INFERENTIAL LEARNING THEORY AS A BASIS FOR MULTISTRATEGY TASK-ADAPTIVE LEARNING

by

R. S. Michalski

Proceedings of the First International Workshop on Multistrategy Learning, MSL-91, Harpers Ferry, WV, 3-18, 1991.

Proceedings of the First International Workshop on

MULTISTRATEGY LEARNING (MSL-91)

November 7-9, 1991
Harpers Ferry

Edited by Ryszard S. Michalski and Gheorghe Tecuci

Sponsored by the Office of Naval Research and
the Center for Artificial Intelligence, George Mason University

Contents

I. Theoretical and General Issues

II. Theory Revision

III. Strategy Selection

IV. Integration of Symbolic and Subsymbolic Methods

V. Cognitive Models

VI. Special Topics and Applications

Author Index

Inferential Learning Theory as a Basis for Multistrategy Task-Adaptive Learning

Ryszard S. Michalski
Center for Artificial Intelligence and
Department of Computer Science
George Mason University
Fairfax, VA 22030
michalski@aic.gmu.edu

Abstract

The development of multistrategy learning systems should be based on a clear understanding of the roles, and the applicability conditions of different learning strategies. To this end, the paper presents initial results on the *Inferential Learning Theory*. that aims at understanding the *competence aspects* of learning processes. The theory views learning as a goal-oriented process of modifying the learner's knowledge by exploring the learner's experience. Consequently, it analyzes learning processes in terms of high-level patterns of inference, called *knowledge transmutations*. Several knowledge transmutations are discussed in a novel way, specifically, inductive and deductive generalization, abstraction, concretion and abduction. A methodology for *multistrategy task-adaptive learning* (MTL), based on the theory, is briefly discussed and illustrated by an example. A MTL learner chooses a strategy dynamically, according to the three factors: input information, learner's background knowledge and the learning goal. The MTL aims at integrating a range of learning strategies, such as abstraction, deductive generalization, empirical and constructive induction, analogy, abstraction and abduction.

Key words: learning theory, multistrategy learning, induction, abduction, generalization, abstraction, knowledge transmutation.

1 Introduction

The last several years have marked a period of great expansion and diversification of methods and approaches to machine learning. Most of the research has been oriented toward *single-strategy* methods that apply one primary type of inference and/or the computational mechanism. Such methods include, for example, learning decision trees or decision rules from examples, explanation-based generalization, deriving numerical equations characterizing a set of data, neural net learning from examples, genetic algorithm based learning, empirical conceptual clustering, and others. The research progress on these and related topics have been reported by many authors, among them Laird (1988), Touretzky, Hinton and Sejnowski (1988), Goldberg (1989), Schafer (1989), Segre (1989), Rivest, Haussler and Warmuth (1989), Fulk and Case (1990), Porter and Mooney (1990), Kodratoff and Michalski (1990), and Birnbaum and Collins (1991).

With the growing understanding of the capabilities and limitations of single-strategy methods, there has been an increasing interest in *multistrategy learning* systems that integrate two or more inference types and/or computational mechanisms. Such systems take advantage of the strengths of different learning strategies, and thus potentially can be applied to a much wider range of practical problems than single-strategy systems. Among well-known multistrategy systems (often called "integrated learning systems") are Unimem (Lebowitz, 1986), Odysseus (Wilkins, Clancey, and Buchanan, 1986), Prodigy (Minton et al., 1987), DISCIPLE-1 (Kodratoff and Tecuci, 1987), GEMINI (Danyluk, 1987 and 1989), OCCAM (Pazzani, 1988), IOE (Dietterich and Flann, 1988), and KBL (Whitehall, 1990). Most of these systems are concerned with integrating an empirical inductive method with an explanation-based learning method. Some, like DISCIPLE, also include a simple method for analogical learning. The integration of these methods is typically done in some predefined, problem-independent way, and without any clear theoretical justification.

An open and very challenging problem is to integrate a whole spectrum of learning strategies in a problem-dependent way, and according to sound theoretical foundations. By a problem-dependent integration we mean an integration in which learning strategy, or a combination thereof, is automatically adapted to changing learning situations.

Humans are clearly able to apply a great variety of learning strategies in such a flexible fashion, according to the demands of the situation. Matching this ability emerges as a fundamental long-term objective for machine learning research. To this end, it is necessary to understand the principles and trade-offs characterizing different learning strategies, to clarify their meaning and interrelationships, to determine conditions for their most effective applicability, and ultimately to develop a unified theory of learning strategies.

This paper reports some early results toward such a goal. It outlines the *Inferential Theory of Learning (ITL)* that analyzes and characterizes learning processes in terms of high level inference patterns that transform initial learner's knowledge to knowledge desired, that is, knowledge satisfying a given learning goal. These inference patterns, called *knowledge transmutations*, represent different types of knowledge transformation. The major goals of the theory are to understand the properties of different knowledge transmutations, as well as their role and implementation techniques in various learning methods and paradigms.

The theory strives to provide a conceptual basis for analyzing diverse learning processes from the viewpoint of their logical capabilities. Its goals are to provide techniques for determining the *competence* of learning systems. These goals differentiate the Inferential Learning Theory from the Computational Learning Theory that is concerned with computational *complexity* of learning processes (e.g., Fulk ad Case, 1990).

The competence aspects address such questions as what types of knowledge the learner is able to learn from what kinds of inputs, given certain prior knowledge; what is the logical relationship between the learned knowledge, the input information and the learner's prior knowledge; what types of inference and knowledge transformations underlie different learning processes, etc.

To present simply the proposed ideas, the presentation consists of mainly conceptual explanations and examples, rather than precise definitions and formal expositions. The presented work is an extension or a refinement of previous ideas described in (Michalski, 1983, 1990a, & 1991).

2. Basic Tenets of the Inferential Theory of Learning

Learning has been traditionally characterized as a behavior change due to experience. While this view is appealing due to its simplicity, it does not provide many clues about how to actually implement learning systems. To build a learning system, one needs to understand, in computational terms, what types of knowledge changes occur in learning, and how they are accomplished in response to different types of experience.

To provide answers to such questions, the Inferential Theory of Learning (ITL) assumes that learning is a goal-guided process of modifying the learner's knowledge by exploring the learner's experience. Such a process can employ any type of inference--deduction, induction or analogy. This process always involves some "background knowledge," that is, the goal-relevant parts of the learner's prior knowledge. Consequently, the information flow in a learning process can be characterized by a general schema shown in Figure 1.

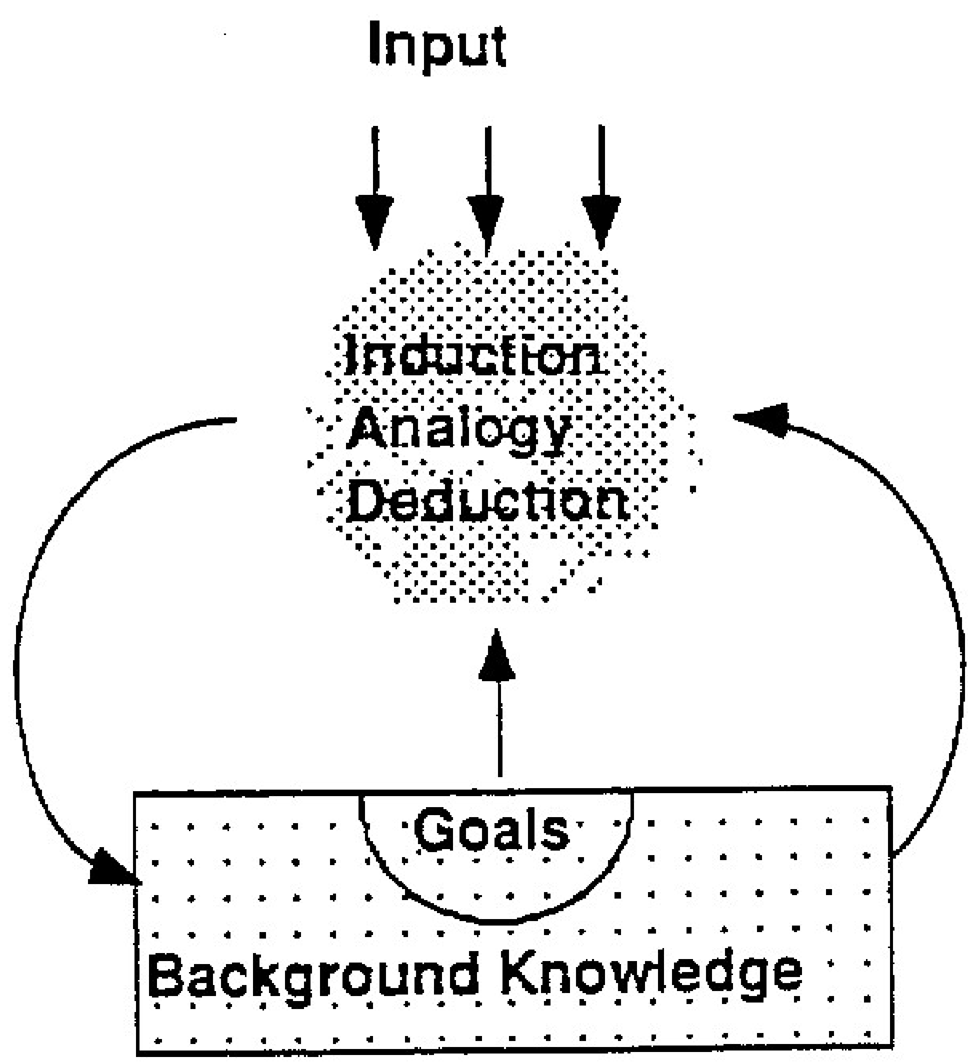

Figure 1. An illustration of a general learning process.

In each learning cycle, the learner analyzes the input information in terms of its background knowledge and its goals, and generates new knowledge and/or a better form of knowledge (depending on the learning goal). The results are fed back to the learner's "knowledge base," and may be used in subsequent learning processes.

The Inferential Learning Theory states that in order to learn, an agent has to be able to perform *inference*, and has to possess *memory* that supplies the BK needed for performing the inference and records the results of the inference for future use. Without either of the two components—the ability to reason and the ability to store and retrieve information from memory—no learning can be accomplished. Thus, one can write an "equation":

$$Learning = Inference + Memory$$

It should be noted that the term "inference" is used here in a very general sense, meaning any type of reasoning or knowledge transformation. The double role of memory, as a supplier of background knowledge, and as a storer of the results, is often reflected in the organization of a learning system. For example, in a neural net, background knowledge resides in the structure of the network (in the type of units used and in the way they are interconnected), and in the initial weights of the connections. The learned knowledge usually resides only in the new values of the weights. In a decision tree learning system, the BK includes an attribute evaluation procedure and knowledge about the domains of the attributes. The knowledge created is in the form of a decision tree. In a "self-contained" rule learning system, all background knowledge and the learned knowledge would be in the form of rules. A learning process would involve modifying prior rules and/or creating new ones. The ultimate learning capabilities of a learning system are determined by what it can or cannot change in its knowledge base during a learning process.

The key idea of Inferential Leaning Theory is that the processes involved in accomplishing a learning goal can be characterized in terms of high-level inference patterns, called *knowledge transmutations*. A knowledge transmutation takes as arguments the input and learner's background knowledge, and generates another piece of knowledge. This paper discusses several basic knowledge transmutations, specifically, generalization, abstraction, similization, and their counterparts, specialization, concretion, and dissimilization.

Knowledge transmutations represent different reasoning methods that transform the input knowledge into derived knowledge using the learner's prior knowledge. For example, an empirical inductive generalization is a transmutation that transforms concept examples and relevant domain knowledge into a concept description that is expressed in the same description space as the initial examples. A constructive generalization transfers concept examples and relevant learner's knowledge into concept descriptions expressed in another description space (Michalski, 1983; Wnek and Michalski, 1991a). An explanation-based generalization (Mitchell, Keller and Kedar-Cabelli, 1986) can be viewed as a transmutation ("constructive deductive generalization") that takes as arguments an example in the "operational description space," a concept description in an "abstract" description space, and relevant domain knowledge, and derives from them a concept description in the "operational" description space. Sec. 3 discusses several basic knowledge transmutations. The main topic of the Inferential Learning Theory is to analyze and explain diverse learning processes in terms of various knowledge transmutations.

Knowledge transmutations represent classes of transformations that can be implemented in many different ways. Depending on the knowledge representation and the computational mechanism, knowledge transmutations are performed explicitly or implicitly. In symbolic learning systems, transmutations are typically implemented in a more or less explicit way, and executed in steps that are conceptually comprehensible. For example, the INDUCE learning system, performs inductive generalization according to certain generalization rules---selective or constructive, and each rule represents a conceptually understandable transformation (Michalski, 1983).

In subsymbolic systems, e.g., neural networks, transmutations are performed implicitly, in steps dictated by the underlying computational mechanism. These steps do not correspond to any conceptual operations. For example, a neural network may generalize an input example by performing a sequence of small modifications of weights of internode connections. These weight modifications are difficult to explain in terms of explicit inference rules, nevertheless, they can produce a global effect equivalent to generalizing a set of examples. Such an effect can be demonstrated by a *diagrammatic visualization* employing a planar representation of a multidimensional space. For example, a diagrammatic visualization of inductive generalization performed by a neural network, genetic algorithm, and symbolic learning systems is described by Wnek and Michalski (1991b - this volume).

The Inferential Learning Theory postulates that a learning process depends on the input information (input), background knowledge and the learning goal. The three components constitute a *learning task.*

An input can be sensory observations, or knowledge from some source, e.g., a teacher or the previous learning step. Such knowledge can be in the form of stated facts, concept instances, previously formed generalizations, conceptual hierarchies, certainty measures, or any combinations of such types.

In humans, declarative ("conceptual') knowledge and procedural knowledge ("skills") seem to reside in different neural structures, and are acquired in different ways. There is a "conscious" access to conceptual knowledge, but there is no such access to the skill knowledge. Therefore, acquiring conceptual knowledge is based primarily on an explicit reasoning and memorizing of the results ("studying"). On the other hand, acquiring skills is based primarily on practice and exercise without much reasoning. A computer system can store and access both declarative and procedural knowledge in the same way. Consequently, acquiring these two forms of knowledge by a computer program can be done by similar mechanisms.

Given an input, and some background knowledge, a learner could potentially generate an infinite number of inferences. To limit the proliferation of choices, a learning process needs to be guided by a *learning goal.* A learning goal determines what parts of prior knowledge are relevant, what knowledge is to be acquired, and how the learned knowledge is evaluated.

There can be many different types of learning goals, e.g., to solve a problem, to perform an action, to "understand" observed facts, to concisely describe given data, to discover a regularity in a collection of observations, to explain or express a regularity in terms of high level concepts, to confirm a given piece of knowledge, etc. A learner may have more than one goal, and the goals may be conflicting. In such situations, the relative importance of different goals affects the decision about the amount of effort to be extended in pursuing any of them. A weakness of some machine learning research is that it considers learning processes separately from the learning goal(s), and as a result many developed systems are method-oriented rather than problem-oriented. Studying the role of goals in learning is an important research topic for machine learning.

In sum, Inferential Learning Theory states that learning is a process of transforming given knowledge into desired knowledge by using input information and background knowledge. Such transformations can be characterized in terms of high-level inference patterns called knowledge transmutations.

3. Types of inference

The central aspect of any knowledge transmutation is the type of underlying inference. This is so, because the type of inference characterizes a transmutation along the truth-falsity dimension, and thus determines the validity of its conclusion. In a learning process, any type of inference may be involved. Consequently, a complete learning theory has to include a complete theory of inference. Such a theory of inference has to account for all possible types of knowledge transformations. To this end, Figure 2 presents an attempt to schematically illustrate all major types of inference.

The first classification is to divide inferences into two basic types: deductive and inductive. In defining these forms, many conventional approaches do not distinguish between the input information and the reasoner's background knowledge. Such a distinction, is, however, important for characterizing learning processes, and leads to a more adequate descriptions of them. To define these forms of inference in a language-independent way, let us consider an entailment:

$$P \cup BK \models C \qquad (1)$$

where P stands for a set of statements, called *premise*, BK stands for the reasoner's *background knowledge*, and C stands for a set of statements, called *consequent*.

Deductive inference is deriving consequent C, given P and BK. Inductive inference is hypothesizing premise P, given consequent C and BK. Thus deduction can be viewed as "tracing forward" the relationship (1), and induction as "tracing backward" this relationship. Because the relationship (1) succinctly explains the relationship between two basic forms of inference, it will be subsequently characterized as the "fundamental equation" for inference.

According to the above formulation, deduction is a truth-preserving inference, and induction is a falsity-preserving inference. The latter means that if C is not true, then P cannot be true.

In a general view of deduction and induction that captures also their approximate or commonsense forms, the "strong" entailment $\models$ may be replaced by a "weak" entailment. A weak entailment may be *plausible, probabilistic or partial*. The difference between a "strong" (valid) and "weak" entailment leads to another major classification of types of inference. Specifically, inferences can be *universal (strong)* or *contingent (weak)*. Universal inferences assume the "strong" entailment, and contingent inferences assume the "weak" entailment. Consequently, universal deductive inferences are "strongly" truth-preserving, and universal inductive inferences are "strongly" falsity-preserving. Contingent deductive inferences are "weakly" truth-preserving, and contingent inductive inferences are "weakly" falsity-preserving.

To illustrate a universal inference, suppose that BK is "If all elements of a set X have a property q, then any specific element of X must have the property q." Such knowledge is universal, because its truth stems from the intrinsic meaning of the statements involved, and does not depend on the specific domain. If an input is "All elements of the set X have property q and x is an element of X," then deriving a statement "x has property q" is a universal deductive inference.

Suppose now that BK contains the same rule as before plus a statement "x is an element of X," and the input is "x has property q." Hypothesizing the statement (premise P) "All elements of X have the property q" is a universal (empirical) induction. If this statement is true, then the input must be true in the context of BK. The inference is falsity-preserving, because if the input is not true (x did not have the property q), then the hypothetical premise must be false. As another example, assume that BK is "All elements of X have property q" and an input is "X has property q." Hypothesizing the statement "x is a member of X" is also a universal induction. If the derived statement is true, then the input must be true in the context of BK. Again, if the

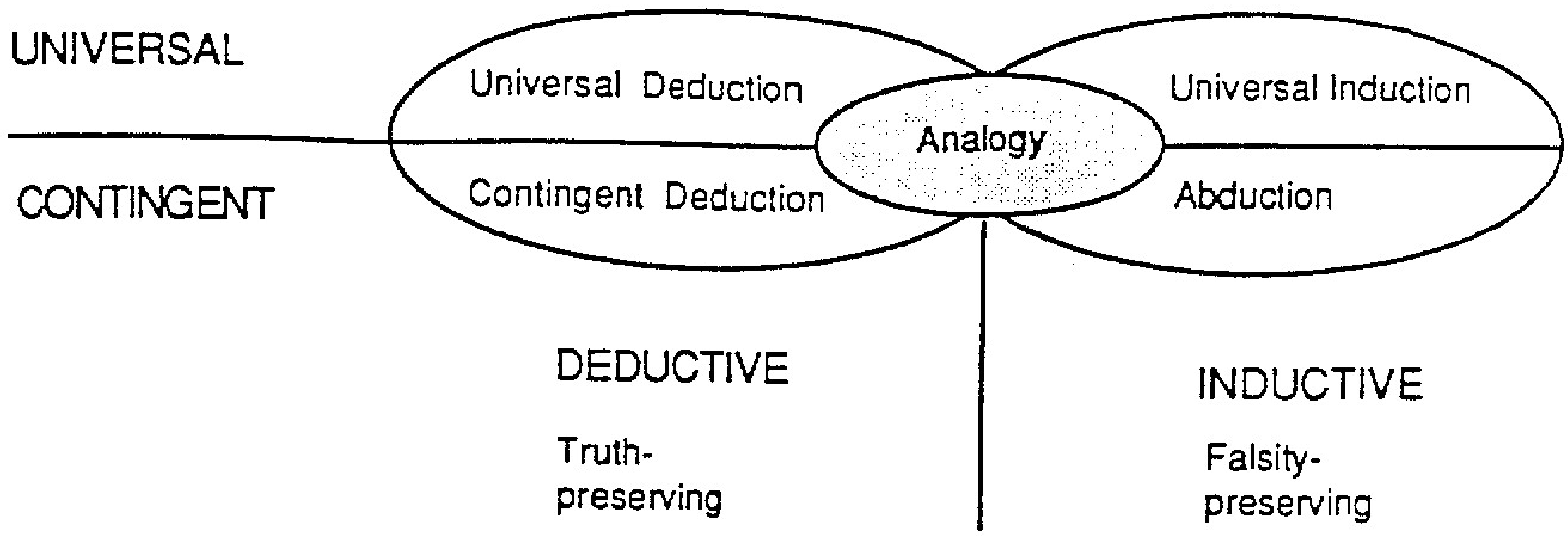

Figure 2. A classification of major types of inference.

input is not true, then the conclusion could not be true.

Contingent inferences use knowledge (in the input or BK) that represents some world knowledge that is not totally certain. It can be in the form of probabilistic dependencies, plausible implications, partial dependencies, etc. The contingency of such relationships is usually due to the fact that they represent incomplete, imprecise or only partially correct information about all the relevant factors. These relationships may hold with different "degrees of strength." The conclusions from inferences based on contingent dependencies (even using valid rules of inference) are therefore uncertain, and may be characterized by different "degrees of belief" (probabilities, degrees of truth, likelihoods, etc.). They also usually hold in both directions, although not with the same strength in each direction (Collins and Michalski, 1989). For example, "If there is fire, then there is usually smoke" is a (bidirectional) contingent dependency. If one sees fire, then one may derive a conclusion that there may be smoke. This is a contingent deduction. The derived conclusion, however, is not certain. Using a reverse direction of reasoning ("tracing backward" the above dependency), observing smoke, one may hypothesize that there is fire. This is a contingent induction or *abduction*. It is also an uncertain inference. Notice that in the latter case, if the input is false, the degree to which the conclusion is false depends on the degree to which the dependency is true.

In the above example both conclusions are uncertain, and this might suggest that there is no principal difference between contingent deduction and abduction. These two types of inferences are different if one assumes that $\models$ in (1) indicates a causal ordering, i.e., P is viewed as a cause, and C as a consequence. Contingent deduction derives a plausible consequent, C, of the causes represented by P. Abduction derives plausible causes, P, of the consequent C. Contingent deduction can thus be viewed as "tracing forward," and abduction as "tracing backward" such contingent, causally-ordered dependencies.

In sum, both contingent deduction and contingent induction are based on domain-dependent relationships. Contingent deduction produces likely consequences of given causes, and contingent induction produces likely causes of given consequences.

Universal deductive inference is strictly truth-preserving, and universal induction is strictly falsity-preserving (if C is not true, then the hypothesis P cannot be true either). A universal deduction thus produces a provable (valid) consequent from a given premise in the context of BK. A universal induction produces a hypothesis that logically entails the given consequent in the context of BK (though the hypothesis itself may be false). Contingent deduction is truth-preserving, and contingent induction is falsity-preserving only to the extent to which contingent dependencies involved in reasoning are true.

The intersection of deduction and induction (that is a truth-preserving and falsity-preserving inference) represents an equivalence-based inference. Analogy can be viewed as an extension of equivalence-based inference (a "similarity-based" inference). An analogy can be characterized as a combination of induction and deduction combined. This is why analogy occupies the central area in the diagram. Induction is involved in detecting an analogical match (i.e., in determining the properties and/or relations similar for the two analogs), whereas deduction uses the hypothesized analogical match to derive unknown properties of the target analog.

4. Types of induction

As mentioned above, universal induction produces a premise that together with BK) tautologically (strongly) implies a given consequent. The tautological implication stems from the set-superset relationship. On the other hand, contingent induction produces a premise that together with BK only weakly implies the consequent. Such a case occurs when a generalization only approximately describes the facts.

Induction underlies two types of knowledge transmutations: *inductive generalization* and *inductive specialization*.

Inductive generalization is central to many learning processes. It transfers knowledge of properties of a subset into knowledge of properties of a set. Such a transfer can be done without changing the description space, or with changing the description space (in the latter case the generated knowledge may involve terms not present in the input descriptions).

In the first case we have an empirical inductive generalization, and in the second--constructive inductive generalization. For example, transfering the input "*bean 1*, *bean 2*, and *bean 3* from bag B are white" into a hypothesis "All beans in bag B are white" is an empirical inductive generalization. Notice that if the hypothesized premise "All beans in bag B are white," is true, then the given consequent (i.e., *bean 1*, *bean 2*, and *bean 3* from bag B are white) must necessarily be true. Thus, the fundamental equation for inference (1) is satisfied without having to involve BK. (For an illustration of constructive generalization, see Figure 3.) Inductive specialization is a less known transmutation. Suppose, for example, that we are told that

"There is a University in Virginia designed by Jefferson." (2)

Suppose that knowing (2), and that Charlottesville is an academic town in Virginia, an agent hypothesizes that

"There is a University in Charlottesville designed by Jefferson." (3)

This is a form of induction because if (3) is true, then (2) must also be true (assuming the background knowledge is true).

In the presented approach, induction is viewed as an inference opposite to deduction. It produces premises that entail consequents, e.g., explanations for the given facts. These explanations can be in the form of generalizations (theories, rules, laws, etc.), causal explanations, or both.

Given a consequent C and non-trivial BK, the fundamental equation (1) could be satisfied by an infinite number of premises, but only few of them may be of any interest. We are usually interested only in "justifiable," "plausible" and/or simple hypotheses. Therefore, we define an *admissible induction* by adding additional constraints. An admissible induction is defined as follows.

Given a consequent C and BK, determine a premise P, consistent with BK, that satisfies the fundamental equation

$$P \cup BK \models C \quad (4)$$

and satisfies *hypothesis selection criteria*.

In different contexts, the selection criteria have been called a *bias* (e.g., Utgoff, 1986), a *comparator* (Poole, 1989), or *preference criteria* (Michalski, 1983). The selection criteria represent extra-logical constraints that specify how to choose among a potentially unlimited number of candidate hypotheses. Ideally, these criteria should reflect the properties of a hypothesis that are desirable from the viewpoint of the learner's goals. Sometimes these criteria (or bias) are hidden in the description language used. For example, an inductive program may use a description language that is limited to only conjunctive statements involving attributes from a predefined set. The selection criteria also may be dictated by the method performing induction. For example, the method based on generating decision trees is automatically limited to using only operations of conjunction and disjunction in the hypothesis representation.

There are three generally desirable characteristics of a hypothesis: *plausibility*, *utility*, and *generality*. The plausibility expresses a desire to find a "true" hypothesis. Because the problem is logically underconstrained, the "truth" of a hypothesis cannot be guaranteed in principle. To satisfy the equation (4), a hypothesis has to be *complete* and *consistent* with regard to the input facts (Michalski, 1983). Experiments have shown, however, that in some situations an inconsistent and/or incomplete hypothesis may give a better overall predictive performance than a complete and consistent one (e.g., Bergadano et al., 1990). The utility criterion requires a hypothesis to be simple to express, easily implementable or easy to apply to an expected set of tasks. The generality criterion seeks a hypothesis that can predict a large scope of new cases.

While the view of induction described above is not universally accepted in machine learning literature, it is consistent with many long-standing thoughts on this subject going back to Aristotle (e.g., Adler and Gorman, 1987; Aristotle). Aristotle, and many subsequent thinkers, e.g., Bacon (1620), Whewell (1857), Cohen (1970) and others viewed induction as a fundamental inference that underlies all processes of creating new knowledge. They did not limit it--what is sometimes done--to only inductive empirical generalization.

Generalizing the earlier mentioned distinction between empirical and constructive generalization, one can classify any form of induction into *empirical* and *constructive*. The difference between the two is usually characterized in terms of the amount of domain knowledge involved in the process of learning (here, domain knowledge means a part of background knowledge that concerns the specific topic of application). Empirical induction uses little domain knowledge, while constructive induction uses more domain knowledge. A more precise way to characterize this distinction is that in empirical induction the description space for examples and for the hypotheses is the same, while in constructive induction these spaces are different.

Inductive inference underlies several important transmutations. Examples of them are presented in Figure 3. (To test whether an inference is inductive, one needs to determine if the input is entailed by the union of the hypothesis and BK.) As mentioned earlier, in the general formulation of induction, the union of the hypothesis and BK may only weakly (e.g., plausibly) entail the consequent. In this case we have a contingent induction. In a weak entailment the hypothesis may be logically inconsistent and/or incomplete in relation to the input. In Figure 3, such a case is illustrated in the example of constructive inductive generalization.

5. Knowledge transmutations: abstraction vs. generalization

As stated earlier, transmutations (also called *derivations*) are patterns of inference which take an input and some background knowledge, and produce new knowledge. The previous section gave examples of transmutations employing inductive inference. Here we will look at some other transmutations.

In general, a transmutation involves a specific type of inference, and makes a certain type of change in the input knowledge. To be more precise, let us define a *knowledge module* as a set of sentences (e.g., in the first order predicate calculus) that describe a set of entities. The set of entities described or referred to by a module is called the *reference set*. A transmutation derives an output knowledge module from an input module and a given BK. Different transmutations change different aspects of the input module. Due to space limitation, we will limit our attention only to two types of changes, and to the corresponding pairs of mutually opposite transmutations:

A. A change in the amount of information (detail) conveyed by a description of a reference set:

abstraction vs. concretion

B. A change in the size of the reference set

generalization vs. specialization

• *Empirical inductive generalization*
(Background knowledge limited)

Input:	The "A girl's face" and "Lvow cathedral" are beautiful painting by Dawski.
BK:	none
Hypothesis:	All paintings by Dawski are beautiful.

• *Constructive inductive generalization*
(Background knowledge intensive)

Input:	The "A girl's face" and the "Lvow cathedral" are beautiful paintings by Dawski.
BK:	Dawski is a known painter. Paintings are pieces of art. Beautiful pieces of art by a known painter are expensive.
Hypothesis:	All paintings by Dawski are expensive.

• *Inductive specialization*

Input:	John lives is Virginia.
BK:	Fairfax is a town in Virginia.
Hypothesis:	John lives in Fairfax.

• *Abduction*

Input:	There is smoke in the house.
BK:	Fire usually causes smoke.
Hypothesis:	There is a fire in the house.

• *General constructive induction: (e.g., constructive generalization plus abduction)*

Input:	Smoke is coming from John's apartment.
BK:	Fire usually causes smoke. John's apartment is in the Golden Key building.
Hypothesis:	The Golden Key building is on fire.

Figure 3. Examples of different inductive transmutations.

Elements of these pairs of transmutations often co-occur, and this is why they are sometimes confused with each other. For example, abstraction is sometimes confused with generalization. Below is an attempt to describe more precisely these two pairs of transmutations.

Abstraction and concretion

Abstraction creates a less detailed description of a set of entities (i.e., the reference set) from a more detailed description. The usual purpose of abstraction is to reduce the amount of information about a set of entities (the reference set) so that information relevant to the learner's goal is preserved, and other information is discarded. For example, abstraction may transfer a description from one language to another language in which the properties relevant to the reasoner's goal are expressed, and other properties are not. An opposite operation to abstraction is *concretion*, which generates additional details about a given entity.

A very simple form of abstraction is to replace in the description of an entity a specific attribute value (e.g., the length in a centimeter) by a less specific value (e.g., the length stated in linguistic terms, such as short, medium and long). A complex abstraction would be to take a description of a computer in terms of electronic circuits and connections, and change it into a description in terms of the functions of the major components.

Abstraction can be characterized as a transformation:

$$D_1(S) \dashrightarrow D_2(S) \qquad (5)$$

such that

$$INF_G(D_1, BK) \supseteq INF_G(D_2, BK) \qquad (5')$$

where $D_1(S)$ and $D_2(S)$ are different descriptions of the set S (in the same or different languages), and $INF_G(D_1)$ and $INF_G(D_2)$ are sets of all deductive inferences, relevant to the goal G, that can be drawn about S from D_1 and D_2, respectively, using BK. If the goal G does not require to remove any parts from the descriptions D_1 and D_2, then (5') is equivalent to saying that D_1 implies D_2, meaning that if an entity has properties stated by D_1, then it has the properties stated by D_2. The goal defines what parts of the description are relevant and cannot be removed, and what parts of the description can be ignored. Often, the goal of an abstraction process is only implicit.

To illustrate the above, consider a *source statement* "John is 6 feet tall, weighs 190 pounds, has blue eyes, and lives in Fairfax." A transformation of this statement into a *target statement:*" John is a big man who lives in Virginia" is an abstraction. To make this abstraction one needs BK that "Being 6 feet tall and weighing 190 pounds classifies one to be called big," and that "Fairfax is a town in Virginia." The implied goal here is that information about the height, weight and the place where a person lives is relevant to the reasoner's goal, while the eye color is not. The abstracted statement clearly tells us less about John, but whatever can be inferred from it about John, can also be inferred from the original statement (given the same BK). The target statement does not introduce or hypothesize any more information about John. The goal is an important component in a general formulation of abstraction, because an abstraction process may introduce information that is incidental, and should not be taken into consideration while making inferences about the entity under consideration. For example, from an abstract drawing of a person one should not infer that the person is made out of paper.

Generalization and specialization

Generalization extends the set of entities described by a description, i.e., the reference set, and specialization reduces the reference set. In order to tell if a given transformation is a generalization, one needs to identify the reference set in the initial description, and see if the set was extended in the derived description. Generalization can be characterized as a transformation:

$$D_1(S_1) \dashrightarrow D_2(S_2) \qquad (7)$$

such that

$$S_2 \supseteq S1 \qquad (7')$$

where $D_1(S_1)$ and $D_2(S_2)$ are descriptions of sets of entities, S_1 and S2, respectively, and D_1 implies D_2.

The reason for the condition "D_1 implies D_2" is that generalizing a set of descriptions usually involves also a removal of information that is not shared by individual descriptions, and this is a form of abstraction. Only in a purely empirical generalization D_1 and D_2 are the same. A constructive generalization typically involves abstraction.

Generalization is typically inductive, which means that the extended set is inductively hypothesized. Generalization can also be deductive, when the more general statement is a logical consequence of the more specific one. For example, transforming a statement "Mary lives in France" into "Mary lives in Europe" is a deductive generalization, assuming background knowledge "France is a part of Europe."

In this example, the first sentence characterizes the set of "land parcels" called France as a place where Mary lives. The second statement applies this description to a larger set of parcels, called Europe. Given background knowledge that France is in Europe, one can deduce the second statement from the first one. The opposite operation to generalization is *specialization* that reduces the reference set. A typical form of specialization is deductive, but, as shown in section 3, there can also be an inductive specialization.

Example

To illustrate the difference between the abstraction and generalization, consider a statement d(S,v), saying that attribute (descriptor) d takes value v for the set of entities S. Let us write such a statement in the form:

$$d(S) = v \qquad (8)$$

Changing (8) to the statement $d(S) = v'$, in which v' represents a more general concept (e.g., a parent node in a generalization hierarchy of values of the attribute d), is an abstraction operation. Changing (5) to a statement $d(S') = v$, in which S' is a superset of S, is a generalization operation. For example, transferring the statement "color(my-pencil) = light-blue" into "color(my-pencil)=blue" is an abstraction operation.

Transforming the original statement into "color(all-my-pencils) = light-blue" is a generalization operation. Finally, transferring the original statement into "color(all-my-pencils)=blue" is both generalization and abstraction. In other words, associating the same or less information with a larger set is a generalization operation; associating less information with the same set is an abstraction operation.

In sum, generalization transforms descriptions along the set-superset dimension, and is typically falsity-preserving. In contrast, abstraction transforms descriptions along the level-of-detail dimension, and is typically truth-preserving. Generalization often uses the same description space (or language), abstraction often 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 reasoning acts involve both of them.

In addition to the transmutations described here, there are other types of transmutations, for example, similization, dissimilization, replication, deletion, selection and generation (Michalski, 1991).

6. Learning strategies

The concept of "learning strategy" has been used somewhat loosely in machine learning literature, often synonymously with a general method or a computational mechanism employed in a learning process. An attempt to make it more precise was done by Carbonell, Mitchell and Michalski (1983) who defined a learning strategy by the type of primary inference used in a learning process. We will slightly modify this characterization, assuming that a learning strategy is defined by the type of primary knowledge transmutation employed. We will also use the term "substrategy" to subclassify a strategy on the basis of knowledge representation and/or the underlying computational mechanism employed in it. Thus, an empirical inductive generalization of examples is a learning strategy. An inductive generalization of examples done by a neural net is a learning substrategy. Learning strategies can be ordered on the basis of the complexity of the primary knowledge transmutation involved in them.

The lowest learning strategy is *rote learning* (or *direct knowledge implantation*) in which the information from a source is copied directly into the learner's knowledge base. The primary knowledge transmutation involved in this strategy is *replication*. The next level strategy, *learning from instruction*, involves selecting parts or the knowledge supplied by a source that is relevant to the learner (a selection transmutation), and performing truth-preserving transformations of it to fit the learner's conceptual structure (a *reformulation* transmutation). The above two strategies change only the form of the information obtained from a source, but not its meaning.

Higher learning strategies require a learner to perform correspondingly more complex knowledge transmutations. In explanation-based generalization the underlying transmutation is deductive generalization. In learning by analogy or case-based learning, the underlying transmutation is similization. In learning causal explanations, it is abductive derivation. In learning from examples, and learning from observation and discovery, it is inductive generalization.

7. Multistrategy Task-adaptive Learning

The presented ideas provide a conceptual framework for the *multistrategy task-adaptive learning* (MTL) methodology that aims at integrating a range of learning strategies. According to the Inferential Learning Theory, three fundamental factors affect a learning process: what information is provided to the learner (i.e., *input* to the learning process), what learner already knows that is relevant to the input (i.e.,*background knowledge*, BK), and what the learner wants to accomplish (the *goal* of learning). These three factors constitute what we call a *learning task*.

The underlying idea of MTL is that a learning strategy should be tailored to the learning task (Michalski, 1990; Tecuci and Michalski, 1991a,b). Given an input information, an MTL system analyzes its relationship to BK and the learning goal, and on that basis determines a learning strategy. If an impasse occurs, a new learning task is assumed, and the learning strategy is determined accordingly.

An input to the MTL learner is assumed to be in the form of logic-style statements, and is either supplied by an external source, or by a previous learning step. It is also assumed that a learning goal is supplied from a supervisory control system. A specific learning goal could be, for example, to create a rule generalizing given 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 past knowledge, to develop a conceptual classification of facts, etc. In the absence of a specific goal, a general learning goal (a *default* goal) is assumed. The general goal is to derive any plausible and useful information from the input, and assimilate it within the BK.

In the first step of a learning process, the input activates segments of the learner's prior knowledge that are relevant to the input and the learning goal. This step thus determines the background knowledge (BK) for the learning process. This is done by exploring the "relevance relationship" between the input and different knowledge structures in the learner's knowledge base (Hieb and Michalski, 1991). The knowledge base is assumed to be in the form of type and part hierarchies interconnected by "traces" that link nodes of different hierarchies. This knowledge representation is called DIH ("Dynamically Interconnected Hierarchies"; see Hieb and Michalski, 1991), and is based on the theory of plausible reasoning introduced by Collins and Michalski (1989). Because of space limitations, DIH is not be discussed here. To give a simple illustration of it, consider a statement "Tulips grow in the Spring." Such a statement would be represented in DIH as a "trace" linking the node "tulips" in the type hierarchy of "Plants", with the node "grow" in the type hierarchy of "actions," and with the node "Spring" in the hierarchy of "Seasons."

The next step of the process is to determine the type relationship between the input information and BK. The method distinguishes five different types of such a relationship. Below is a characterization of these relationships, and a brief explanation of the functions performed by the learner.

1. The input represents new information

An input is "new" to the learner, in the sense that it has no "entailment relationship" with any part of BK (neither subsumes or is subsumed by it, nor contradicts it). The learner tries to identify parts of BK that are siblings under the same node in some hierarchy (e. g., other examples of the concept represented by the input). If this effort succeeds, the related parts are generalized, so that they account now for this input and possibly other information stored previously. The resulting generalizations and the input facts are evaluated for "importance," and those that pass an *importance criterion*, are stored. If the above effort does not succeed, the input is stored, and the control is passed to case 4. Generally, this case involves some form of synthetic learning (empirical learning or constructive induction), or learning by instruction.

2. The input is implied by, or implies a part of BK

This case represents a situation when a part of BK 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 part of BK. Depending on the learning task, 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.,"given state x, it was found that a useful action is y," then storing such a fact as a rule is a form of chunking used 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 criterion is satisfied, the input 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.

3. The input contradicts some part of BK

The system identifies the part of BK that is contradicted by the input information, and then attempts to specialize this part. If the specialization 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 also is 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 is to be revised. In general, this case handles situations requiring a revision of BK through some form of synthetic learning or managing inconsistency.

4. The input evokes an analogy to a part of BK

This case represents a situation when the input does not match any background fact or rule exactly, nor is related to any part of BK in the sense of case 1, but there is a similarity between the fact and some part of BK at a higher abstraction level. In this case, matching is done at this level of abstraction, using generalized attributes or relations. If the fact is "sufficiently important" it is stored with an indication of a similarity (analogy) to a background knowledge component, and with the specification of the aspects (abstract attributes or relations) defining the analogy. 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."

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.

Summarizing, an MTL learner may employ any type of inference. A deductive inference is employed when an input fact is consistent with, implies, or is implied by the background knowledge; analogical inference is employed when the input is similar to some part of past knowledge at some lever of abstraction; and inductive inference is employed when there is a need to hypothesize a new and/or more general knowledge.

8. A simple example

To illustrate simply some of the ideas described above, let us use the widely-known example of learning the concept of "cup" (Mitchell, Keller and Kedar-Cabelli, 1986). The example is deliberately oversimplified, so that major ideas can be presented in a very simple way. Figure 4 presents various learning strategies corresponding to different the inputs, BK and the desired output. The top part of the figure presents

- an abstract concept description (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),
- an operational concept description (operational CD).

An abstract CD describes the *concept* of "cup" in abstract terms, while an abstract OD describes a *specific object* in such terms. The bottom part of the figure specifies different learning strategies and the tasks to which they apply.

9. Summary

The goals of this research are to develop an underlying theoretical framework and a general methodology for integrating major learning strategies. The proposed Inferential Learning Theory provides a new viewpoint for characterizing the "competence" of learning processes (what kind of knowledge a learner can acquire from what kind of inputs). It proposes to analyze learning in terms of patterns of inference called transmutations. Transmutations are characterized by the type of underlying inference (induction vs. deduction), and the type of change they perform in knowledge. The paper discussed transmutations defined by the change in the reference set (generalization vs. specialization), and by the change in the level-of-detail (abstraction vs. concretion).

The presented ideas were used to outline a methodology for multistrategy task-adaptive learning (MTL). An MTL system determines by itself which strategy or combination thereof is most suitable for a given learning task. The current aim of MTL is to integrate such strategies as empirical and constructive induction, abduction, deductive generalization, abstraction, and analogy.

Many ideas discussed here are at an early stage of development, and many topics remain for further research. For example, future research should develop a more precise definition of various transmutations, identify new ones, and investigate different approches to their integration. Another interesting topic is to analyze existing learning algorithms and paradigms in terms of knowledge transmutations. There is also a need to develop a clear understanding of the areas of the most effective applicability of different learning methods and paradigms.

Cup(obj)

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

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)

	Input + BK:		*Learning Goal:*
Abstraction	Example Domain rules	$\triangleright$	Abstract OD
Deductive Generalization	Example Abstract CD Domain rules	$\triangleright$	Operational CD
Empirical Induction	Examples BK'	|<	Operational CD
Constructive Induction (Case of Generalization)	Example(s) Domain rules	|<	Abstract CD
Constructive Induction (Case of Abduction)	Example(s) Abstract CD	|<	Domain rules
Multistrategy Task-adaptive Learning	An integration of all the above, and their application according to the learning task, i.e., the combination of the input, BK anc the learning goal.		

OD and CD stand for object description and concept description, respectively. CUP1 stands for a specific cup; ot denotes a variable. BK' denotes some limited background knowledge, e.g., a specification of the value sets of tl attributes and their types. <---> stands for mutual implication. Symbols $\triangleright$ and |< denote deductive and induc transmutation, respectively.

Figure 4. An illustration of different learning strategies as applied to different learning tasks.

Acknowledgments

The author expresses his thanks to Mike Hieb, Gheorghe Tecuci and Janusz Wnek for useful comments on various topics reported in this paper, and Bill Deichler for editorial help.

This research was done in the GMU Center for Artificial Intelligence, and was supported in part by the Defense Advanced Research Projects Agency under the grants administered by the Office of Naval Research No. N00014-87-K-0874 and No. N00014-91-J-1854, in part by the Office of Naval Research under grants No. N00014-88-K-0397, No. N00014-88-K-0226 and No. N00014-91-J-1351, and in part by the National Science Foundation Grant No. IRI-9020266.

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

Bacon, F., *Novum Organum*, 1620.

Birnbaum, L. and Collins, G., *Proceedings of the 8th International Conference on Machine Learning*, Chicago, June 1991.

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

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.

Cohen, L.J., *The Implications of Induction*, London, 1970.

Collins, A. and Michalski, R.S., "The Logic of Plausible Reasoning: A Core Theory," *Cognitive Science*, Vol. 13, pp. 1-49, 1989.

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 No.* TR-98-066, Philips Laboratories, 1989.

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.

Fulk, M. and Case, J. *Proceedings of the 3rd Annual Workshop on Computational Learning Theory*, University of Rochester, N.Y., August 6-8, 1990.

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

Hieb, M. and Michalski, R.S. "A Knowledge Representation System Based on Dynamically Interconnected Hierarchies: Basic Ideas and Examples," *Reports of Machine Learning and Inference Laboratory*, Center for Artificial Intelligence, George Mason University, 1991 (to appear).

Kodratoff, Y.and Michalski, R.S. (eds.) *Machine Learning: An Artificial Intelligence Approach vol. III*, Morgan Kaufmann Publishers, Inc., 1990.

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.

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., "Toward a Unified Theory of Learning: Multistrategy Task-adaptive Learning," *Reports of Machine Learning and*

Inference Laboratory MLI-90-1, January 1990a.

Michalski, R.S., "A Methodological Framework for Multistrategy Task-adaptive Learning," *Proceedings of the Fifth International Symposium on Methodologies for Intelligent Systems*, Knoxville, (Elsevier Pub.), October 1990b.

Michalski, R.S., "Inferential Learning Theory: A Conceptual Framework for Characterizing Learning Processes," *Reports of Machine Learning and Inference Laboratory*, MLI 91-6, George Mason University, 1991.

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., Kedar-Cabelli,S., "Explanation-Based Generalization: A Unifying View," *Machine Learning*, Vol.1, 1986.

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

Poole, D., "Explanation and Prediction: An Architecture for Default and Abductive Reasoning," *Computational Intelligence*, No. 5, pp. 97-110, 1989.

Porter, B. W. and Mooney, R. J. (eds.), *Proceedings of the 7th International Machine Learning Conference*, Austin, TX, 1990.

Rivest, R., Haussler D. and Warmuth, M., *Proceedings of the Second Annual Workshop on Computational Learning Theory*, University of Santa Cruz, July 31-August 2, 1989.

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

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

Tecuci, G. and Michalski, R.S.,"A Method for Multistrategy Task-adaptive Learning Based on Plausible Justifications," in Birnbaum, L., and Collins, G. (eds.) *Machine Learning: Proceedings of the Eighth International Workshop*, San Mateo, CA, Morgan Kaufmann, 1991a.

Tecuci G., and Michalski R.S., "Input 'Understanding' as a Basis for Multistrategy Task-adaptive Learning," *Proceedings of the 6th International Symposium on Methodologies for Intelligent Systems*, Z. Ras and M. Zemankova (eds.), Lecture Notes on Artificial Intelligence, Springer Verlag, 1991b.

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

Utgoff, P., "Shift of Bias for Inductive Concept Learning," in *Machine Learning: An Artificial Intelligence Approach Vol. II*, Michalski, R.S., Carbonell, J.G., and Mitchell, T. M. (Eds.), Morgan Kaufmann Publishers, 1986.

Whewell, W., *History of the Inductive Sciences*, 3 vols., 3rd edition, London, 1857.

Whitehall, B. L., "Knowledge-based learning: Integration of Deductive and Inductive Learning for Knowledge Base Completion," *Ph.D. Thesis*, Computer Science Department, University of Illinois, 1990.

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

Wnek, J. and Michalski, R.S., "Hypothesis-Driven Constructive Induction in AQ17: A Method and Experiments," *Proc. of the IJCAI-91 Workshop on Evaluating and Changing Representation in Machine Learning*, K. Morik, F. Bergadano, W. Buntine (eds.), pp. 13-22,Sydney, Australia, August 24-30, 1991a.

Wnek, J. and Michalski, R.S., "An Experimental Comparison of Symbolic and Subsymbolic Learning Paradigms: Phase I - Learning Logic-style Concepts," *Proceedings of the First International Workshop on Multistrategy Learning*, R.S. Michalski and G. Tecuci (eds.), GMU Center for Artificial Intelligence, Harpers Ferry, Nov. 7-9, 1991b.