Learning Novel Domains Through Curiosity and Conjecture

Paul D. Scott & Shaul Markovitch
Center for Machine Intelligence
2001,Common wealth Blvd.,
Ann Arbor, Michigan 48105

Abstract

This paper describes DIDO, a system we have developed to carry out exploratory learning of unfamiliar domains without assistance from an external teacher. The program incorporates novel approaches to experience generation and representation generation. The experience generator uses a heuristic based on Shannon's uncertainty function to find informative examples. The representation generator makes conjectures on the basis of small amounts of evidence and retracts them if they prove to be wrong or useless. A number of experiments arc described which demonstrate that the system can distribute its learning resources to steadily acquire a good representation of the whole of a domain, and that the system can readily acquire both disjunctive and conjunctive concepts even in the presence of noise.

1. Introduction

This paper gives an account of DIDO, a learning system we have developed to carry out exploratory learning of unfamiliar domains*. We define the exploratory learning problem as follows:

Situation: An intelligent agent is placed in a novel domain comprised of a large number of entities. The agent has two types of interaction with the domain. It is equipped with a finite set of motor operations which can be applied to entities in the domain. It is also equipped with a finite set of perceptual operations which enable it to perceive the current state of any entities present in its locality. The agent has no prior knowledge of the effect of any of its motor operations on any entity.

Task: The agent knows that it may eventually have to solve problems in the domain but does not have any prior knowledge of what those problems might be. The agent must try to discover what effect its possible actions will have in any circumstances. More specifically, its task is to build a representation of the domain which will enable it to predict the outcomes of each of its motor operations in any situation. This is the basic knowledge the agent would need to engage in problem solving.

Constraints: The domain is very large and the behavior of entities may be non-deterministic and exhibit change over time. No other agent is available to provide assistance in this knowledge acquisition task. The agent has only limited resources of time and memory available. Since it is not known when problem solving will be necessary, it is desirable that the quality of the representation should rise steadily as learning proceeds.

This problem is significantly different from those addressed by other machine learning programs. The absence of prior knowledge means that deductive learning methods cannot be applied. Any solution must be based on inductive inference. The most widely studied type of inductive inference problem is that of learning concepts from sets of classified examples (eg.Winston, 1975; Michalski & Dietterich, 1983; Quinlan, 1983, 1986). The problem stated above differs from this type of problem in two respects: there is no external agent to supply classifications of examples, and there is no predefined classification scheme that the agent is trying to discover. A number of learning programs have been written which do not require such a predefined classification scheme (eg. Michalski & Stepp, 1983; Fisher & Langley, 1986). The task performed by such systems, which is commonly termed 'conceptual clustering', involves developing a classification scheme for a set of examples. Such programs attempt to construct a parsimonious taxonomy to cover the example set on the basis of similarity between examples. The learning problem considered here resembles conceptual clustering in that the learning system must develop its own classification scheme. On the other hand it differs significantly in that the basis for forming classes is not similarity between examples but rather similarity of their behavior when subject to the agent's motor operations. Thus the classification scheme developed is determined both by characteristics of the domain and by characteristics of the agent. In this sense the agent's task is to build an egocentric representation of its world.

Learning involves two distinct but interrelated search processes (Shalin *et al*$_y$ 1987): a search for a good representation in the space of possible representations (Simon & Lea, 1974; Mitchell, 1977, 1979), and a search for informative examples in the space of possible

The work reported in this paper was supported by NSF Grant# MCS-8203956.

experiences (Scott & Vogt 1983). The information obtained from the experiences is used to guide the search for a better representation, while the current state of the representation may be used to guide the search for informative experiences.

The component of a learning system responsible for searching experience space is the *experience generator.* Section 2 describes DIDO's experience generator which is based on a formalization of the concept of curiosity. The search of representation space is carried out by the *representation generator.* DIDO's representation generator is described in Section 3. Section 4 describes results obtained running an implementation of DIDO. Finally, Section 5 contains a discussion of the implications of this work.

2. Curiosity Driven Search of Experience Space

The task of an inductive learning system's experience generator is to select informative examples from the experience space and supply them to the representation generator. An example is informative only if it leads to some change in the representation. The informativeness of an example is not an inherent property of the example since it depends on both the domain and the current representation. Unfortunately examples are not equally informative and some may carry no information at all.

A large number of learning systems avoid the need for a built in experience generator by requiring an external agent (usually human) to serve this role (See Michalski & Dietterich, 1983 for review). Winston (1975) demonstrated that such an external teacher can greatly speed the learning process by selecting examples which are highly informative.

DIDO has no such external assistance. Furthermore DIDO has no preassigned class of problems to solve and so cannot make use of successes and failures at problem solving to guide the search for informative experiences (eg LEX (Mitchell *et al,* 1983); PRODIGY (Carbonell & Gill, 1987)). What DIDO needs is a method of exploring domains which is analogous to human curiosity. One program which attempts to generate its own experiences under similar constraints is AM (Lenat 1982). Lenat's formalization of curiosity was a set of 54 heuristic rules for assessing the 'interestingness' of objects. The objection to using such an elaborate implementation of curiosity is that it is not clear either how much each rule contributes, or how domain specific they were. We have provided DIDO with a simple domain independent method for selecting new experiences.

Given the analogy with human curiosity it might seem reasonable to build an experience generator which favored highly novel objects. However, further consideration shows that such a system would not be very effective. It would provide the system with a large number of highly dissimilar experiences. These would form a poor foundation for a coherent and integrated representation of the domain. Such a system would learn very little about a lot.

The weakness of the novelty heuristic suggests that its opposite might be effective. That is the generator would supply experiences which were as similar as possible to those already encountered. The argument in favor of this approach is that it would seem to have the potential to generate 'near misses' (Winston 1975) quite frequently. However further consideration shows that this familiarity heuristic is likely to be as ineffective as novelty. The experience generator would have no tendency to move outside one small region of the space of possible experiences. Hence such a system would learn a lot about very little.

Clearly what is needed is something that falls somewhere between novelty and familiarity. Such a heuristic would supply experiences which could be related to previous experiences without becoming trapped in a small region of the experience space. This behavior is produced by using a heuristic based on uncertainty. The fundamental idea is that subsequent experiences should be generated in order to resolve uncertainties created through earlier experiences.

DIDO is attempting to build a representation which predicts the outcomes of applying operations to entities with a maximum likelihood of being correct. This representation takes the form of an inheritance network of classes. Each class contains a number of *practical conditionals.* Each practical conditional represents the knowledge that DIDO has acquired about the consequences of applying a particular operation to entities which are members of that class. The form of a practical conditional is as follows:

(Op ((Outcome$_1$ P$_1$)(Outcome2 P$_2$).(OutcomeN PN)))

where each Pj is an estimate, based on experiences, of the probability that the corresponding outcome will occur. Probabilities are estimated using an exponential lag and outcomes with very small probability estimates arc discarded. If there is more than one outcome in the list DIDO is uncertain of the outcome. This uncertainty can be expressed precisely using the Shannon uncertainty function (Shannon & Weaver, 1949) which is defined as:

$$H = -\sum_{i=1}^{n}(p_i \times \log_2(p_i))$$

where the sum is taken over all alternative outcomes. (Note that ID3 (Quinlan, 1983, 1986) also uses this uncertainty function but for a very different purpose. ID3 uses it as part of the representation generator to determine which subclasses to add to the current representation. DIDO uses it as part of the experience generator to determine what new experiences to select.)

There is thus an uncertainty associated with every practical conditional in every class in DIDO's current representation. DIDO's experience generator finds the highest of these uncertainties, and selects a small number of entities that are members of the class concerned. The operation appearing in the most uncertain practical conditional is then applied to each of these entities and the outcomes noted. This procedure is termed a *round of experiments.* Normally the outcomes will change the probabilities and hence lead to a change in uncertainty. DIDO attempts to build a class network in which the uncertainties are minimized. The way this happens is described in the next section.

The use of uncertainty to guide the search of experience space has a number of advantages. First it is simple and domain independent. It can be applied in any system in which it is possible to estimate probabilities of assertions being correct. Second, it generates highly informative experiences. All the experiences produced are examples of classes of experience about which the learning system is doubtful. Third, it makes efficient and effective use of resources. Because the system is continually directing its attention to the area of highest uncertainty in the current representation, the level of uncertainty remains approximately uniform across the whole representation. This means that the system is steadily reducing its uncertainty about the whole of the experience space.

It is also worth noting that this was achieved because the representation included sets of contradictory assertions (alternative outcomes in this case) together with estimates of the likelihood that they were correct. This enabled DIDO to both identify and quantify the degree of doubt associated with what appears to be known. While the need to include measures of uncertainty in representations of uncertain domains has been recognized and explored for many years, the primary use of uncertainty in DIDO is to represent the system's uncertainty about the domain. Such uncertainty can arise either because the domain is inherently uncertain or, as is more commonly the case, because the current representation is inadequate.

3. Conjecture Based Search of Representation Space

DIDO builds an object based representation of a domain. The major feature of this is an inheritance network of classes. The members of each class are the entities which occur in the domain. Every entity is a member of at least one class because DIDO begins with a representation comprising a single class , *Things,* which includes all entities.

Classes have two important components, the intension and the practical conditional set. The intension is a conjunctive predicate whose terms are built up from DIDO's set of sensory operations. Any entity which satisfies the intension is a member of the class. Practical conditionals were described in the preceding section. The practical conditional set represents DIDO's current knowledge of how members of the class behave when an operation is applied to them. It is unnecessary for there to be a practical conditional corresponding to every action in every class because practical conditionals can be inherited from superclasses. A practical conditional for a particular action only appears in a class when it is needed to shadow the practical conditional for the same action in a superclass. Classes may have more than one immediate superclass. Multiple inheritance conflicts do not occur because of the way in which the network is built.

Two different approaches to representation generation are possible. The first is to be very cautious about making changes to the representation and only make changes when enough evidence has been accumulated to establish that they arc highly likely to be correct. The advantage of this approach is that it is unnecessary to retract any changes. The disadvantage is that the system is precluded from making and using conjectures for which there is only limited evidence. DIDO uses the opposite approach. Changes are made tentatively on the basis of small amounts of evidence. If the conjectures prove to be wrong or of no value, they are rapidly discarded. We call such readily retracted changes *conjectures.* (See Markovitch & Scott (1988) for discussion of knowledge which is not beneficial.)

The representation generation process is basically one of learning by specialization (sometimes termed 'discrimination learning'). At the conclusion of each round of experiments, DIDO will have two samples of entities: successes, defined as those that gave the most probable outcome; and failures, that gave some other outcome. Typically these samples are small. DIDO finds the sensory attribute value which correlates most highly with the most probable outcome and forms a new subclass having this as its intension. This subclass may already exist, since it might have been created previously as part of an attempt to reduce the uncertainty of a different practical conditional. If it does not exist DIDO creates it. DIDO then adds the practical conditional to this specialization class. If the specialization class has no other ancestors but the class in which the round of experiments was performed, this is straightforward. DIDO simply copies the practical conditional from the parent class. However, it may be that the specialization class has other ancestors. If there is more than one ancestor containing the practical conditional, the practical conditional with the highest probability for its expected outcome is copied into the specialization class.

This specialization process serves to extend the network to ever more specialized classes. It is balanced by two processes: *retraction* and *generalization.* Retraction is the process by which incorrect or useless conjectures are discarded. Every time the probabilities of a practical conditional are updated it is checked for usefulness. A practical conditional is deemed useless if removing it would not lead either to different predictions or to predictions with lower confidence. Useless practical conditionals are removed. Classes which have neither subclasses nor practical conditionals are also deemed useless and removed.

Practical conditionals containing only one outcome (which must have a probability of 1.0 and zero uncertainty) are said to be *fixated.* No experiments are performed regarding fixated practical conditionals. Hence DIDO's learning process is complete when all practical conditionals in the representation have fixated.

Generalization is a process which guards against overspecialization. Whenever a class fixates, DIDO attempts to form a generalization by searching for other similar classes which contain non-fixated practical

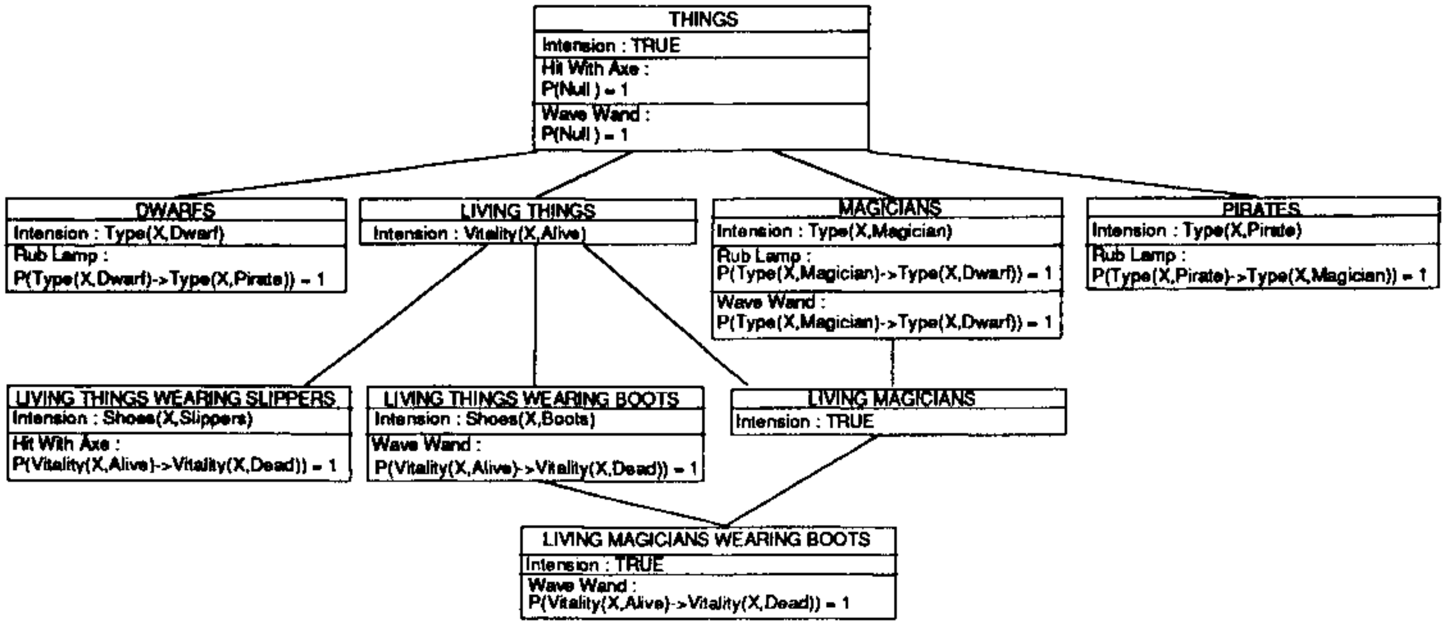
THINGS
Intension : TRUE
Hit With Axe :
P(Null) = 1
Wave Wand :
P(Null) = 1
DWARFS
Intension : Type(X,Dwarf)
Rub Lamp :
P(Type(X,Dwarf)->Type(X,Pirate)) = 1
LIVING THINGS
Intension : Vitality(X,Alive)
MAGICIANS
Intension : Type(X,Magician)
Rub Lamp :
P(Type(X,Magician)->Type(X,Dwarf)) = 1
Wave Wand :
P(Type(X,Magician)->Type(X,Dwarf)) = 1
PIRATES
Intension : Type(X,Pirate)
Rub Lamp :
P(Type(X,Pirate)->Type(X,Magician)) = 1
LIVING THINGS WEARING SLIPPERS
Intension : Shoes(X,Slippers)
Hit With Axe :
P(Vitality(X,Alive)->Vitality(X,Dead)) = 1
LIVING THINGS WEARING BOOTS
Intension : Shoes(X,Boots)
Wave Wand :
P(Vitality(X,Alive)->Vitality(X,Dead)) = 1
LIVING MAGICIANS
Intension : TRUE
LIVING MAGICIANS WEARING BOOTS
Intension : TRUE
Wave Wand :
P(Vitality(X,Alive)->Vitality(X,Dead)) = 1

Figure 1. The final state of the class hierarchy. Note that intension fields only show terms not inherited from superclasses and that class names have been added manually.

conditionals having the same expected outcomes as the fixated practical conditional. If such a class is found a generalization class is formed which is the most specific superclass of the two classes being generalized. The non-fixated practical conditional is copied into this new class. If the fixated practical conditional was an overspecialization it will become useless and be discarded when fixation occurs in the more general class.

DIDO also makes use of a number of other methods for changing the representation. These handle contingencies which only arise occasionally (eg formation of redundant partitions of a class). (See Scott & Markovitch (1988) for details).

4. Experimental Runs

We have implemented DIDO and conducted a large number of experiments, using various domains. Here we describe a selection of these.

A Typical Run

This example, which illustrates DIDO's ability to discover the consequences of each member of the set of motor operations, involves a domain which might be part of an Adventure type game. The entities DIDO may encounter have 5 attributes whose possible values are as follows:

Property:	*Possible Values*
Type:	Dwarf, Magician, Pirate
Size:	Small, Medium, Large
Vitality:	Alive, Dead
Coat Color:	Red, Blue, Green, Yellow, Black, White, Gold
Shoes:	Leather, Boots, Slippers, Sandals, None

The actions available are Hit-With-Axe, Wave-Wand, and Rub-Lamp. Their consequences are:

Hit-With-Axe:	Kills any living thing wearing slippers.
Wave-Wand:	Kills any living thing wearing boots. Turns any magician into a dwarf.
Rub-Lamp:	Turns any dwarf into a pirate. Turns any pirate into a magician. Turns any magician into a dwarf.

Figure 1 shows the final representation DIDO achieved after 417 trials, at which point all the classes shown had fixated. As can be seen, this is a complete, consistent, and correct representation of the domain. Furthermore, it is minimal in the sense that no representation with fewer classes is possible.

Figure 2 shows the steady progress of the system during learning. Figure 2a shows the 'exam score'. During an exam, the learning is turned off and a random batch of 100 entities is shown to the system. The score is the percentage of outcomes predicted correctly. Exams arc administered at regular intervals to monitor the progress of learning. Note that 100% score is reached long before learning is complete. Figure 2b shows the steady decline in average uncertainty. This figure also shows why learning is not complete when DIDO starts scoring 100% in exams. Although the representation is correct, DIDO is not yet certain it is correct. During the last 300 trials DIDO is discovering that the representation achieved by trial 120 is a correct one.

Classification — Conjunction and Disjunction

As we noted in the introduction, the task DIDO performs is different from that performed by other concept learning programs. In particular it differs from those that learn concepts from classified examples since there is no teacher to provide classified examples. This makes it hard to compare DIDO's performance with that of other systems.

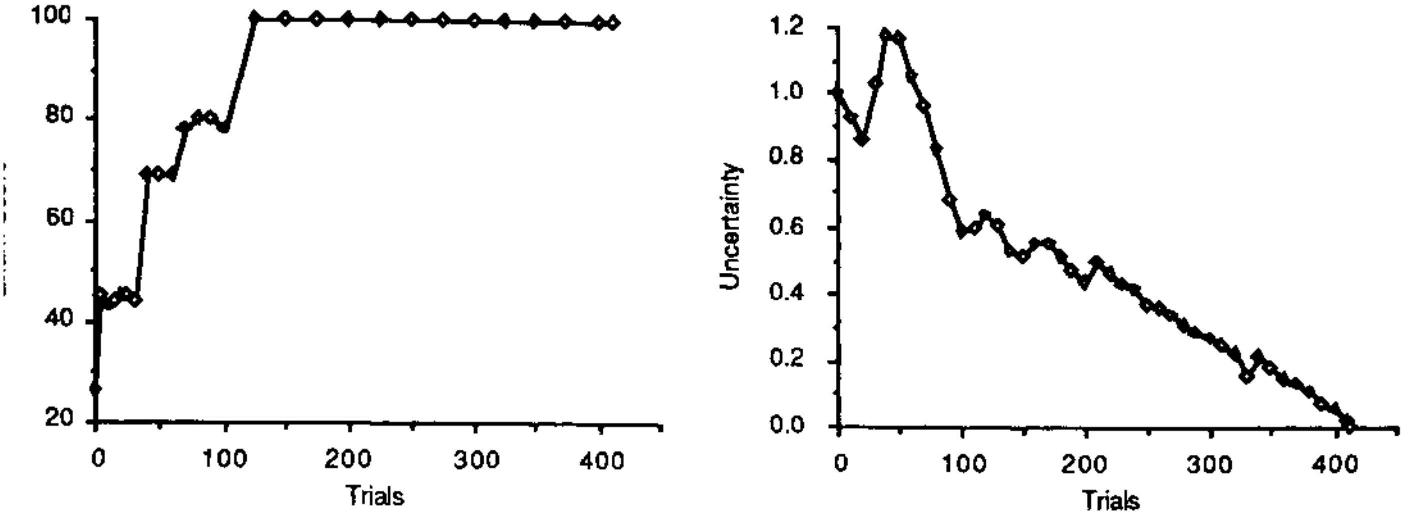

Figure 2. Performance during typical run.

Since DIDO's representation generator involves a modified form of learning by specialization, a comparison with other programs which also learn by specializing general concepts (See Langley 1987 for review) would seem pertinent. In order to achieve this we ran DIDO with a single motor operation Ask-For-Category. The outcome of applying this operation to an entity is a string which is the name of the category to which the entity belongs. In this way it is possible to get DIDO to learn a predefined classification scheme and hence compare the performance of its representation generator with those incorporated in other learning programs.

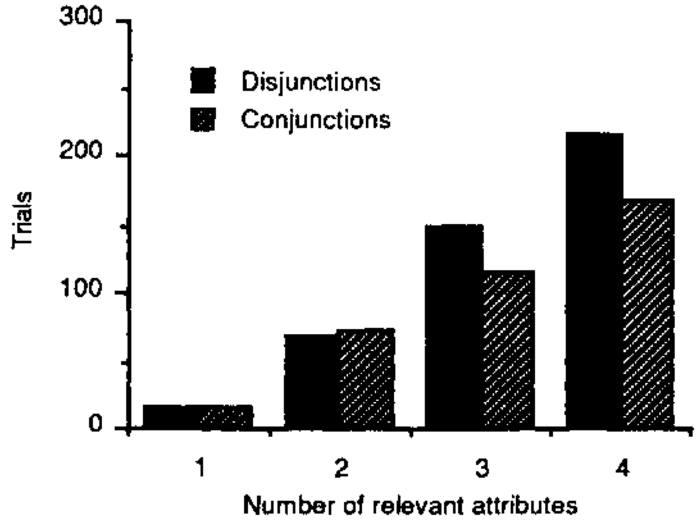

Figure 3. Mean number of trials to reach 100% exam score, taken over 20 runs.

Figure 3 shows the results obtained when DIDO learned conjunctive and disjunctive concepts. Every entity had 5 attributes, and each of these took one of 5 possible values. Relevant attributes were those that formed part of the definition of the category to be discovered. The graphs show the number of trials needed to reach 100% exam scores. As can be seen, both conjunctive and disjunctive concepts are readily discovered and, at least over the range considered, the number of examples needed seems to increase linearly with the number of relevant attributes. Furthermore DIDO takes slightly longer to discover disjunctive concepts. This result is in marked contrast to the results reported by Langley (1987) on PRISM, a program which learns by specialization. He found that conjunctive concepts took much longer to discover than disjunctions, and that the number of examples per relevant attribute increased non-linearly for conjunctions.

Noise And Change

In order to test DIDO's ability to operate successfully in non-deterministic domains, we performed a series of classification tasks using varying levels of noise. Various authors have defined noise level in different ways (eg Langley, 1987; Quinlan, 1986). In our experiments noise level is simply a measure of the probability that the classification of an example will be inverted. This is equivalent to combining positive and negative noise following Langley's (1987) definition. The concept to be learned was a conjunction with 2 relevant attributes. Various noise levels between zero and 30% were tried. To test the system, exams were run with zero noise.

For noise levels up to 15% the effect was simply to slow down the speed at which DIDO converged on the best approximate representation. For noise levels of 20% and above DIDO did not achieve the best approximation in 1000 trials. She did however achieve a non-fixated representation in which the error level in exams (which were noise free) was considerably less than the noise level in the training examples. Thus in the worst case, 30% noise level, she achieved a representation which produced only 11% exam errors. At these higher levels of noise the uncertainty in DIDO's final representation was the same as the inherent uncertainty of the domain.

5. Discussion

How successful is DIDO at solving exploratory problems as defined in the introductory section of this paper? Our original purpose in developing DIDO was to determine whether the concept of uncertainty could form a simple basis for implementing curiosity. The results of our experiments show that using this heuristic to explore experience space does lead to a sensible choice of experiences and a consequent convergence on a good representation. The heuristic leads to efficient use of resources as is clearly shown in Figure 2. DIDO achieves 100% exam scores long before all the practical conditionals have fixated. This happens because learning resources are switched between the operations available so that the amount of remaining uncertainty regarding each is about the same. The success is achieved quickly. There are 6 rules to discover and DIDO achieves 100% success after performing only 6% of the experiments possible in the domain. Complete certainty is achieved after performing 21%. DIDO also uses space parsimoniously. Individual experiences are not retained beyond the round of experiments in which they are generated. The number of classes in the network seldom exceeds double the number in the final solution.

The results obtained in classification experiments and in noisy domains show that the conjecture based method of learning, in which changes in the representation can be made rather freely because mistakes are readily discovered and retracted, may be of use in other types of concept learning. Of particular interest is the absence of any marked non-linearity in the time taken to learn progressively more complex conjunctions and disjunctions. However, there arc limitations to this method of representation generation. DIDO performs poorly at finding exclusive disjunctions and cannot solve multiplexor problems (Wilson 1987). Interestingly, people also find such concepts difficult to discover. In DIDO, these limitations arise as a direct consequence of the features which give rise to the speed at which correct answers are found for conjunction and disjunction. Exclusive disjunction and the multiplexor problems are hard because partial solutions do not yield better than chance results. DIDO's readiness to retract apparently useless conjectures, which is the necessary price of making conjectures freely, leads to the abandonment of partially elaborated correct solutions.

DIDO's learning procedures could be applied to practical problems. In particular DIDO could be used to analyze empirical data such as very large databases composed of case histories paired with their eventual outcomes. The set of case histories would constitute DIDO's problem domain. Using uncertainty guided search, case histories would be sampled in an attempt to learn to predict associated outcomes. The distinctive feature of DIDO's uncertainty guided search is that the sampling of the data set would concentrate on those areas about which remaining uncertainty was highest Thus, once DIDO has learned the commonly occurring cases, resources will not be wasted considering further examples of those types, but will concentrate on seeking out and learning to predict the less common types.

References

CarbonellJ. & Gill,Y. 1987, *Learning By Experimentation,* Proc. 4th International Workshop on Machine Learning, UC Irvine, June 1987

Fisher J). & LangleyP. 1986 , *Conceptual Clustering and Its Relation to Numerical Taxonomy.* In *Artificial Intelligence and Statistics* ed. W.A.Gale, Addison-Wesley

Langley,P. 1987, *A General Theory of Discrimination Learning.* In *Production System Models of Learning and Development,* Eds D.Klahr, P.Langley & R. Neches, MIT Press.

Lenat,D. 1982, *AM: Discovery in Mathematics and Heuristic Search* . In Davis.R. & Lenat,D.B. k*nowledge-based Systems in Artificial Intelligence,* McGraw-Hill, New York

Markovitch,S. & Scott,P.D., 1988, *The Role of Forgetting in Learning,* Proc Fifth International Conference on Machine Learning, Ann Arbor, June 1988.

Michalski,R., CarbonellJ., & Mitchell.T. (eds), 1983, *Machine Learning ,* Tioga Press, Palo Alto

Michalski,R., CarbonellJ., & Mitchell,T. (eds), 1986, *Machine Learning, Vol II,* Morgan Kauffman, Los Altos

Michalski,R. & Dietterich,T., 1983, *A Comparative Review of Selected Methods for Learning from Examples,* In Michalski, Carbonell, & Mitchell (1983)

Michalski,R. & Stepp,R., 1983, *Learning from Observation: Conceptual Clustering.* In Michalski, Carbonell, & Mitchell (1983)

Mitchell,T.M. , 1977 , *Version Spaces: A Candidate Elimination Approach to Rule Learning* , Proceedings Fifth International Joint Conference on Artificial Intelligence, Cambridge, Mass. pp 305-310

Mitchell,T.M. , 1979 , *An Analysis of Generalization as a Search Problem* , Proceedings Sixth International Joint Conference on Artificial Intelligence, Tokyo, Japan.

Mitchell,T.M., Utgoff,P.E. & Banerji,R., 1983, *Learning by Experimentation: Acquiring and Refining Problem Solving Heuristics.* In Michalski, Carbonell & Mitchell (1983)

QuinlanJ., 1983, *Learning Efficient Classification Procedures and their Application to Chess End Games.* In Michalski, Carbonell, & Mitchell (1983)

QuinlanJ., 1986, *The Effect of Noise on Concept Learning Application.* In Michalski, Carbonell, & Mitchell (1986)

Scott, P.D. & Markovitch, S. , 1988 , *Learning Through Curiosity and Conjecture* , CMI Research Paper #030988

Scott, P.D. & Vogt, R.C. , 1983 , *Knowledge Oriented Learning ,* Proc. 8th IJCAI, Karlsruhe, August 1983

Shalin,V.L., Wisniewski,E.J., Levi,K.R. & Scott,P.D., 1987, *A Formal Analysis of Machine Learning Systems for Knowledge Acquisition,* Proc. 2nd AAAI Knowledge Acquisition for Knowledge Based Systems Workshop, Banff, Canada, October 1987

Shannon ,C.E& W.Weaver, 1949, *The Mathematical Theory of Communication,* University of Illinois Press

Simon,H.A. & Lea, G., 1974 , *Problem Solving and Rule Induction: A Unified View,* In Gregg, L.W. (ed) , *Knowledge and Cognition,* Lawrence Erlbaum,Maryland.

Wilson,S.W., 1987, *Classifier Systems and the Animat Problem,* Machine Learning, 2, pp 199-228.

Winston,P.H. , 1975 , *Learning Structural Descriptions from Examples* , In *The Psychology of Computer Vision* ed P.H.Winston,McGraw-Hill