Fabien Torre
(fabien@lri.fr)

Céline Rouveirol
(celine@lri.fr)

Private Properties and Natural Relations in Inductive Logic Programming

This work is partly supported by the Esprit LTR Project n°20237 (ILP^2).

Abstract

Learning in First Order Logic languages raises the problem of searching in very large or even infinite search spaces. In this framework, diverse pruning techniques have been studied. On the one hand, the search space is classically ordered by a generality relation, which can be exploited to dynamically and soundly prune all hypotheses that are more general than an incorrect one (and dually, to prune all hypotheses that are more specific than an incomplete one) [Mitchell, 1982]. On the other hand, language bias provides additional information about which are the relevant hypotheses to the problem at hand, but pruning with respect to language bias may be unefficient in a search space ordered by a generality relation.

We present in this paper the new notion of *private property* that characterizes cases where sound pruning can take place with respect to a property, be it completeness, correctness with respect to the training examples, or hypothesis language restriction. When a generated hypothesis H does not satisfy a private property, all the descendants of H can be safely pruned.

On this notion, we build new quasi-orders, called *natural relations* of a property, to order the search space. We characterize natural relations for conjunctions of such properties. Learning operators that satisfy the natural relation of a set of properties allow pruning with respect to this set of properties to take place beforehand, i.e., before the unappropriate hypotheses are generated.

Résumé

Une difficulté majeure de l'apprentissage en langage du premier ordre est liée à la taille des espaces de recherche devant être explorés. Dans ce cadre, plusieurs méthodes d'élagage ont été élaborées. La première consiste à utiliser la relation de généralité qui ordonne l'espace de recherche [Mitchell, 1982] : cela permet d'élaguer les hypothèses plus générales qu'une hypothèse incorrecte (symétriquement, les hypothèses plus spécifiques qu'une hypothèse incomplète). D'autre part, les biais de langage fournissent des informations supplémentaires sur les hypothèses qui sont pertinentes pour le problème considéré. Cependant, l'élagage par rapport à ces biais peut se révéler particulièrement inefficace dans un espace ordonné par une relation de généralité.

Nous présentons ici la notion de *propriété privée* qui caractérise les cas où un élagage sans risque est possible par rapport à une propriété, qu'il s'agisse de la complétude, de la correction par rapport aux exemples, ou de restrictions sur le langage des hypothèses. Si une hypothèse H ne satisfait pas une propriété privée, toutes les descendantes de H peuvent être élaguée sans risque de perdre une solution.

Sur cette base, nous construisons de nouveaux pré-ordres, appelés *relations naturelles* d'une propriété, pour ordonner l'espace de recherche. Nous caractérisons les relations naturelles pour des conjonctions de propriétés. Un opérateur de raffinement qui satisfait la relation naturelle d'un ensemble de propriétés permet l'élagage par rapport à cet ensemble de propriétés.

1 Introduction

A search problem consists of a set of states (the search space), a set of operators, an initial state and a goal state. The goal state may not be explicitly described, but may rather be specified through a set of properties it must satisfy.

After [Mitchell, 1982], "*generalization problem is essentially a search problem*". In the framework of definite semantics, the search space is a set of definite clauses, the initial state is given by a positive example of the target concept and the operators are learning operators that alter a hypothesis clause for the target concept into a set of new and possibly better hypotheses. The learning goal is classically defined in Inductive Logic Programming (ILP) as follows [Muggleton and De Raedt, 1994]: given a set E^+ of positive examples and a set E^- of negative examples for the target concept, a background knowledge B, find a hypothesis H such that

$$\begin{array}{lll} \forall e^+ \in E^+ : & B \cup H \models e^+ & (H \text{ is complete}) \ , \\ \forall e^- \in E^- : & B \cup H \not\models e^- & (H \text{ is correct}) \ . \end{array}$$

In ILP, the cost of exhaustively exploring the search space in order to find a hypothesis that satisfies the completeness and correctness criteria is prohibitive, and several techniques have been developed to prune the search space, either statically (before search) or dynamically (during search).

On the one hand, a well-known pruning technique [Mitchell, 1982] is to exploit the generality ordering on the learning search space. For instance, given that the search proceeds bottom up with respect to a generality ordering, and that a given hypothesis H in the search space covers a negative example, it is not necessary to develop any of its generalizations, as none of them will ever meet the correctness criterion anymore.

ILP related works have studied different quasi-orderings for First Order Logic (FOL) search spaces: θ-subsumption [Plotkin, 1970], generalized subsumption [Buntine, 1988], T-implication [Idestam-Almquist, 1995], or logical implication [Nienhuys-Cheng and de Wolf, 1996], and have formalized learning operators as *refinement operators* that go through a quasi-ordered space of clauses [Shapiro, 1981, Niblett, 1993, van der Laag, 1995].

On the other hand, pruning techniques exploiting additional constraints on the expected target concept definition known as *learning bias*, have also been extensively studied in ILP (see [Nédellec et al., 1996] for a survey). In particular, *language bias* allows constraint setting on the syntax of the target concept definition. Handling language bias, by pruning hypotheses which are irrelevant with respect to a specific learning problem, adapts the learning process to the problem at hand, and enhances both the quality and the efficiency of learning.

However, in learning systems that search a hypothesis space ordered by a generality relation, the handling of language bias may be expensive: hypotheses generated by the learning operator may not always satisfy the language bias, which therefore have to be tested at each learning step. Even worse, it may well be that a hypothesis does not satisfy a language bias, and that some of its descendants through the learning operator satisfy it: even if a hypothesis fails against the language bias test, its descendants nevertheless have to be generated and tested.

In this paper, we study a smoother and more efficient way to integrate language bias handling in learning. As opposed to systems that only use a generality ordering to explore and prune the search space, the idea is to also take into account the language bias to order the search space. In that aim, we propose an extended definition for the learning task: given a set of properties $\mathcal{P} = \{P_1, \ldots, P_n\}$, find a hypothesis H such that

$$P_1(H) \wedge \ldots \wedge P_n(H) \ .$$

The P_i necessarily include at least completeness or correctness with respect to the examples of the target concept. To deal with this new definition, new quasi-orders called *natural relations* are designed that allow optimal pruning with respect to training example coverage *and* a subset $\mathcal{P}_{ord}$ of $\mathcal{P}$. This pruning is dynamic : when a generated hypothesis H does not satisfy $\mathcal{P}_{ord}$, all the descendants of H can be safely pruned. Roughly speaking, this amounts to pruning the search space with respect to both a generality relation and some language bias, as done previously with respect to a generality order only. This saves the cost of generating and testing inappropriate hypotheses.

The report is organized as follows. In section 2, we give general definitions necessary to introduce our framework. We characterize, in section 3, cases where sound pruning can take place with respect to a property. In section 4, we define $\geq_{\mathcal{P}}$, the *natural relation* for a property set $\mathcal{P}$. We conclude this work by describing its perspectives.

2 Preliminaries

2.1 Notations and definitions

- If F is a formula, then $\forall F$ denotes the universal closure of F, which is the closed formula obtained by adding a universal quantifier for every variable having a free occurrence in F [Lloyd, 1987].
- The semantic of $\overline{F}$ is the negation of F.
- For a given property P, we call the dual property P, the property $\overline{P}$.

Definition 1 (relation)
A binary relation *on a set S is a subset $\mathcal{R}$ of $S \times S$.*

We use $a \,\mathcal{R}\, b$ to denote $(a, b) \in \mathcal{R}$. If $a \,\mathcal{R}\, b$ and $b \,\mathcal{R}\, a$, we note $a \sim_{\mathcal{R}} b$.

- A binary relation $\mathcal{R}$ is called *reflexive* iff $\forall a \in A : a \,\mathcal{R}\, a$.
- A binary relation $\mathcal{R}$ is called *transitive* iff $\forall a, b, c \in A : a \,\mathcal{R}\, b \wedge b \,\mathcal{R}\, c \Rightarrow a \,\mathcal{R}\, c$.
- We note $\mathcal{R}^{-1}$, the *inverse relation* of $\mathcal{R}$, that is the binary relation defined by $x \,\mathcal{R}^{-1}\, y \Leftrightarrow y \,\mathcal{R}\, x$.

Interesting relations for ILP are *quasi-orders*.

Definition 2 (Quasi-order)
A binary relation is called quasi-order *iff it is both reflexive and transitive.*

Definition 3
A quasi-ordered set is a couple $\langle S, \geq \rangle$ *where* S *is a set and* $\geq$ *is a quasi-order on* $S \times S$.

Among quasi-orders, the *generality orderings*

Definition 4 (Generality Ordering)
A generality ordering *is a quasi-ordering* $\geq$ *on clauses such that* $C \geq D$ *implies* $C \models D$.

Several generality orderings have been studied within ILP, among those, θ-subsumption [Plotkin, 1970].

Definition 5 (θ-subsumption)
A clause C *θ-subsumes a clause* D *iff*

$$\exists \theta : C\theta \subseteq D \ .$$

Let us also quote generalized subsumption [Buntine, 1988], T-implication [Idestam-Almquist, 1995] or logical implication [Nienhuys-Cheng and de Wolf, 1996].

Definition 6 (Refinement Operator)
In this paper, we consider a refinement operator *as a binary relation on the search space.*

An operator $\mathcal{O}$ is then represented by the set of pairs (H, H'), such that $H' \in \mathcal{O}(H)$.

Typically, the search space is a set of clauses S ordered by a generality relation $\geq$. A binary relation $\mathcal{O}$ is an *upward refinement operator* for $\langle S, \geq \rangle$ iff

$$\forall C, D \in S : C \ \mathcal{O} \ D \Rightarrow C \geq D \ .$$

In other words, $\mathcal{O}$ is an upward refinement operator iff

$$\forall C \in S : \{D \in S \mid C \ \mathcal{O} \ D\} \subseteq \{D \in S \mid C \geq D\} \ .$$

In the same way, a binary relation $\mathcal{O}$ is a *downward refinement operator* for $\langle S, \geq \rangle$ iff

$$\forall C, D \in S : C \ \mathcal{O} \ D \Rightarrow D \geq C \ .$$

We note $\mathcal{O}(C)$ the set

$$\{D \in S \mid C \ \mathcal{O} \ D\} \ .$$

Then, we will say that an operator $\mathcal{O}$ *satisfies* a relation $\mathcal{R}$ iff

$$\mathcal{O} \subseteq \mathcal{R} \ .$$

2.2 Target concept properties

As quoted in the introduction, we include in the set of expected properties for the target concept completeness and correctness with respect to training examples and language bias constraints.

We can find in ML and ILP literature a large range of language restrictions that a target concept definition should satisfy. We detail in the following the ones we have focused our study on. This list is largely representative of language biases classically used in ILP. Besides, as we will see in next section, it can be considered as indicative, as our framework is general enough to integrate new properties. For each property of this list, the dual property can be considered.

First, COVER$_{\mathsf{e}}$ means a hypothesis must cover a given example e.

One may impose a upper bound on

- VARS$_{\mathsf{n}}$: the number of existential variables (variables not occurring in the head of the clause),
- LENGTH$_{\mathsf{n}}$, the number of literals in the body of the clause (we call this quantity the length of the clause),
- DEPTH$_{\mathsf{d}}$ [Muggleton and Feng, 1990]. The depth of a clause is the maximal depth of its terms. Variables and constants have depth zero. A term $f(\ldots, t_i, \ldots)$ has depth one plus the maximal depth of t_i.
- LEVEL$_{\mathsf{l}}$ [Džeroski et al., 1992]. The level of a clause is the maximum of the level of its variables. Variables of the head have level zero. The level of an existential variable V is one plus the minimal level of variables appearing in the first literal containing V.

One may also impose a hypothesis to be

- RANGE-RESTRICTED: all variables of the head must appear in the body.
- CONNECTED[Quinlan, 1990]: in each literal, at least one variable appears either in the head or in a previous literal.

  There is an other way to describe this property: a clause is connected if the level is defined for all variables of the clause.
- REDUCED for θ-subsumption : C is reduced iff

$$\nexists \theta : C\theta \subset C \ .$$

In the above list, some of the language biases have been specifically developed for FOL languages. On the one hand, those language biases have an interest as they allow expressing meaningful information about the expected form of the concept. For instance, the range-restriction property states that a meaningful definition for the concept "X is the grandfather of Y", should contain constraints on X and Y. On the other hand, those language biases define subsets of FOL for which the coverage test of hypotheses with respect to examples is (relatively) efficient. Finally, any hypothesis that fails against a language bias does not have to be checked against examples, which may save a lot of computation efforts.

3 Private property

The aim is to explore a small search space without risking to miss a solution. It is safe to stop the refinement of a given H that does not satisfy the expected properties (and therefore to prune the search space), iff no descendant of H will ever satisfy those properties. This intuition is illustrated on figure 1, and formalized in the following definition.

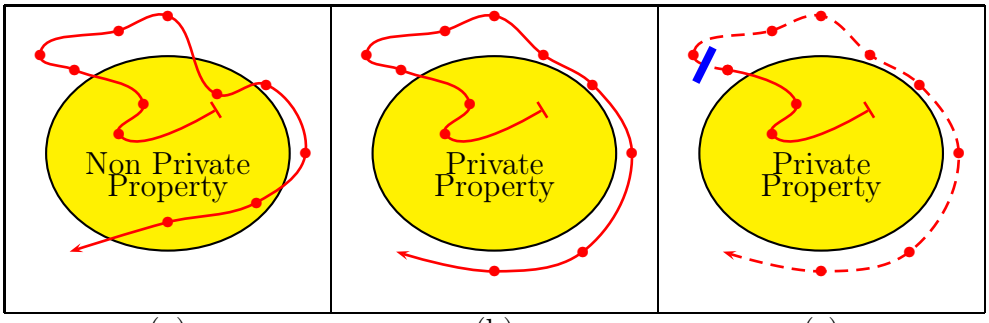


Figure 1: Private and non private properties. Figure (a) illustrates that safe pruning is not possible with respect to non private property. In case (b), pruning is sound and (c) shows how pruning operates.

Definition 7 (private property)
A property P is said private *with respect to the relation $\mathcal{R}$ iff*

$$\forall H, H' \in S : \quad \forall \left(H \mathcal{R} H' \wedge \overline{P(H)} \Rightarrow \overline{P(H')} \right) \quad .$$

Let us assume the property P is expressed by a FOL formula. The parameters of the property are the free variables in this formula.

Example 1 (length of clause)
Let us consider the property that bounds the length of clause to k literals, expressed as $|H| \leq k$. The property $\textsc{Length}_k$ *is private with respect to the relation $\mathcal{R}$ iff*

$$\forall H, H' \in S : \quad \forall k \in \mathbb{N} : (H \mathcal{R} H' \wedge |H| > k \Rightarrow |H'| > k) \quad .$$

A relation $\mathcal{R}$ that satisfies this property is, for instance, the one defined by: $H \mathcal{R} H'$ iff there exists a literal L such that $H' = H \cup \{L\}$.

Table 1 shows properties which are private with respect to a adding literal operator. A quick look on this table shows that not all properties are private with respect to this operator. This observation is true for all classical refinement operators: roughly speaking, for a such operator, properties split up into two sets approximately of the same size: properties which are private for the given operator, and properties which are not. In particular, a property and its dual are never private at the same time.

Table 1: Private properties with respect to adding literal operator

Property P	P is private	$\overline{P}$ is private
$\textsc{Cover}_{\mathsf{e}}$	$\surd$	
$\textsc{Range-Restricted}_{\mathsf{b}}$		$\surd$
$\textsc{Connected}_{\mathsf{b}}$	$\surd$	
$\textsc{Reduced}_{\mathsf{b}}$		
$\textsc{Length}_{\mathsf{n}}$	$\surd$	
$\textsc{Vars}_{\mathsf{n}}$	$\surd$	
$\textsc{Depth}_{\mathsf{d}}$	$\surd$	
$\textsc{Level}_{\mathsf{l}}$	$\surd$	

This means that, if the properties which are relevant for the considered problem are not private with respect to the operator, the search is performed without language biases.

Now, we do not impose the operator anymore. Our aim is to build operators that allow pruning with respect to a given set of properties (the set of properties which are relevant for the problem).

4 Natural Relations

A property is private for many relations. For instance, the empty and identity relations make any property private and are, as a consequence, of little interest, as far as the pruning of the search space is concerned. Besides, they are not very useful as search operators... Indeed, an operator satisfying the identity relation applied to a given hypothesis H, generates at most H itself. We will consider in the following that a natural relation for a given property is one of the largest relations that make this property private.

Definition 8 (natural relation)
$\mathcal{R}$ *is a* natural relation *for the property* P *iff*

1. P *is private with respect to the relation* $\mathcal{R}$*;*
2. *if a relation* $\mathcal{R}'$ *exists such that* P *is private with respect to the relation* $\mathcal{R}'$ *and* $\mathcal{R} \Rightarrow \mathcal{R}'$*, then* $\mathcal{R} = \mathcal{R}'$*.*

Let us try to give a more precise characterization of this natural relation. We will see in the following that the natural relation of a property is indeed unique. Two hypotheses are "naturally" related if, for all possible instantiations of the property parameters, either the first hypothesis satisfies the property or the second one does not.

Proposition 1
A property P *has a single natural relation, denoted* $\geq_P$*.* $\geq_P$ *is defined by*

$$\forall C, D \in S: \quad C \geq_P D \Leftrightarrow \forall \left(P(C) \vee \overline{P(D)} \right) \ .$$

Proof:

- First, we prove that P is private with respect to $\geq_P$. Assume that P is not private with respect to $\geq_P$. Then, there are H and H' in S such that

$$H \geq_P H' \wedge \overline{P(H)} \wedge P(H') \ ,$$

that is

$$[P(H) \vee \overline{P(H')}] \wedge \overline{P(H)} \wedge P(H') \ .$$

This is obviously impossible.

- Second, we have to show that $\geq_P$ is the unique natural relation of P. Assume now there is a relation $\mathcal{R}$ such that P is private with respect to $\mathcal{R}$ and there is a pair (a, b) in $\mathcal{R}$ which is not in $\geq_P$. Then, for some parameters of P, we have $\overline{P(a) \vee \overline{P(b)}}$, that is

$$\overline{P(a)} \wedge P(b) \ .$$

This contradicts the assumption : since P is private with respect to $\mathcal{R}$,

$$a \,\mathcal{R}\, b \wedge \overline{P(a)} \Rightarrow \overline{P(b)} \ .$$

We have shown that P is private with respect to $\geq_P$, and that a relation which is not included in $\geq_P$ cannot make P private. In conclusion, $\geq_P$ is the unique natural relation of P. □

Therefore, a property P is private with respect to the relation $\mathcal{R}$ iff $\mathcal{R} \Rightarrow \ \geq_P$. This result justifies why we have chosen the largest relation as natural relation: a relation makes a property private iff this relation is included in the natural relation of the property. Therefore, safe and dynamic pruning of the search space with respect to a given property can only be achieved through an operator which satisfies its natural relation.

Remark 1
Given the definition of downward refinement operator (definition 6), if $\mathcal{O}$ is downward with respect to $\geq_P$ then P is private with respect to $\mathcal{O}$.

We assume, in the remainder of this paper, that any property can be expressed as $f(H)\,\mathcal{R}\,k$ where

- f is a function from the search space S to a domain $\mathcal{D}_f$;
- $\mathcal{R}$ is a quasi-order on $\mathcal{D}_f \times \mathcal{D}_f$;
- $k \in \mathcal{D}_f$ (k represents the parameter of the property P).

Table 2 shows our properties expressed in this framework. Note that some properties are boolean ($b \in \{true, false\}$): RANGE-RESTRICTED$_\mathsf{b}$, CONNECTED$_\mathsf{b}$ and REDUCED$_\mathsf{b}$.

This assumption on the form of the property allows the simplification of the expression of natural relation.

Table 2: Properties expressed in our framework.

$\textsc{Cover}_{\mathsf{e}}$	$H \models e$
$\textsc{Range-Restricted}_{\mathsf{b}}$	$\mathrm{RR}(H) = b$
$\textsc{Connected}_{\mathsf{b}}$	$\mathrm{Connected}(H) = b$
$\textsc{Reduced}_{\mathsf{b}}$	$\mathrm{Reduced}(H) = b$
$\textsc{Length}_{\mathsf{n}}$	$\lvert H\rvert \leq_{\mathbb{N}} n$
$\textsc{Vars}_{\mathsf{n}}$	$\mathrm{Nv}(H) \leq_{\mathbb{N}} n$
$\textsc{Depth}_{\mathsf{d}}$	$\mathrm{Depth}(H) \leq_{\mathbb{N}} d$
$\textsc{Level}_{\mathsf{l}}$	$\mathrm{Level}(H) \leq_{\mathbb{N}} l$

Proposition 2
Let P be a property defined by $\forall C \in S : P(C) \Leftrightarrow f(C)\,\mathcal{R}\,k$. The natural relation of P is also defined by

$$\forall C, D \in S : \quad C \geq_P D \Leftrightarrow f(C)\,\mathcal{R}\,f(D)$$

Proof: We have to show that, for all C and D in S,

$$\left[\forall k : f(C)\,\mathcal{R}\,k \vee \overline{f(D)\,\mathcal{R}\,k}\right] \Leftrightarrow f(C)\,\mathcal{R}\,f(D)$$

- First, let us prove:

$$\left[\forall k : f(C)\,\mathcal{R}\,k \vee \overline{f(D)\,\mathcal{R}\,k}\right] \Rightarrow f(C)\,\mathcal{R}\,f(D) \ .$$

  Assume that this implication does not hold. Then, there are C and D in S such that

$$[\forall k : f(C)\,\mathcal{R}\,k \vee \overline{f(D)\,\mathcal{R}\,k}] \wedge \overline{f(C)\,\mathcal{R}\,f(D)} \ .$$

  If we choose $k = f(D)$, then $\overline{f(D)\,\mathcal{R}\,k}$ can not hold since $\mathcal{R}$ is reflexive, hence we have

$$f(C)\,\mathcal{R}\,f(D) \wedge \overline{f(C)\,\mathcal{R}\,f(D)} \ .$$

- Now, assume that

$$\left[\forall k : f(C)\,\mathcal{R}\,k \vee \overline{f(D)\,\mathcal{R}\,k}\right] \Leftarrow f(C)\,\mathcal{R}\,f(D) \ ,$$

  is false. Then, there are C, D and k such that

$$\overline{f(C)\,\mathcal{R}\,k} \wedge f(D)\,\mathcal{R}\,k \wedge f(C)\,\mathcal{R}\,f(D) \ .$$

  But, since $\mathcal{R}$ is transitive, we find

$$\overline{f(C)\,\mathcal{R}\,k} \wedge f(C)\,\mathcal{R}\,k \ .$$

We have proved the two implication, therefore the implication is true. □

Example 2
Let us consider the property $\textsc{Cover}_{\mathsf{e}}$ *stating that a hypothesis* H *covers an example* e, $H \models e$. *Function* f *is here identity,* $\mathcal{R}$ *is identified to* $\models$, *the parameter of the function is* e. *By direct application of the above proposition, the natural relation for* $\textsc{Cover}_{\mathsf{e}}$ *is defined by* $\forall C, D \in S : C \geq_{Cover} D \Leftrightarrow C \models D$.

Thus, saying that an operator makes $\textsc{Cover}_{\mathsf{e}}$ private is equivalent to imposing that this operator satisfies a generality ordering. This justifies that a refinement operator must satisfy a generality ordering. The notion of natural relation extends that of generality order to all properties a target concept must satisfy. Table 3 lists the natural relations for all the target concept properties we address.

Table 3: Natural relations.

Property P	Relation $\forall C, D \in S : C \geq_P D \Leftrightarrow$
$\textsc{Cover}_{\mathsf{e}}$	$C \models D$
Range-Restricted	$\mathrm{RR}(C) = \mathrm{RR}(D)$
Connected	$\mathrm{Connected}(C) = \mathrm{Connected}(D)$
Reduced	$\mathrm{Reduced}(C) = \mathrm{Reduced}(D)$
$\textsc{Length}_{\mathsf{n}}$	$\lvert C\rvert \leq_{\mathbb{N}} \lvert D\rvert$
$\textsc{Vars}_{\mathsf{n}}$	$\mathrm{Nv}(C) \leq_{\mathbb{N}} \mathrm{Nv}(D)$
$\textsc{Depth}_{\mathsf{p}}$	$\mathrm{Depth}(C) \leq_{\mathbb{N}} \mathrm{Depth}(D)$
$\textsc{Level}_{\mathsf{l}}$	$\mathrm{Level}(C) \leq_{\mathbb{N}} \mathrm{Level}(D)$

Now that the natural relation for a single property has been characterized, we explore how to compute the natural relation of its dual property and then, and the natural relation of a conjunction of properties, as there are usually more than one property imposed on the target concept definition.

Proposition 3 (dual property)
Let P *be a property and,* $\geq_P$, *its natural relation,*

$$\forall C, D \in S : \quad D \geq_P C \Leftrightarrow C \geq_{\overline{P}} D \ .$$

Proof:

$$\begin{aligned} C \geq_{\overline{P}} D &\Leftrightarrow \forall \left(\overline{P}(C) \vee \overline{\overline{P}(D)}\right) \\ &\Leftrightarrow \forall \left(P(D) \vee \overline{P(C)}\right) \\ &\Leftrightarrow D \geq_P C \end{aligned}$$

This is a direct consequence of the natural relation definition. □

The natural relation of a dual property is the inverse relation of the natural relation. After remark 1, a downward operator with respect to $\geq_P$ makes this property private and allows dynamic pruning of hypotheses that do not satisfy this property. Therefore, the corresponding upward operator deals with the dual property.

Example 3
The natural relation for the dual of the property $\textsc{Cover}_{\mathsf{e}}$

$$\forall C, D \in S : C \geq_{\overline{Cover}} D \Leftrightarrow D \models C \ .$$

Proposition 4 (conjunction of properties)
Let P_1 *and* P_2 *be two properties,* $\geq_{P_1}$ *and* $\geq_{P_2}$*, their associated natural relations,*

$$\forall C, D \in S : \quad C \geq_{P_1 \wedge P_2} D \Leftrightarrow C \geq_{P_1} D \wedge C \geq_{P_2} D \ .$$

Proof:
We have to prove

$$\forall \left((P_1 \wedge P_2)(C) \vee \overline{(P_1 \wedge P_2)(D)} \right) \Leftrightarrow C \geq_{P_1} D \wedge C \geq_{P_2} D$$

or,

$$\forall \left[(P_1(C) \wedge P_2(C)) \vee \overline{P_1(D)} \vee \overline{P_2(D)} \right] \Leftrightarrow C \geq_{P_1} D \wedge C \geq_{P_2} D$$

Let us consider that properties P_1 and P_2 are expressed as follows.

$$\begin{array}{lcl} P_1(H) & \Leftrightarrow & f_1(H)\, \mathcal{R}_1\, k_1 \\ P_2(H) & \Leftrightarrow & f_2(H)\, \mathcal{R}_2\, k_2 \end{array}$$

Then, we have to prove

$$\begin{array}{ll} & \forall k_1, k_2 : \left[(f_1(C)\, \mathcal{R}_1\, k_1 \wedge f_2(C)\, \mathcal{R}_2\, k_2) \vee \overline{f_1(D)\, \mathcal{R}_1\, k_1} \vee \overline{f_2(D)\, \mathcal{R}_2\, k_2} \right] \\ \Leftrightarrow & f_1(C)\, \mathcal{R}_1\, f_1(D) \wedge f_2(C)\, \mathcal{R}_2\, f_2(D) \end{array}$$

- First, let us prove the following implication

$$\begin{array}{ll} & \forall k_1, k_2 : \left[(f_1(C)\, \mathcal{R}_1\, k_1 \wedge f_2(C)\, \mathcal{R}_2\, k_2) \vee \overline{f_1(D)\, \mathcal{R}_1\, k_1} \vee \overline{f_2(D)\, \mathcal{R}_2\, k_2} \right] \\ \Rightarrow & f_1(C)\, \mathcal{R}_1\, f_1(D) \wedge f_2(C)\, \mathcal{R}_2\, f_2(D) \end{array}$$

  Assume that this implication does not hold. This means there are C and D such that

$$\begin{array}{ll} & \forall k_1, k_2 : \left[(f_1(C)\, \mathcal{R}_1\, k_1 \wedge f_2(C)\, \mathcal{R}_2\, k_2) \vee \overline{f_1(D)\, \mathcal{R}_1\, k_1} \vee \overline{f_2(D)\, \mathcal{R}_2\, k_2} \right] \\ \wedge & \overline{f_1(C)\, \mathcal{R}_1\, f_1(D) \wedge f_2(C)\, \mathcal{R}_2\, f_2(D)} \end{array}$$

  or

$$\begin{array}{ll} & \forall k_1, k_2 : \left[(f_1(C)\, \mathcal{R}_1\, k_1 \wedge f_2(C)\, \mathcal{R}_2\, k_2) \vee \overline{f_1(D)\, \mathcal{R}_1\, k_1} \vee \overline{f_2(D)\, \mathcal{R}_2\, k_2} \right] \\ \wedge & \left[\overline{f_1(C)\, \mathcal{R}_1\, f_1(D)} \vee \overline{f_2(C)\, \mathcal{R}_2\, f_2(D)} \right] \end{array}$$

  Now, we choose $k_1 = f_1(D)$ and $k_2 = f_2(D)$. Then, we find

$$[(f_1(C)\, \mathcal{R}_1\, k_1 \wedge f_2(C)\, \mathcal{R}_2\, k_2)] \wedge \left[\overline{f_1(C)\, \mathcal{R}_1\, k_1} \vee \overline{f_2(C)\, \mathcal{R}_2\, k_2} \right]$$

  This does never hold.

- Second, consider the following implication is false.

$$\begin{array}{rl} & \forall k_1, k_2 : \Big[(f_1(C)\,\mathcal{R}_1\,k_1 \wedge f_2(C)\,\mathcal{R}_2\,k_2) \vee \overline{f_1(D)\,\mathcal{R}_1\,k_1} \vee \overline{f_2(D)\,\mathcal{R}_2\,k_2}\Big] \\ \Leftarrow & f_1(C)\,\mathcal{R}_1\,f_1(D) \wedge f_2(C)\,\mathcal{R}_2\,f_2(D) \end{array}$$

Then, there exist C, D, k_1 and k_2 satisfying

$$\begin{array}{rl} & \Big(\overline{f_1(C)\,\mathcal{R}_1\,k_1} \vee \overline{f_2(C)\,\mathcal{R}_2\,k_2}\Big) \wedge f_1(D)\,\mathcal{R}_1\,k_1 \wedge f_2(D)\,\mathcal{R}_2\,k_2 \\ \wedge & f_1(C)\,\mathcal{R}_1\,f_1(D) \wedge f_2(C)\,\mathcal{R}_2\,f_2(D) \end{array}$$

Since $\mathcal{R}_1$ and $\mathcal{R}_2$ are transitive relations, we find

$$\Big(\overline{f_1(C)\,\mathcal{R}_1\,k_1} \vee \overline{f_2(C)\,\mathcal{R}_2\,k_2}\Big) \wedge f_1(C)\,\mathcal{R}_1\,k_1 \wedge f_2(C)\,\mathcal{R}_2\,k_2$$

□

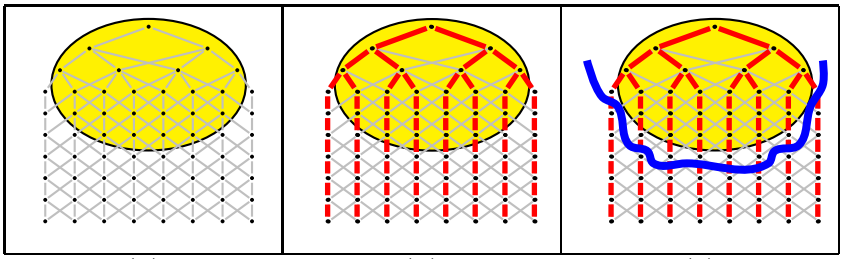
Example 4
By the expression of the completeness criterion as a conjunction of $\textsc{Cover}_{e_i}$ *for* e_i *ranging over the set of positive examples of the target concept, the natural relation for completeness is logical entailment, as expected.*

Example 5
Length and completeness.

$$\mathcal{P} = \{\textsc{Length}_k\} \cup \{\textsc{Cover}_{e_i}\}$$

Then,

$$C \geq_{\mathcal{P}} D \Leftrightarrow (|C| \leq |D|) \wedge (C \models D) \ .$$



(a) (b) (c)

Figure 2: Natural relation. Figure (a) shows the space ordered by the natural relation, that is, the largest relation that makes the property private. Figure (b), one can choose a subset of the natural relation as a refinement operator (the property is private for this operator). Pruning is shown figure (c).

We know now that relations (and operators), that allow sound pruning with respect to a set of properties, are subsets of its natural relation (see figure 2).

5 Conclusion and Future Research

In this paper, we have first introduced the notion of *private property*: this notion characterizes if an operator allows sound pruning with respect to a given property.

Then, we have defined new relations, called *natural relations*, that allow the optimal pruning of the search space. Idestam-Almquist [Idestam-Almquist, 1995] says "*Implication is the most natural and straightforward basis for generalization in inductive learning, since the concept of induction can be defined as the inverse of logical entailment*". We have given here a more formal justification of why logical implication is the most natural relation to order a search space of hypotheses when solutions are defined with respect to at least completeness or correctness criteria. Moreover, we have extended the notion of natural relation to other properties.

The approach of [Shapiro, 1981] is similar to ours. It introduces a language bias, *size*, the aim of which is to make the refinement operator computable. Therefore, strong restrictions are set on *size*: *size* is valued in $\mathbb{N}$, and for a given $n \in \mathbb{N}$, the set of hypotheses H such that $size(H) = n$ is finite. As opposed, we do not set any restriction on the language bias used, except that the property should be expressed as $f(H)\,\mathcal{R}\,k$.

Given our natural relations, we will check whether some ideal operators may exist for the search space ordered by a natural relation, by adapting the conditions introduced in [van der Laag and Nienhuys-Cheng, 1994a, van der Laag and Nienhuys-Cheng, 1994b]. We will then consider our natural relations with respect to other class of operators, such as optimal operators [De Raedt and Bruynooghe, 1993].

References

[Buntine, 1988] Buntine, W. (1988). Generalized subsumption and its application to induction and redundancy. *Artificial Intelligence*, 36:375–399.

[De Raedt and Bruynooghe, 1993] De Raedt, L. and Bruynooghe, M. (1993). A theory of clausal discovery. In Bajcsy, R., editor, *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 1058–1063. Morgan Kaufmann.

[Džeroski et al., 1992] Džeroski, S., Muggleton, S., and Russel, S. (1992). PAC-Learnability of Determinate Logic Programs. In *Proceedings of 5th ACM Workshop on Computational Learning Theory*, pages 128–135. ACM Press.

[Idestam-Almquist, 1995] Idestam-Almquist, P. (1995). Generalization of clauses under implication. *Journal of Artificial Intelligence Research*, 3:467–489.

[Lloyd, 1987] Lloyd, J. W. (1987). *Foundations of Logic Programming*. Springer, Berlin, 2 edition.

[Mitchell, 1982] Mitchell, T. M. (1982). Generalization as search. *Artificial Intelligence*, 18:203–226.

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

[Muggleton and Feng, 1990] Muggleton, S. and Feng, C. (1990). Efficient induction of logic programs. In *Proceedings of the 1st Conference on Algorithmic Learning Theory*, pages 368–381. Ohmsma, Tokyo, Japan.

[Nédellec et al., 1996] Nédellec, C. ., Rouveirol, C., Adé, H., Bergadano, F., and Tausend, B. (1996). Declarative bias in ILP. In Raedt, L. D., editor, *Advances in Inductive Logic Programming*, pages 82–103. IOS Press.

[Niblett, 1993] Niblett, T. (1993). A note on refinement operators. In Brazdil, P. B., editor, *Proceedings of the 6th European Conference on Machine Learning*, volume 667 of *Lecture Notes in Artificial Intelligence*, pages 329–335. Springer-Verlag.

[Nienhuys-Cheng and de Wolf, 1996] Nienhuys-Cheng, S. and de Wolf, R. (1996). Least generalizations and greatest specializations of sets of clauses. *Journal of Artificial Intelligence Research*, 4:341–363.

[Plotkin, 1970] Plotkin, G. (1970). A note on inductive generalization. In *Machine Intelligence*, volume 5. Edinburgh University Press.

[Quinlan, 1990] Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine Learning*, 5(3):239–266.

[Shapiro, 1981] Shapiro, E. Y. (1981). Inductive inference of theories from facts. Technical Report 192, Yale University Department of Computer Science.

[van der Laag, 1995] van der Laag, P. (1995). *An Analysis of Refinement Operators in Inductive Logic Programming*. PhD thesis, Erasmus Universiteit, Rotterdam, the Netherlands.

[van der Laag and Nienhuys-Cheng, 1994a] van der Laag, P. and Nienhuys-Cheng, S. (1994a). Existence and nonexistence of complete refinement operators. In Bergadano, F. and de Raedt, L., editors, *Proceedings of the 7th European Conference on Machine Learning*, volume 784 of *Lecture Notes in Artificial Intelligence*, pages 307–322. Springer-Verlag.

[van der Laag and Nienhuys-Cheng, 1994b] van der Laag, P. and Nienhuys-Cheng, S. (1994b). A note on ideal refinement operators in ILP. In Wrobel, S., editor, *Proceedings of the 4th International Workshop on Inductive Logic Programming*, volume 237 of *GMD-Studien*, pages 247–262. Gesellschaft für Mathematik und Datenverarbeitung MBH.