Multiple-Knowledge Representations in Concept Learning

Thierry Van de Merckt & Christine Decaestecker

IRIDIA, Université Libre de Bruxelles
Av. Franklin Roosevelt 50, 1050 Brussels, Belgium.
Phone: +32.2 - 650 31 69 Fax: +32.2 - 650 27 15
{THVDM, CDECAES@ULB.AC.BE}

Abstract. This paper investigates a general framework for learning concepts that allows to generate accurate and comprehensible concept representations. It is known that biases used in learning algorithms directly affect their performance as well as their comprehensibility. A critical problem is that, most of the time, the most "comprehensible" representations are not the best performer in terms of classification! In this paper, we argue that concept learning systems should employ Multiple-Knowledge Representation: a *deep* knowledge level optimised from recognition (classification task) and a *shallow* one optimised for comprehensibility (description task). Such a model of concept learning assumes that the system can use an interpretation function of the deep knowledge level to build an *approximately* correct *comprehensible* description of it. This approach is illustrated through our GEM system which learns concepts in a numerical attribute space using a Neural Network representation as the deep knowledge level and symbolic rules as the shallow level.

1. Introduction

Concept Learning has much evolve during the last decade. Referring to the goal stated by Michalski in 1983 [Michalski 83] Conceptual Inductive Learning "designates a type of Inductive Learning whose final products are *symbolic descriptions* expressed in high level, *human oriented* terms and forms." Besides the implicit goal of inductive learning, i.e., to produce a theory that is able to explain observed facts and to make correct predictions about unseen cases, Concept Learning also relies on a strong cognitive motivation: the system should be able to express the underlying theory (the concept) under a human-understandable language, which is by essence symbolic. Therefore, concept descriptions have been biased in a way that is close to human way of *understanding and explicating concepts*, that is, by using symbolic descriptions under the form of logical-based languages for Nominal attributes or under the form of intervals (producing orthogonal hyper-rectangles) for continuous ones. Some well-known examples are AQ [Michalski 83], Decision Trees [Quinlan 86a] and Decision Lists [Clark & Niblett 89]. This goal on the representation of induced theories was a characteristic that drew a clear frontier between AI Concept Learning and any "classifier" algorithm issued from Statistical Inference or Pattern Recognition techniques.

Nowadays many subsymbolic algorithms like Neural Networks [Hertz & al. 91], Exemplar-based [Aha & al. 91] and Prototype-based [Kohonen 90; Decaestecker 93] are actively investigated by the machine learning community. Concurrently, symbolic algorithms make an increasing use of techniques issued from Statistical Inference or

Pattern Recognition to improve some aspects of concept recognition: Bayesian Trees [Buntine 89] and Flexible Concept Matching [Esposito & al. 91; Bergadano & al. 92; Van de Merckt 92] are some examples of this trend. A major problem of these algorithms is that it is no longer easy to get the semantic of the knowledge used to encode the concept membership function: they use *interpretation* functions of the encoded knowledge under the form of complex matching mechanisms where the semantic of the concept is (partially) encoded in real valued parameters. In this case, *semantic* means to have a comprehension on how instances are allocated or not to a particular concept. Hence, most of them do not produce a concept *description* in "human-oriented terms and forms" any more. It is clear then that the actual trend of many works done in Machine Learning does not reflect the original definition of Concept Learning. However, these algorithms were developed to answer some important weaknesses of early concept learning systems with respect to classification accuracy. From one side, most early symbolic systems produced *crisp* concept descriptions, i.e., descriptions under the form of explicit concept boundaries that discriminate classes in the description space (see AQ [Michalski 83], ID3 [Quinlan 86a] or CN2 [Clark & Niblett 89]). Whilst more difficult learning tasks have been attacked, it became clear that these algorithms entailed strong limitations, especially regarding graded concepts [Aha & al. 91], noisy and incomplete data [Aha & al. 91; Esposito & al. 91; Van de Merckt 92], and complex concepts [Bergadano & al. 91; Michalski 90]. From another side, it became clear that the relation between biases implemented in the algorithms and their efficiency to infer correct hypothesis is crucial [Utgoff 86; Benjamin 90]. Hence, Brodley speaks about "selective superiority" among different algorithms and concept domains [Brodley 93].

A critical problem of concept learning lies in satisfying two conflicting goals. From one side, one wants the algorithms to produce simple and human-understandable descriptions, which imposes strong (cognitive) constraints. From the other side, one wants them to reach high levels of classification accuracy, which requires one to use complex (as far as human comprehensibility is concerned) knowledge representations. How these two goals can be reconciled? Simply by assuming that an agent might possess multiple-knowledge representations on the same problem. In this paper we present a new approach to Concept Learning, called the Two-functional Model, which is based on this idea. In this framework, we present a system called GEM, which uses a Neural Network to optimise a concept representation using a prototype-based representation and which produces symbolic descriptions reflecting "its knowledge" of the target concept. Section 3 presents our neural-symbolic system. Section 4 presents some experimental results using GEM. Section 5 makes a quick review of closer related works. Section 6 identifies limitations and future works.

2. Concept Learning Revisited: the Two-Functional Model

The basic idea of the TF model is that symbolic descriptions, as far as concepts are concerned, is a characteristic of human beings and hence, results from our high linguistic skill to communicate *what we have in mind.* It does not mean however, that what we communicate is equivalent to the complex knowledge encoded in our brain. Therefore communicating complex concepts, such as *friendship,* entails some kind of reduction in both complexity and efficiency by introducing a human understandable language and cognitive description biases, as simplicity, that constitute a common

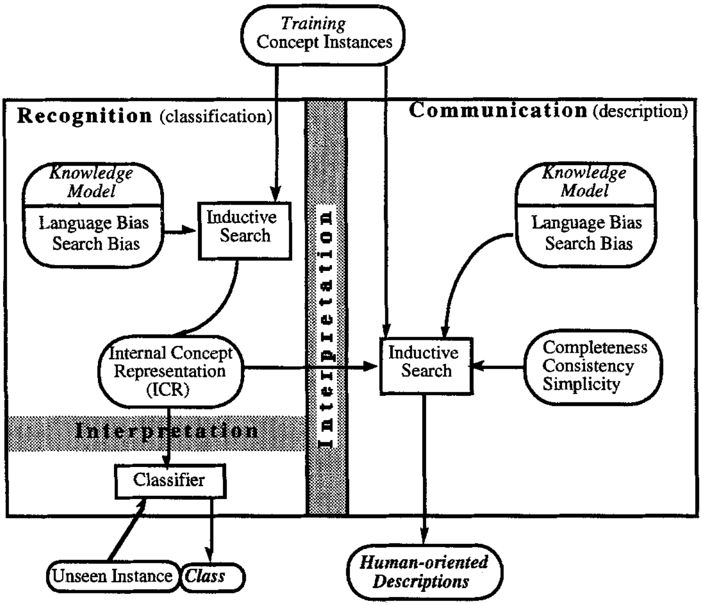


Fig. 1: A flow-chart of the Two-Functional Model of Concept Learning

semantic ground shared by all people. Therefore we argue that there is no need to merge both functions (recognition and description) into one single concept representation as it is done by most symbolic approaches. On the contrary, an *internal concept representation* (ICR), based on any knowledge model[1], should be optimised regarding recognition efficiency without conflict of its further description which is performed by a distinct system that produces a (good) approximation of it. In the Two-Functional model (TF) a "Concept Learner" entails two distinct parts (see Fig. 1):

(i) **Recognition** - It is the "deep knowledge" level which results from an inductive learning process whose bias focus on accuracy of the concept representation. This bias may include any kind of Background Knowledge that helps the system to choose a specific knowledge model regarding the domain (as done by [Brodley 93]) or that encodes *a priori* domain theory that generates constraints on the possible concept representations. The resulting representation, called the *Internal Concept Representation* (ICR), is further used by a classifying function which aims to recognise instances from non-instances of the concept;

(ii) **Communication** - It is the "shallow knowledge" level using human-oriented descriptions that reflect the concept encoded by the ICR. It results from an exploration of the ICR guided by biases focusing on cognitive aspects of the descriptions, such as background knowledge that provides preferences and constraints on the descriptions or that help the system to choose among possible search techniques regarding the type of knowledge model encoded by the ICR. As it is the case for human beings, preferences may entail parameters like

[1] By Knowledge Model we mean the way the knowledge is encoded (prototype, examplar, DNF, NN, etc.) and its interpretation function (matching used for classification).

> completeness ("tell me more about it"), consistency ("be more precise") and simplicity ("I don't care about the exceptional cases"). Because its bias may be very different from those of the ICR, many different descriptions consistent with the ICR and the background knowledge might exist and hence, the communication function performs an inductive search in a space of possible descriptions.

In early symbolic systems, both parts were merged into one single algorithm where the ICR also stood for the human-oriented description. It should be noted that the TF model does not provide a framework for generating symbolic descriptions *independently* of the recognition function. On the contrary, the link between the ICR and the inductive description engine in Fig. 1 entails an interpretation of the semantic content of the ICR (it will be explained in details later) and this interpretation defines the class of classifying functions that could be easily implemented within a particular implementation of the TF model.

Although this model is inspired from Cognitive Science, it offers two major advantages resulting from the clear separation between recognition and communication. The first one is that the classification function may be optimised regarding accuracy by using many kinds of powerful techniques that throw off the yoke of cognitive biases linked to human-oriented descriptions (in our case we use a Neural Network (NN) for that purpose). This allows to get rid of the compromises between accuracy and understandability of complex concepts that are always to be chosen in single concept representation algorithms [Stepp & Michalski 83; Iba & al. 88]. The second advantage is that starting from an optimised ICR and explicitly introducing cognitive biases to generate concept descriptions allow to evaluate the cost of introducing these biases. Indeed, by looking at the loss of accuracy due to their introduction one may *know the cost of being explicit and human understandable* and hence, to evaluate the adequacy of the description bias regarding the target concept. This is not the case of most Symbolic and several Neural Net algorithms [Tshichold & al. 92; Goodman & al. 92] which directly produce biased class descriptions also used for classification.

3. The Hybrid Neural-Symbolic GEM System

GEM has been designed to work in continuous attribute spaces in which cognitive biases encoded in symbolic algorithms are especially constraining. Indeed, many algorithms (ID3-like, AQ's or CN2) produce descriptions under the form of orthogonal hyper-rectangles (whose edges are perpendicular to the description axis). It is well known that this representation may be inadequate for many domains, leading to poor descriptions from a cognitive as well as from a recognition point of view. Many recent algorithms using other kinds of knowledge models like Instance-Based Learning [Aha & al. 91], Prototypes [Decaestecker 93], Neural Trees [Samkar & Mammone 91; Utgoff 88] or Neural Networks [Hertz & al. 91] achieve better results in these domains. Therefore, the whole potential of the TF approach may be particularly highlighted for such numerical-featured concepts. To illustrate the behaviour of GEM, we will use a two-class problem defined in a two-dimension space as shown in Fig. 2, named the Diamond problem. The instances are uniformly distributed in a square of side 30 and allocated to the classes following the decision surface drawn in the figure. A training set of 400 instances has been used.

3.1 The Recognition Function

In GEM, the recognition function uses a Prototype-based *Knowledge Model* implemented through a neural network (a detailed presentation may be found in [Decaestecker 93]).

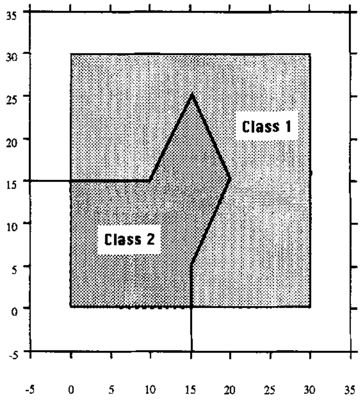


Fig. 2: The Diamond Problem

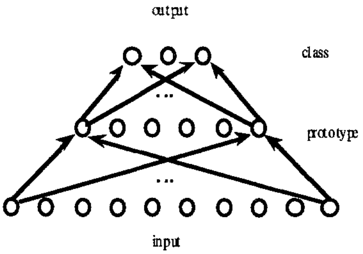


Fig. 3: Network architecture

The Knowledge Model - NNP uses a three-layer, fully connected, feedforward net (Fig. 3). The hidden layer stands for a set of Prototypes whose locations are to be optimised. The weights of the input-to-hidden units are the prototype vector descriptions (location in the pattern space). The hidden-to-output weights are binary and *fixed*: they indicate the class of each prototype. Only the weights of the hidden units are trained. NNP globally optimises the location of prototypes in order to minimise the classification error rate. Hence, the vector descriptions of prototypes are adapted through a gradient procedure which minimises an original error function. A deterministic annealing procedure is introduced to avoid local minima and to distribute the prototypes in each class. The whole optimisation process is biased in order to minimise the resubstitution error rate with a *minimum number of prototypes* (simplicity bias). Hence, redundant prototypes (a prototype is redundant if all its covered instances may be correctly classified by other prototypes of the same class) are eliminated. Hence, the remaining ones are "forced" to cover the largest area of the instance space. At the end of the optimisation, the ICR is constituted by the list of prototypes and its interpretation function for doing classification becomes a simple nearest neighbour rule applied on their locations. Thus the ICR produces Piecewise-linear decision boundaries.

Bias - A major bias of NNP results from the fact that it belongs to Piecewise Linear Classifiers. Indeed, each prototype defines implicitly a convex decision surface by the means of the nearest neighbour competition (see Fig. 4). The set of all individual prototype's decision surface realises a partition of the input space (Voronoï Diagram) and hence, the ICR is always complete but may be partially inconsistent. Besides this language bias, the inductive search algorithm applies a *global optimisation* process focusing on the minimisation of the resubstitution error rate by modifying the prototypes' location. An important bias of NNP is its hill-climbing search for generating *simple* ICR (redundant prototypes elimination): this process imposes a *greatest generalisation* strategy by forcing each prototype to cover the largest area in the instance space, allowing the algorithm to better handle noisy data.
Empirical evaluations of NNP show that it generates highly performing ICR in terms of: (i) classification accuracy and simplicity (small number of prototypes); (ii) regularity (small standard deviations when tested on many different random training sets) and (iii) robustness against noise and domain dependency (even for highly non-

linear concepts). For detailed results see [Decaestecker 93] and [Van de Merckt & Decaestecker 94].

On the Diamond problem, NNP produces an ICR which consists of 6 prototypes. In Fig. 4, the classification boundaries resulting from the prototype's location and from the nearest neighbour competitive process is compared to the underlying target concept boundary. The difference between the real frontier and the decision surface produced by NNP results from the lack of training instances in some places of the instance space.

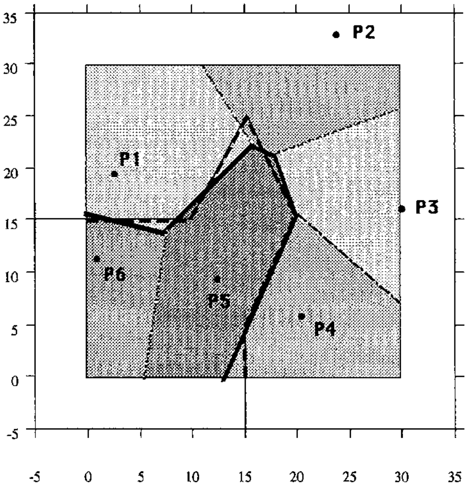


Fig. 4: NNP ICR on the Diamond problem

ICR Interpretation - It is essential, in the framework of the TF model, to have a clear interpretation of the semantic content of the ICR. Indeed, this knowledge must be available to the communication function to generate descriptions that correctly reflect the knowledge encoded in the ICR. The semantic interpretation of the ICR relies upon the identification that each prototype implicitly draws (or covers) a *convex* decision surface in the instance space, called the *Prototypical Region* (PR). Because of NNP's bias for simplicity, we assume that each PR is necessary to approximate the concept's class membership function and hence that, regarding their instance space convexity, each PR stands for a disjunction of the target concept. Therefore, to produce a symbolic description of the concept underlined by the ICR, a description of each individual PR will be searched, each of them being the description of a *disjunct* of the concept.

3.2 The Communication Function

The problem of "understanding" a piece of knowledge representation (as a list of prototypes' location, symbolic descriptions or a matrix of weighted connections of a NN) is related to its *interpretation*, i.e. the function that *gives a protocol on how to use the knowledge*. For NNP, the interpretation is a nearest neighbour rule that entails a competitive process between all prototypes and hence, getting a clear view of the encoded concept's *shape* is a complex calculation problem, as far as human being is concerned.

The Form of Human-oriented Descriptions - The communication function identifies the *classification boundaries* encoded by the ICR and produces a *crisp* description of them. Crisp descriptions have been chosen because they have a "self-contained" meaning: they take the form of DNF implication rules where the precondition part entails disjunctive sets of conjunctive predicates and where the conclusion part specifies the resulting class.

Bias - Each PR of the ICR is approximated by a set of closed geometrical figures. There are three basic biases used by GEM in its search for symbolic descriptions: (i) the shape of the crisp geometrical figures used to approximate each PR; (ii) the use of a "disjunctive view" instead of a "class view" when searching for descriptions; (iii) the assumption that noise has been correctly treated by the recognition function when producing the ICR.

(i) *Symbolic language* - To approximate a PR drawn by the ICR, GEM uses orthogonal hyper-rectangles under the form of intervals defined over the instance space. This approach is widely used by Symbolic Learning systems (see for example ID3 [Quinlan 86a], AQ [Michalski 83] or Nearest Hyperrectangles [Salzberg 91]) because of their natural understanding: an orthogonal box can be easily represented by a conjunctive rule where each term tests a cut-point value of an attribute.

(ii) *A Disjunctive view when searching for descriptions* - The descriptions rely on the interpretation given to PRs which assumes that each of them represents a typical disjunct of the target concept. It is a considerable advantage of the prototypical approach to produce an ICR allowing such a nice interpretation of the distinct areas of its decision surface, as shown in Fig. 4. The target concept is described by approximating each PR individually. As a result, the inductive search focuses on disjunctive terms (the PRs), each one being considered as a distinct new class (called the ICR_class in the algorithms), instead of focusing on real classes given by the training set (called the Training_class). As a consequence, each Training_class represented by *n* prototypes will be described by at least *n* disjunctive terms, one for each PR (see in Fig. 5). The whole concept description is then simply the union of all disjunctive rules. However this bias is *relaxed when looking at near boundary regions of two adjacent PRs of the same class*. Indeed, prototypes also implicitly draw boundaries separating two adjacent PRs of the same class (called adjacent disjunctive PRs) whose locations are arbitrary. Therefore, describing the PRs individually accommodates a relaxing facility near adjacent disjunctive PR boundaries.

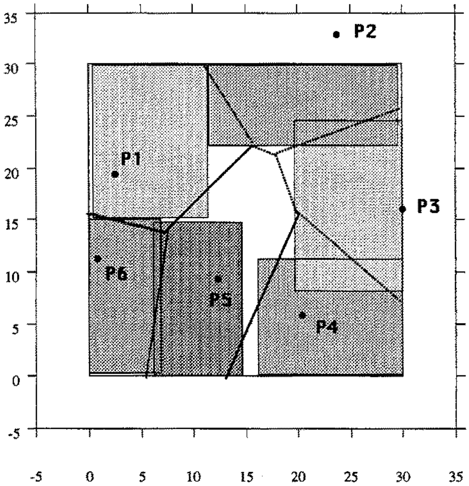


Fig. 5: The ICR and the resulting discriminant descriptions

(iii) *Noise treatment* - A basic assumption of the TF model is that the ICR is better optimised regarding classification accuracy than the symbolic descriptions that are produced using cognitive biases (as orthogonal concept boundaries) and hence, that the optimised ICR avoids noise overfitting. In GEM, noise overfitting avoidance is implemented by the simplicity bias described above. Therefore, training instances that are covered by a PR of a distinct class (inconsistent) are considered to be noisy and are hidden to the description inductive search. As a consequence, the symbolic algorithm relies on the recognition function for the treatment of noise and hence, will not incorporate procedures for taking noise into account.

Preference Criteria - Given these considerations, the inputs of the symbolic description engine are the ICR and the training set. In order to produce symbolic descriptions, a number of other preferences must be decided on: (i) the *simplicity* of the concept description or, in other words, the level of approximation of the ICR's decision surface; (ii) the *consistency* and the *completeness* of the descriptions.

(i) *Simplicity* - Because the language bias differs among the ICR and the description (Piecewise linear versus orthogonal boundaries), more than one hyper-rectangles might be used to correctly approximate the decision surface drawn by one single Prototype. The number of hyper-rectangles needed to correctly approximate one single PR depends on the adequacy of the description's language bias (orthogonality) regarding the target concept. Thus, the number of disjunctive *rules* describing a single PR depends on a preference criterion for simplicity that fixes the maximum number of hyper-rectangles that will be used to approximate one single PR. Using this criterion, one may favour the simplicity of a concept description (sacrificing its consistency or its completeness) or one may ask for complete and consistent descriptions. Once a simplicity level has been chosen, consistency or completeness has to be fixed. In the current state of GEM implementation, there are two simplicity levels available: one box (hyper-rectangle) per prototype and a free number of boxes, which results in producing "perfect" approximations of the target concept.

(ii) *Consistency and Completeness* - Given a level of simplicity, consistency and completeness are related: once a level of consistency for the descriptions has been chosen, the level of completeness is given as a result of the inductive search and inversely. Descriptions may therefore be oriented towards characteristic (100% complete) or discriminant (100% consistent). Any level between 0 and 100% may be asked to the system for consistency or completeness.

Bias Evaluation - Given a level of simplicity and a 100% consistent preference, the level of completeness gives information on the adequacy of the symbolic language towards the domain. Indeed, a PR is a *convex* region in the instance space and hence, if the concept boundaries are orthogonal, one single hyper-rectangle should adequately approximate a PR. If it is not the case, by increasing the complexity of the concept description, one may be able to obtain consistent descriptions that are more complete. In fact, increasing complexity is a mean to produce "closer" complete and consistent descriptions and hence, simplicity is no more a bias to avoid overfitting, as usually in symbolic learning, but *it stands for adjusting biased descriptions (hyper-rectangles) to the underlying shape of the target concept.*

3.3 The Description Algorithm

Two different algorithms have been implemented, one that produces a description with a simplicity level of one box per prototype, and another one, based on ID3 [Quinlan 86a], to produce descriptions of unconstrained complexity (called the free-complexity algorithm). Two important processes are common to the two algorithms: the *Filtering* process that implements the bias related to the treatment of noise and the *Re-Labelling* process that implements the Disjunctive view bias.

The Filtering Procedure - Given the training set, the ICR and its interpretation function, this function eliminates from the training all wrongly covered instances with respect to the ICR. It then returns a list of PR-instances organised into clusters, one per prototype. These filtered ICR_clusters will be further used to build the symbolic description of the target concept.

The Re-Labelling Procedure - This process creates a new attribute for each filtered instance, called the ICR_class, that indicates to which PR they belong. ICR_clusters and ICR_class are used by the algorithm to implement the Disjunctive view bias.

The One-complexity Algorithm - This algorithm follows a bottom-up approach with a *least generalisation strategy*. A PR may be described by two extreme boxes: a *complete Hyper-Box* (complete-HB) and a *discriminant Hyper-Box* (discriminant-HB). The complete-HB is the *smallest* hyper-rectangle covering all PR-instances (least generalisation strategy) and the discriminant-HB is the *largest* consistent hyper-rectangle included in the complete-HB, i.e., which covers the largest part of training examples belonging to the PR-class while covering no training examples of another class (greatest generalisation strategy within complete-HBs). Once the ICR_clusters have been built, producing a complete-HB for one PR is straightforward. It consists in the list of intervals defined by the minimum and maximum values observed among PR-instances for each attribute. In case of scarce training sets, the least generalisation strategy may produce uncovered instance space regions of known class regarding the ICR: the in-between regions of two adjacent disjunctive PRs (of same class). Therefore, complete-HBs are slightly extended, in each direction, towards the closest instance belonging to an adjacent disjunctive PR if it exists (no new negative instances should be included in the box extension). It should be noted that complete-HBs may share large overlapping areas and hence that they may include instances from another class or from an adjacent disjunctive PR, as Fig. 6 shows. The complete-HB is the starting point of the algorithm. To generate an HB of consistency χ, a deflation of the current complete-HB is done by the *Deflate-HB* procedure. Several HBs included into a Complete-HB can be χ-consistent while differing by their cover. Hence, the algorithm performs a hill climbing search biased to *maximise the cover* in terms of completeness and volume (to keep a maximum of positive examples as well as a maximum of the initial complete-HB volume). Starting from the complete-HB and the filtered ICR_clusters, it iteratively searches to shrink the complete-HB along one single direction (an attribute generates 2 directions) to obtain the consistency or completeness level asked for. The iterative procedure excludes *one single* negative example per step. To choose among all possible directions, two

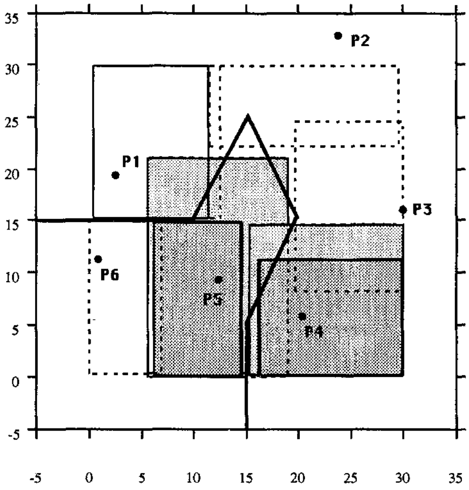


Fig. 6: Effect of the deflation process on the Diamond problem. Dashed-line boxes are complete-HBs where the deflation process didn't shrink the box. For P5 and P4, the darker boxes (discriminant) compared to the lighter ones (complete) show the effect of the deflation process.

heuristics are used: the direction that minimises the number of positive instances excluded is chosen and in case of equality, the one that minimises the reduction in volume is chosen. A *positive* example, regarding an HB, is an instance which shares the same class value (Training-class) than the box (which inherits its class value from its associated PR). Therefore, instances belonging to adjacent disjunctive PRs are considered as positive by the deflation heuristic, relaxing the Disjunctive view bias at near frontier regions. Fig. 6 shows the results of the one-complexity algorithm on the Diamond problem: the complete-HBs and 100%-consistent (discriminant-HBs) are presented. It shows that the deflation is high in the areas *where the orthogonal bias is inadequate to approximate the concept boundary*: P4 and P5 regions are good examples of this. In other regions, like in P2 and P3 where there is (nearly) no contact with the Diamond boundary, or like in P6, where the contact involves an orthogonal boundary, the bias is adequate and hence, the deflation process had nothing to do, leaving discriminant- and complete-HBs being identical. It can be seen from this example that discriminant descriptions may leave large uncovered instance space areas in the neighbourhood of those inadequate regions (see Fig. 5). In the following description of class 2, issued by GEM on this problem, the discriminant description is only about 60% complete, 40% are lost in the area inside the diamond:

```
Characteristic description (simplicity 1):
   2 Prototypes
   P5  (x ⊂ [5.5 19.0] ∧ y ⊂ [0.0 21.5])    (cover 79%; consist 87%)
   P6  (x ⊂ [0.0  7.0] ∧ y ⊂ [0.0 15.0])    (cover 21%; consist 100%)
   ≡ Class2
Discriminant description (simplicity 1):
   2 Prototypes
   P5  (x ⊂ [5.5 14.7] ∧ y ⊂ [0.0 15.0])    (cover 40%; consist 100%)
   P6  (x ⊂ [0.0  7.0] ∧ y ⊂ [0.0 15.0])    (cover 21%; consist 100%)
   ≡ Class2
```

The Free-complexity Algorithm - This algorithm can produce complete and consistent descriptions of the concept encoded in the ICR (with respect to the training of course). It uses a Decision Tree technique similar to ID3 [Quinlan 86a] but in this case the training set has been first Filtered and Re-Labelled. Unlike the one-complexity algorithm, this one follows a top-down approach and uses a *greatest generalisation strategy*. A decision tree is grown on the filtered instances *using the ICR_class*, that is, a partition of the PRs is produced. After this first stage, the relaxation of the Disjunctive bias is done by a simple pruning mechanism: a subtree is pruned if all its children nodes are leaves of the same Training_class. Indeed, due to its top-down search strategy, two leaves of the same Training_class represent a specialisation in a near border region of two adjacent disjunctive PRs. As result,

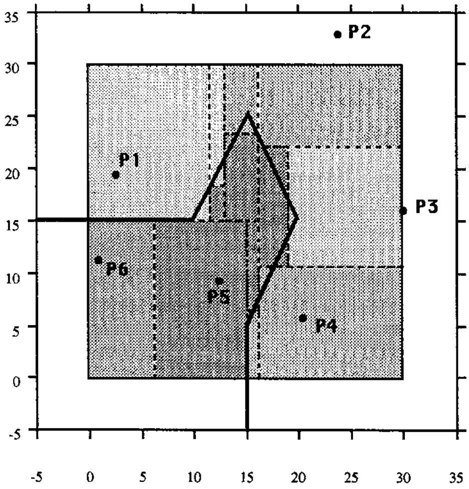


Fig. 7: A description of the Diamond problem with a free complexity

this pruning mechanism merges small hyper-boxes of the same real class value that have been separated due to the Disjunctive view bias. Each leaf is viewed as a disjunctive rule describing a PR. A rule is the conjunction of the predicates tested at each node on the path from the root to the leaf. Fig. 7 shows the result obtained on the Diamond problem. It can be seen that most of the consistent boxes found by the one-complexity algorithm are also produced by the free-complexity one (P1, P4, P5 and P6), despite the drastic change in the search method used. This is a result of the Disjunctive view bias used in both methods and the small number of dimension of the instance space. As a consequence, the main tendency provided by each prototype is preserved and represented by the largest box associated with it, while smaller ones are specialisations in complex shape regions: it can be seen that P1, P4 and P5 recover the "lost" regions due to the deflation process used to generate the discriminant-HBs (compare Fig. 6 and Fig. 7). An advantage, illustrated by this example, of the Disjunctive view bias in GEM is that we know that each main disjunction (resulting from a prototype) is a convex region and therefore that *each internal disjunctive rule (in a PR) is not a real disjunctive decision region in the instance space but rather is an artificial disjunctive term resulting from the inadequacy of the description language towards the domain.*

4. Empirical Evaluation

Experiments did not focus on the efficiency of the recognition function (ICR generated by NNP) of GEM. This aspect only concerns the classification part of the system that has been largely tested against other algorithms and has proven its high performance [Decaestecker 93; Van de Merckt & Decaestecker 94]. Instead, the experiences have been done in order to evaluate the claims concerning the advantages of GEM in the framework of the TF model.

The first aspect concerns the treatment of noise through the Filtering process, the question being: is the ICR Filtering reliable regarding noise? To evaluate this point, we have checked that (i) the ICR does not make noise overfitting and (ii) that the Filtering process is more active when noise is present than when it is absent. This may be evaluated by looking at the number of prototypes generated in the absence versus the presence of noise and the corresponding number of instances filtered out of the ICR_clusters. The second aspect concerns the capacity to evaluate the adequacy of the description bias regarding the target concept. The key point is the observation of the consistency of complete-HBs or its dual aspect, the completeness of discriminant-HBs produced by the one-complexity algorithm. A rating that entails this aspect has been designed: the *Bias Cost* is the difference of the resubstitution omission rate (computed *on the training set*) between discriminant-HB and complete-HB. If, at a given level of simplicity, this measure is too high, the system could propose to increase the complexity level in order to obtain a better approximation of the target concept encoded in the ICR. Our tests aim to evaluate the validity of such an analysis. The last aspect concerns the evaluation of the quality of the descriptions as a mean to communicate the concept encoded in the ICR. To evaluate this point, we used the descriptions as classification rules and compared their results to the classification performed by the ICR on the same data sets. These tests aim to appreciate how the descriptions may represent the central tendencies of the concept and localise "safe" classification areas in the instance space. These aspects have been evaluated along three dimensions: (i) domain dependence: seven different data sets are used presenting different concept shapes, sometimes adapted (Square data) and sometimes not adapted (Geometrical, Wave and Diamond) to the orthogonal bias of symbolic descriptions; (ii) noise dependence: each data set has been tested before and after noise addition; (iii) scarcity of training sets dependence: tests have been done on small and large training sets. The experiments only widely tested the one-complexity algorithm. However, the effects of using the free-complexity algorithm will be commented when appropriate.

4.1 Experimental Set-up

There are two real world data sets and five artificial ones that have been chosen in order to evaluate the TF model against various concept shapes:

- *Iris*: it contains 3 classes of 50 instances each, where a class refers to a type of Iris plant.
- *Diabetes*: it contains 145 records of 3 different diagnostics for Diabetes (the class repartition is C1=26 , C2= 35, C3=84 instances) based on 5 numerical attributes representing clinical tests.
- *Diamond Data*: it is the two-class problem presented at Fig. 2.
- *Wave Forms* [Breiman 84]: it is composed of 3 classes, each of them being a linear combination of three distinct wave forms. Each instance is composed by a vector of 21 continuous values.
- *Geometrical Data*: it is a two-class problem defined in a two-dimension space. The classes are delimited by two circles (centre: (0,0); diameter 20 and 40) entailed in a square (side 60) (see Fig. 8). Class 1 is represented by grey areas and class 2 by the white ring. Instances are uniformly distributed over the whole surface of the square.

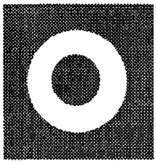


Fig. 8: Geometrical

- *Gauss-Square Data*: it is a three-class problem shown in Fig. 9. Instances in each class are artificially generated by Gaussian distributions (several by class). The centre of each Gaussian is shown by a black triangle in the figure; the standard deviations are relatively small. The instances are attributed to the class corresponding to the nearest centre which have been chosen in order to generate orthogonal implicit decision boundaries (black lines in Fig. 9). The centre of the Gaussian can be considered as optimal prototype's location.
- *Uniform-Square Data*: it is the same as the previous problem where the instances are uniformly distributed in the square and allocated to the class following the decision surfaces showed in Fig. 9.

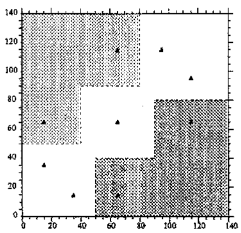


Fig. 9: Square data

Noise Addition - Each data set was tested before and after noise addition. For Diabetes and Iris, a Gaussian noise N(0,σ) on each attribute (with σ equals to 1/2 the standard deviation of the whole population for this attribute) has been artificially added. For Geometrical data, noise was introduced by an overlapping between the clusters. For Wave Forms, a Gaussian noise N(0,1) on each attribute has been added like in [Breiman & al. 84]. The same process has been made on Diamond and Uniform-square.

For Gauss-square, each instance has been reallocated following the location of the centre of the Gaussian that generated it: this process introduces overlapping between the classes.

Training & Test Sets - For each data set, 10 runs were done with two training sets of distinct sizes (small and large). Large sets were built using the small ones by *adding* a number of randomly chosen *new* instances. Table 1 indicates the number of elements in each set. Except for Iris and Diabetes, the test set was generated independently.

Table 1: Sizes of training and test sets.

Data Set	Training	sets	Test set
	small	large	
Iris	20% per class	50% per class	the rest
Diabetes	20% per class	50% per class	the rest
Geometrical	115	575	1000
Wave	10 per class	100 per class	5000
Diamond	100	400	1000
Gauss-square	130	390	1000
Uniform-square	130	390	1000

4.2 Noise Treatment

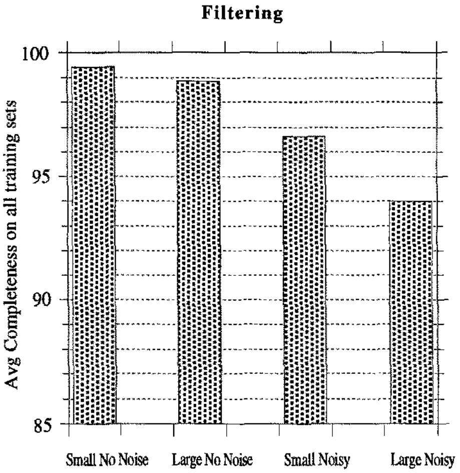
On average, the results obtained from NNP (see [Decaestecker 93]) are: (i) ICR complexity (the number of prototypes) increases very little when moving from small to large sets and (ii) complexity on noisy versions of the data sets are slightly less than on noise-free ones. These results (and others largely analysed in [Van de Merckt & Decaestecker 94]) show that the ICR produced by NNP does not cause overfitting. Concurrently to this general tendency to produce less complex ICR when noise is present, it can be seen in Fig. 10 that the effect of the Filtering procedure works as expected: the percentage of training instances provided to the description algorithm decreases proportionally to the presence of noise and to the size of the training set.

4.3 Evaluation of the Description Bias

The average Bias Cost over all data *for large sets* is presented in Fig. 11. In this chart, the two first bars present the average Bias Cost on noise-free *training* and *test* sets respectively and the two last ones present the same figures after noise addition. This chart shows that the difference in omission between discriminant versus complete descriptions is nearly the same on training and test sets and that this property is observed independently of the presence of noise in data. This result has a strong practical implication: it means that the adequacy of the description bias and hence, the complexity that should be used to correctly approximate the target concept, may be validly evaluated *on the training set*, even in case of noisy data. However, this result should not be misinterpreted: it doesn't mean that the observed level of omission on the training gives a reliable approximation of its level on the test set. On the Geometrical problem, for example, the Bias Cost is about 6% on large noise-free sets, meaning that if the system produces a description under the form of discriminant-HBs, it "looses" a cover of about 6% on the concept instances, but the level of omission of a discriminant description on the test set is about 12% (while complete-HBs omission is 6%). The estimation of the real omission rate depends on the statistical representativeness of the training and hence, the adequacy of description bias should be carefully evaluated regarding the size of the training compared to the dimension of the instance space: the fewer instances we have, the smaller space covered and hence, simple descriptions may appear to correctly approximate the target concept although they don't. When using the free-complexity algorithm, the adequacy of the description bias may be evaluated by looking at the number of disjunctive rules (leaves) necessary to approximate a single PR.

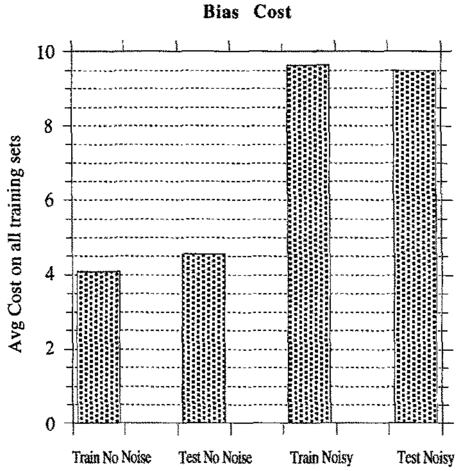
4.4 Evaluation of the Fidelity of the Descriptions

In GEM, a symbolic description should "reflect" the concept encoded in the ICR. This means that, given a level of detail asked by the user through the simplicity parameter, a description *should allow to easily identify the major classification areas entailed in the recognition function.* Therefore, the quality of descriptions may be evaluated by comparing classification results of the descriptions with the ICR's ones *on the test sets.* Fig. 12 presents two bar charts for each type of training, averaged over all data sets. The first bar presents, from bottom to top, the percentage of correct classifications, the omission rate (no decisions) and the error rate (% of incorrect classification) of *discriminant* descriptions produced by the *one-complexity* algorithm. Three main observations may be done from this chart: (i) on average (except on small noisy ones) *simple* descriptions correctly cover a large part (at least 70%) of the concept; (ii) the error



Filtering	Small no N.	Large no N.	Small Noisy	Large Noisy
Iris	98.67	96.27	95.00	89.47
Diabete	99.66	100.00	91.72	93.38
Geometrical	99.65	98.73	94.87	88.75
Wave	100.00	100.00	100.00	94.23
Diamond	99.40	97.50	97.10	95.35
Gauss-square	99.54	99.82	98.85	97.82
Unif-square	99.08	99.64	98.92	99.23

Fig. 10: Effect of the *Filtering* procedure



Bias Cost (Large sets)	Train no N.	Test no N.	Train Noisy	Test Noisy
Iris	0.55	1.86	9.71	10.27
Diabete	5.92	4.59	7.83	5.10
Geometrical	5.87	6.59	9.51	11.98
Wave	0.00	0.00	23.05	20.09
Diamond	15.87	18.32	14.17	16.13
Gauss-square	0.16	0.20	2.18	1.84
Unif-square	0.21	0.20	0.98	0.99

Fig. 11: Effect of deflation on omissions

rate of the descriptions is always less than the ICR, particularly on small training sets (however, at the cost of high omission rate); (iii) the omission rate depends on the size of the training and on the level of noise.

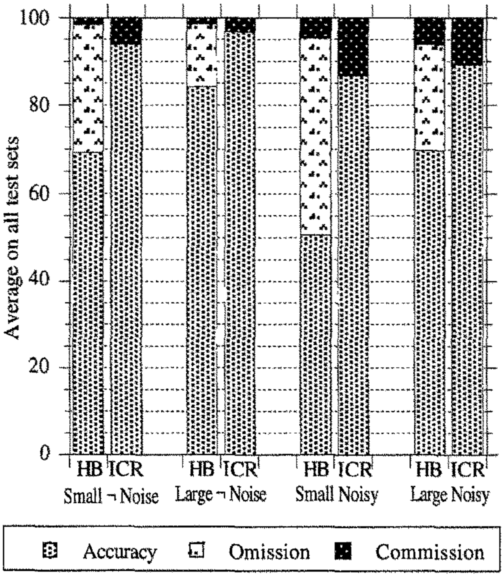


Fig. 12: Average results of 100%-consistent HBs

(i) By "simple" concept description, we mean a description that contains a small number of disjunctions covering a large number of concept instances. In this case, since one box has been used to approximate each single PR, the description is to most simple one GEM can produce. The fact that such simple descriptions correctly cover a large part of the concept is due to the Disjunctive view bias of GEM. On the Diamond problem for example, class 1 could not be correctly described with a higher simplicity than 4 convex regions and hence, the description algorithm produces 4 hyper-rectangles approximating these regions.

(ii) The difference among error rates of the ICR and the descriptions may be explained by their opposite generalisation strategies. The one-complexity algorithm is biased by a least generalisation strategy in order to produce "safe" descriptions adversely to the recognition function which uses a greatest generalisation strategy. In this latter case, when the concept is only partially represented by the training (due to scarce or noisy data) the inductive algorithm does not have enough data in some instance space regions and performs "best guess" generalisation that mainly relies on its *a priori* bias (Piecewise linear and simplicity), resulting in higher chances to perform errors. These results confirm that the descriptions, while being more or less incomplete, depending on the size of the training and the level of noise, correctly capture the major semantic trend of the concepts.

(iii) The least generalisation strategy has the "drawback" of producing incomplete descriptions, depending on the training size and the level of noise in the data. The level of omission is also affected by the adequacy of the description language for the target concept. On the Diabetes data, for example, the omission on small sets (not noisy) is 49% (error is 1%) while the ICR makes 98% of correct recognition. On large training, the level of omission decreases to 29% (error is still 1%) while the ICR is 99% accurate. It is clear in this case that the description bias is inadequate to approximate the concept boundary with an equivalent simplicity as the ICR. Clearly, the Diabetes concept is a good candidate for the free-complexity algorithm.

In conclusion, the symbolic descriptions correctly reflect the semantic content of the concept encoded by the ICR and the level of "correctness" reflects the statistical representativeness of the training set and the adequacy of the description language.

5. Related Works

The idea of coping with accuracy and comprehensibility in the same time is not new. Knight & Gil [91] proposed an architecture for problem-solving composed of an efficient "reasoner" (the problem-solver), such as a NN, and a "rationalizer" which aims to explain the output of the reasoner. However, both modules are completely separated and the rationalizer is a tool for *convincing* a user to accept the solution rather than for *explaining* how its has been reached. The closer work is certainly the one of Towell & Shavlik [93]. They use a special NN architecture (Knowledge-Based NN) in order to improve knowledge given under the form of a set of rules. After optimisation, a new and more accurate set of rules is extracted from the network. Their method differs from GEM in many respects: (i) they use KNN which encode horn clauses describing the domain whilst we use a prototype-based representation; (ii) their algorithm is restricted to discrete (nominal) features whilst GEM is restricted to numerical features; (iii) they don't use the training set to guide the interpretation of the knowledge encoded in the network whilst GEM makes an intensive use of it to constraint its interpretation. However, their algorithm fits the TF model where the recognition function uses a KNN whose accuracy has been empirically demonstrated and the communication function uses m-of-n type of rules.

6. Conclusions

We have presented a hybrid Neural-Symbolic Learning algorithm which implements the Two-Functional model of Concept Learning. This algorithm uses two different inductive engines that use two different knowledge representations: one for building an Internal Concept Representation optimised regarding accuracy, and the other for producing symbolic concept descriptions optimised for comprehensibility. This Multiple-Knowledge Representation schema has shown several advantages over Single-Knowledge Representation concept learning algorithms. Firstly, no compromise should be made concerning accuracy and/or comprehensibility. From the classification point of view, the inductive learning may be optimised without the interference of "human-oriented" biases. From the concept description point of view, stronger cognitive biases may be used (as accepting omission due to a least generalisation strategy). Secondly, the adequacy of biases used to produce concept descriptions may be evaluated regarding the target concept. This allows one to know the cost of being explicit and human understandable. Thirdly, regarding concept descriptions, completeness, consistency and simplicity become "real" preference parameters, since they should not be "optimised" to avoid noise overfitting.

We hope that the TF model approach will provide a framework for integrating many different classifier algorithms as well as to help developing new approaches for generating comprehensible concept descriptions. Indeed, in GEM we have used simple DNF-like rules for describing a concept. However, many different types of descriptions could be produced like m-of-n decision rules (like in [Towell & Shavlik 93]) or a mix among decision rules and typical examples that could be more understandable to an expert than a set of rules. The advantage of the TF model is that the type of descriptions that might be generated could depends on contextual factors such as the level of expertise of the user or explicit preferences for one kind of description among several available ones. From the recognition side and from a theoretical point of view, GEM's TF model implementation could be applied to a whole set of classification functions defined by their ability to account for the two essential biases of the system: (1) Noise Treatment by a Filtering process and (2) Disjunctive view by a Re-Labelling process. However, in practise, GEM benefits from the prototypical knowledge model used by the ICR which performs *generalisation over the instance space.* Using lazy learning algorithms like exemplar-based algorithms would cause problems to apply the Disjunctive view bias. Other kinds of neural networks, such as those using back-propagation, would also cause a problem since these NN creates a single non-linear decision surface for each class (although Towell and Shavlik have open promising ways for KNN). Therefore, GEM's implementation of the TF model may not be applied to any classifier algorithm without extensive work. However, we believe that the idea of the .TF model, i.e., the separation between the knowledge used for prediction and explanation and the "interpretation bridge" between them, could be further explored in order to integrate powerful subsymbolic learning algorithms in the framework of "comprehensible" concept learning.

Our close future work will extend the description algorithm to any complexity level and will better evaluate the performance of GEM with respect to its capacity to communicate the semantic content of the ICR by a closer analysis of the effect of gradual increase of noise as well as gradual increase of the complexity of the tested domains (by increasing the number of dimensions). In a second stage we will also investigate how to extend our approach to mix nominal-numeric attribute spaces as well as how to introduce a feature selection process in NNP and/or in the one-complexity description algorithm.

References

Aha W. David, Kibler D., Albert K. M. (1991) Instance-Based Learning Algorithms, *Machine Learning vol.6, n° 1, January 1991,* Kluwer Academic Publishers.

Bergadano F., Esposito F., Rouveirol C. and Wrobel S. (1991) Evaluating and Changing Representation in Concept Acquisition, *Proceedings of the European Working Session on Learning*, Springer Verlag

Bergadano F., Matwin S., Michalski R.S., Zhang J. (1992) Learning Two-Tiered Descriptions of Flexible Concepts: The POSEIDON System, *Machine Learning vol.8, n° 1,* Kluwer Academic Publishers.

Benjamin Paul D. (Ed) (1990) *Change of Representation and Inductive Bias*, Kluwer Academic Publishers.

Brodley Carla E. (1993) Addressing the Selective Superiority Problem: Automatic Algorithm/ Model Class Selection, *Proceedings of the Tenth International Conference on Machine Learning ML'93*, Morgan Kaufmann.

Buntine Wray (1989) Learning Classification Rules using Bayes, *Proceedings of the Sixth International Workshop on Machine Learning ML'89.* Morgan Kaufmann.

Clark P. and Niblett T. (1989) The CN2 Induction Algorithm, *Machine Learning Vol.3 n°4, March 1989*, Kluwer Academic Publishers.

Decaestecker C. (1993) NNP: a neural net classifier using prototypes, *Proceedings of the IEEE International Conference on Neural Networks*, San Fransisco.

Esposito Floriana, Malerba Donato and Semeraro Giovanni (1991) Flexible Matching for Noisy Structural Descriptions, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence IJCAI'91*, Morgan Kaufmann.

Goodman R.M., Higgins C.M. & Miller J.W. (1992) Rule-based neural networks for classification and probability estimation, *Neural Computation Vol.4 n° 6.*

Hertz J., Krogh A. & Palmer R.G. (1991) *Introduction to the theory of neural computation,* Addison-Wesley.

Knight K. and Gil Y. (1991), Automated Rationalization, *Proceedings of the First International Worshop on Multistrategy Learning,* Ed. by R.S. Michalski and G. Tecuci, Center of Artificial Intelligence, George Mason University.

Kohonen T. (1990) The Self-Organizing Map, *Proceedings of the IEEE, vol. 78, N° 9.*

Iba W., Wogulis J., Langley P. (1988) Trading Off Simplicity and Coverage in Incremental Concept Learning, *Proceedings of the Fith International Conference on Machine Learning ML'88,* Morgan Kaufman.

Michalski Ryszard S. (1983) A Theory and Methodology of Inductive Learning. *Machine Learning, An Artificial Intelligence Approach.* Ed. by Ryszard S. Michalski, Jaime G. Carbonell and Tom M. Mitchell, Tioga Publishing.

Michalski Ryszard S. (1990) Learning Flexible Concepts: Fundamental Ideas and Method Based on Two-Tiered Representation, *Machine Learning: An Artificial Intelligence Approach Vol. III.* Edited by Y. Kodratoff and Ryszard S. Michalski , Morgan Kaufmann.

Quinlan J.Ross (1986a) Induction of Decision Trees. *Machine Learning Vol 1, n°1,* Kluwer Academic Publishers.

Salzberg Steven (1991) A Nearest Hyperrectangle Learning Method. *Machine Learning vol. 6, n° 3, May 1991,* Kluwer Academic Publishers.

Samkar A. & Mammone R.J. (1991) *Neural Tree Networks. Neural Networks, Therory and Applications.* R.J. Mammone & Y. Zeevi Eds, Academic Press.

Stepp Robert E. and Michalski Ryszard S. (1983) Conceptual Clustering: Inventing Goal-oriented Classification of Structured Objects, *Machine Learning, An Artificial Intelligence Approach volII.* Ed. by Ryszard S. Michalski, Jaime G. Carbonell and Tom M. Mitchell, Morgan Kaufmann.

Towell G. G. and Shavlik J. (1993) Extracting Refined Rules from Knowledge-Based Neural Networks, *Machine Learning, vol. 13, n° 1,* Kluwer Academic Publishers.

Tschichold N., Ghazvini M. and Diez D. (1992), M-RCE: a self configuring ANN with rule extraction capabilities, Proceedings of the International Conference on Artificial Neural Networks ICANN'92, Brighton.

Utgoff Paul E. (1986) *Machine Learning of Inductive Bias.* Kluwer Academic Publishers.

Utgoff P.E. (1988) Perceptron Trees: A case Study in Hybrid Concepts Representations, *Proceedings of AAAI-88.*

Van de Merckt T. (1992) NFDT: A Sytem that Learns Flexible Concepts based on Decision Trees for Numerical Attributes. *Proceedings of the Ninth International Conference on Machine Learning ML'92,* Morgan Kaufmann.

Van de Merckt T. and Decaestecker C. (1994), An unifying framework for analysing bias in Similarity Based Learning, *Proceedings of the MlNet Workshop on Declarative Bias,* European Conference on Machine Learning, Catania.