Feature Subset Selection and Inductive Logic Programming

Erick Alphonse ALPHONSE@LRI.FR
Stan Matwin[1] STAN@LRI.FR
LRI - Bt 490
Universit Paris-Sud
91405 ORSAY CEDEX

Abstract

Applicability of ILP to real-world problems is constrained by the high dimensionality of ILP tasks. This paper proposes to reduce the dimensionality of the ILP example space by bringing feature subset selection to the realm of ILP. Seen as a black box, the method reduces ILP examples, in the form of non-recursive Datalog clauses, by removing literals judged irrelevant for the ILP task at hand. The approach exploits existing AVL feature selection (FS) algorithms by changing the representation from the FOL clausal format to a fixed-length AV format, performing FS on the AV approximation of the FOL task, and mapping the results back to the FOL representation. Issues of noise introduced in the data during representation shift are presented, and constraints on the AVL FS to be used in ILP FS are discussed.

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. From the efficiency perspective, one of the main difficulties that prevents ILP from tackling large-size problems is the dimensionality of the learning task, significantly higher than that of attribute-value learning (AVL).

There is no generally accepted theoretical measure of the dimensionality of ILP problems (Fürnkranz, 1997), 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 solution to the overgrown hypothesis space is the idea of a declarative bias. Traditionally in ILP, declarative (static) biases are 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: either the hypothesis language can become too restrictive to express the target concept, or too general so that their constraining effect becomes insignificant. 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, but as the size of the examples in terms of the number of literals by which 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, the presence of irrelevant literals in the examples description may mislead the heuristic search by showing plateau phenomena. 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 successfully 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

[1]On leave from University of Ottawa, Canada

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 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 put forward 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. (Lavrač et al., 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 applied in a straightforward manner. On the other hand, DHDB is too limited to handle current ILP benchmark datasets like mutagenesis (sect. 5) or relational databases.

Our approach to FS in ILP is more general: it works with relational examples that can be expressed as non-recursive Datalog Horn 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 approximating the original FOL problem 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 this 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 the change of representation (a so-called multi-instance propositionalization). In section 4, we discuss some problems facing our approach to example reduction, and illustrate them on a simple 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 limitations of the current approach.

2. Overview of the Architecture

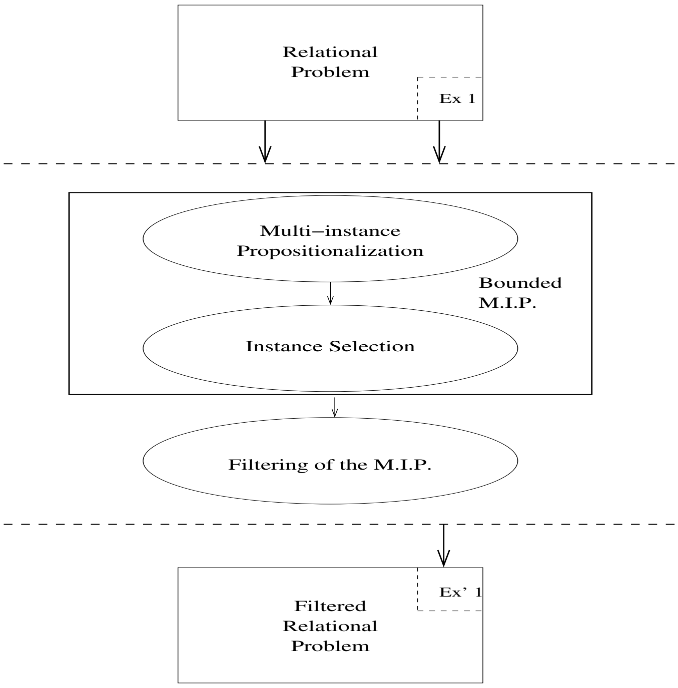

Figure 1. Filtering a relational problem

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 use the same representation language, and the learning system works on the output of the FS process. 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, the idea is to filter the literals of one FOL example at a time. As shown in Fig. 1, we use the set of literals of the example being filtered as a fixed set of attributes, and re-describe the whole relational problem by means of these attributes. Our change of representation converts each relational example into a set of AV examples. This makes us view the propositional representation as a multi-instance problem: a bag of

AV examples is obtained from a single relational example, and it inherits the label of its relational source.

This so-called multi-instance propositionalization (MIP) has been investigated by several researchers (Zucker & Ganascia, 1996; Alphonse & Rouveirol, 2000). In MIP, each ILP example is transformed into a set of fixed-length vectors of AVL attributes. The vector format has a boolean attribute for each literal of the example being filtered. Each boolean attribute describes the fact that its corresponding literal matches (under some substitution) a literal of the transformed ILP example.

Since the coverage test in FOL is NP-complete, the number of matchings 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 matchings, and work with a limited number of such examples. All the vectors resulting from the reformulation of the original relational problem are then given to an AVL feature selection system, engineered to respect constraints imposed by the special characteristics of the new instance space.

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. We reduce the literals judged irrelevant and obtain a filtered version of the relational example. To filter the whole relational problem, we iterate the process taking each example in turn.

3. Multi-instance Propositionalization

We address learning where examples are non-recursive Datalog Horn 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, that generalizes or subsumes all positive examples and does not subsume any negative examples. In this setting, the logical implication is equivalent to θ-subsumption (Gottlob, 1987), which is decidable but NP-complete.

Following (Zucker & Ganascia, 1998), a relational problem can decomposed into two sub-problems: a structural problem and a functional one. As the first attempt to bring feature subset selection to ILP, we restrict ourselves to the structural part of the relational learning as defined at the beginning of this section. This framework is concerned with discrimination by predicate occurrence(s) and variable links between variables, even though the multi-instance propositionalization is general enough to address both learning problems (Zucker & Ganascia, 1998; Sebag & Rouveirol, 1997). The structural problem is 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. For the lack of space, we present in the sequel only those aspects of MIP that are necessary to understand the proposed paradigm for FS in ILP.

The approach initializes P, the pattern of propositionalization, as one of the FOL examples, and the fixed set of literals of P is used as a fixed set of attributes to re-describe 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 re-described representation says whether or not there is a match (true) between this attribute (a literal of P) and some literal of e. In our representation language, there are many different ways to match a given literal and therefore e will be re-described 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 exemplified 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),
                  car(t,c3),short(c3),load(c3,l31),hex(l31).
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 re-describe the whole dataset, we use E' to construct the pattern P. It is built as the following variabilization of E' (omitting the head):

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

We can observe that since in the structural problem constant names are local within clauses (so called Skolem constants), it does not matter whether we work with skolemized clauses, or variabilized clauses so that links between literals are preserved in the variabilization. Given P, we build the new instance space showed in Table 1. 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 and E's literals, the next valid substitution computed is another matching for $car(V, Y)$: $\sigma_{E,2} = \{V/t, W/c1, X/l11, Y/c2\}$. For this substitution, neither the literals $load(Y, Z)$, $short(W)$, $short(Y)$ and $circ(Z)$ nor the variable Z have been matched. Again, another matching will yields $\sigma_{E,i} = \{V/t, W/c2, Y/c3, Z/l31\}$. In our example, six other substitutions (and therefore six other boolean vectors) are obtained when searching for all 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 1).

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., 1997), formerly called a multi-part problem: 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 generalize the equivalence theorem (Zucker & Ganascia, 1996) to our framework:

Theorem 1 *Given P (a formula in the hypothesis language), searching for a solution which is a subset of P (2^P, up to variable renaming) is equivalent to solving the multi-instance problem, obtained from multi-instance propositionalization with P as pattern.*

4. Feature Selection in ILP

In the proposed approach, feature selection is performed by first reformulating the relational problem, and then applying feature subset selection techniques. Consequently, the filter methods used for feature selection must perform well in the new instance space. 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. We focus only on the noise artificially generated by the approximation and not the noise present in the original data. To begin with, the reformulated multi-instance representation of the problem could be seen as a class-noisy representation of the positive data (Blum & Kalai, 1998). Indeed, each positive FOL example is represented by a set of vectors, such that covering only one of them is sufficient to cover the FOL example. However, this problem has been well studied in the ILP community and relaxing the completeness of top-down algorithms is typically applied (Zucker & Ganascia, 1996).

Therefore we focus instead on other sources of noise which are more closely related to our approach. In order to illustrate the impact of noise, we upgraded the boolean-attribute filter FOCUS (Almuallim & Dietterich, 1994) (by relaxing completeness) and performed simple experiments on a train-like artificial problem similar to the one in Sect. 3. FOCUS uses a top-down exhaustive search in the attribute space and is not noise-resistant. Two datasets have been generated, one with a conjunctive target concept , and the other with a disjunctive one.

Below we discuss the impact of noise due to the change of representation for the two kinds of concepts: the conjunctive and the disjunctive ones.

4.1 The Conjunctive 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 an example. As showed in the first row of Table 2, FOCUS finds all relevant literals and only those literals. This behavior is a consequence of theorem 1, as in the learning by implication paradigm, the conjunction of the minimal set of relevant attributes is also a solution of the concept learning task.

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	0	0	0
...								
$\sigma_{E,i}$	1	0	0	0	1	1	1	0
...								
$\sigma_{NE,1}$	1	0	1	0	1	0	1	0
$\sigma_{NE,2}$	1	0	1	0	1	1	1	1
...								
$\sigma_{NE,j}$	1	0	1	1	1	1	1	1
...								

However, in the general case, as pointed out by (Sebag & Rouveirol, 1997), 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 Sect. 5, the potential number of matchings is 40^{40}. But this set is highly redundant and only few vectors are sufficient to represent the whole instance space. Indeed, this set is known to be the set of the most specific vectors in the boolean lattice-like instance space: the nearest-hits and nearest-misses of the pattern (Alphonse & Rouveirol, 2000). A trivial upper-bound of its size is the maximal number of incomparable elements in the above instance space which is known to be $\binom{n}{\lfloor n/2 \rfloor}$, with $n = |P|$. Therefore, instead of dealing with a vector space of size exponential in both $|e|$ and $|P|$, we have a space exponential only in $|P|$ (2^{40} instead of 40^{40}). Algorithms can be designed to approximate this set of non-redundant vectors in order to cope with the intractability of the MIP. 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. Such 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. The resulting noise in the approximating set of vectors is particular in the sense that it is generated artificially by the bounded MIP, inherent to the proposed paradigm. It is strongly related to the quality of the heuristic used to extract as specific vectors as possible. A natural question arises: as the noise is generated by the approximating algorithm, can the amount of so introduced noise be estimated? Unfortunately, in the general case, when we approximate the non-redundant set, the answer is negative. This is due to a negative result concerning the so-called Max-Compatible-Binary-Constraint-Satisfaction problem (equivalent to the problem of computing the exact set of non-redundant elements) that states that we can not approximate the solution within any constant unless $P = NP$ (Jagota, 93). Further investigation could focus on weakening the representation language in order to lift this limitation in the general case (e.g. k-local (Cohen, 1993)).

For now, we have only investigated a very simple bounded MIP scheme, following the idea of (Sebag & Rouveirol, 1997) of sampling the space of matchings. We apply a stochastic process where k matchings are selected to yield a bounded reformulated problem, with k being a user-supplied parameter. Row 2 in Table 2 shows the FOCUS' performance degrading on the approximated problem.

4.2 The Disjunctive Learning Case

In this case, the pattern and the positive example being reformulated may not belong to the same subconcept. That is, some literals discriminate between the sub-concept of P and the sub-concept of the example. These literals, used as attributes to describe the several reformulations of the example will not be matched. Therefore the MIP will produce irrelevant[2] vectors which do not have to be taken into account. In this manner, propositionalization introduces class-

[2]This case is exemplified by the FOL learning problem described in Sect. 3. It can be seen that a conjunctive solution does not exist in the Datalog language. All vectors obtained by propositionalizing the positive FOL example are more general than the negative vector $\sigma_{NE,j}$, and therefore should really be negative in our learning setting.

noise for these particular positive examples which are tagged positive in the training set but which really are negative examples with respect to the pattern. Third row of Table 2 represents this case of noise due to the disjunctive character of the target concept. We can observe that the performance worsens in both measures.

Following the above discussion, we have proposed in (Alphonse & Matwin, 2002) a Relief-like algorithm which is not parameterized with a noise level, but removes fewer literals as the level of noise increases. This is exemplified at the last row of Table 2.

	relevant literals (%)	mean reduction (%)
conj-exhaustive	100	90
conj-bounded	80	90
disj-bounded 1	25	87.8
disj-bounded 2	100	61

Table 2. Filtering on an artificial problem

5. Experiments

In order to validate the proposed paradigm to bring FSS to the realm of ILP, we used a simple instantiation of it and evaluated the training time and the accuracy twice: prior to applying the filter, and after this application. The bounded MIP is performed by sampling the matching space with remplacement, as in Sect. 4.1, and the filter algorithm is the one described in detail in (Alphonse & Matwin, 2002).
We considered several learning systems. It is interesting to see that not all state-of-the-art ILP learning systems can take advantage of the filtered example space in a straightforward manner. First of all, Progol and Aleph (in its default setting) perform an exhaustive search in a bounded concept space (namely a A^* search procedure). Therefore, they are going to find the best concept definition containing only relevant literals, according to their quality measure. An effective gain in time can be realized, and it can be traded-off for accuracy by enlarging the bounded hypothesis space, but we leave investigation of this possibility for future research. Secondly, ILP systems based on interpretation or on the closed-world assumption, e.g. Tilde and FOIL, respectively, rely on a complete description of the examples (de Raedt, 1997). Therefore, these systems can not take advantage of the reduced example space. In our experiments, we have used FOIL 6.4 (Quinlan, 1990) with the CWA turned off, and all other parameters with default values, and PROPAL (Alphonse & Rouveirol, 2000), an AQ-like learning system.

We have evaluated the approach by performing experiments on five tasks from two real-world domains: two from the mutagenesis domain (Srinivasan et al., 1995), a well-known ILP problems used as benchmark tests, and three from the Document Understanding (DU) domain (Esposito et al., 1994). In the mutagenesis domain, the representation language used has been defined from background knowledge B_1, which uses only relational literals, tackling nominal and ordinal arguments as constants.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., 1995). 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. In the DU domain, we work with three tasks: Receiver, Date, and Sender. We used the experimental protocol provided in (Esposito et al., 1994). There are some thirty positive and one-hundred and fifty negative examples in each task. Each example contains approximately 150 literals. In the FS literature, a filtering task involving 150 features is not considered a trivial one.

Table 3 compares the performance of FOIL and Propal on the filtered and not filtered datasets. Each learning time is calculated by performing learning on the whole dataset. In order to include the filter in the validation process, we filtered only instances used for training, for all validation procedures. 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.

In the mutagenesis domain, as expected, the performance of the two learning systems 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. Also, not only FOIL's accuracy has slightly improved, but its running time has decreased almost by a factor of 10. The deterioration of the performance for the case with $k = 500$ is most likely due to the increase of noise by working with a larger sample of the noisy instance

Table 3. Comparison of the performance of FOIL 6.4 and PROPAL on the original datasets and the filtered ones

		FOIL			PROPAL		
		$k = 100$	$k = 500$	unfiltered	$k = 100$	$k = 500$	unfiltered
RF	Accuracy (%)	78.09$\pm$0.34	78.25$\pm$0.15	75.8	85.54$\pm$1.3	85.50$\pm$1.73	81.80
	Time (s.)	496$\pm$1.5	515$\pm$52	4655	69$\pm$26	110$\pm$1	290
RU	Accuracy (%)	-	-	-	80.71$\pm$1.75	82.61$\pm$3.73	71.4
	Time (s.)	-	-	-	6.34$\pm$1.61	5.96$\pm$0.85	10
Receiver	Accuracy (%)	98.5$\pm$0.0	98.5$\pm$0.0	98.16	98.6 $\pm$0.0	98.6$\pm$0.0	97.43
	Time (s.)	< 1	< 1	< 1	< 1	< 1	< 1
Date	Accuracy (%)	98.8$\pm$0.0	98.8$\pm$0.0	98.6	99.33 $\pm$0.0	99.33$\pm$0.0	99.33
	Time (s.)	< 1	< 1	< 1	19.26$\pm$0.35	19.36$\pm$0.57	21.5
Sender	Accuracy (%)	97.36$\pm$0.0	97.36$\pm$0.0	97.36	92.74 $\pm$0.0	92.74$\pm$0.0	94.24
	Time (s.)	< 1	< 1	< 1	< 1	< 1	2.85

space.
In the DU domain, instead of improving an already high performance on the original data, the proposed method does not seem to remove relevants literals which would degrade performance. The results may even improve slightly, while the PROPAL execution time for the two more difficult tasks decreases. This experiment shows that in domains the original training set yields high accuracy performance and its size is limited, the proposed filter is accuracy-neutral but it improves the training time.

The values of k in these experiments are arbitrary. A judicious choice of k needs to be studied further.

6. Limitations of the Current Approach

In the current approach, the input and output of a ILP FS algorithm use the same representation language. Surprisingly, this has some shortcomings, which limit the generality of the paradigm. On the one hand, the application of closed-world assumption (CWA) in some systems (e.g. FOIL) relies on the complete description of the examples to generate negative literals. Inherently, filtering examples will mislead the learning system which will generate the negation of the literals detected empirically irrelevant. A solution would be either to code both the complete description of each example and the relevant literals, or, to move to the learning by implication paradigm and encode negated literals of interest in the background knowledge. On the other hand, by filtering out some predicates, links between variables can be lost. In the DU domain we address this problem by slightly modifying the representation so that all literals are linked to the head. Let us observe that this does not happen in problems similar to the mutagenesis domain. A trivial way to deal with that problem would be to put in the final representation of the filtered dataset the relevant literals together with their links. However, as this seems to favor the bottom-up approach, we will investigate another approach, based on feature construction. The newly constructed feature could represent by a single predicate a group of linked predicates. The learning could then proceed by saturation, although this could turn out to be expensive.
Finally, as we work in the learning by implication paradigm, we didn't investigate filtering negative examples, as the more specific, the more informative they are. The extension of the filtering process to negative examples is an interesting follow-up, but seems to rely intensively on a good filter.

7. Conclusion

We have presented here the first paradigm to apply, in the ILP context, the feature subset selection approach to reduce the dimensionality of the example space, in languages at least as expressive as non-recursive Datalog Horn clauses. It bridges the gap between the flexible representation of instances in the form of FOL clauses, and the fixed-format requirement of the FS filters. In this paradigm, the feature selection techniques work as a front end filter, applied prior to further processing (e.g. (Kramer & De Raedt, 2001)) and to the model building. Moreover, our method is applicable to any partial ordering of the learning space.
In order to empirically evaluate the relevance of the approach, we have proposed a simple instantiation and performed experiments on several tasks from two real-world domains: Mutagenesis, a biochemical domain, considered benchmark in ILP, and the Document Understanding domain. The preliminary results are very encouraging, showing, as expected, an improvement both in time and accuracy, demonstrating the applicability of FS in ILP.

Our approach is based on the approximation of the original ILP problem by a bounded MIP. Unfortunately, as discussed in Sect. 4.1, there is no algorithm

that in polynomial time would guarantee the quality of the approximation. Therefore we plan to further investigate stochastic sampling techniques, e.g. GRASP (Feo & Resende, 1995). An interesting complementary approach would be to investigate biases providing either polynomial-space complexity of MIP (e.g. k-local (Cohen, 1993)) or efficient approximation schemes.
Another consequence of the approximation is the introduction of noise in the data. As far as we know no research has been done on noise-resistant MIP filters, and further work needs to be done in this area.
The method generalizes previous work on FSS in ILP (Lavrač et al., 1999) which is viewed as the special case of our method when only one matching substitution is allowed between the pattern and the examples.

We plan to further investigate the validity of the approach on problems presented by the ML community, still considered out of the scope of current ILP techniques. Specifically, it is interesting to observe that feature selection may shift the localization of ILP problems wrt the phase transition region (Serra et al., 2001).

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, the Computing and Information Technologies Ontario and le Centre National de la Recherche Scientifique are kindly acknowledged.

References

Almuallim, H., & Dietterich, T. G. (1994). Learning Boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, *69*, 279–305.

Alphonse, E., & Matwin, S. (2002). A dynamic approach to dimensionality reduction in relational learning. *XIII. Int. Symposium on Methodologies for Intelligent Systems (to appear)*.

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.

Cohen, W. W. (1993). Learnability of restricted logic programs. *Proceedings of the 3rd International Workshop on Inductive Logic Programming* (pp. 41–72). J. Stefan Institute.

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

Dietterich, T. G., Lathrop, R. H., & Lozano-Pérez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, *89*, 31–71.

Esposito, F., Malerba, D., & Semeraro, G. (1994). Multistrategy learning for document recognition. *Applied Artificial Intelligence*, *8*, 33–84.

Feo, T. A., & Resende, M. G. C. (1995). Greedy randomized adaptive search procedures. *Journal of Global Optimization*, *6*, 109–133.

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).

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

Jagota, A. (93). Constraint satisfaction and maximum clique. In *Working notes, aaai spring symposium on AI and NP-hard problems*, 92–97. Stanford University.

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

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

Kramer, S., & De Raedt, L. (2001). Feature construction with version spaces for biochemical applications. *Proc. 18th International Conf. on Machine Learning* (pp. 258–265). Morgan Kaufmann, San Francisco, CA.

Lavrač, N., Gamberger, D., & 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.

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

Sebag, M., & Rouveirol, C. (1997). Tractable induction and classification in first order logic via stochastic matching. *15th Int. Join Conf. on Artificial Intelligence (IJCAI'97)* (pp. 888–893). Morgan Kaufmann.

Serra, A., Giordana, A., & Saitta, L. (2001). Learning on the phase transition edge. *Proceedings of the 7th Int. Conf. on Artificial Intelligence (IJCAI-01)* (pp. 921–926). Morgan Kaufmann Publishers, Inc.

Srinivasan, A., Muggleton, S., & King, R. (1995). Comparing the use of background knowledge by inductive logic programming systems. *Proc, of the 5th ILP* (pp. 199–230). Scientific Report, K.U.Leuven.

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

Zucker, J.-D., & Ganascia, J.-G. (1998). Learning structural indeterminate clauses. *Proc. of the 8th International Workshop on Inductive Logic Programming* (pp. 235–244). Springer Verlag.