LEARNING TWO-TIERED DESCRIPTIONS OF IMPRECISE CONCEPTS: A METHOD EMPLOYING EXAMPLES OF VARIED TYPICALITY AND AN OPTIMIZED BASE CONCEPT REPRESENTATION: PART II: ALGORITHMS AND EXPERIMENTS

by

F. Bergadano
S. Matwin
R. S. Michalski
J. Zhang

Submitted to the Machine Learning Journal.

LEARNING TWO-TIERED DESCRIPTIONS OF FLEXIBLE CONCEPTS

A Method Employing Examples of Varied Typicality and a Two-staged Construction of the Base Concept Representation

Part II: Algorithms and Experiments

F. Bergadano[1] , S. Matwin[2] , R. S. Michalski, J. Zhang

Artificial Intelligence Center
George Mason University,
Fairfax, VA 22030

MLI 88-9
TR-15-88

November 1988

[1] On leave from Universita di Torino, Italy
[2] On leave from University of Ottawa, Canada

LEARNING TWO-TIERED DESCRIPTIONS OF FLEXIBLE CONCEPTS

Part II: Algorithms and Experiments

ABSTRACT

This paper describes algorithms, their implementation, and experimental results from a system learning two-tiered concept descriptions. A powerful inference method for approximate matching and a general measure for evaluating concept description quality are presented. The learning process is implemented as a two stage procedure. In the first stage, a concept description is obtained from examples using an inductive learning system. The second stage of improving concept descriptions is implemented as a heuristic search in a search space controlled by the quality measure. The heuristics used to trim the search are presented. Experimental results provide evidence that two-tiered concept descriptions are not only simpler than the ones obtained from the existing inductively learning programs, but also have a better predictive power.

ACKNOWLEDGEMENTS

The authors thank G. Thornburg, K. Kaufman, H. Ko, A. Giordana, and F. Oppacher for useful criticism.

This research was done in the Artificial Intelligence Center of George Mason University. Research activities of the Center are supported in part by the Defence Advanced Research Projects Agency under grant, administered by the Office of Naval Research, No. N00014-87-K-0874, in part by the Office of Naval Research under grant No. N00014-88-K-0226, and in part by the Office of Naval Research under grant No. N00014-88-K-0397. The first author was supported in part by the Italian Ministry of Education (ASSI) project, and the second author was supported in part by the NSERC grant No. A2480.

1. INTRODUCTION

A theoretical framework and a method for representing and learning imprecise concepts using a two-tiered concept representation are described in Part I of this paper (Bergadano et al., 88a). In a two-tiered representation, the first tier, called the *Base Concept Representation* (BCR), describes typical properties of the concept in an explicit, comprehensible and efficient format. The second tier, called the *Inferential Concept Interpretation* (ICI), contains inference rules and metaknowledge that define allowable transformations of the concept under different contexts, handle special cases, and cover exceptional instances.

Part I also introduces a system for learning two-tiered concept descriptions. The learning process is organized as a search, guided by a *general description quality* measure. The search operations are truncations of the explicit descriptions. Instances of the concept that are not covered by the truncated descriptions are interpreted by different kinds of inference. The experimental system supports two kinds of such inference. The first kind allows for flexible, probabilistic matching of instances that are not strictly matched by the explicit description. The second kind of inference allows for deductive matching of exceptional instances, using a rule base.

This paper describes the algorithms implementing the above ideas, the two-tiered knowledge representation used, and the experiments with the implemented programs. The knowledge representation schemes used to express the BCR and the ICI are specified. A probabilistic inference method for dealing with approximate matching is introduced. This method is more powerful than the one used in (Michalski et al., 86), since it introduces the idea of the distance between a selector and an example. The deductive inference mechanism using a rule base is also discussed, and the kinds of rules supported by our system are defined. The General concept Description Quality measure (GDQ) is specified in detail, with emphasis on the quality of two-tiered concept descriptions. Finally, the paper presents experimental results of the implemented programs. In these experiments, the system has learned a two-tiered description of a labor-management contract from real-world data.

This study was motivated by the initial work of (Michalski et al., 86), which has demonstrated that by truncating the description obtained from an inductive learning program, one can obtain a significant reduction of memory requirements of a concept representation, without decreasing its performance accuracy.

This paper explores the problem of optimizing an inductively-obtained concept description in greater detail, and it presents a more advanced method of its implementation. Our study confirms the earlier findings of (Michalski et al., 86). Experimental evidence described in sec. 5 of this paper shows that by applying a two-tiered knowledge representation, one can significantly decrease the complexity of a description, and at the same time maintain or even improve its accuracy in recognizing new instances of the concept.

2. KNOWLEDGE REPRESENTATION

In this section, we describe the notation used to represent each of the tiers of a two-tiered concept description. Such concept descriptions are represented by two parts: the base concept representation (BCR) and the inferential concept interpretation (ICI). The ICI consists of a flexible matching function and a set of deductive rules.

2.1. Base Concept Representation

The BCR is represented in Variable-valued Logic System VL (VL_1 or VL_2) notation (Michalski 83). VL is a multiple-valued logic calculus with typed variables. The BCR for a concept is a disjunctive normal form which is called a cover. A cover is a disjunction of complexes. A complex is a conjunction of selectors. A selector is a form:

[L # R] or [L]

where

L is called the referee. In the first form , it is a variable or a function, and in the second form, it is a predicate.

R is called the referent. It is a set of values in the domain of the function or the variable of L.

is one of the following relational symbols : = < > £ ≥ π

Finally quantifiers can be applied to cover, complex and selector.

2.2 Inferential Concept Interpretation: Flexible Matching Function

A flexible matching function F is used as a part of the ICI and it is predefined. The flexible matching function F which is used in our current implemention matches concept descriptions from the set D with events from the set E:

$$F: D \times E \dashrightarrow [0, 1].$$

The value of F of a cover c and an event e is defined as the probabilistic sum of F of its complexes. If c consists of a disjunction of two complexes cpx_1 and cpx_2, we have:

$$F(e, c) = F(e, cpx_1) + F(e, cpx_2) - F(e, cpx_1) * F(e, cpx_2)$$

There is one problem with this definition. Suppose that the cover c consists of many complexes and all values of F of the complexes are very small, say 0.2 Then the value of F(e, c) is close to 1. This is not what we want. We solved the problem by providing a threshold t, such that any value of F(e, cpx) which is smaller than t is treated as 0.

F of a complex cpx and event e is defined as the average of the Fs for a conjunction of its constituent selectors, weighted by the proportion of positive examples covered by the complex:

$$F(e, cpx) = (\sum F(e, sel_i) / n) * \#cpxpos/(\#cpxpos + \#cpxneg)$$

where n is the number of the selectors in cpx and #cpxpos and #cpxneg are the number of positive examples covered by cpx and the number of the negative examples covered by cpx respectively.

F of an event e and a selector sel is defined by the degree of match between the selector and the event weighted by the coverage of positive and negative examples of the selector:

$$F(e, sel) = (1\text{-}DegMatch(e, sel)) * (1+(\#selpos/\#pos - \#selneg/\#neg))/2$$

where #selpos and #selneg are the numbers of positive examples and negative examples covered by the selector respectively. #pos and #neg are the numbers of the positive and negative examples, respectively. Suppose that selector sel is $[x = aj_1 \vee \ldots \vee aj_m]$. DegMatch(e, sel) is then defined as follows:

$$DegMatch(e, sel) = \begin{cases} 1 & \text{if x is nominal and e is covered by sel,} \\ 0 & \text{if x is nominal and e is not covered by sel,} \\ dis(a_k, sel)/\max_{i=1,\ldots,n}(dis(a_i, sel)), & \text{if x is linear} \end{cases}$$

$\mathrm{dis}(a_k, sel) = \min_{k=j1,\ldots,jm}(|i - k|)$

The domain of x is the ordered list $(a_1, a_2,\ldots,a_n)$, and a_k is the value of x of the event e. For example, if the domain of x is [0 .. 10] and the value of x for the event e is 4, then DegMatch(e, [x = 2 v 5]) = $(5\text{-}4)/ \max_{i=0,\ldots,10}(\mathrm{dis}(i, sel)) = 1/5$.

We will not force the system to make a decision when the difference between the values of flexible matching function for two concepts is very small. If the difference is smaller than the preset threshold, the result will be no match.

2.3 Inferential Concept Interpretation: Deductive Rules

The ICI also includes a set of deductive rules, allowing the system to recognize transformed or special cases. In fact, the flexible matching is most useful to cover instances that are close to the typical case. For example, flexible matching could allow us to recognize a sequoia as a tree, although it does not match the typical size requirements, while deductive reasoning would be required to recognize a tree without leaves (in the winter time) or to include in the concept of tree some metaphorical meaning (e.g. a genealogical tree or a search tree).

The deductive rules in the ICI are expressed as Horn clauses. Inference on these rules is implemented using the LOGLISP inference system (Robinson and Sibert, 1982). Numerical quantifiers and internal connectives are also allowed (Michalski 1983).

Finally, the conclusion of a deductive rule can be the special form "Irrelevant(formula)", meaning that "true" can be substituted for "formula" when matching the BCR, if the antecedent of the rule holds. This is especially useful when expressing some transformation of the concept (e.g. season=winter => Irrelevant([has_leaves])). Similar transformation rules have been used in (Bergadano et al., 87). Other rules in the ICI can have as a conclusion either a general predicate (e.g. month=December..march & northern_hemisphere => season=winter) or the classification of the instance, i.e. the predicate *classification=concept_name* or the predicate *classification≠concept_name* (e.g. context=Knuth's_book & acyclic_graph => *classification=tree)*. Rules in the ICI may chain, but simpler deductions are preferred, in order to make the classification easy to understand for the human users. Although the implementation supports recursion, non-recursive rules should be used when possible, and the number of rule activations should be limited.

2.4 Types of Matching

An event can then be covered by a two-tiered description through the following three types of matching:

1. Strict matching: the event matches the BCR exactly, in which case we say that the event is S-covered,
2. *Flexible matching*: the event matches the BCR through a flexible matching function, and we say that the event is F-covered.
3. *Deductive matching*: the event matches the concept through deductive reasoning by using the ICI Rules, and we say that the event is D-covered.

These three sets are made mutually exclusive: if an event is S-covered, then it is not D-covered or F-covered, and if an event is D-covered, then it is not F-covered. Thus, S-covered events are explicitly covered, and F-covered and D-covered events are implicitly covered.

3. QUALITY OF CONCEPT DESCRIPTIONS

Our objective is to obtain concept descriptions of good quality, so the notion of quality has to be introduced, and an operational definition useable in our system has to be given. In sec. 3.1 we discuss the important aspects of description quality, and in sec. 3.2 we focus on these aspects of description quality which are relevant for measuring the quality of two-tiered concept descriptions. Finally, Appendix 1 contains the details of the General Description Quality measure that was implemented in the experimental system. Appendix 2 contains the details of the preference-based evaluation criterion that combines different characteristics into a single measure.

3.1 Criteria for Determining the Quality of Concept Descriptions

The quality of a concept description is influenced by three basic characteristics: the accuracy, the comprehensibility, and the cost. This section discusses these three components, as well as a mechanism for combining them into a a single measure.

The accuracy represents the description's ability to produce correct classifications. A common way to prefer more accurate descriptions is to require that they be complete and consistent with respect to the training events (Michalski, 73; Mitchell, 77; Michalski, 80). Even if a description is incomplete and inconsistent, the number of positive and negative examples it covers provides important information for evaluating its quality. In this case, we can measure the degree of completeness and consistency of a given description. If the description is also sufficiently general and does not depend on the particular characteristics of the training events, these measures can be a meaningful estimate of the accuracy of the description. In order to achieve completeness and consistency in presence of noise, one may generate overly complex and detailed descriptions. Such descriptions, however, may not perform well in future cases and examples. This is the well known phenomenon of overfitting (Watanabe, 69; Sturt, 81).

The comprehensibility of the acquired knowledge is related to subjective and domain dependent criteria. An important requirement of an AI system is that knowledge has to be explicit and easily understandable by human experts. This is important for improving or modifying the knowledge, and for communicating with experts. Since a black box classifier will not be accepted by experts verifying the knowledge of a performance element, therefore knowledge acquired automatically should be easy to understand, should contain the descriptors most frequently used by experts, and should not be syntactically too complex. In practice, only the last feature is easy to obtain.

The cost captures the properties of a description related to its storage and use. Other things being equal, descriptions which are easier to store and easier to use for recognizing new examples are preferred. When considering the cost of a description, two characteristics are of primary importance. The first one is the cost of measuring the values of variables occurring in the description. In some application domains, e.g. in medicine, this may be a very important consideration. The second one is the computational cost of evaluating the description. Again, certain applications in real-time environment, e.g. speech or image recognition, may impose constraints on the evaluation time of a description.

These criteria need to be combined into a single evaluation procedure that can be used to compare different concept descriptions. A possible solution is to have an algebraic formula that, given the numeric evaluations of single criteria, produces a number that represents their combined value. Examples of such approaches are multiplication, weighted sum, maximum/minimum, t-norm/t-conorm (Weber, 1983). Although these approaches are often appropriate, some of them may present disadvantages. Firstly, they usually combine a set of heterogeneous evaluations into a single number, and the meaning of this final number is hard to understand for a human expert. Secondly, they may force the system to evaluate all the criteria, even if it would be sufficient to compare two given descriptions on the basis of the most important one, if one is much better than the other.

In order to overcome some of these problems, we use a lexicografic evaluation functional (LEF) (Michalski, 1972; Michalski, 83) that combines the above mentioned criteria. Appendix 2 discusses in detail the LEF, as well as its modification used in the described system.

The criteria discussed above can also be applied to two-tiered descriptions. The accuracy of the acquired knowledge does not only depend on the explicit information, but also on the implicit reasoning abilities. Inferential Concept Interpretation also affects cost, since it allows the performance system to use a simpler BCR, and reason about special details only in exceptional cases. Finally, the comprehensibility of a two-tiered representation must be carefully evaluated, since one of its implied goals is to state a clear and simple concept description in the BCR and to account for meaningful special cases through a reasoning process.

3.2 The quality of two-tiered concept descriptions

In the previous section, we proposed a general framework for evaluating the quality of concept descriptions:

$$\text{Quality(description)} = <(\text{Accuracy}, \tau_1)\ (\text{Comprehensibility}, \tau_2)\ (\text{Cost}, \tau_3)>$$

which is evaluated using LEF with tolerances τ_1, τ_2, τ_3.

We will now apply the results of this discussion to the task of defining a quality measure appropriate to compare the quality of two-tiered concept descriptions.

The accuracy is the first criterion of concept description quality. Accuracy depends linearly on completeness and consistency of the description, as well as on the typicality of the events covered by the two parts of the description. In evaluating the accuracy of a two-tiered representation, we have to take into account the fact that degree of confidence in the results of inference decreases from deduction to induction (Michalski, 87). These requirements are met by making completeness and consistency dependent on the typicality of the covered examples and on the way these examples are covered. We assume that an expert can provide typicality of examples at the time they are presented to the system responsible for building the initial description. The experts are usually quite good at determining the typicality of events in their area of expertise.

Completeness and consistency of a two-tiered description brings up additional requirements: a good representation should cover the typical examples explicitly, and the non-typical ones implicitly. Moreover, the coverage of typical negative examples in the BCR is particularly detrimental to the quality of the representation. This is important to accuracy because the BCR is mainly obtained or justified by the training events, on an inductive basis. Therefore, one can be confident in the information contained in the BCR only if a sufficient number of examples is available, or if the examples are really typical or representative for the domain. On the contrary, the ICI, being generated by experts or with the available domain knowledge, is appropriate when dealing with rare or exceptional events.

In general, descriptions that cover many typical positive events are most preferred. Completeness is therefore proportional to the typicality of the events covered. Moreover, if negative events are covered, the consistency of the description is inversely proportional to the typicality of the negative events covered.

It is also preferred that the typical events are covered by the BCR , and non-typical, or exceptional events are covered by the ICI. In fact, the BCR is inductively learned from the events provided by user, and it is more reliable when the training events are typical. The ICI, on the contrary, is deductively obtained from the background knowledge, or from a human expert, and relies more on general and domain knowledge. Generally, the ICI is more reliable when dealing with the special or rare cases, since experts often have difficulty in explaining large quantities of typical events. For these reasons, a typical positive explicitly-covered event should contribute to completeness more than implicitly-covered. And vice-versa, nontypical positive implicitly-covered events contribute to completeness more than explicitly-covered. These assumptions are reflected by weights w_S, w_F , w_D , used in the definitions of completeness and consistency (Appendix 1).

Furthermore, since ICI rules are obtained from background knowledge or from a human expert, they are more reliable than the flexible matching function. Consequently, a positive D-covered event should contribute to completeness more than F-covered. We may also observe that flexible matching is not very useful for exceptions whose typicality is very small. A similar argument holds for consistency.

Comprehensibility of a two-tiered representation takes into account the operators occurring in both BCR and ICI, and has to weigh the relative contribution of each part to the comprehensibility of the whole description.

Finally, the notion of cost of a description as introduced in the previous section extends directly to two-tiered descriptions. Details are discussed in Appendix 1.

4. IMPROVING THE QUALITY OF A CONCEPT DESCRIPTION

Learning two-tiered concept descriptions is performed in two stages. In the first stage, a complete and consistent concept description is obtained from an inductive learning system. In our approach, we have relied on AQ15 (Michalski et al., 86a), and INDUCE (Hoff et al., 82) to obtain such descriptions. In this paper, we describe mainly the second stage, which improves the description obtained in the first stage with respect to its GDQ.

4.1 Search Strategies and Operators

The second stage of improving concept descriptions is seen as a state space search (Bergadano et al., 88a). This process is guided by the general description quality discussed in the previous section, and is implemented as a best first search, i.e., the descriptions of better quality are considered first. According to the nature of the quality measure, descriptions can be improved mainly by increasing their accuracy or by decreasing their complexity. For this reason the operators in the search simplify a given description by dropping (truncating) some of its components or by modifying the argument of some predicates. This does not always result in a loss of accuracy, especially when measured on a testing set of new examples, since simpler features might be more stable and depend less on the set of training examples. Table 1 describes the search process:

Search space:	a tree structure, in which the nodes are two-tiered descriptions (BCR + ICI) of a given concept.
Search operators:	selector truncation, complex truncation, referent modification.
Search strategy:	controlled by the quality measure, as described in sec. 4.2

Table 1

Top-level specification of the search algorithm performing optimization of concept descriptions

The goal of this procedure is not necessarily to find an optimal solution, i.e. the description with the highest GDQ value, because this would require a combinatorial search. On the contrary, the system tries to improve the given concept description by expanding a limited number of nodes in the search tree and is guided by a heuristic measure.

One more characteristic of the system should be mentioned: only one operator is applied to the selected (best-quality) node at any one time; therefore, the new node can be selected if its quality is better than the quality of the father node. This is different from standard search procedures, where all the applicable operators are used for the selected node (node expansion). This choice was introduced because the creation of a new node involves the computation of its quality, which, in some cases, can be time-consuming. On the contrary, we try to avoid generating bad quality nodes by selecting the best applicable operator on a heuristic basis, and applying only that operator. The other operators will be used only if the results obtained on this search branch turn out to be unsatisfactory. The heuristics used for selecting the operators will be discussed shortly.

The operators in the search correspond to generalizations or specializations of a given description. In particular, selector truncation is a generalization operator, making the new description cover more positive and negative examples, while complex truncation is a specialization operator, making the set of examples covered by the modified description smaller. Referent modification can be either a specialization or a generalization operator, depending on the type of modification that is being used and on the type of selector involved. The search starts from an initial description, and in this implementation the initial description is supplied by a previous inductive learning phase (using AQ15 or INDUCE). The system can also be applied to the disjunction of all the training events, and the search process can use this as an initial description, although this choice could make the whole system slower. An important issue for further research is related to the integration of the first inductive learning phase and the search in the space of two-tiered concept descriptions.

4.2. Example.

An abstract example of the search process is given in Fig. 3. The nodes contain BCR, ICI, and a graphical representation of the covered examples. The tree is kept in memory throughout the search. The BCR is expressed in disjunctive normal form (it is a "cover"). In this example accuracy is computed according to the formula discussed in sec. 3 and given in Appendix 1,

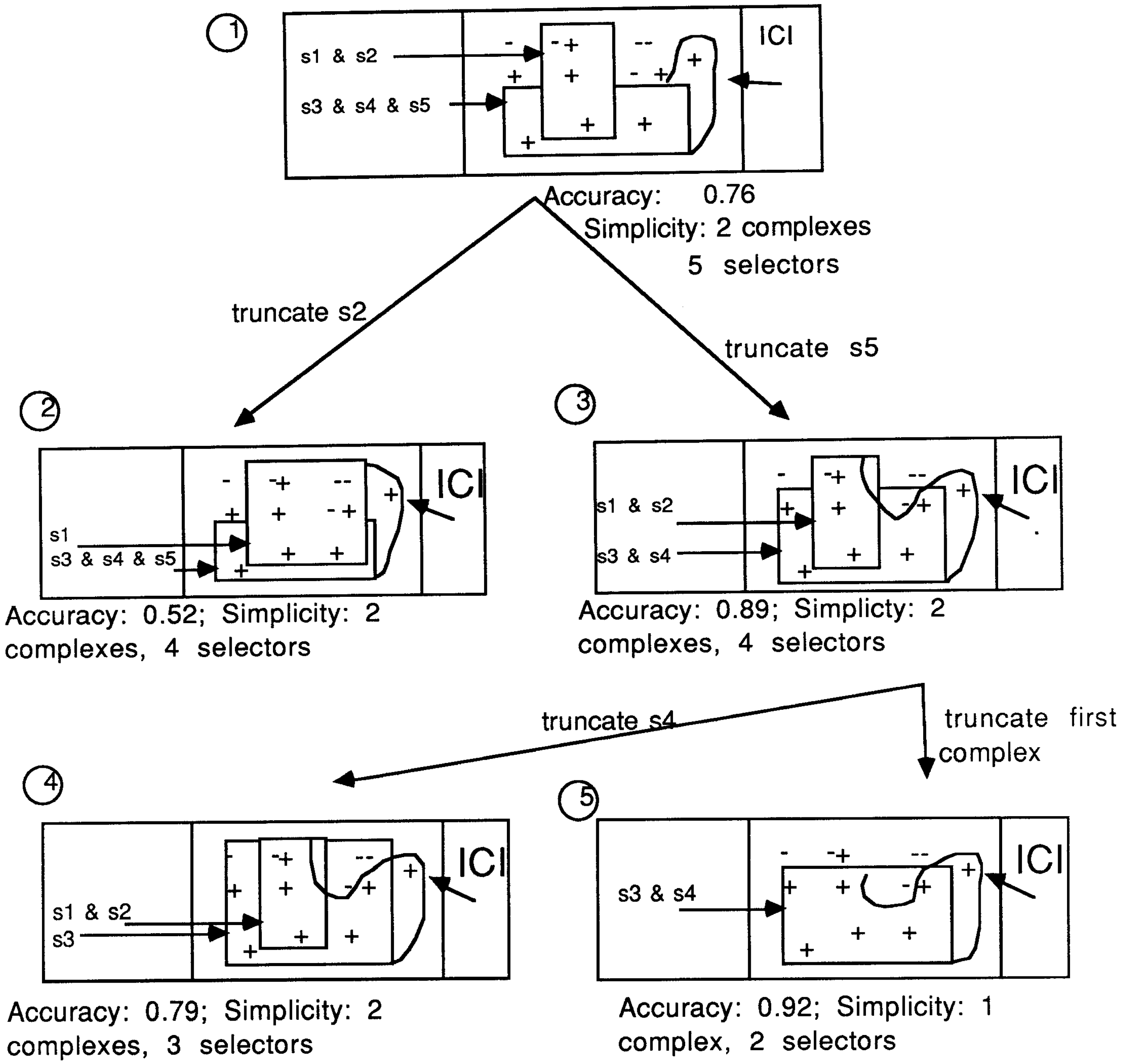
1
s1 & s2
s3 & s4 & s5
ICI
Accuracy: 0.76
Simplicity: 2 complexes
5 selectors
truncate s2
truncate s5
2
s1
s3 & s4 & s5
ICI
Accuracy: 0.52; Simplicity: 2 complexes, 4 selectors
3
s1 & s2
s3 & s4
ICI
Accuracy: 0.89; Simplicity: 2 complexes, 4 selectors
truncate s4
truncate first complex
4
s1 & s2
s3
ICI
Accuracy: 0.79; Simplicity: 2 complexes, 3 selectors
5
s3 & s4
ICI
Accuracy: 0.92; Simplicity: 1 complex, 2 selectors

Fig. 1. Example of the search space organized as a tree. For clarity, the cost is omitted, and the simplicity of the ICI is not taken into account. The simplicity of the BCR depends on the number of complexes and selectors.

assuming the same typicality of all the examples. The initial description is represented in node 1, and contains two disjuncts (complexes). The complexes cover the two corresponding rectangular areas in the graphical representation, containing five positive examples out of eight, and one negative example out of five. The ICI extends this coverage by recognizing one more positive example. By eliminating conjunct (selector) s5 in the second complex we obtain node 3 in the search tree. The accuracy of the description is now improved since all the positive examples are covered. Finally, by truncating the first complex we obtain node 5. It does not cover negative examples any more, and is definitely simpler. This node is then accepted as the improved description resulting from the search. The other nodes lead to inferior concept representations, with respect to GDQ, and are discarded. The quality has been computed with w1=w2=0.5 (see Appendix 1).

4.3. The Search Algorithm

The general search procedure is more precisely presented by the following Search Algorithm:

1. Identify in the search tree the best description D (one with the highest GDQ). Initially, D is the complete and consistent description obtained in stage 1.
2. Apply to D this operator among C_i, S_{ij}, R that potentially improves GDQ of D the best, based on the *Potential Accuracy Improvement* (PAI) heuristics described in sequel:

   C_i: Remove the i-th complex from D.

   S_{ij}: Remove the j-th selector from the i-th complex in D .

   R: Modify the referent in a selector of a complex in D.
3. Compute the GDQ of the node obtained in step 2. If this GDQ is smaller than the GDQ of D, then proceed to step 1.
4. Ask for an explanation of

   (a) the positive examples that are not covered any more

   (b) the negative examples that are now covered

   If such an explanation is found, augment ICI accordingly.
5. Update the GDQ value of the new node, by taking into account the added ICI rules.
6. If the *stopping criterion* is satisfied, then STOP, otherwise proceed to step 1.

We shall now discuss the motivation and details of the algorithm, and explain the search strategy.

In step 1, the nodes are chosen on a best first basis, that is the node in the search space with the highest GDQ value is expanded first. This is not always an optimal choice, since apparently "bad" nodes can lead to better descriptions after a number of truncations. Whether the search will behave in this manner will depend on the adequacy of the GDQ as the measure of concept quality.

In step 2, a search operation on the description is chosen heuristically and applied to the description. Only one operator is applied at one time. The heuristic here is to choose the operation which has the best chance of improving the GDQ of the description. The specific choice of the operations C_i and S_{ij} is determined based on the Potential Accuracy Improvement heuristic (PAI). The idea behind the PAI heuristic is to truncate first a complex which covers uniquely a small number of examples. Then, a specific choice of the operator S_{ij} is made so that the truncation of a selector improves the completeness of the description, while the consistency measure has an acceptable value. Finally, when no other operator is recommended, referent modification can be selected. Referent modification can improve both consistency and completeness measures.

In the worst case, this algorithm will perform a full search of the search space. We can observe, however, that the search is controlled by the heuristic PAI, which is computational much less expensive than the GDQ.

The complex and selector truncation heuristics are implemented together in the following way. Let us first define the PAI. The PAI of truncating a complex is computed as follows:

PAI = #CNI/#NEG - #CPI/#POS

where #CNI (#CPI) is the number of negative (positive) examples no longer covered by the concept description after truncating the complex, respectively. #NEG and #POS are the numbers of negative and positive examples, respectively. The PAI of truncating a selector is more complex and is defined as follows:

PAI = (#SPI/#POS) * P - (#SNI/#NEG) * N

where #SNI (#SPI) is the number of additional negative (positive) examples covered by the concept after having truncated the selector, respectively. N is the proportion of the negative examples which are not covered by the description and, P is the proportion the positive examples which are not covered by the description.

The operation is chosen based on the value of the PAIs. The operation with the largest PAI is chosen. Finally, the PAIs of selector and complex truncation can be weighted differently. More weight can be assigned to PAI of complex truncation, since the complex truncation simplifies the description more than selector truncation.

In step 3, the system computes the GDQ of the new node. It should be noted that, in the GDQ measure, the typical examples covered directly by the BCR can weigh more than those covered through flexible matching. The examples covered by ICI rules should weigh more than the ones covered through flexible matching but less than the ones covered by the BCR.

In step 4, the "explainer" module (Bergadano et al. 88) is used in order to improve the description even further: the BCR description is extended or shrunk by adding ICI rules. Firstly, complex truncation might have caused some positive examples, that were previously covered, to be lost. In this case some new rules could be introduced in the ICI, that would allow the system to reason about such "special" positive examples, and understand why they should still be classified as instances of the concept under consideration. On the other hand, selector truncation might have caused some negative examples to be covered, and new rules in the ICI may be added in order to "shrink" the BCR and avoid these erroneous classifications. Another issue, concerning step 3, is whether an explanation should be required at all, since, in some cases, the chosen truncation operator is not an appropriate one, and will lead to a very poor description. In this case it is not even worth to ask for an explanation, and search can continue in other directions. The current strategy is as follows. Suppose the relation < denotes the GDQ ordering among two-tiered descriptions, n is the node we are expanding and m is the node we obtain after the selected truncation. If m<<n, then no explanation is even tried, otherwise the explainer is asked for an explanation and is told how m compares to n with respect to <, in order to know how important the request for the explanation is for the search procedure.

In step 5, the GDQ of the obtained two-tiered description is updated after the new ICI rules have been added. Since ICI rules are taken into consideration in the GDQ, new ICI rules will change the GDQ value for a concept.

In step 6, the system decides whether to stop or continue the search. The *stopping criterion* is satisfied when the search space obtained is "very large", or when no qualitative improvement has been obtained for a "long time". The particular values of "very large" and "long time" are parameters of the algorithm. They depend on the size of the initial description. When the system stops, the best node in the search space is produced and becomes the modified two-tiered concept description.

5. EXPERIMENTS: LEARNING A TWO-TIERED DESCRIPTION OF A LABOR-MANAGEMENT CONTRACT.

This section describes how the system learns a two-tiered description of a labor-management contract from real data. Unlike the example in Part I, which has been simplified to illustrate the method, rather than its power, the example in this section relies on real data.

The data used in this section comes from *Collective Bargaining* - a review of current collective bargaining issues published by the Government of Canada through its Department of Labor. The data given in *Collective Bargaining* describes labor-management contracts which have been currently negotiated between organizations and those union locals that count at least 500 members. The raw data is divided geographically, as well as by economic sectors. The format of the raw data is pretty standard. Each contract is described by a number of attributes. Since the attributes vary between economic sectors, we have decided to focus on a single sector: personal and business services. This sector includes unions representing hospital staff, teachers, university professors, social workers, and certain classes of administrative personnel of different organizations. With this kind of data, it was natural to represent concepts using the VL_1 formalism.

Our data describes contracts finalized in the second half of 1987 and first half of 1988. Each contract is described by sixteen attributes, belonging to two main groups: issues related to salaries (e.g. pay increases in each year of contract, cost of living allowance, stand-by pay, etc.), and issues related to fringe benefits (e.g. different kinds of pension contributions, holidays, vacation, dental insurance, etc.).

We have run three experiments. In each experiment we were dealing with two descriptions: a description of a contract, and a description of a contract proposal deemed unacceptable by one of the parties. In each experiment we have looked at the number of events correctly and incorrectly covered by both descriptions, and at the number of events that were not covered by either concept. This was done both on the training set and on a testing set of examples not previously seen by the system. The results of the three experiments are given in Tables 2, 3, 4, and 5 are discussed below. In each experiment, the same training and testing sets were used. The training set consisted of 18 positive and 9 negative examples of contracts; the testing set consisted of 19 positive and 11 negative examples.

Factual Knowledge (27 complexes and 432 selectors)

		Correct	Incorrect	No_Match
Strict Match				
	Training Set	100%	0%	0%
	Testing Set	0%	0%	100%
Flexible Match				
	Training Set	100%	0%	0%
	Testing Set	37%	0%	63%

Table 2 Results of Experiment 1

A concept description as a disjunction of training examples.

Initial Description (11 complexes and 28 selectors)

		Correct	Incorrect	No_Match
Strict Match				
	Training Set	100%	0%	0%
	Testing Set	80%	17%	3%
Flexible Match				
	Training Set	100%	0%	0%
	Testing Set	80%	17%	3%

Table 3 Results of Experiment 2.

Concept descriptions derived by AQ15. Completeness and consistency were required.

Optimized Description (9 complexes and 12 selectors)

		Correct	Incorrect	No_Match
Strict Match				
	Training Set	63%	0%	37%
	Testing Set	43%	3%	54%
Flexible Match				
	Training Set	85%	0%	15%
	Testing Set	83%	13%	4%
Inferential Match				
	Training Set	96%	0%	4%
	Testing Set	90%	10%	0%

Table 4 Results of Experiment 3.

Concept descriptions were obtained using the search procedure described in this paper. In the case of inferential matching, expert-provided rules were used in combination with flexible matching.

In the first experiment, we have used the events from the training set as a purely factual concept description (Factual Knowledge): the concept was just the disjunction of the training examples. This description is obviously complete and consistent on the training set but has no predictive power, i.e. it always produces a No_Match answer. This happens because in our experiment, as it often happens when dealing with real data, no testing examples were exactly equal to some training event.

The factual knowledge was also used with a flexible matching technique, based on a measure of the distance between an event and a concept description (see Section 2). Consistency and completeness are achieved in the training set, and this is again obvious since flexible matching is only used when either no match or multiple match has occurred, and this never happens if the factual knowledge is used to classify the training examples. On the contrary, the flexible matching causes some performance improvement for the test set, where 37% of the events are now correctly classified, and we still do not obtain any erroneous classification. This method is similar to statistical techniques in Pattern Recognition, e.g. the nearest neighbor method (Watanabe 85).

In the second experiment, we have used the descriptions learned by AQ15 (Initial Description). Since AQ15 only generates consistent and complete descriptions, classification is 100% correct for the training set, and flexible matching does not affect this performance. For the testing set, the number of correct classifications is still high (80%), and flexible matching does not improve the result in this case. This is partly related to the fact that the descriptions generated by AQ are detailed and specify many alternatives, leaving little space for the no_match case (3%). Moreover, the multiple match case was impossible because AQ15 was run with the "disjoint cover" parameter, causing the generated concept descriptions to have disjoint extensions. In general the flexible matching can improve the performance of the initial description on the test set.

[duration ≠ 1] & [wage_incr_yr2 ≠ 3.0] & [holidays ≠ 10] v
[wage_incr_yr1 ≠ 2.0% v 2.5% v 2.8% v 3.0% v 4.0% v 4.5%] v
[wage_incr_yr1 ≠ 2.0% v 2.5% v 2.8% v 4.0%] & [wage_incr_yr2 ≠ 2.0% v 4.0%] v
[wage_incr_yr1 ≠ 2.0% v 2.5%v 3.0%v 4.0%v 4.5%] & [holidays ≠ 9] v
[wage_incr_yr1 ≠ 2.0% & [vacation = above_average] ::> *acceptable contract*

[wage_incr_yr1 = 2.0% v 2.5% v 4.0%] & [holidays = 10] &
    [vacation = below_average v average] v
[wage_incr_yr1 = 2.0% v 2.5% v 3.0%v 4.0% v 4.5%] & [wage_incr_yr2 = 2.0% v 4.0%] &
    [holidays = 10] & [vacation = below_average v average] v
[duration = 1] & [wage_incr_yr1 = 2.0% v 2.5% v 2.8%v 4.0%] &
    [holidays = 9] & [vacation = below_average v average] v
[wage_incr_yr1 = 2.0% v 2.5% v 4.0%] & [wage_incr_yr2 = 3.0%] &
    [vacation = below_average v average] v
[duration = 1] & [wage_incr_yr1 = 2.0%v 2.5% v 4.0%] &
    [vacation = below_average v average] v
[wage_incr_yr1 = 2.0%] & [wage_incr_yr2 = 3.0%] ::> *unacceptable contract*

Fig. 2. Descriptions Generated by AQ15

If compared with the factual knowledge, the descriptions generated by AQ are definitely superior. Not only their performance is higher, but they are also much simpler (28 selectors compared to 432). Simplicity is closely related to comprehensibility in the given domain, and allows the performance system to recognize new events more efficiently. The descriptions generated by AQ15 are given in Fig. 2.

The third experiment allows us to evaluate empirically the method presented in this paper. We have used the description generated by the search process (Optimized Description), and evaluated its performance both with the flexible matching alone and with the combination of flexible and deductive matching (Inferential Match). For the sake of completeness we also present the performance of the generated descriptions with strict matching, although this would never be used. Strict matching alone yields restricted coverage and poor performance. In fact, the power of the modified description is due to a combination of all three types of matching (strict, flexible and deductive), and all three contribute to the quality measure of a description as computed during the learning process. This represents a new feature of this system, since inferential matching is usually introduced only after the learning phase is completed (Bergadano and Giordana ,to appear; Michalski et al., 86). Fig 2 gives the descriptions output by the system.

[wage_incr_yr2 ≠ 3.0%] v [wage_incr_yr1 ≠ 2.0% v 2.5% v 2.8% v 3.0%v 4.0% v 4.5%] v
[holidays ≠ 9] v [vacation = above_average] ::> *acceptable contract*

[wage_incr_yr1 = 2.0% v 2.5% v 4.0%] & [holidays = 10] v
[wage_incr_yr2 = 2.0% v 4.0%] & [vacation = below_average v average] v
[holidays = 9] v
[duration = 1] & [wage_incr_yr1 = 2.0% v 2.5% v 4.0%] v
[wage_incr_yr2 = 3.0%] ::> *unacceptable contract*

Fig. 3. Optimized Descriptions.

The modified descriptions are simpler than the ones generated by AQ15, and should represent the most important characteristics of the labor management concepts. The performance of this descriptions is slightly better if the flexible matching is used (83% correct classifications for the testing set, compared to 80% for the initial description with flexible matching). It should be noted that, on the contrary, the descriptions generated by AQ15 performed better on the training set (100% vs 85%), suggesting that the data might have been overfitted.This cannot usually be avoided if complete and consistent descriptions have to be obtained, as is required in the implementation of AQ15 used in the experiment.

The training events that were not correctly classified by the description, as it was modified step by step during the search, were analyzed by a domain expert, and the following rules were acquired, as part of the ICI:

[wage_incr_yr1 < 3.0%] & [wage_incr_yr2 < 3.0%] ==> low_wages

low_wages & [wage_incr_yr2 < wage_incr_yr1] ::> *unacceptable contract*

low_wages & [hours ≥ 40]
    & [pension_type = none v retirement_allowance] ::> *unacceptable contract*

[wage_incr_yr1 > 5.5%] & [vacation = above_average] ::> *acceptable contract*

Fig. 4. Deductive rules of the ICI of the optimized description

These rules allow the system to classify almost all the training events (one of them could not be explained by the expert). The combination of modified description and inferential matching (rules plus flexible matching) produces the best results. The description is still simple, although it now includes the ICI rules, and the number of correct classifications is 90%. Moreover, some of the examples that were previously recognized by flexible matching or strict matching are now also correctly recognized by the ICI rules, and this might suggest that the description is more robust, and could perform even better on a larger test set.

Results of this experiment are summarized in Table 5. They indicate that a two-tiered approach to learning allows a system to learn concepts from a small number of examples, and produce descriptions that are simpler than the ones provided by a typical inductive learning system. In Table 5, the Factual Description denotes again the disjunction of training examples, plus the flexible matching function. The Initial Description is provided by AQ15 and also uses the flexible matching; the missing 3% of the training set represent the "no match" situation. The highest performance is realized by the Optimized Description, in which the concept meaning is divided into the BCR, generated by the search, a flexible matching procedure, defined a-priori, and a set of deductive rules given by the expert on the basis of misclassified examples selected automatically during the search process. There are four deductive ICI rules with nine conditions (see Fig. 4); their complexity corresponds to four complexes and nine selectors.

	#Complexes	**#Selectors**	**Performance** (correct/incorrect)
Factual Description	27	432	37%/0%
Initial Description	11	28	80%/17%
Optimized Description			90%/10%
BCR	9	12	
ICI	4	9	

Table 5. Summary of the experimental results.

6. CONCLUSION

In this paper, a two-tiered knowledge representation formalism has been developed, and a system that is able to learn two-tiered concept descriptions has been described. Learning is viewed as a state space search guided by a measure of quality that is applied to concept descriptions. The operators in the search modify a given description by removing some of its components or by simplifying the referent of the selectors. The goal of the search process is to obtain simpler but still accurate descriptions. In this way, the comprehensibility and the predictive power of the acquired knowledge are improved.

The system was applied to the problem of learning the concept of an acceptable labor management contract. The experimental results that we obtained confirm the hypothesis that two-tiered descriptions can be more accurate and easier to understand. The ICI used in the experiments included a flexible matching function and a set of logical rules. The performance of the descriptions produced by the search process on the test set is influenced by the use of the inferential matching. This is due to the fact that ICI is used during learning, in order to choose and modify the best descriptions. This property represents an important difference between the presented system and previous approaches, that tend to apply flexible matching only after the learning process is completed.

Some of the motivations behind the system come from previous work (Michalski et al, 1986), that produced some preliminary results in which flexible matching function was applied during the testing phase. The same research investigated the effect of truncating concept descriptions. In the system presented here, though, the flexible matching function is augmented by a set of rules defining how to extend or modify a concept description at the "knowledge level", by describing symbolically its possible transformations. Selector truncation and referent modification are also introduced (besides complex truncation). Truncations are applied automatically during a search process, whereas in (Michalski et al., 1986) they were applied manually, with a limited number of tries. The search for better two-tiered descriptions is the main process during the learning phase implemented in this system.

An important issue for future research and improvements of the implemented system is the integration of the search procedure with the inductive learning system used to generate the initial description (AQ or INDUCE). The first step in this direction is being experimented with: it allows the two systems to share the same heuristics and the same measure of quality. Further progress is related to the possibility of obtaining partially incomplete and inconsistent description also during the generation of the initial description. More experimentation is also needed in order to evaluate more precisely the performance of the implemented system.

REFERENCES

Bergadano, F., Matwin, S., Michalski, R.S., Zhang, J., "Learning Two-tiered Descriptions of Imprecise Concepts: A Method Employing Examples of Varied Typicality and A Two-staged Construction of the Base Concept Representation", Part 1: Principles and Methods, Research Report, Machine Learning and Inference Group, Artificial Intelligence Center, George Mason University, 1988 (submitted to Machine Learning Journal).

Bergadano, F. , Giordana, A. , Saitta, L., "Automated Concept Acquisition in Noisy Environments", IEEE Transactions on PAMI, July 1988.

Bergadano, F., Giordana, A., "Pattern Classification: An Approximate Reasoning Framework", International Journal of Intelligent Systems, (To appear).

Michalski, R. S., "A Variable-Valued Logic System as Applied to Picture Description and Recognition", in "Graphic Languages", Nake, F. and Rosenfield, A. (Eds.), North Holland, 1972.

Michalski, R. S., "AQVAL/1--Computer Implementation of a Variable Valued Logic System VL1 and Examples of its Application to Pattern Recognition", Proc. of the 1rst International Joint Conf. on Pattern Recognition, Washington, D.C., pp. 3-17, 1973.

Michalski, R. S., "Pattern Recognition as Rule-guided Inductive Inference", IEEE Transactions on PAMI, vol. 2, no. 2,3,4, pp. 349-361, 1980.

Michalski, R.S., "A Theory and Methodology of Inductive Learning", Chapter in the book "Machine Learning, an Artificial Intelligence Approach", Michalski, R. S., Carbonell, J. G., Mitchell, T. M. (Eds.), Tioga Pub. Co., Palo Alto, Ca, 1983.

Michalski, R. S., Carbonell, J. G., Mitchell, T. M., "Machine Learning: An Artificial Intelligence Approach", Tioga Publishing Co., Palo Alto, Ca, 1983.

chalski, R. S., Mozetic, Hong, J.,I., Lavrac, "The Multi-purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains", Proc. 5th AAAI, pp. 1041-1045 1986.

Michalski, R. S., "Two-Tiered Concept Meaning, Inferential Matching and Conceptual Cohesiveness" , Chapter in the Book "Similarity and Analogy", Stella Vosniadou and A. Orton, (Eds), 1987.

Mitchell, T. M., "Version Spaces: An Approach to Concept Learning", Ph. D. dissertation, Stanford University, December 1977.

Robinson J. A. and Sibert E. E., "LOGLISP: an alternative to Prolog" in Machine Intelligence 10, Hayes, J. E. and Michie, D. (Eds.), pp. 399-419, 1982.

Sturt, E., "Computerized Construction in Fortran of a Discriminant Function for Categorical Data", Applied Statistics, vol. 30, pp. 213-222, 1981.

Watanabe, S., "Knowing and Guessing - a Formal and Quantitative Study",Wiley Pub. Co., 1969.

Watanabe, S., "Pattern Recognition Human and Mechanical", John Wiley, 1985.

Weber, S., "A General Concept of Fuzzy Connectives, Negations and Implications based on t-norms and t-conorms", Fuzzy Sets and Systems, vol. 11, pp. 115-134, 1983.

APPENDIX 1.

A detailed General Description Quality Measure.

The purpose of the Appendix is to define in detail the GDQ measure that was implemented in our experimental system. First, we have to define the typicality-dependent completeness (TCOM) and typicality-dependent consistency (TCON) of a description:

$$\text{TCOM} = \frac{\sum_{e^+ \text{ is S-covered}} w_S*\text{Typicality}(e^+) + \sum_{e^+ \text{ is F-covered}} w_F*\text{Typicality}(e^+) + \sum_{e^+ \text{ is D-covered}} w_D*\text{Typicality}(e^+)}{\sum_{e \in \text{PosCov}} \text{Typicality}(e)}$$

$$\text{TCON} = 1 - \frac{\sum_{e^- \text{ is S-covered}} w_S*\text{Typicality}(e^-) + \sum_{e^- \text{ is F-covered}} w_F*\text{Typicality}(e^-) + \sum_{e^- \text{ is D-covered}} w_D*\text{Typicality}(e^-)}{\sum_{e \in \text{NegCov}} \text{Typicality}(e)}$$

where:
PosCov -- set of positive events covered by two-tiered concept description,
NegCov -- set of negative events covered by two-tiered concept description,
typicality(e) -- typicality of the event e specified by the expert when the event is given.
w_s -- if typicality(e) $\geq t_2$ then 1 else w,
w_f -- if $t_2 \geq$ typicality(e) $\geq t_1$ then 1 else w,
w_d -- if $t_2 \geq$typicality(e) then 1 else w,
where, t1 and t2 are thresholds, and $1 \geq t_2 \geq t_1 \geq 0$, $1 \geq w > 0$.

Now *accuracy* can be defined in terms of TCOM and TCON:

$$\text{Accuracy(description)} = w_1*\text{TCOM(description)} + w_2*\text{TCON(description)}$$

where $w_1 + w_2 = 1$.

A measure of comprehensibility of a concept description is difficult to define. We will approximate this measure by a syntactic complexity, defined as:

$$v_1 \sum_{op \in \text{BCR(dsp)}} C(op) + v_2 \sum_{op \in \text{ICI(dsp)}} C(op)$$

where:
- BCR(dsp) -- a set of all operator occurrences in the BCR

- ICI(dsp) -- a set of all operator occurrences in the ICI
- C(op) -- the complexity of an operator. The complexity of operator on the list <interval, internal disjunction, =, <>, not, &, v, implication, predicate> increases with its position on the list. When an operator is a predicate, C increases with the number of the arguments in the predicate.

- v_1 and v_2 are weights, $v_1 + v_2 = 1$. The BCR should describe the general and easy-to-define meaning of the concept, while the ICI is mainly used to handle nontypical or exceptional events, therefore the BCR should be easier to comprehend than the ICI. v_1 should therefore be larger than v_2.

The cost consists of two parts:

Measure-Cost -- the cost of measuring the values of variables used in the concept description, it is defined as the function MC
Evaluation-Cost-- the computational cost of evaluating the concept description, it is defined as the function EC.

$$MC(description) = \sum_{e \in Pos+Neg} \sum_{v \in vars(e)} mc(v)/(|Pos| + |Neg|)$$

$$EC(description) = \sum_{e \in Pos+Neg} ec(e)/(|Pos| + |Neg|)$$

where
vars(e) -- set of all occurrence variables used to evaluate the concept description to classify the event e.
mc(v) -- the cost of measuring the values of the variable v,
ec(e) -- computational cost of evaluating concept description to classify the event e. This could depend on computation time or on the number of operators involved in the evaluation.

We now define the cost of a description:

$$Cost(description) = u_1*MC(description) + u_2*EC(description)$$

where u_1 and u_2 are weights.

With the exception of the weights which can be determined experimentally, we have already defined all three components of the quality measure of concept descriptions. In the next section, we will show how the quality measure evaluates a simple concept description. This quality measure has been experimented with two non-trivial examples, acceptable labor-management contract and the concept of "chair", the results are satisfactory. Currently, we are implementing the quality measure in a two-tiered concept learning system and using it to guide the search for a better two-tiered concept description.

APPENDIX 2

The Preference-based Evaluation Criterion

The problem that we are addressing here is how to combine different criteria used to compare concept descriptions. A possible solution is to have an algebraic formula that, given the numeric evaluations of single criteria, produces a number that represents their combined value. Examples of such approaches are multiplication, weighted sum, maximum/minimum, t-norm/t-conorm (Weber, 1983). Although these approaches are often appropriate, some of them may present disadvantages. Firstly, they usually combine a set of heterogeneous evaluations into a single number, and the meaning of this final number is hard to understand for a human expert. Secondly, they may force the system to evaluate all the criteria, even if it would be sufficient to compare two given descriptions on the basis of the most important one, if one is much better than the other.

In order to overcome some of these problems, we use a *lexicographic evaluation functional* (LEF) (Michalski 1972, Michalski 83) that combines the above mentioned criteria. The general description quality measure is thus defined as:

$$\text{GDQ(description)} = <(\text{Accuracy},\tau_1), (\text{Comprehensibility},\tau_2), (\text{Cost},\tau_3)>$$

where $\tau 1$, $\tau 2$, and $\tau 3$ are tolerance thresholds (which will be discussed later).

In this evaluation scheme, the criteria are ordered according to their importance, and a tolerance threshold is associated with each criterion. If the difference of the evaluation of two expressions under a given criterion is less than the corresponding tolerance, the two descriptions are considered equivalent with respect to that criterion. The most important measure in the LEF is evaluated first, and the subsequent measure is evaluated only if the previous one is a tie.

The LEF evaluation scheme is not affected by the main problems related to algebraic formulas, which we have discussed above, but it may be useful to extend it in some cases. In fact, it can be difficult to determine the tolerance. If the tolerance is too small, we have very little chance of using the other criteria. If the tolerance is too large, some important criterion might be underestimated. Suppose, for example that two descriptions d1 and d2 are such that Accuracy(d1) >> Accuracy(d2) but the difference is within the tolerance, and suppose that Comprehensibility(d2) is slightly better than Comprehensibility(d1) and not in the tolerance. In this case d2 would be preferred although d1 is probably better, since its accuracy is much greater and the comprehensibility of the two descriptions is approximately the same. In order to avoid this problem, the LEF measure can be extended in the following way: first LEF is applied with larger tolerances, in such a way that all the relevant criteria are taken into account; then, if the comparison still results in a tie, a Weighed Evaluation Functional (WEF) is used to combine the measures (i.e. the description having the maximum weighted sum of the measures is preferred).

The above criteria can also be applied to two-tiered descriptions. The accuracy of the acquired knowledge does not only depend on the explicit information, but also on the implicit reasoning abilities. Inferential Concept Interpretation also affects cost, since it allows the performance system to use a simpler BCR, and reason about special details only in exceptional cases. Finally, the comprehensibility of a two-tiered representation must be carefully evaluated, since one of its implied goals is to state a clear and simple concept description in the BCR and to account for meaningful special cases through a reasoning process.