INCREMENTAL LEARNING OF CONCEPT DESCRIPTION: A METHOD AND EXPERIMENTAL RESULTS

by

R. E. Reinke
R. S. Michalski

Chapter in Machine Intelligence II, D. Michie and J. Michie (eds.), 1988

MACHINE INTELLIGENCE 11

Logic and the acquisition of knowledge

---

edited by

J. E. HAYES

Research Associate,
Turing Institute

D. MICHIE

Chief Scientist,
Turing Institute

and

J. RICHARDS

Head of Industrial Studies,
Turing Institute

CLARENDON PRESS · OXFORD
1988

CONTENTS

11

Incremental Learning of Concept Descriptions: A Method and Experimental Results

R. E. Reinke
GTE Laboratories,
Waltham, Massachusetts, USA
R. S. Michalski*
Massachusetts Institute of Technology,
Cambridge, Massachusetts, USA

Abstract

A system for learning concept descriptions incrementally is described and illustrated by a series of experiments in the domains of insect classification, chess endgames and plant disease diagnosis. The system employs a *full-memory* learning method that incrementally improves hypotheses, but does not forget facts. The method is used to form both *characteristic* descriptions, which describe a concept in detail, and *discriminant* descriptions, which specify only properties needed to distinguish a given concept from a given set of other concepts. Experimental results show the advantages of inducing and maintaining only characteristic descriptions during learning and creating discriminant descriptions from them when a classification decision is necessary.

1. INTRODUCTION

Research in the area of concept learning from examples has been concerned mainly with methods for single step, or non-incremental, learning. Such methods can effectively and efficiently induce good descriptions from a given set of examples and, optionally, induce counter-examples (for example Michalski, 1975, 1980a; Quinlan, 1979; Langley *et al.*, 1983). These methods cannot modify concept descriptions which are contradicted by new examples, but must re-learn the descriptions from scratch. In contrast, incremental learning methods modify concept descriptions to accommodate new learning events (Winston, 1975; Michalski and Larson, 1978).

When we observe human learning we clearly see that it is incremental. People learn concept descriptions from facts and incrementally refine

* Current address: c/o George Mason University, 4400 University Drive, Fairfax, VA 22030.

those descriptions when new facts or observations become available. Newly gained information is used to refine knowledge structures and models, and rarely causes a reformulation of all the knowledge a person has about the subject at hand.

There are two major reasons why humans must learn incrementally:

1. Sequential flow of information. A human typically receives information in steps and must learn to deal with a situation long before all the information about it is available. When new information does become available, there is rarely time to reformulate everything known about all the concepts involved.

2. Limited memory and processing power. People cannot store and have easy access to all the information they have been exposed to. They seem to store only the most prominent facts and generalizations, then modify the generalizations when new facts are available.

This paper describes a method for automated learning of concept descriptions from examples which is novel in its use of facts and of concept descriptions. We assume that in practical machine learning systems, only the first of the above constraints is important and that the second may be ignored. The fact that information arrives sequentially cannot be changed, as it reflects the nature of the world. On the other hand, storing and retrieving large amounts of information is not difficult for modern computers. We therefore investigate a *full-memory* incremental learning system which modifies concept descriptions to accommodate new information, but does not forget facts.

A concept description can be assigned a *type* based on two factors: purpose and form. A description's purpose is either to *characterize* or to *discriminate* (Michalski, 1983). A characteristic description of a concept is very specific and tries to capture all the known properties of the concept. Such a description is useful for building a detailed model of the concept and for teaching someone about the concept. On the other hand, discriminant descriptions are used to distinguish one concept from a given set of other concepts and contain only those properties of the concept which are necessary to make such distinctions. Characteristic descriptions attempt to distinguish a given concept not just from a known set of other concepts but from *any* other concepts. Thus, discriminant descriptions are dependent on the class of concepts under consideration while characteristic descriptions are not. In short, characteristic descriptions are used to describe and discriminant descriptions are used to discriminate. Section 2 gives more details and presents a classification of different types of descriptions.

The form of a concept description is directly dependent on the description language used. In the variable-valued logic used in this paper (see next section), a description may be either *conjunctive* or *disjunctive*. We therefore distinguish between four types of description: characteristic conjunctive (CC), characteristic disjunctive (CD), discriminant conjunctive (DC), and discriminant disjunctive (DD).

People are able to learn and use many different types of concept descriptions. Further, the type of description a human uses may depend on the situation. The learning method described here can also be used to form several description types; these may be used in different ways when learning incrementally. We describe experiments designed to test the effectiveness of the new learning method over different description types in different domains.

Section 2 describes the problem area and introduces the relevant terminology. Section 3 describes the new learning methods as they are currently implemented, and presents some possible extensions. Section 4 describes experiments designed to test the learning methods and Section 5 presents the results of these experiments. Finally, Section 6 discusses the implications of the results and some directions for future research.

2. TERMINOLOGY AND DEFINITIONS

This paper deals with that subset of learning from examples known as *symbolic concept acquisition*. *Givens* are observational statements which describe objects (situations, events, etc.) that have been preclassified by a teacher. From these, the learning system is to induce a *concept recognition rule*. If an object satisfies this rule then it is considered an instance of the corresponding concept (class).

Attribute

Throughout this paper, we assume that all objects and concepts are described in terms of a finite number of discrete attributes (variables). Each attribute is assigned a finite *domain* from which it draws values and a *type* that characterizes the structure of the domain. In this study, we distinguish only between two types of attributes: nominal and linear. Nominal attributes have domains where there is no ordering on the values (e.g. 'colour') while linear attributes have domains in which the values are linearly ordered (e.g. 'length').

Event

An *event* is a symbolic description of an object. In this work, an event is represented as a vector of attribute values and is associated with a single concept (class). We assume that each event specifies exactly one legal value for every attribute. If an event is used in the learning phase, the event is called a *training* (*learning*) *event*. If it is used for testing, then it is called a *testing event*.

Selector

A *selector* is a relational statement of the form $[x\#R]$ where x is an attribute, # is a relation (one of $\geq$, $>$, $=$, $<$, $\leq$) and R is a subset of the domain of x. The selector $[x\#R]$ is said to be satisfied by the event e if the value of the attribute x in e has relation # with at least one of the values in R.

Complex

A *complex* is the logical product (conjunction) of selectors. A complex is *satisfied* by an event if every selector in the complex is satisfied by the event.

Concept description

A *concept description* is assumed to be a *disjunction* of *complexes*. A description is *satisfied by* (covers) an event if at least one complex of the disjunction is satisfied.

Decision rule

A *decision rule* is an assertion of the form D ::> C. Here D is a concept description and C is a class (concept) and ::> denotes the class assignment operator. D can therefore be viewed as an hypothesis describing C. The rule above can be interpreted as 'If an event satisfied description D, then the event is an instance of concept C.'

Star

The *star* of an event e *against the set of events* F, denoted $G(e \mid F)$, is the set of all maximal under inclusion complexes satisfied by the event e and not satisfied by any event in the set F. Informally, a star is the set of all maximally general concepts which consistently characterize a given example.

Completeness, consistency, and description types

A concept description learned from examples is *complete* if it is satisfied by all learning events which are known instances of that concept. A description is *consistent* if it is not satisfied by any learning event which is an instance of any other concept. Michalski (1983) defined a characteristic description as an expression that satisfied the completeness condition or the logical product of such expressions while a discriminant description is an expression that satisfies the completeness and consistency conditions or the logical disjunction of such expressions. Ideally, a learning system would learn the *maximal* characteristic description and the *minimal* discriminant description. In this section we will make a further distinction between conjunctive and disjunctive characteristic descriptions. A characteristic concept description is either a single conjunct listing the common properties of all learning instances of that concept or a disjunction of conjuncts which splits the learning instances of the concepts into subclasses. A characteristic disjunctive (CD) description should contain the minimum number of disjuncts and each disjunct should be as specialized (i.e. long) as possible. Note that the disjuncts in a CD description may not be disjoint and that the completeness condition still must hold.

3. METHODS AND IMPLEMENTATION

This section describes in detail the methods developed to learn descriptions incrementally from examples. Section 3.1 presents a very brief sketch of the AQ algorithm (see Michalski, 1975), as it is the base on which the method is built. Section 3.2 describes the modifications necessary to make AQ work incrementally with full memory and introduces an implementation of this method in the GEM program. Finally, Section 3.3 discusses a way to make GEM produce characteristic type descriptions.

3.1. The AQ algorithm

The AQ algorithm was conceived as a quasi-minimal solution to the general covering problem (Michalski, 1969). It has subsequently been recognized as applicable to the problem of inductive inference. This problem can be characterized as follows:

Given: a set of positive events E^+ belonging to the class for which a description is to be formed, and a set of negative events E^- belonging to other classes.

Produce: a description H that is satisfied by (covers) all the events in E^+ and none of the events in E^-.

A simplified version of the AQ algorithm applied to this problem randomly selects a seed event from a given class and generates the star for this seed. During star generation, the seed is generalized against different negative events. The results of these generalizations are intersected together to form a partial star. For efficiency reasons (Hong and Michalski, 1985), the partial star is reduced by selecting from it the preferred complexes as determined by a user-generated preference criterion. Once the reduced star is completed, the best complex in it is selected using the same criterion. The positive events covered by this complex are removed from the list of events to be covered, a new seed is selected from the remaining positive events and the process repeated. Stars are generated until there are no positive events left to cover; the disjunction of the generated complexes is a solution to the problem.

The preference criterion mentioned above is called the LEF (*lexicographical evaluation functional*). A LEF consists of an ordered set of criterion-tolerance pairs. A criterion specifies a metric to be used in judging complexes and a tolerance specifies the estimated relative error in that metric. When selecting the best complex from a list of complexes, AQ orders the complexes based on the first criterion. Complexes that are within the first tolerance of the best complex are ordered by the second criterion, and so on. The LEF provides a means of manipulating the types of descriptions produced by AQ (see Section 3.3).

3.2. Incremental learning with AQ

This section discusses extensions to the AQ algorithm which permit it to form descriptions incrementally (Becker, 1985). As shown in Figure 1, the modified algorithm must be able to apply inference rules either to training examples alone or to training examples and rules. Figure 2 shows a schematic version of the rule-modification process. The incremental method must be able to both specialize a rule so that it no longer covers a negative event and generalize a rule so that it covers a new positive event.

The incremental version of AQ begins by checking each old rule against the new events. It first determines whether any complex in these rules must be specialized. If some complex covers events which it should not, a modified version of AQ is invoked. The modified AQ procedure is

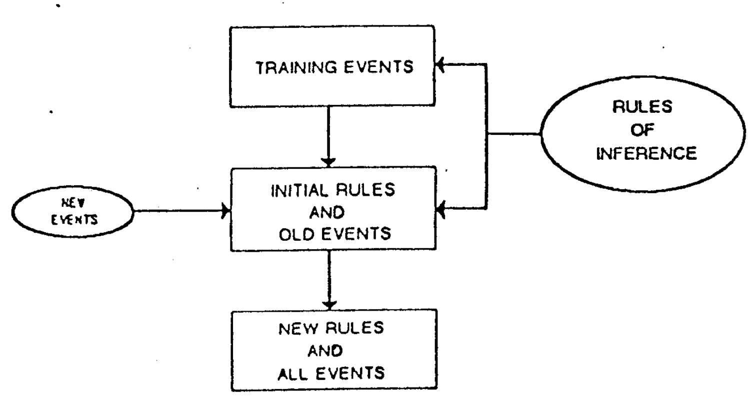

Figure 1. The initial steps in an incremental learning process.

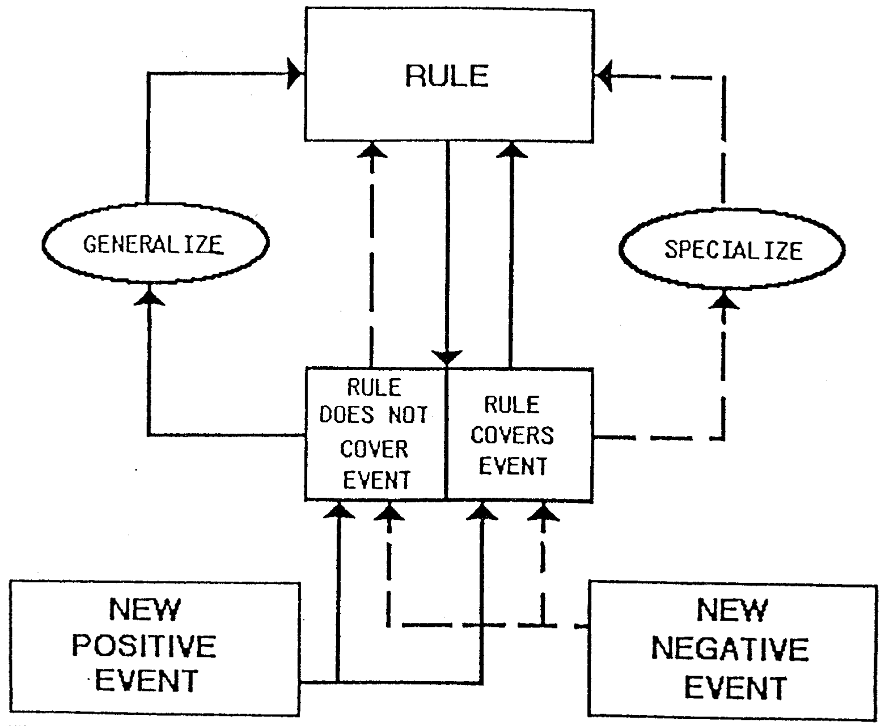

Figure 2. A schematic view of rule modification.

characterized below:

Given: a set of positive events E^+, a set of negative events E^- and a subset of the event space, *SES*.

Produce: A description H, logically contained in *SES*, such that H covers all the events in E^+ and none of the events in E^-.

This is accomplished using the normal star generation technique, except that the first partial star is intersected with *SES*.

So, to specialize a complex, incremental AQ calls the modified algorithm with the following arguments:

E^+ all positive events (both old and new) covered by the old complex

E^- the new negative events covered by the old complex.

SES the old complex.

The result is one or more new complexes, all contained in the original complex, which cover all the positive events that the original did and none of the new negative events. This is the desired specialization.

Once all rules have been specialized, they are re-generalized to cover new positive events. This is done using the standard AQ method, except that the original rules are used as seeds. The result of this second step is a rule which correctly covers both old and new events.

The potential danger here is that the time spent finding every positive event covered by a complex during the specialization step will negate any time gain caused by the retention of old rules. Further, it is possible that the specialization process will produce unduly complex rules by splitting conjuncts into disjuncts. The experiments described in Section 4 were designed to address these issues.

The incremental version of AQ has been implemented in Pascal for efficiency reasons. The program, called GEM (Generalization of Examples by Machine) consists of approximately 3500 lines of code. All input to GEM is in the form of relational tables, allowing the program to interact with the QUIN relational data base system (Spackman, 1983).

3.3. Producing characteristic descriptions with GEM

The LEF (Section 3.1) used by GEM to choose the best complex in a star can be used to manipulate the type of description learned. Typically, the first criterion in the LEF is based on the number of positive events covered by the complex. The second criterion (used to break ties in the first) may be based on the length of the complexes. If the criterion requires that the best complex is the shortest, then GEM will produce discriminant descriptions. If the criterion requires that the best complex is the longest, the result is a more detailed, characteristic type of description. Since the program must sometimes create disjunctions in order to cover all positive events, the result of learning is a CD or DD description (although conjunctive descriptions can result). Two issues must be addressed: how good are these descriptions and what is the best way to use each type in learning?

The quality of a concept description depends on its performance and its comprehensibility. Both characteristic and discriminant descriptions should perform well when tested on previously unobserved events. A good discriminant description will also be easy to use (i.e. brief) while a good characteristic description will be detailed yet easy to understand. The comprehensibility of a description is obviously a subjective matter, but it is very important. If, for example, the descriptions are to be used in an expert system, the domain expert must be able to understand the results of learning.

There are many ways to use different concept description types in learning. The most obvious way is to simply form the type of description desired at whatever time it is needed. Another possibility is to learn incrementally only characteristic descriptions. This method is attractive for two reasons. First, characteristic descriptions are more specific than discriminant descriptions; a specific description contains more information about what is being learned and is less likely to be over-generalized. Second, since GEM can induce over descriptions as well as over events, it may be possible to induce good discriminant descriptions from characteristic descriptions. This second induction step should be very fast, and will allow us to use whichever description type is most appropriate. The question remains as to the quality of discriminant descriptions produced in this way.

4. EXPERIMENTS

In order to test the new incremental learning methodology, three application domains with differing properties were chosen. These domains (described in Section 4.1) varied in size, in type of attributes, and in the degree to which events represented real world objects or situations. This range of problems provides a basis for our tentative conclusions about the effectiveness of the learning methodology. An experiment, to be repeated in all three problem areas, was designed with the following goals in mind:

1. To compare the usefulness of different description types produced by the new incremental learning method.
2. To discover whether the method of inducing discriminant descriptions from characteristic ones produces simple discriminant descriptions that will perform well.
3. To see whether the incremental learning algorithm described in section 3.2 avoids the potential problems in learning with full memory.

4.1. Problems

The first problem was the classification of different species of *Stenonema* mayfly nymphs (Lewis, 1974) based on the use of attributes for describing an individual insect's appearance. Seven species of Interpunctatum group nymphs were described in terms of seven attributes, giving a total event space size on the order of 10^6 possible descriptions. Ten different examples of each species were available.

The second application area was the King–Pawn–King black-to-move chess endgame, where the pawn's side is white. Here, examples were described in terms of 31 boolean attributes (Niblett, 1982); each example actually covered several legal KPK positions. That is, the input examples are generalized representations of the actual board positions. The examples were correctly classified (by a search program) into *Won* for the pawn's side or *Drawn*. A total of 1901 attribute vectors sufficed to represent the entire event space (which has on the order of 10^5 positions) since one attribute vector represents many positions and a large portion of the attribute space consists of illegal, impossible, or symmetrical positions.

The largest application area was the soybean disease diagnosis domain. Diseased soybean crops were described in terms of 50 attributes.

Attribute domains ranged in size from two to 11 values, meaning that approximately 10^{30} attribute vectors were possible. The event set consisted of examples of 17 different soybean diseases common in Illinois; there were 17 different examples of each disease. The data used for these experiments differed from that described by Michalski and Chilausky (1980). For the current experiments, 15 more attributes were used and two new diseases were added to the data. The entire example set was also revised and updated.

4.2. Experimental method

To determine the quality and the usefulness of the full memory incremental learning method, an experiment was devised to simulate rule base development. In each problem area, all the available events were split randomly into two groups, training events and testing events. The basic learning method was to provide GEM with successive sets of new training events, so as to simulate rule base refinement. At each step of the process, the induced rules were tested on available testing events.

In each domain, the incremental learning process started with about 20% of the available learning events. Using this learning set, decisions rules were formed. An enhanced set of learning events was created by adding a random number of learning events of each class to the original learning set. The enhanced event and the rules induced during the first step were input to GEM, which then produced refined rules. The learning set was again enhanced, and new rules produced. This process was repeated until no learning events remained.

In the mayfly nymph domain, for example, there are seven classes and a total of five available learning events per class. The initial learning set was seven events, one per class. From these seven events, rules were induced. Then, seven random numbers were generated, one for each class. The results of this process are shown in Figure 3. For class *Stenonema carolina* the random number was 0.32. There were four events remaining in this class, so one example ($4 \times 0.32 = 1.28$, rounded to 1.0) of a *Stenonema carolina* mayfly nymph was added to the learning set. For this second learning step, a total of seven events were added. So, 14 events were available to GEM for this step, seven old events and seven new ones. These events and the seven rules induced during the first step were used to form new rules.

At each step in the incremental learning process four rule types were formed:

1. A control set of discriminant rules formed using the single-step version of AQ.
2. A set of discriminant rules formed incrementally.
3. A set of characteristic rules formed incrementally.

Class	Generated Number	Available Events	Events to be Added	Total Events for this learning step
Stenonema carolina	0.32	4	1	2
Stenonema candidum	0.53	4	2	3
Stenonema floridense	0.21	4	0	1
Stenonema gildersleevei	0.06	4	0	1
Stenonema interpuc	0.89	4	3	4
Stenonema minnentonka	0.43	4	1	2
Stenonema pallidum	0.11	4	0	1
Total	–	28	7	14

Figure 3. Event selection for the second learning step in the *Stenonema* mayfly nymph domain.

4. A set of discriminant rules induced from the characteristic rules, above.

All three discriminant rule sets were tested against all available testing events. In each domain, the entire experiment was repeated with different combinations of learning and testing events. The results of these experiments are summarized in the next section.

5. EXPERIMENTAL RESULTS

Three facets of rule induction were measured. First, the rule induction time was estimated based on the c.p.u. time used by GEM in forming the rules. All results are for a Pascal version of the GEM program running under the 4.2 bsd version of the UNIX operating system on a VAX 11/780. Second, rule comprehensibility was measured. A rule's complexity, assumed to be the inverse of its comprehensibility, was defined as the sum of the number of selectors, number of different attributes and number of complexes in the rule. The complexity of a set of rules is the average of the complexities of the members. Third, the performance of the rules was estimated. Rules were tested using the ATEST program and testing examples set aside for the purpose [see Reinke (1984) for a description of ATEST and a discussion of the issues involved in rule evaluation].

5.1. Mayfly nymph extension

Figure 4 shows the c.p.u. time used by GEM to induce three different types of discriminant rules for identifying *Stenonema* mayfly nymphs. As expected, inducing DD descriptions from CD descriptions took very little time (less than 1 s of c.p.u. time in every case). The incremental method created descriptions in considerably less time than the single-step method.

Figure 5 shows the complexity of all four rule types at each stage of the learning process. The complexity of the discriminant rules induced incrementally rose at every step, undoubtedly due to the specialization of complexes. There was little difference between the characteristic rules and the discriminant rules induced from them. The second repetition of the experiment, using different learning events, produced more complex characteristic rules and simpler discriminant ones.

The performance of the three discriminant rule types is compared in Figure 6. In this domain, almost all misclassifications took place because several descriptions were satisfied by a testing event. Therefore, the DD

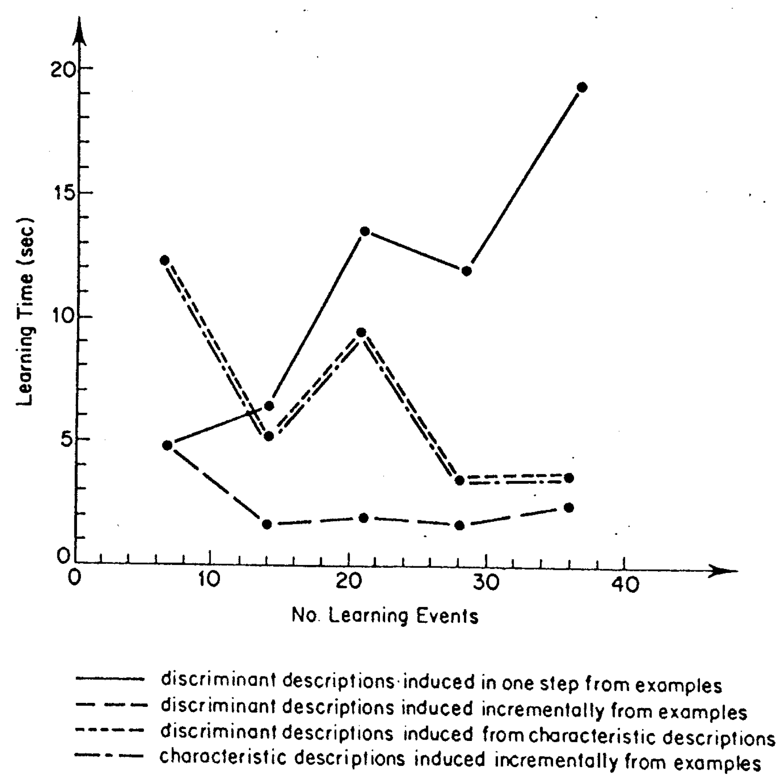

Figure 4. c.p.u. time to induce four description types for identification of mayfly nymphs.

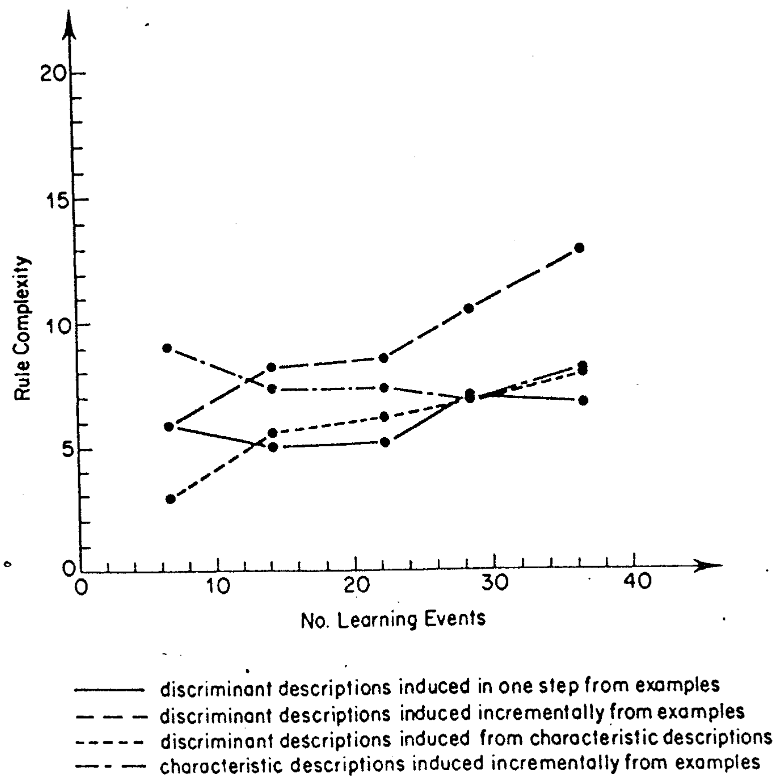

Figure 5. Complexity of four description types for identification of mayfly nymphs.

descriptions induced from CD descriptions were too general in the tests shown in Figure 6. A repetition of the experiment produced CD descriptions from which better DD descriptions were induced. Typical descriptions in this domain are shown in Figure 7.

5.2. Chess endgame position classification

In the chess endgame problem area, it was not possible to generalize the characteristic descriptions produced by GEM. For this reason, Figures 8–10 compare the two types of discriminant rules and the characteristic rules. Figure 8 shows that the incremental method saved a considerable amount of induction time in this domain.

Figure 9 compares the complexity of the three rule types over the course of the learning process. Characteristic and discriminant descriptions differed very little overall. This, and GEM's inability to generalize the long descriptions, is probably due to the nature of the attributes used to describe events. Since each input vector is really a generalization of several actual chess positions, one event may not generalize easily to

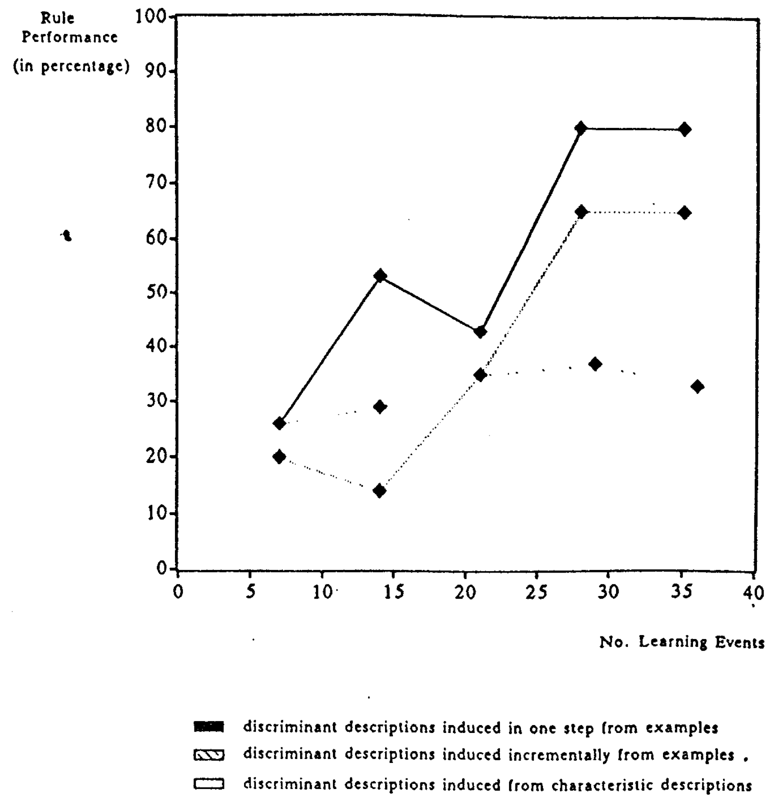

Figure 6. Performance of three description types for identification of mayfly nymphs.

cover another. This hypothesis is partially borne out by the fact that the descriptions produced were very disjunctive, containing an average of 20 complexes each. Each of these complexes would be highly specialized (i.e. characteristic) by nature, and therefore impossible to generalize.

Figure 10 shows the performance of all three description types. Unsurprisingly, the choice of learning events was very important in this domain. Two rules sets were produced by induction over two learning sets of exactly the same size, yet the rules were more than 90 per cent correct during the run shown and about 50 per cent correct during the other. This suggests that events of a given class appear in many distinct regions of the event space, and explains the highly disjunctive nature of descriptions in this domain. If learning events are taken from only a few of the regions, then rule performance will be poor. If, however, the learning events contain elements from almost all the regions, the rules should have relatively good performance. This hypothesis suggests that the better rules should have a larger number of complexes than the

Characteristic description:

[maxilla_crown_spines = 10][maxilla_lateral_setae = 21,26,28,30][inner_canine_teeth = 2]
[outer_canine_teeth = 7.8][terga_dark_posterior_margins = absent]

Discriminant description induced from characteristic description:

[maxilla_crown_spines = 10][inner_canine_teeth = 2][terga_dark_posterior_margins = absent]

Discriminant description induced from examples:

[terga_mid_dorsal_pale_streaks = absent]

Figure 7. Examples of different description types for the class *Stenonema carolina* in the mayfly nymph domain.

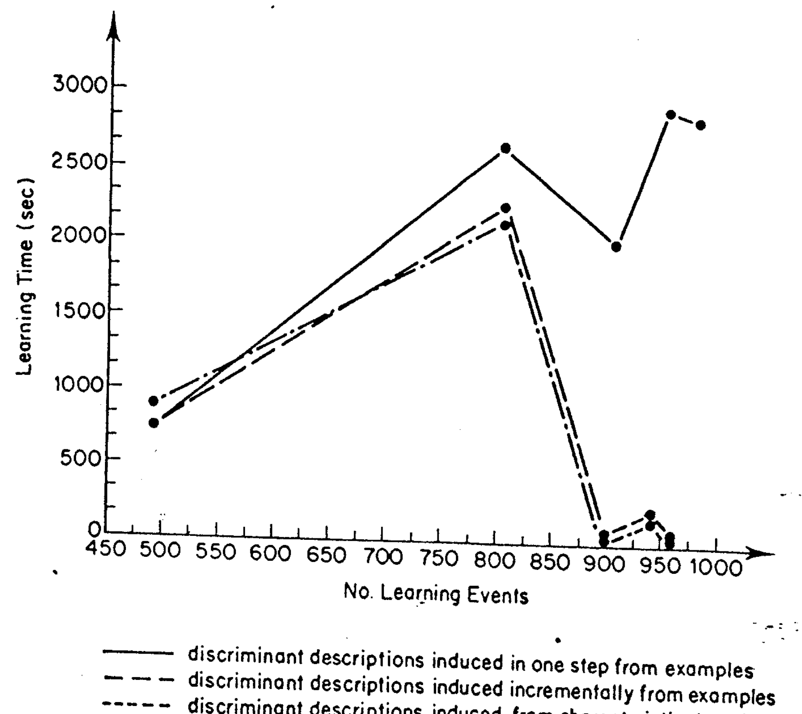

Figure 8. c.p.u. time to induce three description types for classification of KPK endgame positions.

poorer. This was indeed the case—the good rules had, on the average, almost twice as many complexes as the poorer rules. It should be noted that this effect would probably not have been observed if a chess expert had chosen the examples. Typical descriptions for this domain are shown in Figures 11 and 12.

5.3. Soybean disease diagnosis

The results for the soybean disease problem are summarized in Figures 13–15. The event space for this problem was by far the largest of the three, so rule induction took considerably longer. The time saved by using the incremental method was considerable. Again, inducing DD descriptions from CD descriptions took very little time.

Figure 14 shows the complexity of the four description types over the learning process. As expected, the characteristic descriptions were the most complex. The DD descriptions induced from CD descriptions were more complex than DD descriptions induced directly from examples.

All of the discriminant rules performed well, as shown in Figure 13. In comparison, the most recent rules written by plant pathologists were

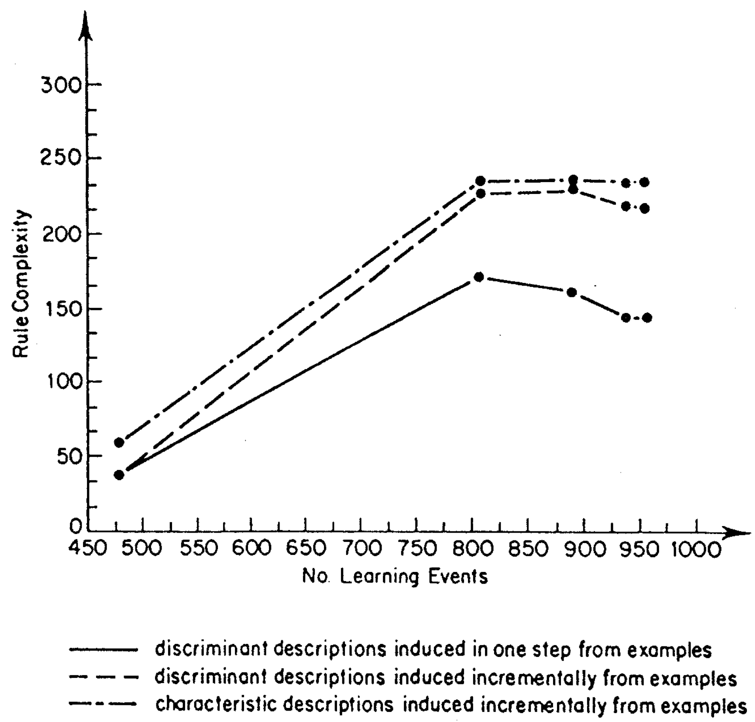

Figure 9. Complexity of three description types for classification of KPK endgame positions.

about 80% correct for these testing events. These results are similar to earlier results in the same domain (Michalski, 1980b; Niblett, 1982). Typical descriptions for this domain are shown in Figure 16.

6. SUMMARY

The experimental results are summarized below in terms of the goals sets forth in Section 4:

1. The relative quality of the various description types varied widely with the domain. In the mayfly nymph recognition domain, the incrementally learned descriptions performed poorly compared to the single step descriptions (83% correct compared to 60%). In the chess endgame domain they performed at about the same level (98–96 per cent) and in the soybean disease diagnosis domain the incrementally learned rules performed slightly better (88–82 per cent). Overall, incrementally learned discriminant disjunctive descriptions were slightly more complex than descriptions formed in a single step. Characteristic disjunctive descriptions were even more complex, as expected, but were unfortunately also more disjunctive (averaging six complexes per descrip-

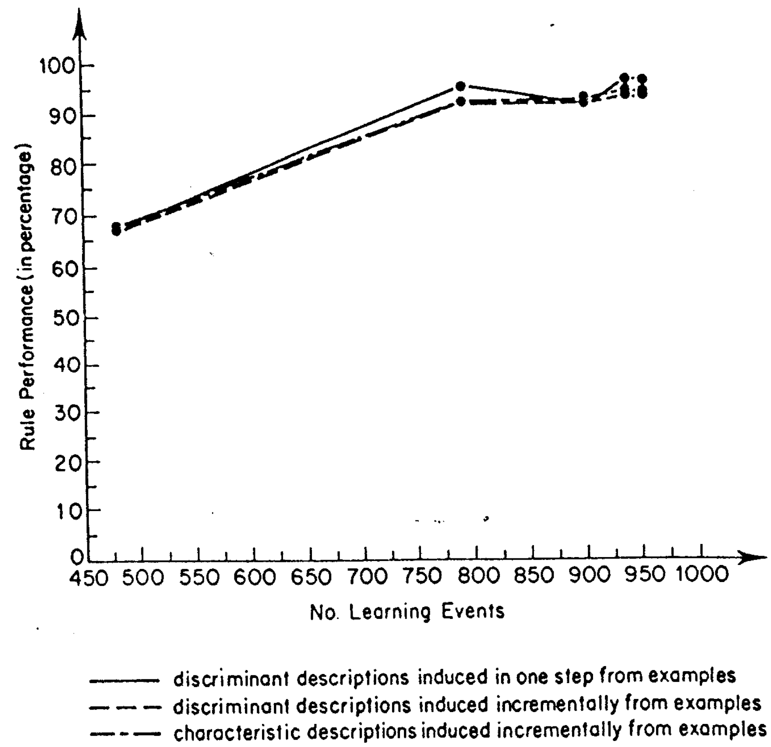

Figure 10. Performance of three description types for classification of KPK endgame positions.

tion over the three domains compared to four complexes per description for discriminant disjunctive).

2. The discriminant disjunctive descriptions formed from characteristic disjunctive descriptions performed better than the discriminant disjunctive descriptions learned from examples in two of the three domains. Overall, the performance of these descriptions was about four per cent better than that of the discriminant disjunctive descriptions induced from examples. Unfortunately, inducing discriminant disjunctive descriptions from characteristic disjunctive makes the discriminant disjunctive description more complex (the average complexity of indirectly induced descriptions was 58, compared to 41 for descriptions induced directly from examples).

3. Both incremental methods were significantly faster than single step learning (between two and five times as fast overall). Summing over all experiments in all domains, the single step method took approximately 4.2×10^3 c.p.u. minutes, the incremental method took 0.7×10^3 c.p.u. min and the characteristic disjunctive to discriminant disjunctive incremental method took 2.6×10^3 c.p.u. min.

[cimmt = f][cplu2 = f][cplu1 = f][cahea = f][cwksa = f][rrp2 = f][rsta1 = f]V
[cimmt = f][mmp2 = t][btop5 = f][spra7 = f][srfil = f]V
[cimmt = f][cplu2 = f][cplu1 = 1][cahea = f][rneac = f][rrp2 = f][rsta] = f]V
[cimmt = f][rrp2 = f][mp5 = t][spra7 = f][spran = t][srfil = f]V
[cimmt = f][cplu1 = f][rneac = f][rrp2 = f][rnear = f][rstal = f][mpmov = t][spra7 = f][srfil = t]V
[cimmt = f][cplu1 = f][cahea = f][rneac = f][rrp2 = f][rstal = f]
[mmp2 = t][srfil = t][nxto7 = t]V
[cimmt = f][cahea = f][rnear = f][btop5 = f][spra7 = t][srfil = f][nxto7 = t]V
[cimmt = f][ccrit = f][mmp1 = t][mp5 = t][spra7 = f][smain = t][srfil = f]V
[cimmt = f][rrp2 = f][mpmov = t][mp5 = f][spra7 = f][smain = t][srfil = f]V
[cimmt = f][ccrit = f][btop5 = f][spra7 = t][srfil = f][nxto7 = t]V
[cimmt = f][btop5 = f][spra7 = f][spran = 7][srfil = f]V
[cimmt = f][ccrit = f][mdiro = t][btop5 = f][spra7 = f][srfil = f]V
[cimmt = f][ccrit = f][spra7 = f][smain = t][srfil = f][sint = t]V
[cimmt = f][ccrit = f][diro5 = t][spra7 = t][spra7 = f][spran = t][srfil = f]V
[cimmt = f][cplu1 = f][cahea = f][cwksa = f][ccrit = t][rrp2 = f][rstal = f][mp5 = f][srfil = t][nxto7 = t]V
[cimmt = f][cplu1 = f][ccrit = f][rneac = f][rstal = f][mmp1 = t][mp5 = t][spra7 = f]V
[cimmt = f][cplu1 = f][rneac = f][rrp2 = f][rstal = f][mmp2 = t][mp5 = t][spra7 = f]V
[cimmt = f][cplu1 = f][ccrit = f][rneac = f][rrp1 = t][rstal = f][spra7 = f][srfil = t]V
[cimmt = f][cplu1 = f][cwksa = f][rrp2 = f][rstal = f][mpmov = t][mp5 = f][spra7 = f][srfil = t]V
[cimmt = f][rrp2 = f][r5p6 = t][spran = t][srfil = f][nxto7 = t]V
[cimmt = f][ccrit = f][mpmov = t][spra7 = f][smain = t][srfil = f]V
[cimmt = f][ccrit = t][rneac = f][rrp2 = f][rstal = f][mmp2 = t][mpmov = t][spra7 = f][srfil = t]V
[cimmt = f][cahea = f][ccrit = t][rrp2 = f][rnear = f][mp5 = t][srfil = f][nxto7 = t]V
[cimmt = f][rneac = f][rstal = f][mmp2 = t][btop5 = f][spra7 = f]

Figure 11. Typical character description for the class *Won for white* in the KPK chess endgame domain.

[cimmt = f][cplu2 = f][cplu1 = f][cahea = f][cwksa = f][rpsq = f][rrp2 = f]V
[cimmt = f][cplu1 = f][cahea = f][rneac = f][rrp2 = f[[mmp2 = t][nxto7 = t]V
[cimmt = f][cplu1 = f][ccrit = f][rneac = f][mpmov = t]V
[cimmt = f][cplu2 = f][cplu1 = f][cahea = f][cwksa = f][rrp2 = f][mp5 = t][nxto7 = t]V
[cimmt = f][cplu2 = f][cplu1 = f][cahea = f][rneac = f][rrp2 = f][srfil = t]V
[cimmt = f][cplu1 = f][rneac = f][rrp2 = f][rnear = f][mmp1 = t][mp5 = t]V
[cimmt = f][cplu1 = f][cwksa = f][rneac = f][rrp1 = t][rrp2 = f]V
[cimmt = f][rnea8 = f][rrp2 = f][rnear = f][mpmov = t][smain = t]V
[cimmt = f][cplu2 = f][cplu1 = f][rrp2 = f][mpmov = t][smain = t]V
[cimmt = f][rneac = f][mmp2 = t][btop5 = f][nxto7 = t]V
[cimmt = f][cwksa = t][rneac = f][mmp1 = t][btop5 = f]V.
[cimmt = f][cwksa = f][rrp2 = f][mp5 = t][spran = t]V
[cimmt = f][cplu1 = f][cahea = f][cwksa = f][rrp2 = f][rnear = f][mp5 = t][spra7 = t][nxto7 = t]V
[cimmt = f][cplu1 = f][cahea = f][cwksa = f][ccrit = f][mp5 = t][spra7 = t]V
[cimmt = f][cwksa = f[btop5 = f][spran = t]V
[cimmt = f][rrp2 = f][mmp2 = t][mp5 = t][smain = t]V
[cimmt = f][cplu1 = f][rrp2 = f][rnear = f][mdiro = t][mp5 = t][smain = t]V
[cimmt = f][cplu1 = f][ccrit = f][smain = t][sint = t]V
[cimmt = f][cwksa = f][rrp2 = f][diro5 = t][spran = t]V
[cimmt = f][cwksa = f][ccrit = t][rnea8 = f][rrp2 = f][rnear = f][mpmov = t][mp5 = f][srfil = t]V
[cimmt = f][cplu1 = f][cahea = f][cwksa = f][ccrit = t][rrp2 = f][rnear = f][mp5 = f][srfil = t]V
[cimmt = f][cplu1 = f][cwksa = t][rrp2 = f][mpmov = t][smain = t]V
[cimmt = f][cplu1 = f][cahea = t][rneac = f][rrp2 = f][mmp2 = t][mp5 = t]V
[cimmt = f][ccrit = f][mmp1 = t][mp5 = t][smain = t]V
[cimmt = f][cplu1 = t][rrp2 = f][r5p6 = t][spran = t]V
[cimmt = f][ccrit = f][rneac = f][mpmov = t][smain = t]V
[cimmt = f][cahea = f][cwksa = f][ccrit = t][rrp2 = f][rnear = f][mp5 = t][spra7 = t][nxto7 = t]V
[cimmt = f][cplu1 = t][rrp2 = f][mp5 = t][spran = t]V
[cimmt = f][cplu1 = t][ccrit = t][rneac = f][rrp2 = f][rnear = f][mpmov = t][srfil = t]V
[cimmt = f][cahea = t][ccrit = f][mdiro = t][btop5 = f][smain = t]

Figure 12. Typical discriminant description for the class *Won for white* in the KPK chess endgame domain.

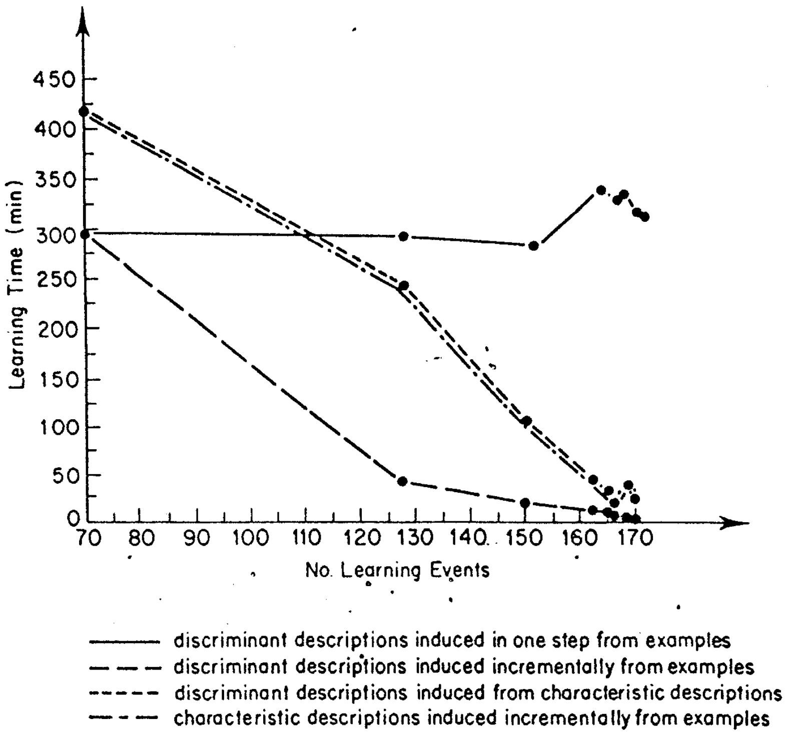

Figure 13. c.p.u. time to induce three description types for soybean disease diagnosis.

The success of the full memory incremental learning method was obvious. In all the application areas, GEM took considerably less time to form rules when it had old rules to modify. The rules produced using the incremental method were slightly more complex and performed slightly less well than those produced in a single step, but the time saved was large and the differences in performance and complexity were small. The method of inducing discriminant disjunctive descriptions from characteristic disjunctive descriptions proved workable, but produced more complex rules. This may have been due to the nature of the characteristic descriptions produced by GEM.

The incremental method could be further enhanced by simplifying both the specialization and generalization steps using the reunion operator. That is, a complex could be simplified by taking the union of the events it covers. New positive events could be covered by taking the union of the events and some complex. The method currently used could serve as a back-up, invoked only when reunion produces a complex which does not satisfy specified constraints.

The characteristic descriptions produced by GEM were sometimes unattractive because they were long and disjunctive. A combination of

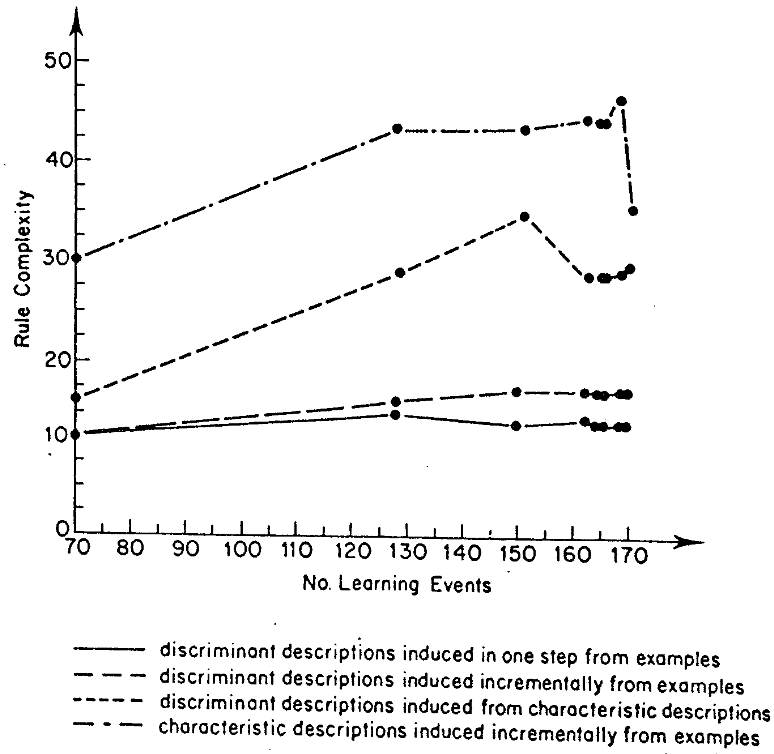

Figure 14. Complexity of four description types for soybean disease diagnosis.

two factors was responsible: the individual concepts in each domain tended to be divided into subparts and GEM always produces consistent *and* complete descriptions. Formally (see Section 2), characteristic descriptions are not necessarily consistent. Nevertheless, discriminant descriptions induced from these characteristic descriptions were often quite good.

A simple method could be used to produce conjunctive descriptions. If the disjuncts in a characteristic disjunctive concept description produced by GEM correspond to subclasses, a tree-structured concept description could be formed in the following way:

1. Induce a characteristic disjunctive description incrementally from examples.

2. Treat each disjunct as a separate class within the concept and induce a description to characterize each subclass.

Another possibility is to use a conceptual clustering method such as that described by Michalski and Stepp (1983) to divide each class into subclasses.

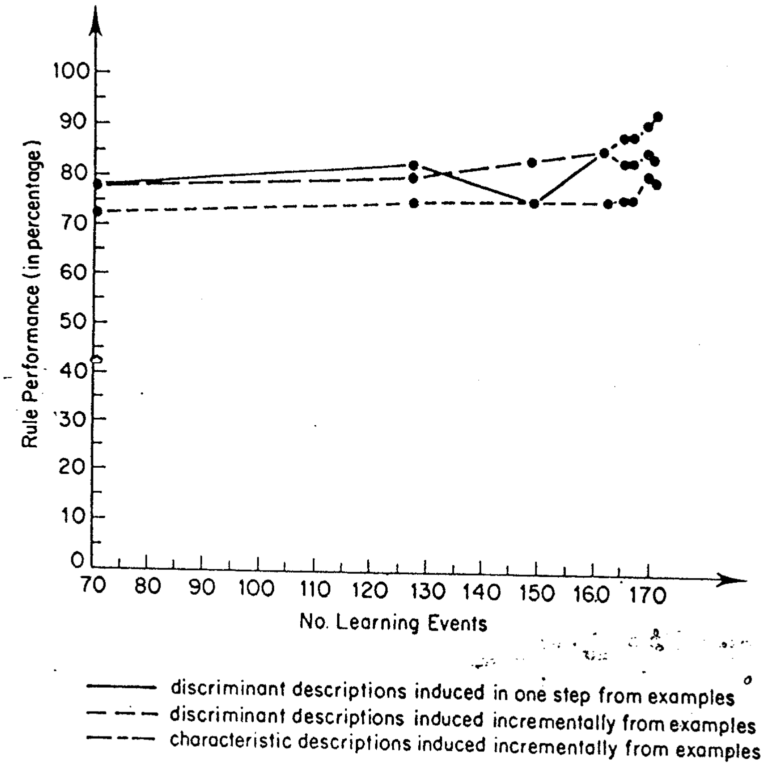

Figure 15. Performance of three description types for soybean disease diagnosis.

A more difficult extension to the method would use *partial memory* and *exceptions* (Michalski and Winston, in press). A partial memory incremental learning system would have to be able to recognize and remember 'important' events. Something like this is done by ID3 (Quinlan, 1979), which remembers one event in each parcel of events that contributed to rule formation. A true partial memory incremental learning system will need some criteria recognizing importance. Exception events which violate the consistency of conjunctive characteristic descriptions are interesting candidates. The method would have to form a characteristic conjunctive description while creating as few exception events as possible.

Unless a data base of examples is excessively large, the full memory incremental learning method provides the best way to induce reliable concept descriptions. For three real world problems, the full memory method took considerably less time and no more memory than the single step method (which must have all the events in memory anyway). Further, it appears that the best way to learn incrementally is to maintain

Characteristic description:

[precipitation = above_normal][temperature = normal . . . above_norm][severity = minor . . . potentially_severe]
[condition_of_leaves = abnormal][leaf_spot_color = brown]
[leaf_spot_growth = scattered_with_concentric_rings,necrosis_across_veins][leaf_spot_size = greater_than_eighth_inch]
[shot_holing = present][position_of_affected_leaves = scattered_on_plant]
[condition_of_leaves_below_affected_leaves = unaffected][stem_cankers = does_not_apply][fruit_spots = colored_spots]

Discriminant description induced from characteristic description:

[leaf_spot_color = brown][leaf_spot_growth = scattered_with_concentric_rings,necrosis_across_veins]
[position_of_affected_leaves = scattered_on_plant][fruit_spots = colored_spots]

Discriminant description induced from examples:

[leaf_spot_growth = scattered_with_concentric_rings,necrosis_across_veins]

Description written by domain expert;

[leaf_spot_growth = scattered_with_concentric_rings]:0.90
+
[time_of_occurrence = august . . . october][shot_holing = present]:0.50
+
[leaf_spot_size = greater_than_eighth_inch]:0.45
+
[time_of_occurence = august . . . october][fruit_pods = diseased][fruit_spots = colored_spots]:0.10
+
[seed_discoloration_color = black]:0.05
+
[leaf_spot_margins = water_soaked]:0.05
+
[yellow_leaf_spot_halos = absent]:0.05

Figure 16. Examples of different description types for the class *Alternaria Leaf Spot* in the soybean disease diagnosis domain.

characteristic descriptions of classes. Such descriptions are more appealing to humans than terse, disjunctive descriptions. The results here show that characteristic type descriptions also contain enough information that good discriminant descriptions may be induced from them in a very short amount of time.

Acknowledgements

The first phase of this research was done at the University of Illinois, where it was supported in part by the National Science Foundation under grant DCR 84-06801, and in part by the Office of Naval Research under grant N00014-82-K-0186. Subsequent research was done while the first author was at GTE Laboratories and the second author was at the Artificial Intelligence Laboratory at the Massachusetts Institute of Technology. Support for the Artificial Intelligence Laboratory's research is provided in part by the Defense Advanced Research Projects Agency under Office of Naval Research contract N00014-80-C-0505. The authors would like to thank A. Paterson for providing the chess endgame data, and Dr D. Prerau and D. Laukaitis for reading drafts of this paper.

REFERENCES

Becker, J. (1985) Topics in incremental learning of discriminant descriptions, *UIUCDCS-F-85-935*. Department of Computer Science, University of Illinois at Urbana-Champaign.

Hong, J. R. and Michalski, R. S. (1985) The general covering problem: an extension matrix method for generating disjoint stars. Report of the Intelligent Systems Group. Department of Computer Science, University of Illinois at Urbana-Champaign.

Langley, P., Bradshaw, G., and Simon, H. A. (1983) Rediscovering chemistry with the BACON System. In *Machine learning, an artificial intelligence approach* (eds R. S. Michalski, J. B. Carbonell and T. Mitchell) pp. 307–330. Tioga, Palo Alto.

Lewis, P. (1974) Taxonomy and ecology of *Stenonema* mayflies (Heptageniidae: Ephemeroptera), *EPA*-670*A*-74-006. Environmental Protection Agency, Washington DC.

Michalski, R. S. (1969) On the quasi-minimal solution of the general covering problem. *Proc. 5th Int. Symp. on Information Processing* (*FCIP* 69), Vol. 13, pp. 125–127, Bled, Yugoslavia.

Michalski, R. S. (1975) Variable-valued logic and its applications to pattern recognition and machine learning. In *Computer science and multiple-valued logic theory and applications* (ed. D. C. Rine) pp. 506–534. North-Holland, Amsterdam.

Michalski, R. S. (1980a) Pattern recognition as rule-guided inductive inference, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2**, 349–361.

Michalski, R. S. (1980b) Knowledge acquisition through conceptual clustering: a theoretical framework and an algorithm for partitioning data into conjunctive concepts. *Int. J. Policy Analysis and Information Systems* 4(3), 219–44.

Michalski, R. S. (1983) A theory and methodology of inductive learning. In *Machine learning, an artificial intelligence approach* (eds R. S. Michalski, J. B. Carbonell and T. Mitchell, pp. 83–124. Tioga, Palo Alto.

Michalski, R. S. and Larson, J. B. (1978) Selection of most representative training examples and incremental generation of VL_1 hypotheses: the underlying methodology and descriptions of programs ESEL and AQ11, *Report* 867. Department of Computer Science, University of Illinois at Urbana-Champaign.

Michalski, R. S. and Chilausky, R. L. (1980) Learning by being told and learning from examples: an experimental comparison of two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *Int. J. Policy Analysis and Information Systems* 4(2), 125–160.

Michalski, R. S. and Stepp, R. E. (1982) Revealing conceptual structure in data by inductive inference. *Machine intelligence* 10 (eds J. E. Hayes, D. Michie & Y.-H. Pao) pp. 173–195. Ellis Horwood, Chichester; Halsted, New York.

Michalski, R. S. and Dietterich, T. G. (1983) A comparative review of selected methods for learning from examples. *Machine learning, an artificial intelligence approach* (eds R. S. Michalski, J. B. Carbonell and T. Mitchell) pp. 41–75. Tioga, Palo Alto.

Michalski, R. S. and Stepp, R. E. (1983) Learning from observation: conceptual clustering. *Machine learning, an artificial intelligence approach,* (eds R. S. Michalski, J. B. Carbonell and T. Mitchell) pp. 331–363. Tioga, Palo Alto.

Michalski, R. S. and Winston, P. H. (1985) Variable precision logic, *AI Memo 857,* Cambridge: Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Mass.

Michalski, R. S., Davis, J. H., Bisht, V. S., and Sinclair, J. B. (1983) A computer-based advisory system for diagnosing soybean diseases in Illinois. *Plant Disease* 67(4), 459–63.

Niblett, T. B. (1982) A provably correct advice strategy for the endgame of king and pawn versus king. *Machine intelligence* 10 (eds J. E. Hayes, D. Michie and Y.-H. Pao) pp. 101–122. Ellis Horwood, Chichester/Wiley, New York.

Quinlan, J. R. (1979) Discovering rules by induction from large collections of examples. *Expert systems in the micro electronic age* (ed. D. Michie) pp. 168–201. Edinburgh University Press, Edinburgh.

Quinlan, J. R. (1982) Semi-autonomous acquisition of pattern based knowledge. *Machine Intelligence* 10 (eds J. E. Hayes, D. Michie and Y.-H. Pao) pp. 159–172. Ellis Horwood, Chichester/Wiley, New York.

Reinke, R. E. (1984) Knowledge acquisition and refinement tools for the ADVISE meta-expert system, *UIUCDCS-F*-84-921. Department of Computer Science, University of Illinois at Urbana-Champaign.

Spackman, K. A. (1983) QUIN: integration of inferential operators within a relational data base, *UIUCDCS-F*-83-917. Department of Computer Science, University of Illinois at Urbana-Champaign.

Winston, P. H. (1975) Learning structural descriptions from examples. In *The psychology of computer vision* (ed. P. H. Winston). McGraw-Hill, New York.