

LEARNING STRATEGIES AND AUTOMATED
KNOWLEDGE ACQUISITION: AN OVERVIEW

by

Ryszard S. Michalski

Report #926, Department of Computer Science, University of Illinois, Urbana, Illinois, November, 1984.

File No. UIUCDCS-F-84-926

**LEARNING STRATEGIES
AND
AUTOMATED KNOWLEDGE ACQUISITION:
An Overview**

by

Ryszard S. Michalski

**Artificial-Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139†**

ISG 84-6

November 1984

†On leave from the Department of Computer Science, University of Illinois, Urbana, IL 61801.

This research was supported in part by the National Science Foundation under grant DCR 84-06801 and in part by the Office of Naval Research under grant No. N00014-82-K-0186. Support for MIT's Artificial-Intelligence Research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-C-0505.

LEARNING STRATEGIES
AND
AUTOMATED KNOWLEDGE ACQUISITION:
An Overview

by

Ryszard S. Michalski

Artificial-Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139†

ABSTRACT

Fundamental learning strategies are discussed in the context of knowledge acquisition for expert systems. Special consideration is given to inductive learning strategies: learning from examples (concept acquisition) and learning from observation (concept formation without teacher). A specific form of learning from observation is discussed in detail, namely, *conceptual clustering*, i.e., structuring given observations into a hierarchy of conceptual categories.

An inductive learning system generates knowledge by drawing inductive inferences from the facts or observations under the guidance of *background knowledge*. The background knowledge defines goals of learning, the criteria for evaluating generated hypotheses, the properties of initial attributes and relations characterizing observed events, and inference rules and heuristics for deriving higher level descriptive concepts.

1. INTRODUCTION

Learning ability is no doubt central to human intelligence. This ability permits us to adapt to the changing environment, to develop a great variety of skills, and to acquire expertise in an almost unlimited number of specific domains. The human ability to learn is truly remarkable: people are capable of learning from information carried by multiple physical media and expressed in an unbounded variety of forms. The information can be stated at many different levels of abstraction, with different degrees of precision, with or without errors, and with different degrees of relevancy to the knowledge ultimately acquired.

†On leave from the Department of Computer Science, University of Illinois, Urbana, IL 61801.

Implanting learning capabilities in machines is one of the central goals of Artificial Intelligence. It is the subject of a new field of Machine Learning. Due to the enormous complexity of learning processes, development of general-purpose, versatile learning systems is a long-term goal. With the development of expert systems, however, implementing some forms of machine learning has become an urgent task, even if the forms of such implementation are very limited.

The urgency of this task stems from an explosive growth of interest and social need to develop expert systems for many different applications, from medicine and agriculture to law, education and computer design. Expert systems are computer programs (or devices) that simulate the expertise of a human expert in solving problems in some specific domain. They are capable of conducting formal inference on their knowledge base in the interaction with the external information provided by a user, in order to provide a solution to a problem or an advice in decision making. Examples of specific expert systems include:

- DENDRAL (developed at Stanford University) for determining the molecular structure of organic compounds from mass spectrograms.
- MACSYMA (developed at MIT) which serves as a general mathematical aids system (e.g., for symbolic integration, simplification of mathematical expressions, etc.).
- R1 (developed at Carnegie-Mellon University) for determining configurations of VAX computer systems.
- INTERNIST (developed at the University of Pittsburgh) for diagnosing diseases of interest in internal medicine.
- PLANT/ds and PLANT/cd (developed at the University of Illinois) - two related agricultural expert systems, the first for diagnosing soybean diseases, and the second for predicting black cutworm damage to corn.

The major component of an expert system is its knowledge base, i.e., formally represented knowledge in the given domain of application. Building such a knowledge base is typically done as a cooperative effort between a "knowledge engineer" and a domain expert. The knowledge engineer conducts interviews with an expert and codifies the expert's knowledge in some knowledge representation system.

Typically, such a system consists of production rules (condition-action rules) or a semantic network (a graph whose nodes represent concepts or conditions and whose links represent relations between the

concepts). These two forms of knowledge representation have special appeal, because of their comprehensibility and relative ease of use for implementing inference processes. For some applications, however, these forms may not be sufficient. For example, in system PLANT/cd, a large part of the domain knowledge is encoded as a set of procedures that form a simulation model of the growth of corn versus the growth of black cutworms (Boulanger, 1983).

Encoding expert knowledge into a system is a time-consuming, difficult process that is prone to error. For this reason, knowledge acquisition is a "bottleneck" in the development of expert systems. The importance of machine learning research to further progress in the development of expert systems has recently been indicated in a report on the state-of-the-art in Artificial Intelligence (Waltz et al., 1983). The process of knowledge acquisition can be simplified by applying interactive programming aids for developing and debugging rule bases. Such an aid is provided, for example, by the system TEIRESIAS, developed by Davis (1978).

In this paper we review basic strategies of learning and discuss them in the context of automated knowledge acquisition. We specifically concentrate on knowledge acquisition through inductive learning, which encompasses two strategies: learning from examples, and learning by observation and discovery.

2. FUNDAMENTAL LEARNING STRATEGIES

The knowledge acquisition process can be greatly simplified if an expert system can learn decision rules directly from examples of decisions made by human experts, or from its own errors via feedback. This type of learning strategy is called *learning from examples* (or *concept acquisition*). It has been studied widely in the last ten years or so, and many important results have been obtained (e.g., Winston, 1970; Michalski, 1972; Lenat, 1976; Mitchell, 1978; Buchanan et al. 1979; Pao and Hu, 1982, Hu and Pao, 1982; Dietterich and Michalski, 1983; Langley, Bradshaw and Simon, 1983; Michalski, 1983; Rendell, 1983b; Winston, 1983).

Learning from examples is one of several fundamental learning strategies. The identification of these

strategies stems from viewing a learning system as an inference system. Thus, the strategies are distinguished on the basis of the type and amount of inference the learning system (human or machine) must perform on the information provided, in order to derive the desired knowledge. At one extreme, the system performs no inference, but directly accepts and uses the information given to it (or built into it). At the other extreme, the system performs a complex, often indirect, random search-based inference that on occasion leads to discovery of new knowledge. The following learning strategies are important points along the above spectrum:

A. *Direct Implanting of Knowledge*

As mentioned above, this strategy, also called *rote learning*, requires no inference or information processing on the part of the learner. It includes learning by being constructed, programmed, or modified by an external agent (designer, programmer, etc.). This strategy is a widely used method for providing knowledge to a computer system: we incorporate knowledge into its hardware, we program it, and we build databases for all kinds of applications. Although building databases is not typically considered as machine learning, it can be considered this way in our context. Some databases go beyond this learning strategy, as they can perform some limited inference, usually mathematical or statistical.

B. *Learning from Instruction*

In this form of learning, also called *learning by being told*, a learner transforms the knowledge from the input language to an internally-usable representation and integrates it with prior knowledge for effective retrieval and use. This is the most widely used strategy of human learning: learning from teachers, books, publications, exhibits, displays, and similar sources. A machine version of this strategy is a system capable of accepting instruction or advice and applying the learned knowledge effectively to different tasks. Simple versions of this strategy constitute the basic method for providing knowledge to expert systems today (e.g., Davis, 1978, Hass and Hendrix, 1983).

C. *Learning by Deductive Inference*

A learning system that uses this strategy not only translates the knowledge from the input

language, but also conducts deductive (truth-preserving) inference on it. This inference permits the system to restructure given knowledge into more useful or more effective forms, or to determine important consequences of the knowledge. For example, given a set of numbers: 1, 2, 6, 24, 120, 720 we can represent them equivalently as $n!$, $n=1..6$. An example of such a system is an automatic theorem prover capable of storing theorems that it has already proved, and then using the stored theorems to prove new ones.

D. *Learning by Analogy*

This strategy involves transforming or extending existing knowledge (or skill) applicable in one domain to perform a similar task in another domain. For example, the learning-by-analogy strategy might be applied to learn water skiing when a person already knows snow skiing. Learning by analogy requires a greater amount of inference on the part of the learner than does learning from instruction. Relevant knowledge or skill must be retrieved from the memory and appropriately transformed to be applicable in a new situation or to a new problem. Examples of systems capable of learning by analogy are described by Carbonell (1983), Winston (1984) and Burstein (1984).

E. *Learning from Examples*

Given a set of examples and (optionally) counter-examples of a concept, the learner induces a general concept description. The amount of inference performed by the learner is greater than in learning by analogy, because the system does not use prior knowledge from which it can derive the desired knowledge by appropriate transformations.

Learning from examples, also called *concept acquisition*, can be a one-step (batch) process or a multi-step (incremental) process. In the batch case, all examples are presented at once. In incremental learning, examples (positive or negative) are introduced one-by-one or in small groups; the learner forms one or more tentative hypotheses consistent with the data at a given step, and subsequently refines the hypotheses after considering new examples. The latter strategy is commonly used in human learning.

Examples of a concept may be provided by a human teacher or by another module of the system. They can be generated by a deliberate effort of a teacher, or by a random, heuristic or exhaustive search through a space of operators acting upon given situations. If an operator produces a desired result, then we have an positive example, otherwise a negative example. The inductive learning system then generalizes these examples to form general decision rules or control heuristics.

When a system determines examples by a search or other active effort, we have a form of learning called *learning by experimentation*. Such a method was used, for example, in the LEX symbolic integration learning system (Mitchell, Utgoff and Banerji, 1983).

F. *Learning by Observation and Discovery*

This "learning without teacher" strategy includes a variety of processes, such as creating classifications of given observations, forming a theory to explain a given phenomenon, or discovering new relationships and laws. The learner is not provided with a set of instances exemplifying a concept, nor given access to an oracle (or teacher) who can classify internally-generated instances as positive or negative. Also, rather than concentrating attention on a single concept at a time, the learner may have to deal with observations that represent several concepts. This adds a new difficulty, namely solving the focus-of-attention problem, a problem of deciding how to manage the available time and resources in acquiring several concepts at once.

Learning from observation can be subclassified according to the *degree of interaction* between the learner and the external environment. Two basic cases can be distinguished:

- (a) *passive observation*, where the learner builds a description of a given set of observations. For example, such a description may be a taxonomy of the observations (e.g., Michalski and Stepp, 1983), or an empirical law characterizing the observations, as in the BACON system (Langley, Simon & Bradshaw, 1983).

- (b) *active experimentation*, where the learner makes changes in the environment and observes the results of those changes (e.g., Rendell, 1983a). The changes may be random or dynamically controlled by some heuristic criteria. The choice of tasks and directions in the experimentation can be controlled by criteria such as utility or *interestingness* (e.g., Lenat, 1976).

The learning strategies, A to F, are presented above in order of increasing amounts of effort on the part of the learner and decreasing amounts of effort on the part of the teacher. This order also reflects the increasing difficulty of constructing a learning system capable of using the given learning strategy.

In human learning, the order of the strategies also corresponds to the increasing confidence in the acquired knowledge. We all know that when we are given a general rule (a directive, a theory) without explanation and examples supporting it, our confidence in it will not be very high; it will directly depend on the trust we have in the giver. Our confidence in a rule will be much greater if we can try the rule on examples, and still greater, if we develop the rule through our own experience. On the other hand, it is much harder to determine correct knowledge by induction than by being told. This means that the higher the strategy, the higher is the risk of arriving at incorrect knowledge.

The above relation holds under the assumption that the input information to the learner has no errors. That is, e.g., that in learning by instruction we have a "perfect" teacher. Because this assumption may not hold in reality, the learning by instruction strategy is also associated with a risk of acquiring incorrect knowledge. This explains the emphasis educators place on providing students with best teachers.

The higher the learning strategy, the more inference, and thus the more cost and effort (on the part of the learner) is involved in deriving the desired knowledge. It is much easier to be told how to solve a problem than to discover the solution on one's own. Learning by instruction works well when there is a teacher who knows the algorithm or the definition of the concept, and is capable of articulating it in the language of the learner. But when such a teacher is not available, another strategy must be used. For example, it is difficult to define the concept of a chair, or the shape of the characters of the alphabet.

When we teach children these concepts, therefore, we teach them by showing examples.

In many situations, the best way to explain a concept is to relate it to a similar concept and show the similarity and differences. This is learning by analogy. This form of learning requires learner to know the concepts which are referenced. The more knowledgeable a learner is, the more potentially effective learning by analogy is. It seems reasonable to infer that learning by analogy is more effective with adults than with children.

There are lessons for machine knowledge acquisition to be drawn from the above considerations. One is that if we know precisely how to solve a problem, we should tell the computer the solution directly (i.e., program it). In this case, to teach the system by instruction is simpler and better than to engage it in an inductive learning process. Such learning will be facilitated by having an appropriate knowledge representation language. Because there are many areas in which we know solutions and can define relevant concepts, this strategy has wide applications. Therefore, the development of appropriate knowledge representation languages (both general and specific to a given domain) constitutes a major research area.

There are, however, many areas of application where precise algorithms are unknown or difficult to construct. Examples of such areas are medical or agricultural diagnosis, visual pattern recognition, voice recognition, machine design, robot control in automated assembly lines, and many others. Also, people often have difficulties in articulating their expertise, even when they know well how to perform a given task. In all such cases, applying an analogical or inductive machine learning strategy seems quite desirable.

A prerequisite for applying an analogical learning strategy is that the system possess a knowledge base of concepts and solutions to problems that are similar to the ones the system will be solving. In addition, the system must be able to recognize the similarity between any new problem and a problem for which it already knows a solution, and must be able to modify the known solution appropriately. These are

difficult and complex operations. For that reason it is often easier for the system to start from scratch than to modify a known solution. This phenomenon is well known to programmers, who sometimes prefer to write a program anew rather than to modify an existing program. An interesting problem that arises here is how to decide which way is better in any given situation. Analogical inference can be viewed as a combination of inductive and deductive inference: the first one determines the analogy between problems and formulates appropriate knowledge transformations, and the second one performs these transformations on the known solution.

In the remainder of the paper we will discuss in greater detail the inductive learning strategy. We will start by giving a more precise meaning to this type of learning.

3. INDUCTIVE LEARNING: General Description

Inductive learning is a process of acquiring knowledge by drawing inductive inferences from teacher- or environment-provided facts. This process involves operations of generalizing, transforming, correcting and refining knowledge representations in order to accommodate given facts and satisfy various additional criteria. It is the fundamental strategy through which we learn about our environment. This strategy has one basic weakness: except for special cases, the acquired knowledge cannot, in principle, be completely validated. This is so because inductive inference produces hypotheses with a potentially infinite number of consequences, while only a finite number of confirming tests can be performed. This is a well-known predicament of induction, as observed by Hume in the 18th century.

Also, given any set of facts, one can potentially generate an infinite number of hypotheses in explanation of these facts. Therefore, in order to perform inductive inference one needs some additional knowledge (*background knowledge*) to constrain the possibilities and guide the inference process toward one or a few most *plausible* hypotheses. This background knowledge must include the goals of learning, criteria for deciding the preference among candidate hypotheses, the methods for interpreting the observations, and the knowledge representation language with corresponding inference rules for manipulating representations in this language, as well as the knowledge of the domain of inquiry.

There are two aspects of inductive inference: the generation of plausible hypotheses, and their confirmation. Only the first is of significant interest to machine inductive learning. The second one (impossible in principle) is considered of lesser importance, because it is assumed that the generated hypotheses will be judged by human experts and tested by known methods of deductive inference and statistical confirmation.

Bearing in mind these considerations, let us formulate a general paradigm of inductive inference:

Given:

- (a) *observational statements (facts), F*, that represent specific knowledge about some objects, situations or processes,
- (b) *a tentative inductive assertion (which may be null),*
- (c) *background knowledge* that defines the goal of inference, the preference criterion for choosing plausible hypotheses, specifies assumptions and constraints imposed on the observational statements and the candidate inductive assertions, and supplies any relevant problem domain knowledge.

Find:

an inductive assertion (hypothesis), H, that tautologically implies the observational statements and satisfies the background knowledge.

An hypothesis H tautologically implies a set of facts F , if F is a logical consequence of H , that is the implication $H \Rightarrow F$ holds under all interpretations. Since an infinite number of assertions H can satisfy such an implication, a preference criterion is used to reduce the choice to one hypothesis or a few most preferable ones. Such a criterion may require, for instance, that the hypothesis be the shortest or the most economical description of all given facts.

Inductive learning programs already play an important role in the acquisition of knowledge for some expert systems. In some relatively simple domains they can determine decision rules by induction from

examples of decisions made by experts. This form of knowledge acquisition relieves the expert from the tedious task of defining rules himself. Moreover, it requires the expert to do what he can do best: make decisions. Experts are typically not trained to analyze and explain to others their decision making processes, especially if they must express them in a formal way; therefore, such tasks are usually difficult for them to perform.

A less direct yet important application of inductive learning is to the refinement of knowledge bases initially developed by human experts. Here, inductive learning programs together with other supporting software can be used to detect and rectify inconsistencies, to remove undesirable redundancies, to cover gaps or to re-express the given rules in a simpler way (e.g., Reinke, 1984). Also, starting with initial human expert-based rules, an inductive learning program can improve these rules through feedback in the form of an evaluation of the results of the expert system's decisions.

Another use for inductive learning is to generate meaningful classifications of given sets of data, or to organize the sets of data (e.g., collections of rules) into a structure of conceptually simple components (Michalski and Stepp, 1983). We will illustrate this application by an example in section 5.

Most of the above applications have already been tried successfully on some relatively simple problems. Current research tries to extend the machine learning techniques in a number of directions: using richer knowledge representation languages, exploring constraints of a domain to control generalization, constructing multi-level concept descriptions, automating the process of generating new attributes and operators (the so called *constructive induction*, the *new term problem*), coping with the uncertainty of data, integrating different learning strategies, and so on.

As mentioned above, we can distinguish between two types of inductive learning: learning from examples and learning by observation and discovery. We will now discuss these two types of learning in greater detail.

4. LEARNING FROM EXAMPLES

Within the category of learning from examples we can distinguish two major types: *instance-to-class* generalization and *part-to-whole* generalization. In the instance-to-class generalization, given are independent instances (examples) of some class of objects and the task is to induce a general description of the class. The instances can be representations of physical objects, sounds, images, actions, processes, abstract concepts, etc. Most research on learning from examples is concerned with this type of problem. For example, such research includes learning descriptions of block structures (e.g., Winston, 1977) or automatically inducing diagnostic rules for soybean diseases (Michalski and Chilausky, 1980). For a review of methods of such generalization see (Dietterich and Michalski, 1983).

In part-to-whole generalization, given are only selected parts of an object (a scene, a situation, a process) and the task is to hypothesize a description of the whole object. A simple example of this type of problem is to determine a rule characterizing a sequence of objects (or a process) from seeing only a part of the sequence (or process). A specific case of such a problem occurs in the card game Eleusis, where players are supposed to discover a "secret" rule governing a sequence of cards. A computer program capable of discovering such rules has been described by Dietterich and Michalski (1983).

In learning from examples, facts can be viewed as implications of the form

$$Event ::> Class$$

where *event* is a description of some object or situation and *class* represents a decision (class) to be assigned to this object or situation. (We denote the implication between a fact or pattern, and the class associated with it by the symbol " $::>$ ", in order to distinguish it from the ordinary implication, denoted " \Rightarrow ".) The result of learning is a rule

$$Pattern ::> Class$$

where *Pattern* is an expression in some formal language describing all events that belong to the given *Class*, and no events that do not belong to this class. When an unknown event satisfies the *Pattern* then it is assigned to *Class*.

The pattern description can be expressed in a many forms, e.g., a propositional or predicate logic expression, a decision tree, a formal grammar, a semantic network, a frame, a script, a computer program. The complexity of the process of inducing a pattern description depends on two factors: 1) the complexity of the description language used (i.e., the number and the type of operators the system understands), and 2) the intricacy of the pattern description to be induced.

If the pattern description involves no intermediate concepts then the above rule describes one-level class descriptions. In multi-level class descriptions there are intermediate rules between the lowest level concepts involving only measurable properties of objects, and a top level description involving higher level concepts directly related to the given class.

Another important classification of learning techniques is based on the degree to which *descriptors* (attributes, relations, predicates, operators) used in the observational statements are relevant to the decision classes (in other words, the degree to which initial descriptors are related to the descriptors used in the class descriptions). At the lowest level, the descriptors used in the observational statements are the same as the ones used in the class descriptions. That means that the given descriptors are directly relevant to class descriptions (such a case is typical in many methods). At the next level, the initial descriptors should contain the relevant ones, though not all contained must be relevant. In this case, the system must have the ability to determine the most relevant descriptors among many given descriptors. At the highest level, initial descriptors may be completely different from the ones used in the final definition of the concept. We illustrate this case in section 5, where given descriptors are physical properties of some objects (in this case trains), and the final descriptors are concepts such as "trains with toxic or non-toxic loads".

Let us illustrate *learning from examples (concept acquisition)*, and differentiate it from *learning from observation*, by an example problem known as "East-bound and West-bound trains" [Michalski & Larson, 1977] shown in Fig. 1. Given are two collections of trains, those that are "East-bound" (A to E) and those that are "West-bound" (F to J). The task is to determine a simple descriptive rule

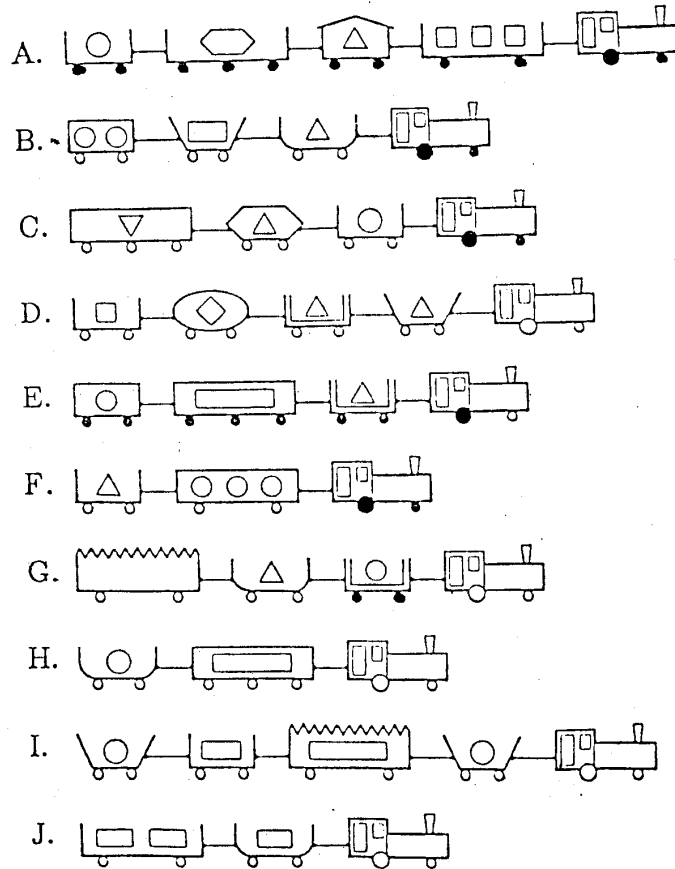


Figure 1. The TRAINS problem

distinguishing between the East-bound and West-bound trains using examples of the trains.

These trains are highly structured; each consists of a sequence of cars of different shapes and sizes. The trains have different number of cars, and cars have different lengths. Thus, an adequate description of trains involves both qualitative and quantitative descriptors (e.g., numerical attributes such as *number of trains*, the *length of a car*, or the *number of loads in a car*, categorical attributes such as *shape of a car*,

and relations such as *contains, in-front-of*). To illustrate one possible solution, let us present the discriminant descriptions of East-bound and West-bound trains found by the program INDUCE/2 [Hoff, Michalski, Stepp, 1983]. These discriminant descriptions (i.e., rules for distinguishing between the two classes of trains) are expressed in the Annotated Predicate Calculus (APC), which is a typed predicate calculus with additional operators (Michalski, 1983). The descriptions are:

East-bound $\Leftarrow:: \forall(\text{train}) \exists(\text{car}) [\text{contains}(\text{train}, \text{car})][\text{length}(\text{car})=\text{short}][\text{shape}(\text{car})=\text{closed}]$

("A train is East-bound if it contains a short, closed car.")

West-bound $\Leftarrow:: \forall(\text{train}) [\text{num-cars}(\text{train})=2] \vee$

$\exists(\text{car}) [\text{contains}(\text{train}, \text{car})][\text{shape}(\text{car})=\text{jagged top}]$

("A train is West-bound if there are two cars in the train or if there is a car with a jagged top.")

An early practical application of the learning from examples strategy to building the knowledge base of an expert system is described in the paper by Michalski and Chilausky (1976). In the follow-up paper (Michalski and Chilausky, 1980), the learning from examples strategy was compared with the strategy of learning by being told in the context of building the afore-mentioned expert system PLANT/ds. This experiment resulted in inductively derived diagnostic rules (i.e., those obtained by machine learning from examples) that outperformed the rules determined by interviewing an expert (i.e., those acquired by the learning from instruction strategy). Recently, Reinke (1984) described a system for testing the consistency and completeness of a rule base using techniques of inductive inference.

5. LEARNING FROM OBSERVATION

The learning from observation strategy is applied when a collection of facts (observations) is given and one wants to develop a general description (a theory) explaining the facts. It is assumed that there is no teacher who can explain the facts or identify important or relevant concepts applicable to them.

The first step in developing a theory about a collection of facts is usually the creation of a classification (taxonomy). Such a classification can be considered a general description of these facts. Creating simple yet useful classifications is a challenging intellectual process of great importance.

So far, the problem of automatically creating classifications has been studied mainly in the areas of numerical taxonomy and cluster analysis. In these areas, the basic principle for creating a classification is to form classes of objects using some mathematical measure of similarity between the objects. This measure is defined over a finite, a priori set of attributes given to characterize the objects. Objects in the same class have a high degree of similarity and objects in different classes have a low degree of similarity.

One difficulty with this approach is that classes (concepts) formed solely on the basis of some fixed measure of similarity are often difficult to interpret conceptually. In fact, the interpretation of obtained classifications is assumed in this approach to be the task of a data analyst. This approach does not take into consideration possible goals for classification, nor does it use general concepts or linguistic constructs that characterize a collection of observations as a whole (i.e., concepts that capture Gestalt properties). For example, if a collection of points forms a "T-joint", then in order to describe it this way, the system must contain in its background knowledge base the definition of such a concept. Even if the computation of similarities (here, reciprocal of distances) puts all the points forming a "T-joint" into the same class, the system still would not "know" that the collection can be described this way.

Another approach to creating classifications is based on *conceptual clustering* (Michalski, 1980; Michalski and Stepp, 1983). In this approach, observations are partitioned into classes that represent some conceptual entities. Instead of *similarity*, the approach uses the measure of *conceptual cohesiveness* between objects. The similarity of objects A and B is a function only of properties of these objects, i.e., is a two-argument function $f(A,B)$. In contrast to this, the conceptual cohesiveness is a function of the properties of objects A and B, of the surrounding objects, E (the environment), and of a set of concepts, C, available in the given description language for describing these two objects together. Thus, the conceptual cohesiveness is a four-argument function $f(A,B,E,C)$.

In *conjunctive conceptual clustering* objects are assembled into classes that represent conjunctive concepts closely circumscribing the given objects (i.e., "fitting" the objects well) and satisfying some

additional criteria measuring *clustering quality* (Michalski and Stepp, 1983). These criteria take into consideration the relation of the classes to a set of possible goals of classification, the complexity of generated class descriptions, their "disjointness", and other factors (Michalski and Stepp, 1983). The conjunctive concepts are descriptions in the form of conjunctions of statements specifying properties (attribute values) of objects representing the given concept, the relations among the object parts and the properties of the parts.

For illustration, let us consider an example (borrowed from Stepp and Michalski, 1984). Suppose that trains in Fig.1 are not assigned to any classes, and the task is to create a meaningful classification(s) of these trains. What criteria would people use to create such a classification?

To answer this question, cognitive psychologist Doug Medin at the University of Illinois performed an experiment with 31 subjects, who were asked to solve this problem (Medin, Wattenmaker and Michalski, 1984). The subjects devised a total of 93 classifications of the trains. The most popular criterion for classification (used in 17 classifications) was the number of cars in the train. Trains were classified into 3 groups: 2-car, 3-car and 4-car trains respectively. The second most popular classification (7 cases) was based on the color of engine wheels. Trains were classified to two groups: a group in which all engine wheels are white, and the group which does not have this property.

These results suggest that even in the absence of clear goals for a classification, people tend to use similar criteria for creating a classification. This similarity pattern is not very strong, however, as evidenced by a large number (40 out of 93) of unique classifications, i.e., different people proposed different solutions. The same problem was given to the recently developed program CLUSTER/S (Stepp, 1984). The program generated several classifications. Two of them are shown in Fig. 2. The first classification A, uses as classification criterion the *number of different car shapes* in the train. In the second classification B, the criterion used is whether the wheels on all cars in a train are the same or not. Although these classifications are different from the most frequent classifications made by people, they seem to be reasonable, and even appealing.

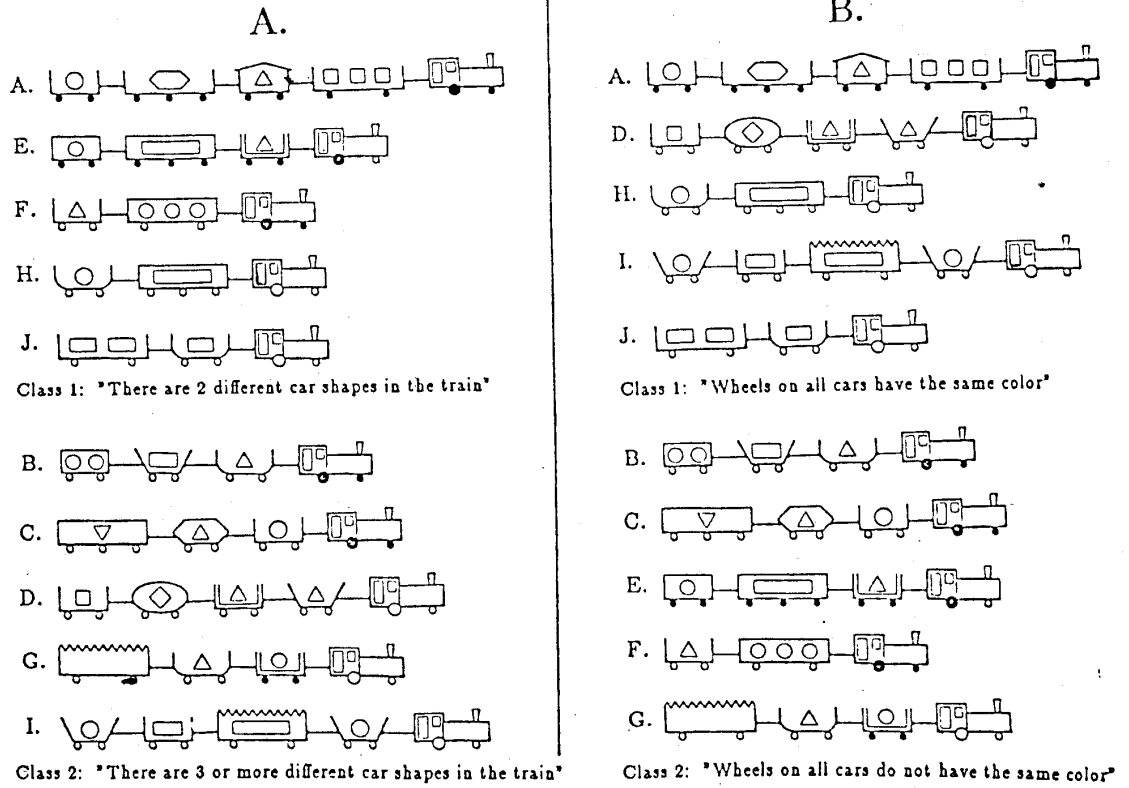


Figure 2. Two classifications of TRAINS created by program CLUSTER/S

It should be mentioned that the initial descriptions of the trains (the observational statements) did not include statements about the number of different car shapes or whether the car wheels have the same color. How did the program generate such statements and use them in creating classifications?

The background knowledge of the program contains inference rules that, when applied to the original descriptions of the trains, can generate new descriptors (attributes and relations) characterizing given

objects (here, trains) or their parts. Using various heuristics, the program selectively generates new descriptors and attempts to apply them in the process of determining tentative classifications. The program evaluates the tentative classifications according to the *classification quality* criterion LEF (lexicographic evaluation functional). The LEF criterion takes into consideration various properties of a classification, such as the degree to which it satisfies a set of goals (defined in the program's background knowledge), the degree of *fit* between a classification and the observed events (objects), and the importance of descriptors occurring in the class descriptions (see, Michalski, 1980, Medin, 1982; Michalski and Stepp, 1983 a,b,c). To illustrate the concept of *importance* of a descriptor, let us assume that background knowledge of the system includes a rule which defines cars in the train that carry toxic chemicals. Suppose that such a rule is :

$$[\text{contains}(\text{train}, \text{car})][\text{car-shape}(\text{car})=\text{opentop}][\text{cargo-shape}(\text{car})=\text{circle}][\text{items-carried}(\text{car})=1] \\ \iff [\text{toxic-chemicals}(\text{train})]$$

In this rule, the equivalence relation is used to state that the negation of the condition part is sufficient to assert the negative of the consequence. After applying this rule to each train description, the right-hand side of the rule will be appended to the description (as an additional predicate) to state the presence or absence of toxic-chemicals on the given train. This predicate will in turn trigger other inference rules that are part of the background knowledge base:

- toxic chemicals are dangerous,
- dangerous things are important,
- important things should have high selection value (high preference score).

As a result of this inference the program will propose a candidate classification of trains into those containing toxic chemicals and those not containing such chemicals. The descriptor generation process outlined above constructs new attributes from combinations of existing attributes. Certain heuristics of attribute construction are used to guide the process. For example, two attributes that have linearly ordered value sets can be combined using arithmetic operators. When the attributes have numerical values (as opposed to symbolic values such as "small," "medium," and "large") a trend analysis is used to suggest appropriate arithmetic operators, as in BACON 4 (Langley, Bradshaw, Simon, 1983).

Predicates can be combined by logical operators to form new attributes. For example, the rule

$$[\text{cold-blooded}(a1)][\text{offspring birth}(a1)=\text{egg}] \Rightarrow [\text{animal-type}(a1)=\text{reptile}]$$

yields a new attribute "animal-type" with a specified value "reptile." Using this rule and similar ones, one might classify some animals into groups of reptiles, mammals, and birds (even though the type of each animal is not stated in the original data). Such classification construction problems occur when one wants to organize and classify observations that require structural descriptions. Problems of this type include classifying physical or chemical structures, analyzing genetic sequences, building taxonomies of plants or animals, characterizing visual scenes, and splitting a sequence of temporal events into episodes with simple meanings. In an expert system, a classification construction program could be used to structure a large knowledge base of decision rules, or to structure the database of facts about a given problem.

6. SUMMARY

Fundamental learning strategies have been discussed including *direct implantation of knowledge*, *learning by instruction*, *learning by deductive inference*, *learning by analogy*, *learning from examples* and, finally, *learning by observation and discovery*. The order of these strategies reflects the increasing complexity of the inference process that a learning system must perform on the given information in order to derive the desired knowledge.

Learning from examples and learning from observation are two basic forms of inductive learning. The paper discussed the importance of using background knowledge in applying these learning strategies. The capability to incorporate background knowledge in inductive learning is an important prerequisite for the successful application of this form of learning.

ACKNOWLEDGMENT

It is this author's pleasant duty to thank Gail Thornburg, Robert Stepp and Larry Rendell for comments and proofreading of the paper. The work presented here was done in part at the Artificial Intelligence Laboratory of the University of Illinois, Department of Computer Science and in part at the

Artificial-Intelligence Laboratory of the Massachusetts Institute of Technology. The support for the Artificial Intelligence Laboratory at the University of Illinois is provided in part by the National Science Foundation under grant DCR 84-06801 and in part by the Office of Naval Research under grant No. N00014-82-K-0186. Support for MIT's Artificial-Intelligence Research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-80-C-0505.

REFERENCES

- Boulanger, A., "The Expert System PLANT/cd: A case study in Applying the General Purpose Inference System ADVISE to Predicting Black Cutworm Damage in Corn," M.S. Thesis, UIUCDS-R-83-1134, Department of Computer Science, University of Illinois, Urbana, IL, July 1983.
- Buchanan, B.G., Mitchell, T.M., Smith, R.G. and Johnson, C.R., Jr., "Models of learning systems", Tech. Rept. STAN-CS-79-692, Computer Science Department, Stanford University, 1979.
- Burstein, M.H., "Concept formation by incremental analogical reasoning and debugging," Proceedings of the 2nd International Machine Learning Workshop, Allerton House, University of Illinois at Urbana-Champaign, June 22-24, 1983.
- Carbonell, J.G., "Learning by analogy: formulating and generalizing plans from past experience, chapter in *Machine Learning*, R.S. Michalski, J. Carbonell and T. Mitchell, (eds.), Tioga Publishing Company, 1983.
- Davis, R., "Knowledge acquisition in rule based systems: Knowledge about representation as a basis for system construction and maintenance." Chapter in *Pattern-Directed Inference Systems*, D.A. Waterman and F. Hayes-Roth (eds.), Academic Press, 1978.
- Dietterich, T.G. and Michalski, R.S., "A comparative review of selected methods for learning from examples," chapter in *Machine Learning*, R.S. Michalski, J. Carbonell and T. Mitchell, Eds., Tioga Publishing Company, 1983.
- Hass, N. and Hendrix, G.G., "Learning by Being Told: Acquiring Knowledge for Information Management", IBID
- Hoff, W.A., Michalski, R.S., Stepp, R.E., "INDUCE 2: A program for learning structural descriptions from examples," Technical Report, Department of Computer Science, University of Illinois, Urbana, Illinois, September, 1983.
- Hu, Chi-Heng and Yoh-Han Pao, "Processing of pattern based information, Part II: Description of inductive inference in terms of transition networks, Technical Report 102-82 Part II, Case Institute of Technology, Case Western Reserve University, Cleveland, Ohio 44106, 1982.

Langley, P., Bradshaw, G.L., Simon, H.A., "Rediscovering chemistry with the BACON system," chapter in *Machine Learning*, R.S. Michalski, J. Carbonell and T. Mitchell, (eds.), Tioga Publishing Company, 1983.

Lenat, D., AM an artificial intelligence approach to discovery in mathematics as heuristic search, Computer Science Department, Rept.STAN-CS-76-750, Stanford University, Stanford, CA, 1976.

Lenat, D., Hayes-Roth, F., Klahr, Ph., "Cognitive economy in a fluid task environment, Proceedings of the International Machine Learning Workshop, R.S. Michalski (ed.), Allerton House, June 22-24, 1983.

Medin, D.L., "Structural principles in categorization," *Developments of perception and cognition*, 1982.

Medin, D.L., Wattenmaker, W.S., Michalski, R.S., "Constraints on rule induction in classification," to be published, 1985.

Michalski, R.S., A variable-valued logic system as applied to picture description and recognition," Chapter in the book *Graphic Languages*, F. Nake and A. Rosenfeld (eds.), North-Holland Publishing Co., 1972.

Michalski, R.S., "Knowledge acquisition through conceptual clustering: a theoretical framework and an algorithm for partitioning data into conjunctive concepts," *Policy Analysis ad Information Systems*, Vol. 4, No. 3, pp. 219-244, 1980b.

Michalski, R.S., "A theory and methodology of inductive learning," chapter in *Machine Learning*, R.S. Michalski, J.G. Carbonell and T.M. Mitchell(eds.), Tioga Publishing Company, 1983.

Michalski, R.S., Chilausky, R.L., "Knowledge acquisition by encoding expert rules versus computer induction from examples: a case study involving soybean pathology", *International Journal for Man-Machine Studies*, No.12, pp.63-87, 1980.

Michalski, R.S. and Stepp, R.E., "Automated construction of classifications: conceptual clustering versus numerical taxonomy," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 4, pp.396-410, July, 1983a.

Michalski, R.S. and Stepp, R.E., "Learning from observation: conceptual clustering," chapter in *Machine Learning*, R.S. Michalski, J. Carbonell and T. Mitchell (Eds.), Tioga Publishing Company, 1983b.

Michalski, R.S., and Stepp, R.E., "How to structure structured objects," *Proc. Intern. Machine Learning Workshop*, R.S. Michalski (ed.), Monticello, Illinois, June 22-24, 1983c.

Mitchell, T.M., Version spaces: an approach to concept learning, Ph.D. Thesis, Stanford University, Stanford, CA 1978.

Mitchell, T.M., Utgoff, P. E., Banerji, R., "Learning by experimentation: acquiring and refining problem-solving heuristics," chapter in R.S. Michalski, J. G. Carbonell and T. M. Mitchell (eds.), Tioga Publishing Company, 1983.

Pao, Yoh-Han and Hu, Chi-Heng, "Processing of pattern based information, Part I: inductive inference methods suitable for use in pattern recognition and Artificial Intelligence, Technical Report 101-82, Case Institute of Technology, Case Western Reserve University, Cleveland, Ohio 44106, 1982.

Reinke, R., "Knowledge acquisition and refinement tools for the ADVISE meta-expert system," M.S. Thesis, Department of Computer Science, University of Illinois, Urbana, Illinois 1984.

Rendell, L.A., "A new basis for state-space learning systems and a successful implementation," *Artificial Intelligence*, vol.20, No.4, pp.369-392, 1983a.

Rendell, L.A., "Toward a unified approach for conceptual knowledge acquisition," *The AI Magazine*, Winter 1983b.

Stepp, R.E., "Conjunctive conceptual clustering: a methodology and experimentation," Ph.D. thesis, Department of Computer Science, University of Illinois, Urbana, 1984.

Stepp, R.E. and Michalski, R.S., "Discovering meaningful classifications of structured objects: further progress on conceptual clustering," *Machine Learning: An Artificial Intelligence Approach* vol.II, in preparation, 1985.

Waltz, D., Genesereth, M., Hart, P., Hendrix, G., Joshi, A., McDermott, J., Mitchell, T., Nilsson, N. and Wilensky, R., "Artificial Intelligence: An assesment of the state-of-the-art and recommendation for the future directions, *AI Magazine*, vol.4, No.3, Fall 1983.

Winston, P.H., "Learning Structural Descriptions from examples, Tech. Report AI TR-213, MIT, AI Lab, Cambridge, MA, 1977.

Winston, P.H., "Learning by augmenting rules and accumulating censors," *Proceedings of the 2nd International Machine Learning Workshop*, Allerton House, University of Illinois, Urbana, June 22-24, 1983.

| | | | |
|--|--|--|--|
| BIBLIOGRAPHIC DATA SHEET | 1. Report No. File No. UIUCDCS-F-84-926 | 2. | 3. Recipient's Accession No. |
| | 4. Title and Subtitle Learning Strategies and Automated Knowledge Acquisition: An Overview | | 5. Report Date November 1984 |
| 7. Author(s) Ryszard S. Michalski | 8. Performing Organization Rept. No. | | 6. |
| 9. Performing Organization Name and Address Department of Computer Science University of Illinois at Urbana-Champaign Urbana, IL 61801 | | 10. Project/Task/Work Unit No. | 11. Contract/Grant No. DCR 84-06801 |
| 12. Sponsoring Organization Name and Address National Science Foundation Washington, DC | | 13. Type of Report & Period Covered | 14. |
| 15. Supplementary Notes | | | |
| 16. Abstracts <p>Fundamental learning strategies are discussed in the context of knowledge acquisition for expert systems. Special consideration is given to inductive learning strategies: learning from examples (concept acquisition) and learning from observation (concept formation without teacher). A specific form of learning from observation is discussed in detail, namely, conceptual clustering, i.e., structuring given observations into a hierarchy of conceptual categories.</p> <p>An inductive learning system generates knowledge by drawing inductive inferences from the facts or observations under the guidance of background knowledge. The background knowledge defines goals of learning, the criteria for evaluating generated hypotheses, the properties of initial attributes and relations characterizing observed events, and inference rules and heuristics for deriving higher level <u>descriptive concepts</u>.</p> | | | |
| 17. Key Words and Document Analysis. 17a. Descriptors Machine Learning Inductive Inference Knowledge Acquisition Expert Systems Inference Systems Learning Strategies | | | |
| 17b. Identifiers/Open-Ended Terms | | | |
| 17c. COSATI Field/Group | | | |
| 18. Availability Statement | | 19. Security Class (This Report) UNCLASSIFIED | 21. No. of Pages 26 |
| | | 20. Security Class (This Page) UNCLASSIFIED | 22. Price |