# Bias Windowing for Relational Learning

**Frédéric Koriche** [1]

**Abstract.** A central issue in relational learning is the choice of an appropriate bias for limiting first-order induction. The purpose of this study is to circumvent this issue within a uniform framework inspired from the paradigm of windowing. A *bias window* is a restricted subclass of the relational space determined by some parameters. The idea is to learn a theory in a small window first, and iteratively adjust the window in order to find the optimal bias from which to choose the final theory. To this end, our model integrates a logical notion of window-based induction, a learning algorithm that implements this mechanism, and a windowing technique that monitors the learning process using a metric-based criterion. Experiments on the Mutagenesis dataset show that, after a period of underfitting, windowing converges on hypotheses which are stable and very effective.

## 1 INTRODUCTION

The relational learning problem [11] seems to be caught between a rock and hard place. On the one hand, relational domains typically involve multiple objects and relationships between them. To this end, first-order logic provides a very expressive language which enables the learner to induce structural patterns in the observed sample, and to represent this knowledge into a compact form. On the other hand, first-order induction is very much demanding from a computational point of view. Even in the finite, function-free case, the learner is confronted with hypothesis spaces which are generally much larger than concept classes usually addressed in propositional learning.

This dilemma is exacerbated still further by the statistical evidence that induction in large hypothesis spaces can substantially reduce both the accuracy and stability of classifiers. As explained by the Bernoulli's theorem in [18], the difference between the empirical error made by the learner on the training set and the generalization error measured on a separated set of test data grows with the size of the concept class. This stems from the fact that large spaces contain many descriptions that behave similarly on the observed sample and yet behave quite differently in larger populations, thus diminishing the ability to distinguish relevant hypotheses from irrelevant ones.

In the ILP literature, two main approaches have been investigated to handle these issues: the *heuristic-based* approach and the *representation-based* approach. As to the former, the idea is to limit exploration in hypothesis spaces by using strong procedural biases. The blueprint is the top-down greedy search algorithm governed by appropriate heuristics. However, although greedy-based learning is very efficient, it is generally bound to miss relevant theories, especially in relational domains that lie in the phase transition [8]. Several strategies have been proposed to address this myopic limitation, including pruning methods [6] and genetic techniques [3]. Yet, these strategies clearly cannot scale up beyond a limited context.

In the later approach, the idea is to restrict the dimension of the space in order to obtain tractable forms of induction. For this purpose, a wide variety of representation biases have been proposed, and range from standard syntactic parameters to complex grammars that model relational languages [5]. Based on a bias that efficiently narrows the hypothesis space for a given domain, the learner can perform a systematic search in this space. Recent experiments have shown that the resulting theories are often stable and effective [2]. However, a key issue, which is often deferred to the user, is to select the "appropriate" representation bias for the problem. Since different domains typically require different biases, it is important to make further steps in the direction to automated methods of bias.

In this setting, the purpose of the present study is to investigate the paradigm of *windowing*, a technique primarily due to Quinlan [14] and which has recently been generalized by Fürnkranz in [7]. The main concern of this paradigm is to provide a trade-off between efficiency and effectiveness by enabling the learner to concentrate on different subparts of its data and/or hypothesis space. Regardless of the specificity of the search space, the idea is to maintain a subclass of this space, the so-called *window*, from which a theory is learned. If the quality is estimated insufficient, the window is adjusted by increasing the search space, and a new theory is learned.

Most existing approaches to windowing are *data-oriented* and aim at reducing the size of large databases. Yet, windowing may also be *bias-oriented* by reducing hypothesis spaces. Actually, CLINT [12] and NINA [1] have adopted a similar idea by enabling the learner to shift its bias from a predefined sequence of languages. Preliminary results are encouraging and state that using a sequence of biases and shifting the bias can be more economic and effective than learning in the union of the languages in the sequence.

In this paper, we develop a model of bias windowing for relational learning. The basic building block lies in the notion of *bias window*, which determines a restricted subclass of the relational space. Based on this notion, the model integrates three key components:

- A logical notion of robust induction which enables the learner to infer a theory given *any* bias window available. In particular, the learner may induce hypotheses even if they are not perfectly consistent with the training examples.
- A learning algorithm that attempts to find, in time polynomially bounded by the bias parameters, a solution in the window which is as consistent as possible with the examples.
- A windowing algorithm that monitors the learning progress. The method uses a metric-based selection principle, inspired from [15], which attempts to identify the best window for the domain.

The paper is organized as follows. In section 2, we introduce the logical part of the model. Sections 3 and 4 are devoted to the algorithmic aspects. Experiments on the mutagenesis dataset are reported in section 5. Finally section 6 concludes the paper.

[1] LIRMM, 161 rue Ada 34392 Montpellier Cedex 5, France email: koriche@lirmm.fr

## 2 BIAS WINDOWS

This section introduces the logical setting of windowing. As stated earlier, the key component lies in the notion of bias window which captures the hypothesis class available to the learner. For sake of generality, we need a corresponding form of induction that enables the learner to infer theories given any bias window available. The main motivation here is to allow the learner to robustly induce hypotheses which, albeit not necessarily consistent with the training data, can capture enough important patterns to be accurate on test data. We thus begin to review the standard notion of induction, and then, we extend this notion to robust induction.

### 2.1 Basic Terminology

Our framework basically addresses two-class learning problems. In this setting, relational theories are usually represented in first-order DNF, i.e., disjunction of conjunctive formulas.

In this paper, a *relational vocabulary* consists in a set of predicates and a set of constants. For sake of simplicity, we shall assume throughout that the sets of predicates and constants, and all the arities are finite. Furthermore, we suppose that the maximum of the arities in the set of predicates is fixed. A term is a variable or a constant. An atom is a predicate whose arguments are terms. A literal is an atom or its negation. A *rule* is an existentially quantified conjunction of literals and a *hypothesis* is a disjunction of rules. In following, we represent rules as sets of literals and hypotheses as sets of rules.

Now, we need to formalize the notion of bias window. As stated earlier, several bias languages have been proposed in the literature. In this paper, we exploit a simple bias scheme inspired from the so-called $xk$-DNF concept class introduced by Valiant in [16]. A *bias window* consists in a pair $[x, k]$ where $x$ and $k$ are positive integers. The *window space* of $[x, k]$, denoted $\mathcal{H}_{xk}$, is the space of all hypotheses composed of rules which contain at most $x$ distinct variables per rule and at most $k$ literals per rule. In the following, any rule generated from the window bias $[x, y]$ is called a *$xk$-rule*. Intuitively, the parameter $x$ is used to limit the complexity of the covering test, while $k$ is used to reduce the dimension of the relational space.

### 2.2 Standard Induction

Various formalizations of relational induction have been proposed in the ILP setting. Our framework uses the learning from interpretations principle. An expression is called ground if it does not contain any occurrence of variable. A ground substitution is a mapping from variables to constants. An *interpretation* is a set of ground atoms. Given a ground substitution $\gamma$, an instance $A\gamma$ of an atom $A$ is true in an interpretation $I$ if $A\gamma \in I$. A negative literal $\neg A\gamma$ is true in $I$ is $A\gamma \notin I$. A rule $R$ is true in $I$ if there exists a ground substitution $\gamma$ such that all literals in $R\gamma$ are true in $I$. Finally, a hypothesis $H$ is true in $I$ if there exists a rule $R \in H$ such that $R$ is true in $I$.

Let $\mathcal{I}$ denote the set of all possible interpretations generated from the vocabulary. A *classifier* is an assignment from $\mathcal{I}$ to $\{0, 1\}$, where $0$ is the negative class and $1$ the positive class. Any hypothesis $H$ can be extended to a classifier, also denoted $H$, such that $H(I) = 1$ iff $I$ is true in $H$. An *example* is a pair $e = (I_e, c_e)$ where $I_e$ is an interpretation and $c_e \in \{0, 1\}$. An example $e$ is called positive if $c_e = 1$ and negative otherwise.

**Definition 1.** *Given a set of examples $E$ and a bias window $[x, k]$, the* standard version space *of $E$ w.r.t. $[x, k]$ is given by:*

$$\mathcal{H}_{xk}(E) = \{H \in \mathcal{H}_{xk} : H(I_e) = c_e \textit{ for every } e \in E\}.$$

A bias window $[x, k]$ is called *consistent* with a sample $E$ if $\mathcal{H}_{xk}(E)$ is nonempty, and *inconsistent* otherwise. The consistency criterion cannot always be guaranteed especially in presence of strong representation biases. In fact, if $[x, k]$ is inconsistent with $E$, then the version space collapses and induction fails into triviality. So, we need to make appropriate formal steps in this direction.

### 2.3 Robust Induction

In presence of inconsistency, we need to introduce a metric in the hypothesis space in order to select the theories which are as consistent as possible with the observed sample of examples given the available representation bias. To this end, the standard empirical error measure meets our requirements. Given a hypothesis $H$ and a training set $E$, the distance between $H$ and $E$ is defined by:

$$d(H, E) = \frac{|\{e \in E : H(I_e) \neq c_e\}|}{|E|}$$

It can be shown that this measure induces a total pre-ordering $\leq_E$ between hypotheses, where $H \leq_E H'$ iff $d(H, E) \leq d(H', E)$. Thus, the aim of robust induction is to retain those hypotheses in the window space which are minimal according to $\leq_E$.

**Definition 2.** *Given a training set $E$ and a bias window $[x, k]$, the* robust version space *of $E$ w.r.t. $[x, k]$ is given by:*

$$\mathcal{H}^*_{xk}(E) = \min(\mathcal{H}_{xk}, \leq_E).$$

We now examine several semantical properties that clarify the interest of this model in the setting of windowing. In the following result, the first property assures that a robust version-space is always nonempty and well-defined. The second property suggests that, if possible, the result of robust induction is simply the standard version space. The third property embodies a *quasi-decomposability* principle which is particularly useful for covering algorithms. Intuitively, if we could find two subgroups of examples which agree on at least one hypothesis, then the result of robust induction will be exactly those hypotheses the two groups agree on. The last property captures a notion of *quasi-monotonicity* which advocates the use of specific-to-general windowing techniques. Namely, it states that if two bias windows, a small one and a large one, agree on at least one hypothesis, then any solution returned by the small window is guaranteed to be a solution for the large window.

**Proposition 3.** *Let $E$, $F$, $G$ be sets of examples such that $F \cap G = \varnothing$ and $F \cup G = E$, and let $x$, $x_1$, $x_2$ and let $k$, $k_1$, $k_2$ be representation biases such that $x_1 \leq x_2$ and $k_1 \leq k_2$. Then window-based induction satisfies the following properties:*

1 $\varnothing \subset \mathcal{H}^*_{xk}(E) \subseteq \mathcal{H}_{xk}$
2 *If* $\mathcal{H}_{xk}(E) \neq \varnothing$ *then* $\mathcal{H}^*_{xk}(E) = \mathcal{H}_{xk}(E)$
3 *If* $\mathcal{H}^*_{xk}(F) \cap \mathcal{H}^*_{xk}(G) \neq \varnothing$ *then* $\mathcal{H}^*_{xk}(E) = \mathcal{H}^*_{xk}(F) \cap \mathcal{H}^*_{xk}(G)$
4 *If* $\mathcal{H}^*_{x_1k_1}(E) \cap \mathcal{H}^*_{x_2k_2}(E) \neq \varnothing$ *then* $\mathcal{H}^*_{x_1k_1}(E) \subseteq \mathcal{H}^*_{x_2k_2}(E)$

*Proof.* Property 1 follows from definition 2 and the fact that $\mathcal{H}_{xk}$ is never empty (even when $x = 0$ and $k = 0$). Property 2 stems from the fact that $d(H, E) = 0$ for any $H \in \mathcal{H}_{xk}(E)$. For property 3, we only prove that $\mathcal{H}^*_{xk}(F) \cap \mathcal{H}^*_{xk}(G) \subseteq \mathcal{H}^*_{xk}(E)$ since an analogue strategy holds for the dual part. Suppose that $H \in LHS$ and $H \notin RHS$. Then, there must exist $H' \in \mathcal{H}^*_{xk}(E)$ such that $d(H', E) < d(H, E)$. Since $d(H', E) = d(H', F) + d(H', G)$ it follows that either $d(H', F) < d(H, F)$ or $d(H', G) < d(H, G)$. In both cases, this contradicts the initial assumption. Finally, for property 4, let $H \in \mathcal{H}^*_{x_1k_1}(E) \cap \mathcal{H}^*_{x_2k_2}(E)$. Then, for any $H' \in \mathcal{H}^*_{x_1k_1}(E)$, we have $d(H', E) = d(H, E)$. Hence, it follows that $H' \in \mathcal{H}^*_{x_2k_2}(E)$. □

## 3 WINDOW-BASED LEARNING

In the previous section, we examined window-based induction at the logical level: a process that determines the set of possible theories given the available data and bias. In this section, we examine this notion at the algorithmic level. The problem can be formulated as follows: given a sample $E$ and a bias window $[x, k]$, find a hypothesis $H$ such that $d(H, E)$ is minimal. Interestingly, this problem is closely related to the *agnostic learning* [9] issue, in which no assumption is made on the target function. An important point is that, for most interesting classes, agnostic learning is known to be intractable. Even the class of monotone monomials is not efficiently learnable unless RP = NP. Using similar arguments (i.e. reduction to "Set-Cover"), the window-based learning problem can be shown NP-hard.

This computational barrier incites us to seek for *approximation algorithms* that run in polynomial time and yet guarantee a bound on the suboptimal solution for the problem. In light of this approach, we develop an approximation algorithm for the window-based learning problem which can be seen as a generalization of the greedy cover method used to find the simplest theories in Occam learning.

To this end, we need additional definitions. Let $P(E)$ ($N(E)$) be the set of positive (negative) examples which occur in the sample $E$. Given a rule $R$, let $P(R, E)$ ($N(R, E)$) be the set of all positive (negative) examples $e$ in $E$ such that $R$ is true in $I_e$. Given a bias window $[x, k]$ and a training set $E$, the *positive cost* $\alpha$ of $[x, k]$ w.r.t. $E$ is the maximum number of positives $|P(R, E)|$ covered by any $xk$-rule $R$. The *negative cost* $\beta$ of $[x, k]$ w.r.t. $E$ is the maximum ratio of negatives $|N(R_1, E) \cap N(R_2, E)|/|E|$ which are mutually covered by two distinct $xk$-rules $R_1$ and $R_2$.

The algorithm is shown in figure 1. It is important to remark here that the method performs a greedy search in the space of $xk$-hypotheses, yet a systematic search in the space of $xk$-rules. Despite its apparent simplicity, the algorithm embodies the property that it tends to approximate the optimal error to within a logarithmic factor *plus* an additional inclusion-exclusion penalty.

**Theorem 4.** *Let $[x, k]$ be a bias window, let $E$ be a set of examples, and let $\alpha$ and $\beta$ be the positive cost and the negative cost of $[x, k]$ w.r.t. $E$. Now, let $G$ be the hypothesis returned by* $\textsc{Learn}(E, x, k)$. *Then for any hypothesis $H \in \mathcal{H}^*_{xk}(E)$*

$$d(G, E) \leq \ln(\alpha) \left( d(H, E) + \beta^{\left(\frac{1}{\sqrt{|H|}\log(|H|)}\right)} \right)$$

*Proof.* The demonstration closely follows the "weighted set cover approximation" proof [4], with two important variations: (1) the covers can be incomplete and (2) the weights can be dynamic. Part 1 is handled using a simple completion method. Part 2 is circumvented using an inclusion-exclusion approximation technique [13].

We assume here that $E$ is clear from the context. Thus, $P(E)$ is abbreviated as $P$, $P(R)$ is abbreviated as $R$, and so on. Let $\mathcal{R}$ be the smallest set defined by the following conditions: (1) any $xk$-rule $R$ is an element of $\mathcal{R}$ and (2) any subset $S$ of $P$ is an element of $\mathcal{R}$. By definition, we set $P(S) = S$ and $N(S) = S$.

Now, let $G = \{R_1, \cdots, R_g\}$ be the solution found by the greedy algorithm and let $G^* = G \cup \{R_{g+1}\}$ be an extension of $G$ such that $R_{g+1}$ is the set of positives not covered by $G$. For each positive $e$ in $P$, let $\tau(e)$ be the iteration where $e$ is covered the first time. Each positive is assigned a cost only once, when it is covered the first time. Let $c(e)$ be the cost of $e$. If $e$ is covered the first time by $R_{\tau(e)}$ then

$$c(e) = \frac{|N(R_{\tau(e)}) - (N(R_1) \cup \cdots \cup N(R_{\tau(e)-1}))|}{|P(R_{\tau(e)}) - (P(R_1) \cup \cdots \cup P(R_{\tau(e)-1}))|}.$$

```
Input: A training set $E$ and a window bias $[x, k]$.
1. set $H = \varnothing$;
2. if $P(E) = \varnothing$ then goto step 5;
3. find a $xk$-rule $R$ that minimizes the quotient $\frac{|N(R,E)|}{|P(R,E)|}$; in case of tie, take $R$ which maximizes $|P(R, E)|$;
4. if $|N(R, E)| < |P(R, E)|$ then set $H = H \cup \{R\}$, set $E = E - (P(R, E) \cup N(R, E))$ and return to step 2;
5. return $H$;
```

**Figure 1:** $\textsc{Learn}(E, x, k)$

Let $R$ be any member of $\mathcal{R}$ and let $t = \max\{\tau(e) : e \in P(R)\}$, be the iteration when the last positive of $R$ is covered by the algorithm. Let $p_i = |P(R) - (P(R_1) \cup \cdots \cup P(R_{i-1}))|$. We have

$$\begin{aligned}
\sum_{e \in P(R)} c(e) &= \sum_{i=1}^{t} \sum \{c(e) : e \in P(e), \tau(e) = i\} \\
&= \sum_{i=1}^{t} \frac{|N(R_i) - (N(R_1) \cup \cdots \cup N(R_{i-1}))|}{|P(R_i) - (P(R_1) \cup \cdots \cup P(R_{i-1}))|} (p_i - p_{i+1}) \\
&\leq |N(R)| \sum_{i=1}^{t} \frac{p_i - p_{i+1}}{p_i} \\
&\leq |N(R)| \ln(\alpha)
\end{aligned}$$

Notice that the first inequality arises from the heuristic of the greedy algorithm chosen in step 3. Now let $H = \{R'_1, \cdots, R'_h\}$ be an optimal solution and let $H^* = H \cup \{R_{h+1}\}$ be an extension of $H$ such that $R_{h+1}$ is the set of positives not covered by $H$. We have

$$\begin{aligned}
d(G, E) &= \sum_{i=1}^{g+1} \frac{|N(R_i) - (N(R_1) \cup \cdots \cup N(R_{i-1}))|}{|E|} \\
&= \sum_{i=1}^{g+1} \sum \frac{\{c(e) : \tau(e) = i\}}{|E|} \\
&\leq \ln(\alpha) \sum_{i=1}^{h+1} \frac{|N(R'_i)|}{|E|} \\
&\leq \ln(\alpha) \left( d(H, E) + \beta^{\Omega\left(\frac{1}{\sqrt{h}\log(h)}\right)} \right)
\end{aligned}$$

The last inequality uses the inclusion-exclusion approximation technique. Notice that if $\beta$, the ratio of false positives shared between $xk$-rules, is small then the inclusion-exclusion term is close to zero. □

A second important aspect of this algorithm is that its complexity is polynomially bounded by the window parameters $x$ and $k$. Let $p$ be the total number of predicates, $c$ the total number of constants, and $a$ the maximum of the arities in the set of predicates. Let $m$ be the number of examples in $E$ and $g$ the maximal number of ground atoms in any example. Notice that $g$ is in $O(pc^a)$. For any $xk$-rule $R$ and any example $e$, one can test whether $R$ is true in $e$ by enumeration in time $O(kg^x)$. Moreover, the total number $r$ of $xk$-rules is bounded by $(p(x + c)^a)^k$. If we assume that $x$ is very small by comparison with $c$, then $r$ is in $O(g^k)$. Thus, using only $O(mg)$ space, step 3 requires at most $O(mrg^{x+k})$ time. Step 4 requires only $O(m)$ time. Finally, since there are at most $O(m)$ iterations in the main loop, the overall time bound is therefore $O(km^2g^{x+k})$.

# 4 WINDOWING

Based on the learning algorithm developed in the previous section, we now turn to the main windowing scheme. The idea is to start from a small window $[x_0, k_0]$ and to induce a hypothesis from this bias. We then adjust the window by modifying the parameters $x$ and $k$, and induce a new theory from this window. This process is iterated until the best current hypothesis is judged satisfying or a maximal bound $[x_n, k_n]$ is reached. In formal terms, the *window selection* problem can be stated as follows: given a set of examples and a collection of windows $[x_0, k_0] \leq [x_1, k_1] \leq \cdots \leq [x_n, k_n]$ organized in a lattice, identify the optimal window from which to choose the final theory.

This setting is the realm of *model selection* techniques used to find the optimal hypothesis class for a given problem. These methods can broadly being divided into three categories. *Data-oriented methods*, like cross-validation, use separate data to learn and validate hypotheses. Yet, they are often computationally intensive and reduce the available data for learning. *Complexity penalization methods* seek to avoid this problem by using the same data for training and validation, but penalize the hypotheses which are likely to overfit using a complexity parameter, such as the VC dimension. However, they typically produce overly broad bounds especially in relational learning. Finally, *metric-based methods* [15] lie in-between by taking advantage of the *unlabeled* examples, in order to introduce a complexity penalty. Since real-world databases typically contain large amounts of unlabeled data not used by supervised learners, this technique is worth to be investigated in relational learning.

In our framework, a metric-based method can easily be conceived by taking opportunity of the metric introduced in window spaces. Given two hypotheses $H$, $H'$ and a set of examples $E$, let

$$d_E(H, H') = \frac{|\{e \in E : H(I_e) \neq H'(I_e)\}|}{|E|}$$

Now, suppose we are given a training set $E$ and a set $U$ that contains $E$ *and* a nonempty set of unlabeled interpretations. Let $H_{xk}$ be the hypothesis induced by the learning algorithm on $E$ and $[x, k]$. The *adjusted distance* between $H_{xk}$ and $E$ (w.r.t. $U$) is defined by

$$\hat{d}(H_{xk}, E) = d(H_{xk}, E) \max_{[x_0,k_0] \leq [x_i,k_i] < [x,k]} \frac{d_U(H_{xk}, H_{x_i k_i})}{d_E(H_{xk}, H_{x_i k_i})}$$

Intuitively, the method attempts to penalize complex hypotheses which have an erratic behavior by comparison with simpler theories generated previously. The windowing algorithm, presented in figure 2, is based on this principle. It operates a lexicographic search in the lattice of windows and iteratively attempts to identify the best current theory using the notion of adjusted distance. The EXTRACT procedure implements the quasi-monotonicity property of window spaces. Given a hypothesis $H = \{R_1, \cdots, R_h\}$, the procedure returns the maximal subsequence $R_1, \cdots, R_i$ of rules which contain no false positive (i.e. $N(R_j, E) = \varnothing$, $1 \leq j \leq i$).

A key feature of metric-based selection is to provide a guarantee on the performance of the algorithm. Let $\epsilon(H)$ be the *generalization error* of $H$. Then, under some reasonable assumptions, the algorithm cannot overfit the optimal error by a factor much greater than 3.

**Theorem 5.** *Let $H_{xk}$ be the optimal theory in the sequence generated by the algorithm and let $H_{x'k'}$ be the hypothesis selected by the algorithm. If $[x, k] \leq [x', k']$ and $\hat{d}(H_{xk}, E) \leq \epsilon(H_{xk})$ then*

$$\epsilon(H_{xk}) \leq \left(2 + \frac{d(H_{x'k'}, E)}{d(H_{xk}, E)}\right) \epsilon(H_{x'k'})$$

*Proof.* By specialization of proposition 2 in [15]. □

```
Input: A training set $E$, an unlabeled set $U$, a minimal window
       $[x_0, k_0]$ and a maximal window $[x_n, k_n]$

1. set $[x, k] = [x_0, k_0]$, $E' = E$, $G' = \varnothing$ and $\hat{d}_{\min} = \infty$;
2. set $G = G' \cup \text{LEARN}(E', x, k)$ and $\hat{d}_g = \hat{d}(G, E)$;
3. set $G' = G' \cup \text{EXTRACT}(G)$ and $E' = E' - P(G')$;
4. if $\hat{d}_g < \hat{d}_{\min}$ then set $H = G$ and $\hat{d}_{\min} = \hat{d}_g$;
5. lexicographically increment $[x, k]$;
6. if $P(E') \neq \varnothing$ and $[x, k] \leq [x_n, k_n]$ then return to step 2;
7. return $H$;
```

**Figure 2:** Relational Bias Windowing

# 5 EXPERIMENTS

We have evaluated the windowing algorithm by performing experiments on the Mutagenesis dataset, a well-known ILP problem used as a benchmark test [10]. In the dataset, each example consists of a structural description of a molecule, and some numerical information describing its biochemical properties. The available data consists of 233 molecules of which 188 are “regression-friendly” and used for training and validation, and 45 are “regression-unfriendly” and generally not used by ILP learners. In our setting, the first pool is viewed as a labeled set of examples, which have to be classified into mutagenic and non-mutagenic ones, while the second pool is an unlabeled set examples used by the metric-based selection heuristic.

Four different sets of background knowledge have been identified for this problem and range from $B_1$ which uses only information on atoms and bonds to $B_4$ which involves high-level information on the molecules. We have focused on descriptions $B_2$, $B_3$ and $B_4$. For numerical data we have employed an “equal-width binning method” and for estimating predictive accuracy we have used the 10-fold cross-validation suggested by the authors.

Figure 3 reports the accuracy and run-time results obtained by the windowing algorithm. The times are measured on a Pentium IV 1GHz. For each experiment, the maximal window $[x_n, k_n]$ is progressively incremented until reaching a upper bound of 3 variables per rule and 5 literals per rule. In light of these results, we remark that bias windowing provides a natural trade-off between accuracy and efficiency. Indeed, the learner is able to return accurate theories even for strong biases. Furthermore, we observe that after a period of underfitting the algorithm converge on hypotheses which are stable and effective. Interestingly, the length of this period is correlated with background knowledge. For poorly informed domains such $B2$, the algorithm needs large windows to provide accurate learners. On the other hand, for $B4$ the algorithm quickly converges on small windows that lead to very accurate hypotheses. This phenomenon is closely related to phase transition effects reported in [8]. In particular, the variance of underfitting periods observed in windowing corroborates the evidence that an appropriate use of background knowledge tends to limit phase transition effects.

The table below compares the performance of windowing with the standard learners FOIL and PROGOL, a recent greedy-based learner ICL [17] and the genetic learner G-NET [3]. Note that ICL provides a multi-class theory that combines the hypotheses learned from each separate class. From this table, it can be concluded that windowing generates theories which are stable and very accurate. Notably, for descriptions $B3$ and $B4$, windowing finds in few seconds theories for which effectiveness encompasses the best current techniques.

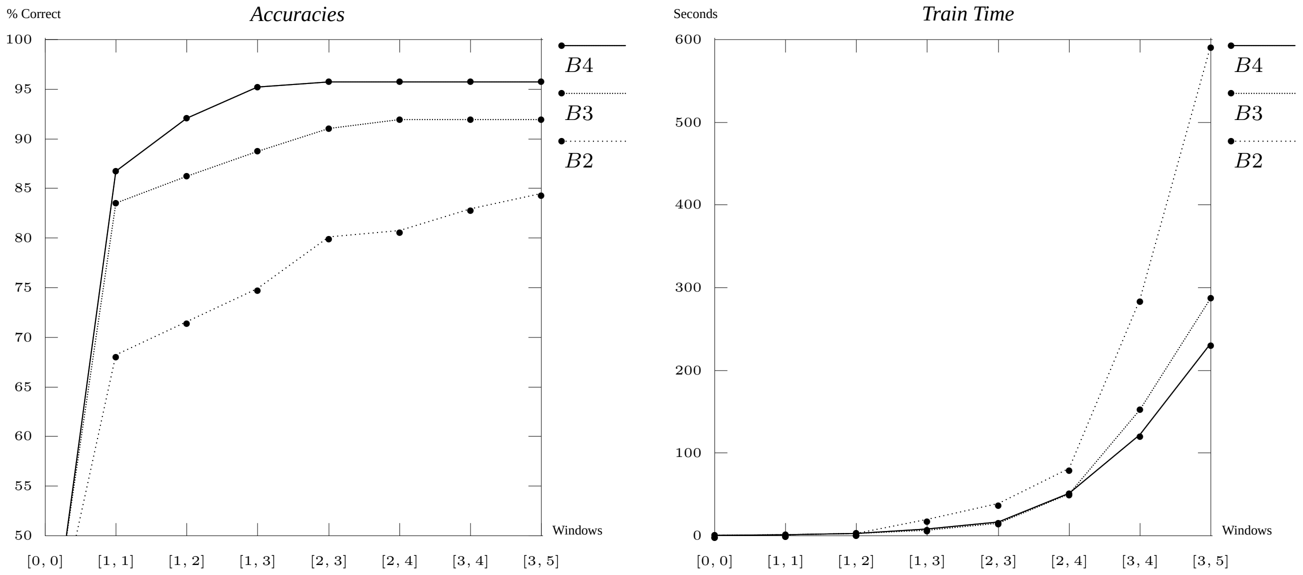

**Figure 3:** Results in the Mutagenesis domain

| | *Accuracies* | | |
|---|---|---|---|
| | $B2$ | $B3$ | $B4$ |
| Foil | $61 \pm 6$ | $83 \pm 3$ | $86 \pm 3$ |
| Progol | $81 \pm 3$ | $83 \pm 3$ | $88 \pm 2$ |
| Icl | $82 \pm 7$ | $87 \pm 10$ | $88 \pm 8$ |
| G-Net | NA | NA | $92 \pm 8$ |
| Windowing $[3, 5]$ | $84 \pm 3$ | $92 \pm 3$ | $96 \pm 2$ |

## 6 CONCLUSION

It has long been recognized that representation biases can help to circumvent the computational issue involved in relational learning by substantially reducing the hypothesis space. Moreover, if used appropriately, a representation bias can limit overfitting effects by enabling the learner to focus on small theories which are stable and accurate. However, choosing the good representation bias for the domain at hand is a notoriously hard task which is often left to the user.

This work is an attempt to automatically select small classes from which theories are learned. Our model of "bias windowing" is logically settled on a notion of robust induction that allows the learner to infer hypotheses given any bias available. The algorithmic part integrates two components: a learning method that attempts to approximate the best hypothesis for a given window, and a selection technique which attempts to identify the best window for a given domain. The only proviso is that sufficient unlabeled training data be available. Experiments on the Mutagenesis dataset reveal that windowing tends to converge on stable and accurate hypotheses.

Several directions of future research are possible. Clearly, more experiments need to be done to study the performance profiles of windowing. The development of a competence map for phase transition problems is a subject of on-going research. Furthermore, the efficiency of window-based learning could be improved by using a beam search strategy whose width progressively decreases as the size of the window is enlarged. Finally, the notion of bias window could be extended by incorporating the atoms which are relevant for the application domain (see e.g. [2]). In this setting, forward selection approaches such as feature selection would be particularly interesting for governing exploration in the lattice of windows.

## REFERENCES

[1] H. Adé, L. De Raedt, and M. Bruynooghe, 'Declarative bias for specific to general ILP systems', *Mach. Learning*, **20**, 119–154, (1995).

[2] J. Ales Bianchetti, C. Rouveirol, and M. Sebag, 'Constraint-based learning of long relational concepts', in *Proceedings of the 19th International Conference on Machine Learning*, pp. 35–42, (2002).

[3] C. Anglano, A. Giordana, G. Lo Bello, and L. Saitta, 'An experimental evaluation of coevolutive concept learning', in *Proceedings of the 15th International Conference on Machine Learning*, pp. 19–27, (1998).

[4] V. Chvatal, 'A greedy heuristic for the set covering problem', *Mathematics of Operation Research*, **4**(3), 233–235, (1979).

[5] L. De Raedt and L. Dehaspe, 'Clausal discovery', *Machine Learning*, **26**, 99–146, (1997).

[6] J. Fürnkranz, 'Pruning algorithms for rule learning', *Machine Learning*, **27**, 139–171, (1997).

[7] J. Fürnkranz, 'Integrative windowing', *Journal of Artificial Intelligence Research*, **8**, 129–164, (1998).

[8] A. Giordana and L. Saitta, 'Phase transitions in relational learning', *Machine Learning*, **41**(2), 217–251, (2000).

[9] M. Kearns, R. E. Schapire, and L. M. Sellie, 'Toward efficient agnostic learning', *Machine Learning*, **17**(2–3), 115–142, (1994).

[10] R. D. King and A. Srinivasan, 'Relating chemical activity to structure: An examination of ILP successes', *New Generation Computing, Special issue on Inductive Logic Programming*, **13**(3–4), 411–434, (1995).

[11] N. Lavrač and S. Džeroski, *Relational Data Mining*, Springer, 2001.

[12] L. De Raedt and M. Bruynooghe, 'Interactive learning and constructive induction by analogy', *Machine Learning*, **8**, 107–150, (1992).

[13] N. Linial and N. Nisan, 'Approximate inclusion-exclusion', *Combinatorica*, **10**, 349–365, (1990).

[14] J. R. Quinlan, 'Learning efficient classification procedures and their application to chess end games', in *Machine Learning. An Artificial Intelligence Approach*, volume 1, 463–482, Morgan Kaufman, (1983).

[15] D. Schuurmans and F. Southey, 'Metric-based methods for adaptative model selection and regularization', *Machine Learning*, **48**, 51–84, (2002).

[16] L. G. Valiant, 'Learning disjuctions of conjunctions', in *Proceedings of the 9th International Joint Conference on Artificial Intelligence*, pp. 207–232, (1985).

[17] W. Van Laer, *From Propositional to First Order Logic in Machine Learning and Data Mining*, Ph.D. dissertation, Department of Computer Science, K.U.Leuven, Belgium, 2002.

[18] V. Vapnik and A. Chervonenkis, 'Necessary and sufficient conditions for the uniform convergence of means to their expectations', *Theory of Probability and Its Applications*, **26**, 532–553, (1981).