NOTICE WARNING CONCERNING COPYRIGHT RESTRICTIONS:
The copyright law of the United States (title 17, U.S. Code) governs the making of photocopies or other reproductions of copyrighted material. Any copying of this document without permission of its author may be prohibited by law.

CMU-CS-84-108

APPROACHES TO MACHINE LEARNING

Pat Langley
Jaime G. Carbonell
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213
February 16 1984

Abstract

The field of *machine learning* strives to develop methods and techniques to automate the acquisition of new information, new skills, and new ways of organizing existing information. In this article, we review the major approaches to machine learning in symbolic domains, covering the tasks of learning concepts from examples, learning search methods, conceptual clustering, and language acquisition. We illustrate each of the basic approaches with paradigmatic examples.

To appear in *Journal of the American Society for Information Science.* Correspondence should be addressed to Pat Langley, The Robotics Institute, Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213.

This work was supported in part by Contract N00014-82-C-50767 from the Office of Naval Research.

1. Introduction: Why Machine Learning?

Learning is ubiquitous in intelligence, and it is natural that Artificial Intelligence (AI), as the science of intelligent behavior, be centrally concerned with learning. There are two clear reasons for this concern, one practical and one theoretical. With respect to the first, AI has now demonstrated the utility of expert systems, but these systems often require several man-years to construct. An expert system consists of a symbolic reasoning engine plus a large domain-specific knowledge base. Expert systems that rival or surpass human performance at very narrowly defined tasks are proliferating rapidly as AI is applied to new domains. A better understanding of learning methods would enable us to automate the acquisition of the domain-specific knowledge bases for new expert systems, and thus greatly speed the development of applied AI programs. On the theoretical side, expert systems are unattractive because they lack the *generality* that science requires of its theories and explanations. On this dimension, the study of learning may reveal general principles that apply across many different domains.

A third research goal is to emulate human learning mechanisms, and thus come to a better understanding of the cognitive processes that underly human knowledge and skill acquisition. In addition to improving our knowledge of human behavior, studying human learning may produce benefits for AI, since humans are the most flexible and robust (if slow) learning systems in existence. Hence, one objective of machine learning is to combine the capabilities of modern computers with the flexibility and resiliance of human cognition. As Simon [1] has pointed out, if learning could be automated and the results of that learning transferred directly to other machines which could further augment and refine the knowledge, one could accumulate expertise and wisdom in a way not possible by humans – each individual person must learn all relevant knowledge without benefit of a direct copying process. Thus, no single mind can hold the collective knowledge of the species.

2. A Historical Sketch

Historically, researchers have taken two approaches to machine learning. Numerical methods such as discriminant analysis have proven quite useful in perceptual domains, and have become associated with the paradigm known as *Pattern Recognition*. In contrast, Artificial Intelligence researchers have concentrated on symbolic learning methods,[1] which have proven useful in other domains. The symbolic approach to machine learning has received growing attention in recent years, and in this paper we review some of the main approaches that have been taken within this paradigm, and outline some of the work that remains to be done.

Within the symbolic learning paradigm, work first focused on learning simple concepts from examples. This originally involved artificial tasks similar to questions found in intelligence tests given to children, such as "What do all these pictures have in common?" and "Does this new picture belong in the group?" Such tasks involve the formulation of some hypothesis that predicts which instances should be classified as examples of the concept. Not too surprisingly, psychologists were among the active researchers in this early stage (e.g., Hunt, Marin and Stone [3]). Subsequent work focused on learning progressively more complex concepts, often requiring larger numbers of exemplars. Recent work has focused on more complex learning tasks, in which the learner does not rely so heavily on a tutor for instruction. For example, some of this research has focused on learning in the context of problem solving, while others have explored methods for learning by observation and discovery. Learning by analogy with existing plans or concepts has also received considerable attention.

In the following pages, we examine four categorical tasks that have been addressed in the machine learning literature – learning from examples, learning search heuristics, learning by observation, and language acquisition. These four representative tasks do not, by any means, cover all approaches to machine learning, but they should provide an illustrative sample of the issues, methods, and techniques of primary

[1] Samuel's [2] early checkers learning system was a notable exception to the later trend, relying mainly on a parameter fitting methods to improve performance.

concern to the field. In each case, we describe the task, consider the main approaches that have been employed, and identify some open problems in the area. As is typical in a survey article, we can only highlight the best known approaches and results in the area of machine learning, giving the reader a feeling for where the field as a whole has been and where it is heading. The serious reader is encouraged to digest other reviews of machine learning work by Mitchell [4], Dietterich and Michalski [5], and Michalski, Carbonell, and Mitchell [6].

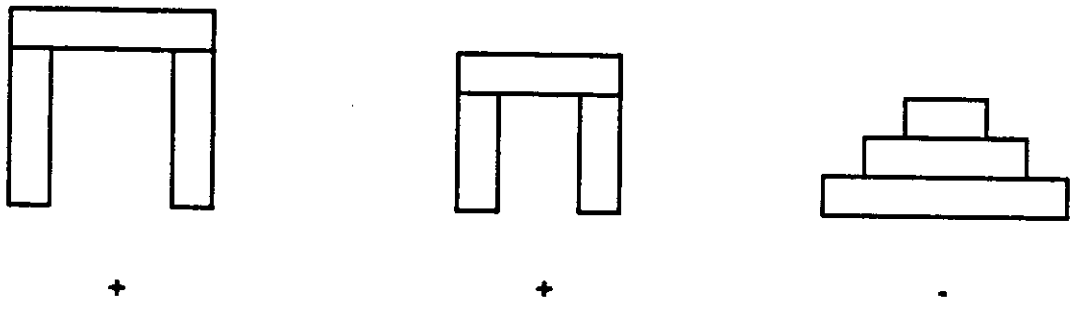

Figure 1. Positive and negative instances of "arch".

3. Learning Concepts From Examples

Methods for learning concepts from examples have received more attention than any other aspect of machine learning. The task appears straightforward: given a set of positive and negative instances of a concept, generate some rule or description that correctly identifies these and all future examples as instances or non-instances of the concept. However, despite its apparent simplicity, the approaches taken to solving this problem are nearly as numerous as the people who have worked on it. Below, we consider one approach to learning from examples, and then examine some of the dimensions along which different approaches to this problem vary. After this, we discuss some open issues in learning from examples that remain to be addressed.

3.1. An Example

Perhaps the best known research on learning from examples is Winston's [7] work on the "arch" concept. Figure 1 presents two examples of this concept and one counterexample that are very similar to those presented to Winston's system. Given these instances, one might conclude that

"An ARCH consists of two vertical blocks and one horizontal block".

This hypothesis covers both positive instances and excludes the negative one. Alternately, one *could* define "arch" as simply a union of all positive examples of ARCH ever encountered. However, the principles of brevity and generality preclude us from formulating such a definition, since we would like our concept to be as simple as possible, and for it to be able to predict new positive and negative instances. Given the first hypothesis, there is hope that a simple and general definition of "arch" will converge and help us recognize future examples of arches.

Now let us consider the two instances shown in Figure 2. Upon considering the positive instance, we realize that our concept of arch is too restrictive, since it excludes this instance. Therefore, we revise the concept to

"An ARCH consists of two vertical blocks and one horizontal *object*".

However, this new hypothesis covers some of the negative instances, suggesting that it is overly general in some respect. Revising the definition to exclude these instances, we might get:

"An ARCH consists of two vertical blocks *that do not touch* and a horizontal object *that rests atop both blocks.*

One can continue along these lines, gradually refining the concept to include all the positive but none of the negative examples. New positive instances that are not covered by the current hypothesis (errors of omission)

tell us that the concept being formulated is overly specific, while new negative examples that are covered by the hypothesis (errors of commission) tell us it is overly general. We have not been very specific about how the learner responds to these two situations, but we consider some of the alternatives below. All systems that learn from examples employ these two types of information, though we will see that they use them in quite different ways.

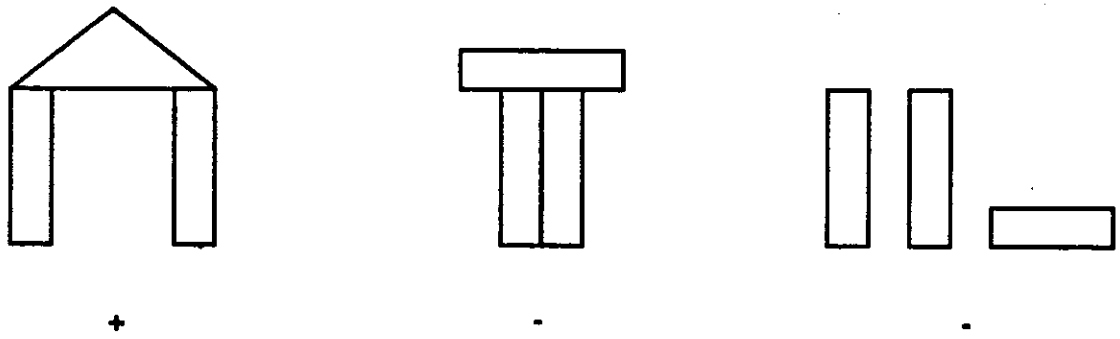

Figure 2. Additional positive and negative examples of "arch".

Lest the reader get the false impression that modifying an existing definition of a concept to accommodate a new positive or negative exemplar is always a simple process, we offer the positive and negative examples in Figure 3. We challenge the reader to devise an automated process that can modify "ARCH" to account for these examples. One insight that arises from these instances is that our concept of ARCH might involve some *functional* aspects as well as the structural ones we have focused on so far. We shall have more to say on this matter later.

3.2. The Dimensions of Learning

As Mitchell [4] and Dietterich and Michalski [5] have pointed out, all AI systems that learn from examples can be viewed as carrying out *search* through a space of possible concepts, represented as recognition rules or declarative descriptions. Moreover, this space is partially ordered[2] along the dimension of generality, and it is natural to use this partial ordering to organize the search process. However, at this point the similarity between systems ends. The first dimension of variation relates to the direction of the search through the rule space. *Discrimination-based* concept learning programs begin with very general rules and make them more specific until all instances can be correctly classified, while *generalization-based* systems begin with very specific rules and make them more general. Since these two methods approach the goal concept from different directions and more than one concept may be consistent with the data, the two methods need not arrive at the same answer. Dietterich and Michalski have called the rules learned by discrimination systems *discriminant* descriptions, and the rules learned by generalization systems *characteristic* descriptions. In general, the latter will be more specific than the former.

A second dimension of variation relates to the manner in which search through the rule space is controlled. Some systems carry out a *depth-first* search through the space of rules, while others employ a *breadth-first* search. In depth-first search, the learner focuses on one hypothesis at a time, generating more general or more specific versions of this (depending on the direction of the search) until it finds a description that accounts for the observed instances. In breadth-first search, the system considers a number of alternate hypotheses simultaneously, though many are eliminated as they fail to account for the data. Breadth-first search strategies have greater memory requirements than depth-first methods, but need never back up through the search space.

A third dimension of variation involves the manner in which data is handled. *All-at-once* systems

---

[2] It is this *partial* ordering that leads to branching, and thus to search. If the space were completely ordered, then the task of learning rules would be much simpler.

require all instances to be present at the outset of the learning process, while *incremental* systems deal with instances one at a time. The former tend to be more robust with respect to noise, while the latter are more plausible models of the human learning process. Finally, concept learning programs differ in the *operators* they use to move through the rule space. *Data-driven* systems incorporate instances in the generation of new hypotheses, while *enumerative* systems[3] use some other source of knowledge to generate states, and employ data only to *evaluate* these states.

\+ -

Figure 3. Still more positive and negative instances of "arch".

Given these four dimensions, we can determine that $2^4 = 16$ basic types of concept learning systems are possible, at least in principle. New researchers in machine learning might take as an exercise the task of classifying existing systems in terms of these dimensions, and brave individuals might attempt to develop a learning system that fills one of the unexplored combinations. In order to clarify the dimensions along which concept learning systems vary, let us examine two programs that lie at opposite ends of the spectrum on each dimension. For the sake of clarity, we will simplify certain aspects of the programs. The first is Quinlan's ID3 system [8], which has been tested in the domain of chess endgames, where the concepts to be learned are "lost in one move", "lost in two moves" and so forth. The second is Hayes-Roth and McDermott's SPROUTER [9] which has been tested on a number of complex relational instances like those in Figure 1 through 3.

ID3 represents concepts in terms of discrimination networks, as with the disjunctive concept **((large and red) or (blue and circle and small))**, shown in Figure 4. The system begins with only the top node of a network, and grows its decision tree one branch at a time. For instance, the system would first create the **(red or blue)** branch emanating from the top node. Next, it would create a branch coming from one of the new nodes, if necessary. The tree is grown downward, until terminal nodes are reached which contain only positive or negative instances. Thus, the system can be viewed as *discrimination-based*, moving from very general rules to very specific ones. At each point, it must select one attribute as more discriminating than others, so it carries out a *depth-first* search through the space of rules. ID3 is given a list of potentially relevant attributes by the programmer, so that in deciding which branch to create, it uses the data only in evaluating these attributes. The system is thus *enumerative* rather than data-driven in its search through the rule space. Finally, the program has all data available at the outset, so that it can use statistical analyses to distinguish discriminating attributes from undiscriminating ones; as a result, ID3 is an *all-at-once* concept learning system rather than an incremental one. The exact evaluation function Quinlan uses to direct search is based on information theory, but Hunt, Marin, and Stone [3] have used another evaluation function, and the exact function seems to be less important than the overall search organization.

Hayes-Roth and McDermott's SPROUTER [9] is historically interesting, since it was one of the first alternatives to Winston's early work on learning from examples. This program attempts to learn conjunctive

---

[3] Mitchell [4] has called these *generate and test* systems, while Dietterich and Michalski [5] have called them *model-driven* systems. However, AI associates the first term with systems that proceed exhaustively through a list of alternatives, and associates the second term with systems that rely on large amounts of domain-specific knowledge. We prefer the term *enumerative*, since a learning system can enumerate a set of alternate hypotheses at each stage in its search, without being either of these.

characteristic descriptions for a set of data, moving from a very specific initial hypothesis based on the first positive instance to more general rules as more instances are gathered. Thus, Hayes-Roth and McDermott's concept learning system is *generalization-based* rather than discrimination-based. SPROUTER also differs from ID3 in carrying out a *breadth-first* search through the rule space, rather than a depth-first search. With respect to positive instances, the system is *data-driven*, since it uses these instances to generate new hypotheses by finding common structures between them and the current hypotheses. However, the program is *enumerative* with respect to negative instances, since it uses these only to eliminate overly general hypotheses. Similarly, SPROUTER processes positive instances in an *incremental* fashion, reading them in one at a time and generalizing its hypotheses accordingly. However, it retains all negative instances in order to evaluate the resulting hypotheses, and processes them in an *all-at-once* manner. Thus, SPROUTER is something of a hybrid system in that it treats positive and negative instances in quite different ways.

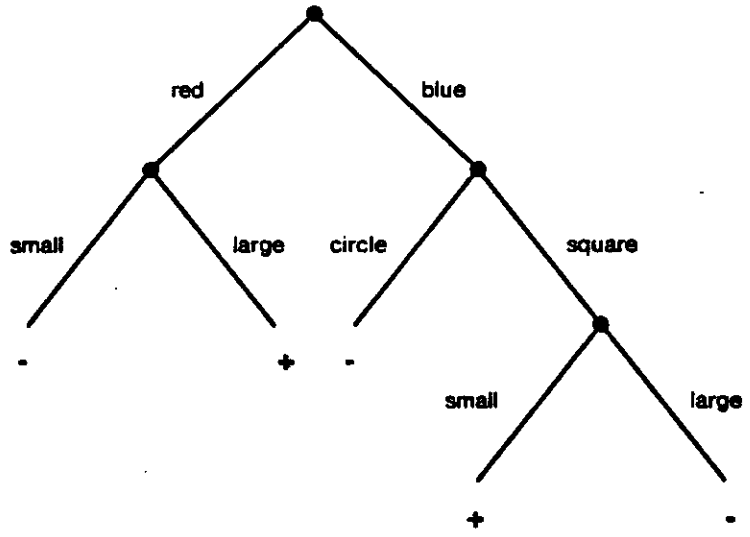

Figure 4. A concept expressed as a discrimination network.

3.3. Open Problems in Learning from Examples

A number of problems remain to be addressed with respect to learning from examples. Most of these relate to simplifying assumptions that have typically been made about the concept learning task. For instance, many researchers have assumed that no noise is present (i.e., all instances are correctly classified). However, there are many real-world situations in which no rule has perfect predictive power, and heuristic rules that are only usually correct must be employed. Some learning methods (such as Quinlan's) can be adapted to deal with noisy data sets, while others (such as Hayes-Roth and McDermott's) seem less adaptable. In any case, one direction for future work would be to identify those approaches that are robust with respect to noise, and to identify the reasons for their robustness. Most likely, tradeoffs exist between an ability to deal with noise and the number of instances required for learning, but it would be useful to know the exact nature of such relationships.

A related simplification is that the correct representation is known. If a learning system employs an incomplete or incorrect representation for its concepts, then it may be searching a rule space that does not contain the desired concept. One approach is to construct as good a rule as possible with the representation given; any system that can deal with noise can handle incomplete representations in this manner. A more interesting approach is one in which the system may improve its representation. This is equivalent to changing the space of rules one is searching, and on the surface at least, appears to be a much more challenging problem. Little work has been done in this area, but Utgoff [10] and Lenat [11], have made an interesting start on the problem.

A final simplifying assumption that nearly all concept learning researchers have made is that the concept to be acquired is *all or none*. In other words, an instance either is an example of the concept or it is not; there is no middle ground. However, almost none of our everyday concepts are like this. Some birds fit our bird stereotype better than others, and some chairs are nearer to the prototypical chair than others. (Is a Dodo a bird? Is a Platypus a better bird? If a person sits on a log, is it a chair? Is it a better chair if we add stubby legs and use a second log as a backrest?) Unfortunately, all of the existing concept learning systems rely fairly heavily on the sharp and unequivocal distinction between positive and negative instances, and it is not clear how they might be modified to deal with fuzzily-defined concepts such as birds and chairs. This is clearly a challenging direction for future research in machine learning.

The vast majority of work on learning concepts from examples has assumed that a number of instances must be available for successful learning to occur. However, recently a few machine learning researchers have taken a somewhat different approach. DeJong [12] has explored the use of causal information to determine the relevant features in a positive instance of a complex concept, such as *kidnapping*. By focusing on causal connections between events (such as the reason one would pay money to ensure another's safety), his system is able to formulate a plausible hypothesis on the basis of a single positive instance and *no* negative instances. Winston [13] has taken a similar approach to learning concepts such as *cup*. His system is presented with a *functional* description of a cup (e.g., that it must be capable of containing liquid, that it must be capable of being grasped) and a single positive instance of the concept. The system then uses its knowledge of the world to decide which structural features of the example allow the functional features to be satisfied, again using causal reasoning. These structural features are used in formulating the definition of the concept. Both approaches rely on causal information, and both relate this to some form of *functional* knowledge. This new approach promises concept learning systems that are much more efficient than the traditional *syntactic* methods, while retaining the generality of the earlier approaches. We expect to see much more work along these lines in the future.

4. Learning Search Methods

One of the central insights of AI is that intelligence involves the ability to solve problems by *searching* the space of possible actions and possible solutions, and to employ knowledge to constrain that search. In fact, one of the major differences between novices and experts in a complex domain is that the former must search extensively, while the latter use domain-specific heuristics to achieve their goal. In order to understand the nature of these heuristics, and how they may be learned, we must recall that search involves *states* and *operators*. A problem is stated in terms of an initial state and a goal, and operators are used to transform the initial state into one that satisfies the goal. Search arises when more than one operator can be applied to a given state, requiring consideration of the different alternatives. Of course, some constraints are usually given in terms of the *legal* conditions under which each operator may apply, but these constraints are seldom sufficient to eliminate search. In order to accomplish this, the learner must also acquire *heuristic* conditions on the operators. For example, Figure 5 presents a simple search tree involving two operators (O1 and O2), with the solution path shown in bold lines. If the problem solver knew the heuristic conditions on each operator, it would be able to generate the steps along the solution path without considering any of the other moves. The task of learning search methods involves determining these heuristic conditions.

The problem of learning search heuristics from experience can be divided into three steps. First, the system must generate the behavior upon which learning is based. Second, it must distinguish good behavior from bad behavior, and decide which part of the performance system was responsible for each. In other words, it must assign credit and blame to its various parts. Finally, the system must be able to modify its performance so that behavior will improve in the future. Different learning programs can vary on each of these three dimensions. For instance, though their initial performance component will carry out search, it may use depth-first search, breadth-first search, means-ends analysis, or any one of many other methods for directing the search process. Below we consider some alternative approaches to dealing with credit assignment and modification of the performance system.

Given this framework, the task of learning from examples is easily seen as a special case task of learning search heuristics, in which a single operator is involved and for which the solution path is but one step long. No true search control is necessary for the performance component, since feedback occurs as soon as a single "move" has been taken. Credit assignment is trivialized, since the responsible component is easily identified as the rule suggesting the "move". However, the modification problem remains significant, and in fact the task of learning from examples can be viewed as an artificial domain designed for studying the modification problem in isolation from other aspects of the learning process. In a similar fashion, the task of learning search heuristics can be seen as the general case of learning from examples, in which a different "concept" must be learned for each operator. Learning heuristics is considerably more difficult than learning from examples, since the learner must generate its own positive and negative instances, and since the credit assignment problem is nontrivial.

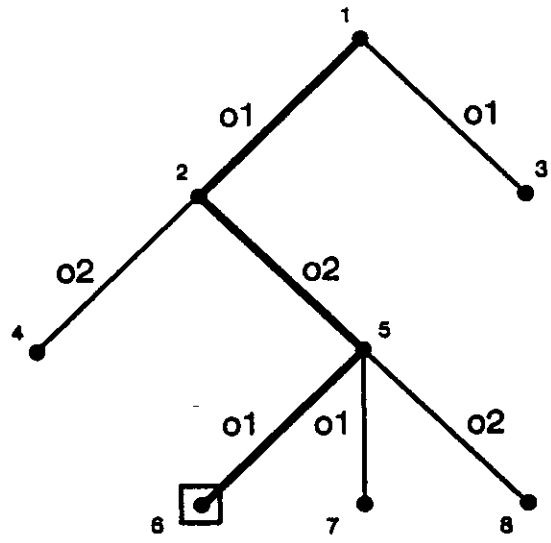

Figure 5. A simple search tree.

4.1. Assigning Credit and Blame

As we have discussed, if a learning system is to improve its behavior, it must decide which components of its performance system are responsible for desirable behavior, and which led to undesirable behavior. In general, assigning credit and blame can be difficult because many actions may be taken before knowledge of results is obtained, and any one of these actions may be responsible for the error. For instance, if the performance component is represented as a set of production rules, one must decide which of those rules led the system down an undesirable path. The problem of credit assignment is trivial in learning from examples since feedback is given as soon as a rule applies. However, the task is more formidable in the area of learning search heuristics, and recent progress in this area has resulted mainly from new insights about methods for assigning credit and blame.

The most straightforward of these approaches relies on waiting until a complete solution path to some problem has been found. Since moves along the solution path led the system toward the goal, one can infer that every move on this path is a positive instance of the rule that proposed the move. Similarly, moves that lead one step *off* of the solution path are likely candidates for negative instances of the rules that proposed them (though it is possible that alternate solutions starting with these moves were overlooked). Let us return to the problem space in Figure 5, with the solution path shown in bold. The move from state 1 to state 2 and from state 5 to state 6 would be classified as good instances of operator O1, while the move from state 2 to state 5 would be marked as a good instance of operator O2. In contrast, the moves from state 1 to state 3, and from state 5 to state 7 would be labeled as bad instances of O1, while the moves from state 2 to 4, and from state 5 to 8 would be noted as bad instances of O2. Moves more than one step off the solution path (these are

not shown in the figure) are not classified; since they were not responsible for the initial step away from the goal, they are not at fault. At least two recent strategy learning systems – Mitchell, Utgoff, and Banerji's LEX and Langley's SAGE – have used this heuristic as their basic method for assinging credit and blame to components of their performance systems. Other systems, including Brazdil's ELM [14] and Kibler and Porter's learning system [15], have used a similar technique, though their programs required the solution path to be provided by a benevolent tutor. Sleeman, Langley, and Mitchell [16] have discussed the advantages of this method for "learning from solution paths".

One limitation of this approach is that it encounters difficulty in domains involving very long solution paths and extensive problem spaces. Obviously, one cannot afford to search exhaustively in a domain such as chess. In response, some researchers have begun to examine other methods that assign credit and blame while the search process is still under way. These include such heuristics as noting loops and unnecessarily long paths, noting dead ends, and noting failure to progress towards the goal. Systems that incorporate such "learning while doing" methods include Anzai's HAPS [17], Ohlsson's UPL [18], and Langley's SAGE.2 [19]. Ironically, these systems have all been tested in simple puzzle-solving domains, where the "learning from solution paths" method is perfectly adequate. One obvious research project would involve applying these and other methods to more complex domains with long solutions and extensive search spaces.

4.2. Modifying the Performance System

Once credit and blame has been assigned to the moves made during the search process, one can modify the performance system so that it prefers desirable moves to undesirable ones. If the performance component is stated as a set of condition-action rules, then one can employ the same methods used in learning from examples. In other words, one can search the space of conditions, looking for some combination that will predict all positive instances but none of the negative instances. However, since multiple operators are involved, one must search a separate rule space for each operator. When one or more rules have been found for each operator, they can be used to direct search through the original problem space; if these rules are sufficiently specific, they will eliminate search entirely.

However, the task of learning search heuristics does place some constraints on the modification method that is employed. In particular, the learning system must be able to generate both positive and negative instances of its operators. This poses no problem for discrimination-based learning systems, since they begin with overly general move-proposing rules that lead naturally to search.[4] However, generalization-based systems are naturally conservative, preferring to make errors of omission rather than errors of commission. Such an approach works well if a tutor is present to provide positive and negative examples, but it encounters difficulties if a system must generate its own behavior. Ohlsson [18] has reported a mixed approach in which specific rules are preferred, but very general move-proposing rules are retained and used in cases where none of the specific rules are matched. However, in its pure form, generalization-based methods do not seem appropriate for heuristics learning.

4.3. Open Problems in Heuristics Learning

We have seen that heuristics learning can be viewed as the general case of learning from examples, and many of the open problems in this area are closely related to those for concept learning. For instance, one can imagine complex domains for which no perfect rules exist to direct the search process. In such cases, one might still be able to learn probabilistic rules that will lead search down the optimum path in *most* cases. This situation is closely related to the task of learning concepts from noisy data. Similarly, one can imagine attempting to learn search heuristics with an incorrect or incomplete representation. Finally, there are many domains in which some moves are better than others, but for which no absolute good or bad moves exist. As with learning from examples, most of the existing heuristics learning systems assume that "all or none" rules

---

[4] Neither does any problem arise for bi-directional approaches such as Mitchell's version space method, since these can use the general boundary in proposing moves.

exist. Thus, even if one could modify the credit assignment methods to deal with such continuous classifications, it is not clear how one would alter the modification components of these systems. Each of these problems have been largely ignored in the machine learning literature, but we expect to see more work on them in the future.

One recent departure from the *syntactic* methods we described above corresponds closely with the causal reasoning approach to learning from examples. Rather than relying on multiple solution paths to learn the heuristic conditions on a set of operators, Mitchell, Utgoff, and Banerji [20] have explored a method for gathering maximum information from a single solution path. This method involves reasoning backwards from the goal state, and determining which features of each previous state allowed the final operator in the sequence to apply. This method is used for each operator along the solution path, resulting in a macro-operator that is guaranteed to lead to the goal state. This method is very similar to that employed by Fikes, Hart, and Nilsson [21] in their early STRIPS system. Carbonell [22, 23] has explored a somewhat different but related approach in his work on problem solving by analogy. During its attempt to solve a problem, Carbonell's system retains information not only about the operators it has applied, but about the *reasons* they were applied. Upon coming to a new problem, the system determines if similar reasons hold there, and if so, attempts to solve the current problem by analogy with the previous one. Both Mitchell's and Carbonell's methods involve analyzing the solution path in order to take advantage of all the available information. As with learning from examples, this approach to learning search heuristics has definite advantages over the more syntactic approaches, and we expect it to become more popular in the future.

5. Learning from Observation: Conceptual Clustering

For the moment, let us return to the task of learning concepts from examples. Another of the simplifying assumptions made in this task is that the tutor provides the learner with explicit feedback by telling him whether an instance is an example of the concept to be learned. However, if we examine very young children, it is clear that they acquire concepts such as "dog" and "chair" long before they know the words for these classes. Similarly, scientists form classification schemes for animals, chemicals, and even galaxies with no one to guide them. Thus, it is clear that concept learning can occur without the presence of a benevolent tutor to provide feedback. The task of learning concepts in this way is sometimes called learning by *observation*.

5.1. The Conceptual Clustering Task

There are different types of learning by observation, but let us focus on what Michalski and Stepp [24] have called *conceptual clustering*, since this bears an interesting relation to learning from examples. In the conceptual clustering paradigm, one is presented with a set of objects or observations, each having an associated set of features. The goal is to divide this set into classes and subclasses, with similar objects being placed together. The result is a taxonomic tree similar to those used in biology for classifying organisms. In fact, biologists and statisticians have developed methods for generating such taxonomies from a set of observations. However, these methods (such as cluster analysis and numerical taxonomy) allow only numeric attributes (e.g., length of tail), while the conceptual clustering task also allows symbolic features.

Consider the set of objects shown in Figure 6, which vary on four binary attributes – size, shape, color, and thickness of the border. Only four out of the sixteen possible objects are observed, and the task is to divide these into disjoint groups that cover the observed objects, but that do not predict any of the unobserved ones. The classification tree shown in the figure satisfies these constraints while reflecting the regularities in the data. For instance, size and shape are the only features that are completely correlated, since all large objects are red, and all small objects are blue. Thus, these two features are ideal for dividing the observations into two groups at the highest level. However, within these groups finer distinctions can be made, and the features of border-thickness and shape are useful at this level.

This example points out two additional complexities in the conceptual clustering task over learning from examples. First, classification schemes nearly always involve disjunctive classes, and any successful

method must be able to handle them. (A conjunctive clustering task would be one in which only a single object was observed, and would not be very interesting.) Second, concepts must be learned at *multiple* levels. For instance, in the above example the "concept" **((large and red) or (small and blue))** must be generated at the first level, while the concepts **((thick and square) or (thin and circle))** and **((thick and circle) or (thin and square))** must be learned at the second level. Thus, the task of conceptual clustering can be viewed as a version of learning from examples that is more difficult along a number of dimensions – namely the absence of explicit feedback, the presence of disjuncts, and the need for concepts at multiple levels of description.

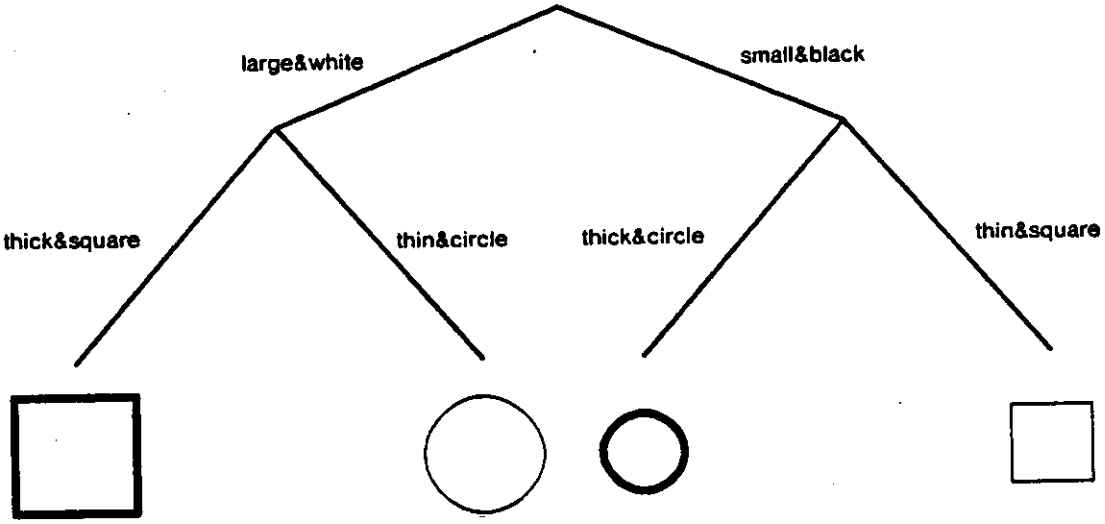

Figure 6. A simple classification tree.

5.2. Approaches to Conceptual Clustering

Michalski and Stepp's [24] approach to conceptual clustering takes advantage of this relationship. Basically, they employ a method for learning conjunctive concepts from examples to determine the branches (or concepts) at each level in the classification tree, starting at the top and working downward. In order to do this, their system must have a set of positive and negative instances. These are based on a small set of N randomly selected *seed* objects, and concepts are learned for each of these seed objects in such a manner that they do not cover any of the other seeds. Based on these concepts, a new set of seeds are produced which represent the central tendency of each concept, and the process is repeated, generating a revised set of concepts. This strategy continues until the seed objects stabilize, giving an optimal set of N disjoint classes. In addition, the system must decide *how many* classes should be used at each level in the classification tree. This is done by considering different numbers of seeds, and evaluating the resulting sets of concepts on their fit to the data. The best of these sets is used to add branches to the tree, and objects are sorted down the appropriate branches. The entire process is then repeated on each of these subsets of objects, in order to add lower level branches to the classification scheme.

As with learning from examples, approaches to conceptual clustering can vary along a number of dimensions. For instance, though Michalski and Stepp's method requires all data to be present at the outset, one can imagine systems that work in an incremental fashion. In fact, Lebowitz [25] has reported such an incremental system. These two systems also differ in the way they organize search through the space of classification trees. Both systems carry out a depth-first search through this space, starting at the top with more general classes and adding more specific subclasses later. However, since Michalski and Stepp's approach has all relevant data available at the outset, it can use this information to select the best branch at each point. In contrast, Lebowitz's system is sometimes forced to restructure a classification tree as new observations are made; this is equivalent to backing up through the space of classification trees, and trying an alternate path. This appears to be another case of the well-known AI tradeoff between knowledge and search: the more

knowledge that is available (in this case in the form of data), the less search is required (in this case through the space of classification trees).

A final dimension of variation involves the order in which the classification tree is constructed. Both Michalski and Stepp's and Lebowitz's approaches begin at the top of the tree and work downward. For example, given the objects in Figure 6, the distinction between large red objects and small blue objects would be made first, followed by the "finer" distinctions at lower levels in the tree. However, there is no reason why a taxonomic scheme could not be generated in the opposite order, classifying the most similar objects together first, and grouping the resulting classes afterwards. In fact, two systems that form conceptual clusters in this manner have been described in the AI literature. Wolff's [26] MK10 and SNPR [27] programs, which operate in the domain of grammar acquisition, form classes such as *noun*, *verb*, and *adjective* early in the learning process, and form more abstract classes in terms of these at a later time. Similarly, the GLAUBER program described by Langley, Zytkow, Bradshaw, and Simon [28] discovers regularities in chemical reactions first by defining classes such as *alkalis* and *metals*, and only later defines classes such as *bases* in terms of them. Hopefully, future work will reveal the advantages and disadvantages of different approaches to the conceptual clustering task.

5.3. Open Problems in Conceptual Clustering

Most of the existing conceptual clustering systems are designed to handle attribute-value representations. Thus, one direction for future research in this area would involve extending these approaches to deal with relational or structural information. In addition, the reader may recall that the task of learning from examples can be transformed into the conceptual clustering task by removing the simplifying assumption of explicit feedback. However, most work in conceptual clustering retains the assumption that the learned concepts are "all or none". Thus, a second direction for research would involve extending these methods, enabling them to learn inexact concepts such as dog or chair in which some features are more central than others. Since conceptual clustering methods do not rely on a strong distinction between positive and negative instances, this should be reasonably straightforward. It simply has not been a major focus of the researchers in this area.

A final research area relates to the importance of *function* in our everyday concepts. Nelson [29] has argued that children's very early concepts are often functional in nature. For example, a ball is something that one can bounce, and a chair is something that one can sit on. Only later, Nelson claims, are structural features added to these concepts. This suggests that a child's *goals* play an important role in the way he organizes his view of the world. Moreover, this ties in with Winston's approach to learning from examples, in which the learner uses a functional description to simplify the learning of structural descriptions. One can imagine a learning system that, starting with certain goals, formulated a set of function-based core concepts without using explicit feedback, and which then used Winston's method to add structural information. This would be a radically different approach to conceptual clustering, but one which appears to have considerable potential for modeling the human process of concept formation.

6. Language Acquisition

A fourth major area of machine learning research has dealt with the acquisition of language. In many ways, the literature on language learning stands apart from other work in the field. For instance, more of the researchers in this area have been concerned with modeling the human learning process than have workers in other areas of machine learning. In addition, relatively little contact has been made between work in this area and the work on concept learning and strategy learning. For this reason, and for lack of space, we will not attempt to cover AI approaches to language acquisition is as much detail as we have other areas. Rather, we will attempt to state the problem and provide a simple example. More detailed reviews of computational approaches to language learning can be found in Anderson [30], Pinker [31], and Langley [32].

Early research on language acquisition focused on inducing grammars to predict a set of sample sentences [33, 34]. More recently, most workers have reformulated the task in terms of learning a *mapping*

between a set of sentences and their meanings. Anderson [30] has argued that this situation is similar to that encountered by children, since early sample sentences generally refer to some situation or event present in the child's environment. Figure 7 presents such a sample sentence and its meaning. Some workers have focused on sentence generation (most of the psychological data concerns children's utterances), others have studied learning to *understand* sentences, and still others have been concerned with both issues. Some researchers have assumed that connections between concepts and their associated words are already known, while others attempt to learn this mapping along with the relation between meaning structures and grammatical structures.

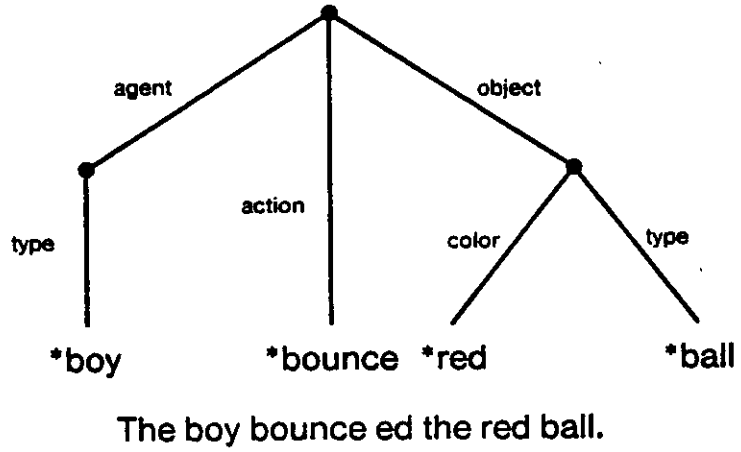

Figure 7. A simple sentence and its meaning.

In modeling language acquisition, the learning system is presented with a set of legal sentences and their associated meanings. The reader will recall that negative instances play an important role in learning from examples and learning search methods, and one would expect a similar situation here. Thus, the fact that only *legal* sentences are presented might be viewed as a serious problem for language learning systems. However, recall that the task is to learn a *mapping* between sentences and their meanings. This mapping is never carried out by a single rule, but rather by some *set* of rules. For a given sentence-meaning pair, some of these rules may apply correctly, some may fail to apply when they should, and still others may apply when they should not. The latter two cases correspond to positive instances (errors of omission) and negative instances (errors of commission), respectively. Thus, at the appropriate level of analysis, both positive and negative instances do arise in the language learning task.

For example, in order to describe the meaning structure in Figure 7, the learner must have some rule for saying the word "the", another for "boy", another for "bounce", perhaps another for "ed", and so forth. Each of these rules may be overly specific or overly general, leading to errors of omission or commission. In terms of finding the correct conditions on such rules, the language learning task is more difficult than the others we have examined, since arbitrary exceptions often occur. Thus, the learner may decide to say "ed" after the word for any past action, and then discover the numerous exceptions to this rule. In fact, young children often produce overgeneralizations like "runned" and "hitted", though they eventually recover from these problems.[5] In addition, in order to organize its knowledge, the language learner may also need intermediate level rules for describing the agent of an event, the action, and so on. This further complicates the learning task, since errors can occur at different levels in such hierarchical schemes, making credit and blame difficult to assign.

In summary, the language acquisition task involves learning a mapping between sentences and their meanings. In turn, this provides the equivalent of positive and negative instances, letting the learner acquire

[5] Selfridge [35] has developed a computational model of this process of overgeneralization and recovery.

rules in much the same fashion as in other areas of machine learning. However, the task is more difficult than most in that it often involves arbitrary exceptions, as well as intermediate level rules for which one can never attain complete feedback. The language acquisition task is complex enough that we cannot hope to cover it adequately here; however, this brief overview may have given the reader some idea of its relation to, and differences from, other areas of machine learning.

7. Conclusions

In this paper, we examined some of the task domains studied by researchers in machine learning – learning from examples, learning search methods, conceptual clustering, and language acquisition – and considered some relations between those domains. A number of common threads emerged from this examination. One of these was the notion of search through a space of rules, and various methods for directing the search through this space. Another was the idea that learning from examples can be viewed as a simpler version of the more complex tasks of learning search heuristics and conceptual clustering, in that credit assignment is simplified and feedback is present. We found that some areas, such as data-driven approaches to learning from examples, appear to relatively well understood, while in other areas, such as learning during the search process, much work remains to be done. In each of the domains we examined, we found a number of open issues that remain to be explored. Among the most exciting of these was the potential for using functional or causal information in directing the learning process.

In addition to those aspects of machine learning we have covered, ongoing research is addressing a number of exciting topics we have not had the space to discuss. One of these involves attempts to automate the process of scientific discovery [11, 36]; ultimately this may lead to advisory systems that aid scientists in their research. Another area that has received considerable attention recently concerns methods for reasoning by analogy with prior experience [23]; systems that solve problems in this manner could be considerably more flexible than existing AI programs. Another research focus is learning from instruction, in which the system acquires knowledge directly from a textbook or tutor. This is probably the most immediately applicable of all machine learning methods, due to recent advances in natural language processing. Machine learning, despite its recent emergence, has developed nearly as many fascinating problems as researchers to pursue those problems. As a result, more colleagues are always welcome, and we hope we have communicated some of the excitement in this rapidly developing field to the reader.

8. References

[1] Simon, H. A.
Why should machines learn?
In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (editors), *Machine Learning: An Artificial Intelligence Approach.* Tioga Press, Palo Alto, CA, 1983.

[2] Samuel, A. L.
Some studies in machine learning using the game of checkers.
IBM Journal of Research and Development 3:210-229, 1959.

[3] Hunt, E. B., Marin, J., and Stone, P. J.
Experiments in Induction.
Academic Press, New York, 1966.

[4] Mitchell, T. M.
Generalization as search.
Artificial Intelligence 18:203-226, 1982.

[5] Dietterich, T. G and Michalski, R. S.
A comparative review of selected methods for learning from examples.
In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (editors), *Machine Learning: An Artificial Intelligence Approach.* Tioga Press, Palo Alto, CA, 1983.

[6] Carbonell, J. G., Michalski, R. S., and Mitchell, T. M.
Machine learning: A historical and methodological analysis.
In *AI Magazine*, pages 69-79. Fall, 1983.

[7] Winston, P. H.
Learning structural descriptions from examples.
Technical Report AI-TR-231, Massachusetts Institute of Technology, 1970.

[8] Quinlan, R.
Learning efficient classification procedures and their application to chess end games.
In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (editors), *Machine Learning: An Artificial Intelligence Approach.* Tioga Press, Palo Alto, CA, 1983.

[9] Hayes-Roth, F. and McDermott, J.
An interference matching technique for inducing abstractions.
Communications of the ACM 21:401-410, 1978.

[10] Utgoff, P. E. and Mitchell, T. M.
Adjusting bias in concept learning.
In *Proceedings of the International Machine Learning Workshop*, pages 105-109. 1983.

[11] Lenat, D. B.
The role of heuristics in learning by discovery: Three case studies.
In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (editors), *Machine Learning: An Artificial Intelligence Approach.* Tioga Press, Palo Alto, CA, 1983.

[12] DeJong, G.
An approach to learning from observation.
In *Proceedings of the International Machine Learning Workshop*, pages 171-176. 1983.

[13] Winston, P. H.
Learning by augmenting rules and accumulating censors.
In *Proceedings of the International Machine Learning Workshop*, pages 2-11. 1983.

[14] Brazdil, P.
Experimental learning model.
In *Proceedings of the Third AISB/GI Conference*, pages 46-50. 1978.

[15] Kibler, D. and Porter, B.
Perturbation: A means for guiding generalization.
In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 415-418. 1983.

[16] Sleeman, D., Langley, P., and Mitchell, T.
Learning from solution paths: An approach to the credit assignment problem.
In *AI Magazine*, pages 48-52. Spring, 1982.

[17] Anzai, Y.
Learning strategies by computer.
In *Proceedings of the Canadian Society for Computational Studies of Intelligence*, pages 181-190. 1978.

[18] Ohlsson, S.
A constrained mechanism for procedural learning.
In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 426-428. 1983.

[19] Langley, P.
Learning effective search heuristics.
In *Proceedings of the Eighth International Joint Conference on Artificial Intelligence*, pages 419-421. 1983.

[20] Mitchell, T. M., Utgoff, P., and Banerji, R. B.
Learning problem solving heuristics by experimentation.
In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (editors), *Machine Learning: An Artificial Intelligence Approach*. Tioga Press, Palo Alto, CA, 1983.

[21] Fikes, R. E., Hart, P. E., and Nilsson, N. J.
Learning and executing generalized robot plans.
Artificial Intelligence 3:251-288, 1972.

[22] Carbonell, J. G.
Learning by analogy: Formulating and generalizing plans from past experience.
In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (editors), *Machine Learning: An Artificial Intelligence Approach*. Tioga Press, Palo Alto, CA, 1983.

[23] Carbonell, J. G.
Derivational analogy in problem solving and knowledge acquisition.
In *Proceedings of the International Machine Learning Workshop*, pages 12-18. 1983.

[24] Michalski, R. S. and Stepp, R. E.
Learning from observation: Conceptual clustering.
In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (editors), *Machine Learning: An Artificial Intelligence Approach*. Tioga Press, Palo Alto, CA, 1983.

[25] Lebowitz, M.
Concept learning in a rich input domain.
In *Proceedings of the International Machine Learning Workshop*, pages 177-182. 1983.

[26] Wolff, J. G.
Grammar discovery as data compression.
In *Proceedings of the AISB/GI Conference on Artificial Intelligence*, pages 375-379. 1978.

[27] Wolff, J. G.
Data compression, generalisation, and overgeneralisation in an evolving theory of language development.
In *Proceedings of the AISB-80 Conference on Artificial Intelligence*, pages Wolff 1-10. 1980.

[28] Langley, P., Zytkow, J., Bradshaw, G. and Simon, H. A.
Mechanisms for qualitative and quantitative discovery.
In *Proceedings of the International Machine Learning Workshop*, pages 121-132. 1983.

[29] Nelson, K.
Some evidence for the cognitive primacy of categorization and its functional basis.
Merrill-Palmer Quarterly of Behavior and Development 19:21-39, 1973.

[30] Anderson, J. R.
Induction of augmented transition networks.
Cognitive Science 1:125-157, 1977.

[31] Pinker, S.
Formal models of language learning.
Cognition 7:217-283, 1979.

[32] Langley, P.
Language acquisition through error recovery.
Cognition and Brain Theory 5:211-255, 1982.

[33] Solomonoff, R.
A new method for discovering the grammars of phrase structure languages.
In *Proceedings of the International Conference on Information Processing*. UNESCO, 1959.

[34] Feldman, J. A., Gips, J., Horning, J. J., and Reder, S.
Grammatical complexity and inference.
Technical Report No. CS 125, Computer Science Department, Stanford University, 1969.

[35] Selfridge, M.
Why do children say "goed"? – A computer model of child generation.
In *Proceedings of the Third Conference of the Cognitive Science Society*, pages 131-132. 1981.

[36] Langley, P., Bradshaw, G., and Simon, H. A.
Rediscovering chemistry with the BACON system.
In R. S. Michalski, J. G. Carbonell, and T. M. Mitchell (editors), *Machine Learning: An Artificial Intelligence Approach*. Tioga Press, Palo Alto, CA, 1983.