2nd copy

An earlier and shorter version of this paper was presented
at the ONR Workshop on Knowledge Acquisition
Arlington, VA, November 6-7, 1989.

TOWARD A UNIFIED THEORY OF LEARNING:
Multistrategy Cooperative Learning

Ryszard S. Michalski
Center for Artificial Intelligence
George Mason University
4400 University Drive
Fairfax, VA 22030

Submitted to Machine Learning Journal

May 21, 1990

89-20

Abstract

As the field of machine learning is experiencing great diversification and multiplication of research directions, there is a need for developing theoretical frameworks that would unify different paradigms and methods, and clarify relationships among them. This paper discusses an *inference-based theory* of learning that is intended to provide such a framework, and serves as a basis for an integrated task-responsive learning methodology, called *multistrategy cooperative learning* (MCL).

In any process of learning, an MCL learner applies the strategy, or a combination of strategies, that is most appropriate for achieving the desired goal in the context of the relationship between the input information and the learner's prior knowledge. The input information to a learning act can be external, from a teacher or environment; or internal, produced as an intermediate result of the previous learning act. Any input information to the system is first evaluated for potential "relevance" to the learner's goal(s), and it it passes the test, a learning process is activated.

If the input information is "novel," but not inconsistent with the learner's prior knowledge, it is assimilated within the prior knowledge. This process may involve a generalization of relevant segments of prior knowledge, as well as storing representative facts. If the input is inconsistent with prior knowledge, and is believed to be correct, the appropriate parts of prior knowledge undergo modification. This process typically involves specialization, and/or weakening parameters indicating the strength of relevant knowledge segments. If the input is similar to some part of the prior knowledge at some level of abstraction, an analogy is established, and explored in the context of the learner's goals. If the input is recognized to be already a part of the learner's prior knowledge, is implied by it, or implies it, the relevant segments are reinforced and/or restructured to facilitate their future use. Finally, if the input information is evaluated as irrelevant to the learner's goal(s), it is ignored. Thus, any input causes construction of either new knowledge or better knowledge.

The proposed methodology is intended to ultimately integrate empirical learning, constructive induction, learning by instruction, constructive deduction, explanation-based learning, reinforcement learning, conceptual clustering and learning by analogy (hence, the term *multistrategy*). An important property of the MCL methodology is that in a given act of learning, different strategies may cooperate to achieve the desired outcome (hence, the term *cooperative)*. To implement such a collaboration, the learner must have the ability to apply knowledge gained from one learning act in another learning act (this property is called *closed-loop learning*). It is hoped that the developed ideas will serve as a theoretical basis for building advanced integrated learning systems.

For every belief comes either through syllogism or from induction.
Aristotle, Prior Analytics, Book II, Chapter 23 (p.90)
ca 330 BC.

1. INTRODUCTION

Machine learning strives to develop principles underlying learning processes in whatever form they occur and to construct computational learning methods based upon them. The last several years have seen an extraordinary growth of this field, and one can expect a continuation of this trend in the predictable future.

While previously established areas, such as empirical symbolic learning and discovery systems, have continued to be very active (e.g., Laird, 1988; Segre, 1989), several new areas have been rapidly expanding, such as explanation-based learning (e.g., G. DeJong, 1988), and computational learning theory (e.g., Haussler and Pitt, 1988; Ehrenfeucht, 1988). There has also been a tremendous new interest in subsymbolic learning approaches, such as connectionist and neural network learning (e.g., Barto and Anderson, 1985; Touretzky, Hinton and Sejnowski, 1988; and Fisher et al., 1989), and mixed approaches, such as classifier systems and genetic algorithms (Holland, 1986, 1987; Davis, 1987; Goldberg, 1988; and Schafer, 1989).

Another important development has been an increasing interest in building systems that integrate different learning strategies. Among the most well-known such systems are Unimem (Lebowitz, 1986), Odysseus (Wilkins, Clancey, and Buchanan, 1986), Prodigy (Minton et al., 1987), DISCIPLE-1 (Kodratoff and Tecuci, 1987), the GEMINI (Danyluk, 1987 and 1989), OCCAM (Pazzani, 1988), and IOE (Dietterich and Flann, 1988). For other examples see (Segre, 1989).

In the context of great diversification of research directions and methods, there is a need for developing a general theoretical framework that would clarify the relationship among these directions and methods, and provide a conceptual foundation for building multistrategy learning systems. Such a framework might also provide insights for determining the most desirable new research directions.

This paper presents our initial results toward developing such a theoretical framework. It outlines the *inference-based learning theory* that views learning as a process of knowledge transformation based on inference, and classifies learning methods into two basic types: synthetic and analytic. Synthetic methods involve a generation of *hypothetical* explanations of the input information, and use induction as the primary type of inference. Analytic methods involve a generation of *derivational* explanations of the input, and use deduction as the primary type of inference. The theory is then used as a basis for an integrated task-responsive learning methodology, called *multistrategy cooperative learning* (MCL).

The MCL methodology is intended to integrate empirical learning, constructive induction, constructive deduction (see sec. 4), reinforcement learning, learning by instruction, explanation-based learning, conceptual clustering and learning by analogy. It postulates that a learner should be able to learn something from any input information (even from facts that it already "knows"), and should employ the strategy that is most effective for a given combination of dynamically changing input facts, the learner's prior knowledge and the goal of learning. Another postulate is that whatever the learner learned should be reusable in subsequent acts of learning (the "closed-loop" assumption). An interesting consequence of these assumptions is that to meet the needs of different strategies, a new form of knowledge representation has to be employed.

The presented work is an extension and an elaboration of the ideas presented earlier in (Michalski and Ko, 1988; Michalski and Watanabe, 1988; and Ko and Michalski, 1989). It also draws upon many ideas developed by others in the field of machine learning and cognitive science. Its intention is to provide a conceptual analysis of different learning methods and describe the intuition behind various ideas, rather than present a formal and rigorous treatment of the topics considered. References to identified sources are made in the text. We start with an outline of the inference-based theory of learning and a discussion of basic learning strategies.

2. INFERENCE-BASED THEORY OF LEARNING

A key idea in symbolic learning is that the learner typically acquires desired knowledge through some form of reasoning - inductive, analogical, or deductive. In special cases, learning involves only copying the information provided by a source ("rote learning"), or syntactically transforming it and/or selecting from it some parts ("learning by instruction'). These cases, however, are not central to understanding the learning behavior, and will not be discussed here. The reasoning process is activated by input facts, obtained from a teacher or from the environment. It involves the learner's prior knowledge ("background knowledge") and is guided by the goal of learning. The learning goal sets the criteria for determining the relevant parts of prior knowledge, choosing the learning strategy, and selecting the most preferred hypothesis among candidate ones. The goal also specifies the amount of effort to be extended in pursuing any specific strategy.

Thus, learning can be viewed as a process of transforming input information into the desired knowledge by the use of inference and under the guidance of the learner's goal. Because a learning act may involve any type of inference, a complete theory of learning must therefore include also a theory of inference.These ideas provide a framework for what we call the *inference-based theory of learning* (Figure 1).

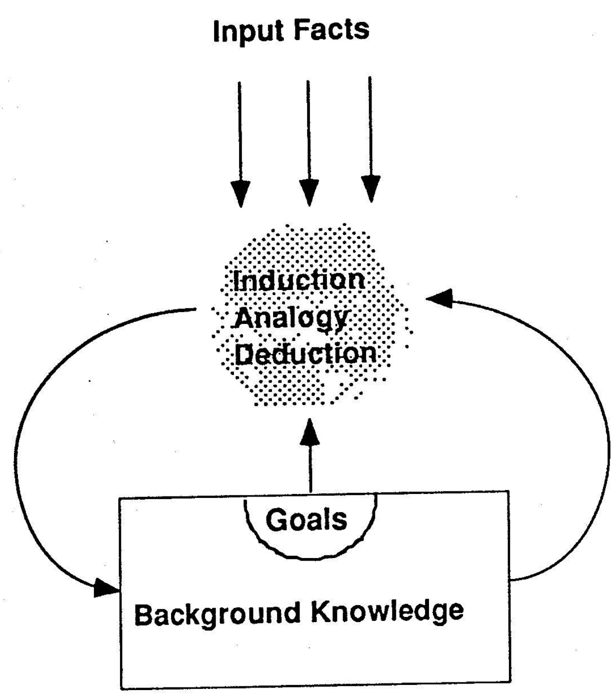


Figure 1. An Illustration of the inference-based theory of learning

The input information ("input") can be facts, observations, concept instances, previously formed generalizations, or other knowledge. The input activates segments of background knowledge that are most relevant to it from the viewpoint of the learning goal(s). This is done by making matches between the input and hierarchically organized knowledge segments (see knowledge representation in section 5).

Depending on the goal of learning and the relationship between the input and the activated knowledge segments, a different primary inference is involved. This inference may be analogical, if the input is "similar" to what the learner already knows, and the goal is to make a decision about the input based on the past experience. It can be inductive, if the input consists of one or more facts, or previously generated descriptions, and the goal is to generalize these facts and/or descriptions. It can be deductive, if the input is a special case of what the learner already knows, but the possessed knowledge is not efficient or directly applicable, and the goal of learning is to evaluate future similar inputs as efficiently as possible. The results of the inference are assimilated into Background Knowledge (BK), so that the next act of learning will involve the modified BK. The above implies that whatever is learned must be expressed in the form compatible with the form in which any other knowledge is stored. Thus, a segment of knowledge in BK can itself be an input to a learning process. This aspect is called *closed-loop learning.*

In summary, the inference-based theory states that in order to learn, an agent needs to be able to perform transformations of knowledge, i.e., to perform *inference*, and to have *memory* which supplies the learner with the background knowledge needed for performing the inference, and records the results of the inference *in the form useful for future use.* Without either of the two components there cannot be learning (except for "rote" learning). Thus, one can say that:

Learning = Inference + Memory

While these ideas seem clearly to apply to symbolic learning, one may ask if they also apply to methods such as those used in connectionist systems or genetic algorithms. A short answer is that they do apply, because both the latter approaches are capable of generalizing or specializing information, although implicitly rather than explicitly, as in symbolic systems. They also have the ability to memorize results of their learning for future use, although, again, they do it in a different way.

Another question may be what is the main goal of this theory, in particular, whether it is intended to be a cognitive theory, which explains information processes in human (or animal) learning. The answer is that our intention is to develop a theory that is sufficiently general that unessential biological and implementational aspects can be ignored, but sufficiently specific that it can account for major information processes occuring in diverse forms of learning, whether they exist in nature or not. The success of the theory can be measured by its ability to characterize and explain conceptual relationships among methods and paradigms studied in machine learning or in human learning. We deliberately avoid being too formal, so that we can primarily stress the ideas and intuitions behind different learning strategies and their interrelationships.

Many proposed ideas have been inspired by observing how people or animals learn, but no claims are made that this is a cognitive theory. It is hoped, however, that the presented outline of the theory will be helpful for fostering our understanding of learning processes in general, and for implementing more advanced learning systems.

3. THE ROLE OF EXPLANATION IN LEARNING

It is a well-known fact that it is difficult to learn anything without understanding it. At the early stage of development humans, of course, acquire a lot of information by rote, but among mature individuals rote learning plays a minor role. Understanding, in turn, is a result of self-creating or receiving a satisfactory explanation. The notion of explanation, however, needs clarification.

In explanation-based learning (Mitchell, Keller and Kedar-Cabelli, 1986; DeJong and Monney, 1986; DeJong, 1986 and 1988;), an explanation is the process of deductively applying the learner's prior knowledge (domain knowledge and an abstract definition of the goal concept) to demonstrate that a given example is an instance of the goal concept. The obtained proof is then used to restructure the learner's concept definition, to make it more efficient or useful ("operational").

In inductive learning, producing some form of an "explanation" of the observed facts is the main purpose of learning. Through induction, a scientist builds a theory explaining an observed phenomenon, or a medical researcher develops a general description of a disease (Michalski, 1983). In the latter case, such a description, especially if it is stated in terms of causal relations, is is a form of "explanation" of the disease and of the patient's symptoms.

In the field of machine learning, inductive learning has been implemented in several forms. In empirical inductive learning, generated "explanations" are simple generalizations of the given facts or examples. These facts or examples are typically expressed in terms of attributes that are selected a priori. Such generalizations hardly deserve to be called explanations. For example, the rule "if there is smoke, there is fire" may be a generalization of some observations, but it does not give a "real" explanation of the phenomenon. Only in knowledge-based induction, i.e., constructive induction or abduction (see sec. 4), can the result of learning be a "true" explanation. This is because such induction is capable of creating descriptions in terms of high level concepts and/or causal relationships (e.g., Hoff, Michalski and Stepp, 1983; Mehler, Bentrup and Riedesel, 1986; O'Rorke, Morris and Schulenburg, 1989).

It is clear that the word explanation can be used in several senses, and it is not easy to capture its meaning formally. The Webster's Third International Dictionary states that "an explanation is an act or process of explaining," and that it "consists in successfully comparing a new phenomenon with an older and more familiar one." While such an "explanation" of explanation sounds intuitively correct, in order to make it computational, one needs to better specify the meaning of "comparing" in this definition.

It appears that one can adequately capture the meaning of "comparing" by assuming that it means a demonstration that the new phenomenon is a logical consequence of the old familiar one. Thus, the process of explanation involves proving such a logical consequence, i.e., it involves deduction.

We postulate that creating an explanation of some observation involves in general constructing two components (Michalski and Ko, 1988 and Michalski and Watanabe, 1988):

- an *explanatory hypothesis*, which, together with a reasoner's background knowledge, entails the observation ("strongly" or "weakly"), and
- an *explanatory structure*, which demonstrates this entailment.

where a "strong" entailment means a logical entailment, and a "weak" entailment means a plausible or probabilistic entailment.

To illustrate the above components, let us use an example concerning the U.S. space shuttle Challenger disaster on January 28, 1986, and Richard Feynman's experiment to explain the reason for this disaster. During a launch of a shuttle there are always vibrations that cause the rocket joints to move a little. Inside the joints are the so-called O-rings, which are supposed to expand to make a seal. However, if the O-rings do not expand, hot gas would escape through the joints, which could start a fire. Thus, if the O-rings lose their resiliency just for a second or two, this could cause an accident.

In a televised meeting of the presidential commission investigating the accident, Richard Feynman made the famous O-ring ice-water demonstration. He put a sample of the rubber from the O-rings in a clamp, and submerged it for a while in a glass of ice-water. Then he took the rubber out, and showed that when the clamp was undone, the rubber did not spring back.

Thus, the experiment showed that the rubber of the O-rings loses resiliency at low temperatures. This information is what we previously described as explanatory hypothesis. To explain the accident one also needs the above mentioned (background) knowledge, that if the rings lose resiliency, hot gas can escape through the O-rings, which, in turn, can cause the observed fire, and the rocket explosion. One also needs to know that on the morning of the launch, the temperature was 29 ^{o}F, which is low. The sequence of reasoning steps that lead from the explanatory hypothesis combined with background knowledge to the phenomenon being explained is called the explanatory structure.

Defining Explanation

We call an observation, a process, or anything that is supposed to be explained, the *explanatory target* (ET). To explain an ET to an agent (or to oneself) one needs to show that the *explanatory hypothesis* (EH), plus the agent's *background knowledge* (BK) entails the ET. As we stated before, the sequence of steps demonstrating this entailment is the explanatory structure (ES). In brief, ES demonstrates that EH and BK strongly or weakly entails ET, which we write as:

$$\text{EH} \ \& \ \text{BK} \mid\!> \text{ET} \qquad (1)$$

In some situations, the explanatory hypothesis does not need to be constructed, because the background knowledge itself entails ET. In the example about the Challenger accident, if it were well known that the rubber used in O-rings loses resiliency at low temperatures, there would have been no need for Feynman's demonstration. In such situations, the explanatory structure simply demonstrates that:

$$\text{BK} \mid\!> \text{ET} \qquad (2)$$

In the most general case, the background knowledge may be incorrect or inconsistent with respect to an observation, and the total explanatory hypothesis may also involve modifications of the background knowledge. In such a case, instead of BK, we would have a modified BK*:

$$\text{EH} \ \& \ \text{BK}^* \mid\!> \text{ET} \qquad (3)$$

For example, suppose that in the Challenger example, the background knowledge includes an erroneous belief that nothing could be wrong with the O-rings. In this case, the explanatory hypothesis would include not only the knowledge that the rubber O-rings are not resilient at low temperatures, but also a correction of the erroneous belief.

The explanatory hypothesis (in the Challenger example, "the rubber of O-rings loses resiliency at low temperatures") may be obtained in several ways. It may be obtained through an experiment, and a generalization of the results. In our example, the observation "the rubber loses resiliency in a glass of ice-water" can be generalized to "the rubber loses resiliency at low temperatures" (thus, it will lose resiliency when exposed to the cold air). Such a process of making generalized statements solely on the basis of experimental observations is *empirical induction.*

To come up with the idea to make such an experiment, however, one needs to perform knowledge-based induction, i.e., constructive induction (see section 4). The background knowledge might involve rules, such as:

> "If there is a leak, the gas escapes through. If the gas is very hot, and gets in contact with flammable material, then it might cause a fire."

A fire has been observed. By reasoning backward (from consequences to premises), one would hypothesize that there might have been a leak. What could cause a leak? Many things could cause a leak. For example, if rubber of the O-rings would shrink due to vibrations, and did not spring back, this would cause a leak. Since properties of materials change with temperature, let us then see how the rubber of O-rings behaves in the kind of temperature observed on the day of the flight.

Conducting such reasoning leads one to the idea of making an experiment, like the one made by Feynman.

An alternative way to get the explanatory hypothesis is to deduce it from a general theory. In our example, it would be a theory about the behavior of different rubber materials at different temperatures.

Finally, the explanatory hypothesis can be received directly from a source of information. For example, Feynman, instead of making a demonstration, might have referred to a technical document stating the properties of this specific rubber material. This is a form of learning by instruction. (Of course, the information in the document could have been a result of prior experiments).

The above methods of creating an explanation of observations correspond to basic strategies of learning - learning by empirical induction, constructive induction, by deduction, and by instruction, respectively.

Types of Explanation

We distinguish between two basic types of explanation, one in which the explanatory hypothesis is not needed because background knowledge is sufficient to explain the ET, and the second in which explanatory hypothesis has to be created through some form of plausible inference (empirical induction, constructive induction or analogy). Accordingly to these two cases we have two types of explanation:

- *derivational* (or *deductive*) explanation, which consists only of an explanatory structure demonstrating through deductive reasoning that the knowledge already possessed (knowledge supplied by some source plus the agent's prior knowledge) implies the explanatory target, and
- *hypothetical* (or *inductive*) explanation, which consists of an explanatory hypothesis and an explanatory structure demonstrating that this hypothesis together with the learner's background knowledge implies the explanatory target. The explanatory hypothesis is created by inductive reasoning or analogy (which can be viewed as induction and deduction combined).

Learning processes that involve making primarily derivational explanations are called *analytic*. Their main result or purpose is restructuring prior knowledge into a form that is better in some sense (e.g., more efficient, easier to understand, or operational), or strengthening the belief in prior knowledge. This form of learning has also been called "learning at the symbol level" (Dietterich, 1986).

Learning processes that involve making primarily inductive explanations are called *synthetic*. Their main result or purpose is to hypothesize new knowledge, i.e., knowledge not contained in the deductive closure of the learner's background knowledge. This form of learning has also been called "learning at the knowledge level" (Dietterich, 1986). Synthetic learning methods can, in turn, be classified into *empirical induction* (in which the role of prior knowledge is limited), and *constructive induction* (in which prior knowledge plays a significant role).

One may ask how these ideas relate to skill acquisition through practice, since such processes seem to involve little reasoning. This question can also be extended to learning in connectionist systems or genetic algorithms. The above learning processes do not explicitly execute any symbolic rules of inference. However, by comparing the input information (e.g., training examples or practice exercises) with the observed behavior on new cases, one can say that, from the conceptual viewpoint, the above processes do implicitly perform operations that are logically equivalent to those of generalization, analogy or specialization. For example, the famous Pavlov's experiments have shown that dogs can perform instinctively certain limited generalizations of sound, smell or other signals, without any reasoning. Connectionist systems, as well as genetic algorithms are clearly capable of generalization, specialization or analogy, though again, not in an explicit form.

In the next section we use the above ideas to compare synthetic and analytic paradigms, and then to describe the multistrategy constructive learning.

4. BASIC TYPES OF LEARNING

Inference-based learning theory treats learning as an inference process that involves input facts and the learner's prior knowledge. If the results of this inference are evaluated as important, they are stored for future use, and this completes a single learning process. The major type of inference involved in a learning process defines the learning strategy. The lowest level strategy is when there is no inference done by the learner, and the inputs are stored as they are received (*direct knowledge implantation* or *rote learning*). The next level strategy is when there is only a selection of information from and/or syntactic transformation of the source information (*learning by instruction*). The above two forms involve little knowledge transformation (inference) on the part of the learner and are not central research topics in the field of machine learning. We will, therefore, concentrate here only on forms of learning that involve a substantial transformation of the source knowledge, specifically analytic and synthetic.

Before we discuss these two major strategies in more detail, let us briefly classify learning processes on the basis of the form of knowledge that a learner starts with, and the form of knowledge acquired. From this viewpoint, learning methods can be classified into four basic classes:

DD (Declarative to Declarative) - The initial knowledge is in a declarative form, and the derived knowledge is also in a declarative form. For example, in explanation-based generalization (Mitchell, Keller and Kedar-Cabelli, 1986), the initial knowledge is an example (a declarative description of some fact or observation with an associated class) plus the learner's background knowledge (domain knowledge, goal concept plus some domain-independent knowledge, i.e., relevant inference rules). The deductively derived output is "operational" knowledge (knowledge in the form useful for a given application). Another example is empirical induction from examples, where the input is typically a collection of observations stated in a declarative form, and the output is a generalization, also stated in such a form. Knowledge compression (reformulation) is also an example of DD learning.

DP (Declarative to Procedural) - The initial knowledge is in a declarative form, and the derived knowledge is in a procedural form. For example, advice taking or automatic programming are forms of the DP analytic method, because the input is some advice or a program specification typically in declarative form, and the output is a procedure for actually accomplishing the desired task (e.g., Bierman, Guiho and Kodratoff, 1984, Amarel, 1986; Mostow, 1986). Acquiring new skill can also be viewed as a DP learning task in which the initial knowledge includes a mental representation of what one should be able to do and observed results of trying, and the output knowledge is the improved control mechanism.

PD (Procedural to Declarative) - The initial knowledge is in a procedural form, e.g., an algorithm or a process developing in time, and the output is in a declarative form, e.g., a declarative description of this algorithm or process. If the input is a complete process, and the produced description just builds a "true" declarative description of it, then we have PD analytic learning. On the other hand, if a learner generalizes a description beyond the observable process, then we have a synthetic PD learning. Program SPARC that discovers rules for predicting sequences of objects is an example of synthetic PD learning (e.g., Dietterich and Michalski, 1986; Michalski, Ko and Chen, 1987).

PP (Procedural to Procedural) - The starting knowledge is procedural, as is the derived knowledge. Skill improvement with practice, automatic program optimization and the analogy-based program transformation are examples of PP learning.

As shown above, this classification cuts across both synthetic and analytic methods, and is useful for characterizing a variety of learning processes.

In the previous section we made a distinction between derivational and hypothetical explanations and, based on it, defined the synthetic and analytic learning processes. We will now analyze these two major types of learning in greater detail, and then discuss the proposed approach, *multistrategy constructive learning,* which attempts to provide a unifying framework for both of them.

4.1 Analytic Learning

As mentioned earlier, analytic learning involves an analysis of the input information in terms of the learner's relevant prior knowledge (domain-dependent and domain-independent), and then creation of desirable knowledge on the basis of this analysis. The primary type of inference engaged in this process is deductive, although in more recent versions of explanation-based learning, there can be some inductive inference involved also. A "pure" analytic learning method performs only a truth-preserving knowledge transformation, and thus the validity of the derived knowledge depends entirely on the validity of the input information and the background knowledge. If the initial knowledge is valid, so is the derived knowledge. Such pure analytic learning creates no "new" knowledge, but a more useful reformulation or specialization of the initial knowledge (i.e., the learner's prior knowledge, plus input information, such as a concept example supplied by a teacher).

Explanation-based learning

The most well-known form of analytical learning is explanation-based learning (Mitchell, Keller and Kedar-Cabelli, 1986; DeJong and Mooney, 1986). Other forms include "operationalization" (Mostow, 1983) and automatic program synthesis (e.g., Bierman, Guiho and Kodratoff, 1984).

Let us analyze explanation-based learning (EBL) in terms of the ideas presented above. In EBL, given an instance of a concept, the learner first determines an explanatory structure (proof) showing that the instance is indeed an example of the concept. An abstract concept definition ('goal concept"), relevant domain knowledge and domain-independent knowledge (inference rules) are assumed to be known to the learner a priori. All these components constitute what we call learner's background knowledge. The produced explanatory structure is then used to create a reformulation of the concept definition, so that it is more useful ("operational") for classifying future instances. This operational concept description is a generalization of the original instance and a specialization of the abstract concept definition. The underlying assumption is that future concept instances to be classified will be in the same form as the initial (training) instance.

Thus, (pure) EBL assumes that the learner's background knowledge (BK) is adequate to establish an explanatory structure explaining a given instance (i.e., the explanatory target ET), and there is no need for an explanatory hypothesis. The learner only seeks the explanatory structure that demonstrates that

$$BK \models ET \qquad (4)$$

holds, where $\models$ denotes logical (strong) entailment. The explanatory structure can be in the form of a proof tree generated by a theorem prover showing that the abstract concept definition (goal concept) is satisfied by a given example. The explanatory structure is then used to develop more effective knowledge. For example, in the ARMS system (Segre, 1987), given an initial plan for joining two assembly components, an example provided by a teacher and BK describing simpler goals, the system determines a more effective plan for joining components.

In short, in EBL, the learner relies primarily on BK, and the example serves as a focus of attention for learning. Once an explanatory structure is established, it is used to generalize the input example to the extent that the created operational knowledge BK* still subsumes ET, i.e., BK* $\models$ ET.

If BK is inadequate (inconsistent, incomplete or intractable), an explanatory structure cannot be established without postulating change in the background knowledge or hypothesizing some new knowledge. Thus, the applicabiltiy of this approach and the validity of its results depend on whether the background knowledge is complete and valid.

Constructive deduction

While studing properties of constructive induction (see next section) it occured to us that one could formulate a symmetric form of learning, which we call *constructive deduction*. This form uses background knowledge to deductively transfrom input information to a more abstract description, more general description or both. Creating a more abstract description is called *abstraction*; while creating a more general description by deduction is called *deductive generalization*. In both cases, learning involves applying truth-preserving rules of inference (domain-dependent or domain-independent) to the input information. Abstraction transfers a description of an entity from a more specific language to a less specific, in which certain details are ignored. Generalization extends the set of entities that are referred to in a description. These two processes often co-occur, and this is probably the reason why these two terms are sometimes confused.

For example, transforming a statement "My workstation has a Motorolla 25-MHz 68030 processor" to "My workstation is quite fast" is an abstraction. To make such a transformation, one needs (domain-dependent) background knowledge that a processor with the 25-MHz clock speed is quite fast, and therefore the computer can be viewed as quite fast. On the other hand, transforming a statement "John lives in Fairfax, Virginia" to "John lives in the United States" is a deductive generalization, because the set of locations where John lives is extended. This is a deductive generalization, because the resulting statement is a logical consequence of the initial input description and background knowledge. To make such a transformation one needs domain-dependent background knowledge that Virginia is a part of the United States, and that if somebody lives in some subarea, which is a part of a greater area, then the person also lives in the greater area. The last piece of knowledge is a special case of the transitivity of set membership, which is domain independent knowledge. For a more discussion of deductive generalization, see (Michalski and Zemankowa, 1989). Finally, consider a transformation of a statement "John has two cats, Kicia and Vicia, in his appartment" into "John has pets in his residence." This trasformation involves both abstraction and deductive generalization.

To simply characterize the difference between an abstraction and a generalization, consider an expresssion

$$d(A) = p \qquad (5)$$

which states that a descriptor (an attribute or predicate) takes value p for the set of entities A. Changing (5) to a statement in which d and/or p is substituted by a more abstract/general descriptor is an abstraction. Changing (5) to a statement in which A is replaced by a larger set is a generalization (can be deductive or inductive, depending on the meaning of d).

Inference rules used in constructive deduction may change the terms used in a description from low level observable concepts to highly abstract and/or general concepts. This way, constructive deduction is a vehicle for creating abstract descriptions. Notice, that constructive deduction generates knowledge that is a logical consequence of given premises (input information and the background knowledge), and therefore its pure form does not introduce elements of uncertainty. It can introduce uncertainty, if the rules of inference are plausible rather than crisp. Because constructive deduction uses deduction as the primary form of inference, it is classified as a form of analytic learning.

4.2 Synthetic Learning

In synthetic learning, the system strives to create desired knowledge by hypothesizing it through some form of inductive inference. Although the primary inference type involved is inductive, a synthetic learning process always involves also some deductive inference (e.g., to test whether a generated hypothesis accounts for an observation).

Unlike deduction, induction has been a subject of a long-standing debate, and different authors have defined it in different ways. One view is that it is merely empirical reasoning from particulars to universals without using prior knowledge. Another view is that induction includes every inference process under uncertainty, i.e., any inference that is not strictly deductive (e.g., Holland et al, 1986). These two views seem to be extreme points of a spectrum. The first one is too narrow, as it does not reflect the basic scientific thought on this subject going back to Aristotle, which characterizes induction as the fundamental inference underlying any process of creating new knowledge (see the reference under Aristotle). The second view seems to be overly general, as it includes processes such as approximate deduction. Our view is that induction is simply a process opposite of deduction. While deduction is a derivation of consequents from given premises, induction is a process of hypothesizing premises that entail given consequents. Strict deduction is truth-preserving, and strict induction is falsity-preserving. The intersection of these two types is tautological inference, which is both truth-preserving and falsity-preserving (i.e., equivalence preserving).

Empirical reasoning from particulars to universals, which we call *empirical inductive generalization*, is a simple, knowledge-poor, form of reasoning from effects to premises. As we show below, it can be viewed as a reasoning that traces backward certain *domain-independent* rules of inference ("generalization rules"; Michalski, 1983).

A more general form is *constructive induction*, which may trace backward both *domain-dependent* rules, as well as domain-independent background knowledge rules. In this formulation, constructive induction includes *constructive inductive generalization*, which uses BK rules to create higher level generalizations, *constructive inductive specialization*, which uses BK to hypothesize specializations (Michalski and Zemankowa, 1990; see an example below), and *abduction*, a form of reasoning introduced by Peirce in his classic and very influential treatise on Elements of Logic (see the reference under Peirce).

Abduction, as defined by Peirce, also called by him *retroduction*, is "the operation of adopting an explanatory hypothesis ...that would account for the facts (or some of them)." Here is an exerpt from his treatise (chp. II, sec.1):

> *The surprising fact, C, is observed;*
> *But if A were true, C would be a matter of course*
> *Hence, there is reason to suspect that A is true.*

Thus, abduction is creating a hypothesis that would entail the observation ("..if A were true, C would be a matter of course"). It is interesting to observe that this definition is equivalent to the definition of induction, if one interprets the undefined concept of "explanatory hypothesis" more broadly, namely that it can be in the form of a generalization. In this sense, this definition includes, as a special case (probably unintentionally), empirical inductive generalization. Clearly, a generalization of an observed fact must account for the fact. For example, suppose one observes that a particular painting of Polacci was sold very high, and hypothesizes that perhaps all paintings of Polacci are very expensive. If such a generalization is adopted as true, then the statement that the particular Pollaci's painting is expensive would be "a matter of course."

As we show below, in Peirce's abduction a hypothesis is created by "tracing backward" certain *domain-dependent rules*. Because constructive induction places no constraints on what type of background knowledge rules are employed, it can create causal explanations or other explanatory hypotheses, as well as inductive generalizations.

The initial formulation of constructive induction (Michalski, 1983) emphasized using domain knowledge to develop new concepts or attributes, beyond those supplied in the input. Depending on the type of domain knowledge involved, the new concepts so created can serve as explanatory hypotheses. Therefore, in general, the idea of constructive induction includes a knowledge-based creation of explanations, and this has led us to the use of the term "constructive induction" in the current form. Both, empirical inductive generalization and constructive induction can be viewed as forms of "reverse" reasoning, as opposed to deduction that can be viewed as "forward" reasoning. Therefore, we find it conceptually more attractive to consider them as two forms of inductive inference, rather than to view them as totally distinct forms of inference.

There is another issue related to abduction. Peirce was not very concerned with the issue of a preference criterion for choosing an explanatory hypothesis. This issue, however, is important when there is more than one hypothesis that explains the given facts. Thus, in general, a preference criterion has to be included as an important component of processes of creating hypotheses.

In view of the above, our general formulation of inductive inference is that, given an observation statement (OS) and background knowledge (BK), the reasoner searches for a hypothesis (H), consistent with BK, such that H & BK strongly or weakly entails OS, which we write as:

$$H \,\&\, BK \mid\!> OS \qquad (6)$$

and that the hypothesis satisfies a *preference criterion*. The preference criterion expresses the desirable properties of the hypothesis from the viewpoint of the reasoner's goals. For example, the reasoner may have a preference (or a bias) for a simpler hypothesis, and/or more plausible one according to the BK, and/or a hypothesis that uses concepts easy to test, etc. A preference criterion may also allow to generate an inconsistent and/or incomplete hypothesis, if such a hypothesis is more effective for its expected use. For example, we usually prefer to use the Newton's laws of motion, although they are, in general, less consistent with the facts than Einstein's theory.

By identifying H with explanatory hypothesis EH and OS with explanatory target ET, equation (6) becomes identical to (1), which characterizes the concept of explanation. Thus, the above shows that induction can be viewed as a process of creating explanations that satisfy some preference criterion. We distinguish between two basic types of induction:

- *Empirical induction*, in which the system creates an inductive hypothesis primarily on the basis of the given facts, that is without much use (or need) of domain-dependent background knowledge BK. Empirical induction involves primarily domain-independent generalization rules. Domain-dependent knowledge plays only a supportive role, for example, that of providing the constraints on the set of possible attribute values, specifying relations that hold among these attribute values and influencing the preference criterion.

- *Constructive induction (knowledge-based induction)*, in which the process of creating a hypothesis depends strongly on the domain-dependent background knowledge, as well as domain-independent.

In the literature, the terms empirical induction and inductive generalization are often viewed as equivalent. This view is not correct, because inductive generalization can not only be empirical, but also constructive, that is, it may involve a significant amount of domain knowledge. For example, creating a general scientific theory describing a class of entities (e.g., creating a physical law) is a form of inductive generalization, but may not be what we would call an empirical induction, because it may involve concepts far beyond the observables. Another example is a generalization of the statement "I saw John in his office on Monday, Wednesday and Sunday evenings" to "John is an unusually hard working employee." The second statement is a constructive inductive generalization, but strictly speaking is not an empirical induction. This is because to generate such a hypothesis one also needs to know about the work of other employees and to know that being in one's office in the evenings means working beyond normal expectations.

Finally, one should note that induction and generalization are two different processes. As indicated earlier, just as induction does not always produce a generalization, generalization is not always inductive (Michalski and Zemankova, 1989).

Let us now consider in greater detail empirical and constructive inductive learning.

Empirical inductive learning

In empirical inductive learning, BK is small and inadequate for constructing an explanatory structure for a given observation(s). The learner generalizes examples observed to create a consistent and complete description of them in terms of concepts used in describing observations (or closely related ones). Such a description implies the observed facts and, thus, can be viewed formally as an explanatory hypothesis, EH (an " empirical " generalization or explanation).
In machine learning, programs constructing empirical generalizations typically use only descriptive concepts that are selected from among those used in describing original observations. Such "surface" induction is called *selective induction*.

In learning from examples, P denotes a description that characterizes all positive examples and none of the negative examples (assuming that they are distinct). There can be many Ps which consistently and completely characterize a given finite set of examples, and therefore empirical learning needs some preference criterion for judging such admissible hypotheses. The essence of practical implementations of empirical learning is determining the simplest or most efficient expression for P.

The above describes a *crisp* empirical induction, which creates generalizations that strongly (or strictly) imply the observations. For example, after observing that John has come to various meetings punctually, one might hypothesize that he always comes to meetings punctually. The crisp empirical induction is falsity-preserving (if the input includes a false statement, the generalization is necessarily false). Another form is *soft* empirical induction that produces generalizations that weakly imply the observations. The latter form of empirical induction is not necessarily falsity-preserving. For example, observing someone coming late to a meeting several times, one might generalize that this person *usually* comes late to meetings.

Statements produced by empirical induction are usually not causal explanations, because they do not typically involve any causal relationships, but only correlations. They tend to be used, however, as explanations in everyday reasoning. For example, a person may ask, "Why is this tennis table green?" and someone may answer, "All tennis tables are green." This is not a "real" explanation, but people give such answers as "explanations."

Empirical induction has been the most active research area in machine learning, and there are many successful implementations of empirical learning programs. Most of them either generate rules [e.g., the AQ-based family of programs (Michalski, 1973)], or decision trees [the ID3-based family of programs (Quinlan, 1979)].

Constructive induction

In constructive induction (Michalski, 1983), the learner uses *domain-dependent* as well as domain independent background knowledge to hypothesize concepts and/or relations that characterize input information. The hypothesized concepts can be generalizations of the input facts, can be causal explanations of the facts, or they can be specializations of the input knowledge. If the engaged background knowledge involves causal dependencies that are "traced back," then the created hypothesis provides a causal explanation of the observation(s). If the input is general knowledge rather than specific facts, constructive induction involves using background knowledge to hypothesize lower level or more specific knowledge (which implies the more general one). To illustrate the latter, suppose that input information is that azalias grow in Virginia. From that general knowledge, one my hypothesize that azalias may also grow in Fairfax, a city in Virginia. This type of reasoning is called *inductive specialization* (Michalski and Zemankowa, 1989).

As we mentioned earlier, we view inductive inference as a general form of inference that includes empirical generalization and constructive induction. Such a view is consistent with a long tradition of science - starting with views of Aristotle, as expressed in his fundamental treatise *Posterior*

Analytics (see reference under Aristotle). Such a view is also quite satisfying intellectually, because it treats both empirical induction and constructive induction as different forms of reverse reasoning, namely as a reasoning from effects to premises that imply them. Such premises can be generalizations or causal descriptions. A simple form of constructive induction can be characterized as follows.

Given:

- Background knowledge consiting of domain-dependent rules

$$(\text{For all } p \in \mathbf{P},\ Q(e, p) \implies (\text{For all } t \in \mathrm{T},\ S(e, t)) \tag{7}$$

where $Q(e, p)$ and $S(e, t)$ are certain predicates,

and domain-independent rules used in empirical induction, such dropping a condition, climbing generalization tree, etc. (Michalski, 1983).

- Input $S(e, t_1)$, $S(e, t_2)$, $S(e, t_3)$,..., where $t_1, t_2, t_3, ... \in \mathrm{T}$.

Hypothesize:

$$\text{For all } p \in \mathbf{P},\ Q(e, p) \tag{8}$$

For example, suppose one believes that being well-organized, i.e., consistently well-organized over time, implies the ability to come to meetings punctually. If one observes John coming to several meetings punctually, then one might *constructively* hypothesize that John is well-organized.

As another example, suppose that one believes that being hardworking implies working after hours. If one sees several students from the AI Center working after hours a few times, then one might hypothesize that all students of the AI Center are hardworking. In these examples one can see how constructive induction may involve both empirical generalization (over the students), and an abduction of an abstract concept ("hardworking").

To show that the above form includes abduction, consider a classic example of abductive inference presented by Peirce (see reference under Peirce):

Given

BK: Location(bean, BAG) ==> Color(bean, white)

("Beans in this BAG are white")

Input: Color (Bean1, white)

("Color of Bean1 is white")

Determine

Hypothesis: Location(Bean1, BAG), i.e., "Bean1 is from the BAG".

As one can see, the above inference can be interpreted as "tracing backward" a domain-dependent rule. For another illustration of abduction consider, for example, the problem of recovery from failed proofs (Cox and Pietrzykowski, 1986; Duval and Kodratoff, 1990). In these works, the system abductively creates the minimal hypothesis needed for completing a proof by "tracing backward" certain domain knowledge rules.

In general, constructive induction is reasoning that may trace backward and/or forward certain domain-independent rules (e.g., rules of generalization), and/or domain-dependent rules (expressing domain knowledge), so that the result is a hypothesis that together with BK entails the initial input. Thus, constructive induction can be viewed as the most general form of induction, and abduction as a special type of such constructive induction.

A major limitation of inductive learning (empirical or constructive) is that it produces hypotheses that may be incorrect, because induction is not, in general, a truth-preserving inference. Even if the input facts are all correct, the produced generalization may not be correct. On the other hand, analytic learning, if it is based on strict deduction, guarantees that the improved knowledge is correct.

It may be interesting to point out a certain symmetry between synthetic (inductive) and analytic (deductive) methods. Analytic learning produces correct knowledge only to the extent to which the learner's initial knowledge (handcrafted into the system, or induced from cases) is *correct and complete*. If the initial knowledge is incorrect or incomplete, the results may be incorrect also. On the other hand, empirical inductive learning may also produce provably correct results. This is the case when the set of input facts (examples) is *correct and complete*, in the sense that it spans all representative examples (this does not necessarily mean the whole space of examples). Such a situation is described, for example, in (Michalski and Negri, 1977), where an inductive learning program produced provably correct rules for distinguishing between a win and draw positions in a chess endgame. Analytic and synthetic methods are not mutually exclusive, but are overlapping; methods that perform an equivalence-preserving knowledge transformation are both analytic and synthetic.

The uncertainty of inductive inference is a property inherently connected with any process of knowledge creation, including all scientific activities, and cannot be avoided in principle. The certainty of deduction is based on the certainty of the premises, but the premises have originally been created by induction.

5. A MULTICRITERION CLASSIFICATION OF LEARNING PROCESSES

Learning processes can be classified according to many criteria. Among particularily relevant criteria are the type of learning strategy used, the research orientation, the type of knowledge representation employed, the application area, etc. Classifications based on such single criteria have been discussed in (Carbonell, Michalski and Mitchell, 1983) and (Michalski, 1986).

This section proposes a classification of learning processes that is based on several interrelated criteria (Figure 2). In one general stucture, the classification shows basic characteristics of all major machine learning approaches and paradigms. Its primary purpose is to help the reader to get a general view of the whole field of machine learning.

As any classification, the classification can be judged by the degree to which it illustrates important distinctions and relations among various categories. The categories presented are not to be viewed as having precisely delineated borderlines, but rather as labels of central tendencies that can transmute from one to another by differently emphasizing various principal components. This interpretation reflects our view of multistrategy learning as an integration of basic inference processes, which are combined in different ways appropriately for the task. The classification criteria include the primary purpose of the learning method, the type of input information, the type of primary inference employed, and finally, the role of the learner's prior knowledge in the learning process.

As discussed above, from the viewpoint of the primary purpose, learning methods can be classified into synthetic and analytic. The primary purpose of synthetic methods is to create new or better knowledge. The primary purpose of the analytic methods is to transform the prior knowledge into a better form, so that is can better serve some goal. The knowledge so transformed does not allow the learner to solve more problems, but to solve them more effectively.

If the input to a synthetic learning method are examples classified by an independent source of knowledge, for example, a teacher, an expert, or an "oracle," then we have *learning from examples*. When the input are facts that need to be described or organized into a knowledge structure by the learner itself, then we have *learning from observation*. The latter is exemplified by learning by discovery, conceptual clustering and theory formation.

The primary type of inference used in synthetic learning is induction. As described in section 4.1, inductive learning can be empirical (BK-poor) or constructive (BK-intensive). Most work in empirical induction has been concerned with empirical generalization of concept examples using attributes selected from among those present in the descriptions of the examples (hence, such induction is sometimes called "selective" ; Michalski, 1983). Another form of empirical learning includes quantitative discovery, in which learner constructs a set of equations characterizing given data. Learning methods employed in neural nets or genetic algorithms are also viewed as forms of empirical inductive learning. They typically rely on relatively small amounts of BK, and their primary inference type is inductive. This inference, however, is not executed in an explicit way, like in typical symbolic methods, but in an implicit way.

In contrast to empirical induction, constructive induction is knowledge-intensive, as it uses BK to create high-level characterizations of the input information. This input information can be in the form of low level specific facts or already generalized descriptions. As described above, abduction can be viewed as a form of constructive induction, which "traces backward" certain domain-dependent knowledge rules.

For completness, let us mention that there are two other classifications of inductive methods, not shown in this classification. One is based on the way facts or examples are presented to the learner. If examples are presented and processed all at once, then we have one-step or non-incremental inductive learning. If they are processed one by one, or in portions, and the system may have to modify the hypothesis after each input, we have an incremental inductive learning.

The second classification is based on the method of interpreting or matching instances with concept descriptions. Mathing an instance with a concept description can be done in a direct way, or can employ a substantial amount of background knowledge and inference. For example, case-based or exemplar-based methods employ matching procedures that allow the system to recognize new examples that do not directly match any past example (e.g., Bareiss, Porter and Wier, 1990). Such a process can be characterized as a "dynamic" induction that is performed during the matching process (or the recognition process). Learning methods based on the two-tiered concept representation (Bergadano et al., 1990) also use a sophisticated matching procedure. In general, they can employ any type of inference in matching an instance with a concept representation (Michalski, 1990).

Analytic methods can be divided to those that are guided by an example in the process of knowledge reformulation (example-guided) and those that start with a specification (specification-guided). The former category includes explanation-based learning (e.g., DeJong et al. 1986), explanation-based-generalization (Mitchell et al., 1986), and explanation-based specialization (Minton, 1986; Minton et al., 1987). The primary inference method in analytic learning is deduction. If deduction is based on axioms ("domain theory"), then it is called axiomatic. Explanation-based generalization can be viewed as an example of an axiomatic method, because it is based on a pure deductive process that utilizes complete and consistent background knowledge. This knowledge playes the role analogous to the axioms in formal theories.

Analytic methods that involve deductive transformations of description spaces and/or imperfect background knowledge and/or plausible rules of deductive inference are classified as methods of "constructive deduction." This class also includes abstraction, as it utilizes background knowledge to create descriptions at a lower level of detail, while basically preserving the truth of the description. Results of abstraction are typically statements expressed in a higher level language.

Another form of constructive deduction is deductive generalization that creates more general statements, in the sense that they include more entities. Such statements are logically implied by the source statements, in contrast to statements generated by inductive generalization, which imply the source statements. These two processes are put into a dotted rectangle, to indicate they do not seem to correspond to any major current research areas. They are simply suggested as potential research areas, as a result of making the above classification. This can be viewed as a kind of the "Mendeleiev periodic table effect."

In general, constructive deduction is a knowledge-based process of transforming descriptions from one representation space or language to another, which preserves information important for an assumed goal. Abstraction is classified as a constructive deduction, which transforms a description at a high level of detail to a description at a low level of detail, while preserving the truth of the relations and/or properties relevant to the assumed goal. In other words, while reducing the total information content of the original description, abstraction preserves the information important to performing an implicitly or explicity defined goal. Depending on the goal, a given description can be abstracted in many different ways. Each such process is essentially deductive, as it is not supposed to introduce or hypothesize any information that is not contained in the initial description or information source, or which cannot be deductively inferred from it using the learner's BK. The difference between constructive deduction and what we call axiomatic deduction is that the former emphasizes a change in the representation space or language, and may use a variety of knowledge transformations, rather then strictly logic-based formal methods. A "pure" constructive deduction is truth-preserving; however, in general, it can involve rules of plausible reasoning, and in this case ceases to be truth-preserving.

As mentioned before, abstraction is sometimes confused with generalization. Note that generalization transforms descriptions along the set-superset dimension and may be falsity-preserving, as in the case of inductive generalization, or truth-preserving, as in the case of deductive generalization (Michalski and Zemankowa, 1990; see also below). In contrast, abstraction transforms descriptions along the level-of-detail dimension, and is truth-preserving with regard to the characteristics of the entity(ies) important for the assumed goal. While generalization often uses the same description space (or language), abstraction typically involves a change in the representation space (or language). The reason why generalization and abstraction are frequently confused may be attributed to the fact that many processes include both of them.

Deductive generalization is concerned with making generalizations that are logical deductions from the base knowledge. It differs from abstraction as it moves from considering a set to considering a superset, and typically uses the same representation formalism. For example, transforming a statement " George Mason University is in Fairfax" to "George Mason University is in Virginia" is a deductive generalization. In contrast, changing a high resolution digitized satellite image of Fairfax into a low resolution image is a simple form of abstraction. A more sophisticated abstraction would be to use the high resolution image and appropriate BK to create a map of Fairfax, which emphasizes important (according to the goal) aspects of the area. Research on problem representation, transformation of problem representation spaces, determination of a representative set of attributes, deductive transformation of a knowledge base, and related topics can be classified under the rubric of constructive deduction.

Systems that combine several basic strategies are called *multistrategy learning systems.* In parallel to multistrategy systems, one can also distinguish multirepresentation learning systems (not shown in the classification). Such systems would employ various forms of constructive deduction or induction to create and use representations at different levels of abstraction, and/or apply different description languages in the process of learning. The use of these descriptions and languages would depend on the task at hand and on the application domain. Such systems thus are capable of changing the representation of the original problem statements. The importance of this area has been acknowledged very early by pattern recognition researchers (Bongard, 1970), as well as by AI researchers (Newell, 1969; Amarel, 1970). Nevertheless, it received relatively little attention during recent years. Among notable exceptions are (Amarel, 1986; Mozetic, 1989)

Summarizing, one can distinguish three pairs of reasoning and learning mechanisms. Each pair contains two opposite processes, and is concern with different aspects of reasoning and knowledge transformation. Two of these pairs have been relatively well-explored in machine learning: deduction/induction and generalization/specialization. The third pair, which has been relatively less studied, consists of abstraction and its reverse, which may be called *concretion* (Webster's dictionary defines it as being a process of concretizing something).

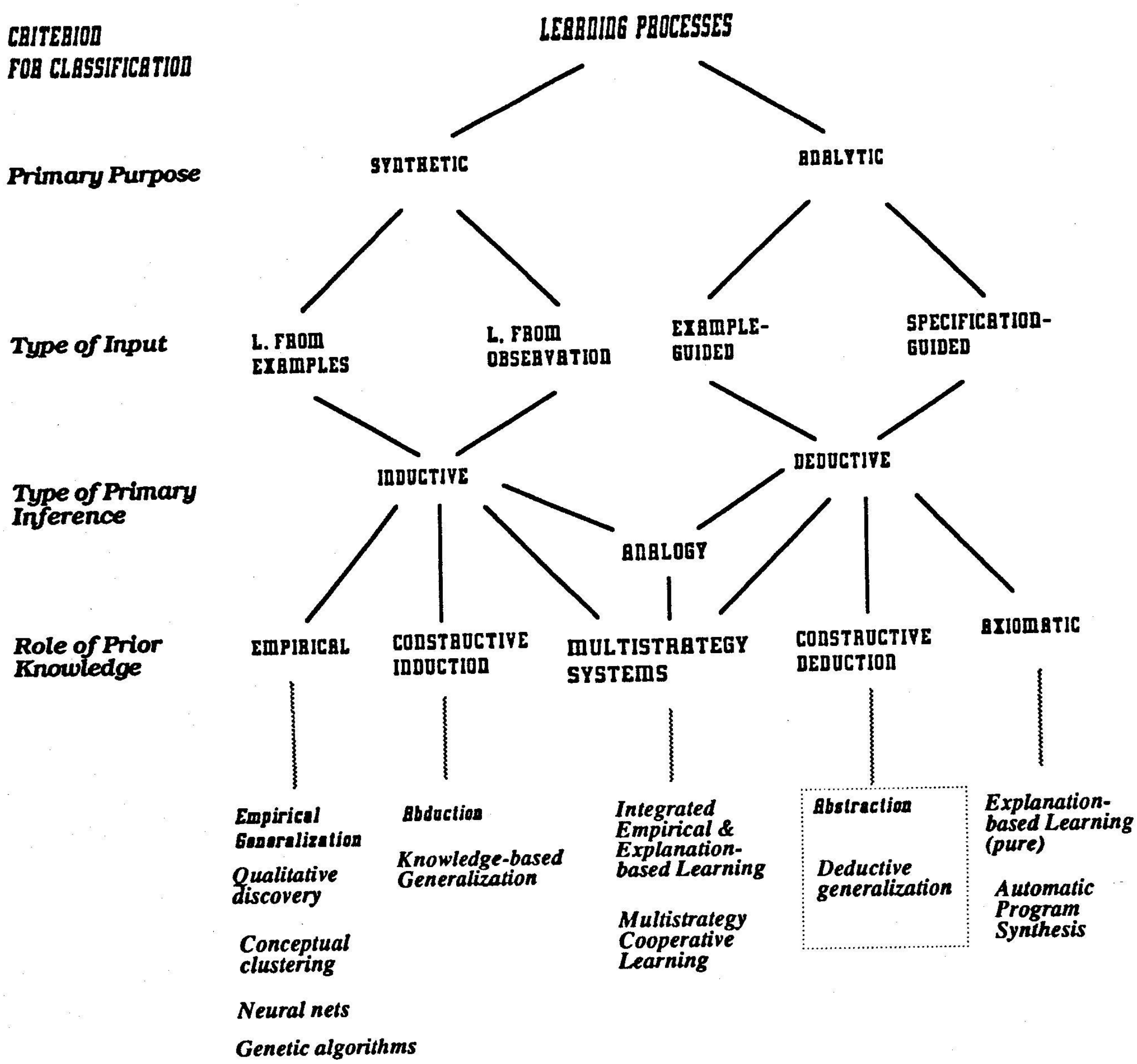


Figure 2. A multicriterion classification of learning processes.

These three types of mechanisms can be combined in different ways, giving some classical, well known reasoning mechanisms and some less known. The classical ones include inductive generalization and deductive specialization. Less investigated are inductive specialization, abstraction, deductive generalization, inductive concretion and other.

The above "grand" classification appears to be the first attempt to characterize and relate to each other all major methods and subareas of machine learning within one general scheme. As such, this attempt may suffer from various weaknesses and imprecision, and can be criticized on various grounds. Its primary purpose is to try to help the reader, especially a novice in this field, to view different learning mechanisms and paradigms as parts of a one general structure, rather than as a collection of unclearly related components and research efforts. By analyzing this classification, the reader may be stimulated to improve it or to develop a new, more adequate one.

6. MULTISTRATEGY COOPERATIVE LEARNING

The ideas presented above have shown the relationships among different forms of explanation and different types of learning. They have shown, in particular, the relationship between the two most active and complementary methodologies for building learning programs: empirical learning, which primarily exploits data, and analytical learning, which primarily exploits prior knowledge. While both these methodologies are useful for some domains of application, most practical learning problems seem to fit neither the empirical nor the analytic paradigm. This is because most practical problems involve to a significant extent both prior knowledge and new facts, and the prior knowledge is often incomplete and/or not totally correct.

This section discusses a general approach to learning that attempts to unify several learning approaches and to build a learning system of much greater capability than those using only one type of approach. The proposed *multistrategy cooperative learning* integrates empirical learning, constructive induction, learning by instruction, explanation-based learning and conceptual clustering. Ultimately, it is intended to integrate also learning by analogy and case-based reasoning (which can be viewed as a form of analogical reasoning).

Given a fact or an observation, one can distinguish five types of relationship between the fact and the learner's prior knowledge. First, the fact may be new or partially new to the learner, neither confirming nor disconfirming the learner's prior knowledge, or, if it is not economical to test for this property, one assumes that the fact is new. Second, the fact may contradict some segment (a rule or a rule set) of the learner's prior knowledge. Third, the fact may be implied (or may imply) some segment of the learner's prior knowledge. Fourth, the fact may be similar in certain respects (in terms of abstract relations, rather than low level attributes) to some segment of the learner's knowledge. Fifth and finally, the fact may be already known to the learner (i.e., strictly match some knowledge segment).

Current empirical and constructive induction systems are concerned primarily with handling the first and the second cases. "Pure" explanation-based learning is concerned with handling the third case. The more recent methods of explanation-based learning attempt to address situations in which the learner's knowledge is insufficient (first case), or inconsistent with the prior knowledge (second case), or the prior knowledge is intractable (first case). Learning by analogy and case-based reasoning are concerned with handling the fourth case. Very few symbolic learning systems handle the fifth case other than by ignoring such inputs (Slimmer and Granger, 1986). In neural networks and genetic algorithms a repetition of the input is not ignored, but those systems do not have the ability to recognize that the input is repeated.

Our work on the MCL learning methodology is intended ultimately to handle all five cases in an integrated fashion. Thus, to explain how this methodology works, one needs to explain how it would handle all these cases. Before we move to this topic, however, we first need to introduce the knowledge representation to be used in the proposed methodology.

Knowledge representation

A multistrategy cooperative learning system (briefly, an *MCL learner*) has to be able to represent and use knowledge created by different learning strategies. This means, in particular, that it has to be able to employ knowledge created by one learning process as an input to another learning process. The other learning process may be using the same learning strategy or a different strategy. As mentioned earlier, learning with such a property is called the closed-loop learning.

Another property of a constructive learner concerns its reaction to repetitive information. Traditional symbolic learning methods typically use a knowledge representation (e.g., rules or semantic networks) that does not change if a new instance repetitively satisfies the given concept description. A constructive learner needs a representation that could change even if an instance is shown to satisfy a concept description. The reason is that such cases should be used for increasing a degree of confirmation of the concept description. An MCL learner should also be able to decrease such a degree in some situations.

To satisfy the above requirements, we assume that the basic component of the learner's background knowledge (BK) is a *parameterized association rule* (PAR), whose general form is:

CTX: AS: L-PREMISE <---> R-PREMISE: *M-Parameters* (9)

where

L-PREMISE and R-PREMISE denote the left and right sides of a PAR, respectively. They are expressed as conjunctive statements or terms. The statements may include *internal disjunction* and terms may be *compound* (Michalski, 1983).

<---> denotes a bi-directional *association*, which is instantiated into a more specific relationship according to AS.

AS stands for the *association specification*, which defines the type and properties of the association. An association can be instantiated to many specific types and defined with different degree of precision. For example, AS may state that the association is an implication between statements or a mutual dependency between terms. A term dependency is, e.g., that "smoking is related to lung cancer." When more knowledge about this dependency is obtained, the association may become a functional dependency, e.g., that "smoking two or more packs a day shortens the life span by 10 years on the average." In general, the association may be a strong (logical) or weak (plausible or probabilistic) implication or equivalence, mutual dependency between terms, equality, correlation, causal relationship, decision assignment, precedence relation, and other. AS may include a quantification expressed in the from of an ordinary quantifier or a *numerical quantifier* (Michalski, 1983). A numerical quantifier may state, e.g., that there are two or more objects in a set S to which the PAR applies or that there are specifically three objects in S to which the PAR applies. When AS is not specified, the association takes a default meaning. The default meaning may be that if the premises are statements, then the association is implication; and if the premises are terms, it is *mutual dependency* (Collins and Michalski, 1989).

CTX denotes a *context* for the association, that is a characterization of the conditions under which the PAR applies. When the CTX is not specified, a default context is assumed.

M-Parameters (*merit* parameters) represent numerical or qualitative properties of the association, which characterize its strength in both directions, the number of times the association has been satisfied or not satisfied by input events, and other parameters, such as those introduced in the theory of plausible reasoning (Collins and Michalski, 1989). Each time a PAR is evoked and either satisfied or not satisfied, the appropriate parameters are updated.

An input fact may match the whole or part of either premise, or both premises of a PAR. For example, a description of an object may match the left premise, and its classification by a teacher may match the right premise. PARs are organized into *segments*. A segment is collection (a "parset") of rules that are related in some way. For example, a segment may describe a single concept or a few closely related ones. Segments may be (statically or dynamically) organized to larger units, called *schemas*.

This form of knowledge representation is based on ideas stated in the theory of human plausible reasoning proposed by Collins and Michalski (1989), and in the *annotated predicate calculus* (Michalski, 1983). In particular, the PAR is a generalization of the rules for mutual implication and term dependency. The concepts used in PARs (attributes, relations, etc.) are organized into *dynamic hierarchies*, such as described in that theory.

Although PARs are significantly more general than productions used in genetic algorithms (e.g., Holland, 1986 and 1987), they share with them the property of having some numerical score(s) attached to them. PARs permit one to represent a large class of descriptions and relationships. For example, a single PAR may represent a condition-action rule, such as "if the second valve is broken, call a mechanic"; a term dependency, such as "pressure and volume are inversely proportional "; a causal relationship, such as "if the pressure goes beyond 3 atmospheres, this indicator will move up"; a quantified implication, such as " 60% of objects observed have property P", as well as ordinary implications or equivalences.

Outline of the method

We will now outline our preliminary ideas about how an MCL learner might learn in different situations, in particular, how it could react to the above described different types of relationship between an input, background knowledge (BK) and a learning goal. We assume that the input consists of information (e.g., a fact, a concept example or a rule) supplied by an external source, or information resulting from an impass in processing of an input according to some strategy. In the latter case, processing of the input may involve activating another learning strategy. For example, in the process of determining if a fact is implied by BK (i.e., in attempting to explain the fact), the learner finds that some parts of it are explainable by BK, and some other parts may represent new information. The parts that are explainable are processed by an analytical learning strategy, and the parts that are new may activate a synthetic learning process.

We assume that the general learning goal (a default goal) is to derive any "useful" information from the input, make sense of it and assimilate it into the knowledge base. More specific learning goals, such as to generalize facts to generate a rule, to create a conceptual classification of facts, to reformulate a part of BK into a more efficient knowledge, to determine new knowledge on the basis of an analogy between the input and some past knowledge, etc. are supplied from a supervisory control system.

Presented ideas are concerned only with aspects of building or updating a knowledge base, and not with issues of using the knowledge for problem solving. Given input information, the learner determines which of the five cases above ("processing methods") is involved. The rules and segments in BK are indexed in various ways to facilitate this process. The learner performs a "deductive" matching of the information with BK to determine if it satisfies (or is satisifed by) some rule, or at least is consistent with the rules. Such a matching is called "deductive" because it may involve several steps of deduction.

A limited amount of resources is available for this process, and if they are exceeded, a failure is communicated. In such a case, the information is assumed to be (pragmatically) new to the system. Rule generalization or specialization is done by applying various inference rules, such as those described in (Michalski, 1983). Any input is first evaluated for "relevance" to the learner's goal(s). Such an evaluation is based on a quick classification of the input to some category, and the category is related to the goal(s). If the input passes such a "relevance test," a learning process is activated.

1. The input represents pragmatically new information

In this case, the learner searches for the part of BK that is "sufficiently related" to the input information. For example, it may be a part describing the concept being exemplified by the input.

If this effort succeeds, the relevant part is generalized, so that it accounts for this information and possibly other information that was stored previously. The input is also evaluated for "importance" and, if it passes an *importance criterion*, it is stored (this is called learning with partial memory of the past).

If there is no knowledge "sufficiently related" to the input, the input is stored without involving the importance criterion, and the control is passed to strategy 4. In general, this case handles situations that require some form of synthetic learning (empirical learning or constructive induction), or merely learning by instruction.

2. The input contradicts some part of the learner's background knowledge

The system identifies the part of BK that is contradicted by the input information, and then attempts to modifies this part. If this modification involves too much restructuring, and/or the confidence in the input is low, no change to this part of BK is made, but the input is stored. When some part of BK has been restructured to accommodate the input, the input is also stored, but only if it passes an "importance criterion." If contradicted knowledge is a specific fact, this is noted, and any knowledge that was generated on the basis of the contradicted fact may have to be revised. In general, this case handles situations requiring a correction of BK through some form of synthetic learning and, generally, managing inconsistency.

3. The input is implied by, or implies a part of the background knowledge

This case represents a situation when it is determined that there is a part of BK that accounts for the input, or is a special case of it. The learner creates a derivational explanatory structure that links the input with the involved BK part. Depending on the learning goal, this structure can be used to create a new ("operational") knowledge that is more adequate for future handling of such cases. If the new knowledge passes an "importance criterion," it is stored for future use. This mechanism is related to the ideas on the utility of explanation based-learning (Minton, 1988).

If the input represents a "useful" result of a problem solving activity, e.g.," for given state x, it was found that the best action is y", then storing such a fact as a rule is similar to chunking in SOAR (Laird, Rosenbloom, and Newell, 1986). If the input information (e.g., a rule supplied by a teacher) implies some part of BK, then an "importance criterion" is applied to it. If the input passes this criterion, it is stored, and an appropriate link is made to the part of BK that is implied by it. In general, this case handles situations requiring some form of analytic learning, in particular, explanation-based learning.

4. The input evokes an analogy to a part of the background knowledge

This case represents a situation when the input does not match any prior fact or rule exactly, nor is closely related to any part of BK in terms of low level properties, but there is a similarity between the fact and some part of BK at a higher level of abstraction. That is, unlike in case 1, in which the system tries to directly match the fact with a knowledge segment, in this case, the matching is done using abstract attributes or relations. An analogy is established, and explored in the context of the learner's goals. For example, an input describing a lamp may evoke an analogy to the part of BK describing the sun, because both lamp and sun match in terms of an abstract attribute "produces light." Knowing that the sun produces heat, and that there is often a mutusal dependency between light and heat (Collins and Michalski, 1989), a learner may hypothesize that the lamp may also produce heat.

5. The input is already known to the learner

This case occurs when the input matches exactly some part of BK (a stored fact, a rule or a segment). In such a situation, a measure of confidence associated with this part is updated.

In summary, in multistrategy cooperative learning, any act of receiving information may activate a learning process. The learner employes deductive inference when an input fact is consistent with, implies, or is implied by the prior knowledge, analogical inference when it is similar to some part of past knowledge, and inductive inference when there a need to generalize knowledge or hypothesize new knowledge. It also learns when input facts confirm its knowledge, by reinforcing current beliefs.

7. SIMPLE EXAMPLE: Learning the Concept of a Cup

To illustrate briefly some of the ideas described above, let us use a well-known example of learning the concept of "cup" (Mitchell, Keller and Kedar-Cabelli, 1986). The example is deliberatly oversimplified, so that major ideas can be presented in a very simply way (Figure 3).

The top part of the figure presents an abstract concept definition (abstract CD) for the concept "cup," the domain rules, a description of an example of a cup (specific object description or specific OD), an abstract object description (abstract OD), and an operational concept description (operational CD). An abstract concept definition describes the concept of "cup" in abstract/general terms, while an abstract object definition describes the specific object in such terms.

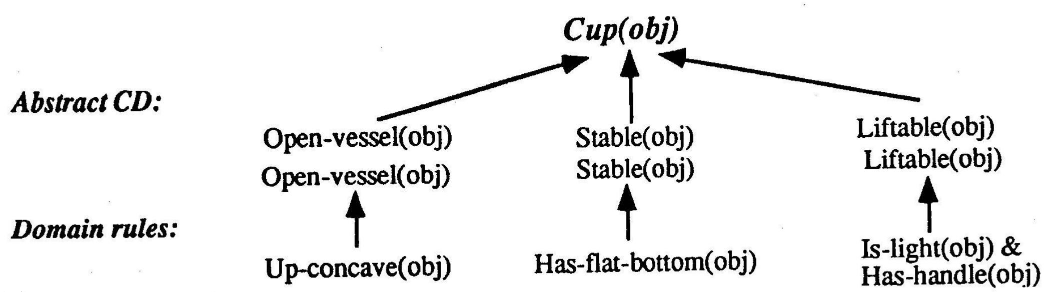
The bottom part of the figure summarizes information that is assumed to be given and to be learned using different learning approaches: constructive deduction (abstraction), explanation-based learning, empirical induction, constructive induction (in cases of generalization and abduction), and the proposed multistrategy cooperative learning. For simplicity, some details are omitted, and the example does not illustrate the mechanism of updating the strength of the rules, nor analogical reasoning. Figures 4, 5 and 6 give more details about some of the learning processes, specifically, about abstraction, constructive generalization and abduction.

A more practical, but less general example is described in (Ko and Michalski, 1989). It shows how a system learns a general schema for creating a plan for putting together simple assemblies, for example, a bell. The schema is developed by an incremental improvement and testing of intermediate schemas.

8. CONCLUSION

The aims of this work are to create a theoretical framework for characterizing and unifying basic learning strategies, and to develop an experimental integrated learning system based on it. The proposed MCL methodology stems from the inference-based theory of learning that considers learning as an inference process, whose useful results are stored for future use. This process involves input information, the learner's prior knowledge, and the goal of learning. It may employ any kind of inference - deductive, analogical or inductive. Among important assumptions for this work are that a learning system should be capable of acquiring knowledge from any input and be able to use knowledge gained in one learning task in any new learning task (i.e., be capable of the "closed-loop" learning).

The MCL methodology is intended ultimately to include capabilities for empirical learning, chunking, constructive induction, learning by instruction, reinforcement learning, explanation-based learning, conceptual clustering, learning by analogy and case-based reasoning.



Example (Specific OD):
Up-concave(CUP1) & Has-flat-bottom(CUP1) & Is-light(CUP1) & Has-handle(CUP1) & Color(CUP1) = red& Owner(CUP1) = RSM & Made-of(CUP1) = glass &...<---> Cup(CUP1)

Abstract OD:
Open-vessel(CUP1) & Stable(CUP1) & Liftable(CUP1) & ... <---> Cup(CUP1)

Operational CD:
Up-concave(obj) & Has-flat-bottom(obj) & Is-light(obj) & Has-handle(obj) <---> Cup(obj)

	Given:		To be learned:
Constructive Deduction (Abstraction)	Example Domain rules	$\triangleright$	Abstract OD
Explanation-based Learning	Abstract CD Domain rules Example	$\triangleright$	Operational CD
Empirical Induction	Examples Partial BK'	$\mid\!<$	Operational CD
Constructive Induction (Generalization)	Domain rules Example(s)	$\mid\!<$	Abstract CD
Constructive Induction (Abduction)	Example(s) Abstract CD	$\mid\!<$	Domain rules
Multistrategy Cooperative Learning	Any of the above and other combinations, depending on what is the input, what the learner knows already and what is to be learned		

OD and CD denote object and concept descritpion, respectively. CUP1 stands for a specific cup; obj denotes a variable. BK' denotes some partial background knowledge, e.g., a specification of the value sets of the attributes and the type of the attributes. Operators $\triangleright$ and $\mid\!<$ denote deduction and induction, respectively.

Figure 3. Learning various aspects of the concept of "cup" using different strategies.

Learning Process: EX&DR --> OD

Constructive Deduction (Abstraction)	Example Domain rules	$\mid\!<$	Abstract OD

Given:

1. INPUT

New Example:

Up-concave(CUP1) & Is-light(CUP1) & Has-handle(CUP1) & Owner(CUP1)=RSM & Color(CUP1)=red & Made-of(CUP1)=glass & Has-flat-bottom(CUP1) <----> Cup(CUP1)

2. BACKGROUND KNOWLEDGE

Domain rules

Open-vessel(obj) $\uparrow$ Up-concave(obj)

Stable(obj) $\uparrow$ Has-flat-bottom(obj)

Liftable(obj) $\uparrow$ Is-light(obj) & Has-handle(ot

Other Relevant Knowledge

Container(obj) <---- Open-vessel(obj) & Stable(obj)

....

3. GOAL

To derive an abstract description of this example.

STEPS:

1. Determine relevant domain rules
2. Apply the rules to the given example and create an abstract OD

Learned:

An abstract OD:

Cup(CUP1) <---> Open-vessel(CUP1) & Stable(CUP1) & Liftable(CUP1) &...

After applying other relevant knowledge, even more abstract OD can be created:

Cup(CUP1) <---> Container(CUP1) & Liftable(CUP1) &...

Figure 4. An illustration of abstraction.

Learning Process: **EX&DR --> AC**

Constructive Generalization Example(s) Domain rules |< Abstract CD

Given:

1. INPUT

Examples::

Up-concave(CUP1) & Is-light(CUP1) & Has-handle(CUP1) & Owner(CUP1)=RSM & Color(CUP1)=red & Made-of(CUP1)=glass & Has-flat-bottom(CUP1) <----> **Cup(CUP1)**

Up-concave(OBJ2) & Is-heavy(OBJ2) & Has-handle(OBJ2) & Owner(OBJ2)=RSM & Color(OBJ2)=grey & Made-of(OBJ2)=wood & Has-flat-bottom(OBJ2) <----> **Jar(OBJ2)**

Up-concave(OBJ3) & Is-light(OBJ3) & Made-of(OBJ3)=glass & Has-flat-bottom(OBJ3) No-handle(obj) <----> **Jar(OBJ3)**

2. BACKGROUND KNOWLEDGE

Cup(obj)

Unknown Abstract CD

Open-vessel(obj) Stable(obj) Liftable(obj)

Open-vessel(obj) Stable(obj) Liftable(obj)

Domain rules

Up-concave(obj) Has-flat-bottom(obj) & Made-of(obj)=hard-material Is-light(obj) & Has-handle(obj)

Other relevant knowledge

Made-of(obj)=hard_material <---- Made-of(obj)=glass
Made-of(obj)=hard_material <---- Made-of(obj)=wood
Is-light(obj) =/= Is-heavy(obj)

3. GOAL

To create an abstract description of the concept of cup.

STEPS:

1. Analyze the relationship between the input and the BK in the context of GOAL
2. If the current abstract description of the cup BK is incomplete, hypothesize additional rule(s), to make it complete

Learned:

An abstract concept description:

Cup(obj) <---- Open-vessel(obj) & Stable(obj) & Liftable(obj)

Figure 5. An illustration of constructive generalization

Learning Process: **EX&AC --> DR**

Abduction Example(s) Abstract CD |< Domain rules

Given:

1. INPUT

An example:
Up-concave(CUP1) & Is-light(CUP1) & Has-handle(CUP1) & Owner(CUP1)=RSM & Color(CUP1)=red & Made-of(CUP1)=glass & Has-flat-bottom(CUP1) <----> Cup(CUP1)

2. BACKGROUND KNOWLEDGE

Abstract CD

Cup(obj)

Open-vessel(obj) Stable(obj) Liftable(obj)

Incomplete domain rules

Open-vessel(obj) Liftable(obj)

Up-concave(obj) Is-light(obj) & Has-handle(obj)

Other relevant knowledge

Stable(obj) <~~/~~~> Owner(obj) & Color(obj) (no mutual dependency)
Stable(obj) <~~~~> Type-of-bottom(obj) (mutual dependency)
Type-of-bottom(obj) = {uneven, flat, leg-supported,...}
Type-of-bottom(obj) = <value> ==> Has-<value>-bottom(obj)

3. GOAL
To determine domain knowledge that justifies the abstract concept definition.

STEPS:
1. Analyze the relationship between the input and the BK in the context of GOAL
2. If the BK is insufficient, hypothesize additional domain rule(s), that are consistent with example and BK.

What is learned :

A new domain rule:
Stable(obj) <== Has-flat-bottom(obj)

Figure 6. An illustration of abduction.

An important component of the MCL methodology is the cognitive theory of plausible reasoning (Collins and Michalski, 1989), which provides a formal structure for implementing various forms of such reasoning. In the complete implementation of the MCL, plausible reasoning is supposed to play a double-level function. The first-level function is to generate lines of reasoning that relate the input information to the learner's background knowledge and goals, and determine the most plausible conclusions. These conclusions are to be stored as results of learning, and integrated with the rest of the learner's knowledge. The second-level function is to generate, on request, explanations of the results of learning in terms of high-level human-oriented concepts and structures. It is our strong believe that an advanced learning system should not only be able to learn, but also to explain to human counterparts what knowledge it acquired or modified during any learning process. When a learning system is a part of a knowledge system (e.g., an expert system), the explanatory capabilites for learning should be integrated with explanatory capabilities for the system's performance. It may be worth mentioning, that while a significant amount of research has been done on the development of explanatory capabilities for performance of knowledge-based systems (e.g., Tanner, 1990), relatively little has been done so far on the development of explanatory capabilites for learning systems.

To demonstrate some aspects of MCL learning, a prototype system, called NOMAD, has been implemented (Ko, 1989). NOMAD is a planning system that learns from planning experiences, and has been developed in connection with the Intelligent Explorer (IEX) project at the GMU Center for Artificial Intelligence. In the future work we plan to explore the utility of the INDUCE 4 program for incremental structural learning (Mehler, Bentrup, and Riedesel, 1986), and the DISCIPLE integrated learning system (e.g., Tecuci and Kodratoff, 1990) for implementing an MCL system.

The presented ideas are at an early state of development, and many issues have not been resolved. For example, future resaerch should address the question of the development of a flexible control of the execution of different learning strategies, handling input information whose different components need to be processed separately, but in a globally coordinated way, the access and manipulation of a large collection of parameterized association rules (PARs), or the methods for updating and using various parameters associated with PARs. Future research may also explore the utility of genetic algorithms in the MCL methodology. A genetic algorithm might be used, e.g., for evolutionary optimization of many parameters involved in MCL.

In closing, our goals in developing the MCL methodology are to explore research issues involved in the integration of different learning strategies, and to understand how various strategies can best be utilized in different learning situations. This understanding may give insights into learning processes in general, and help to build more powerful and efficient multistrategy learning systems.

Such systems are needed for many practical tasks in which a learning system starts with inadequate knowledge, and needs to use facts or experience to extend or improve it in a goal-oriented way. There are two general areas where such systems may be particularily useful: extraction of knowledge from large databases and knowledge acquisition for expert and advisory systems. In both these areas, to derive useful knowledge no single learning strategy is usually sufficient, and knowledge learned must be understandable by a human user. Among specific application tasks one can list all kinds of diagnostic problems, decision making, planning systems, system design, economical prediction, resource management, robot navigation, automated assembly, and sensory signal interpretation.

Acknowledgements

The author expresses his gratitude to Hugo De Garis, Ken De Jong, Kejitan Dontas, Bob Giansiracusa, Heedong Ko, Yves Kodratoff, David Littman, Elizabeth Marchut, Pawel Stefanski, Gheorge Tecuci, Janusz Wnek and Jianping Zhang for many insightful comments and suggestions that significantly helped to improve earlier versions of the paper. Thanks also go to Gara Barham and Janet Holmes for stylistic suggestions and proofreading.

This research was supported in part by the Defense Advanced Research Projects Agency under the grant administered by the Office of Naval Research No. N00014-K-85-0878, and in part by the Office of Naval Research under grants No. N00014-88-K-0397 and No. N00014-88-K-0226.

References

Adler, M. J., Gorman (Eds.) The Great Ideas: A Synoptic of Great Books of the Western World, Vol. 1, Ch. 39, *Encyclopedia Britannica*, 1987.

Aristotle, Posterior Analytics, in *Great Books of the Western World*, vol.8, R.M. Hutchins (ed.), Encyclopedia Britannica, Inc. 1987.

Amarel, S., "Program Synthesis as a Theory Formation Task: Problem Representations and Solution Methods," in *Machine Learning: An Artificial Intelligence Approach Vol.II*, Morgan Kaufmann, Los Altos, CA, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), 1986.

Aristotle, *Posterior Analytics*, in *The Works of Aristotle Volume 1*, R. M. Hutchins (Ed.), Encyclopedia Britannica, Inc., 1987.

Bareiss, E. R., Porter, B. and Wier, C.C., PROTOS, An Exemplar-based Learning Apprentice, in *Machine Learning: AN Artificial Intelligence Approach vol. III*, MOrgan Kaufmann, 1990.

Barto, A.G., and Anderson, C.W., "Structural Learning in Connectionist Systems," *Proceedings of the 7th Conference of the Cognitive Science Society*, pp. 43-53, 1985.

Bergadano, F., Matwin, S., Michalski, R.S. and Zhang, J., Learning Two-tiered Descriptions of Flexible Concepts: The POSEIDON System, *Machine Learning and Inference Raports, No. MLI-3*, Center for Aritificial Intelligence, George Mason University, 1990.

Bierman, A. W., Guiho, G., and Kodratoff, Y. (Eds.), *Automatic Program Synthsis Techniques*, Macmillan Publishing Company, New York, pp. 517-552, 1984.

Carbonell, J. G., Michalski R.S. and Mitchell, T.M., An Overview of Machine Learning, in *Machine Learning: AN Artificial Intelligence Approach*, Michalski, R.S., Carbonell, J.G., and Mitchell, T. M. (Eds.), Morgan Kaufmann Publishers, 1983.

Collins, A., and Michalski, R.S., "The Logic of Plausible Reasoning: A Core Theory," *Cognitive Science*, Vol. 13, No. 1, pp.1-49, 1989.

Cox, P.T., Pietrzykowski, T., "Causes for Events: Their Computation and Applications" *Proceedings of the Eighth International Conference on Automated Deduction*, Oxford, 1986.

Danyluk, A.P., "The Use of Explanations for Similarity-Based Learning," *Proceedings of IJCAI-87*, pp. 274-276, Milan, Italy, 1987.

Danyluk. A. P., "Recent Results in the Use of Context for Learning New Rules," Technical Report TR-98-066, Philips Laboratories, 1989.

Davis, D., *Genetic Algorithms and Simulated Annealing*, Pitman Press, Cambridge, MA, 1987.

DeJong, G. and Mooney, R., "Explanation-Based Learning: An Alternative View," *Machine Learning Journal*, vol 2, 1986.

DeJong, G.F., "An Approach to Learning from Observation," in *Machine Learning: An Artificial Intelligence Approach, Vol. II*, Morgan Kaufmann, Los Altos, CA, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), pp. 571-590, 1986.

DeJong, G. F. (Ed.), *Proceedings of the 1988 Symposium on Explanation-Based Learning*, Stanford University, March, 1988.

DeJong, K.A., and Spears, W., "Using Genetic Algorithms to Solve NP-Complete Problems," *Proceedings of the 3rd International Conference on Genetic Algorithms and their Applications*, Fairfax, VA, 1989.

de Kleer, J., "An Assumption-Based Truth Maintenance System," *Artificial Intelligence*, vol. 28, No. 1, 1986.

Dietterich, T.G., "Learning at the Knowledge Level," Machine Learning, Vol. 1, No. 3, pp. 287-316, 1986.

Dietterich, T.G., and Flann, N.S., "An Inductive Approach to Solving the Imperfect Theory Problem," *Proceedings of 1988 Symposium on Explanation-Based Learning*, pp. 42-46, Stanford University, 1988.

Dietterich, T.G., and Michalski, R.S., Learning to Predict Sequences, in *Machine Learning: An Artificial Intelligence Approach Vol. II*, Morgan Kaufmann, Los Altos, CA, R.S. Michalski, J.G. Carbonell and T.M. Mitchell (Eds.), pp. 63-106, 1986.

Duval B., and Kodratoff Y. "A Tool for the Management of Incomplete Theories: Reasoning about Explanations" in *Machine Learning, Meta-Reasoning and Logics*, P. Brazdil and K. Konolige (Eds.), Kluwer Academic Press, pp. 135-158, 1990.

Ehrenfeucht, A., Haussler, D., Kearns, M., and Valiant, L., "A General Lower Bound on the Number of Examples Needed for Learning," *Proceedings of the 1988 Workshop on Computational Learning Theory COLT'88*, pp. 139-154, MIT, 1988.

Fisher, D.H., McKusick, K., Mooney, R., Shavlik, J.W., and Towell, G., "Processing Issues in Comparisons of Symbolic and Connectionism Learning Systems," *Proceedings of the 6th International Workshop on Machine Learning*, pp. 169-173, Ithaca, NY, 1989.

Goldberg, D.E., *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, 1989.

Haussler D. and Pitt, L. (Eds.), *Proceedings of the 1988 Workshop on the Computational Learning Theory* (COLT 88), Morgan Kaufmann Publishers, San Mateo, CA, 1988.

Hoff, B., Michalski, R.S. and Stepp, R., INDUCE 2 - a program for learning structural descriptions from examples, Intelligent Systems Group Reports, No.83-1, Department of Computer Science, University of Illinois at Champaign-Urbana, 1983.

Holland, J.H., "Escaping Brittleness: The Possibilities of General-Purpose Learning Algorithms Applied to Parallel Rule-based Systems," in *Machine Learning: An Artificial Intelligence Approach, Vol. II*, Morgan Kaufmann, Los Altos, CA, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), pp. 499-570, 1986.

Holland, J.H., "Genetic Algorithms and Classifier Systems: Foundations and Future Directions," *Proceedings of the 2nd International Conference on Genetic Algorithms*, pp. 82-89, Cambridge, MA, 1987.

Holland J. H. Holyoak K. J., Nisbett R. E. and Thagard, P R., Induction: Processes of Inference, Learning and Discovery, The MIT Press, Cambridge, MA 1986.

Ko, H., "Assembly Planning Through Constructive Learning Approach," to appear in *Machine Learning and Inference Reports*, Center for AI, George Mason University, 1989.

Ko, H. and Michalski, R.S., "Types of Explanation and Their Role in Constructive Learning," to appear in *Machine Learning and Inference Reports*, Center for AI, George Mason University, 1989.

Kodratoff, Y., and Tecuci, G., "DISCIPLE-1: Interactive Apprentice System in Weak Theory Fields," *Proceedings of IJCAI-87*, pp. 271-273, Milan, Italy, 1987.

Laird, J.E., (Ed.), *Proceedings of the Fifth International Conference on Machine Learning*, University of Michigan, Ann Arbor, June 12-14, 1988.

Laird, J.E., Rosenbloom, P.S., and Newell A., "Chunking in SOAR: the Anatomy of a General Learning Mechanism," *Machine Learning*, Vol. 1, No. 1, pp. 11-46, 1986.

Lebowitz, M., "Integrated Learning: Controlling Explanation," Cognitive Science, Vol. 10, No. 2, pp. 219-240, 1986.

Mehler, G., Bentrup, J., and Riedesel, J., "INDUCE.4: A Program for Incrementally Learning Structural Descriptions from Examples," *Reports of Intelligent Systems Group*, Dept. of Computer Science, University of Illinois at Urbana-Champaign, 1986.

Michalski, R.S., "Discovering Classification Rules using Variable-valued Logic System VL_1," *Proceedings of the Third International Joint Conference on Artificial Intelligence*, Stanford, 1973.

Michalski, R. S., "Theory and Methodology of Inductive Learning," *Machine Learning: An Artificial Intelligence Approach*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (Eds.), Tioga Publishing Co., 1983.

Michalski, R.S., Understanding the Nature of Learning: Issues and Research Directions, in *Machine Learning: AN Artificial Intelligence Approach Vol. II*, Michalski, R.S., Carbonell, J.G., and Mitchell, T. M. (Eds.), Morgan Kaufmann Publishers, 1986.

Michalski, R. S., "Concept Learning," *Encyclopedia of Artificial Intelligence*, E. Shapiro (Ed.), John Wiley & Sons, January, 1987.

Michalski, R.S., LEARNING FLEXIBLE CONCEPTS: Fundamental Ideas and a Method Based on Two-tiered Representation, in *Machine Learning: AN Artificial Intelligence Approach vol. III*, MOrgan Kaufmann, 1990.

Michalski, R. S., Ko, H. and Chen, K., "Qualitative Prediction: The SPARC/G Methodology for Describing and Predicting Discrete Processes," in *Expert Systems*, P. Dufour and A. Van Lamsweerde (Eds.), Academic Press Incorporated, pp. 125-158, 1987.

Michalski, R.S.and Ko, H., "On the Nature of Explanation or Why Did the Wine Bottle Shatter," *Proceedings of the AAAI Workshop on Explanation-based Learning*, Stanford University, March 1988.

Michalski, R.S. and Negri, P., "An Experiment on Inductive Learning in Chess End Games," *Machine Representation of Knowledge, Machine Intelligence 8*, E. W. Elcock and D. Michie (Eds.), Ellis Horwood Ltd., New York, pp. 175-192, 1977.

Michalski, R. S. and Watanabe, L., "Constructive Closed-loop Learning: Fundamental Ideas and Examples," *Reports of Machine Learning and Inference Laboratory*, 1988.

Michalski, R. S. and Zemankova, M., "What is Generalization: An Inquiry into the Concept of Generalization and its Types," to appear in *Reports of Machine Learning and Inference Laboratory*, MLI 90-6, Center for Artificial Intelligence, George Mason University, 1990.

Minton, S., "Quantitative Results Concerning the Utility of Explanation-Based Learning," *Proceedings of AAAI-88*, pp. 564-569, Saint Paul, MN, 1988.

Minton, S., Carbonell, J.G., Etzioni, O., et al., "Acquiring Effective Search Control Rules: Explanation-Based Learning in the PRODIGY System," *Proceedings of the 4th International Machine Learning Workshop*, pp. 122-133, University of California, Irvine, 1987.

Mitchell, T. M., Keller, T., and Kedar-Cabelli, S., "Explanation-Based Generalization: A Unifying View," *Machine Learning Journal*, Vol. 1, January 1986.

Mostow, J., "Machine Transformation of Advice into a Heuristic Search Procedure," in *Machine Learning: An Artificial Intelligence Approach, Vol. I*, pp. 367-403, R. S. Michalski, J. G. Carbonell and T. M. Mitchell (Eds.), Morgan Kaufmann, Los Altos, CA, 1986.

O'Rorke, P.V., Morris, S., and Schulenburg, D., "Theory Formation by Abduction: Initial Results of a Case Study Based on the Chemical Revolution," *Proceedings of the 6th International Workshop on Machine Learning*, pp. 266-271, Ithaca, NY, 1989.

Pazzani, M.J., "Integrating Explanation-Based and Empirical Learning Methods in OCCAM," *Proceedings of EWSL-88*, pp. 147-166, Glasgow, Scottland, 1988.

Peirce, C.S., *Elements of Logic* in Collected papers of Charles Sanders Peirce (1839-1914), Ch. Hartshorne and P.Weiss (Eds.), The Belknap Press Harvard University Press, Cambridge, MA, 1965.

Quinlan, J. R., "Discovering Rules from Large Collections of Examples: A Case Study", *Expert Systems in the Microelectronic Age*, D. Michie (Ed.), Edinburgh Univeristy Press, Edinburgh, 1979.

Schafer, D. (Ed.), *Proceedings of the 3rd International Conference on Genetic Algorithms*, George Mason University, June 4-7, 1989.

Segre, A. M., "Explanation-Based Learning of Generalized Robot Assembly Plans," PhD thesis, *UILU-ENG-87-2208*, Coordinated Science Laboratory, University of Illinois, January 1987.

Segre, A. M. (Ed.), *Proceedings of the Sixth International Workshop on Machine Learning*, Cornell University, Ithaca, New York, June 26-27, 1989.

Slimmer, J.C. and Granger, R.H., Jr., "Incremental Learning from Noisy Data," *Machine Learning*, 1, pp.317-354, 1986.

Smolensky, P., "Connectionist AI, Symbolic AI, and the Brain," *Artificial Intelligence Review 1*, pp. 95-109, 1987.

Tanner, M.C., Explaining Kowledge Systems:Justifying Diagnostic Conclusions, PhD thesis, Department of Computer Science and Information Science, Ohio State University, 1990.

Tecuci G. and Kodratoff Y., "Apprenticeship Learning in Imperfect Domain Theories," in *Machine Learning: An Artificial Intelligence Approach, Vol. III*, Y. Kodratoff and R. S. Michalski (Eds.), Morgan Kaufmann, Los Altos, CA, 1990.

Touretzky D., Hinton, G., and Sejnowski, T. (Eds.), *Proceedings of the 1988 Connectionist Models*, Summer School, Carnegie Mellon University, June 17-26, 1988.

Wilkins, D.C., Clancey, W.J., and Buchanan, B.G., *An Overview of the Odysseus Learning Apprentice*, Kluwer Academic Press, New York, NY, 1986.