limitations due to the phase transition phenomenon (Giordana et al., 2000). Section 3 presents the paper goal, that is, learning rare concepts from large relational examples. It is claimed that this goal is important from a relational data mining perspective, although not illustrated by popular ILP benchmarks. Therefore, as a proof of principle for this exploratory work, the approach is validated within the CSP-based relational setting. The core of the *cBIL* algorithm is presented in section 4, together with an average complexity analysis. Experimental results are detailed in section 5, and the limitations of the approach are extensively discussed. The paper ends with some perspectives for further research.

2. Phase Transition in ILP

This section formalizes the relational setting devised to study the phase transition phenomenon, and summarizes the main findings reported in (Giordana et al., 1999; Giordana et al., 2000).

2.1 Formalization

The hypothesis language is the first order logic language defined from a number m of literals and a number n of variables. For computational convenience, the number n of variables is set to 4 in all experiments.

Definition 1 *($\mathcal{L}_m$)*
$\mathcal{L}_m$ is the set of all m-literals conjunctive formulas h, such that all literals in h are built on distinct binary predicate symbols noted $p_1, \ldots p_m$, and involve n variables[1] $X_1, \ldots X_n$.

$$\mathcal{L}_m = \{h \mid h = \bigwedge_{k=1}^{m} p_k(X_i, X_j),\ i,j \text{ in } [1..n]\}$$

An artificial relational learning problem is defined from the target concept tc, belonging to the hypothesis language $\mathcal{L}_m$, and a dataset. Each example e in the dataset is expressed in the example language defined from the number L of distinct constants allowed in an example, and the number N of literals per predicate symbol. The working assumption (discussed in 5.2) is that all predicate symbols in the example language are relevant, i.e. are the same $p_1, ..p_m$ as those occurring in the hypothesis language. For computational convenience, the number N of literals per predicate symbol is set to 100 in all experiments.

Definition 2 *($\mathcal{E}_L$)*
The example language $\mathcal{E}_L$ is the set of all ground conjunctive formulas e

$$\mathcal{E}_L = \{e \mid e = \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{N} p_i(a_{ij_1}, a_{ij_2})\}$$

where literal $p_i(a_{ij_1}, a_{ij_2})$ is built on predicate symbol p_i, constants a_{ij_1} and a_{ij_2} belong to $\Omega = \{a_1, \ldots a_L\}$, and all literals in e are distinct.
Each example is labeled as positive iff it is subsumed by the target clause at hand, and negative otherwise.

An artificial relational learning problem noted $\Pi_{m,L}$ is defined from i) a target concept tc uniformly selected in $\mathcal{L}_m$ and ii) a training and test set of 200 examples each, with equal number of positive and negative examples, obtained by uniformly sampling $\mathcal{E}_L$ under the balanced distribution requirement.

2.2 The Failure Region

Earlier experiments on uniformly generated learning problems (Giordana et al., 1999; Giordana & Saitta, 2000) have shown that for most value pairs (m, L) and almost all (clause h, example e) pairs, where h belongs to $\mathcal{L}_m$, and example e belongs to $\mathcal{E}_L$), h either almost surely subsumes e, noted $h \prec e$, or h almost surely rejects e. In the former case, pair (m, L) is said to belong to the YES region; in the latter case, (m, L) belongs to the NO region.

In between the YES and NO region lies the narrow phase transition region, where the probability for h to subsume e abruptly drops from 1 to 0 (Fig. 1). Large-scale experiments conducted on artificial relational learning problems with[2] FOIL (Quinlan, 1990), ML-Smart (Botta & Giordana, 1993) and G-NET (Anglano et al., 1997), have shown that both top-down and genetic-algorithm -based search tend to visit

[1]Further restrictions were used in (Giordana et al., 1999) to guarantee the fairness of the experiments: formulas in $\mathcal{L}_{m,n}$ must be connected; and in order to take advantage of the symmetry of the predicates, it was assumed that all literals $p_k(X_i, X_j)$ in h are such that $i < j$ (Giordana et al., 1999).

[2]Other experiments done with PROGOL (Muggleton, 1995) or STILL (Sebag & Rouveirol, 2000) were unsuccessful: the domain-knowledge based heuristics in PROGOL were inefficient since no domain knowledge was available; and the stochastic heuristics in STILL were inefficient due to the uniform generation of the learning examples.

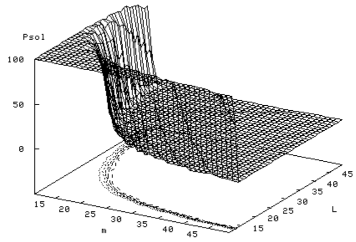

Figure 1. Average success of θ-subsumption ($h \prec e$), plotted in plane (m, L), estimated from $1,000$ pairs (h, e) uniformly selected in $\mathcal{L}_m$ and $\mathcal{E}_L$. See (Giordana et al., 2000) for more details.

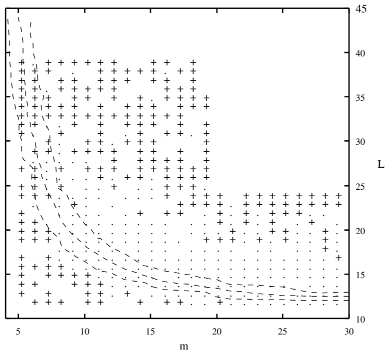

Figure 2. FOIL Competence Map and Failure Region.
'+' : Error $< 20\%$ '.' : Error $\approx 50\%$

again and again the phase transition region, and select hypotheses lying in this region (Giordana et al., 2000). In retrospect, this appears only natural: from the above remark, the non-trivial hypotheses lie in the phase transition.

The artificial datasets considered involve uniformly generated medium-size examples (one hundred literals per predicate symbol)[3]. Though indeed moderate from a mining perspective, the total example size (up to 5,000 literals) is larger than in most ILP applications (Dzeroski & Lavrac, 2001). Learning from such large examples reveals an important limitation of the current ILP search criteria (information gain, Gini criterion or MDL principle) with respect to scalability.

Practically, these criteria appear to mislead the learning search, as follows. Top-down search starts with very general hypotheses, lying in the YES region wrt the examples. In this region, a hypothesis almost surely covers any example and the number of substitutions or models highly fluctuates. Therefore, any criterion based on the hypothesis coverage or number of models will be mistaken with high probability. Experimentally, a *failure region* is detected; for all learning problems in this region, FOIL misses the true target concept, it retains a number of seemingly random hypotheses, and the predictive accuracy is comparable to that of random guessing (Fig. 2).

In optimization terms, what happens is that greedy optimization on a rugged plateau (the YES region) is trapped into a local optimum; and most local optima (hypotheses in the PT region) are very poor.

Two possibilities have been evoked to overcome this failure region problem. One is concerned with changing the domain representation, in such a way that the rewritten learning problem escapes the failure region. This perspective is that of abstraction and reformulation (Saitta & Zucker, 2000). A second approach is based on intensive search, randomly sampling the set of hypotheses sufficiently close to the PT region and computing their number of models in the examples. These samples are used to detect relevant candidate hypotheses, and starting from these, FOIL can be made to learn truly accurate hypotheses (Serra et al., 2001). A third approach is presented in this paper.

3. Learning Goal

The paper goal is learning in the NO region, i.e. when the target concept hardly covers any random example with same size as the training examples. Such problems are relevant indeed from a relational data mining perspective, for many interesting phenomena (e.g. fraud detection, diseases, etc) are rare.

Although learning in the NO region makes sense, existing ILP benchmarks do *not* belong to the NO region. Typically, the mutagenesis problem (King & Srinivasan, 1995) would appear on the extreme left of the competence map (Fig. 2), with $m = 2$ (the basic molecular representation involves predicates *atom* and *bond*), L (the number of atoms in the molecules), ranging in [22..40], and N similarly ranges in [22..40]. The only current possibility for validating a rare concept learner is therefore within the artificial relational setting[4] (Giordana et al., 1999).

[3] All details and the datasets are available at `http://www.di.unito.it/~mluser/challenge/index.html`.

[4] As a further excuse for validating our approach on artificial problems, it must be noted that all existing ILP benchmarks can be addressed by one or another judicious variant of greedy search based learning, which is not representative either of the whole range of learning problems. Actually, the limitations of greedy search become more and more apparent as the problem size increases (Chevaleyre & Zucker, 2001).

Practically, 20 relational problems lying in FOIL failure region will be considered. Each problem involves 200 training examples and 200 test examples, equally distributed among positive and negative examples. The number of literals per example ranges from 1,000 (for the smallest problem) to 2,900 (for the largest problem). On each problem, the goal is to find a formula h in language $\mathcal{L}_m$ with an arbitrarily bounded generalization error, ideally subsuming all positive training examples (*completeness*), and subsuming no negative training examples (*soundness*) (De Raedt, 1997). By construction, such a solution exists since tc itself belongs to $\mathcal{L}_m$.

It must be noted that the size of the search space $\mathcal{L}_m$ is not amenable to exhaustive exploration. Each hypothesis in $\mathcal{L}_m$ is uniquely defined (up to variable renaming) by its variable bindings, partitioning the set of $2\times m$ arguments involved in m binary literals, into four subsets corresponding to the allowed $n = 4$ distinct variables. The search space size thus is in $\mathcal{O}(n^{2\times m})$ (ranging from 10^{17} for $m = 15$ to 10^{35} for $m = 30$).

4. Constrained Bottom-Up Induction

This paper presents an alternative to top-down learning from relational data, considering a target hypothesis language $\mathcal{L}_m$ such that any hypothesis in this language is unlikely to cover any random example in the example language $\mathcal{E}_L$.

Accordingly, the number of hypotheses in $\mathcal{L}_m$ covering t examples in $\mathcal{E}_L$ decreases exponentially fast with t. The *cBIL* algorithm is based on a constraint solving mechanism, iteratively narrowing the set $\mathcal{S}$ of candidate hypotheses covering t positive examples, and incrementing t until the size of $\mathcal{S}$ makes it amenable to exhaustive validation on the rest of the training set. Assuming that the target concept belongs to language $\mathcal{L}_m$ and that examples are not noisy, *cBIL* provably discovers the target concept.

The relaxation of these working assumptions, and the cost of dealing with noisy datasets and unknown target language $\mathcal{L}_m$, will be considered in section 5.2.

4.1 Language-based generalization

Let e denote an example, and let θ be a mapping from the set of variables $\{X_1, \ldots, X_n\}$ onto $\Omega = \{c_1, \ldots, c_L\}$. For the sake of notational simplicity, $\theta = \{X_1/c_{i_1}, \ldots, X_n/c_{i_n}\}$ is identified to its image $(c_{i_1}, \ldots, c_{i_n})$, i.e. an element of Ω^n.

Let us introduce the superset of $\mathcal{L}_m$ noted $\hat{\mathcal{L}}$, including all conjunctive formulas involving any number of literals of the form $p_k(X_i, X_j)$, for k in $[1..m]$ and i, j in $[1..n]$.

Let $e|\theta$ denote the element of $\hat{\mathcal{L}}$, defined as the conjunction of all literals $p_k(X_i, X_j)$ such that literal $p_k(X_i, X_j)\theta$ belongs to e:

$$e|\theta = \bigwedge_{p_k(X_i,X_j)\theta \ in \ e} p_k(X_i, X_j)$$

By definition (Plotkin, 1970), any hypothesis h in $\mathcal{L}_m$ subsumes e iff it subsumes $e|\theta$ for some θ in Ω_n:

$$\forall\ h\ \in \mathcal{L}_m \quad ((h \prec e)\ \Leftrightarrow\ (\exists \theta / h\theta \subseteq e)\ \Leftrightarrow\ (h \prec e|\theta))$$

Inversely, any $e|\theta$ can be subsumed by some hypothesis h in $\mathcal{L}_m$ iff it involves all predicate symbols $p_1, \ldots, p_m$. More generally, any formula f in $\hat{\mathcal{L}}$ is said to be admissible, noted $Adm(f)$, iff it includes all predicate symbols $p_1, , \ldots, p_m$; one shows without difficulty that f is admissible iff it is subsumed by a hypothesis in $\mathcal{L}_m$.

Admissibility constraint is formalized through the boolean functions $M_{i,j,k}$, defined as the indicators of literal $p_k(X_i, X_j)$ for k in $[1..m]$ and i, j in $[1..n]$:

$$\begin{array}{rccl} M_{i,j,k}\ : & \hat{\mathcal{L}} & \mapsto & \{true, false\} \\ & f & \mapsto & M_{i,j,k}(f) = (p_k(X_i, X_j) \in f) \end{array}$$

For any f in $\hat{\mathcal{L}}$, one has:

$$Adm(f) = (\forall\ k\ \in [1..m],\ \exists\,(i,j)\ \in\ [1..n]^2\ s.t.\ M_{i,j,k}(f))$$

Theorem 1 *Let $\mathcal{S}(e)$ be defined as the set of all admissible formulas $e|\theta$, for e in $\mathcal{E}_L$ and θ ranging over Ω^n, and let h be a hypothesis in $\mathcal{L}_m$. Then, h θ-subsumes e iff it subsumes some $e|\theta$ in $\mathcal{S}(e)$.*

$\mathcal{S}(e)$ thus provides a compact representation of all hypotheses in $\mathcal{L}_m$ subsuming example e. Next step is to construct a compact representation for all hypotheses in $\mathcal{L}_m$ covering several examples $e_1, \ldots, e_t$.

4.2 Example-based generalization

In order to allow for a recursive construction, let f be any formula in $\hat{\mathcal{L}}$, let e be a positive training example, and let $\theta \in \Omega^n$ be a substitution; $f \otimes e|\theta$ is defined as the conjunction of all literals in f that also belong to $e|\theta$. It comes:

Theorem 2 *Let $\mathcal{S}(e_1, ..e_t)$ be recursively defined as the set of all admissible formulas $f \otimes e|\theta$, for f in $\mathcal{S}(e_1, ..e_{t-1})$, e_t in $\mathcal{E}_L$ and θ ranging over Ω^n.*

$$\begin{array}{rl} \mathcal{S}(e_1, ..e_t) = \{ & f\ \otimes\ e_t|\theta_t\ s.t.\ f \in \mathcal{S}(e_1, ..e_{t-1}), \\ & \theta_t\ \in\ \Omega^n\ \text{ and }\ Adm(f\ \otimes\ e_t|\theta_t)\ \} \end{array}$$

Then, let h be in $\mathcal{L}_m$; h subsumes $e_1, ..e_t$ iff h subsumes some formula in $\mathcal{S}(e_1, ..e_t)$.

$$(\forall i = 1..t \;\; h \prec e_i) \Leftrightarrow (\exists f \in \mathcal{S}(e_1, .., e_t) \;/\; h \prec f)$$

Proof sketch for $t = 2$.
$\Rightarrow$. If h subsumes e_1 and e_2, there exists θ_1 and θ_2 in Ω^n such that h subsumes $e_1|\theta_1$ and $e_2|\theta_2$, which implies that h subsumes $e_1|\theta_1 \otimes e_2|\theta_2$. Therefore $e_1|\theta_1 \otimes e_2|\theta_2$ is admissible and belongs to $\mathcal{S}(e_1, e_2)$.
$\Leftarrow$. Let f belong to $\mathcal{S}(e_1, e_2)$ and let h be any formula in $\mathcal{L}_m$ subsuming f. Up to variable renaming, h is obtained from f by dropping all literals except one built on predicate symbol p_k, for each $k = 1..m$. As all literals in h also belong to $f = e_1|\theta_1 \otimes e_2|\theta_2$, it follows that $h\theta_1 \subset e_1$ and $h\theta_2 \subset e_2$, which concludes the proof. □

4.3 *cBIL* Algorithm

cBIL constructs iteratively the set $\mathcal{S}_t = \mathcal{S}(e_1, \ldots, e_t)$, increasing t until the number of elements in $\mathcal{S}_t$ falls below a user-prescribed threshold B. $\mathcal{S}$ is initialized to the most specific formula in $\hat{\mathcal{L}}$, given as the conjunction of all literals $p_k(X_i, X_j)$ with $k = 1..m$ and $i, j = 1..n$.

```
cBIL(I+ : set of positive training examples;
      B : positive integer)
Init :
  S_0 = { ∧ p_k(X_i, X_j) }; t = 0
Repeat
  t ← t + 1
  Draw e_t from I+ ; I+ ← I+ \ {e_t}
  S_t = ∅
  For each f in S_{t-1}
    S_{t-1} ← S_{t-1} \ {f}
    For each θ in Ω^n
      If Adm(f ⊗ e_t|θ)
        S_t ← S_t ∪ {f ⊗ e_{t|θ}}
Until the size of S_t is less than B.
Solution ← ∅
For each f in S_t,
  For each h in L_m s.t. h ≺ f
    Assess h wrt the training set
    If err(h) = 0
      Solution ← Solution ∪ {h}
Return Solution.
```

Let us illustrate *cBIL* principle on a simplified problem ($n = 4$, $N = 3$, $m = 3$ and $L = 10$). The transformation of example e_1 and e_2 respectively in $e_1|\theta_1$ and $e_2|\theta_2$ is shown in Table 1; both $e_1|\theta_1$ and $e_2|\theta_2$ are admissible, but $e_1|\theta_1 \otimes e_2|\theta_2$ is not since it does not involve predicate p_3.

e_1

p_1	p_2	p_3
(c_1, c_2)	(c_4, c_8)	(c_2, c_4)
(c_1, c_3)	(c_1, c_4)	(c_2, c_8)
(c_9, c_2)	(c_1, c_2)	(c_2, c_3)

$e_1|\theta_1$ with $\theta_1 = (c_1, c_2, c_3, c_4)$

p_1	p_2	p_3
(X_1, X_2)		(X_2, X_4)
(X_1, X_3)	(X_1, X_4)	
	(X_1, X_2)	(X_2, X_3)

e_2

p_1	p_2	p_3
(c_6, c_7)	(c_6, c_9)	(c_6, c_7)
(c_8, c_9)	(c_5, c_8)	(c_4, c_3)
(c_1, c_4)	(c_8, c_9)	(c_2, c_7)

$e_2|\theta_2$ with $\theta_2 = (c_6, c_7, c_8, c_9)$

p_1	p_2	p_3
(X_1, X_2)	(X_1, X_4)	(X_1, X_2)
(X_3, X_4)		
	(X_3, X_4)	

$\Longrightarrow$ $e_1|\theta_1 \otimes e_2|\theta_2$

p_1	p_2	p_3
(X_1, X_2)	(X_1, X_4)	

Table 1. Construction of $e_1|\theta_1 \otimes e_2|\theta_2$

4.4 Average complexity

The central result of the paper is that *cBIL* is tractable for learning problems in the NO-region (section 3).

$\mathcal{S}(e)$ is constructed with complexity $\mathcal{O}(L^n \times n\,m\,N)$: the total number of substitutions θ is L^n and checking the admissibility of $e|\theta$ is done with linear complexity in the size of θ and e (respectively n and $2mN$).
The average size of $\mathcal{S}(e)$ is $L^n \times Pr(Adm)$, where $Pr(Adm)$ is the probability for $e|\theta$ to be admissible, i.e. to involve all predicate symbols $p_1, \ldots, p_m$.

Let p_k be a given predicate symbol; $e|\theta$ involves p_k iff there exists (at least) one variable pair X_i, X_j such that $p_k(X_i, X_j)$ is in $e|\theta$. For each (i, j), this occurs with probability N/L^2. The probability that $e_{|\theta}$ *does not* involve predicate p_k is thus lower-bounded by $(1 - \frac{N}{L^2})^{n^2}$ (the lower bound is reached for θ injective).

And since all literals built on distinct predicate symbols are independent, the probability for $e_{|\theta}$ to be admissible is upper-bounded by

$$Pr\left(Adm\left(e|\theta\right)\right) \leq \left(1 - \left(1 - \frac{N}{L^2}\right)^{n^2}\right)^m$$

As an approximation, positive examples $e_1, ..e_t$ will be considered independent; even though they all include a model of the same target concept, $100 - \frac{1}{N}\%$ (here, 99%) of their literals are constructed uniformly and independently. Therefore:

$$Pr(Adm(e_1|\theta_1 \otimes ... \otimes e_t|\theta_t)) = Pr(Adm(e|\theta))^t$$

And since the total number of such elements is $L^{n \times t}$, the average size of $\mathcal{S}_t$ is upper bounded as:

$$E(|\mathcal{S}_t|) \leq L^{n\,t} \times \left(1 - \left(1 - \frac{N}{L^2}\right)^{n^2}\right)^{m\,t} \approx \left(\frac{(n^2 N)^m}{L^{2m-n}}\right)^t$$

The approximation follows from the fact that N is small compared to L^2 (the problem range is in the NO region). First thing is to ensure that the number of elements in $\mathcal{S}_t$ does not go to infinity as t increases, i.e.:

$$\left(n^2N\right)^m < L^{2m-n} \tag{1}$$

From equation (1), the size of $\mathcal{S}_t$ decreases exponentially fast with the number t of considered examples. The average number t of examples needed to restrict the size of $\mathcal{S}_t$ under some user-supplied threshold B is given as:

$$t \approx \frac{logB}{mlog(n^2N) - (2m-n)logL}$$

This estimate for t allows us to use a depth-first exploration of $\mathcal{S}_t$ instead of a beam search in the algorithm, which ensures that the memory size remains linearly bounded in B.

Last step is to extract complete and correct hypotheses from $\mathcal{S}_t$. The completeness of each formula f in $\mathcal{S}_t$ wrt any positive example e is first checked as there must exist a substitution θ such that $f \otimes e|\theta$ is admissible, which is done with (worst-case) complexity $L^n \times n^2m$. From each complete formula f in $\mathcal{S}_t$, are extracted possibly many complete hypotheses h in $\mathcal{L}_m$, by independently selecting exactly one literal among all literals in f built on each predicate symbol. The soundness of each such hypothesis h is then checked wrt all negative examples e in the training set.

5. Validation and Discussion

5.1 Experimental setting and results

Twenty artificial problems have been considered. These are selected on a vertical segment in the NO region (the value of m is fixed to 14 while L varies from 17 to 26), and a horizontal segment in the NO region (the value of L is fixed to 18 while m varies from 14 to 27); both segments largely overlaps FOIL failure region (Fig. 2). Each problem involves 200 training examples and 200 test examples, equally distributed among positive and negative examples.

As *cBIL* is a randomized algorithm, 15 independent runs have been launched for each problem. For each problem is reported the number t of positive examples considered, the number of formulas in $\mathcal{S}_t$ and the runtime needed to construct $\mathcal{S}_t$ (in seconds on Pentium-II 300MHz), averaged over the 15 runs (Table 2).
In order to allow for a fair comparison, parameter B is set in such a way that the total number of hypotheses h in $\mathcal{L}_m$ subsuming some f in $\mathcal{S}_t$ is 200. The exhaustive test of these h on the rest of the training set, using the fast Django θ-subsumption algorithm (Maloberti & Sebag, 2001), requires less than one hour (on Pentium-II). In all cases, the target concept was found out.

Problems		Results		
m	L	t	$\lvert\mathcal{S}_t\rvert$	time
14	17	7.5 ± 0.5	37.1 ± 44	$1.3\ 10^6$
14	18	5.6 ± 0.5	2.6 ± 1.8	$6.4\ 10^3$
14	19	5 ± 0	1.4 ± 0.4	$1.2\ 10^3$
14	20	4.1 ± 0.3	3.7 ± 4.8	245
14	21	4 ± 0	1 ± 0.3	92
14	22	3.7 ± 0.5	1.2 ± 0.4	37
14	24	3 ± 0	1.2 ± 0.4	6
14	25	3 ± 0	1.3 ± 0.5	3
14	26	3 ± 0	1.4 ± 0.5	3
14	18	5.6 ± 0.5	2.6 ± 1.8	6402
15	18	5 ± 0	1.9 ± 1.4	3065
16	18	5 ± 0	1.6 ± 1.2	1715
18	18	4 ± 0	2 ± 1.1	620
19	18	4 ± 0	2.8 ± 2.1	392
20	18	4 ± 0	2.5 ± 2.6	284
21	18	4 ± 0	3.4 ± 3.3	218
23	18	4 ± 0	2.2 ± 1.0	143
24	18	4 ± 0	2.8 ± 2.6	117
26	18	4 ± 0	3.0 ± 2.6	78
27	18	3.2 ± 0.4	1.1 ± 13.5	66

Table 2. cBIL: Number t of positive examples considered, size of $\mathcal{S}_t$, and runtime, on two problem sets in FOIL failure region.

Compared to the previous, search intensive approach proposed to address the FOIL failure region (Serra et al., 2001), *cBIL* improves by two orders of magnitude regarding the computational cost.

5.2 Limitations

Let us consider the assumptions done insofar and see to which extent they limit the proposed approach. The central assumption regards the target hypothesis language $\mathcal{L}_m$, fixing the number n of variables, the set of relevant predicate symbols $p_1, \ldots, p_m$, and the number of literals per predicate symbol (here, one). Indeed, this language specification reflects a deep prior knowledge into the problem at hand, which cannot be assumed to be available in the general case for real-world applications.

It thus remains to see how the needed information can be acquired in case it is not available, and the acquisition cost. A second limitation to be considered is that of noise-free datasets.

Target number of variables. Let n^* denote the (unknown) target number of variables, and let us evaluate the cost of estimating n^*, everything else being equal. The standard heuristics would be to run *cBIL* with gradually increasing values of n, until a solution can be found out. The overall cost would then be $Cost(2) + \ldots + Cost(n^*)$; and since *cBIL* cost is exponential in n, the computational overhead required to estimate n^* is linear in the total learning cost.

Target number of literals per predicate symbol. Let us assume that the target concept tc actually belongs to the language $\mathcal{L}_m^{k_1,..,k_m}$, involving conjunctions of k_1 literals built on p_1,.. k_m literals built on p_m.
It must be noted that *cBIL* actually works in the intermediate language $\hat{\mathcal{L}}$, which contains all $\mathcal{L}_m^{k_1,..,k_m}$: each formula in $\mathcal{S}_t$ involves any number of literals per predicate symbol between 1 and n^2 (the number of variable pairs). Therefore the construction of $\mathcal{S}_t$ needs not be modified: tc will still be discovered by *cBIL* as tc subsumes at least one formula in $\mathcal{S}_t$.
The only impact of varying the number of literals per predicate symbol is that it might be the case that, although $\mathcal{L}_m^{k_1,..,k_m}$ belongs to the NO region wrt the examples, $\mathcal{L}_m$ does not. In such cases, the admissibility constraint as defined in section 4 does not allow for an efficient restriction of $\mathcal{S}_t$ and it must be revised accordingly (e.g. requiring the total number of literals in any formula in $\mathcal{S}_t$ to be greater than a given threshold, to be determined wrt the examples and the phase transition location).

Target predicate symbols. Let us assume that the examples involve M predicate symbols, and let the (unknown) target number of predicate symbols be m^*. Without any prior knowledge, the simplest heuristics would be to run *cBIL* with all subsets of 1, 2,.. (and $M-1$, $M-2$,..) predicates among the M initial symbols, entailing a multiplicative computational overhead of $2\,min\,(M^{m^*}, M^{M-m^*})$ compared to the overall cost when the relevant predicate symbols are known.
The selection of the relevant predicate symbols thus appears as the current central limitation of the approach.
On one hand, this limitation must be considered in the perspective of the state of the art. Actually, the predicate symbol selection done in e.g. FOIL, PROGOL, TILDE among others, relies on the greedy optimization of some coverage-based criteria. But, *cBIL* is devoted to relational learning precisely in regions where greedy optimization fails. Further efforts will be devoted to the case of highly complex relational problems, insofar unsuccessfully tackled by the ILP community, and examine whether the main complexity factor actually comes from the number of predicate symbols (tables) considered.

Noise-free data. A simple and elegant heuristics has been proposed by Kearns and Li (Kearns & Li, 1993) to turn a robust algorithm with small sample complexity in the noise-free case, into a learning algorithm robust wrt malicious noise. Along the same lines, *cBIL* can be run independently K times. Assuming that the error rate ε is less than $\frac{1}{t}$, where t is the number of examples required for *cBIL* to adequately restrict the number of candidate hypotheses, one of these runs will select noise-free positive examples only (with probability $1-(t\varepsilon)^K$); hence the target concept will be one among the learned hypotheses. Evaluating the learned hypotheses on a disjoint validation set should allow with high confidence to distinguish between the true target concept (the predictive accuracy of which should be 100% - ε) and spurious hypotheses (the average predictive accuracy of which should be around 50% since they belong to the NO region wrt the examples).

6. Conclusion and Perspectives

This paper brings some positive results on learning long and rare concepts from large relational examples.

The proposed approach takes into account the fact that, on large problem domains, coverage-based criteria do not provide enough information to reliably explore the generalization lattice. The difficulty is due to the fact that no elementary part of the (long) target solution is found sufficiently discriminant by itself.

As an alternative, the *cBIL* algorithm considers a target hypothesis language with large but fixed complexity. The choice of this search space equally prevents greedy search (all hypotheses have same size) and exhaustive search (there are too many of them to explore them all). To our best knowledge, the approach is novel wrt the main ILP trend, as most prominent ILP learners are based on top-down search (Quinlan, 1990; Muggleton, 1995), occasionally combined with backtracking, pruning or look-ahead (Cohen, 1995; Blockeel & De Raedt, 1998).

cBIL constraint solving mechanism exploits the structure of the search space together with a handful of positive examples, to determine the *a priori* relevant hypotheses, i.e. complete wrt the positive examples. The good news is that a few positive examples makes it feasible to restrict the candidate hypotheses to such an extent, that their exhaustive assessment becomes feasible. Further, the approach provably discovers the target concept or a generalization thereof, if it belongs to the target hypothesis language. A proof of principle of the approach has been presented on artificial datasets, improving by two orders of magnitude in terms of computational cost on a previous search intensive approach (Serra et al., 2001).

The current key limitation of the approach is the selection of the target hypothesis language, more precisely the selection of the relevant predicate symbols. On-going research is concerned with applying the approach on relational problems which are not amenable to short-clause learning.

Acknowledgments

We gratefully acknowledge Lorenza Saitta, Attilio Giordana and Marco Botta for many enjoyable discussions about Phase Transition and its implications on ILP. Thank also to Jérome Maloberti, who made publicly available the fast θ-subsumption algorithm Django.

References

Anglano, C., Giordana, A., Lobello, G., & Saitta, L. (1997). A network genetic algorithm for concept learning. *Proc. of the 7th Int. Conf. on Genetic Algorithms* (pp. 434–441). Morgan Kaufmann.

Blockeel, H., & De Raedt, L. (1998). Top-down induction of first order logical decision trees. *Artificial Intelligence*, *101*, 285–297.

Botta, M., & Giordana, A. (1993). SMART+: A multi-strategy learning tool. *Proc. of the 13th Int. Joint Conf. on Artificial Intelligence* (pp. 937–943). Chambéry, France.

Cheeseman, P., Kanefsky, B., & Taylor, W. (1991). Where the really hard problems are. *Proceedings 12th Int. Joint Conf. on Artificial Intelligence* (pp. 331–337). Sidney, Australia.

Chevaleyre, Y., & Zucker, J.-D. (2001). A framework for learning rules from multiple instance data. *Proc. of the 12^{th} European Conf. on Machine Learning* (pp. 49–60). Springer Verlag.

Cohen, W. W. (1995). Fast effective rule induction. *Proc. of the 12^{th} Int. Conf. on Machine Learning* (pp. 115–123).

De Raedt, L. (1997). Logical settings for concept learning. *Artificial Intelligence*, *95*, 187–201.

De Raedt, L., & Dehaspe, L. (1997). Clausal discovery. *Machine Learning*, *26*, 99–146.

Dzeroski, S., & Lavrac, N. (Eds.). (2001). *Relational data mining*. Springer Verlag.

Giordana, A., Botta, M., & Saitta, L. (1999). An experimental study of phase transitions in matching. *Proc. of the Int. Joint Conf. on Artificial Intelligence* (pp. 1198–1203). Morgan Kaufmann.

Giordana, A., & Saitta, L. (2000). Phase transitions in relational learning. *Machine Learning*, *41*, 17–251.

Giordana, A., Saitta, L., Sebag, M., & Botta, M. (2000). Analyzing relational learning in the phase transition framework. *Proc. of 17th Int. Conf. on Machine Learning* (pp. 311–318). Morgan Kaufmann.

Hogg, T., Huberman, B., & Williams, C. (1996). Phase transitions and the search problem. *Artificial Intelligence*, *81*, 1–15.

Kearns, M., & Li, M. (1993). Learning in the presence of malicious errors. *SIAM J. Comput.*, *22*, 807–837.

King, R., & Srinivasan, A. (1995). Relating chemical activity to structure: An examination of ILP successes. *New Generation Computing, Special issue on Inductive Logic Programming*, *13*, 411–434.

Maloberti, J., & Sebag, M. (2001). Theta-subsumption in a constraint satisfaction perspective. *Proc. of 11th Int. Conf. on Inductive Logic Programming* (pp. 164–178). Springer Verlag.

Muggleton, S. (1995). Inverse entailment and PROGOL. *New Generation Computing*, *13*, 245–286.

Muggleton, S., & De Raedt, L. (1994). Inductive logic programming: Theory and methods. *Journal of Logic Programming*, *19*, 629–679.

Plotkin, G. (1970). A note on inductive generalization. In *Machine intelligence*, vol. 5. Edinburgh University Press.

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

Saitta, L., & Zucker, J. D. (2000). Abstraction and phase transitions in relational learning. *Proc. of SARA 2000, LNAI 1864* (pp. 291–302). Springer Verlag.

Sebag, M., & Rouveirol, C. (2000). Resource-bounded relational reasoning: Induction and deduction through stochastic matching. *Machine Learning*, *38*, 41–62.

Serra, A., Giordana, A., & Saitta, L. (2001). Learning on the phase transition edge. *Proc. of the Int. Joint Conf. on Artificial Intelligence* (pp. 921–926). Morgan Kaufmann.