ANALOGICAL PROCESSES

IN MACHINE LEARNING

ALEX BAVELAS, NEWMAN LAM, ERIC LEE, JAMES MACGREGOR

University of Victoria

Victoria, B.C.

and

IAN MORRISON

Carnegie-Mellon University

Pittsburgh, PA

DOC Contract No. 19SV.36100-4-4153

DOC Scientific Authority: Bill Treurniet

ANALOGICAL PROCESSES

IN MACHINE LEARNING

ALEX BAVELAS, NEWMAN LAM, ERIC LEE, JAMES MACGREGOR

University of Victoria

Victoria, B.C.

and

IAN MORRISON

Carnegie-Mellon University

Pittsburgh, PA

DOC Contract No. 19SV.36100-4-4153

DOC Scientific Authority: Bill Treurniet

TABLE OF CONTENTS

ABSTRACT

The objective in machine learning is to develop programs which can improve their performance by learning. The focus of the report is on learning by analogy, one of the most powerful, yet least investigated, forms of learning. For people, analogy plays an important role in creativity, in scientific discovery, in language, and in common-sense reasoning. The extent to which machines can capitalize on this source of power presumably depends on the extent to which those characteristics that make analogical reasoning a source of power in human thinking can be implemented in machine form. We argue that programs that learn by analogy have been relatively unsuccessful. Our diagnosis suggests that the problem arises from an inadequate understanding of the nature of analogy and of analogical reasoning. Our prescription for remedying the problem is to examine in some depth the concept of analogy from the perspective of both machine learning and psychology. This examination reveals fundamental differences in the definitions of analogy between the two fields. From a psychological perspective, these differences indicate that the implemented programs learn by a process more akin to generalization -- literal similarity -- than by analogy. More recent theoretical developments in the field of machine learning come much closer to what we understand by analogical processes in human cognition, but they have yet to be implemented in program form. We conclude, therefore, that the ideas which have been implemented do not involve learning by analogy, while the ideas that involve learning by analogy have not been implemented.

1. INTRODUCTION

Artificial intelligence (AI) is the study of how to make computers do things at which, at the moment, people are better (Rich, 1983). One respect in which people are markedly superior to computer programs is their capacity to learn. Most programs today perform a given task in the same way every time they are used. Performance of such programs can only be improved by the time-consuming, costly process of reprogramming. They cannot learn from experience, from past mistakes, or from observing the behaviour of others. Even minor changes in task can require major changes in programming, changes which must be made by programmers and not by the program itself. In short, most programs do not learn.

Machine learning, a relatively new area in AI, attempts to redress this deficiency. The objective is to develop programs which can improve their performance by learning. But learning can take many different forms: rote learning (memorization), learning from instruction, learning from examples, learning by discovery and observation, and learning by analogy, for example.

The focus of the present paper is on the last of these alternative ways of learning -- learning by analogy. For people, analogy plays an important role in creativity (Evans, 1968; Billow, 1977), in scientific discovery (Oppenheimer, 1956; Dreistadt, 1968; Hesse, 1970), in language and how we perceive the world we use language to talk about (Ortony, Reynolds, & Arter, 1978; Billow, 1977), and in common-sense reasoning (Carbonell & Minton, 1983). The ubiquity of metaphor in everyday language and

reasoning, for example, is easily demonstrated by counting the number of metaphors and analogies on a newspaper or magazine page. As analogy plays such a central role in human learning, it is important that machine programs tap the same source of power.

Until recently, however, machine learning confined its attention primarily to the simpler, more basic forms of learning such as rote learning (or memorization). Attesting to this is the fact that in 1982 the Handbook of Artificial Intelligence (Cohen & Feigenbaum, 1982, vol.3), while noting the area's potential importance, included no material on the topic, since "... this area has not received much attention" (p.334). This situation has changed markedly since 1980, however. Analogy has assumed a central role in more recent machine learning programs, even to the extent of being elevated to the status of one of nine "sources of power", or keys, to intelligent problem solving (Lenat, 1984).

Undoubtedly, a stimulus for this interest is the belief that analogical reasoning is an important component of human thought processes. Carbonell (1983), for example, states that "... analogy is one of the central inference methods in human cognition..." (p.137). For Lenat (1984), the ability to understand and reason by analogy is the "... source of power at the heart of human intelligence..." (p.209). Winston (1980) makes the more modest claim, that "Much thinking is done by analogy" (p.689).

The extent to which machines can capitalize on this source of power presumably depends on the extent to which those characteristics that make analogical reasoning a source of power in human thinking can be implemented in machine form. Do we know enough about what these characteristics are to achieve non-trivial

implementations of analogical reasoning? Assuming that we did indeed know enough to model analogical reasoning, at least in outline, there may remain problems associated with modelling the component processes. It is widely accepted that analogical reasoning involves the comparison or mapping of one domain to another. Do we know enough about the cognitive encoding of these domains to establish knowledge representations which can capture the essential characteristics of analogical reasoning? Similarly, do we know enough about the comparison or mapping processes involved?

The following paper attempts to address these questions from both a machine learning and a psychological perspective, and indicates some of the differences which appear to exist between these approaches to the understanding of analogy. The next section specifically addresses the issue of program performance. We shall argue that the machine learning programs that learn by analogy have been relatively unsuccessful. Our diagnosis suggests that the problem arises not so much from deficiencies in the size of knowledge base utilized -- as has been suggested by others such as Lenat (1984) -- but from an inadequate understanding of the very nature of analogy. Our prescription for remedying the situation, therefore, is to examine the concept of analogy in some depth. The third section of the paper addresses the issue of what exactly an analogy is. Succeeding sections consider the mapping processes in analogical reasoning, the representation of knowledge in analogical reasoning, and some general conclusions. Appendix A provides a more technical review of these and related issues in machine learning for the interested reader.

2. PROGRAM PERFORMANCE

Extravagent claims have sometimes been made for the power and accomplishments of AI programs, and the field of machine learning is no exception. Performances of learning-by-analogy programs simply do not match the claims made for them. We have often found the examples cited by the authors provide a more accurate reflection of a program's power and performance. With this caution in mind we will examine the programs created by four of the major researchers in the field -- Evans, McDermott, Carbonell, and Lenat.

Early work on programming analogical reasoning focused on the kind of artificial situation presented in creativity and intelligence tests (Evans, 1968). Evans' program was capable of recognizing analogies between geometric figures. Problems were of the form "A is to B as C is to ??", and five possible responses were provided in a multiple-choice format. The program could not be generalized to other, more realistic problem domains, however.

McDermott's (1979) program, ANA, was designed to learn how to do new tasks by "analogy" with similar known tasks. When given an unfamiliar task, the program searches for a highly similar task which it already knows how to do, and modifies the method slightly to accomplish the new task. To illustrate the operation of his program, McDermott described an example in which ANA, already knowing how to "paint table" was confronted with the new task of having to learn how to paint a blue chair red. Although ANA did learn to "paint chair" successfully, the example serves to illustrate several limitations of the program. First, the program

did not really learn to "paint chair" in any detailed sense. Rather, it learned to construct a new command, "paint chair", which had the effect of changing the colour of the item painted. Second, the program depends critically on the existence of a store of "almost adequate methods". If an almost identical method is not already known, then the program cannot learn the new task. Third, learning to "paint chair" when one already knows how to "paint table" does not seem much of an accomplishment.

Carbonell (1983) describes an application of his program MEA in which it proved that the product of two odd numbers is odd after being instructed how to prove that the product of two even numbers is even. This was accomplished by applying almost exactly the same method used to prove the first theorem to the second theorem. As for ANA, the two situations must be highly similar for the program to work.

Lenat (1977, 1983a, 1983b, 1984) has developed two learning programs, each of which incorporates learning by analogy as one of several learning techniques. AM was developed to "discover" concepts and conjectures in elementary mathematics. The program appeared to be quite successful. To quote Lenat (1977):

> "AM began with scanty knowledge of a hundred elementary concepts of finite set theory. Most of the obvious set-theoretic concepts and relationships were quickly found (e.g. de Morgan's laws, singletons)... Prime pairs, Diophantine equations, the unique factorization of numbers into primes, Goldbach's conjecture -- these were some of the nice discoveries." (p.839)

The success of AM appears, however, to have more to do with the fact that LISP, the programming language on which AM was based, possesses a mathematical structure that embodies much that was discovered (Lenat, 1983a; Ritchie & Hanna, 1984). Moreover, according to Ritchie and Hanna, "analogies are one of the less satisfying parts of AM" (p.262), for it is not at all clear how the program actually uses analogies in mathematical reasoning. As even Lenat (1983a) admits "AM's ultimate failure apparently was due to its inability to discover new, powerful, domain-specific heuristics for the various new fields it uncovered." (p.61).

EURISKO was designed to avoid some of the problems associated with AM. This program applies "analogical" reasoning in different task domains. For the purposes of exposition we confine our discussion to a task domain in which EURISKO is considered to have made a major contribution -- the design of naval fleets in the TCTS (Traveller Trillion Credit Squadron) wargame. EURISKO won both national (USA) championship tournaments (1981 and 1982) for which it designed a fleet of ships. Given that Lenat had never played the game before or seen a competition, the performance of the program appears impressive. However, closer examination of the tournament games suggest that the wins may be more easily attributable to the exploitation by EURISKO of the finer details of the game than to its analogical reasoning engine. In Lenat's (1983a, p.83) own words: "What EURISKO found were not fundamental rules for fleet and ship design; rather, it uncovered anomalies, fortuitous interactions among rules, unrealistic loopholes that hadn't been forseen by the designers of the TCS simulation system."

To summarize, learning-by-analogy programs have tackled problems of analogy in intelligence tests, naval wargames, household tasks such as painting chairs, computer chip design, and elementary number theory. It would be misleading to assert that these programs have accomplished little, for much has been learned. Nevertheless, compared to human learning-by-analogy, program performance to date has been rather poor.

Why is performance so poor? Researchers in the field, such as Lenat (1984), have advocated expansion of the knowledge base to improve program performance. In defence of this position, they have argued that people can access millions of situations, actions, objects, and concepts upon which analogies can be based. Programs, on the other hand, lack this rich store of information. Researchers, so the argument runs, have not only failed to provide programs with such large stores of "experiences" but have also failed to provide them with the capacity for storing large numbers of experiences as they occur. These programs may store experiences over short periods of time, but are typically restarted on new problems with most or all of their memories erased. Moreover, even such memories that are retained over time are impoverished relative to those of people.

While the size of knowledge base may, indeed, limit the effectiveness of learning-by-analogy programs, a more fundamental problem may underlie their poor performance. We shall argue that the real problem lies in the question of what constitutes an analogy. Most current programs perform poorly because they fail to capture the essential nature of analogy. Ortony et al. (1978, p.921) suggested essentially the same reason for the lack of

progress among philosophers on the nature of metaphor: "It is our contention that a prime reason for this [lack of progress] is the relative inexactness and inadequacy of the dominant philosophical theories and definitions of metaphor. A good definition is needed..." To explore this issue, we shall first compare the definitions and theories of analogy as used in machine-learning programs with those from psychology.

3. WHAT IS AN ANALOGY?

Within both the AI and psychological communities, it appears to be generally held that analogy, simile, and metaphor are closely related phenomena involving a comparison between objects (or their attributes, feature sets, predicates, structural properties, etc.) or a comparison between relations between objects (Carbonell & Minton, 1983; Miller, 1979). While in a strict sense, analogies are associated with a comparison of relations between objects, this distinction is not considered to be of theoretical significance to an understanding of the psychological processes involved (Ortony, 1979b). Consequently, we will use "analogy" in the broad sense in which similes and metaphors are both considered to express analogies (Miller, 1979).

In their simplest form, similarity statements of this kind involve two terms, traditionally known as the "topic" and "vehicle", though in AI parlance, "target" and "source" have more currency. In the analogy "the brain is like a computer", the first term is the topic or target, the second the vehicle or source.

It is widely recognized in the psychological literature that the relationship between source and target is asymmetrical. The transfer of information is generally from source to target, (although interactions may be involved. See, for example, Black, 1962). This means that reversing the positions of the terms may result in a loss or change of meaning. For example:

"The old lady fought like a prizefighter"

has a very different meaning from

"The prizefighter fought like an old lady".

In general, it appears that asserting that A is like B is not necessarily equivalent to asserting that B is like A. This type of asymmetry poses serious problems for some psychological theories of analogy. For example, Rumelhart & Abrahamson (1973) proposed a model of analogical reasoning in which the similarity between objects was a direct function of their psychological distance in multidimensional space. One difficulty for this type of geometrical approach is that the distance (A,B) equals the distance (B,A), which makes it difficult to account for the asymmetry of analogy statements, or indeed of similarity statements more generally.

Tversky's (1977) theory of similarity provided a partial resolution to the problem of asymmetry. In this formulation, if A represents the set of all features in the target, and B the set of all features in the source, then the similarity of source to target, S(a,b), is given by:

$$S(a,b) = \theta f(A \cap B) - \alpha f(A - B) - \beta f(B - A)$$

That is, the similarity of a to b is a weighted function of the number of common features minus weighted functions of the distinctive features of a and of b. It follows from this that $S(a,b) = S(b,a)$ if either $\alpha = \beta$, which implies that the task is non-directional (ie. in what ways are a and b alike, rather than in what ways is a like b?), or if $f(A-B) = f(B-A)$, which implies that the feature sets A and B are of equal size. Otherwise the similarity will be asymmetrical.

A case where Tversky's approach runs into difficulty occurs if the analogy creates the perceived overlap, rather than reflects

an existing overlap. (Black, 1962; Ortony, 1979b). To take an extreme example, the concept "zaglob" may have no features for English speakers, and consequently no shared features. It's similarity with other concepts will therefore be zero, using Tversky's measure. However, the simile "zaglobs are like giraffes" may provide information for English speakers, even in the absence of any a priori shared features between the two terms. What appears to occur is that salient characteristics of the source are attributed to the target, rather than selected from a pre-existing feature set. The similarity is created, rather than recognized, which Ortony (1979b) refers to as "attribute-introduction" as opposed to "attribute-promotion".

The relative salience or importance of both objects and object attributes is another concern which psychological approaches to analogy typically attempt to grapple with. With respect to objects, people show decided preferences when asked to complete similarity statements of the form "A __ is like a __". In general, they appear to prefer placing the "better exemplar", or "better pattern" or more "meaningful" term in the second, source, position. For example, Rosch (1975) found that when subjects were asked to place a pure focal red and a slightly "off" red stimulus in these relative positions, they exhibited strong preferences for placing the focal red in the second position. Similar results were obtained using numbers and line orientations as stimuli. In a pilot study at the University of Victoria, we obtained extreme examples of asymmetries of a similar kind. The stimuli were nonsense syllables, which varied in terms of high or low "meaningfulness" ratings. When presented with a high and low

pair, subjects exhibited a preference to place the more "meaningful" nonsense syllable in the source position, the less "meaningful" in the target position".

With respect to the salience of object attributes, similar directional characteristics are evident, in that the more salient characteristics of the source term serve to select or "promote" less salient attributes in the target term. To use the previous example of the prizefighter and the old lady, to say that "the old lady fought like a prizefighter" attributes to the old lady what are high-probability or highly salient prizefighter properties. When comparing the prizefighter's performance to that of an old lady, the opposite attributions occur. Ortony (1979a) reports data which support this point.

The fact that the source draws out or highlights what were previously non-salient or even non-existent properties of the target serves to illustrate another point: that analogies are not concerned with the similarities between two similar domains but rather with drawing out certain similarities between two otherwise dissimilar domains. To illustrate, "birds" and "heat engines" are not superficially alike. Birds have wings, feathers, beaks, eat worms, and so on; heat engines are typically metallic, have polished exterior surfaces, are large and heavy, consume fossil fuels, etc. The analogy that "birds are like heat engines" does not invite us to attempt to match up these salient characteristics, but rather draws attention to certain less salient characteristics of birds in such a manner as to expose or suggest unknown or unrealized properties. The analogy proposes that we use heat engines, and their thermodynamic properties, as

a model which applies to certain characteristics of birds. To the extent that the model is a "good" one, it will suggest hypotheses or explain facts about birds which were not part of our previous knowledge about birds: for example, that there must be a lower limit to the viable size of birds, that smaller birds will consume more calories per gram of body weight than large birds, that small birds will be less common in extreme latitudes, and so on. It is in this sense, of analogies as models, that practicing scientists as well as psychologists appear to understand the essential nature of analogical reasoning. Craik, for example, asserts that models are analogies, and that a model is "...any physical or chemical system which has a similar relation-structure to that of the process it imitates...The model need not resemble the real object pictorially...but it works in the same way in certain essential respects" (Craik, 1968, p.284). Oppenheimer proposes a similar interpretation, that by analogy he means "...a special kind of similarity, which is the similarity of structure, the similarity of form, a similarity of constellation between two sets of structures, two sets of particulars, that are manifestly very different but have structural parallels. It has to do with relation and interconnection" (Oppenheimer, 1956, p. 129).

In answer to the question of what is an anlogy, therefore, there are certain characteristics on which there is wide agreement in the psychological literature. Analogies are based on similarity, but not the similarity of appearance, or the similarity between highly salient properties of source and target. They are based on a similarity of structure, of "inner form" or process, which is more characteristic of, or more clearly

represented in the source than in the target. The source therefore acts as a model which "maps on" to the target. For this reason, the similarity relationship is directional, not symmetrical.

In contrast, on examining the descriptions of learning-by-analogy programs, we were struck (a) by the general lack of clarity in defining analogy, (b) by the discrepancy between stated definitions of analogy and what actually seems implemented in a program when analogy is defined, and (c) by the rather naive view of analogy implemented in many of these programs. Given the centrality of the notion of analogy to these programs, one might be forgiven for expecting the concept of analogy to be clearly defined by each author. All too frequently the term "analogy" has been used with little or no explication (McDermott, 1979; Cohen & Feigenbaum, 1983; Lenat, 1983a, 1984; Lenat & Brown, 1983: Carbonell, 1983; Winston, 1980).

Descriptions of the programs and illustrative examples provide some indication of the concept of analogy employed in a particular program. The prevalent view implemented in these programs is that analogy equals similarity (Winston, 1980; Lenat, 1984; McDermott, 1979). According to this view, the more similar two situations are, the more likely some form of profitable analogy might be found. For example, Winston (1980, p.693) asserts that "Analogy is based on the assumption that if two situations are similar in some respects, then they must be similar in other respects as well." From his description of the "matcher" component of his program, it seems that he counts the number of shared relations in common and makes an analogy on the basis of the

permutation having the greatest score. For Lenat (1984, p.213) "it is meaningful between two concepts only if they share many of the same attribute names, and it is useful or cost-effective if in addition the concepts are actually similar in some of their qualities, that is, if certain of their attribute values are comparable." Carbonell (1982, 1983) hypothesizes that analogical problem solving is a four-step process with the first step being the recall of one or more past problems that bear strong similarity to the new problem: "When encountering a new problem situation, a person is reminded of past situations that bear strong similarity to the present problem."

More recent theories of analysis proposed by AI researchers (e.g., Gentner, 1983; Carbonell & Minton, 1983) have been much closer to the psychological theories discussed earlier. For example, Gentner (1983) has proposed that the domain of a concept should be represented as a system of objects, object attributes, and relations between objects. This view is shared by Gick & Holyoak (1980, 1983) and resembles the representation suggested by Carbonell (1982, 1983).

Gentner, in furthering his proposal, suggests that it is the relational characteristics that play an important role in an analogy. For example, an electric battery is like a reservior because they both represent a source of energy being held by a container. He also attempts to distinguish the differences between an analogy, a literal similarity, and an abstraction. An analogy, as defined by Gentner, is a comparison in which most relational characteristics are shared between the concepts, but few or no object attributes are matched. A literal similarity statement

differs from an analogy in that it has more of the object attributes matched. Hence, the statement "The X12 star system in the Andromeda galaxy is like our solar system" is not an analogy because the X12 star and our sun have a lot of object attributes in common. An abstraction is distinguished from an analogy by the fact that one of the concepts must be an abstract relational structure with generalized physical entities and that all objects and relational features should be matched. For example, the statement "The hydrogen atom is a central force system" is an abstraction. Even though these distinctions may be semantically correct, they are rarely regarded in the modelling of analogical reasoning. This departure in defining analogy can be seen in an example provided by Carbonell (1983). The example involves a person planning to travel from Pittsburgh to New York city. This person usually travelled by planes but had discovered that all the flights were booked. He had never travelled using the intercity train, but by analogical problem solving, he could reason that he needed to withdraw sufficient money from the bank to buy a ticket, find out where to buy the ticket, call the ticket office to make a reservation, and later go to the station to board the train. In this example, the target problem and the analog have a lot of object attributes in common - the person, Pittsburgh, New York city, and the bank are identical in both situations; the ticket offices could be expected to have many shared physical features; even the train and the airplane are both made of metal and supported by wheels. Applying Gentner's definitions, this example may be interpreted as a case of literal similarity rather than an analogy.

Carbonell & Minton (1983) also deviate from the simplistic view of analogy as similarity. Although analogy also seems equated with similarity for them ("Analogical reasoning is the process by which one recognizes that a new situation is similar to some previously encountered situation ..."), they go further in exploring the idea that analogy is based on structural similarity, that the role of analogy is to transfer information from familiar situations to unfamiliar situations, and that only the salient features of the source are transferred to the target.

Thus, both Gentner's (1983) and Carbonell & Minton's (1983) theories of analogy reflect many of the considerations raised by the psychological literature. Neither researcher, however, has yet developed a program implementing these new ideas. (See Appendix A, pages 12-18 for additional technical information.)

Given some agreement between the more recent learning theories (which remain to be implemented) and the psychological theories, the next section addresses the issue of how characteristics are mapped from the source to the target.

4. MAPPING PROCESSES IN ANALOGICAL REASONING

Carbonell (1982, 1983), Gentner (1983), and Gick and Holyoak (1980, 1983) have all proposed that the recognition of an analogy occurs through a mapping process. There is considerable agreement amongst their theories and the main points of similarity are listed below:

(1) The mapping process involves comparing features of two domains.

(2) If one says that a T is like a B, the mapping is from B to T. B will be called the base or source, and T the target.

(3) The object features are the least important ones in the mapping.

(4) The relational features (Gentner, 1983) or structural similarities (Carbonell, 1983) are the important characteristics.

(4) Learning can be facilitated by transferring features from the source domain to the target domain.

(6) In order to have learning facilitated, the source must be a well-understood domain and the target a comparatively ill-structured one.

Gentner appears to hold the view that the strength of an analogy is dependent on the degree of overlap of the relational features. However, there is no clear indication of the amount of overlap required to distinguish an analogy which "works" from one which fails.

Carbonell (1983) provides a more detailed plan for assessing

the similarities in an analogy. His theory is a modification of the traditional means-ends analysis of Newell & Simon (1972). Means-ends analysis (MEA) identifies the domain of a problem as a problem space which consists of:

(1) A set of possible problem states.

(2) One state designed as the initial state.

(3) One or more states (if there is more than one solution) designated as goal states.

(4) A set of operators with known preconditions that transform one state into another in the space.

(5) A difference function that computes differences between two states.

(6) A method for indexing operators as a function of the difference between the current and the goal states they can reduce.

(7) A set of global path constraints that must be satisfied in order for a solution to be viable.

Carbonell's modification involves comparing the following features between the source and the target domains:

(1) the initial states,

(2) the final states,

(3) the path constraints,

(4) proportions of preconditions of the transferred operators satisfied in the two domains.

Carbonell refers to this as a comparison of structural similarity. It is obvious that the elements being compared are also relational features, using Gentner's terminology. Since an analog is not identical to its target, features of the analog,

when mapped onto the target, may be varied in order to satisfy the current constraints. In a study of two hundred metaphors, Carbonell (1982) discovered that certain kinds of features are less likely to vary in the mapping process. Based on this finding, Carbonell proposes that a "hierarchy of relative invariance" can be created by listing the features according to its tendency to remain unchanged in a mapping process. This suggests that the strength of an analogy can be assessed by measuring the invariance at each level of the hierarchy, presumably with the higher level features being given more weight. The following is a hierarchy proposed by Carbonell:

(1) Goal expectation - if the analogy implies a task to be done, the goal of the actor is usually preserved in the mapping (e.g. "inflation is like a disease" - the goal is to cure and be healthy).

(2) Planning and counterplanning strategies - the means for achieving the goal (e.g. controlling the growth of virus to cure disease implies controlling money supply to hamper inflation).

(3) Causal structures - the cause and effect relationships (e.g. medicine cures disease; hence economic measures will hamper inflation).

(4) Functional attributes - the function of objects involved (e.g. a doctor to administer medicine; hence the finance ministry to plan economic policies).

(5) Temporal orderings - the order of events to occur.

(6) Natural tendencies - the natural laws governing the behaviour of the objects.

(7) Social roles - the social relations between the actors.

(8) Structural relations - the physical relations between objects.

(9) Descriptive properties - physical attributes of the objects (i.e. object features).

(10) Object identity - this is very rarely mapped in an analogy.

The proposals made by Carbonell, Gentner, and Gick and Holyoak all represent a means for assessing the similarity between the source and target domains. However, these are only applicable when both the target problem and the potential analog are given. The problem remains of how can a potential analog be retrieved from the memory. It would be impractical to search through all events in the memory to find an analogy. Sternberg (1977) suggests that an analogy can be identified by asking "if A is like B, what is C like". The problem of this method is that it required a parallel analogy to be created first. It would be as difficult to find an applicable parallel as to find an analogy for the target. Besides, this approach would limit the kind of analog available to be retrieved (Gick & Holyoak, 1983). Carbonell (1983) proposes a practical method for narrowing the search paths by generalizing solution plans that bear strong similarities. This suggestion is consistent with Kintsch and Van Dijk's (1978) theory of prose representation and Gick and Holyoak (1980, 1983)'s notion of a schema. Gick and Holyoak used Duncker's (1945) "radiation problem" to obtain empirical proof that generalization procedures are actually adopted in human reasoning. Duncker's problem involves a doctor faced with a patient with a malignant tumor. It is

impossible to operate on the patient, but the patient will die if the tumor were not destroyed. There is a kind of ray that can destroy the tumor, but the rays with intensity high enough to kill the tumor will destroy the healthy tissue at the same time. The question is how can the tumor be destroyed without hurting the tissue. In the experiments, Gick and Holyoak provided subjects with military stories about how a fortress can be captured. Their findings suggest that subjects generalize the radiation problem and the military stories to form a convergence schema similar to the one presented below:

- Initial state

  Goal: Use force to overcome a central target.

  Resources: Sufficiently great force.

  Constraint: Unable to apply full force along one path.

- Solution plan: Apply weak forces along multiple paths simultaneously.

- Outcome: Central target overcome by force.

Kintsch and Van Dijk's theory involves the application of a set of inference rules to generate an abstract macrostructural representation of a problem. This process would produce an output similar to Gick and Holyoak's schema. Gick and Holyoak suggest that the schema would have a perfect ovarlap with the target domain since only the matched features are being generalized. This implies not only a saving on memory space and search time but also an improvement of decision quality, if only the schema are stored.

Carbonell's generalization procedures are more complicated than Gick and Holyoak's schema and Kintsch and Dijk's prose

representation. He suggests that all solution plans be fed into an inductive machine (Dietterich & Michalski, 1983; Michalski, 1983) which would cluster similar plans together to form generalized plans that resemble Schank's notion of a script (Schank & Abelson, 1977; Schank, 1980). More specifically, Carbonell's suggestion involves the following procedures:

(1) When an analogical plan is created for a new problem, the plan has to be tested in the external enviroment.

(2) Feedback should be obtained to indicate whether the plan is sucessful.

(3) The successful and unsuccessful plans are fed into a inductive machine which generates a plan encompassing all the successful solutions and none of the unsuccessful ones.

(4) A comparison between the successful plans and the unsuccessful ones to identify features that would discriminate the plans.

(5) If the machine fails to generate a solution from an analogy, it indicates that the difference function (for measuring the differences between states) must have omitted some crucial aspects of the analogy and the function criterion should be refined.

Since Carbonell's model involves generalizing a class of problem solutions to form a plan, the generalized plan should represent a higher level of abstraction than Gick and Holyoak's schema or Kintsch and Van Dijk's prose representation. Consequently, more interpretation would be required in the mapping process. Carbonell's model, on the other hand, should have the

advantage of further reducing search time. It is unclear whether these generalized plans can be categorized into a hierarchical structure to facilitate a faster search process. In the case of a very complicated problem for which an analogy cannot be found, Carbonell suggests that the problem may be decomposable into sub-problems for some of which analogies may be obtainable. This suggestion introduces flexibility into the learning system and allows it to handle a large variety of tasks.

There is another advantage and disadvantage to the system proposed by Carbonell. The advantage is that the system is an adaptive one which allows it to broaden its applicability and improve its decision quality as more problems are solved. The disadvantage is that the model requires human assistance to provide feedback for the system. One may argue, however, that interaction with an external environment is a necessary condition for the acquisition of knowledge, and need not be regarded as a weakness.

Most of the research on analogical learning (Gentner, 1983; Gick and Holyoak, 1980, 1983) has been psychological in its focus. Consequently, the models created in these studies may be expressed in abstract terms that are difficult to translate into algorithms suitable for machine processing. While there are a number of models expressed in formal logic (Kling, 1971; Stelzer, 1983), these models solve mainly problems in mathematics, the domain of which is well defined. One model that does handle ill-defined domains can be found in the framework proposed by Carbonell (1983). This framework provides a detailed description of the steps required to transfer operators from the analog to the

target. As mentioned earlier, a generalized plan for a class of problems can be developed by an inductive machine. The generalized plan should consist of a sequence of states and operators. When presented with a new problem, the problem solver would try to search for a plan that begins with the same initial state and ends with a state closer to the desired goal. The ending state of the plan would then become the current state in the new problem space. The search process would begin again to find a plan that would further narrow the gap between the current and the goal states. This process would continue until the goal state is attained. By this method, a new problem space would be created with a number of retrieved plans connected together. Carbonell refers to this space as the analogy transform problem space (T-space). The retrieved plans would become the states in this space.

Carbonell's approach may appear to resemble the notion of a macro-operator (Korf, 1985; Fikes & Nilsson, 1971) but there are distinct differences between them. A macro-operator, like a solution plan, is represented by a sequence of states and operators. One distinct characteristic of a macro-operator is that it allows certain (non-serializable) subgoals of the problem to be temporally violated in its application. This characteristic allows macro-operators to become extremely useful in solving problems like Rubik's Cube. Macro-operators, however, are not applicable in Carbonell's model for several reasons. First, macro-operators perform very specific tasks. A slight variation in the problem situation would require new macro-operators to be formed. In order to solve problems that have many variations, the system would have to store a large number of macro-operators. The combinatorics

involved in storing and searching all these operators could easily become unmanageable. Second, the application of macro-operators does not consider path constraints. A macro-operator would easily become invalid since a new problem usually carries a different set of path constraints. Third, there is no provision for adding, deleting, or substituting operators in a macro-operator. These operations are crucial elements in Carbonell's model.

Carbonell allows a set of operators for shaping the retrieved plans into potential solution sequences. To avoid confusion, these operators are referred to as T-operators. The following are some of the functions these T-operators can perform:

(1) Insertion of a new operator into a sequence.

(2) Deletion of an operator from a sequence.

(3) Subsititioon of an original operator by another operator or a sequence of operators.

(4) Concatenation of one solution sequence to another.

(5) Merging of two sequences to form a new sequence.

(6) Reordering of operators in a sequence.

(7) Substitution of an object in the original problem with an object in the new problem.

(8) Truncation of a sequence of operators from the original sequence.

(9) Inversion of the order of the operators in a sequence.

Carbonell suggests that T-operators may be indexed in a difference table. Entries in this table would take the form "To reduce X, apply a member of T-operator set Y". Since these T-operators do not involve performance in the external world, they are not subjected to the restriction of any path constraints.

Carbonell suggests to incorporate the path constraints in the difference table. He proposes that the comparison of the initial states, final states, path constraints in the two domains, applicability of the retrieved solution in the new problem scenario should individually be represented by a difference function. These four functions should be combined to form a difference matrix to represent the differences between the retrieved solution and the desired solution. A viable solution is said to have been found if all these functions indicate a zero difference. (The reader is referred to pages 12-18 in Appendix A for additional technical information relevant to this section.)

5. KNOWLEDGE REPRESENTATION IN ANALOGICAL REASONING

To the extent that analogical reasoning is based on a recognition of structural similarity, then machine implementations capable of this type of reasoning must have (a) knowledge representation which embodies the relevant structural characteristics, and (b) operators which are capable of extracting them. Are some forms of knowledge representation likely to be more useful in this respect than others?

There has been considerable debate among cognitive psychologists in recent years concerning knowledge representation in humans, and two points of view have emerged, the "propositional" and the "analog". The propositional view holds that knowledge is encoded in an essentially language-like medium in which both objects and relations between objects are represented symbolically. The "analog" view holds that representations are in some sense "spatial", in which both objects and their relationships are intrinsic to the representation. That is, the "image" represents the objects in relationships which are functionally equivalent to the relationships which they hold in the external world.

One proposed difference between these approaches is that of extrinsic versus intrinsic representation of relations. In the propositional approach, relationships are "extrinsic" in the sense that they are "added-on" as relational elements to the set of object elements. In analog representations, the relationships are given in the same representation as the objects, and are therefore

considered to be "intrinsic" (Palmer, 1978). However, it is quite possible to imagine propositional representations in which non-represented relationships can nevertheless be derived. For example, "the cat sat on the stove and the dog lay on the rug" could be represented propositionally by six elements, four object elements and two relational elements. An additional relation, that the cat is probably higher than the dog, could be derived from knowledge of the given elements. The non-represented relationship is therefore intrinsically available in the knowledge representation. The real difference may have more to do with the fact that in the propositional representation, the additional relationship must be inferred, whereas in an analog representation, this knowledge is held to be given immediately in the representation, rather than extracted by deduction.

A related difference between the two approaches has to do with the types of constraint which exist within the representational medium itself (Shepard, 1981). Propositional media are highly unconstrained, in the sense that, in principle, any object may be related to any other object by any available relation. The medium itself imposes no constraints on the combinatoric possibilities. On the other hand, analog media themselves possess a structure which constrains the forms of representation possible. As a low-level example, the perception of smell involves the "fit" between specific molecululars structures and specific receptors of complementary structure. At the level of visual perception, this point of view implies that the representational medium itself contains structure which is isomorphic to structure in the world it represents. For example,

it may contain an "up" and "down" or a functionally equivalent dimension. The representation of objects in this dimensional medium would consequently preserve their dimensional relationships simply as "part of the picture".

These points of view are not mutually exclusive, and it is quite possible for a cognitive system to be endowed with both analogical and propositional forms of representation, in which the first system forms images, and the second system describes them (Attneave, 1981). However, it is probably no accident that proponents of the propositional view typically try to model cognitive events using a highly unstructured medium (computer models) while proponents of the analog view typically employ physical or physicalistic types of model (Palmer, 1978). As a consequence, the field of AI has adopted an almost exclusively propositional approach to knowledge representation. It is possible, however, that the form of analogical reasoning that is "powerful" in humans derives more from an analog form of representation. If this is the case, then it will be necessary to build structure into knowledge representations. The remainder of the paper considers several extended examples of how this might be achieved in simple cases.

The first example begins with a consideration of Gibson's (1966) view of visual perception. Gibson proposed that perception is "direct". That is, the perceptual system does not "model" or "structure" the external world, it simply "picks-up" information which is already structured. He considered that this structure was preserved in the ambient light array. However, since the perceiver is typically in motion, this information undergoes

continual transformation as the visual "station-point" changes. An important component of Gibson's theory was the view that what was extracted by the perceptual system were those stimulus characteristics which remained invariant as the stimulus array changed over time, which Gibson referred to as "higher-order invariants". (The idea recalls Carbonnell's notion of a "hierarchy of relative invariance"). Consider, for example, a perceiver walking down a long corridor with doorways on each side. The projection of the optical array on a two-dimensional plane will undergo continuous transformations in which the far end wall gradually swells in size. Distant side doorways first appear as slits which gradually resolve into trapezoidal shapes, and so on. Out of this visual flux, the perceiver picks-up certain invariant relationships which afford the perception of a stable corridor with rectangular doorways at each side.

For present purposes, an interesting consequence of this point of view is that one can consider that the "structure" of the space itself undergoes transformation, in that any object placed in the corridor will undergo analagous transformation. Thus, although the specific changes which occur will depend on the objects themselves, any object placed in the same space will undergo the same general transformation.

One method by which changes in the "structure" of space may be represented was introduced by D'Arcy Thompson (1961) in a different context. He attempted to demonstrate relationships between different species by illustrating that the form of a body part, or in some cases the whole body, when replotted on a transformed co-ordinate system, yielded the form of another

species. Figure 1 shows one illustration of the technique, where the second form is a point-for-point mapping of the first form onto the transformed co-ordinate system.

This general technique has been employed by Pittenger, Shaw and Mark (1979), in an attempt to explain the fact that we can generally recognize people's faces despite changes caused by the process of aging. They showed that at least one change related to aging is a continuous topological transformation of head shape which can be represented by changes in a co-ordinate system using Thompson's method. Figure 2 illustrates the application of this transformation to cartoon drawing of the heads of three species. People judge the ages of the animals in order from top to bottom, the top being younger. Given the consistency of these judgments, we would expect people to be able to solve proportional analogies using this type of stimulus, where the relationship that has to be extracted has to do with the topological transformation which has been applied: for example, the top bird is to the bottom bird as the top dog is to the bottom dog. It is difficult to see how existing AI programs based on purely propositional forms of representation could extract this relationship.

The second example we wish briefly to explore begins with a method of solving proportional analogies proposed by Klein (1982), and based on a proportional form of stimulus representation[1].

[1]We are indebted to W. Treurniet for introducing us to this example, and for his participation in developing it.

The method applies to stimuli which can be described in terms of sets of binary features. Figure 3a shows an example of a feature set of this kind, with the values on each feature represented by 0 or 1. Form A in Figure 3b can therefore be represented as 1000, form B as 1110, and form C as 0001. Klein's algorithm solves for D in the proportional analogy A is to B as C is to D. It does so by comparing A and B and extracting an operator based on the logical operation of exclusive disjunction, which defines the transformation required to change A to B. This operator is then applied to C to solve for D, as illustrated below.

A	B	Operator	C	Operator	D
1	1	1	0	1	0
0	1	0	0	0	1
0	1	0	0	0	1
0	0	1	1	1	1

The procedure appears to work for stimulus sets of this kind, but fails when a spatial component is entered into the analogy. For example, consider the proportional analogy below, in which the value 1 represents a filled cell, the value 0 an empty cell in a 3x3 matrix.

A				B				C				D
1	0	0		0	0	1		1	1	1		
1	0	0	is to	0	0	1	as	0	0	0	is to	?
1	0	0		0	0	1		0	0	0		

Reading the values across the rows and applying Klein's procedure yields:

A	B	Operator	C	Operator	D
1	0	0	1	0	1
0	0	1	1	1	1
0	1	0	1	0	0
1	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	0	1
1	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	0	1

This produces the apparently anomalous solution:

	D	
0	1	0
1	0	1
1	0	1

If, however, C is encoded by reading off the values vertically rather than horizontally and if, after applying the operator, D is decoded in a similar manner, then we obtain the intuitively more acceptable solution:

D

0 0 0

0 0 0

1 1 1

The difficulty arises from the fact that the encoding of the information fails to preserve the structure necessary for the solution, that is the distinction between the vertical and horizontal dimensions. Only by providing the necessary rotation through 90° -- by changing the order of reading the data from rows to columns -- is the procedure able to find the solution. The example serves to illustrate once again the distinction between propositional and analog forms of representation.

However, by incorporating directional information into the coding scheme, it is possible to adapt Klein's procedure so that it can successfully solve spatial analogies of this type. Consider, for example, the analogy shown in Figure 4. People tend to produce either of two possible solutions to this problem, illustrated here as D1 or D2. D1 is obtained in the following way. A is transformed into B by mapping point 1 in A to point 1 in B and rotating the figure through 90° in the plane. The analogous transformation is then applied to C, to give D1. D2 is obtained by mapping point 5 in A to point 1 in B and performing a rotation through the third dimension. D2 is then obtained by performing the analagous operation to C.

Coding the line segments to produce solutions of this kind can be achieved by arbitrarily assigning binary values to the cardinal points of the compass, such that N = 00, E = 01, S = 10,

and W = 11.

Encoding the line segments in A using this scheme gives, reading from point 1, A = 01, 00, 01, 10, 01. Similarily, B = 10, 01, 10, 11, 10. Mapping A1 to B1, we obtain the transformation table given below:

Value in A	Value in B	Operator
00	01	10
01	10	00
10	11	10
11	00	00

Encoding C by reading from point 1, and applying the same transformational operator gives D1, as illustrated below:

C	Operator	D1
00	10	01
01	00	10
10	10	11
01	00	10
01	00	10

Decoding D1 gives the form D1 shown in Figure 4. The alternative solution is obtained by following the same procedure, but mapping A5 to B1. To do so, A is read from A5, providing the code A = 11, 00, 11, 10, 11. The code for B remains unchanged, resulting in a new operator, shown below:

Value in A	Value in B	Operator
00	01	10
01	00	10
10	11	10
11	10	10

Similarily, C is read from C5 and the operator applied:

C	Operator	D2
11	10	10
11	10	10
00	10	01
11	10	10
10	10	11

This provides the form shown as D2 in Figure 4.

The present section began with a discussion of propositional versus analog forms of knowledge representation, and proposed that these need not be regarded as mutually exclusive: a cognitive system may be endowed with both forms. However, it was suggested that present AI applications tend to focus almost exclusively on propositional forms. It was proposed that analog forms of representation might be better suited to certain forms of analogical reasoning, particularly those which embody structural components of a spatial form. Several examples were proposed of

how coding and mapping schemes might be developed which have a spatial component.

6. CONCLUSION

The paper focussed on machine learning by analogy as a potentially powerful technique for creating artificial intelligence systems capable of learning from experience. A review of the psychological literature indicated that analogical and metaphorical processes in human cognition are understood to involve:

(1) a mapping of "elements" from a source to target domain, where

(2) the "elements" are typically relational or structural rather than simply featural, and where

(3) the initial overall similarity between the two domains is low: that is, the domains are highly dissimilar, except for the analogical relationships.

A review of analogical learning programs which have been implemented indicated that analogies, as operationalized, involve:

(1) a mapping of elements from a source to target domain, where

(2) the elements are typically featural, rather than relational or structural, and where

(3) the initial overall similarity between the two domains is high, and indeed must be high for the analogy to be recognized.

From a psychological perspective, these differences in definition indicate that the implemented programs learn by a process more akin to generalization -- literal similarity-- than

by analogy as understood in human cognition. This in itself may be a useful and worthwhile development, but it is unlikely to tap what the program developers seem to mean by the "power"of human analogical reasoning. In the human sense, the power and economy of a good analogy can to some extent be gauged by the degree to which it surprises us, and evokes insight. The programs implemented to date show little potential for doing either.

However, more recent developments in cognitive science approaches indicate an awareness of a need to go beyond literal similarity. Gentner (1983) and Carbonnell (1983) emphasize relational and structural characteristics, and relegate featural similarity to the least important role in analogy recognition. Carbonnell's model allows features to be relaxed during the mapping process, which allows for the possibility that invariants may be recognized in mapping two otherwise dissimilar domains. The idea of a hierarchy of invariance further allows for the possibility that a small set of overlapping properties, provided that they are sufficiently high in the hierarchy, could trigger the recognition of an anlogy between highly dissimilar domains. These developments seem highly promising, and come much closer to what we understand by analogical processes in human cognition. However, they do not appear to have been implemented as operational programs.

The distinction between literal and non-literal similarity represents one area in which machine learning has not yet grasped the full complexity of human learning by analogy. A second important distinction is between propositional and analogical forms of knowledge representation. Existing programs rely

exclusively on propositional representation. However, it may be that human analogical reasoning, especially where spatial components are involved, is better represented analogically. Conjecturally, certain types of scientific and mathematical problem solving may depend on just such forms of spatial reasoning, and it may be that if the full power of human analogical reasoning is to be tapped, then analogical as well as propositional forms of knowledge representation will have to be explored.

7. REFERENCES

Attneave, F. Perceptual organization: Comments on views of Hochberg, Shepard, and Shaw and Turvey. In M.Kubovy & J.R.Pomeranz (Eds.), Perceptual organization. Hillsdale, NJ: Erlbaum, 1981, 417-421.

Billow, R.M. Metaphor: A review of the psychological literature. Psychological Bulletin, 1977, 84, 81-92.

Black, M. Metaphor. In M.Black, Models and metaphors. Ithaca, NY: Cornell University Press, 1962.

Boden, M. Artificial intelligence and natural man. New York: Basic Books, 1977.

Carbonell, J.G. Metaphor: an inescapable phenomenon in natural language comprehension. In N. Lehnert and M. Ringle (Eds.), Knowledge representation for language processing systems. Hillside, NJ: Lawrence Erlbaum, 1982.

Carbonell, J.G. Learning by analogy: formulating and generalizing plans from past experience. In R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning. Palo Alto, Calif: Tioga, 1983, 137-161.

Carbonell, J. & Minton, S. Metaphor and common-sense reasoning. Report CMU-CS-83-110 for the Office of Naval Research, Contract number N00014-79-C-0661, 1983.

Cohen, P.R. & Feigenbaum, E.A. (Eds.) The handbook of artificial intelligence. Stanford, CA: HeurisTech Press, 1982.

Craik, K.J.W. Hypothesis on the nature of thought. In P.C.Wason & P.N.Johnson-Laird (Eds.), Thinking and reasoning. Baltimore, MD: Penguin, 1968, 283-290.

Dietterich, T.G. & Michalski, R.S. A comparative review of selected methods for learning from examples. In R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning. Palo Alto, CA: Tioga, 1983.

Dreistadt, R. An analysis of the use of analogies and metaphors in science. Journal of Psychology, 1968, 68, 97-116.

Duncker, K. On problem Solving. Psychological Monographs, 1945, 58.

Evans, T.G. A program for the solution of geometric-analogy intelligence test questions. In M.Minsky (Ed.), Semantic Information Processing. Cambridge, MA: MIT Press, 1968, 271-353.

Fikes, R.E. & Nilsson, N.J. Strips: a new approach to the

application of theorem proving to problem solving. Artificial Intelligence, 1971, 2, 189-208.

Genter, D. Structure-mapping: a theoretical framework for analogy. Cognitive Science, 1983, 7, 155-170.

Gibson, J.J. The senses considered as perceptual systems. Boston, MA: Houghton Mifflin, 1966.

Gick, M.L. & Holyoak, K.J. Analogical problem solving. Cognitive Psychology, 1980, 12, 306-355.

Gick, M.L. & Holyoak, K.J. Schema induction and analogical transfer. Cognitive Psychology, 1983, 15, 1-38.

Hesse, M. Models and analogies in science. Notre Dame, IN: University of Indiana Press, 1970.

Kintsch, W. & van Dijk, T.A. Toward a model of text composition and production. Psychological Review, 1978, 85, 363-394.

Klein, S. Culture, mysticism & social structure and the calculation of behavior. Proceedings of the 1982 European Conference on Artificial Intelligence, Orsay, France, 1982, 141-146.

Kling, R.E. A paradigm for reasoning by analogy. Artificial Intelligence, 1971, 4, 147-178.

Korf, R.E. Toward a model of representation changes. Artificial Intelligence, 1980, 14, 41-78.

Korf, R.E. Marco-operators: A weak method for learning. Artificial Intelligence, 1985, 26, 35-77.

Lenat, D.B. Computer software for intelligent systems. Scientific American, 1984, 251, 204-213.

Lenat, D.B. Automated theory formation in mathematics. Proceedings of the Fifth International Joint Conference on Artificial Intelligence, Cambridge, MA, 1977, 833-842.

Lenat, D.B. EURISKO: A program that learns new heuristics and domain concepts. Artificial Intelligence, 1983a, 21, 61-98.

Lenat, D.B. The role of heuristics in learning by discovery. In R.S.Michalski, J.G.Carbonell & T.M.Mitchell (Eds.), Machine learning. Palo Alto, CA: Tioga, 1983b, 243-306.

Lenat, D.B. & Brown, J.S. Why AM and EURISKO appear to work. Artificial Intelligence, 1984, 23, 269-294.

McDermott, J. Learning to use analogies. Report for the Defence Advanced Research Projects Agency (DOD), ARPA Order No. 3597, Contract No. F33615-78-C-1151, 1979, pp. 568-576.

Michalski, R.S. A theory and methodology of inductive learning. In R.S. Michalski, J.G. Carbonell, T.M. Mitchell (Eds.), Machine Learning. Palo, Alto, CA: Tioga, 1983.

Miller, G.A. Images and models, similes and metaphors. In A.Ortony (Ed.), Metaphor and thought. New York: Cambridge University Press, 1979, 202-250.

Newell, A. & Simon, H.A. Human problem solving. New Jersey: Prentice-Hall, 1972.

Oppenheimer, R. Analogy in science. American Psychologist, 1956, 11, 127-135.

Ortony, A. Beyond literal similarity. Psychological Review, 1979a, 86, 161-180.

Ortony, A. (Ed.) Metaphor and thought. New York: Cambridge University Press, 1979b.

Ortony, A., Reynolds, R.E., & Arter, J.A. Metaphor: Theoretical and empirical research. Psychological Bulletin, 1978, 85, 919-943.

Palmer, S.E. Fundamental aspects of cognitive representation. In E.Rosch & B.B.Lloyd (Eds.), Cognition and categorization. New York: Erlbaum, 1978, 259-303.

Pittinger, J.B., Shaw, R.E., & Mark, L.S. Perceptual information for the age level of faces as a higher order invariant of growth. Journal of Experimental Psychology: Human Perception and Performance, 1979, 5, 478-493.

Rich, E. Artificial intelligence. New York: McGraw-Hill, 1983.

Richie, G.D. & Hanna, F.K. AM: A case study in AI methodology. Artificial Intelligence, 1984, 23, 249-268.

Rosch, E. Cognitive reference points. Cognitive Psychology, 1975, 7, 532-547.

Rumelhart, D.E. & Abrahamson, A.A. A model for analogical reasoning. Cognitive Psychology, 1973, 5, 1-28.

Schank, R.C. & Abelson, R.P. Scripts, goals, plans and understanding. Hillside, NJ: Lawrence Erlbaum, 1977.

Schank, R.C. Language and memory. Cognitive Science, 1980, 4, 243-284.

Shepard, R.N. Psychophysical complementarity. In M.Kubovy & J.R.Pomeranz (Eds.), Perceptual organization. Hillsdale, NJ: Erlbaum, 1981, 279-341.

Stelzer, J. Analogy and axiomatics. International Journal of Man-Machine Studies, 1983, 18, 161-174.

Sternberg, R.J. Intelligence, information processing, and analogical reasoning: The componential analysis of human abilities. Hillside, NJ: Lawrence Erlbaum, 1977.

Thompson, D'Arcy On growth and form. New York: Cambridge University Press, 1961.

Tversky, A. Features of similarity. Psychological Review, 1977, 84, 327-352.

Winston, P. Learning and reasoning by analogy. Communications of the ACM, 1980, 23, 689-703.

Figure 1: Analagous forms produced by topological transformation of the co-ordinate system (from Thompson, 1961).

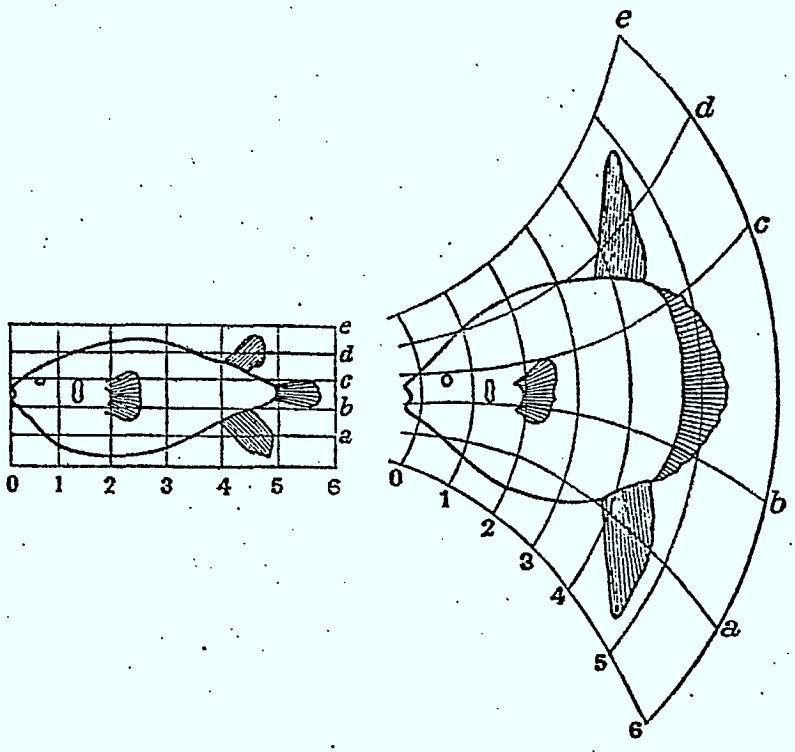


Figure 2: Analagous forms of cartoon profiles produced by topological transformation of the co-ordinate system (from Pittenger, Shaw & Mark, 1979).

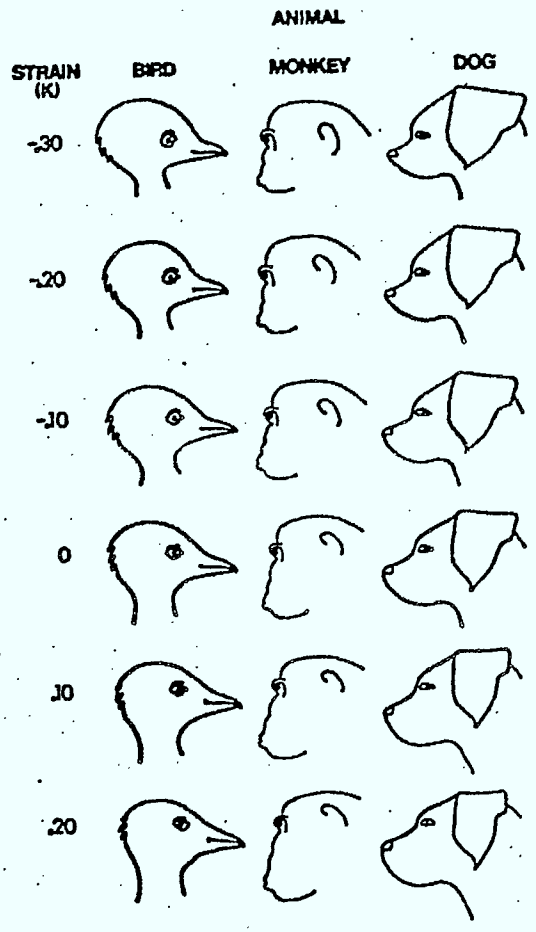


Figure 3a: Feature set and coding

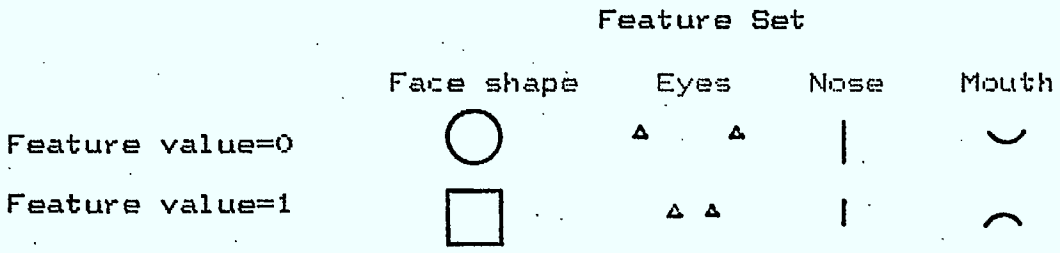


Figure 3b: Proportional analogy with face stimuli

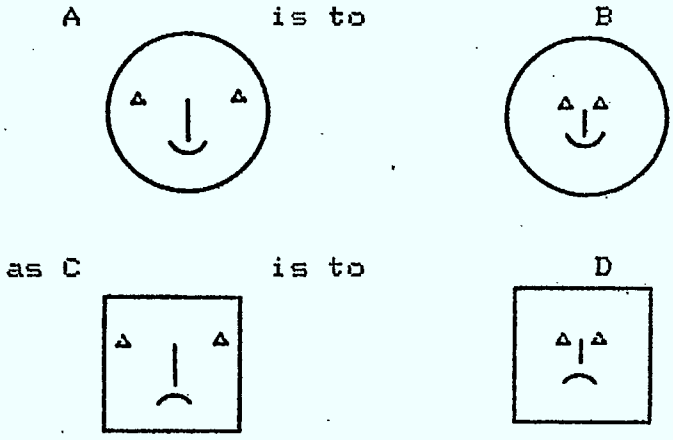
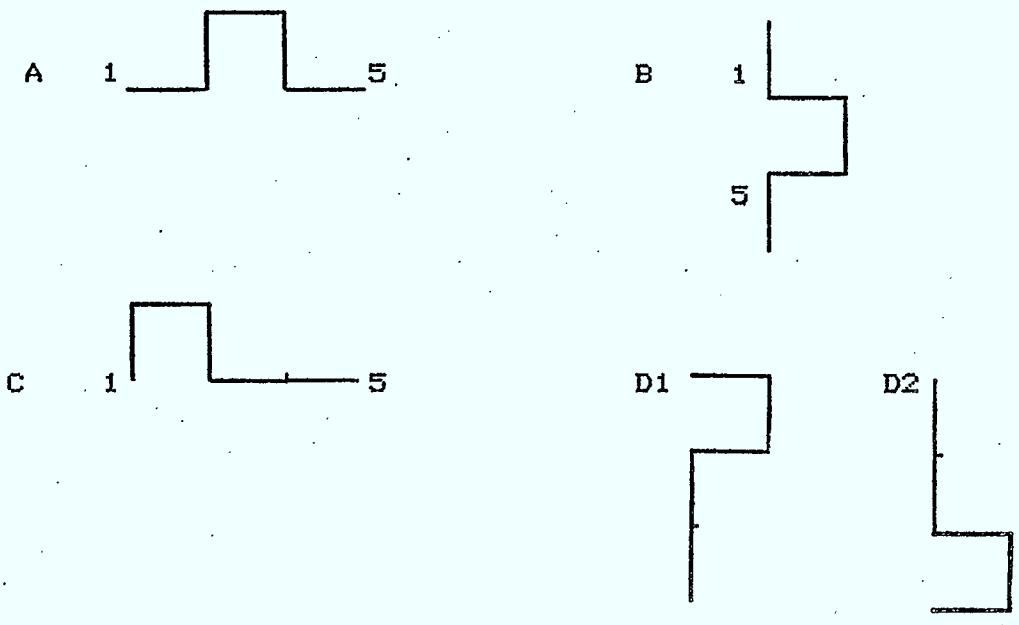


Figure 4: Proportional analogy using stick figures



APPENDIX A

A SELECTIVE REVIEW OF MACHINE LEARNING

A Selective Review of Machine Learning

Ian Morrison
Carnegie-Mellon University
Pittsburgh, Pennsylvania

Abstract

A selective review of machine learning summarizes a new approach to learning realized in two different production systems, two approaches to learning by analogy, and presents one example of a connectionist model.[1] The section on production systems describes learning by knowledge compilation in ACT* (Anderson, In Press) and learning by the chunking of subgoals in Soar. Both systems learn by combining productions that realize contiguous goals and focus attention on information related to those goals. An analysis of metaphor in common sense reasoning indicates the need for a complex representation in analogy (Carbonell & Minton, 1983) and how pragmatic considerations help analogical reasoning (Holyoak, In Press) (Holyoak, ress). The connectionist Boltzmann Machine (Hinton, Sejnowski & Ackley, 1984) model of the micro-structure of cognition demonstrates how a learning mechanism can adapt to any new situation by building and modifying connection strengths between individual processing units.

---

[1]Prepared under contract to the Communications Research Centre in Ottawa for Dr. Brian A. Schaefer. The original papers, although liberally para-phrased here to facilitate description, should be consulted for a deeper appreciation of the complexity of each theory.

Introduction

Machine learning is experiencing perhaps the greatest growth period of all the fields in Artificial Intelligence because programs that can adapt have definite advantages over those that cannot. More people have become interested in machine learning since the publication of Machine Learning: An Artificial Intelligence Approach (Michalski, Carbonell & Mitchell, 1983). An expanded version of the first chapter of the book appeared in an article in AI Magazine (Carbonell, Michalski & Mitchell, 1983). The article mentions two reasons why human learning is important to machine learning: (1) humans are our best examples of complex learning mechanisms and (2) learning machines must be understable to the humans with whom they must interact. The articles reviewed in this section all reflect some degree of concern with human learning.

In the article Carbonell, Michalski, and Mitchell (1983) identify two major components of learning: knowledge acquisition and skill refinement. Knowledge acquisition is learning which allows the learner to explain more situations with greater accuracy, i.e., to be a better predictor of its environment. Skill refinement is "the gradual improvement of motor and cognitive skills through practice." Knowledge acquisition is argued to be a conscious, symbolic process and skill refinement an unconscious, non-symbolic process but most human learning is regarded as a mixture of both. The non-symbolic nature makes skill acquisition more difficult to capture by AI techniques but Newell and Rosenbloom (1981) developed a successful symbolic model of skill refinement. An adapted version of the learning mechanism used by Newell and Rosenbloom is incorporated in the **Soar** architecture, one of the new production system approaches to learning reviewed below.

Carbonell et al. (1983) chose three dimensions to describe machine learning research. One dimension was the type of knowledge or skill required by the learner, that is, the representation of knowledge. One form of knowledge representation is the rules used in production systems. Each of these rules embody a set of conditions and a set of actions. If all the conditions of a rule are true then it becomes "instantiated" and its actions are taken.[2] Knowledge of the world, then, is represented by production rules where the variety of detectable situations is represented by the conditions and range of responses is represented by the actions of production rules. The world is decribed by a set of assertions

---

[2]In many production systems only one rule is actually fully instantiated, for example OPS5, requiring a **conflict resolution** strategy to choose which of the "partially" instantiated rules will become fully instantiated. Parallel production systems, for example CAPS, fully instantiate all rules whose conditions are met.

in working memory and all conditions test some aspect about working memory; for example, that a particular thing is there or not there, that one thing is greater than another, and so on. Actions typically add, remove, or change things in working memory, some of which may be, enable, or disable conditions of other production rules. There are four basic methods of acquiring new, or refining old, knowledge.

1. Creation of a new rule.
2. Generalization of a rule to apply to a wider range of situations.
3. Specialization or discrimination of a rule to apply to a narrower range of situations.
4. Composition of one or more rules to create a new rule to permit faster reaction to a particular situation, since only one rule needs to be instantiated.

Both the knowledge compilation technique used by Anderson in the ACT* production system and the Soar architecture accomplish generalization as a side-effect of creating new rules. Anderson (In Press) has suggested that the same approach may be applied to discrimination. Anderson's approach is the other production system we review. While ACT* and Soar are based on production systems both super-impose additional structure on the production system. Each super-imposes a goal structure, ACT* adds an additional long memory structure, and Soar adds other structure described below.

A second dimension used by Carbonell et al. (1983) to describe machine learning research concerned how much inference the underlying learning strategy applied to input information, ranging from the total lack of inference in rote learning through increasing degrees of inference in learning from instruction, learning from analogy, learning from examples, and learning from observation and discovery. Most work on machine learning has focused on learning from examples but learning by discovery and learning by analogy has recently been attracting more interest. Two approaches reflecting the complexities in representing and applying analogies are reviewed here (Carbonell & Minton, 1983; Holyoak, In Press)

Carbonell et al. (1983) sketch a history of machine learning centred around three paradigms. Early attempts used general purpose learning mechanisms with little task or domain-specific knowledge, called neural nets or self organizing systems, the best examples of which are Rosenblatt's perceptrons and Selfridge's pandemonium. The approach ended in failure and was dismissed (Rich, 1983) but has been revived in the form of connectionist models. We will review one particularly promising connectionist model, the Boltzmann

Machine (Hinton, Sejnowski & Ackley, 1984).

Contiguous Learning in Production Systems

Brownston, Farrell, and Kant (In Preparation) provide an overview of learning methods in production systems. The most common are generalization, discrimination, composition, and proceduralization. In **generalization** a new rule is built from other rules or by learning from examples. The new rule will be applicable in all the situations that the old rules were, and possibly more. The most common technique is to delete a condition which does not apply very often, or applies to some situations but not others. Unfortunately the deletion technique often leads to overly general productions which apply in too many situations. In **discrimination** the system creates one or more variants of a rule, usually by adding conditions, so that each variant is instantiated in fewer situations. Discrimination is, therefore, the process of restricting an overly general production. **Composition** is a mechanism which creates a new rule by combining the conditions and actions (removing redundant elements) of two production rules which are instantiated in sequence. When the new rule is instantiated it will be favoured by conflict resolution because of its greater specificity (i.e., more detailed conditions) in situations where the old rules are also instantiated. **Proceduralization** is a learning mechanism that attempts to reduce the size of composed productions. Variables in conditions and actions are replaced by the actual values so that the next time the conditions arise long term memory[3] need not be accessed. Anderson (In Press) has argued that the processes of composition and proceduralization, collectively termed knowledge compilation, can account for inductive learning.

Knowledge Compilation in ACT*

Anderson (In Press) recently noted that most learning systems implicitly assume that inductive learning cannot occur by association through contiguity. These systems assume that noncontiguous examples must be compared to formulate hypotheses and then take appropriate actions. But two papers at the 1983 Machine Learning Conference, one by Anderson and the other by Rosenbloom and Newell adopted the contiguity character. The common ground for the two different approaches was that (1) behaviour was controlled by a hierarchical goal structure used in problem solving rather than by specific inductive

---

[3]Remember that Anderson's ACT* uses a separate long term memory in addtion to working memory.

processes, and (2) learning occurred by creating single production rules that accomplished the task previously requiring multiple rules. Anderson argued that these two architectural assumptions are sufficient to account for inductive learning within the scope of the ACT* theory of learning (Anderson, 1983).

In ACT* a learner begins with declarative knowledge relevant to the execution of a skill and general interpretive procedures to apply to these facts. Knowledge compilation operates on the traces of two interpretive procedures, general problem-solving procedures and general analogy procedures, to create more efficient productions specific to the task domain. Knowledge compilation is subdivided into two subprocesses called composition and proceduralization which operate as described above.

Anderson drew several conclusions from analysis of a subject's protocol[4] during a lisp programming episode which was simulated using GRAPES, a representation of ACT* used to simulate programming episodes. The first conclusion emphasized the importance of structural analogy in bridging the gap between current and desired behaviour. Knowledge appears to be isolated, requiring something, such as analogical processing, to transfer from one context to another. The second conclusion was that problem-solving for novice programmers is organized as a hierarchical goal structure in which the goals are expanded in a depth-first and left-to-right manner. This implies that novices follow the implications of one goal as far as possible before processing related goals that could be explored at each level in the goal hierarchy. This is important because the structure of the hierarchical goal tree is crucial to the compilation process because it identifies which parts of the problem-solving episode belong together and which do not. Anderson notes that a breadth-first expansion has been found for experts (Jeffries, Turner, Atwood & Polson, 1981). The third conclusion was that knowledge compilation is an important mechanism for building new productions which can streamline later performance. The subject's learning and the GRAPES simulation could be described as an episode of inductive learning but problem solving through analogy coupled with knowledge compilation are sufficient to explain the results without recourse to explicit inductive mechanisms.

GRAPES distinguishes between **inherent goals**, intrinsic parts of the task whose achievement solves part of the original problem, and **planning goals** goals whose results guide the solution process but are not part of the problem solution itself. One way to

---

[4]A protocol is a subject's monologue of what comes to his mind as he is working through a problem.

perform composition is to eliminate the planning goals intermediate between two inherent goals. Knowledge compilation produces new rules which preserve the inherent goals specific to the task domain and lose the goals from general processes like structural analogy.[5]

> The interesting observation is that compilation results in clause deletion and replacement of constants with variables. Compilation deletes clauses associated with omitted goals and with planning. Variables from planning productions can remain in the compiled productions. This is how we are able to get the effect of generalization through compilation. ... Specifically, it appears that generalizations can be formed through the process of compiling analogies.

This path to generalization is distinguishable from the standard inductive path because it is generated from a single item[6] and it has added flexibility because of its problem-solving origins.

Discrimination is handled by having the system deliberately follow the steps to form a discrimination, i.e., through problem-solving productions rather than an automatic process "watching" the system. If a sequence of productions results in a discrimination then compiling the sequence results in a discriminate production. This requires that the system must make an error, correct that error, and identify the relevant features distinguishing the current instance from prior instances in that category (i.e., make a deliberate hypothesis). Lewis and Anderson report experimental evidence that these conditions are met when discrimination occurs. Anderson makes the following conclusion:

> The fundamental point then is that the induction process occurs as a conscious problem-solving effort to find a basis for dealing with a new case. ... The fundamental category of behaviour is problem-solving not induction. This theory is not one of learning by temporal contiguity but learning by contiguity in the problem-solving goal structure. There is no such thing as unconscious induction of features.

Universal Subgoaling and Chunking in Soar

A recent approach to both problem solving and learning is represented by the Soar architecture. Soar combines ideas from two doctoral theses supervised by Newell (Laird, 1984, Rosenbloom, 1983). Production systems were viewed as efficient but computationally

---

[5]New rules, it should be emphasized, supplement old rules in special circumstances rather than replacing them.

[6]Explicit generalization mechanisms require more than one item in order to generalize, i.e., they replace a number of different constants with a single variable descriptor.

limited representations of knowledge so additional structure was super-imposed on a production system[7] to build a general problem solver called Soar.

Attention in Laird's Soar is focused by a **current context** and much processing during problem solving concerns search for the appropriate elements to fill the slots in the current context. The slots in the current context are the **goal** of problem solving, the **problem space** in which problem solving occurs, the **state** describing the problem, and the **operator**, or action, which changes (or adds some aspect to) the state. The problem space is the set of states that can be generated given the set of operators and an initial state. Problem solving stops when the current state satisfies the goal; until then apropriate elements must be chosen to fill the slots in the current context. Soar detects difficulties stemming from any slot of the current context (regardless of the slot's particular content) or difficulties in specifying slot membership for the next problem solving cycle and generates a goal to resolve the difficulty; thus Soar is a reflective problem solver because it can reason about its own problem solving activity. Since all goals are generated in this fashion regardless of the specific problem domain Soar is said to exercise **Universal Subgoaling**.

Rosenbloom developed a learning algorithm which modeled the power law of practice and was based on the chunking[8] theory of learning (Newell & Rosenbloom, 1981). Performance gradually improved as newly built productions based on chunks required less frequent subgoal decomposition. Laird, Rosenbloom, and Newell (1984) hypothesized that combining their general problem solver and chunk-based practice mechanism may produce a general intelligent agent capable of more interesting types of learning than just speeding up performance (as in practice). Chunks in the combined system were built based on the parameters and results of goals experienced during problem solving.

The current context focuses attention in Soar because problem solving can only occur on one goal, in one problem-space, on one state with one operator at a time.[9] Membership for the current context slots is determined by **preferences**. A major function of production rules in Soar is to make preferences for particular items to occupy the current context slots at particular points during problem solving. Making preferences for some

---

[7] A modified version OPS5 with conflict resolution removed.

[8] A chunk (Miller, 1956) is a single unit previously recognizeable only as distinct units. For example, a useful chunk using the letters B, I, and M might be IBM.

[9] Actually the most recent version of Soar allows some operators to be instantiated in parallel.

elements rather than others constrains search (i.e., they reduce the set of candidates for a slot) so productions that make preferences control search, and thus problem solving.

There are two major processing phases in Soar, an **elaboration** phase when the state is manipulated and preferences are made, and a **decision** phase when items are chosen to fill each slot in the current context for the next elaboration-decision cycle. During elaboration the state may be manipulated (i.e., some attributes added or changed) but the major aim of the elaboration phase is to build preferences for the slots in the current context, i. e., what should be the next goal, problem-space, state, or operator. A preference for an item specifies the slot for which it is preferred, the context in which it is preferred (i.e., the value of the other slots), the value of the preference (acceptable, reject), and perhaps a partial ordering compared to other preferences (worst, worse, equal, indifferent, better, best). A special value (parallel) allows some operators to be processed simultaneously, effectively permitting multiple objects in the operator slot.

The decision phase decides which items should fill the slots in the current context on the next elaboration-decision cycle. The Soar archictecture follows a fixed procedure for this determination based on the knowledge encoded in preferences. If preferences do not isolate a unique object for a slot, or no preferences exist for any slot, Soar detects a **difficulty** and creates a subgoal to resolve it. This is the only way to create a goal in Soar; deliberate subgoaling is not permitted. Earlier versions of Soar permitted deliberate subgoaling in user-defined productions but, and as somewhat of a surprise, it was found that the situations requiring deliberate subgoals were detectable as difficulties by Soar. Soar recognizes four difficulties concerning the items in the current context slots: **resolve-tie**, **resolve-no-change**, **resolve-rejection**, and **resolve-conflict**. A resolve-tie subgoal is created if the preferences for a slot do not lead to the selection of a single object; a resolve-no-change subgoal is created if there is no change during a decision cycle, a resolve-rejection subgoal is created if all objects with acceptable preferences for a slot also have reject preferences, and a resolve-conflict subgoal is created if at least two objects have conflicting preferences. Soar maintains special goals and problem-spaces designed to resolve each of these difficulties. They are special only because Soar automatically generates them when the difficulties arise. It should be noted that detection of difficulties and resolution of preferences is independent of domain knowledge.

The basic assumption underlying Soar as a general learning mechanism is that all complex behaviour, including learning, occurs as search in problem spaces (Newell, 1980). Learning is simply a recorder of experience which determines the form of what is learned.

Since chunking acts as a recorder of goal-based experience it is a good candidate for a learning mechanism. Chunking caches the processing of a subgoal so that a chunk can substitute for the usual processing of the subgoal the next time it, or a similar one, is generated. The operation is task-independent, occurring during processing through experience with processed goals, and requires no extensive analysis either before, during, or after performance. Only goal-related things are chunked, or learned so superficial, non-goal-related variations are irrelevant to a chunked production rule, providing an implicit generalization mechanism. To become a general learning mechanism, however, the chunking learning algorithm must be combined with a general problem solver. A good candidate is Soar, a "reflective" problem solver which can reason about its own problem solving behaviour by creating subgoals (in the same format as subgoals for other problem solving activities) to do so. Laird, Rosenbloom, and Newell (1984) note four contributions towards chunking as a general learning mechanism accomplished by implementing chunking within Soar.

1. Chunking can be applied to a general solver to speed up its performance.
2. Chunking can improve all aspects of a problem solver's behaviour.
3. Significant transfer of chunked knowledge is possible via the implicit generalization of chunks.
4. Chunking can perform strategy acquisition, leading to qualitatively new behaviour.

In summary, in Soar both problems and routine tasks are formulated as heuristic search. A problem-space consists of a set of states and a set of operators that transform one state into another. Problem solving begins with an initial state and proceeds through the application of operators to a desired state. Operators, tests for goal satisfaction or failure, and search control are implemented as productions. Domain dependent knowledge can guide search control (through preferences) but spaces which have only operators and goal recognizers will work correctly given enough time (or production cycles). Directly available knowledge (i.e., that available within the current context) may not be sufficient to resolve search control or to apply an operator to a state. Soar recognizes such a difficulty and creates a subgoal to resolve it, just as for any other problem: i.e., Soar selects a problem-space for the subgoal where success is finding a state which resolves the subgoal. Thus, Soar builds a hierarchy of goals and problem-spaces. The hierarchy contains special goals and problem-spaces (special only in the sense that Soar automatically generates then when difficulties arise) to resolve difficulties that can occur in any domain. In this

organization all aspects of the system's behaviour are open to problem solving when necessary; this is called universal subgoaling.

Universal subgoaling and learning by chunking is a potentially powerful combination which may encompass forms of learning previously thought too complex for a simple chunking algorithm. The learning algorithm can be simple because what is learned is determined by the goal-related problem solving.

> The power of chunking in Soar stems from Soar's ability to automatically generate goals for problems in any aspect of its problem-solving behavior: a goal to select among alternatives leads to the creation of a production that will later control search; a goal to apply an operator to a state leads to the creation of a production that directly implements the operator; and a goal to test goal-satisfaction leads to a goal-recognition production. As search-control knowledge is added, performance improves via a reduction in the amount of search. If enough knowledge is added, there is no search; what is left is a **method** -- an efficient algorithm for a task. In addition to reducing search within a single problem space, chunks can completely eliminate the search of entire subspaces whose function is to make a search-control decision, apply an operator, or recognize goal-satisfaction.

Because of the uniformity of its problem-space representation and universal subgoaling Soar can produce **within task** and **across task** learning without explicitly attempting to do so. A task that shares subgoals with another task can produce chunks that are useful for the other, yielding across task transfer of learning. Within task learning occurs when a subgoal arises more than once while attempting to solve the task. Since many aspects of the context in which a chunk was created are ignored (i.e., those irrelevant to the goal) generalization to similar situations sharing the same goal-related objects but possessing superficial differences occurs implicitly without an explicit attempt to do so. In other words, the chunk ignores all irrelevant information and is instantiated in all situations where the relevant information is present. An unfortunate problem with chunking (from the programmer's perspective but perhaps not from the psychological modeler's viewpoint) is that it produces overly general productions, a problem leading to negative transfer in humans. Methods of recovery from overly general productions are required. Laird, Rosenbloom, and Newell (1984) suggest that the way humans recover from over-generalization should be investigated so that the problem solving activities involved in the recovery can be used to build chunks which will override the over-general ones.

Soar has demonstrated its application to learning in typical vehicles for computer learning programs (e.g., tic-tac-toe, eight-puzzle) and in a complex problem solving environment: a version of the R1 program for configuring computers at Digital Equipment

Corporation was converted to Soar (Rosenbloom, Laird, McDermott & Newell, 1984). Van de Brug and Rosenbloom (In Preparation) have extended the R1-Soar implementation and investigated several configurations of problem-spaces for learning and processing efficiency. The work identified a number of positive properties of Soar:

1. The modularity of problem-spaces, which allows local reasoning and can broadly resemble the human components of a task, facilitates both maintenance of the system and investigation of various potential approaches to task solution.

2. Rules tend to be similar within each problem-space, each having rules to initialize the problem-space and states, propose acceptable operators, implement search control, apply operators, and recognize success (or failure).

3. Control knowledge is clearly separated from domain knowledge. Conflict resolution is replaced by the preference system.

4. The representation leads to a separation of "active" from "passive" (or volatile versus fixed) state information.

A Note About ACT* and Soar

Soar and ACT* are similar in their learning strategy in that they compose productions which act on goals contiguous in the goal hierarchy and only include items in the new productions which are goal-relevant. However, the two systems are quite different. ACT* was designed as a model of human cognition but the designers of Soar, while cognizant of human cognitive psychology, were after a system which could approximate an ideal, general intelligent agent, human or not. Soar does not posit an additional long term memory as ACT* does, and its goal differentiation is different from ACT*'s (e.g., there are no inherent or planning goals). The uniform representation in problem-spaces and the reflective nature of Soar also differs from ACT*. As Soar and ACT* are extended to a wider variety of problems, and thus a greater overlap, comparison should elucidate the advantageous parts of each approach.

Learning by Analogy

Anderson considered analogy to be one of the basic problem-solving methods available to the human learner. The structural analogy process which he described, however, is the simplest. Carbonell and Minton (1983) focus on the mapping problem in analogy, i.e., the determination of what parts of one situation are relevant to another, and how pragmatic information can aid the process. They also argue that a complex representation of the

mapping process itself facilitates analogical reasoning. Holyoak (In Press), in concert with a number of colleages and expressed in a forthcoming book (Holland, Holyoak, Nisbett & Thagard, In Preparation), has developed an approach which de-emphasizes the syntactic description of analogy and emphasizes the goal-related, problem solving nature of analogical processing. Holyoak also emphasizes how pragmatic information can aid analogical reasoning.

Metaphor and Common Sense Reasoning

Carbonell and Minton (1983) think the weight of empirical evidence has not yet "tipped the academic scales" in favour of metaphor and analogy as the basic process involved in common sense reasoning, perhaps a good use of a pun since the focus of their investigation was on the use of the balance scales analogy. It plays a major role, however, as evidenced in a report published by Carbonell only a month later (Carbonell, Larkin & Reif, 1983) describing the general reasoning processes involved in scientific endeavour. Carbonell and Minton's central hypothesis is the:

> **Experiential reasoning hypothesis**: Reasoning in mundane, experience-rich recurrent situations is qualitatively different from formal, deductive reasoning evident in more abstract, experimentally contrived, or otherwise non-recurrent situations (such as some mathematical or puzzle-solving domains).

The authors claim that formal modes of thought are not dominant in mundane situations because they are seldom necessary and require more effort when used than analogical reasoning, which is at least partly responsible for common sense reasoning. Analogical reasoning is defined roughly as "the process by which one recognizes that a new situation is similar to some previously encountered situation, and uses the relevant prior knowledge to structure and enrich one's understanding of the new situation." Metaphorical reasoning is "that subset of analogical reasoning in which the analogy is explicitly stated or otherwise made evident to the understanderer."

Analogy requires access to large amounts of past knowledge, reaching conclusions without benefit of formal deductive reasoning, and consists of a target, a source, and an analogical mapping. One particularly prevalent metaphor concerns reasoning about abstract entities as if they were weights, i.e., using the balance principle as an analogy. Based on the observation that language is heavily endowed with words that describe physical attributes and people use these words to describe abstract entities, Carbonell and Minton propose another hypothesis:

> **Physical metaphor hypothesis**: Physical metaphors directly mirror the underlying inference processes. Inference patterns valid for physical attributes are used via the analogical mapping to generate corresponding inferences in the target domain.

The inference pattern for the balance principle is straight forward. Given an input set of signed quantities, whose magnitudes are analogous to the "weights" and whose signs are analogous to the sides of a binary issue, the side with the greatest weight is chosen with a corresponding qualitative calculation of how far out of balance the system is. Carbonell and Minton argue that this simple analogy accounts for human inferences in many situations and summarize the four stages in Carbonell's (1983) model of analogical problem solving:

1. Recalling one or more past problems that bear strong similarity to the new problem.
2. Constructing a mapping from the old problem solution process into a solution process for the new problem, exploiting known similarities in the two problem situations.
3. Instantiating, refining and testing the potential solution to the new problem.
4. Generalizing recurring solution patterns into reusable plans for common types of problems.

The greater part of processing in that model concerned building a mapping from the similar past problem situation to help in the new situation. The role of metaphor is to capture and communicate mappings from well known experiential situations to new, less structured domains. Although such mappings often fail to provide deep insight into the new situation they often convey quick, superficial understanding sufficient for normal everday functioning.

The central issue in metaphor comprehension is the analogical mapping problem, i.e., the identification of the relevant parts of the source to map to the appropriate parts of the target. A mapping based on simple similarity is not sufficient to capture the complexity of most analogies. The matching process used in mapping must be focused to eliminate spurious, unimportant similarities from consideration. Focusing strategies utilize pragmatic considerations to enormously constrain the matching process. First, the typical use of the metaphor may be known, i.e., the methaphor is at least partially frozen, reducing the number and allowable elements to be matched. Second, salient features can guide matching in the mapping process when novel metaphors are used. Finally, an analysis of more complicated extended metaphors, such as scientific analogies, shows that not all details of the mapping are needed; the establishment of "beachheads" enables the gist of the metaphor and allows further elaboration as required.

Analogies require a link between two domains. Single LIKE links are too simplistic to capture the complexity of analogical processing and multiple sets of LIKE links are insufficient because they rely on a reductionistic representation of analogy. A mapping structure allowing only explicitly specified information to be transferred from one domain to another is proposed. The mapping structure implements the LIKE relation, can contain meta-information about the mapping, and can be identified as the source of new inferences in the target domain made as as result of the analogy. The analogy can therefore be extended incrementally over time making inapplicable parts susceptible to retraction processes. In inheritance networks complex IS-A relations allow the specification of exactly what information to transfer, and what not to transfer, from a particular superordinate to members of a class. Both LIKE and IS-A relations require a mapping between concepts. Carbonell and Minton (1983) refer to the analogical mapping process as lateral inheritance as opposed to the vertical inheritance of IS-A relations.

The Pragmatics of Analogical Transfer

Holyoak (In Press) begins with the observation that current expert systems typically do not learn from experience because their brittleness (Holland, In Press) prevents the transfer of experiential knowledge to novel situations. He integrates studies of analogical problem-solving with a pragmatic framework for induction (Holland et al., In Preparation) which argues that progress has been slowed by misguided attempts to specify purely syntactic contraints on induction without considering the goals of the system or the context in which induction occurs.

> From the pragmatic perspective, the central problem of induction is to specify processing constraints ensuring that the inferences drawn by a cognitive system will tend to be (a) relevant to the system's goals and (b) plausible. What inductions should be characterized as plausible can only be determined with reference to the current knowledge of the system. Induction is thus highly context dependent, being guided by prior knowledge activated in particular situations that confront the system as it seeks to achieve its goals. The study of induction becomes the study of how knowledge is modified through its use.[10] The key ideas are that induction is (a) directed by problem solving activity and (b) based on feedback regarding the success or failure of predictions generated by the system.

---

[10] In a related view Scott and Vogt (Scott & Vogt, 1983. Scott, 1983) argue that the goal of learning is the construction of an organized representation of experience rather than improved performance. This view is particularly important when considering the Boltzmann Machine which focuses on learning rather than expert performance.

A mental model is a representation that makes predictions about some part of the environment and the purpose of induction is to refine mental models. Internal mental models can be described in terms of morhpisms. A morphism is a set of states and a **transition function** T that relates each state to its successor. The components of environmental states and system outputs are organized into categories by a **mapping function h**; category members are indistinguishable to the system. Homomorphic models are **commutative**: a transition in the environment and determination of the resulting state's category is the same as determining the category of the initial state and then carrying out the transition in the model. Commutativity will sometimes fail in realistic mental models because a prediction of the model does not match receptor input, triggering inductive change. A basic inductive change involves generating new subcategories for abberant cases and refining the transition function, i.e., discrimination. As with new rules in knowledge compilation the new, specialized part of the transition function need not replace the more general one. The old, general expectations can act as defaults in cases where they are not overridden by more specific expectations.

The characterization of mental models as morphisms is important for an account of analogical problem solving for the following reasons. First, a solution plan can be viewed as a model in which the initial state is a problem representation, the final state is a representation of the class of goal satisfying states, and the transition function specifies a plan for transforming the former into the latter. Second, an analogy can itself be viewed as a morphism which can help separate the important from the less important differences between analogs. Finally, the initial solution plan constructed from an analogy is often imperfect in much the same way as a mental model may in general be imperfect, triggering more inductive corrections.

The mapping function **h** and the transition function T are represented by condition-action rules in a production system[11]. There are three types of rules in the system: (1) empirical rules describing the environment and its behaviour, (2) diachronic rules describing the transition function and generating temporal expectations about the behaviour of the environment, and (3) synchronic rules describing the mapping function and performing temporal categorizations of the components of environmental states. Synchronic rules capture the kind of categorical and associative information often representated in static semantic networks while diachronic rules represent information about the expected effects of

[11] Specifically, the model is couched Holland's production system.

system actions, such as problem-solving operators. A example of a synchronic rule is "If an object is small, feathered, and builds nests, then it is a bird" whereas an example of a diachronic rule is "If an object is a bird, and it is chased, then it will fly away." The common rule format subjects both types of information to the same inductive pressures and the same processing constraints determine their activation.

Several principles govern the organization and processing of rules (Holland, In Press). Rules, receptors, and declarative memory stores can post messages that guide system behaviour. Active messages are matched against rule conditions and those rules completely matched compete for execution by placing bids.[12] Three factors determine the size of bids: the specificity of the rule, the strength of the rule (a numerical value based on past usefulness), and the support accruing to the rule from the messages that matched it (a measure of the current activation level of the messages satisfying the rule's condition). Rules thus compete for the "right" to post messages. Inductive mechanisms favour the development of clusters of rules that often work well together. Conflict resolution is minimized because rules only post messages, and contradictory messages can coexist until one attains sufficient support to suppress its alternatives or until the need for an effector actions demands a decision. Goal attainment is the basic source of "reward".

Analogy aids the construction of new rules in a novel domain by transferring knowledge from a better understood source domain. The overall similarity of target and source domains varies from the mundane to the methaphorical. Examples, according to Holyoak, are commonly used as analogical models for problem solving when overall similarity is mundane[13]. Holyoak's own experience has been in the metaphorical end of the continuum. Analogy differs from other inferential mechanisms because it does not dwell on the immediate problem situation but requires information outside the immediate problem. Precisely because few strong rules are available to apply to an ill-defined problem, weak synchronic rules that activate associations to the target can direct processing. Four basic steps are involved in analogical problem solving:

1. Constructing mental representations.

---

[12]The number of rules acting simultaneously is limited.

[13]For example, student's solutions of geometry or computer programming problems as in the knowledge compilation section above (Pirolli & Anderson, In Press).

2. Selecting the source as a potentially relevant analog to the target.

3. Mapping the components of the source and the target.

4. Extending the mapping to generate a solution to the target.

Holyoak uses variations of the convergence analogy in his experiments and illustrations. In one example the target domain requires a stomach tumour to be irradiated (surgery is impossible) or the patient will die. However, the ray intensity needed to destroy the tumour is too intense for intervening tissue (Duncker, 1945). The solution is to use many weaker rays to converge on the same spot. A similar source domain is presented to some subjects but not others. Here a general wishes to capture a fortress in the centre of a country but cannot send all his troops down one road or rail line. The solution is to use many paths to the fortress. The abstract structure common to the two problems is a schema for convergence problems, i.e., a class of problems for which the convergence solution may be possible. Analogy is thus closely related to the induction of categories by generalization. Because the information in a problem schema can be represented by a set of interrelated synchronic and diachronic rules, a schema is represented as a rule cluster.

Two pragmatic problems concerning analogical problem solving are the efficient retrieval of a relevant source analog and the determination of which properties of the source analog to use in developing a model of the target problem. Useful source analogs share multiple, goal-related properties with the target. Goal-related diachronic rules of the source analog provide the basis for generating new diachronic rules appropriate to the target problem. Therefore, syntactic approaches to analogy (e.g., Gentner, 1983), which do not consider the impact of goals on analogical transfer, fail. Syntactic approaches miss the fundamental relation between synchronic and diachronic rules, i.e., between the mapping and the transition functions, which are affected by goals in particular problem contexts.

Analogy involves "second-order modeling", i.e., a model of the target domain is constructed by "modeling the model" of the source domain. The ideal case occurs when the mapping is one-to-one, or isomorphic. Even in the ideal case not all elements must be transferred to the target domain. Only the parts relevant to the solution plan are needed. These are the goal which is the reason for it, the resources which enable it, the contraints which prevent alternative plans, and the outcome which is the result of executing the plan. The definition of analogy as a relation between problem models makes it possible to specify the information transferred from source to target in a principled manner.

The initial mapping typically involves detection of an abstract similarity between

corresponding goals, constraints, object descriptions, and operators common to the two analogs. Once established the inital mapping can be extended. As the source is unpacked a model is built of the target. Unpacking continues until a solution is found or the analog begins to break down. Two important mapping relations are **structure preserving** differences which allow construction of corresponding operators and **structure violating** differences which prevent the construction of corresponding operators. An analogy breaks down at the level where differences prove to be mostly of the structure violating type and the completeness of an analogy is measured by the degree to which all differences are structure preserving. However, the usefulness of an analogy is determined by pragmatic considerations. Imperfect analogies can be useful for first approximations which lead to further refinements.

When do people notice the relevance of potential analogies? A summation principle ensures that analogs sharing multiple properties with the target domain will be activated. Superficial similarities do play a role, although a minor one because goal-related properties tend to dominate. Plausible source analogs share multiple components with the target problem. For example, a source activated by both an intial state and goal state is likely to have common diachronic rules transforming the initial state to the goal state. Remote analogs are more difficult to retrieve specifically because they share few surface properties but greater concentration of solution-related (or structural) features helps to retrieve a remote, but useful, source analog. The definition of a feature as surface or structural depends on the problem solver's goal (i.e., a structural similarity in one situation may be a surface similarity in another) and a person's ability to distinguish them is imperfect (otherwise no need to employ an analogy). Once retrieved, surface properties have less impact on the mapping process than structural features; i.e., they have a greater impact in the selection of a source analog than on the mapping. Experimental evidence supporting these views is presented.

A Connectionist Model: Learning in a Parallel Network

As we discussed in the introduction connectionist models are not new. Connectionist models are comprised of simple processing units connected by links which can vary in association strength. A unit is activated if the sum of its input links exceeds a threshhold value and activation is passed along output links moderated by the strengh of the link. A "symbol" in such a network is usually described by a "pattern" of activity in a number of units. The Boltzmann Machine (Hinton, Sejnowski & Ackley, 1984), which we describe below,

has this basic organization except that all links between units are symmetrical so there is no conception of input and output, per se, to a single unit. Although we concentrate on the Boltzmann Machine much related work is available.[14]

The Boltzmann Machine

The Boltzmann Machine is a massively parallel network of simple "neuron-like" units in which knowledge is stored in the strengths of the associations between units. Some tasks that require massive amounts of similar computations (e.g., vision) can be tremendously speeded if the computations can be accomplished as simultaneously as possible. Propagating constraints between units is one way of accomplishing simultaneity but the paths for constraint propagation must be known in advance to set up a useful network. The Boltzmann Machine is a massively parallel network which can learn constraint paths between appropriate units. It can adapt its internal structure to any problem by simply being shown examples from the domain. From these "lessons" the network adjusts its connection strengths so that it can produce examples with the same statistical probability as found in the domain. However, although the learning algorithm used in the Boltzmann Machine is guaranteed to build an appropriate internal representation, it is very slow.

Constraint satisfaction methods typically involve strong constraints (Waltz, 1975, Winston, 1984) that must be satisfied by any solution but the Boltzmann Machine is better suited to tasks involving weak constraints that involve some cost if violated, but are not rejected by such violation. A weak constraint can be seen as a matter of degree whereas a strong constraint is absolute. The quality of any solution is measured by the total cost of violations and is reflected in its plausibility. The mechanics of the Boltzmann Machine is described in the extended quotation below which, because of the technical nature of the Boltzmann Machine, is reprinted almost in its entirety.

> The machine is composed of primitive computing elements called **units** that are connected to each other by bidirectional **links**. A unit is always in one of two states, **on** or **off**, and it adopts these states as a probabilistic function of the states of its neighboring units and the **weights** on its links to them. The weights can take on real values of either sign. A unit being on or off is taken to mean that the system currently accepts or fejects some elemental hypothesis about the

---

[14] This and other work is described in two forthcoming volumes edited by D. E. Rumelhart and J. L. McClelland under the title Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Cambridge: MA, Bradford Books, In Press. See also (Feldman & Ballard, 1982, Sutton & Barto, 1981a, Sutton & Barto, 1981b, Klopf, 1979, Klopf79b, Granger, 1983, Granger & McNulty, 1984).

domain. The weight on a link represents a weak pairwise constraint between two hypotheses. A positive weight indicates that the two hypotheses tend to support one another; if one is currently accepted, accepting the other should be more likely. Conversely, a negative weight suggests, other things being equal, that the two hypotheses should not both be accepted. Link weights are **symmetric**, having the same strength in both directions (Hinton & Sejnowski, 1983).

The resulting structure is related to a system described by Hopfield (1982), and as in his system, each global state of the network can be assigned a single number called the "energy" of that state. With the right assumptions, the individual units can be made to act so as to **minimize the global energy**. If **some** of the units are externally forced or "clamped" into particular states to represent a particular input, the system will then find the minimum energy configuration that is compatible with that input. The energy of a configuration can be interpreted as the extent to which that combination of hypotheses violates the constraints implicit in the problem domain, so in minimizing energy the system evolves towards "interpretations" of that input that increasingly satisfy the constraints of the problem domain.

A simple algorithm for finding a combination of truth vaues that is a **local** minimum is to switch each hypothesis into whichever of its two states yields the lower total energy given the current states of the other hypotheses. If hardware units make their decisions asynchronously, and if transmission times are neglibible, then the system always settles into a local energy minimum (Hopfield, 1982). Because the connections are symmetric, the difference between the energy of the whole system with the k^{th} hypothesis rejected and its energy with the k^{th} hypothesis accepted can be determined locally by the k^{th} unit (i.e., the energy gap). ... Therefore, the rule for minimizing the energy contributed by a unit is to adopt the **on** state if its total input from the other units and from outside the system exceeds its threshold. This is the familiar rule for binary threshold units.

The simple, deterministic algorithm suffers from the standard weakness of gradient descent methods: It gets stuck in **local** minima that are not globally optimal. This is not a problem in Hopfield's system because the local energy minima of his network are used to store "items": If the system is started near some local minimum, the desired behavior of to fall into that minimum, not to find the global minimum. For constraint satisfaction tasks, however, the system must try to escape from local minima in order to find the configuration that is the global minimum given the current input.

A simple way to get out of local minima is to occasionally allow jumps to configurations of higher energy. An algorithm with is property was introduced by Metropolis et al. (1953) to study average properties of thermodynamic systems (Binder, 1978) and has recently been applied to problems of constraint satisfaction (Kirkpatrick, Gelatt & Vecchi, 1983). We adopt a form of the Metropolis algorithm that is suitable for parallel computation. ...

The decision rule ... is the same as that for a particle which has two energy states. A system of such particles in contact with a heat bath at a given temperature will eventually reach thermal equilibrium and the probability of finding the system in any global state will then obey a Boltzmann distribution. Similarly, a network of units obeying this decision rule will eventually reach "thermal equilibrium" and the relative probability of two global states will follow the Boltzmann distribution.

> The Boltzmann distribution has some beautiful mathematical properties and it is intimately related to information theory. In particular, the difference in the log probabilities of two global states is just their energy difference (at a temperature of 1).[15] The simplicity of this relation and the fact that the equilibrium distribution is independent of the path followed in reaching equilibrium are what make Boltzmann machines interesting.
>
> At low temperatures there is a strong bias in favor of states with low energy, but the time required to reach equilibrium may be long. At higher temperatures the bias is not so favorable but equilibrium is reached faster. A good way to beat this trade-off is to start at a high temperature and gradually reduce it. This corresponds to a physical annealing system (Kirkpatrick et al., 1983). At high temperatures, the network will ignore small energy differences and will approach equilibrium rapidly. In doing so it will perform a search of the coarse overall structure of the space of global states, and will find a good minimum at that coarse level. As the temperature is lowered, it will begin to respond to smaller energy differences and will find one of the better minima within the coarse-scale minimum it discovered at high temperature. Kirpatrick et al. (1983) have shown that this way of searching the coarse structure before the fine is very effective for combinatorial problems like graph partitioning, and we believe it will also prove useful when trying to satisfy multiple weak constraints, even though it will clearly fail in cases where the best solution corresponds to a minimum that is deep, narrow and isolated.

One of the more interesting aspects of the Boltzmann Machine is its domain independent learning algorithm which modifies connection strengths such that the network adopts an internal model capturing the underlying structure of the environment. For complex learning a network must contain elements which are not directly constrained by the input but also identify which connections were at fault when the network does something wrong. This credit assignment problem led to the demise of Perceptrons (Rosenblatt, 1961) which could guarantee the training of a single layer of decision units but not of the hidden units in multiple layers required for complex learning. The Boltzmann Machine can solve this credit-assignment problem by running the appropriate stochastic decision rule and running the network until it reaches equilibrium. Because the energy is a linear function of the weights in the network there is a simple relationship between the log probabilities of global states and the individual connection strenghs.

> The units of a Boltzmann Machine partition into two functional groups, a non-empty set of **visible** units and a possibly empty set of **hidden** units. The visible units are the interface between the network and the environment; during training all the visible units are clamped into specific states by the environment: when testing

---

[15] The temperature, T, is one of the variables used in the equations; it indicates the degree of "shaking" applied to prevent entrapment by local minima.

> for completion ability any subset of the visible units may be clamped. The hidden units, if any, are never clamped by the environment and can be used to "explain" underlying constraints in the ensemble of input vectors that cannot be represented by pairwise constraints among the visible units. A hidden unit would be needed. for example, if the environment demanded that the states of three visible units should have even parity -- a regularity that cannot be enforced by pairwise interactions alone. Using hidden units to represent more complex hypotheses about the states of the visible units, such higher-order constraints among the visible units can be reduced to first and second-order constraints among the whole set of units.

Hinton et al. (1984) describe an information-theoretic measure **G** of the discrepancy between the network's internal model and the environment which includes components reflecting the probability of the state of visible units when their state is determined by the environment and the probability of the state of visible units when the network is running freely with no input from the environment. The **G** metric is sometimes called asymmetric divergence. or information gain. Since the components of **G** reflecting the probability of visible states depends on the weights between units **G** can be altered. Hinton et al. (1984) use a rule to minimize **G** which depends on the probability of two units both being on when the environment is clamping the visible units and the corresponding probability when environmental input is absent.

> The surprising feature of the rule is that is uses only **locally available** information. The change in a weight depends only on the behavior of the two units it connects. even though the change optimizes a global measure, and the best value for each weight depends on the values of all the other weights. If there are no hidden units, it can be shown that G-space is concave (when viewed from above) so that simple gradient descent will not get trapped at poor local minima. With hidden units, however, there can be local minima that correspond to different ways of using the hidden units to represent the higher-order constraints that are implicit in the probability distribution of environmental vectors. ... Once **G** has been minimized the network will have captured as well as possible the regularities in the environment. and these regularities will be enforced when performing completion.

Hinton et al. (1984) investigate the ability of the Boltzmann Machine to learn what they refer to as the encoder task. The reader is referred to the original report for a description of the task and the learning process but a few points are recounted here. Hinton et al. (1984) believe that the G-spaces for which the learning algorithm is well-suited are those involving many possible solutions but the very best one is not essential. For large networks to a learn in a reasonable amount of time a sufficient number of units and weights and a liberal specification of the task are required so that no single unit or weight is essential. Good performance on completion tests requires a gentle annealing schedule. As the

annealing rate increases the error rate also increases reflecting the speed/accuracy tradeoff often observed in human reaction time experiments. Finally, later in the report another task (the shifter task) is used to demonstrate the necessity of hidden units for complex learning, a task that simple Perceptrons could not learn.

Connectionist models can differ in their representation. The Boltzmann Machine adopts a distributed representation where a concept is represented as a pattern of activity over a group of units and alternative concepts are different patterns of activity over the same units (Hinton, 1981) as opposed to a local representation where the activation of one or a few units represents a concept (Feldman & Ballard, 1982). A good argument in favour of local representations is their modularity, making connections easy to modify. Distributed representations, while less susceptible to hardware damage, make modification more difficult. However, in a Boltzmann Machine a distributed representation corresponds to an energy minimum and the problem of creating a collection of good concepts is the problem of developing a good energy landscape; the learning algorithm used by the Boltzmann Machine is capable of solving this problem.

Despite the fact that the tasks were small scale learning took a long time, a slowness which Hinton et al. (1984) use to raise several questions for which they feel they do not have good answers.

1. How does the learning time scale with the size of the problem?

2. Can the learning algorithm be generalized to exhibit the kind of "one-shot" learning in which a person is told a fact once and then remembers it for a long time?

3. How much faster is the learning when the connectivity of the network and the initial values of the weights are approximately correct for the task at hand?

4. Do good solutions generally have a particular statistical structure? If so, it may be possible to impose strong **a priori** domain-independent constraints on the values of the weights or the connectivity that will constrain the search for a good set of weights to a subspace.

However, in the discussion following the presentation of the above questions Hinton et al. (1984) demonstrate the possibility of one-shot learning and identify the factors involved in the learning-time scaling problem to be the ratio of hidden to visible units, the number of connections per unit, the number of constraints in which each visible unit is involved, the order of the underlying constraints, and the compatibility of the constraints.

Hinton et al. (1984) note that the visible units in their simulations behaved correctly but

in large, practical problems this may be unreasonable. In these situations broad degenerate minima where visible units are not strongly constrained to be on or off may be sufficient. Broad minima would probably be easier to construct than narrow minima where the state of each visible unit is crucial. Within broad minima similar concepts can be differentiated by modifying the shape of the minimum's floor to establish a set of related minima separated by small energy barriers.

Hinton et al. (1984) discuss the insufficiency of a similar formulation to the Boltzmann distribution, Bayes theorem. The Bayes rule is similar if the probability of a unit is identified with the probability of a hypothesis. However, Bayes rule is insufficient in that it provides no way for the negation of evidence to affect the probability of the hypothesis, it does not lead to symmetrical weights when two units affect each other, and although it can be generalized to cases where there are many independent pieces of evidence, it is more difficult to generalize to cases where pieces of evidence are dependent. The learning algorithm in the Boltzmann Machine focuses on the worst violations of independence and develops a set of "causal rules", represented as connections between visible and hidden units and each other, to explain them.

The Boltzmann Machine with its symmetric links is incapable of sequential behaviour. Hinton et al. (1984) suggest that a set of symmetrically connected modules asymmetrically connected to one another could solve this problem. This is not unlike a "production system architecture in which all the heavy computational work is done by a parallel recognition process that decides which rule best fits the current state of working memory." Touretzky and Hinton are working on the implementation of a production system in a Boltzmann Machine architecture.[16]

The main points of the Boltzmann Machine are that noise can aid search, that credit can be assigned on the basis of local information, and that features can be created that model the external environment. The system learns to find the appropriate representation by finding the lowest point in an energy landscape. Ackley (1984) has proposed a Boltzmann-like parallel machine, but using a reinforcement learning algorithm and backward propagation of feedback, to play the role of an evaluation function in an otherwise more traditional game playing program and has produced some interesting preliminary results. The Boltzmann Machine approach is clearly flexible, an important point given the current interest in building

---

[16] AI seminar at Carnegie-Mellon University, March 12, 1985.

parallel hardware.[17]

Many connectionist models rely on comparisons with the neuronal structure of the brain to partially justify their existence. Although the Boltzmann machine does not claim to model the functioning of the neuron it does claim to be a valid model of the micro-structure of cognition based, perhaps, on units larger than the single neuron. In this respect we note five devices identified by Crick and Asanuma (In Press) as favourites of theorists but not justified by physiological evidence. These are neurons which excite some cells and inhibit others, neurons which merely change sign, neurons which connect to all other cells of the same type, neurons with distinctive synapses which do elaborate computations, and a neuron which, by itself, can fire another cell. Two justified but absent features are (1) veto cells, which appear to veto many other cells and probably need the summed activity of several distinct inputs to fire them, and (2) the various diffuse inputs, from the brain stem and elsewhere, which may be important, not only for the general level of arousal of the cortex but also for potentiating the synaptic modification involved in laying down a memory. An additional point important for the Boltzmann Machine organization is that most cortical projections are reciprocal if not symmetrical in all details.

Perhaps more important to the Boltzmann machine is the parcellation process described by Ebesson (1984). The parcellation process involves increased neural migration and increased number of certain select neurons at the cost of selective loss of certain connections. The process occurs in both evolutionary and ontogenetic development of neural circuitry. In many neural systems axons do not invade unknown territories but rather follow the path of their ancestors. If the connection is later lost it reflects neural specialization of function. It should be apparent that if weights in a connectionist model involve more than one connection then the selective growth and loss of connections is an important mechanism for adjusting weights, and therefore an important empirical finding. It is interesting that in the Boltzmann Machine a number of hidden units become obsolete (i.e., their connection strengths tend to zero) as learning progresses. Finally, brains seem to have an inherant capacity for the overproduction of neurons. An important finding by Hinton et al. (1984) was that larger networks with excess capacity speeded learning.

---

[17] For example, the Production System Machine at Carnegie-Mellon University, the Ultracomputer project at New York University, the Thinking Machine Company's connection-machine, and others.

Conclusion

The selections in the current report indicate that goal-directed systems that learn as a side-effect of problem solving are good candidates for general machine learning systems. Explicit generalization and discrimination mechanisms become unnecessary and irrelevant, non-goal related features of the problem are automatically ignored. These systems should adapt well to the use of analogy which also appears best considered as a goal-related problem solving exercise. Even the Boltzmann Machine, with its very different architecture, maintains its own goal: that of minimizing total energy as seen from the local level. The Boltzmann machine has the added capability of developing new features in reaction to a new environment; this may prove to be its most important ability.

The reflection of Soar on its internal state suggests that it is not just the detection of success or failure that is important in learning but the old problem of the correct feedback of some kind at the appropriate time. The occurrence of explicit feedback in the form of errors and positive results is salient because of observability. External feedback invokes the credit assignment problem at its maximum degree (i.e., the farthest from the source) and is probably much less frequent in real problem solving situations. Humans clearly obtain other forms of feedback, especially in extended problem solving episodes, and if we want our machines to be competent learners, they should too. Issuing feedback closer to the source would be an important aid to learning in any mechanism. Forms of internal feedback other than those present in Soar may be necessary. Holyoak's description of analogy breakdown may be helpful in developing a similar analysis which can be applied to general problem solving to provide internal feedback. Brown and Van Lehn's (1980) Repair Theory reacts to "impasses" in problem solving which indicate that an error exists in current hypotheses and repairs are required. Another possibility is the "adventurous coefficient" suggested by Berliner (1985) as a measure of making progress. Adventurousness in a game playing program is the ratio of acceptance of non-intuitive to good moves.

References

v

Ackley, D. H. Learning evaluation functions in stochastic parallel networks. Thesis Proposal, Department of Computer Science, Carnegie-Mellon University, December. 1984.

Anderson, J. R. (1983). The Architecture of Cognition. Cambridge, MA: Harvard University Press.

Anderson, J. R. (In Press). Knowledge compilation: The general learning mechanism. In Machine Learning: An Artificial Intelligence Approach, Vol. 2 Palo Alto, CA: Tioga Press.

Berliner, H. Superpuzz and some insights on searching. AI Seminar, Computer Science Department, Carnegie-Mellon University, February 12, 1985.

Binder, K. (1978). The Monte-Carlo Method in Statistical Physics. New York: Springer-Verlag.

Brown, J. S., and Van Lehn, K. (1980). Repair theory: a generative theory of bugs in procedural skills. Cognitive Science, 4, .

Brownston, L., Farrel, R., and Kant, E. Production Systems: A Tutorial Introduction. In Preparation.

Carbonell, J. G., and Minton, S. (March. 1983). Metaphor and common sense reasoning. Technical Report CMU-CS-83-110, Computer Science Department, Carnegie-Mellon University.

Carbonell, J. G., Larkin, J. H., and Reif, F. (April, 1983). Towards a general scientific reasoning engine. Technical Report CMU-CS-83-120, Computer Science Department, Carnegie-Mellon University.

Carbonell, J. G., Michalski, R. S., and Mitchell, T. M. (1983). Machine learning: A historical and methodological analysis. AI Magazine, 4(3), 69-79.

Crick, F., and Asanuma, C. (In Press). Certain aspects of the anatomy and physiology of the cerebral cortex. In J. L. McClelland and D. E. Rumelhart (Eds.), Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol2. Applications Cambridge, MA: Bradford Books.

Duncker, K. (1945). On problem solving. Psychological Monographs, 58, Whole No. 270.

Feldman, J. A., and Ballard, D. H. (1982). Connectionist models and their properties. Cognitive Science, 6, 205-254.

Gentner, D. (1983). Structure mapping: A theoretical framework for analogy. Cognitive Science, 7, 155-170.

Granger, R. H. (1983). Identification of components of episodic learning: The CEL process model of early learning and memory. Cognition and Brain Theory, 6, 5-38.

Granger, R. H., and McNulty, D. M. (1984). Learning and memory in machines and animals: An AI model that accounts for some neurobiological data. Technical Report

227, Artificial Intelligence Project, University of California, Irvine.

Hinton, G. E. (1981). Implementing semantic networks in parallel hardware. In G. E. Hinton and J. A. Anderson (Eds.), Parallel Models of Associative Memory Hillsdale, NJ: Lawrence Erlbaum Associates.

Hinton, G. E., and Sejnowski, T. J. (1983). Analyzing cooperative computation. Cognitive Science Society

Hinton, G. E., Sejnowski, T. J., and Ackley, D. H. (May, 1984). Boltzmann machines: Constraint satisfaction networks that learn. Technical Report CMU-CS-84-109, Computer Science Department, Carnegie-Mellon University.

Holland, J. H. (In Press). Escaping brittleness: The possibilities of general purpose learning algorithms applied to parallel rule-based systems. In Machine Learning: An Artificial Intelligence Approach, Vol. 2 Palo Alto, CA: Tioga Press.

Holland, J. H., Holyoak, K. J., Nisbett, R. E., and Thagard, P. Induction: Processes of Inference, Learning, and Discovery. In Preparation.

Holyoak, K. J. (In Press). The pragmatics of analogical transfer. In Gordon H. Bower (Ed.), The Psychology of Learning and Motivation New York: Academic Press.

Hopfield, J. J. . Neural networks and physical systems with emergent collective computational abilities. (pp. 2554-2558). National Academy of Sciences USA

Jeffries, R., Turner, A. A., Polson, P. G., and Atwood, M. E. (1981). The processes involved in designing software. In J. R. Anderson (Ed.), Cognitive Skills and their Acquisition Hillsdale, NJ: Lawrence Erlbaum Associates.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by simulated annealing. Science, 220, 671-680.

Klopf, A. H. (1979). Goal seeking systems from goal-seeking components: Implications for AI. The Cognition and Brain Theory Newsletter, 3, 54-62.

Klopf, A. H. (1979). The Hedonistic Neuron: A Theory of Memory, Learning, and Intelligence. Washington, DC: Hemisphere.

Laird, J. E. . (May, 1984). Universal Subgoaling. Technical Report CMU-CS-84-129, Computer Science Department, Carnegie-Mellon University.

Laird, J. E., Rosenbloom, P., and Newell, A. (1984). Towards chunking as a general learning mechanism. AAAI.

Lewis, C.H., and Anderson, J. R. The role of feedback in discriminating problem-solving operators. Manuscript submitted for publication.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. . Journal of Chemical Physics, 1953, 6, 1087.

Michalski, R. S., Carbonell, J. G., and Mitchell, T. M. (1983). Machine Learning: An Artificial Intelligence Approach. Palo Alto, CA: Tioga Press.

Miller, G. A. (1956). The magical number seven, plus or minus two: Some limits on our capacity for processing information. Psychological Review, 63, 81-97.

Newell, A. (1980). Reasoning, problem solving and decision processes: The problem space as a fundamental category. In N. J. Nickerson (Ed.), Attention and Performance, VIII Hillsdale, NJ: Lawrence Erlbaum Associates.

Newell, A. and Rosenbloom, P. (1981). Mechanisms of skill acquisition and the law of practice. In J. R. Anderson (Ed.). Cognitive Skills and Their Acquisition Hillsdale, N. J.: Lawrence Erlbaum Associates.

Pirolli, P. L., and Anderson, J. R. (In Press). The role of learning from example in the acquisition of recursive programming skills. Canadian Journal of Psychology, . .

Rich, E. (1983). Artificial Intelligence. New York: McGraw Hill.

Rosenblatt, F. (1961). Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Washington, DC: Spartan.

Rosenbloom, R. S. (August, 1983). The chunking of goal hierarchies: A model of practice and stimulus-response compatibility. Technical Report CMU-CS-83-148, Computer Science Department, Carnegie-Mellon University.

Rosenbloom, P., and Newell, A. (1983). The chunking of goal hierarchies: A generalized model of practice. Machine Learning Workshop.

Rosenbloom, P. S., Laird, J. E., McDermott, J., and Newell, A. (unknown, 1984). R1-SOAR: An experiment in knowledge-intensive programming in a problem-solving architecture. Technical Report unknown, Computer Science Department, Carnegie-Mellon University.

Scott, P. D. (1983). Learning: The construction of a posteriori knowledge structures. AAAI.

Scott, P. D., and Vogt, R. C. (1983). Knowledge oriented learning. IJCAI.

Sutton, R. S., and Barto, A. G. (1981). Toward a modern theory of adaptive networks: Expectation and prediction. Psychological Review, 88, 135-170.

Sutton, R. S., and Barto, A. G. (1981). An adaptive network that constructs and uses an internal model of its world. Cognition and Brain Theory, 4, 217-246.

Van de Brug, A., and Rosenbloom, P. R1-Soar: Problem-spaces, search control and learning. Forthcoming technical report. Department of Computer Science, Carnegie-Mellon University.

Waltz, D. L. (1975). Understanding line drawings of scenes with shadows. In The Psychology of Computer Vision New York: McGraw-Hill.

Winston, P. H. (1984). Artificial Intelligence. Reading, MA: Addison-Wesley.

ANALOGICAL PROCESSES IN MACHINE LEARNING