Empirical Bias for Version Space

Jacques Nicolas
IRISA INRIA RENNES
Campus de Beaulieu - 35042 Rennes Cedex - France
Te1 (33) 99 36 20 00

Abstract

The ability to generalize remains one of the central issues of concept learning. A general generalization algorithm -the Candidate Elimination Algorithm- exists but practical applications of this algorithm are still limited, due to its low convergence. The issue has shifted to the design of a useful "bias" limiting the size of the Version Space. This paper proposes a new kind of bias, called empirical bias, and a new general algorithm, ICE, for generalization in presence of bias. This proposition is founded on the concept of focus set, which provides a very flexible way to express expectations or constraints on the space of generalizations.

1. Introduction

We assume the reader to be familiar with the main concepts elaborated by Mitchell [1978] in his work on generalization as a search in a version space (VS). We just want to emphasize in this section the main limitation of the approach and the need for a more constrained algorithm.

1.1 The issue : the bias

The candidate elimination algorithm (CEA) is a double constrained search in a space lying between and represented with two sets of generalizations: a boundary set S of most specific generalizations and a boundary set G of most general generalizations. The main limitation we want to address in this work is the constraint on acceptable generalization languages due to the quadratic dependency of the complexity of the algorithm to the size of the boundary sets. Due to the cost of an elementary treatment for a single instance, the size of S and G becomes too large for applications where the version space contains large "slices", that is large sets of >_uncomparable elements [1] (sec § 2.1). The issue is to impose further restrictions on the shape of the version space.

Several means have been explored in order to express constraints on the class of admissible and inadmissible patterns for generalizations. The main ideas have been to take into account a theory of the domain of learning [Michalski,1983], to introduce a bias on the language of generalization (and/or. on the generalization operators) and then refine it by necessity [Utgoff, 1986], and to specify a more constrained preference criterion [Blumer, 1987; Nicolas, 1988].

1.2 What is needed

All these attempts share the common characteristic to introduce formalized knowledge in the generalization mechanism. Though it is the neatest way to focus the search mechanism, we claim that this represents only one part of the control knowledge. We need a more empirical ability to express constraints, according to the empirical approach chosen in generalization from examples. Concept learning is dedicated to a performance element. This performance element may have some ideas about the shape of the concept definitions it needs. However, it is not coherent to ask for this knowledge only in an abstract way, since it is assumed that the main work of the generalization algorithm is abstraction on the data of the performance element

We illustrate our purpose on several cases.

•Assume the performance element to be a human specialist Even if he does not know the precise definition, nor formalize enough knowledge to help the search of the concept he is looking for, he seldom has no idea of it. He can generally give an approximate definition of this concept This generalization is not consistent in most cases but provides useful hints on the type of generalizations that the specialist is looking for. This may be considered as a form of learning by instruction.

•When the performance element is automated, it may also be the case that approximate generalizations are available.

[1] Note that it only weakly depends on the branching of the partial ordering >.

Our ideas originate in fact from such a situation. We are tackling a "real world" problem, the conceptual clustering of phlebotomine sandflies of French Guiana [Lebbe *et al.*, 1989]. On the one hand, we need a rich representation language which precludes the simple application of the CEA. On the other hand, we use a statistical module which is able not only to build some clusters, but also furnish some clues concerning the form of each cluster characterization. It provides information about attributes involved in the emergence of a cluster which can be combined to build some approximate generalizations.

•Finally, another way to generate approximate descriptions is to use a generalization algorithm with a strong bias on a training set of instances. The bias leads on one side to a search in a high level abstracted space and ensure a quick convergence of this search. On the other side the learned generalizations are rough (they are likely to become inconsistent with the treatment of more instances) and represent only an indication of the desired ones. Note that if we know some means to exploit these approximations, this process may be iterated. For example, we may first restrict the generalization langage to kCNF descriptions, with k fixed, and progressively increase k by a constant step p.

2. The version space revisited

We assume now that approximate descriptions (generalizations) are available as an input for the C.E. algorithm. The issue then becomes : How to extend it in order to make profit of these supplementary data without loosing the good properties of the original algorithm? The two subproblems are: 1) How to formalize an approximation and 2) How to modify the original algorithm. The development of the first point has led to the concepts of focus set and seed set, which are presented in this section. The second point is the subject of section 3.

The focus set is a set of consistent descriptions (approximations). The elements of the seed set may be not consistent.

2.1 Definition of a focus set

We introduce in this paragraph the basic concept of our extension. The Version Space approach requires the initialization of two sets of descriptions, namely the S-set and the G-set. We require the consideration of a new set of descriptions for our algorithm, called the Focus set (F-set). This set is intended to provide a structural bias on the version space. Indeed, a version space may be viewed as a collection of paths between some element of S and some element of G. The focus set restricts the allowed paths by giving intermediate nodes along these paths.

Let us first introduce a preliminary definition.

Definition Let x, y, z be descriptions.
x and y are *$\geq$_comparable* or there exists *a path* between x and y iff $x \geq y \vee y \geq x$. We note $x \lesseqgtr y$
y is *between* x and z (resp. *strictly between*)
iff $x \geq y \geq z \vee z \geq y \geq x$ (resp $x > y > z \vee z > y > x$).
We note $y \in [x \lesseqgtr z]$ (resp. $y \in]x \lesseqgtr z[$)

These definitions may be naturally extended for sets.

The focus set is simply a set of descriptions which have to belong to the initial VS (we say VS *definable),* such that every concept description c (a target generalization) is "attainable" by at least one of these (say fbest). we now make precise this concept of attainability in a VS. The casual reader may skip to the proposition without inconvenience. To be attainable means at least to be comparable with. However, we need a supplementary property, governing the consistency of the whole set. This property will guarantee, during the search for generalizations, that no interaction between two or more elements of the focus set precludes the paths leading from fbest to c to be eventually followed by the search algorithm. For this purpose, we introduce the notion of representativeness. A set of descriptions F is representative of a set of concepts if, when there exists a path from a element of F to a concept, there exists a no-deviation path between these two elements. That is, a set of descriptions F is representative of a set of concepts if, for every path starting at the same element of F, one going through c and the other (the deviation) through another element f of F not comparable with c, there exists a third distinct element of F which is comparable with c and not comparable with f.

Formally, it leads to the following definition.

Definition Let C be the set of target (consistent with any proposable instance and VSdefinable) concept descriptions.
F is *representative* of C iff $\forall c \in C\ \exists f \in F\ \forall f1 \in F$
$f \lesseqgtr c \wedge (f \lesseqgtr f1 \wedge \neg f1 \lesseqgtr c \Rightarrow \exists f2 \in F\ \neg f1 \lesseqgtr f2 \wedge f2 \lesseqgtr c)$
F is a *focus set* iff F is VSdefinable and representative of C.

For instance, consider the version space containing the formulas $p \vee q \vee r$, $p \vee q$, $p \vee r$, p,q ,r, $p \wedge q$, $p \wedge r$, $p \wedge q \wedge r$. Assume C={q} to be the target concept and $f1 \geq f2$ iff $f1 \Rightarrow f2$. Then F={$p \vee q$,p} is not representative of C, because $p \vee q$ is the only element of F comparable with q, $p \geq p \vee q$ and $\neg p \lesseqgtr q$. However, {$p \vee q$, p, q} and {$p \vee q$, $p \wedge q$) are focus sets.

We next state that the focus set is an extension of the boundary set concept (proofs may be found in [Nicolas 91]).

Proposition

1 The S-set S and G-set G of any VS of C are focus sets.

2 Let F be a focus set. Then $S \cup F$ and $G \cup F$ are focus sets.

2.2 Interpretation of a focus set

We now want to show that, if the focus set is an extension of the boundary set concept, the size of the version space itself, with a natural interpretation of this focus set, may only be *reduced* with the input of such a set. This is because it leads to an extension of the constraints on the set of consistent generalizations. We define for this purpose the interpretation of the focus set in terms of a search inside a version space.

We first recall the interpretation of the boundary sets in the Candidate Elimination Algorithm of Mitchell, with the point of view of focus sets. Consider the union of boundary sets $U = S \cup G$ characterizing a version space. It is a focus set (propositions 1&2). Now, the goal of the CEA. may be defined in terms of the maintenance of this focus set with the input of new instances. The updating method is a kind of bidirectionnal search, with the following meaning: in order to maintain one of the boundary sets B, consider all the paths from each inconsistent element b of B to the consistent elements of $U \setminus B$ $\geq$-comparable with b.

We keep this same point of view to describe the search when considering an additionnal focus set F. Then, the set of generalizations to be maintained becomes $U = S \cup G \cup F$. The updating method is a bidirectionnal search, with the following meaning: in order to maintain one of the boundary sets B, consider all the paths from each inconsistent element b of B to the elements of $U \setminus B$ $\geq$-comparable with b *and* "nearest" from b (that is most specific if B=S or most general if B=G). Otherwise stated, we do not explore the paths between two comparable points from the boundary sets but we explore the paths between two comparable points from the focus set. Since the focus set is representative, we ensure that eventually, a path leading to a concept definition will be explored.

We end this section with an illustration of the decrease of complexity of the search on a simple but enlightening example. Assume the language of generalizations to be representable with a boolean lattice of boolean functions with n variables (2^n atoms). The initial version space is characterized with S = {False} and G={True} and contains exactly 2^{2^n} consistent generalizations (these can be built from the variables and the $\vee$, $\wedge$ and $\neg$ connectives). Assume the focus set F to be composed of elements involving only m variables $m \leq n$. Then it is easy to show that the search for consistent generalizations will be focussed such that none of the elements of the inner version space, which is the boolean lattice generated from the n-m other variables, will be explored. That is, the size of the search space is reduced by at least $2^{2^{n-m}}$. A related problem consists of determining how far we can go in the reduction, given a suitable focus set. In our example, the answer is quite simple. The maximum of reduction is reached when there is only one path in the version space to be considered. In the boolean lattice, all paths are 2^n long. So, one can ultimately reduce the size of the search to 2^n, that is, the elements of the focus set itself. More generally, if the focus set contains two adjacent nodes of the lattice, the space is reduced by 2^n-1 elements.

2.3 Definition of a seed set

One of the basic points common to all generalization methods consists of effectively producing the set of concept definitions, starling from a well-chosen set of descriptions and then progressively refining them by application of generalization and specialization operators, in order to achieve the selection criteria. The focus set provides such a set of descriptions.

However, providing a set, whose elements belong to the version space, are representative of the concept descriptions, and differ from S or G, may be too strong a constraint in practice. This motivates the introduction of the seed set. This definition is much less constraining than the previous one. It is intended to formalize the intuitive notion of "generalization pattern". The aim of the seed set is to capture the minimal assumptions to be made about such models or patterns, in order to guarantee that a solution can be reached. The next section provides a definition of the projection of the seed set on the version space, leading to the production of an associated focus set

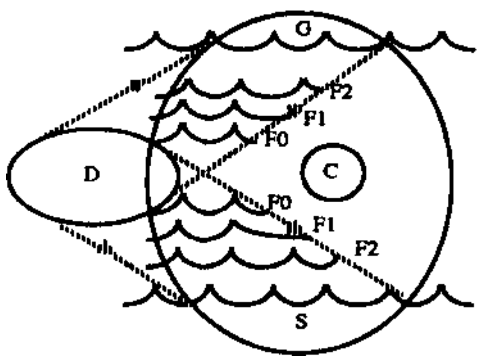
Definition Let D be a set of descriptions
D is a *seed set* iff D and $S_0 \cup G_0$ (initial version space) are comparable.

Note that this condition is not really restrictive, since D may be completed with as many elements of $S_0 \cup G_0$ as necessary to obtain the comparability. The definition no longer requires that the elements of the set be consistent or representative of the concepts. However, even if it does not appear in the definition, recall the basic underlying assumption that elements of the seed set share strong links with the target generalizations.

The issue then is to map the seed set on to a focus set, taking advantage of this assumption. The idea is to keep the same notion of proximity between descriptions as in the generalization algorithm itself. That is, a description d_j is close to a description d if they are comparable and a description d_1 is closer to a description d than a description d_2 if d_1 is between d_2 and d.

2.4 Seed set and focus set

These last definitions rely on the focus set and the seed set concepts in such a way that the seeds are minimally generalized or specialized (closest to their original description). The notion of projection on a version space reflects these minimal refinements. F_0 is proposed first. If it is not sufficient to find the concept C, then the new projection Fj is produced and the algorithm restart from these new approximations, etc...The focus set is defined as the convergence of a sequence of trials F_0, F_1 F_2...(see figure) progressively loosening the strength of the links between the seed set D and its projection (in the direction of S or G), until the boundary S or G is reached.



Definition Let S and G be the boundary sets of a version space. Let D be a seed set.

f is a *projection* of $d \in D$ on (S,G) with respect to a set of descriptions E iff $\forall x \in (]f \leq\geq d]\setminus E)$

f SGdefinable) $\wedge f \leq\geq d \wedge f \notin (E \setminus S \cup G) \wedge$ x not SGdefinable

F is the *closest focus set* from D w.r.t. E iff $\forall$f

$f \in F \Leftrightarrow \exists d \in D$ s. t. f is a projection of d on (S,G) w.r.t. E

The next proposition reflects the desired properties of this definition.

Proposition For every seed set D, there exists a set E s.t the closest focus set F from D w.r.t. E is a focus set

proof: It is sufficient to choose for E the set of all descriptions. In that case, since D and $S_0 \cup G_0$ are comparable, the projection reduces to $S_0 \cup G_0$, which is a focus set

However, this choice for E represents a worst case, where the search occurs in the whole initial version space. The seed set is not pertinent, i.e., it conveys no additional information with respect to $S_0 \cup G_0$. The idea of the algorithm is to start from a closest focus set F where $E = \emptyset$ ("nearest" from D) and in case of failure, to restart with E:=F. More generally, each time a failure occurs, E is replaced with the union of the previous E and the closest focus set w.r.t this previous E.

Note that this search for the closest focus set can not be stated as a search in a version space, as there arc no instances involved here.

3 The Informed Candidate Elimination algorithm

We designed a generalized version of the CE A, able to take into account the seed set as a new source of knowledge (this is not concurrent with the inclusion of other types of control knowledge). The CEA has two interesting properties that we need to preserve. First, its strategy is a bidirectionnal search. And second, if a concept is SoGo-definable (i.e., in the original version space), then it will be found eventually. Since we have no idea of how accurate are the elements of the focus set for the concept at hand, the algorithm has to include a "repair" strategy, loosening the bias until the production of consistent generalizations for the whole set of instances. In fact, if the focus set is of no help for the definition of the concept, the behaviour of the algorithm converges towards that of the CEA.

The general schema of the algorithm remains unchanged with respect to Mitchell's work. However, taking into account the duality of S and G and in order to emphasize the parameters and components of the generalization process, we present a parameterized form of this algorithm. A more detained algorithm may be found in [Nicolas, 1991].

```
proc Informed_Candidatc_Elimination (S0,G0,F0)
 S:=S0; G:=G0; F:=F0; reset-instances;
 until single solution or no solution or no more instances
 %(S=G)∧singleton(S) or S=Ø ∨ G=Ø or end_of_file%
 do
  I:= readjnstances;
  case type(I) in
 positive —> remove_in_boundary (incomplete, G)
             update_boundary(incomplete, S)
             remove__equivalent_or__notj)iefeiTed (S)
 negative —» remove_in_boundary(weakly__inconsistent, S)
            update_boundary(weakly_jnconsistent,G)
            remove_equivalent_or_not_jjrefemed (G)
end case end untill
 exit no generalization: if F0=Ø
 %unsucessfull search in the whole version space SO,GO
           then print no generalization
           else Fl:=repair_focus_set(FO,SO,GO);
              Informed_Candidate_Elimination(SO,GOJFI)
 exit single generalization: print learning completed
 exit no more instances: print current version space
end proc
```

The CEA requires the specification of two types of transformations on generalizations : most specific generalizations and most general specializations. Note that our algorithm requires the definition of two new simplified types of transformations on descriptions during the repair strategy. The specification of a projection is indeed a search for most specific generalizations and most general specializations, but it does not involve any instance. An illustration of these new operators is given in the next section.

4. Example: a feature value intervals space

This problem derives from the one described in [Mitchell, 1978]. It must be clear that it remains didactic.

The goal is to define a concept, whose characteristics are attributes taking their values in the domain of integerss. An instance is a vector (conjunction) of attributes with particular values. Assume here that the descriptions are represented with only two attributes. Following the single representation trick, instances arc expressed in the same langage as generalizations. Assume the chosen representation to be:

$\langle concept\rangle \to (\langle interval\ (x)\rangle, \langle interval\ (y)\rangle)$

$\langle interval\ (x)\rangle \to \langle int\rangle \langle op\rangle\ x\ \langle op\rangle \langle int\rangle$

$\langle int\rangle \to integer\ (\in Z)\ |\ -\infty\ |\ +\infty \qquad \langle op\rangle \to\ <\ |\ \leq$

For example, the instance (5,3) is represented as $(5 \leq x \leq 5, 3 \leq y \leq 3)$. The matching predicate and the order of specificity between concepts is simply the inclusion relation between couples of intervals. For example, the concept $(3 \leq x \leq 7, 2 \leq y \leq 3)$ is more specific than $(-1 \leq x \leq 7, 0 \leq y < +\infty)$. Due to space restrictions, we can not give the diagrams illustrating the behaviour of the algorithm. Intermediary and final results of the algorithm are presented with the previous syntax, but we strongly encourage the reader to draw a graphical representation of the concepts.

Assume the following scenario for the initialization part.

The algorithm reads a first positive instance (1,1).

G is the singleton $\{(-\infty < x < +\infty, -\infty < y < +\infty)\}$

S is initialized to the singleton $\{(1 \leq x \leq 1, 1 \leq y \leq 1)\}$

Furthermore, we dispose of two approximative models, leading to the seed set

$D=\{(1 \leq x \leq 1, 2 \leq y \leq 3), (0 \leq x \leq 1, 0 \leq y \leq 1)\}$

Then, a procedure computes the closest focus set from D and leads to the following result (notation: each element of the focus set has the form <identification number, description, list of immediate predecessors, list of immediate successors, list of comparable elements of S, list of comparable elements of G>):

$$F=\{\langle 1, (1 \leq x \leq 1, 1 \leq y \leq 1), \emptyset, \{3,4\}, \{1\}, \{2\}\rangle,$$
$$\langle 2, (-\infty < x < +\infty, -\infty < y < +\infty), \{3,4\}, \emptyset, \{1\}, \{2\}\rangle,$$
$$\langle 3, (1 \leq x \leq 1, 1 \leq y \leq 3), \{1\}, \{2\}, \{1\}, \{2\}\rangle,$$
$$\langle 4, (0 \leq x \leq 1, 0 \leq y \leq 1), \{1\}, \{2\}, \{1\}, \{2\}\rangle\ \}$$

Elements 1 and 2 of F are just the elements of S and G. The other elements are the least definable generalizations f of each element of D, that is, such that $s \geq f \geq g$.

We describe now the behaviour of the algorithm with the input of instances in the following order (+ for positive instances, - for negative instances):

1) -(3,3), 2) +(2,2), 3) -(0,0), 4) +(3,2), 5) -(4,1)

The computation gives for S et G, after step (instance) 2:

$S = \{(1 \leq x \leq 2, 1 \leq y \leq 3),, (0 \leq x \leq 1, 0 \leq y \leq 1)\ \};$

$G = \{(-\infty < x \leq 2, -\infty < y < +\infty), (-\infty < x < +\infty, -\infty < y \leq 2)\}$

S has "absorbed" the focus points and contains fairly general concepts. After step (instance) 3 :

$S = \{(1 \leq x \leq 2, 1 \leq y \leq 3)\};$

$G = \{(1 \leq x \leq 2, -\infty < y < +\infty), (-\infty < x \leq 2, -\infty < y \leq 2)\}$

Since the next positive instance is not included in any element of G, the algorithm find no generalization at step 4. The procedure repairjbcus.set then computes a new focus set. In this example, assume the generalization operator of the projection is replacing one of the bounds of the elements in the focus set with the corresponding bound of a comparable element of the boundary sets. We are sure that the process converges in at most 4 calls for the repair procedure (one for each bound in a couple). Each time, instances are considered from the beginning.

The new focus set contains the following concepts:

$$F=\{(1 \leq x \leq 1, 1 \leq y < +\infty), (1 \leq x \leq 1, -\infty \leq y \leq 3),$$
$$(1 \leq x < +\infty, 1 \leq y \leq 3), (-\infty < x \leq 1, 1 \leq y \leq 3),$$
$$(1 \leq x \leq 1, 0 \leq y \leq 1), (0 \leq x \leq 1, 1 \leq y \leq 1)\ \}$$

With this new focus set, learning may be completed after step 5 and leads to the following boundary sets:

$S=G=\{(1 \leq x \leq 3, 1 \leq y \leq 2)\}$

The convergence of the algorithm is remarkable in this case, where we only need 6 instances. Without D, the S set would be unchanged, but the G set would be

$G = \{(1 \leq x \leq 3, -\infty < y \leq 2), (-\infty < x \leq 3, 1 \leq y \leq 2)\}$

5. Related work

Many studies involving the bias problem have been developed since Mitchell emphasized the issue [Mitchell, 1980], The common goal shared by all these studies is to hasten the convergence of the boundary sets. If one more deeply investigates the existing methods, one can divide them further in three main directions:

1) Definitions of constraints or properties on the set of preferred generalizations. Bias is included in the selection criteria. Michalski handles for this purpose a Lexicographic Evaluation Function concept [Michalski, 1983], to be defined at a user level. At a theoretical level, we proposed some general logical criteria [Nicolas, 1988] for the selection of generalizations when the representation language includes negations and disjunctions.

2) Multiplication of the effect of an instance. Bias is included in a theory. Hirsch's approach [1989] consists of using a theory able to determine for some instances their type

(positive or negative). This theory allows the generalization of the explained instances with an EBG technique and to generate in such a way multiple instances of the same type and same explanation of their type. A related technique is the perturbation [Kibler and Porter, 1983], wich automatically explores the "nearest" instances from the observed instance.

3) Change of the representation language. Bias is included in the language. Utgoff's algorithm [Utgoff, 1986] initially searches for generalizations in a restricted space defined with a restricted language. In case where no generalization is found, the language is augmented with new terms which are linked to the previous one so as to produce a generalization. Russel and Grosof [Russel and Grosof, 1990] use a determination knowledge base, expressing dependencies between concepts, to build a pertinent version space for the concept to be learned.

Our approach may be viewed as a fourth direction, striving towards the inclusion of bias in the search control sets which are the boundary sets. One of the basic assumptions of learning from examples is that the knowledge is easily accessible at a specific, "instanced" level, but hard to produce at an abstracted, "theoretical" level. Our proposition is based on the same assumption. We require the bias at an empirical level.

Rendell's classification of biases [Rendell, 1987] considers three kinds of biases. Fixed bias is defined once for all, independently of any application. Parameterized bias is adjusted for some application by the user. Variable bias may evolve during learning, according to its results on some particular problems. The kind of bias we propose is basically a parameterized one. However, the focus set may be not only user defined, but also automatically generated from the results of another module (like a data analysis tool). This initial bias may be then refined, if no satisfying generalization is found. Consequently, the bias possesses also a variable characteristics.

6. Conclusion

We have proposed a general framework enabling the development of further studies in the field of constrained generalization algorithms. It shifts the limitations on "blind" inductive learning [Dietterich, 1989] to the issue of building interesting approximations of the concept to be learned.

The worst case complexity remains unchanged in terms of S and G but it has to be defined in terms of the structure of the focus set or, even better, in terms of the structure of the seed set A basic interesting property is that the growth of the focus set is solely dependent on the growth of boundary sets: the other elements can just be discarded by the algorithm. The other main issue that remains to be explored is the design of an algorithm optimizing the adaptation of the bias (that is, choosing the new elements in the focus set) when the ICE algorithm fails to produce a generalization. A related question occurs with the design of the seed set. Actually, it seems reasonnable not to produce this set once for all, but iteratively, with progressive refinements towards more and more detained approximations.

References

[Blumer *et al.*, 1987] A. Blumer, A.Ehrenffeucht, D. Haussler, M. Warmuth *Occam's Razor* Information Processing Letters 24 1987 pp377-386.

[Dietterich, 1989] T. Dietterich *Limitations on Inductive Learning* Proc. 6th Machine Learning Jun 1989 Cornell Univ. Ithaca N.Y.ppl24-128.

[Hirsh, 1989] H. Hirsh *Combining Empirical and Analytical Learning with Version Spaces* Proc. 6th Machine Learning 89 June 1989 Cornell Univ. Ithaca N.Y.pp29-33.

(Lebbe *et al.*, 1989] J. Lebbe, I.C Lerman, J. Nicolas, P. Peter, R. Vignes *Conceptual Clustering in Biology : Application and Perspectives* Proc. of the Conference on Data Analysis, Learning Symbolic and Numeric Knowledge Antibes Sept 1989 Diday Ed. Nova Science Pub.

[Michalski, 1983] R. Michalski *A theory and methodology of inductive learning* Machine Learning : an Artificial Intelligence approach Tioga 1983.

[Mitchell, 1978] T. Mitchell *Version Spaces : an approach to concept learning* PhD dissertation. Standford University. Dec 1978.

[Mitchell, 1980] T. Mitchell *The need for bias in learning generalizations* T.R. CBM-TR-117 Rutgers University Dept of Comp. Science 1980.

[Mitchell, 1982] T. Mitchell *Generalization as search* Artificial Intelligence 18 pp 203-226 1982.

[Mitchell *et al.,* 1983] T. Mitchell, R.Keller, S. Kcdar-Cabelli *Explanation Based Learning : a Unifying View* Machine Learning 1.

[Nicolas, 1988] J. Nicolas *Consistency and Preference Criteria for Generalization Languages handling Negation and Disjunction* Proc. of the European Conference on Artificial Intelligence, Aug 88 Munehen.

[Nicolas, 1991] J. Nicolas *An Informed Generalization Algorithm* Intern. Rep INRIA/IRISA Rennes May 91.

[Kibler and Porter, 1983] D. Kibler, B. Porter *Perturbation : A Means for Guiding Generalization* Proc. of UCAI 83 pp 415-418.

[Rendell 87] L. Rendell *More robust concept learning, using dynamically variable bias* Proc. of 5th ML87 Irvine.

[Russel and Grosof, 1990] S. Russell, B. Grosof *A sketch of Autonomous Learning using Declarative Bias* Machine Learning, Meta-reasoning and Logics P. Brazdil & K. Konolige eds Kluwer Academic Pub. 1990.

[Utgoff, 1986] Utgoff *Machine Learning of Inductive Bias* Kluwer Academic Pub. 1986.