Feature Subset Selection and Relational Learning

Erick Alphonse ALPHONSE@LRI.FR
Stan Matwin STAN@LRI.FR
LRI - Bt 490
Universit Paris-Sud
91405 ORSAY CEDEX

Abstract

This paper argues that in order to perform data mining on large relational databases with multiple tables, one needs to go beyond the traditional attribute-value learning (AVL) techniques. Inductive Logic Programming lifts the expressivity of both the examples and the hypotheses to the level of first-order logic, well-suited for this task. Several subsets of FOL with different expressive power have been proposed in ILP. The Datalog language is expressive enough to represent realistic learning problems when data is given directly in a multi-relational database. The difficulty lies in the fact that the more expressive the hypothesis language the learner works with, the more critical the dimensionality of the learning task. The dimensionality problem, addressed for decades in Machine Learning, is typically tackled by Feature Subset Selection (FS) techniques. The idea of re-using these techniques in ILP runs immediately into a problem as examples have variable size and do not share the same set of literals. We propose here the first paradigm that brings Feature Subset Selection to the level of ILP, in languages at least as expressive as Datalog. The main idea is to approximate the original relational problem by a multi-instance problem, a representation suitable for FS techniques. The method acts as a filter, preprocessing the relational data, prior to the model building, which outputs relational examples with empirically relevant literals. An implementation of the paradigm is proposed and successfully applied to the biochemical mutagenesis domain.

1. Introduction

Even though the expressiveness of Inductive Logic Programming is attractive for many modern applications, it is generally agreed that there exists a dichotomy between expressiveness of the representation language and the efficiency of learning(Rouveirol, 2000). From the efficiency perspective, one of the main difficulties that prevents ILP from tackling large-size problems is the dimensionality of the task, significantly higher than that of attribute-value learning (AVL).

There is no generally accepted theoretical measure of the dimensionality of ILP problems, but researchers agree that the major factor is the complexity of the hypothesis space. Hypothesis spaces in ILP are considerably larger (even infinite under θ-subsumption) than in AVL. The standard Machine Learning solution to the overgrown hypothesis space is the idea of a declarative bias. Traditionally in ILP, declarative (static) biases were defined to constrain the hypothesis language. Such biases were mainly constraints on the hypothesis language, introduced by the user (Nedellec et al., 1996). Their effect may be detrimental for the learning task, heavily relying on the user's intuition about the solution: either the hypothesis language can become too restrictive to express the target concept, or too general so that their constraining effect becomes unsignificant. In this paper, we focus on reducing another factor influencing dimensionality of ILP: the complexity of the example space. The *complexity* of the example space is understood here not as the number of examples as in (Fürnkranz, 1997), but as the size of the examples in terms of the number of literals by wich the examples are expressed. Such example complexity also contributes to the dimensionality of the ILP task for the following reasons. Firstly, the hypothesis language is based on the example language, so the simpler the example language, the smaller the hypothesis space. Secondly, the coverage test in ILP (involving logical matching of FOL formulae) is NP-complete, and therefore, in the worst case, is exponentially more expensive in the size of the examples than the same operation in AVL. Thirdly, if we are able to filter out some predicates from all examples, hypothesis containing only such literals do not need to be retained, so the search space will shrink, improving the heuristic function. All the three situations make it interesting to decrease the size of the examples by eliminating some literals from them.

In the AVL context, the task of limiting the size of examples has been successfuly handled by Feature Selection (FS) methods. Several successful approaches to FS have been proposed (Kira & Rendell, 1992; Kohavi & John, 1997), and are widely used not only in research but also in industrial practice, as feature selection functions are included in commercial data mining systems (e.g. Mineset). And yet, as pointed out by Fürnkranz (Fürnkranz, 1997) the idea of dynamic, data-driven reduction of the the example space, common in AVL in the form of feature selection, has been little researched in ILP. Since the FS approach to control dimensionality of the learning task has proven fruitful in AVL, we investigate in this paper how a similar data-driven transformation of examples could be applied in ILP.

However, the idea of performing FS in an ILP setting runs immediately into a problem: what is an attribute in ILP? All the dynamic FS methods rely on the values of a fixed set of attributes to evaluate their relevance. In ILP, however, there is no fixed set of attributes for a given problem: predicate symbols may change from example to example, and examples have a variable number of literals. In this paper, we show effective methods that address these and other problems and show how feature selection can be brought to the realm of ILP.

It has to be noted that Lavrač et al. (Nada Lavrač & Jovanoski, 1999) have proposed a feature selection framework in ILP, using a constrained language named Deductive Hierarchical DataBase (DHDB). On the one hand, since DHDB does not allow non-determinate existential variables, the coverage test complexity is quadratic and all problems described in DHDB can be compiled into propositional logic in quadratic time, and then feature selection techniques can be appliedin a straightforward manner. On the other hand, DHDB is too limited to handle current ILP benchmark datasets like mutagenesis (sect. 5) or relational databases [ADD REF TO FINANCIAL CHALLENGE FROM YC THS more information??] [11].

Our approach to FS in ILP is more general: it works with relational examples that can be expressed as non-recursive Datalog clauses. Seen as a black box, our FS filter processes FOL examples one by one, and removes from them literals which are judged irrelevant. This is achieved by first performing the change of representation which approximates a FOL probalem as an attribute-value problem, and then applying a purposefully modified version of an attribute-value feature-selecting filter. We are going to show that the change of representation has certain properties which result in interesting constraints on the design of the FS algorithm. Feature selection algorithms working under these constraints, as far as we know, have not been investigated in the literature.

In the next section we describe, using generic components, the main idea of our approach. We then introduce a version of the multi-instance propositionalization which produces our approximation of the original ILP task. In section 4 we discuss some problems facing our approach to example reduction,and illustrate them on astandard, artificial domain. The method is then empirically validated on the mutagenesis problem, a real-word dataset in ILP used as benchmark section 5. We conclude after discussing the experiments with our approach.

2. Overview of the architecture

We present here the first approach to feature selection in ILP, capable of dealing with ILP problems which are not easily transalted to AVL problems. Our method works with ILP problems in which examples are non-recursive Datalog clauses. The general idea of our approach is illustrated in Fig. 1. First of all, both the inputs and the outputs of our method are relational examples, which we view as sets of literals. In that sense, the method is similar to AVL-type feature selection, where the inputs and the outputs are vectors of attributes. All AVL feature selection methods rely on a fixed set of attributes, the same for all examples, to evaluate their relevance for the learning task. In order to bridge the gap between the flexible format of ILP examples and the fixed representation requirement of FS in AVL, we use example's literals as a fixed set of attributes, and attempt to reduce them. As shown in Fig. 1, each example is subject to a change of representation, with all the other examples participating in this change. Our change of representation is a propositionalization, in which each relational example is converted into a set of AV examples. This makes us view the propositional representation as a multi-

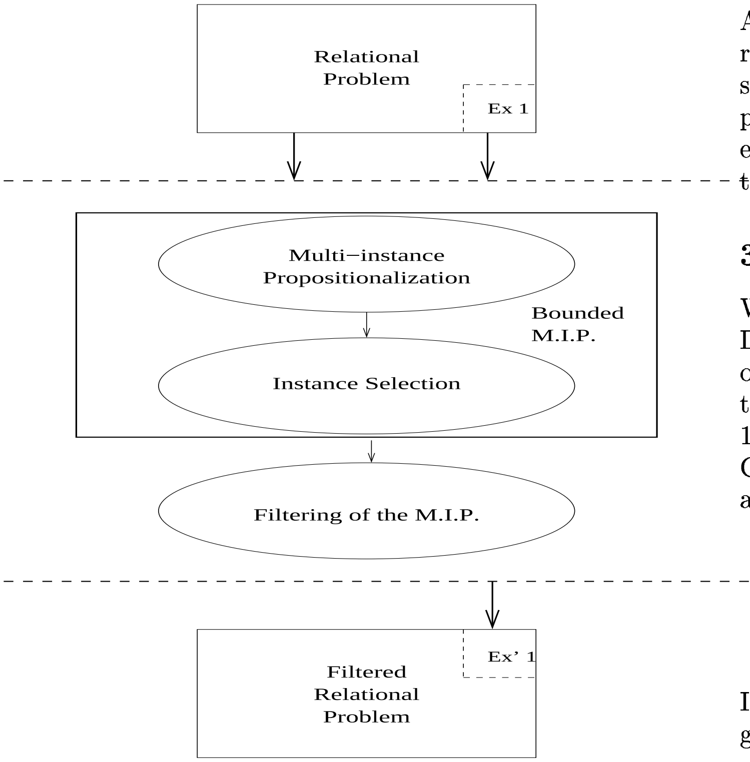

Figure 1. Filtering a relational problem

instance problem: a bag of AV examples is obtained from a single relational example, and all examples in the bag inherit the label of their relational source.

This so-called multi-instance propositionalizationis (MIP) has been investigated by several researchers (**?**; Zucker & Ganascia, 1996; Alphonse & Rouveirol, 2000) UPDATE BIBLIO WITH ML 2000 SEB. In MIP, each ILP example is tranformed into a set of fixed-length vectors of AVL attributes. The vector format, same for each positive example, has a Boolean attribute for each literal of the example. Each boolean attribute describes the fact a literal of this example matches (under some substitution) a literal of some other, remaining example. Having obtained a set of AV vectors from an ILP example, we are able to evaluate the relevance of each attribute in the AV setting, and consequently the relevance of the literal that corresponds to it in the ILP setting.

Since the coverage test in FOL is NP-complete, the number of mtachings between MIP instances corresponding to the original ILP problem may be exponential. As MIP produces an AV vector for each matching, we may obtain an exponential number of AV vectors. Consequently, we perform instance selection on the space of all the vectors corresponding to mtachings, and work with a limited number of such examples. All the propositional examples resulting from a single relational example are then given to an AVL feature selection system, engineered to respect constraints imposed by the special chaarteristics of the propositioanl examples (e.g. special kind of noise), which result from the change of represenation as we perform it.

3. Multi-instance Propositionalization

We address learning whare examples are non-recursive Datalog clauses, i.e. non recursive Horn clauses without function symbols other than constants. We use the typical learning by implication paradigm (de Raedt, 1997), described as follows:
Given a set of positive examples E^+ and negative examples E^-, find a hypothesis, h, such that:

- $\forall e^+ \in E^+, h \geq e^+$
- $\forall e^- \in E^-, h \not\geq e^-$

In other words, we need to find a hypothesis which generalizes or subsumes all positive examples and does not subsume any negative examples. In this setting, the logical implication is equivalent to θ-subsumption (Plotkin, 1970; Gottlob, 1987), which is decidable but NP-complete.

Aspointed out by (Sebag & Rouveirol, 2000) there are two kinds of literals in ILP. Structural literals represent relations between functional attributes that represent properties of the examples. As the first attempt to bring feature subset selection to ILP, we restrict ourselves to the structural part of the relational learning, which is only concerned with discrimination by predicate occurrence(s) and variable links between variables, even if the multi-instance propositionalization is general enough to address both learning problems (**?**; Sebag & Rouveirol, 2000). The structural problem is typically the non-determinate part of the learning problem, and therefore the part where the dimensionality is the most critical.

We use here the multi-instance propositionalization proposed in (Alphonse & Rouveirol, 2000). (Zucker & Ganascia, 1996) first investigated this kind of change of representation but their transformation is tailored for the REPART system, and is not general enough for our purpose. In our approach the multi-instance propositionalization is a representation change that reformulates the FOL learning problem as a multi-instance problem, aiming at preserving the expressive power of the original problem. We present in the following only the relevant information needed to understand the FS paradigm proposed in ILP.

The approach initializes the pattern P as one of the FOL examples, and the fixed set of literals of P is used as a fixed set of attributes to redescribe each other FOL example e. Given a substitution, we match (a subset of) P's literals to those of e. The value of each attribute in the redescribed representation says wheteher or there is a match (true) between between this attribute (a literal of P) and some literal of e. In our representation language, there are many differents ways to matched a given literal and therefore, e will be redescribed as several attribute-value vectors. For convenience, we will not distinguish in the remainder of the paper a matching (substitution) from its associated boolean vector.

This procedure outlined above is examplified as follows. Let us consider the non-recursive Datalog clausal space ordered by θ-subsumption. Let E, and E' be two positive examples and NE be a negative example of the target concept, inspired by R. Michalski's trains problem:

E': train(t) :-	car(t,c1),short(c1),load(c1,l11),rect(l11), car(t,c2),short(c2),load(c2,l21),circ(l21).
E: train(t) :-	car(t,c1),long(c1),load(c1,l11),rect(l11), car(t,c2),long(c2),load(c2,l21),hex(l21), car(t,c3),short(c3).
NE: train(t) :-	car(t,c1),long(c1),load(c1,l11),hex(l11), car(t,c2),short(c2),load(c2,l21),circ(l21), car(t,c3),long(c3),load(c3,l31),rect(l31).

To redescribe the whole dataset, we use E' to construct the pattern P. It is built as the following variabilization of E' (omitting the head):

$$P \quad : \quad \text{car(V,W),short(W),load(W,X),rect(X),} \\ \text{car(V,Y),short(Y),load(Y,Z),circ(Z).}$$

We can observe that since constant names are local within clauses, it does not matter whether we work with skolemized clauses, or clauses variablized so that links between literals are preserved in the variabilization. Given P, we build the new instance space showed in Table 3. Let us more closely consider how example E is reformulated. The propositionalization algorithm searches for all substitutions which, when applied to literals of P, will result in literals belonging to E. Without loss of generality, we use the links between variables to constrain the matching space being searched. The propositionalization process first builds the substitution $\sigma_{E,1} = \{V/t, W/c1, X/l11, Y/c1, Z/l11\}$. Note that the literals $short(W)$, $short(Y)$ and $circ(Z)$ of P have not been matched to any literal of E, hence the value "false" (0). When searching for another possible matching for P's literals, the next valid substitution computed is another matching for $car(V,Y)$: $\sigma_{E,2} = \{V/t, W/c1, X/l11, Y/c2, Z/l21\}$. Again, another matching of this literal with E yields $\sigma_{E,3} = \{V/t, W/c1, X/l11, Y/c3, Z/l31\}$. Note that for this substitution, neither the literal load(Y, Z) have been matched. In our example, six other substitutions (and therefore six other boolean vectors) are obtained when searching forall possible partial matchings of variables of P with constants of E.

As each vector represents a matching of P's literals to the ones of the FOL examples, each reformulated example is described by a set of vectors more general than or equal to P. Therefore, by construction, the propositionalization pattern P is represented by the bottommost element of the new instance space (σ_P in table 3). There exists a one-to-one mapping between each vector and the corresponding FOL description where the syntactic representative is simply a subset of the chosen propositionalization pattern.

As pointed out in (Zucker & Ganascia, 1996), the new instance space is no longer a set of positive and negative vectors but is now a so-called *multi-instance problem*(Dietterich et al., 1996): for each positive FOL example, at least one of its associated vectors is positive, and for each negative example, all its associated vectors are negative.

We extend the equivalence theorem (Zucker & Ganascia, 1996) to our more general framework:

Theorem 1 *In the learning by implication paradigm, where examples are non-recursive Datalog clauses, and a pattern P (a formula of the language), searching for a generalization of P that is a subset of P (up to a variable renaming) is equivalent to solving the multi-instance problem, obtained from multi-instance propositionalization given P and $\preceq$*

4. Feature Selection in ILP

In the proposed approach, feature selection is performed by first reformulating the relational problem, and then by applying feature subset selection techniques. Consequently, the filter methods used for feature selection must perform well in the new instance space. As far as we know, that has not been investigated yet by the machine learning community. We are going to gradually introduce the underlying problems. More precisely, the approximation of the original relational problem proposed will inevitably produce noise, and therefore, the issue of noise resistance has to be addressed while designing a filter algorithm.
First of all, it has been pointed out that the underlying multi-instance representation of the reformulated problem could be seen as a class-noisy representation of the positive data (Blum & Kalai, 1998; Chevaleyre & Zucker, 2000). According to definition **??**, each

Table 1. The tabular representation of a FOL problem

P	car(V,W)	short(W)	load(W,X)	rect(X)	car(V,Y)	short(Y)	load(Y,Z)	circ(Z)
σ_P	1	1	1	1	1	1	1	1
$\sigma_{E,1}$	1	0	1	1	1	0	1	0
$\sigma_{E,2}$	1	0	1	1	1	1	0	0
$\sigma_{E,3}$	1	0	1	0	1	1	0	0
...								
$\sigma_{NE,1}$	1	0	1	0	1	1	1	1
$\sigma_{NE,2}$	1	0	1	0	1	0	1	0
...								
$\sigma_{NE,j}$	1	0	1	1	1	1	1	1
...								

positive FOL example is represented by a set of vectors, such that covering only one of them is sufficient to cover the FOL example. This problem has been studied in the ILP community and relaxing the completeness of top-down algorithms is typically applied (Zucker & Ganascia, 1996; **?**; Alphonse & Rouveirol, 2000).

In order to illustrate the impact of noise, we upgraded the boolean-attribute filter FOCUS (**?**), as indicated above, and performed simple experiments on a train-like artificial problem similar to the one of sect. **??**. FOCUS uses an exhaustive search in the attribute space and is not noise-resistant. Two datasets have been generated wherein a conjunctive and a disjunctive concept were used to split them. Each train contains five cars, giving a matching space under θ-subsumption of size 5^5.

4.1 The conjonctive learning case

We validate first the relevance of the approach developed in the paper, by multi-instance propositionalizing the FOL problem and retaining all vectors, that is, we explored the whole matching space between the pattern and am example. As showed in the first row of table **??**, FOCUS finds all relevant literals and only them. This behaviour is a corollary of theorem **??**, as in the learning by implication paradigm, the minimal set of relevant attributes is also a solution of the concept learning case.

However, as pointed out by (Sebag & Rouveirol, 1994), attribute-value algorithms working on the reformulated problem must deal with data of exponential size wrt the FOL problem. For example, under θ-subsumption and given a pattern P, a FOL example is theoretically to be reformulated as a set of $\prod n_i^{m_i}$ vectors, where n_i and m_i are the number of occurrences of non-determinate literals based on the predicate symbol p_i in the FOL example e and in P, respectively. For instance, in the mutagenesis dataset, described in section 5 the potential number of matchings is 40^{40}. But this set is highly redundant (Alphonse & Rouveirol, 2000) and only few vectors are indeed sufficient to represent the whole instance space. Algorithms can be designed to approximate this set of non-redundant vectors in order to cope with the intractability of the MIP. Indeed, this set is known to be the set of the most specific vectors in the boolean lattice-like instance space (the nearest-misses and nearest-hits of the pattern). An approximation can be achieved by working with a subset of vectors whose elements are as close as possible to the non-redundant, minimal elements. We will refer to this approximation of the initial relational problem as *bounded* multi-instance propositionalization. This approximation will produce a generalization of the most specific vectors. This generalization can be viewed as obtaining the most specific vectors from a source of noise, in which some attributes' values "true" have been flipped to values "false", introducing attribute noise. Due to our particular learning setting, the attribute-noise impact differs between negative and positive examples. For the former, the so generalized negative examples are still negative. For the latter, flipping true to false can transform the positive vector into a negative one.

For now, we have only investigated a very simple bounded MIP scheme, following the idea of (**?**) of sampling the matching space. We apply a stochastic process where k matchings are selected to yield a bounded reformulated problem, with k being a user-supplied parameter. This approach has been successfully used in the learning system STILL (Sebag & Rouveirol, 2000). Row 2 in table **??** shows FOCUS' result on the approximated problem.

5. Experiments

To evaluate the approach we use the ILP system PROPAL (Alphonse & Rouveirol, 2000) as learning algorithm.

	relevant literals (%)	size of clauses
CE	100	3
CS	80	3
DS	25	4

Table 2. Filtering on an artificial problem

We have evaluated the approach by performing experiments on the two Mutagenesis datasets, well-known ILP problems used as benchmark tests. In these problems, each example consists of a structural description of a molecule as a definite clause. The molecules have to be classified into mutagenic and non-mutagenic ones. The representation language used has been defined from background knowledge B_0, which uses only relational literals, tackling nominal and ordinal arguments as constants (see (Srinivasan et al., 1994) for a detailed explanation). In a few words, positive and negative examples of the target concept are molecules described in terms of atoms (between 25 and 40 atoms) and bonds between some of these atoms.

One of the two datasets is "regression-friendly", in which a good regression analysis can be performed, composed of 188 molecules, and the other one, "regression-unfriendly", composed of 42 molecules. The experimental protocol is the one provided in (Srinivasan et al., 1994). The accuracy of the learned theory for the regression-friendly dataset (RF) is evaluated by a 10-cross-validation (the 10 folds being already given), and the accuracy on the regression-unfriendly (RU) is evaluated by a leave-one-out procedure. Each learning time is calculated by performing learning on the whole dataset.

Table 3 compares the PROPAL's performance on the Mutagenesis datasets filtered and not filtered. The time to filter the two databases is negligible. Due to the stochastic process of the bounded propositionalization used, each accuracy and time have been averaged over 10 runs; the standard deviation is given.
As expected, we can see that the performance of PROPAL has been improved, both in accuracy and time. It is interesting to notice the small standard deviation obtained for the accuracy, although the total number of vectors extracted is really small compared to the size of the matching space in mutagenesis. That could be evidence that this space is highly redundant, and the filtering step does not suffer from the poor bounded propositionalization scheme, at least for the mutagenesis problems. The deterioration of the performance for the case with $k = 500$ is most likely due to the increase of noise into the data by working with a larger sample of the noisy instance space.

6. Discussion

Several remarks are in order to summarize the implementation of the paradigm used for the validation. Firstly, we use a very simple bounded MIP scheme which does not depend on the application and the structure of the space of matchings being searched. There is need for an informed sampling scheme here, one which would take into account the partial ordering between instances in order to extract vectors as specific as possible and exploit particularities of the application. Secondly, we can observe that the parameter k, specifying the size of the example space sampled from the entire search space, is arbitrary in the approach described here. At this point, we do not have a good grasp of the range of values of k which will allow efficient and yet precise learning from the sample. We can observe that although one could anticipate k to be large, our experiments indicate that even a very small k (wrt the total size of the matching search space) is adequate, at least in the mutagenesis application. Thirdly, with the upgrade of feature selection techniques, there is also an upgrade of the filter algorithm, which has to take into account our particular settings. Tackling a bounded MIP needs to be further investigated. We proposed here a Relief-like algorithm for its ability to cope with attribute-noise and numerical attributes (not yet evaluated). The method described does not use any positive examples other that P. Clearly, there is room for improvement here: use of positive examples should allow for a tighter, more focused search space, and consequently a better approximation of the relevant literals. However, the restricted approach proposed here is worth considering further due to its low complexity, and its constant space requirement which could one allow to handle arbitrarily large sample size.

7. Conclusion

On the one hand, relational problems typically have to cope with dimensionality challenges harder than those in AVL learning. On the other hand, Machine Learning research has developed effective techniques for dealing with dimensionality by means of feature selection filters. It is not obvious, however, how to perform feature selection in ILP because there is no notion of a fixed set of features for a given ILP task. We have resolved this problem and proposed a general paradigm enabling feature subset selection techniques to be applied in ILP, in languages at least as expressive as Datalog. Our design criteria focused on the ability to handle FOL problems expressive enough to support KDD applications. This ability means that

Table 3. Comparison of the PROPAL's and FOIL's performance on the original datasets and the filtered ones

		PROPAL			FOIL		
		$k = 100$	$k = 500$	original	$k = 100$	$k = 500$	original
RF	Accuracy (%)	78.09±0.34	78.25±0.15	75.8	85.54 ±1.3	85.50 ±1.73	81.80
RF	Time (s.)	??±0.34	??±0.15	??	69 ±26	110 ±1	290
RU	Accuracy (%)	80.71 ±1.75	85.50 ±1.73	71.4	-	-	-
RU	Time (s.)	??±0.34	??±0.15	??	69 ±26	110 ±1	290

a FOL representation must include free, existentially quantified variables. As far as we know, it is the first feature selection method which works with reasonably expressive subsets of FOL. Moreover, our method is applicable to any partial ordering of the learning space (e.g. theta-subsumption).

The main idea is to consider filtering one relational example at a time. This uses its literals as a fixed set of attributes for approximating the relational problem by a MIP. Filtering the MIP allows us to assess the relevance of the example's literals. To filter the whole relational problem, we consider each example in turn. In this paradigm, the feature selection techniques work as a front end filter, applied prior to further processing and to the model building.

The reformulated problem which has to be filtered is a noisy MIP, which has not been investigated yet by the machine learning community. We have proposed a simple solution to this problem but further research is needed. A possible extension is to develop a model of noise introduced by the change of representation as described in Sect. **??**, and then to come up with an AVL FS under this model, maybe following (Chevaleyre & Zucker, 2000).

In order to evaluate the relevance of the approach, we have proposed a simple implementation, and we have performed experiments on two datasets in mutagenesis, a real-word biochemical domain, considered benchmark tests in ILP. The results are encouraging, showing as expected, an improvement both in time and accuracy. We plan to further investigate the validity of the approach on problems presented by the KDD community, still considered out of the scope of current ILP techniques.

Acknowledgements

We are very grateful to Michèle Sebag and Céline Rouveirol for discussing their works with us. The support of the Natural Sciences and Engineering Research Council of Canada and of the Computing and Information Technologies Ontario are kindly acknowledged.

References

Alphonse, E., & Rouveirol, C. (2000). Lazy propositionalization for relational learning. *Proc. of the 14th European Conference on Artificial Intelligence (ECAI'2000)* (pp. 256–260). Berlin: IOS Press.

Blum, A., & Kalai, A. (1998). A note on learning from multiple-instance examples. *Machine Learning*, *30*, 23–29.

Chevaleyre, Y., & Zucker, J. (2000). Noise-tolerant rule induction for multi-instance data. *ICML 2000, Workshop on Attribute-Value and Relational Learning.*

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

Dietterich, T., Lathrop, R., & Lozano-Perez, T. (1996). Solving the multi-instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*, 31–71.

Fürnkranz, J. (1997). Dimensionality reduction in ILP: A call to arms. *Proceedings of the IJCAI-97 Workshop on Frontiers of Inductive Logic Programming* (pp. 81–86). Nagoya, Japan.

Gottlob, G. (1987). Subsumption and implication. *Information Processing Letters*, *24*, 109–111.

Kira, K., & Rendell, L. A. (1992). A practical approach to feature selection. *Proc. of the Ninth International Conference on Machine Learning.* (pp. 249–256). MK.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*, 273–323.

Nada Lavrač, D. G., & Jovanoski, V. (1999). A study of relevance for learning in deductive databases. *Journal of Logic Programming*, *40*, 215–249.

Nedellec, C., Rouveirol, C., Ade, H., Bergadano, F., & Tausend, B. (1996). *Advances in Inductive Logic Programming*, chapter Declarative Bias in Inductive Logic Programming, 82–103. Raedt de L. (ed.), IOS Press.

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

Rouveirol, C. (2000). Expressiveness/Efficiency: the dilemma of the Inductive Logic Programming. LRI, Université Paris. HDR, in french.

Sebag, M., & Rouveirol, C. (1994). Induction of maximally general clauses consistent with integrity constraints. *Proceedings of the 4th International Workshop on Inductive Logic Programming* (pp. 195–216). Gesellschaft für Mathematik und Datenverarbeitung MBH.

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

Srinivasan, A., Muggleton, S., King, R., & Sternberg, M. (1994). Mutagenesis: ILP experiments in a non-determinate biological domain. *Proceedings of the 4th International Workshop on Inductive Logic Programming* (pp. 217–232). Gesellschaft für Mathematik und Datenverarbeitung MBH.

Zucker, J.-D., & Ganascia, J.-G. (1996). Changes of representation for efficient learning in structural domains. *Proc. of 13th International Conference on Machine Learning*. Morgan Kaufmann.