A Constraint-Based Induction Algorithm in FOL

Michèle Sebag
LMS CNRS URA 317
Ecole Polytechnique, 91128 Palaiseau France
and LRI, Université Paris-XI Orsay, 91405 Orsay France
sebag@xela.polytechnique.fr

Abstract

We present a bottom-up generalization which builds the maximally general terms covering a positive example and rejecting negative examples in first-order logic (FOL), i.e. in terms of Version Spaces, the set G.
This algorithm is based on rewriting negative examples as constraints upon the generalization of the positive example at hand. The constraints space is partially ordered, inducing a partial order on negative examples ; the near-misses as defined by Winston can then be formalized in FOL as negative examples minimal with respect to this partial order. As expected, only near-misses are necessary to build the set G. Moreover, constraints can be used directly to classify further examples.

1 INTRODUCTION

Recently, the number of empirical induction algorithms in first order logic (FOL) is rapidly increased : to cite but a few, see FOIL (Quinlan 90, Quinlan 93), ML-Smart (Bergadano et al. 88, Botta Giordana 93), Golem (Muggleton Feng 90), KBG (Bisson 90), (Bisson 92)...

We focus on discriminant learning from positive and negative examples, and this paper is motivated by the following remark. Near-misses are the real important samples according to (Winston 75) :

> *Small differences permit the machine to localize some part of its current opinion about a concept for improvement. If one wants the machine to learn that the uprights of an arch cannot marry, one should show it a scene that fails to be an arch only in this respect.*

In spite of this claim, near-misses seem to be of no use to FOL learners up to now. As a matter of fact, they are not used (not usable ?) in top down learning, such as in FOIL or ML-Smart, where the stress is put on the discriminant power of features with respect to the whole data. In bottom-up learning, negative examples are seldom used. In KBG (Bisson 90) generalization is guided by clustering the examples on the basis of a sophisticated similarity (Bisson 92) ; but keeping apart positive and negative examples may be ensured only by means of weights (by increasing the weight of the predicate to learn). In Golem (Muggleton Feng 90), negative examples allow to prune a least general generalization. But no mention is made of sorting negative examples with respect to the allowed pruning, i.e. detection of near-misses.

As an attempt to address this remark, we propose a bottom-up approach using an intermediate representation of the negative examples. Negative examples are transformed into constraints on the generalization of positive examples. The constraint space being partially ordered, a partial order on negative examples is therefore derived. Near-misses then are negative examples minimal with respect to this partial order. Constraints are sufficient to build maximally discriminant terms, i.e. the G set in terms of Version Spaces (Mitchell 82). And, as expected, constraints derived from near-misses are sufficient to build G. This result extends that of (Smith Rosenbloom 90) and (Sebag 94) to FOL.

This paper is organized as follows. Section 2 describes the formalization and details the rewriting of negative examples as constraints on the generalization of a positive example. Section 3 shows how to use constraints in building explicitly discriminant terms ; another possibility (section 4) is to use constraints to classify further cases. The complexity of this classification procedure is studied. Last, the advantages and limitations of the proposed algorithm, called CBI (for Constraint-Based Induction) are briefly discussed with respect to some related works.

2 FORMALIZATION

We restrict ourselves to a subset of the first-order logic, excluding negation, existential quantifiers and functions.

2.1 FORMALIZATION OF AN EXAMPLE

Let P_i, $i \in I$, be the predicates involved in describing example Ex. The description of Ex is :

$$Ex = \bigwedge_{i \in I} \bigwedge_{j=1}^{J_i} P_i(a_{i,j,1}, a_{i,j,2}, \ldots, a_{i,j,K_i})$$

where
• K_i denotes the arity of P_i,
• J_i is the number of instanciations of P_i in Ex, (e.g. predicate *Wagon* appears several times in a train description);
• $a_{i,j,k}$ denotes the instanciation of the k-th variable in the j-th instanciation of P_i in Ex.

Example. A West train is described by the conjunction :
Ex : Locomotive(a,sloped_roof,squared_wheel), Wagon (b,1), Roof (b,flat), Wheel (b, front, white), Wheel (b, back, dark).

A variabilization of Ex is obtained by replacing each $a_{i,j,k}$ by variable $X_{i,j,k}$. Let $\mathcal{X}$ be the set of variables $X_{i,j,k}$:

$$\mathcal{X} = \{X_{i,j,k},\ i \in I,\ 1 \leq j \leq J_i,\ 1 \leq k \leq K_i\}$$

Let $\mathcal{F}(\mathcal{X})$ denote the term obtained through this variabilization[1] :

$$\mathcal{F}(\mathcal{X}) = \bigwedge_{i \in I} \bigwedge_{j=1}^{J_i} P_i(X_{i,j,1}, X_{i,j,2}, \ldots, X_{i,j,K_i})$$

By construction, Ex matches $\mathcal{F}$: there exists at least one application σ from $\mathcal{X}$ onto the set $\{ a_{i,j,k} \}$ such that $\mathcal{F}(\sigma(X_{i,j,k})) = Ex$.

Among variables involved in a given predicate, we distinguish variables instanciated by objects, (the *object-variables*), from those instanciated by values, (the *valued-variables*) ; this distinction is ultimately done by experts. In predicate *Roof(X,Y)* for instance, according to instanciation *Roof(b,flat)*, variable X is an object-variable and variable Y is a valued-variable. Set $\mathcal{X}$ can thus be divided into the set of object-variables denoted $\mathcal{O}$, and the set of valued-variables denoted $\mathcal{V}$. Valued-variables are restricted to be either tree-structured, or linear (real- or integer-valued).

[1] A given variable then appears in a unique predicate, that is said to *determine* the variable.

2.1.1 Partitioning Object-Variables

A matching σ of $\mathcal{F}$ by Ex induces a partition over $\mathcal{O}$, denoted $\mathcal{O}_\sigma$: an element of this partition is the set of variables instanciated by the same object. An element (non singleton) of $\mathcal{O}_\sigma$ states that several variables *are* instanciated by the same object in Ex ; it can also be interpreted as : these variables *must* be instanciated by the same object, i.e. are equal. Let $\mathcal{L}_\sigma$ ($\mathcal{L}$ for *link*) be defined as :

$$\mathcal{L}_\sigma(\mathcal{O}) = \bigwedge_{\mathcal{E} \in \mathcal{O}_\sigma,\ |\mathcal{E}|>1,\ \mathcal{E}=\{X_{i_1},\ldots,X_{i_{|\mathcal{E}|}}\}} (X_{i_1} = \ldots = X_{i_{|\mathcal{E}|}})$$

where $|\mathcal{E}|$ stands for the size of set $\mathcal{E}$.

Example : Term $\mathcal{F}$ associated to the West train is (for the sake of readability, variables are noted X, Y, Z, ... in the example) :
$\mathcal{F}(\mathcal{X})$ = *Locomotive(X,Y,Z), Wagon(Q,R), Roof(S,T), Wheel(U,V,W), Wheel(U',V',W')*
The set of object-variables is : $\mathcal{O} = \{X, Q, S, U, U'\}$
Let σ denote the matching of $\mathcal{F}$ by Ex (the first possible one given the description order of Ex) ; objects a and b induce the following partition of $\mathcal{O}$:
$\mathcal{O}_\sigma = \{\sigma^{-1}(a) = \{X\},\ \sigma^{-1}(b) = \{Q, S, U, U'\}\}$
The corresponding $\mathcal{L}_\sigma$ is
$\mathcal{L}_\sigma(X, Q, S, U, U') = (Q = S = U = U')$

2.1.2 Valued Variables

The information related to valued-variables is stored into a term denoted $Values_\sigma$; this approach and the delayed handling of valued-variables are inspired by the flattening operator of (Rouveirol 90). Term $Values_\sigma$is set to the conjunction of selectors
$[X_{i,j,k} = \sigma(X_{i,j,k})]$.

Example. In the West train, one has :
$\mathcal{V} = \{Y, Z, R, T, V, W, V', W'\}$,
and
Values$_\sigma(\mathcal{V})$ = *[Y = sloped_roof], [Z = squared_wheel], [R=1], [T= flat], [V=front], [W=white], [V'=back], [W'=dark].*

2.2 REPRESENTATION OF G

Example Ex is completely defined (up to a permutation of the values of object-variables) by formulas $\mathcal{F}$, $\mathcal{L}_\sigma$, and $Values_\sigma$. Since several matchings between $\mathcal{F}$ and Ex exist in general, we will consider the matching σ defined by $\sigma(X_{i,j,k}) = a_{i,j,k}$, called *canonical matching* of $\mathcal{F}$ by Ex. In the following, terms $\mathcal{L}_\sigma$ and $Values_\sigma$ will always refer to this canonical matching σ.

We limit ourselves to finding generalizations of Ex expressed as conjunctions of :
• a formula f, generalization of $\mathcal{F}$ obtained by the dropping rule ;

• a formula ℓ, (possibly empty) generalization of $\mathcal{L}_\sigma$ obtained by dropping equalities between variables, and such that all variables in ℓ also appear in f;

• a formula v, (possibly empty) generalization of $Values_\sigma$ obtained by either dropping selectors, or generalizing the set of values involved in a selector, and such that all variables in v also appear in f.

Limitation.
Note that the above formalization does not allow for expressing all possible generalizations of Ex ; consider positive example Ex and negative example Ce as follows :
Ex : Wagon(a), Wagon(b), Color(a,red), Color(b,red).
Ce : Wagon(c), Color(c,red).
Then

$$\begin{aligned}\mathcal{F}(X,Y,S,T,S',T') &= Wagon(X), Wagon(Y),\\ &\quad\; Color(S,T), Color(S',T')\\ \mathcal{L}_\sigma(X,Y,S,S') &= ((X=S),(Y=S'))\\ Values_\sigma(T,T') &= [T=red],[T'=red]\end{aligned}$$

As Ce matches $\mathcal{L}_\sigma$ and $Values_\sigma$, Ce also matches any generalization of $\mathcal{L}_\sigma$ and $Values_\sigma$.
A discriminant generalization of Ex such as :
$Wagon(X), Wagon(Y),\ (X \neq Y)$
does not fit in the chosen formalism.

Hypothesis.
We restrict ourselves to deal with discrimination problems that can be solved within this formalism.

The goal then is to find (all) conjunctions of formulas f, ℓ and v, respectively generalizations of $\mathcal{F}$, $\mathcal{L}_\sigma$ and $Values_\sigma$, maximally general such that they reject any negative example.

2.3 REPRESENTATION OF A NEGATIVE EXAMPLE

Negative example Ce is described by a conjunction of instanciations of predicates P_i, analogous to the description of positive example Ex.

To discriminate Ce from Ex, we shall first consider the predicates involved in their descriptions (section 2.4). Then, we shall suppose that Ex and Ce are described by the same collection of predicates ; so their discrimination will be based either on the links among object-variables (section 2.5) or on the values taken by valued-variables (section 2.6).

The discrimination of several negative examples will be considered in 3.1.

2.4 DISCRIMINATION BASED ON PREDICATES

Among the predicates appearing in the descriptions of Ex or Ce, we distinguish three types of predicates :

1- The predicates that appear in Ce only ;
2- The predicates that appear in Ex only ;
3- The predicates that appear in both Ex and Ce.

1- Predicates of type 1 are not useful for the discrimination — given the chosen formalism. In of fact, if $Q()$ denotes a predicate appearing in Ce only, the term $Not\ Q_j()$ is a discriminant generalization of Ex (under the closed world assumption), but does not fit in the formalism (2.2).

2- Predicates of type 2 provide maximally discriminant generalizations. Let $\mathcal{Q}(Ex - Ce)$ denote the set of predicates appearing in Ex and not appearing in Ce ; any predicate $P()$ in $\mathcal{Q}(Ex - Ce)$ generalizes Ex and discrimines Ce by construction. However, these generalizations are neither the only possible ones, nor the most interesting ones, in general.

3- So, we shall now on suppose that all predicates are of type 3 ; the same predicates P_i are involved in Ex and Ce (the number of occurences of P_i in Ex and Ce may be different of course), and we assume that Ex and Ce can still be discriminated by terms fitting in the chosen formalism.

Both Ex and Ce match $\mathcal{F}$ as all predicates in $\mathcal{F}$ belong to the description of Ex and Ce. A discrimination process thus has to compare applications σ and τ, where σ (respectively τ) denotes a matching between $\mathcal{F}$ and Ex (resp. Ce). Several matchings between $\mathcal{F}$ and Ex (or Ce) are possible if a predicate occurs several times in the description of Ex (resp. Ce). Consider now the permutation of two occurences of a given predicate. This transposition does not alter Ex and Ce (the description of an example does not depend on the order of its descriptive features) ; nor does it alter the differences between σ and τ. This remark leads to the following pruning : instead of comparing all possible matchings between $\mathcal{F}$ and Ex, to all possible matchings between $\mathcal{F}$ and Ce, we limit to compare one matching σ between $\mathcal{F}$ and Ex (the canonical one, 2.2) to all possible matchings τ between $\mathcal{F}$ and Ce.

2.5 DISCRIMINATION BASED ON LINKED VARIABLES

Consider first the case where $\mathcal{F}$ only involves object-variables. Comparing Ex and Ce then reduces to comparing the matchings of object-variables, or, equivalently (up to renaming their values) comparing partitions $\mathcal{O}_\sigma$ and $\mathcal{O}_\tau$.
Example. Consider positive example Ex and negative example Ce :
Ex = P(a,b,c), Q(b,d), R(b,d), Q(a,d)
Ce = P(e,f,g), Q(e,f), R(f,g)
Ex and Ce both match term $\mathcal{F}$
$\mathcal{F} = P(X,Y,Z), Q(S,T), Q(S',T'), R(U,V)$
Given $\mathcal{F}$, Ex is completely described by :
$\mathcal{O}_\sigma = \{\{X,S'\},\{Y,S,U\},\{Z\},\{T,T',V\}\}$

And Ce is described by

$\mathcal{O}_\tau = \{\{X,S,S'\},\{Y,T,T',U\},\{Z,V\}\}$

A given element of $\mathcal{O}_\sigma$

• either is a subset of an element of $\mathcal{O}_\tau$; e.g. $\sigma^{-1}(a) = \{X,S'\}$ is included into $\tau^{-1}(e) = \{X,S,S'\}$. This case does not bring any information from the discrimination point of view : link $(X = S')$ is compatible with τ since $\tau(X) = \tau(S')$.

• or it overlaps several elements in partition $\mathcal{O}_\tau$; e.g. $\sigma^{-1}(b) = \{Y,S,U\}$ overlaps $\tau^{-1}(e) = \{X,S,S'\}$ and $\tau^{-1}(f) = \{Y,T,T',U\}$.

This case does provide some discriminant information : link $(Y = S = U)$ is not compatible with τ since $\tau(Y) = \tau(U) \neq \tau(S)$.

The hypothesis made in 2.2 precisely requires one element at least in $\mathcal{O}_\sigma$ to overlap several elements in partition $\mathcal{O}_\tau$. We call *O-constraint derived from matching* τ, noted $Obj(Ce,\tau)$, the set of pairs of variables (X,Y) such that both variables have the same image according to σ and different images according to τ. This notion is inspired from the discriminant link defined by (Ganascia 87). The point is that this definition directly provides a way of building discriminant terms :

Proposition 0. *Let τ be a matching of $\mathcal{F}$ by Ce, and let X and Y be two variables of $\mathcal{F}$ such that (X,Y) belongs to $Obj(Ce,\tau)$ (i.e. $\sigma(X) = \sigma(Y)$ and $\tau(X) \neq \tau(Y)$). Now let f be the conjunction of predicates (or the unique predicate) of $\mathcal{F}$ determining X and Y. If τ is assumed to be the unique matching of $\mathcal{F}$ by Ce, then term $f \bigwedge (X = Y)$ is maximally general such that it covers Ex and rejects Ce.*

Proof. Term $f \bigwedge (X = Y)$ generalizes Ex by construction. It remains to show that it rejects Ce. In opposition, let τ_f denote a matching of f by Ce. One can extend τ_f by $\tau_\mathcal{F}$, where $\tau_\mathcal{F}$ is a matching of $\mathcal{F}$ by Ce : all predicates in $\mathcal{F}$ appear in Ce. But if there exists a unique matching τ between $\mathcal{F}$ and Ce, then $\tau_\mathcal{F} = \tau$, with $\tau(X) \neq \tau(Y)$ by hypothesis. Moreover, variables X and Y appear in f by hypothesis; hence $\tau(X) = \tau_f(X)$ and $\tau(Y) = \tau_f(Y)$; so $\tau_f(X) \neq \tau_f(Y)$, and this shows that there is no matching between f and Ce compatible with equality $(X = Y)$.

On the other hand, if an element in G includes link $(X = Y)$, it must also include term f that determines variables X and Y, according to 2.2 ; and since term f alone generalizes Ce, conjunction $f \bigwedge (X = Y)$ is maximally discriminant. □

When several possible matchings between $\mathcal{F}$ and Ce are possible, we extend this approach by considering subsets of linked variables. Let us call *discriminant partition of object-variables* a partition $\mathcal{P}$ of $\mathcal{O}$ such that

• for any element $\mathcal{E}$ of $\mathcal{P}$, all variables in $\mathcal{E}$ have same instanciation through σ [2], and

• for any matching τ between $\mathcal{F}$ and Ce, there exist two variables X and Y belonging to a same element $\mathcal{E}$ of $\mathcal{P}$ and with $\tau(X) \neq \tau(Y)$ (so pair (X,Y) belongs to constraint $Obj(Ce,\tau)$).

A variable is called *active* in $\mathcal{P}$ if it does not belong to a singleton of $\mathcal{P}$.

The generalization of $\mathcal{F}$ associated to a discriminant partition $\mathcal{P}$, noted $f_\mathcal{P}$, is defined as the conjunction of predicates determining the variables active in $\mathcal{P}$; the generalization of $\mathcal{L}_\sigma$ associated to $\mathcal{P}$, noted $\ell_\mathcal{P}$, sets that variables belonging to a same element $\mathcal{E}$ of $\mathcal{P}$are to be equal[3]:

$$\ell_\mathcal{P}(\mathcal{O}) = \bigwedge_{\mathcal{E}\in\mathcal{P},\ |\mathcal{E}|>1,\ \mathcal{E}=\{X_{i_1},\ldots,X_{i_{|\mathcal{E}|}}\}} (X_{i_1} = \ldots = X_{i_{|\mathcal{E}|}})$$

Then :

Proposition 1. *Let $\mathcal{P}$ be a discriminant partition of object-variables. Then, term $g_\mathcal{P} = f_\mathcal{P} \bigwedge \ell_\mathcal{P}$ generalizes Ex and rejects Ce.*
Furthermore, if $\mathcal{P}$ is a discriminant partition minimal with respect to the partial order on partitions, then $g_\mathcal{P}$ is maximally general discriminating Ce.

Proof. Follows from proposition 0.

As an illustration, consider positive example Ex and term $\mathcal{F}$ as above, and let Ce be a negative example such that there exist several possible matchings between $\mathcal{F}$ and Ce. :

Ex : P(a,b,c), Q(b,d), Q(a,d), R(b,d).

$\mathcal{F} = P(X,Y,Z), Q(S,T), Q(S',T'), R(U,V)$.

Ce : P(e,f,g), P(g,h,f), Q(f,f), R(g,h).

Partition $\mathcal{O}_\sigma$is :

$\mathcal{O}_\sigma = \{\{X,S'\},\{Y,S,U\},\{Z\},\{T,V,T'\}\}$

There exist two matchings between $\mathcal{F}$ and Ce [4], which respectively derive partitions $\mathcal{O}_{\tau_1}$and $\mathcal{O}_{\tau_2}$:

$\mathcal{O}_{\tau_1} = \{\{X\},\{Y,S,T,S',T'\},\{Z,U\},\{V\}\}$

$\mathcal{O}_{\tau_2} = \{\{X,U\},\{Y,V\},\{Z,S,T,S',T'\}\}$

Associated constraints are :

$Obj(Ce,\tau_1) =$
$\{(X,S'),(Y,U),(S,U),(T,V),(T',V)\}$

$Obj(Ce,\tau_2) =$
$\{(X,S'),(Y,S),(Y,U),(S,U),(T,V),(T',V)\}$

A discriminant partition of object-variables is

$\mathcal{P} = \{\{X,S'\},\{Y\},\{Z\},\{S\},\{T\},\{T'\},\{U\},\{V\}\}$

One sees that corresponding $g_\mathcal{P}$

$$g_\mathcal{P} = f_\mathcal{P} \bigwedge \ell_\mathcal{P} = P(X,Y,Z),Q(S',T'),(X = S')$$

eventually generalizes Ex and rejects Ce.

We notice that $Obj(Ce,\tau_1) \subset Obj(Ce,\tau_2)$; the constraint derived from matching τ_2 thus brings no information from the discrimination point of view, and it can be pruned. This point will be developed later (in 2.7).

[2] So, by definition of the partial order on the partitions of $\mathcal{O}$, one has $\mathcal{P} \leq \mathcal{O}_\sigma$.

[3] It follows from $\mathcal{P} \leq \mathcal{O}_\sigma$ that $\ell_\mathcal{P}$ generalizes $\mathcal{L}_\sigma$.

[4] Tuple (X,Y,Z) can be instanciated by (e,f,g) or (g,h,f).

2.6 DISCRIMINATION BASED ON VALUES

Assume now that term $\mathcal{F}$ only contains valued variables, and that discriminating Ex and Ce is still possible within the chosen formalism. Valued variable $X_{i,j,k}$ takes its values in domain $\mathcal{D}_{i,j,k}$. We restrict ourselves to tree-structured and linear (integer or real-valued) variables.
Following (Mellish 91), every domain $\mathcal{D}_{i,j,k}$ is augmented if necessary, to include an upper bound noted $\top$ (value more general than any value in $\mathcal{D}_{i,j,k}$) and a lower bound noted $\perp$ (value more specific than any value in $\mathcal{D}_{i,j,k}$) ; e.g. in case where $\mathcal{D}_{i,j,k}$ is the space of reals R, value $\top$ is R itself, and $\perp$ is the empty value ϕ.

Set $\mathcal{D}$ is defined as the cross product of domains $\mathcal{D}_{i,j,k}$:

$$\mathcal{D} = \bigotimes_{X_{i,j,k} \in \mathcal{V}} \mathcal{D}_{i,j,k}$$

Consider vector $(\sigma(X_{i,j,k}))$, where $X_{i,j,k}$ belongs to the set of valued-variables $\mathcal{V}$; an abbreviated notation of this vector is $\sigma(Ex)$. To every matching τ between $\mathcal{F}$ and Ce is associated vector $(\tau(X_{i,j,k}))$, denoted $\tau(Ce)$. The discrimination task then reduces to discriminate vector $\sigma(Ex)$ from all vectors $\tau(Ce)$ in propositional logic. The approach proposed in (Sebag 94) thus directly applies. We shall briefly recall this previous work in order to make this paper self-contained.

Example. Let a positive example Ex be a wagon transporting three red circles, and let negative example Ce be composed of 2 wagons, the former transporting seven grey rectangles and the latter one blue oval :
Ex : Wagon (a, red, 3 , circle).
Ce : Wagon (b, grey, 7, rectangle),
Wagon (c, blue, 1, oval).

$\mathcal{F}$ is given by, where all variables except the first one are valued variables :

$$\mathcal{F}(X,Y,Z,T) = Wagon(X,Y,Z,T)$$

Vector $\sigma(Ex)$ is : *(red, 3, circle)*. There are two possible matchings between $\mathcal{F}$ and Ce : τ_1 (respectively τ_2) maps X onto b (resp. c). Accordingly, one has
$\tau_1(Ce)$ = *(grey, 7, rectangle)*, and
$\tau_2(Ce)$ = *(blue, 1, oval)*.
Now, let variable Y (the color of the transported stuff) be valued in a tree-structured domain (Fig. 1) :

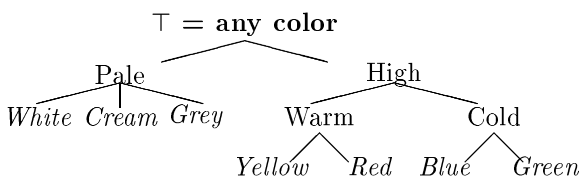

Figure 1 : Domain of variable Y

Variable Z (the number of stuff transported) is integer-valued ; let its domain be [0,5000]. Let variable T (the form of the transported stuff) be valued in a tree-structured domain (Fig. 2) :

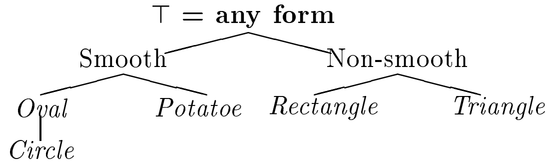

Figure 2 : Domain of variable T

Consider variable Y in order to discriminate $\sigma(Ex)$ and $\tau_1(Ce)$; the most general selector (Michalski 83) covering Ex and rejecting $\tau_1(Ce)$ is $[Y = high]$[5], for *high* color is the most general color encompassing *red* and rejecting *grey*. By similarly considering variables Z and T, we build the disjunction of the most general selectors covering $\sigma(Ex)$ and rejecting $\tau_1(Ce)$:

$$[Y = high] \bigvee [Z = [0,6]] \bigvee [T = smooth]$$

More formally, for any instanciation τ of $\mathcal{F}$ by Ce, for any variable $X_{i,j,k}$, let $Val(Ce,\tau)_{i,j,k}$ denote the most general value[6] covering $\sigma(X_{i,j,k})$ and rejecting $\tau(X_{i,j,k})$ — if such a value exists — otherwise $Val(Ce,\tau)_{i,j,k}$ is set to $\perp$.

We call *V-constraint associated to matching* τ, noted $Val(Ce,\tau)$, the vector of $\mathcal{D}$ whose components are built as detailed above.

So, matching τ_1 gives rise to the V-constraint
$Val(Ce,\tau_1)$ = *(high, [0,6], smooth)*
and similarly
$Val(Ce,\tau_2)$ = *(warm, [2, 5000],* $\perp$*)*
Now, consider vector V = *warm,* $\top$*, smooth)* V is more specific, with respect to one variable at least, than any constraint. We may associate to V a conjunction

$$Wagon(X,Y,Z,T), [Y = warm], [T = smooth]$$

and one verifies this term (wagon transporting stuff with a warm color and a smooth form) is discriminant.

More generally, let V be a subset of $\mathcal{D}$ such that for any matching τ between $\mathcal{F}$ and Ce, there exists a variable $X_{i,j,k}$ with

$$\perp < V_{i,j,k} \leq Val(Ce,\tau)_{i,j,k}$$

V is called a *discriminant vector* ; variables $X_{i,j,k}$ such that $V_{i,j,k}$ is distinct from both $\perp$ and $\top$ are called variables *active* in V.

The generalization of $\mathcal{F}$ associated to vector V, denoted f_V, is defined as the conjunction of predicates determining the active variables of V ; similarly, the

[5] If we restrict to selector '='.

[6] Which is a value if $X_{i,j,k}$ is a tree-structured variable and an interval of values if $X_{i,j,k}$ is a linear variable.

generalization of $Values_\sigma$ associated to V, denoted v_V, is set to the conjunction of selectors $[X_{i,j,k} = V_{i,j,k}]$, where $X_{i,j,k}$ is an active variable of V. Note that the partial order defined on tree-structured domains and the set inclusion defined on the space of real values (or integer values), classically induce a partial order on $\mathcal{D}$. We then have (Sebag 94):

Proposition 2. *Let V be a discriminant vector ; then term $g_V = f_V \bigwedge v_V$ generalizes Ex and rejects Ce.*
Furthermore, if V is a discriminant vector maximal with respect to the partial order defined on $\mathcal{D}$, then g_V is a maximally general term covering Ex such that it discriminates Ce.

2.7 GENERAL CASE

In general, term $\mathcal{F}$ includes both object- and valued-variables. A *global discriminant set* with respect to a negative example Ce, denoted $\mathcal{H} = (\mathcal{P}_\mathcal{P}, V)$ is then defined as a pair of a partition of variables $\mathcal{P}$ and a vector V such that for any matching τ between $\mathcal{F}$ and Ce, one at least among the following conditions holds:

1. there exist two object-variables X and Y such that (X, Y) belongs to constraint $Obj(Ce, \tau)$ (cf 2.5) and (X, Y) is included into an element of $\mathcal{P}$.
2. there exists one valued-variable $X_{i,j,k}$ such that $\perp < V_{i,j,k} \leq Val(Ce, \tau)_{i,j,k}$ (cf 2.6).

Given a global discriminant set $\mathcal{H} = (\mathcal{P}, V)$, the generalization of $\mathcal{F}$ associated to $\mathcal{H}$, noted $f_\mathcal{H}$, is defined as the conjunction of predicates determining variables active in $\mathcal{P}$ or in V. Let $\ell_\mathcal{P}$ be the generalization of $\mathcal{L}_\sigma$ associated to $\mathcal{P}$, and v_V the generalization of $Values_\sigma$ associated to V ; then, term $g_\mathcal{H}$ associated to the global discriminant set $\mathcal{H}$ is set to the conjunction $f_\mathcal{H} \bigwedge \ell_\mathcal{P} \bigwedge v_V$. It comes :

Proposition 3. *Let $\mathcal{H}$ be a global discriminant set. Then term $g_\mathcal{H}$ generalizes Ex and rejects Ce. Furthermore, if $\mathcal{H}$ is such that $\mathcal{P}$ is minimal with respect to the order on the partitions over $\mathcal{O}$ and V is maximal with respect to the order on $\mathcal{D}$, then $g_\mathcal{H}$ is maximally general such that it covers Ex and discriminates Ce.*
Proof. Follows from propositions 1 and 2.

Remark. O-Constraints can be ordered according to the partial order defined on partitions over $\mathcal{O}$; similarly, V-constraints can be ordered according to the partial order defined on $\mathcal{D}$. These partial orders induce a partial order on the matchings between $\mathcal{F}$ and Ce : we define

$$\tau_1 < \tau_2 \iff (\quad (Obj(Ce, \tau_1) \subset Obj(Ce, \tau_2)) \text{ and } (Val(Ce, \tau_1) < Val(Ce, \tau_2)) \quad)$$

It follows from proposition 3 that only minimal matchings according to this order relation, are to be considered when building a global discriminant set.

3 BUILDING G

3.1 CONSIDERING SEVERAL NEGATIVE EXAMPLES

Set $\mathcal{Q}(Ex - Ce)$ was defined (section 2.3) as the set of predicates used to describe Ex and not used to describe Ce. Let $\mathcal{F}^*$ be the generalization of $\mathcal{F}$ obtained by dropping these predicates. By construction, Ce matches $\mathcal{F}^*$ and every matching τ between $\mathcal{F}^*$ and Ce derives an O-constraint (whose elements are pairs of variables in $\mathcal{F}^*$) and a V-constraint (which is a vector of $\mathcal{D}^*$, with $\mathcal{D}^*$ being the cross-product of domains of valued variables occuring in $\mathcal{F}^*$). When several negative examples $Ce_1, \ldots Ce_N$, are considered, one has to merge the different spaces the constraints belong to. The set of object variables in $\mathcal{F}^*$ is embedded into the set of object variables in $\mathcal{F}$; $\mathcal{D}^*$ is embedded into the cross-product $\mathcal{D}$ of the domains of valued variables occuring in $\mathcal{F}$[7].

Now, given positive example Ex and negative examples $Ce_1, ..Ce_N$, a *multi-discriminant set* is defined as a triplet $(\mathcal{Q}, \mathcal{P}, V)$, where $\mathcal{Q}$ is a set of predicates, $\mathcal{P}$ a partition on the object variables, and V a vector of $\mathcal{D}$, such that for any Ce_i, one condition at least among the followings, holds :

1. $\mathcal{Q}$ includes a predicate of $\mathcal{Q}(Ex - Ce_i)$;
2. let $\mathcal{F}^i$ be the generalization of $\mathcal{F}$ obtained by dropping predicates in $\mathcal{P}(Ex - Ce_i)$; then for every matching τ between $\mathcal{F}^i$ and Ce_i,
   - Either there exist two variables X and Y such that (X, Y) belongs to $Obj(Ce_i, \tau)$ and both X and Y belong to some element $\mathcal{E}$ in $\mathcal{P}$;
   - Or there exists a valued variable $X_{l,j,k}$ with $\perp < V_{l,j,k} \leq Val(Ce_i, \tau)_{l,j,k}$.

Given a multi-discriminant set $\mathcal{M} = (\mathcal{Q}, \mathcal{P}, V)$, the corresponding discriminant term noted $g_\mathcal{M}$, is defined as follows. The generalization of $\mathcal{F}$ associated to $\mathcal{M}$, noted $f_\mathcal{M}$, is given by the conjunction of the predicates belonging to $\mathcal{Q}$, and the predicates determining variables active in $\mathcal{P}$ or in V. Let $\ell_\mathcal{P}$ be the generalization of $\mathcal{L}_\sigma$ associated to $\mathcal{P}$, and let v_V be the generalization of $Values_\sigma$ associated to V ; then term $g_\mathcal{M}$ is given as the conjunction $f_\mathcal{M} \bigwedge \ell_\mathcal{P} \bigwedge v_V$, and it follows from Proposition 3 :

Proposition 4. *Let $\mathcal{M} = (\mathcal{Q}, \mathcal{P}, V)$ be a multi-discriminant set with respect to positive example Ex and negative examples $Ce_1, ..Ce_N$. Then term $g_\mathcal{M}$ generalizes Ex and discrimines all Ce_i. Furthermore, if $\mathcal{Q}$ and $\mathcal{P}$ are minimal w.r.t. the set inclusion and the partition orders, and V maximal for the relation*

[7] The embedding is done by completing vectors of $\mathcal{D}^*$ with extra components ; these extra components are set to value $\perp$.

order defined on $\mathcal{D}$, then $g_{\mathcal{M}}$ is maximally discriminant.

Remark. A partial order relation on the negative examples can be derived from the partial order defined on the constraints and on the sets $\mathcal{Q}(Ex - Ce_i)$. An extended definition of a near-miss (Winston 75) is then an example deriving minimal constraints w.r.t. this partial order ; and it is straightforward to show that only minimal negative examples with respect to this order relation are to be considered for defining a multi-discriminant set. This result extends to FOL the pruning proposed in propositional logic by (Smith Rosenbloom 90) in case of a convergent data set, and generalized to the case of a non-convergent data set in (Sebag 94).

3.2 BUILDING G

The multi-discriminant sets (and the corresponding elements of G) are found by a depth-first exploration of the constraints. For every Ce_i, array $M(i)$ stores whether or not discrimination from Ce_i is done by considering predicates $\mathcal{Q}(Ex - Ce_i)$. In the first case, array $p(i)$ stores the currently considered predicate ; otherwise, array $m(i, \tau)$ stores for any matching τ whether the specialization needed to exclude τ is done by considering O-constraint $Obj(Ce, \tau)$ (index $o(i, \tau)$ then stores the considered element of $Obj(Ce, \tau)$) ; otherwise specialization is done by considering $Val(Ce, \tau)$, and index $v(i, \tau)$ tells which component of $Val(Ce, \tau)$ to consider.

Initialization. The multi-discriminant set $\mathcal{M}$ is set to $(\mathcal{Q} = \phi, \mathcal{P} = \mathcal{P}_\perp$[8]$, V = (V_{i,j,k} = \top))$.
Arrays $M(i)$ and $m(i, \tau)$ are set to *True*.
Indices $p(i)$, $o(i, \tau)$ and $v(i, \tau)$ are set to -1.
Loop on negative examples.
For every negative example Ce_i,

1 The previous specialization made according to Ce_i, if any, is removed,

2 Arrays and indices are updated,

3 If constraints derived from Ce_i are not yet satisfied by $\mathcal{M}$, then the corresponding specialization is performed :

3.1 If $M(i) = True$, the $p(i)$-th predicate in $\mathcal{Q}(Ex - Ce_i)$ is added to $\mathcal{Q}$;

3.2 Else, **Loop on matchings** τ between $\mathcal{F}^i$ and Ce_i :

3.2.1 If $m(i, \tau) = True$,
Let $\mathcal{E}_1$ and $\mathcal{E}_2$ be the elements of $\mathcal{P}$ including the two variables in the $o(i, \tau)$-th element of $Obj(Ce_i, \tau)$.
Then update $\mathcal{P}$ by merging $\mathcal{E}_1$ and $\mathcal{E}_2$.

3.2.1 Else, let $X_{i,j,k}$ be the $v(i, \tau)$-th active variable in $Val(Ce_i, \tau)$; then make

$$V_{i,j,k} = V_{i,j,k} \bigcap Val(Ce, \tau)_{i,j,k}$$

Solutions Storage. When all negative examples have been considered, set $\mathcal{M}$ is a multi-discriminant set ; the corresponding term $g_{\mathcal{M}}$ is added to G ; backtracking is performed to find other solutions.

This depth-first exploration finds out all maximally discriminant terms ; however, terms found may not be all maximally discriminant : a condition added to satisfy a given constraint may become useless because of a condition added further on and satisfying also this constraint.

So we have to check whether a term $g_{\mathcal{M}}$ is maximally general or not. This is done by generalizing $g_{\mathcal{M}}$ against $Ce_1, ..Ce_N$ again (the proposed approach applies on formulas, be they completely instanciated or not).

4 CHARACTERIZING G

As shown by (Haussler 88), the number of conjunctive terms in G may be huge. When this happens, an explicit expression of G does not make sense ; but let us show that constraints still enable a tractable classification of further examples.

4.1 CLASSIFYING A FURTHER CASE

Let E be a further example ; let $\mathcal{F}$ be the variabilization of positive example Ex built as in 2.1, and let $\mathcal{F}^*$ be the generalization of $\mathcal{F}$ obtained by dropping the predicates not used to describe E. A computable characterization of G is given by:
Proposition 5.
Example E belongs to $G(Ex, Ce_1, \ldots, Ce_N)$ iff, for every negative example $Ce_i, i = 1..N$, one condition at least among the followings, holds :
(a) One predicate in $\mathcal{Q}(Ex - Ce_i)$ belongs to $\mathcal{F}^*$
(b) There exists a matching σ[9] between $\mathcal{F}^*$ and E, such that for any τ matching of $\mathcal{F}^i$ by Ce_i
(b.1) Either there exists (X, Y) in $Obj(Ce_i, \tau)$ with $\sigma(X) = \sigma(Y)$;
(b.2) Or there exists $X_{l,j,k}$ active in the valued constraint $Val(Ce_i, \tau)$ such that
$\sigma(X_{l,j,k}) \leq Val(Ce_i, \tau)_{l,j,k}$.
Proof.
$\Rightarrow$ The proof is constructive in that that conditions (a), (b.1) and (b.2) allow to build a term covering Ex and E and rejecting $Ce_1, ..Ce_N$. Let g_E be the term defined by the conjunction, for all negative examples Ce_i, of :
• the predicates satisfying condition *(a)* ;

[8] Where $\mathcal{P}_\perp$ denotes the trivial partition composed of singletons.

[9] Same matching σ for all negative examples.

• the predicates determining variables X and Y involved in condition *(b.1)* and the corresponding link $(X = Y)$;
• the predicate determining variable $X_{i,j,k}$ involved in condition *(b.2)* and the corresponding selector $[X_{i,j,k} = Val(Ce_i, \tau)_{l,j,k}]$.
By construction, term g_E covers Ex and E, and rejects $Ce_1, ..Ce_N$. Hence E belongs to G. □
$\Leftarrow$ Conversely, if E belongs to G, there exists a multi-discriminant set $\mathcal{M} = (\mathcal{Q}, \mathcal{P}, V)$, such that E is covered by $g_\mathcal{M}$. Hence, all predicates in $\mathcal{Q}$ appear in $\mathcal{F}^*$ (since E matches $f_\mathcal{M}$) ; similarly, E must satisfy $\ell_\mathcal{P}$; last, E must satisfy v_V ; then, by definition of a multi-discriminant set, conditions *(a), (b.1)* and *(b.2)* hold for E. □

Remark. A matching σ between $\mathcal{F}_E$ and E satisfying *(a), (b.1)* and *(b.2)* eventually induces a structural matching between Ex and E : term g_E built as above generalizes both Ex and E and *ipso facto* defines a correspondence between objects involved in both descriptions. As a by-product, one obtains one or several solutions to the structural matching problem (Ganascia 87, Bisson 90) — any such structural matching being relevant from the discrimination point of view — but not based on any similarity.

4.2 COMPLEXITY

Let I be the number of (distinct) predicates in Ex; checking condition *(a)* then has a complexity $\mathcal{O}(I)$.
Let O be the number of object-variables in $\mathcal{F}$. The size of an O-constraint is upper bounded by $\frac{O.(O-1)}{2}$. For a given matching τ, checking condition *(b.1)* thus is in $\mathcal{O}(O^2)$.
Similarly, let L be the number of valued variables in $\mathcal{F}$; for a given matching τ, checking condition *(b.2)* is in $\mathcal{O}(L)$.

It then remains to bound the number of possible matchings between term $\mathcal{F}$ and a positive or negative example : here lies the exponential complexity. Let P be a predicate having K occurences in the description of Ex, and J occurences in the description of Ce, and assume for a moment that P is the only predicate in the description of Ex. Then, the number of possible matchings between $\mathcal{F}$ and Ce is J^K.

In the general case, let P_i, for $i \in I$ be the (distinct) predicates occuring in the description of Ex, and let K_i (respectively J_i) be the number of occurences of P_i in Ex (resp. Ce) ; then the number of possible matchings between $\mathcal{F}$ and Ce is given by the product $\Pi_{\ i \in I} \ \ J_i^{K_i}$.
Assume the number of occurences of a given predicate in the description of a positive or negative example to be upper bounded by K, then the number of possible matchings between $\mathcal{F}$ and Ce is in $\mathcal{O}(K^{I \times K})$.

Now, if N denotes the number of negative examples, the complexity of the characterization of $G(Ex, Ce_1, ..Ce_N)$ proposed in 4.1 is upper bounded by

$$\mathcal{O}(N \times (I + K^{2.(I \times K)} \times (O^2 + L))).$$

4.3 DISCUSSION

The main factor of complexity in the above expression comes from the number of possible occurences of a predicate in both positive and negative examples. It has to be outlined that the difficulty is general.

In FOIL (Quinlan 93), it is possible to specialize a predicate by considering the possible links between its variables and variables occuring in ancestor nodes ; however, a predicate appearing several times in positive as well as in negative examples is unlikely to be considered, as its information gain should be small. In ML-Smart+ (Botta Giordana 93), a quantity of information related to such possible links between ancestor and current variables is considered ; so ML-Smart+ may face a huge number of possible links, too. However, this eventuality is unlikely, as the candidates for the current node are sorted on the basis of a statistical criterion, and predicates having many occurences in both positive and negative examples are not given a high discriminant score. In KBG (Bisson 90), there is no possibility for an object to instanciate several variables.

A main difference with other discriminant strategies is the following. In algorithms using negative examples such as Golem (Muggleton Feng 90) and Charade (Kodratoff Ganascia 86), the primary goal is to perform structural matching among positive examples. Negative examples are used to sort candidate generalizations or to prune them. In opposition, constraint-based induction (CBI) performs structural matching between positive examples as a by-product of the classification procedure (4.1). The primary goal of CBI is to determine the possible matchings between a positive and a negative example, and to forbid them.

Of course, a balanced strategy would be to combine the structural matching of positive examples on the one hand, and the forbidding of structural matching with negative examples on the other hand.

However, our focusing on negative examples can be justified when
• the concept to learn may be disjunctive — so performing structural matching among positive examples may be misleading.
• there are plenty of negative examples, and few positive examples. This happens for instance when one tries to characterize a failure ; the positive examples (examples of failures) are much less than the negative examples (examples of normal-state).

5 CONCLUSION AND PERSPECTIVES

The originality of the presented constraint-based induction algorithm lies in the combination of three characteristics. First, learning from sparse examples is possible because building the set G of maximally discriminant generalizations leads to both general (hence widely appliable) and redundant (hopefully more robust) rules. Second, the intermediate representation of negative examples provided by the constraints enables the comparison and thus the pruning of negative examples. This point should significantly reduce the search if near-misses are purposely provided by the expert ! Last, it provides means to classify further examples without explicitly building G ; the resulting computational savings can be imagined from propositional logic, where the number of terms in G can be exponential (Haussler 88) while the number of constraints is shown polynomial (Sebag 94).

A major limitation of this algorithm lies in dealing with multiple occurences of a given predicate in both positive and negative examples. Further research will investigate how to reduce the number of matchings to consider ; for instance next matchings could be ascribed to satisfy constraints derived from the previously considered matchings.

Another perspective is to extend control strategy from propositional logic to FOL. For instance, dealing with noisy data could be achieved by considering only a fraction of the constraints generated by negative examples. More sophisticated strategies should involve interacting with the expert in order to filter erroneous data by using heuristics inspired from (Bergadano Giordana 88) and (Sebag 94).

Acknowledgements

I thank Marc Schoenauer, from CMAP, Ecole Polytechnique, for his invaluable and smiling criticism. Thank also to the whole Equipe I & A, LRI, Orsay, and especially to Yves Kodratoff, Celine Rouveirol and Claire Nedellec, for bunches of comments and critics.

References

G. Bisson (1990). *KBG A Knowledge Based Generalizer* 7^{th} International Conference on Machine Learning, B. Porter R. Mooney Eds, Morgan Kaufmann, Austin Texas.

G. Bisson (1992). *Learning in FOL with a similarity measure* 10^{th} AAAI, San Jose.

F. Bergadano, A Giordana, L. Saitta, (1988). *Automated Concept Acquisition in Noisy Environments*, IEEE Trans on Pattern Analysis and Machine Intelligence, PAMI-10, pp 555-578.

F. Bergadano, A Giordana, (1988). *A Knowledge Intensive Approach to Concept Induction*, ICML-88, Morgan Kaufman, pp 305-317.

M. Botta, A Giordana, (1993). *SMART+ : A Multi-Strategy Learning Tool*, in IJCAI-93, pp 937-943.

D. Haussler, (1988). *Quantifying Inductive Bias : AI Learning Algorithms and Valiant's Learning Framework*, Artificial Intelligence, 36, pp 177-221..

Y. Kodratoff, J.-G. Ganascia, (1986). *Improving the Generalization Step in Learning* Machine Learning: An Artificial Intelligence Approach, I, R.S. Michalski, J.G. Carbonnell, T.M. Mitchell Eds, Vol 2, Morgan Kaufman, pp 215-244.

J.G. Ganascia (1987). *AGAPE et CHARADE, deux techniques d'apprentissage symbolique appliquées à la construction de bases de connaissances*, Thèse d'Etat, may 1987, Orsay.

C. Mellish (1991). *The description identification problem*, Artificial Intelligence Vol 52-2, pp 151-167.

R.S. Michalski (1983). *A theory and methodology for inductive learning* Machine Learning: An Artificial Intelligence Approach, I, R.S. Michalski, J.G. Carbonnell, T.M. Mitchell Eds, Springer Verlag, p 83-134.

T.M. Mitchell, (1982). *Generalization as Search*, Artificial Intelligence Vol 18, pp 203-226.

S. Muggleton, C. Feng, (1990). *Efficient Induction of Logic Programs*, Algorithmic Learning Theory, S. Arikawa, S. Goto, S. Ohsuga, T. Yokomori Eds, Springer Verlag, pp 368-381.

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

R. Quinlan, R.M. Cameron-Jones, (1993). *FOIL ; A Midterm Report*, ECML 93, P.B. Bradzil Ed, Springer-Verlag, pp 3-20.

C. Rouveirol, (1990). *Saturation : Postponing Choices for Inverting Resolution*, Proc. 9^{th} ECAI, Pitman, pp 557-562.

B. Smith, P. Rosenbloom, (1990). *Incremental non-backtracking focussing : A polynomially- bounded generalization algorithm for version space*, Proc. National Conference on Artificial Intelligence.

M. Sebag, (1994). *Using Constraints to Building Version Spaces*, ECML-1994, L. de Raedt F. Bergadano Eds, Springer Verlag, pp 257-271.

P.H. Winston, (1975). *Learning Structural Descriptions from Examples* The Psychology of Computer Vision, P.H. Winston Ed, Mc Graw Hill, New York, pp 157-209.