In R.S.Michalski and G.Tecuci (eds), Machine Learning: A Multistrategy Approach, Morgan Kaufmann, 1994.

AN INFERENCE-BASED FRAMEWORK FOR MULTISTRATEGY LEARNING

Gheorghe Tecuci
(George Mason University and Romanian Academy)

Abstract

This chapter describes a general framework for multistrategy learning. One idea of this framework is to view learning as an inference process and to integrate the elementary inferences that are employed by the single-strategy learning methods. Another idea is to base learning on building and generalizing a special type of explanation structure called plausible justification tree which is composed of different types of inference and relates the learner's knowledge to the input. In this framework, learning consists of extending and/or improving the knowledge base of the system so that to explain the input received from an external source of information. The framework is illustrated with a specific method that integrates deeply and dynamically explanation-based learning, determination-based analogy, empirical induction, constructive induction, and abduction.

1 INTRODUCTION

The Knowledge Base (KB) of an intelligent system could be regarded as an internal model of the system's application domain. The more adequately this model approximates the domain, the better is the system's behavior. Therefore, a general learning goal of the system is to continually improve its domain model, either from the point of view of competence or from the point of view of performance. The system is improving its competence if it learns to solve a broader class of problems and to make fewer mistakes in problem solving. Also, it is improving its performance, if it learns to solve more efficiently the problems from its area of competence.

Research in machine learning has elaborated several single-strategy learning methods like, for instance, empirical induction, explanation-based learning, learning by abduction, learning by analogy, case-based learning, which are based on a primary type of inference and illustrate different ways in which a system can learn [Michalski, Carbonell and Mitchell, 1983, 1986; Kodratoff and Michalski, 1990; Shavlik and Dietterich, 1990]. Each of these learning methods could be characterized by the learning task performed. The learning task is defined by the input from which the system learns, the knowledge base (which contains the knowledge that the system can use during learning and which represents the domain model), and the learning goal (which indicates what the system tries to learn). That is, the learning task defines both the applicability conditions and the results of the corresponding learning method. For instance, in the case of empirical induction, in which the primary type of inference is inductive generalization / specialization, the input may consist of *many (positive and/or negative) examples* of some concept C, the KB usually contains only a *small amount of knowledge* related to the input, and the goal is to *learn a description of C* in the form of an inductive generalization of the positive examples that does not cover the negative examples. This description extends the domain model and may improve the competence of the system. In the case of explanation-based learning, in which the primary type of inference is deduction, the input may consist of only *one example* of a concept C, the KB should contain *complete knowledge about the input*, and the goal is to *learn an operational description of C* in the form of a deductive generalization of the input example. This description is a reorganization of some knowledge pieces from the domain model and may improve the performance of the system. In the case of learning by analogy (and case-based learning) the input may consist of *a new entity I*, the KB should contain an *entity S which is similar to I*, and the goal is to *learn new knowledge about the input I* by transferring it from the known entity S. In the case of learning by abduction the input may be *a fact F*, the KB should contain *causal knowledge related to the input* and the goal is to *learn a new piece of knowledge* that would account for the input. Learning by analogy, case based-learning, and learning by abduction extend the domain model with new pieces of knowledge, and usually improve the competence of the system.

This brief characterization of the learning tasks of different single-strategy learning methods shows that these methods have a limited applicability because each requires a special type of input and of background knowledge, and learns a specific type of knowledge.

On the other hand, the complementary nature of these requirements and results naturally suggests that by properly integrating the single-strategy methods, one could obtain a synergistic effect in which different strategies mutually support each other, and compensate for each other's weaknesses. As a result, one may build a multistrategy learning system that may be applicable to a wider spectrum of problems. Each of the multistrategy learning systems that have been built in the last several years illustrates a specific way in which several single-strategy methods could be integrated in order to perform a learning task that could not be performed by a single-strategy method [e.g., Lebowitz, 1986; Danyluk, 1987; Minton and Carbonell, 1987; Pazzani, 1988; Tecuci, 1988; Flann and Dietterich, 1989; Hirsh, 1989; Bergadano and Giordana, 1990; Genest, Matwin and Plante, 1990; Shavlik and Towell, 1990; Tecuci and Kodratoff, 1990; Whitehall, 1990; Wilkins, 1990; Cohen, 1991; Gordon, 1991; De Raedt, 1991; Mooney and Ourston, 1992; Morik, 1992; Ram and Cox, 1992; Reich, 1992; Saitta and Botta, 1992; Widmer, 1992].

After the development of many methods and techniques for the integration of learning strategies, the research in machine learning started to address the problem of defining general principles and frameworks for the design of advanced multistrategy learning systems. One such framework for a multistrategy learning system consists of a cascade of single strategy learning modules, in which the output of one module is an input to the next module. Another framework consists of a global control module and a tool box of single strategy learning modules, all using the same knowledge base. The control module analyzes the relationship between the input and the knowledge base and decides which learning module to activate.

This chapter presents another general framework for multistrategy learning. One idea of this framework is to regard learning as an inference process and to integrate the elementary inferences (like deduction, analogy, abduction, generalization, specialization, abstraction, concretion, etc.) that are employed by the single-strategy learning methods. As a consequence, instead of integrating learning strategies at a macro level (as it is done in most of the current multistrategy systems), one could integrate the different elementary inferences that generate individual learning strategies. Another idea of the framework is to base learning on building and generalizing a special type explanation structure called plausible justification tree which is composed of different types of inference and relates the learner's knowledge to the input. In this framework, learning consists of extending and/or improving the knowledge base of the system so that to explain the input received from an external source of information.

The next section presents this general framework for multistrategy learning. The rest of the paper illustrates the general framework with a specific concept learning and theory revision method that integrates deeply and dynamically explanation-based learning, determination-based analogy, empirical induction, constructive induction, and abduction.

2 INFERENCE-BASED MULTISTRATEGY LEARNING

As mentioned, the KB of an intelligent system could be regarded as a model of the system's application domain. During problem solving, the system uses this model in order to answer questions about the application domain or to find solutions to different problems. For instance, it can answer if a certain fact is true in the real world by simply checking if the fact is explicitly represented in the model or derives form facts that are explicitly represented.

One could regard learning as a reverse process in which the system receives information about the external world as, for instance, a new fact (or a solution to a certain problem), and tries to improve its domain model so that to be able to infer the received input, as well as similar ones (or to solve the input problem and similar ones).

Based on this observation, one could define a general learning scenario in which the system has an incomplete and partially incorrect knowledge base (domain model) and receives new input information about the application domain.

The input may take different forms. It may be a ground fact. It may consist of one or several positive and/or negative examples of a concept. It may also consist of one or several positive (and negative) examples of problem solving episodes, each episode specifying a problem and its solution.

The goal of the system is to improve its KB so as to consistently integrate the information contained in the input. More precisely, after learning from an input I, the KB should be such that a generalization of I is inferable from the KB.

The general learning strategy is based on understanding the input in terms of the current KB. This means that the system will try to build a plausible justification tree that demonstrates that the input is a plausible consequence of the knowledge from the KB.

A plausible justification tree is like a proof tree, except that the inferences which compose it may be the result of different types of reasoning (not only deductive, but also analogical, abductive, probabilistic, fuzzy, etc.).

Let us suppose, for instance, that the learner's input is a new fact

$$P_n(a,b) \tag{1}$$

where 'P_n' is a predicate and 'a', 'b' are object names or object properties.

To "understand" this input the learner would try to build a plausible justification tree as the one from Figure 1. The root of the tree is the input fact, the leaves are facts from the KB, and the intermediate nodes are intermediate facts generated during the "understanding" process. The branches connected to any given node link this node with facts, the conjunction of which *certainly* or *plausibly implies* the fact at the node, according to the learner's KB. The notion "plausibly implies" means that the target (parent node) can be inferred from the premises (children nodes) by some form of plausible reasoning, using the learner's KB. The branches together with the nodes they link represent individual inference steps which could be the result of different types of reasoning.

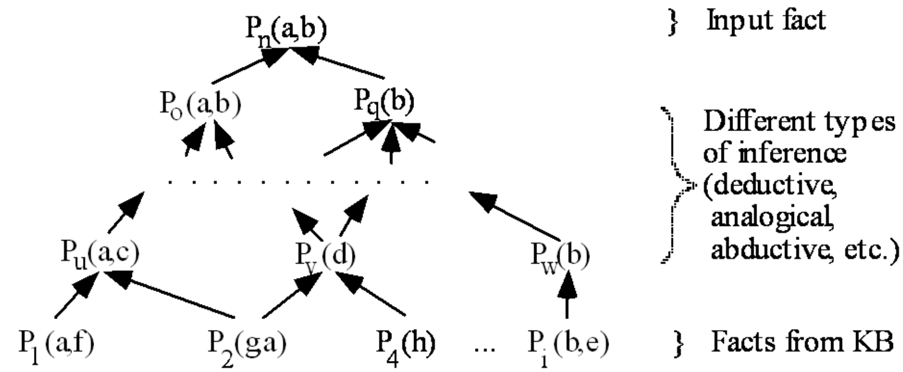


Figure 1: A plausible justification tree for the input $P_n(a,b)$

For example, the inference step[*]

$$P_1(a,f)\ \&\ P_2(g,a) \square P_u(a,c) \tag{2}$$

(see Figure 1) may be the result of deduction based on the following deductive rule from the KB:

$$\forall s\ \forall t\ \forall z\ \forall p\ (P_1(s,t)\ \&\ P_2(z,s) \square P_u(s,p)) \tag{3}$$

Also, the inference step

$$P_2(g,a)\ \&\ P_4(h) \varnothing P_v(d) \tag{4}$$

may be the result of analogy with the following implication from the KB:

$$P_2(g',a')\ \&\ P_4(h') \square P_v(d') \tag{5}$$

Indeed, suppose that the system determined that g, a, h and d are similar to g', a', h' and d', respectively. By analogy, the system concludes that from $P_2(g,a)$ & $P_4(h)$ one can plausibly infer $P_v(d)$, and hence (4).

The inference step

$$P_i(b,e) \square P_w(b) \tag{6}$$

may be the result of abduction based on the following causal relationship from the KB

$$\forall r\ (P_i(r,e) \square P_w(r)) \tag{7}$$

Indeed, let us suppose that the predicate $P_w(b)$ would need to be true in order to build the plausible justification tree in Figure 1. Because $P_w(b)$ matches the right hand side of (7), one may trace backward this rule and hypothesize that $P_i(b,e)$ is true.

An inference step could also result from a combination of empirical generalization and deduction, which is called inductive prediction (see section 4.4). To illustrate this, let us suppose that the KB contains the following examples of the concept $P_n(x,y)$:

---

[*] Throughout this paper we use : : > to denote concept assignment, □ to denote certain (deductive) implication, ∅ to denote plausible implication, and --> to denote plausible determination.

$$P_o(c,f) \ \&\ P_q(c) \ \&\ P_j(b,e) \ ::>\ P_n(c,f) \quad (8)$$

$$P_o(d,g) \ \&\ P_q(d) \ \&\ P_k(b) \ ::>\ P_n(d,g)$$

These examples can be empirically generalized to the rule

$$\forall x \forall y\ (P_o(x,y) \ \&\ P_q(y) \varnothing P_n(x,y)) \quad (9)$$

Rule (9) could then be used to produce the following plausible inference step:

$$P_o(a,b) \ \&\ P_q(b) \varnothing P_n(a,b) \quad (10)$$

In a more complex case, available examples may not be so easily generalizable to (9), and the system may have to use constructive induction.

Thus, the tree in Figure 1 shows that $P_n(a,b)$ is a plausible consequence of facts that are explicitly represented in the system's KB.

The understanding process proceeds in the same way when the input is an example of a concept or a specific solution to some problem.

Indeed, let us suppose that the input is the following example of the concept $P_n(x,y)$:

$$P_1(a,f) \ \&\ P_2(g,a) \ \&\ P_3(b) \ \&\ P_4(h) \ \&...\&\ P_j(b,c) \ ::>\ P_n(a,b) \quad (11)$$

Then the system will try to "understand" it by building the plausible justification tree in Figure 1 in which the leaves are facts from the left hand side of (11), the intermediate nodes are intermediate facts generated during the "understanding" process, and the top is the right hand side of (11). Thus, the plausible justification tree shows that (11) is indeed an example of the concept $P_n(x,y)$.

Let us now suppose that the input is the following example of problem solving episode:

to achieve the goal: $P_n(a,b)$ (12)

perform the actions: $P_u(a,c)$, $P_w(b)$, ... , $P_m(c)$

In this case, the top of the justification tree is the goal $P_n(a,b)$. The intermediate nodes are the actions $P_u(a,c)$, $P_w(b)$, ... , $P_m(c)$, or facts that are preconditions of these actions, or facts that are effects of these actions, or other generated facts that plausibly imply these preconditions or derive from these effects. The leaves are facts from the knowledge base which are either preconditions of some of the actions $P_u(a,c)$, $P_w(b)$, ... , $P_m(c)$, or plausibly imply these preconditions. Therefore, in this case, the plausible justification tree is a plan which shows that, in the context of the current KB, the system could achieve the goal $P_n(a,b)$ by performing the actions $P_u(a,c)$, $P_w(b)$, ... , $P_m(c)$.

An important result of building the plausible justification tree is the generation of new pieces of knowledge which extend the KB so that to infer the input.

In the case of the justification tree in Figure 1, these new pieces of knowledge are:

$P_2(g,a) \ \&\ P_4(h) \varnothing P_v(d)$	(generated through analogy)
$P_i(b, e)$	(generated through abduction)
$P_o(a,b) \ \&\ P_q(b) \varnothing P_n(a,b)$	(generated through inductive prediction)

By asserting these pieces of knowledge into the KB, the system is able to deductively infer the input. The new pieces of knowledge are supported by the fact that they allow building a logical connection (the plausible justification tree) between a KB that represents parts of the real world, and a piece of knowledge (the input) that is known to be true in the real world.

In general, a learning system should try to learn as much as possible from any input it receives. In the case of the considered learning scenario, it may do so by generalizing the plausible justification tree as much as allowed by the knowledge used to build it in the first place. By this, it will generalize the hypothesized knowledge, so that the resulting KB will entail not only the received input, but also similar ones.

One way to generalize the plausible justification tree is presented in section 4.6. It consists of replacing each implication with a plausible generalization and then in unifying the connections between these generalized implications.

To illustrate this process, take, for instance, the inference step (2), i.e., $P_1(a,f)$ & $P_2(g,a) \square P_u(a,c)$. This inference step can be locally generalized into the rule (3) that allowed it, i.e., $\forall s \forall t \forall z \forall p\ (P_1(s,t)\ \&\ P_2(z,s) \square P_u(s,p))$. The branches of the tree in Figure 1, corresponding to the original inference step (2), are then replaced by the appropriate components of this rule. This is a *deductive generalization.*

Let us now consider the inference step (4), i.e., $P_2(g,a)$ & $P_4(h) \varnothing P_v(d)$. This step was made by analogy with implication (5), i.e., $P_2(g',a')$ & $P_4(h') \square P_v(d')$. The generalization of this analogical inference is based on the idea that a similarity of an entity to a given entity generates an equivalence class of all entities similar to the given entity. Following this idea, one may generate a conjunctive generalization that covers all the inferences that could be derived by analogy with (5):

$$P_2(g'',a'')\ \&\ P_4(h'') \varnothing P_v(d'') \qquad (13)$$

where g'', a'', h'' and d'' represent classes that contain g and g', a and a', h and h', d and d', respectively. This is a *generalization based on analogy.*

The abductive step (6) $P_i(b,e) \square P_w(b)$ is replaced by $P_i(r,e) \square P_w(r)$, according to the general rule (7), i.e. $\forall r\ (P_i(r,e) \square P_w(r))$. This generalization is justified because a system abducing $P_i(b,e)$ in order to explain $P_w(b)$, would also abduce $P_i(r,e)$ in order to explain $P_w(r)$, for any r. This is a *generalization based on abduction.*

Finally, the inference step (10) $P_o(a,b)$ & $P_q(b) \varnothing P_n(a,b)$ was done by applying the rule (9), i.e. $\forall x \forall y (P_o(x,y) \& P_q(y) \varnothing P_n(x,y))$, which was obtained by empirical generalization. Then the corresponding branch is replaced by rule (9). This is an *empirical inductive generalization.*

As one could notice, the generalization of an implication depends of the type of inference that generated it and of the system's knowledge. This idea opens a new research direction in the theory of generalization by suggesting that a specific type of generalization may be associated with each type of inference. Consequently, a learning system may perform not only deductive and empirical generalizations, but also generalizations based on analogy, on abduction, etc.

As mentioned above, after all the inference steps are locally generalized, the system unifies their connections. In particular, it makes the following unifications: (x=s=a", z=g", y=r). In this way the system builds the general plausible justification tree from Figure 2.

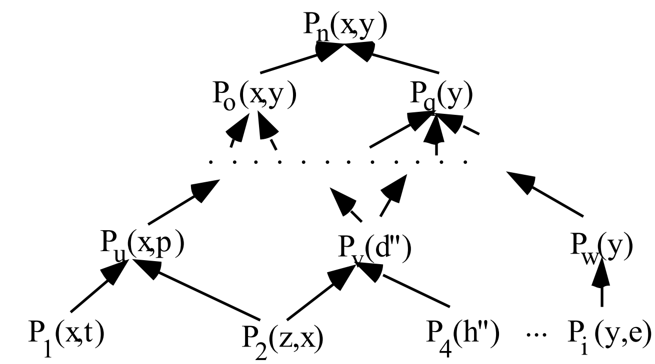


Figure 2: A generalization of the justification tree from Figure 1

During the generalization of the plausible justification tree, some of the previously learned knowledge may also be generalized as, for instance, the analogical implication (4) $P_2(g,a)$ & $P_4(h) \varnothing P_v(d)$, which was generalized to the rule (13) $P_2(g",a")$ & $P_4(h") \varnothing P_v(d")$.

Other general knowledge pieces have been generated during the building of the plausible justification tree in Figure 1. An example of such a knowledge piece is the rule (9).

The generalized plausible justification tree shows how a generalization of the input is inferrable from the KB. However, this tree was built by making plausible inferences and plausible generalizations and is therefore less certain.

The system may improve the generalized justification tree, as well as the knowledge pieces learned during its buiding, by learning from additional input (other facts of the form $P_n(a_k,b_k)$, concept examples, or examples of problem solving episodes).

For each new positive example E_i, the system will generalize the plausible justification tree in Figure 2 so as to cover a plausible justification of E_i. Also, some of the knowledge pieces from the KB may be generalized so as to cover inferences from the plausible justification of E_i.

For each new negative example N_j, the system will specialize the general plausible justification tree so as to no longer cover a plausible justification tree which would show that N_j is a positive example. This may also require the specialization of some knowledge pieces from the KB as, for instance, the rule (9) or the rule (13).

After all the examples have been processed, the system may extract several (operational or abstract) concept definitions from the final general justification tree. For instance, if the final general justification tree is the one from Figure 2, then its leaves represent an operational definition of $P_n(x,y)$:

$$P_1(x,t) \ \& \ P_2(z,x) \ \& \ P_4(s) \ \&...\& \ P_i(y,e) \ ::> P_n(x,y) \tag{14}$$

Also, the top part of the justification tree represents the most abstract characterization of $P_n(x,y)$:

$$P_o(x,y) \ \& \ P_q(y) \ ::> P_n(x,y) \tag{15}$$

Other learnable knowledge pieces are various abstractions of the input examples. For instance, one abstraction is obtained by instantiating the variables in the above abstract characterization (eq.(15)), to specific arguments in a certain example:

$$P_o(a,b) \ \& \ P_q(b) \ ::> P_n(a,b) \tag{16}$$

Other abstractions would correspond to lower levels of the generalized justification tree.

As a result of this learning process the system may increase its problem solving abilities both in terms of competence and performance. Indeed, if the input was a new fact, then the system will be able to predict that certain similar facts are true in the real world. Or, if the input was a sequence of positive (and negative) examples of some concept, then the system will be able to predict if a new instance is (or is not) an example of the learned concept. Also, if the system retains operational definitions like (14), then it may be able to make these predictions faster. If the input was an example of a problem solving episode, represented by a problem P and its solution S, then the system will be able to solve problems similar to P, by proposing solutions similar to S.

Table 1 synthesizes the main steps of this learning methodology.

Table 1: The learning methodology

- For the first positive example I_1:
  - build a plausible justification tree T_1 of I_1;
  - build the plausible generalization T_u of T_1;
  - generalize the KB so that to entail T_u.
- For each new positive example I_i:
  - generalize T_u so that to cover a plausible justification tree T_i of I_i;
  - generalize the KB so that to entail the new T_u.
- For each new negative example I_i:
  - specialize T_u so that no longer to cover any plausible justification tree T_i of I_i;
  - specialize the KB so that to entail the new T_u without entailing the previous T_u.
- After all the examples have been processed:
  - extract different concept definitions from T_u.

By generalization of the KB one means any transformation that results in an increase of knowledge inferable from the KB. The KB may be generalized by generalizing knowledge pieces or by simply adding new knowledge pieces.

Similarly, by specialization of the KB one means any transformation that results in a decrease of knowledge inferable from the KB. The KB may be specialized by specializing knowledge pieces or by simply removing knowledge pieces from the KB.

It is important to stress that the types of inference from the plausible justification trees (see Figure 1 and Figure 2) and the order in which they are performed, depend of the relationship between the input and the KB. Consequently, a learning method developed in the presented framework will integrate dynamically different types of elementary reasoning mechanisms. Moreover, as will be shown in section 7, if a particular learning task corresponds to a single-strategy method, then the behavior of the system should correspond to the application of such a method. Such an adaptive integration of different learning strategies (that seems also to be a characteristic of human learning) has been called multistrategy task-adaptive learning, or MTL for short (Michalski, 1990, 1992; Tecuci and Michalski, 1991a,b).

The next sections present a specific Multistrategy Task-adaptive Learning method that was developed in this framework. The method, called MTL-JT (Multistrategy Task-adaptive

Learning based on building plausible Justification Trees), integrates deeply and dynamically explanation-based learning, determination-based analogy, empirical induction, constructive induction, and abduction, in order to learn from one or from several positive (and negative) concept examples.

3 THE LEARNING TASK OF MTL-JT

The learning task of MTL-JT is presented in Table 2. As one could notice, it is both a theory revision task and a concept learning task. The main steps of the MTL-JT method are those presented in Table 1. A detailed presentation of the method is given in (Tecuci, 1992a).

Table 2: The learning task of MTL-JT

Input: *one or several positive (and negative) examples of a concept.*
The examples are represented as conjunctions of first-order predicates, are considered noise-free, and are presented in sequence.

Knowledge Base: *incomplete and partially incorrect.*
The KB may include a variety of knowledge types (facts, examples, implicative or causal relationships, determinations, etc.), represented with first-order predicates.

Goal: *improve the KB and learn different concept definitions from the input example(s).*
The learned concept definitions may be operational or abstract, and the KB is improved by both generalizing and specializing it, so that to entail these definitions.

One may notice the generality of the learning goal. In a specific application of this learning method, this goal would need to be specialized. For instance, some of the learnable concept definitions may not be useful and, consequently, will not be learned.

In order to illustrate this learning task and the corresponding learning method, we shall consider the case of a learning system in the area of geography. The purpose of the system is that of acquiring geographical data and rules in order to answer questions about geography.

Let us consider, for instance, that the knowledge base is the one from Table 3. It contains several ground facts, two examples of fertile soil, a plausible determination rule and four deductive rules.

Table 3: A sample of an incomplete and partially incorrect KB

Facts:

terrain(Philippines, flat), rainfall(Philippines, heavy), water-in-soil(Philippines, high)

Examples *(of fertile soil):*

soil(Greece, red-soil) : : > soil(Greece, fertile-soil)

terrain(Egypt, flat) & soil(Egypt, red-soil) : : > soil(Egypt, fertile-soil)

Plausible determination:

rainfall(x, y) --> water-in-soil(x, z)

Deductive rules:

∀ x, soil(x, loamy) □ soil(x, fertile-soil)

∀ x, climate(x, subtropical) □ temperature(x, warm)

∀ x, climate(x, tropical) □ temperature(x, warm)

∀ x, water-in-soil(x, high) & temperature(x, warm) & soil(x, fertile-soil) □ grows(x, rice)

Let us also consider that the input consists of the sequence of concept examples from Table 4. The left hand side of each positive example (negative example) is the description of a country that grows rice (does not grow rice), and the right hand side is the statement that the respective country grows rice (does not grow rice).

Table 4: Positive and negative examples of "grows(x, rice)"

Positive Example 1:

rainfall(Thailand, heavy) & climate(Thailand, tropical) & soil(Thailand, red-soil) &
terrain(Thailand, flat) & location(Thailand, SE-Asia) : : > grows(Thailand, rice)

Positive Example 2:

rainfall(Pakistan, heavy) & climate(Pakistan, subtropical) & soil(Pakistan, loamy) &
terrain(Pakistan, flat) & location(Pakistan, SW-Asia) : : > grows(Pakistan, rice)

Negative Example 3:

rainfall(Jamaica, heavy) & climate(Jamaica, tropical) & soil(Jamaica, loamy) &
terrain(Jamaica, abrupt) & location(Jamaica, Central-America) : : > ¬ grows(Jamaica, rice)

The different types of knowledge pieces learned from the above KB and input examples are presented in Table 5.

Table 5: The learned knowledge

Concept definitions

Operational definition of "grows(x, rice)":

{ rainfall(x, heavy) & terrain(x, flat) & [climate(x, tropical) ∆ climate(x, subtropical)] & [soil(x, red-soil) ∆ soil(x, loamy)] } : : > grows(x, rice)

Abstract definition of "grows(x, rice)":

water-in-soil(x, high) & temperature(x, warm) & soil(x, fertile-soil) : : > grows(x, rice)

Abstraction of Example 1:

water-in-soil(Thailand, high) & temperature(Thailand, warm) & soil(Thailand, fertile-soil) : : > grows(Thailand, rice)

Improved KB

New facts:

water-in-soil(Thailand, high), water-in-soil(Pakistan, high)

New plausible rule:

∀ x, soil(x, red-soil) ∅ soil(x, fertile-soil)

with the positive examples: (x<-Greece), (x<-Egypt), (x<-Thailand).

Specialized plausible determination:

rainfall(x, y) & terrain(x, flat) --> water-in-soil(x, z)

with the positive examples: (x<-Philippines, y<-heavy, z<-high),
(x<-Thailand, y<-heavy, z<-high),
(x<-Pakistan, y<-heavy, z<-high).

with the negative example: (x<-Jamaica, y<-heavy).

One result of learning consists of several concept definitions [Michalski, 1990].

The first definition in Table 5 is an operational definition of "grows(x, rice)", expressed with the features present in the input examples. The second definition is an abstract definition of "grows(x, rice)", expressed with more general features, derived from those present in the input examples (since this rule was already known, the new knowledge is just that it represents an abstract definition). The third definition is an abstraction of Example 1 that was obtained by instantiating the previous abstract definition.

The other result of learning is the improvement of the KB so that to entail the learned concept definitions.

The KB was generalized by learning two new facts and a rule. It was also specialized, by conjunctively adding a literal to the left hand side of the plausible determination.

As indicated in Table 5, the system also retains all the examples of the learned knowledge pieces in order to update them when new knowledge becomes available. These examples have been generated through different forms of plausible reasoning and have been validated during the learning process. Therefore, they also represent an improvement of the KB.

The next three sections illustrate the MTL-JT method, which follows the steps presented in Table 1.

4 LEARNING FROM THE FIRST EXAMPLE

4.1 Building the plausible justification tree

As shown in Table 1, the first step of the learning method consists of building a plausible justification tree for the first example received by the system. In MTL-JT, the system builds an AND/OR tree by conducting a top-down uniform-cost search [Nilsson, 1971].

The developed AND/OR tree contains several AND trees, each having a cost that estimates its global plausibility. The cost of a partial AND tree is computed as a tuple (m, n), where m represents the number of the deductive implications in the tree, and n represents the number of the non-deductive implications (which, in the case of MTL-JT, could be obtained by analogy, inductive prediction or abduction). The ordering relationship for the cost function is defined as follows:

$$(m1,\ n1) < (m2,\ n2) \text{ if and only if } n1<n2 \text{ or } (n1=n2 \text{ and } m1<m2)$$

This cost function guarantees that the system will find the justification tree with the fewest number of non-deductive implications. In particular, it will find a deductive tree (if one exists) and the deductive tree with the fewest implications (if several exist).

As a general strategy, the system always tries to justify a given predicate (for instance, "grows(Thailand, rice)" in Figure 3) by deduction. If it succeeds, then it tries to justify the resulting predicates (i.e., "water-in-soil(Thailand, high)", "temperature(Thailand, warm)" and "soil(Thailand, fertile-soil)"). However, if it fails, then it tries to justify the predicate by using as many plausible reasoning methods as possible. It will try these methods in order: first analogy, then inductive prediction and lastly abduction. If one of them produces a plausible inference step, then the system tries the remaining ones in order to confirm or to contradict it. If no contradiction is found, the inference step is accepted. This method (although quite simple and definitely a necessary topic of future research) is related to that employed by humans. Indeed, Collins and Michalski (1989) argue that people solve problems by pursuing different "lines of reasoning". They estimate the "strength" of each line of reasoning, and make their conclusion on the basis of this evaluation. If the lines lead to the same conclusion, they have a strong belief in the result. If the lines lead to different conclusions, and the associated "strengths" are roughly similar, people restrain from making any decisive conclusion.

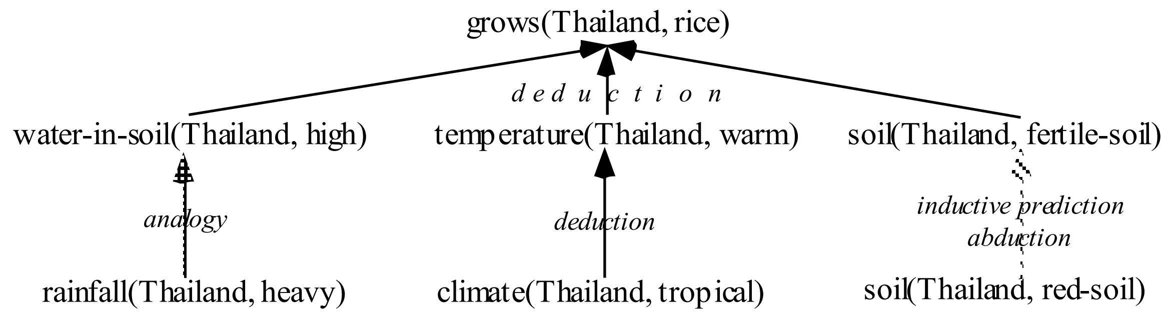


Figure 3: The plausible justification tree T_1 of Example 1

It should be noticed that, although at the level of a given inference step, the different reasoning methods are tried in a predefined order, globally (at the level of the resultant justification tree) there is no predefined order. For instance, in the case of the justification tree in Figure 3, the order of the inference steps was: deduction, analogy, deduction, inductive prediction and abduction. In general, this order depends of the relationship between the KB and the input. Therefore, the MTL-JT method is an example of a dynamic and deep (i.e. at the level of individual inference steps) integration of single strategy learning methods (each learning method corresponding to a specific type of inference).

The next sections present briefly the way the different inference steps in Figure 3 have been made.

4.2 Deduction

Two inference steps in Figure 3 are the results of deductions based on the deductive rules from Table 2, as illustrated in the following:

$$\begin{array}{l} \forall\ x, \text{climate}(x, \text{tropical}) \rightarrow \text{temperature}(x, \text{warm}) \\ \text{climate}(\text{Thailand, tropical}) \\ \hline \text{temperature}(\text{Thailand, warm}) \end{array}$$

4.3 Analogy

Analogical inference is the process of transferring knowledge from a known entity S to a similar but less known entity T. S is called the *source* since it is the entity that serves as a source of knowledge, and T is called the *target* since it is the entity that receives the knowledge. The central intuition supporting this type of inference is that if two entities, S and T, are similar in some respects, then they could be similar in other respects as well. Therefore, if S has some feature, then one may infer by analogy that T has a similar feature.

In MTL-JT one uses a simple form of analogy based on plausible determinations defined as follows:

P(x, y) --> Q(x, z) (P plausibly determines Q) meaning
$\forall$S, $\forall$T { If $\exists$y [P(S, y) & P(T, y)] then it is probably true that $\exists$z [Q(S, z) & Q(T, z)] }
where P and Q are first order logical expressions.

Otherwise stated, if the source S and the target T are characterized by the same feature P (i.e., P(S, y_0)=true and P(T, y_0)=true) then it is probably true that they are also characterized by a same feature Q. Therefore, if Q(S, z_0)=true then one may infer by analogy that Q(T, z_0)=true.

We use the term "probably true" to express that the determination-based analogy we are considering is a weak inference method that does not guarantee the truth of the inferred knowledge. This is similar to the mutual dependency rules introduced by [Collins and Michalski, 1989; Michalski, 1992], but different from the determination rules introduced by [Davies and Russell, 1977] which guarantee the truth of the inferred knowledge.

The analogical inference step in Figure 3 was made by using the plausible determination

rainfall(x, y) --> water-in-soil(x, z)

(i.e. the rainfall of an area determines the quantity of water in the soil of that area). Indeed, Philippines and Thailand are similar from the point of view of "rainfall" (in both cases this is heavy). Therefore, one may infer by analogy that the two countries are also similar from the point of view of "water-in-soil". Thus, the system concluded that "water-in-soil(Thailand, high)" from the fact "water-in-soil(Philippines, high)".

One should notice that a plausible determination rule indicates only what kind of knowledge could be transferred from a source to a target (knowledge about "water-in-soil", in the case of the considered determination), and in what conditions (the same type of "rainfall"). It does not indicate, however, the exact relationship between the type of the rainfall (for instance, "heavy") and the quantity of water in soil ("high"). The exact relationship is indicated by the source entity ("Philippines"). Therefore, a plausible determination rule alone (without such a source entity), cannot be used in the inference process.

In general, the MTL methods are intended to incorporate different forms of analogy, based on different kinds of similarities, such as similarities among causes, relations, and meta-relations.

4.4 Inductive prediction

Inductive prediction consists in finding an inductive generalization of a set of examples of a concept and in applying it in order to predict if a new instance is (or is not) a positive example of the concept.

The generalization of the examples could be obtained through a process of empirical or constructive generalization. The generalization process is empirical if it involves only descriptors from the description space of the examples, and is constructive if it introduces new descriptors which do not belong to the description space of the examples. A detailed characterization of empirical and constructive generalization is given in [Michalski, 1992].

In MTL-JT one uses an inductive generalization method that determines the most specific generalization of a set of positive examples that does not cover any of the negative examples. Moreover, the system is retaining all the examples in order to update the generalization when new examples become available.

One inference step in Figure 3 was the result of inductive prediction. Indeed, in order to prove that "soil(Thailand, fertile-soil)" is true, the system looked into the KB for examples of "fertile-soil". Then it inductively generalized them to a rule that was used to predict the inference step from Figure 3, as indicated in Table 6.

It is important to stress that the system keeps the learned rule in the KB as an inductive generalization. Therefore, future applications of this rule are also inductive predictions. Let us also notice that the rule in Table 6 was obtained through an empirical generalization process because it is expressed only in terms of the descriptors used in the examples.

Table 6: Making an inference through inductive prediction

Examples from KB:

soil(Greece, red-soil) : : > soil(Greece, fertile-soil)

terrain-type(Egypt, flat) & soil(Egypt, red-soil) : : > soil(Egypt, fertile-soil)

Inductive generalization:

∀ x, soil(x, red-soil) ∅ soil(x, fertile-soil)

with the positive examples: (x<-Greece)(x<-Egypt)

Predicted inference:

soil(Thailand, red-soil) ∅ soil(Thailand, fertile-soil)

4.5 Abduction

In general, abduction is defined as follows [Josephson, 1991]:

D is a collection of data;
H explains D;
<u>*No other hypothesis is able to explain D as well as H does;*</u>
Therefore, H is probably true.

According to this definition, abduction involves two steps, generation of explanatory hypotheses and selection of the "best" hypothesis.

In MTL-JT one considers two forms of abduction:

a) tracing backward a deductive rule

If D is to be explained and H □ D then hypothesize H.

In particular, *if* $H=H_1$ *&* H_2 *& ... &* H_n *and* H_2 *& ... &* H_n *is true then hypothesize* H_1.

b) hypothesizing an ISA relationship (i.e. d_1 ISA d_2)

If $P(a, d_2)$ *is to be explained and* $P(a, d_1)$ *is true*
then hypothesize that $P(a, d_1)$ ∅ $P(a, d_2)$.

Choosing the "best" abductive hypothesis is the most difficult problem of abductive learning. This is somewhat simplified in the context of MTL-JT because the system is trying to make an inference step through as many plausible inference methods as possible and abduction is the last one to try (as shown in section 4.1). Therefore, if an inference "H ∅ D" has been made through some other form of reasoning, abduction is used only to confirm this inference or to contradict it (i.e. to prove "H ∅ C", where D & C = false).

In the absence of the above criterion, the system chooses the abductive hypotheses in the following order:

- prefer the ISA abductions;
- prefer to backtrace the rule H_1 & H_2 & ... & H_n □ D with the highest number of true antecedents;
- prefer to backtrace the rule that has the highest number of known instances;
- prefer the simplest hypothesis.

In the case of the plausible justification tree in Figure 3, the system made an ISA abduction confirming the previously made inductive prediction from Table 6. Indeed, "soil(Thailand, fertile-soil)" needed to be proven and "soil(Thailand, red-soil)" was known to be true. Therefore, the system abduced the ISA relationship:

soil(Thailand, red-soil) ∅ soil(Thailand, fertile-soil) 17)

4.6 Generalization of the plausible justification tree

Once a justification tree was successfully created, the system analyzes the individual implications associated with the elementary inference steps to determine if these implications could be locally generalized within the constraints of the KB that were used to make the inference steps. After the implications are generalized locally, the system unifies them globally, and builds a generalized justification tree. This technique is an extension of the one elaborated by [Mooney and Bennet, 1986]. The extension concerns the way individual implications are generalized, by using the knowledge from which they were derived. The idea is to replace each implication A ∅ B (or A □ B) with the least general generalization of all the similar implications that could be obtained from the knowledge that produced it [Tecuci & Michalski, 1991b].

A deductive implication is replaced by the deductive rule that generated it. This is a *deductive generalization.* For instance

water-in-soil(Thailand, high) & temperature(Thailand, warm) & soil(Thailand, fertile-soil)
□ grows(Thailand, rice)

is replaced by

∀ x, water-in-soil(x1, high) & temperature(x1, warm) & soil(x1, fertile-soil) □ grows(x1, rice).

An analogical implication is generalized by considering the knowledge used to derive it. In our example, the implication

rainfall(Thailand, heavy) ∅ water-in-soil(Thailand, high)

was obtained by analogy with "rainfall(Philippines, heavy)" and "water-in-soil(Philippines, high)", based on the plausible determination

rainfall(x, y) --> water-in-soil(x, z).

Because the system would infer "water-in-soil(x2, high)" for any x2 such that "rainfall(x2, heavy)", the analogical implication is generalized to:

$\forall$ x2, rainfall(x2, heavy) ∅ water-in-soil(x2, high).

This is a *generalization based on analogy.*

An implication obtained through inductive prediction is generalized to the rule that produced it. Therefore, the predicted inference from Table 6 would be replaced with the *empirical inductive generalization* from Table 6.

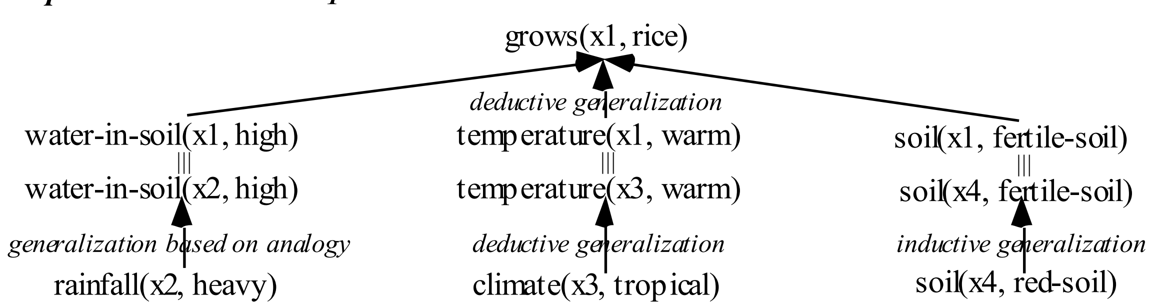
An abductive implication obtained by tracing backward a deductive rule would be generalized to that rule. However, for an abduced ISA relationship there is no knowledge that could be used to generalize it. Therefore it would remain unchanged in the explanation structure.

An implication obtained through several forms of reasoning is generalized to the least general expression that covers the generalizations corresponding to individual reasoning methods. Therefore, the implication from Figure 3, which was obtained both through inductive prediction and abduction, is generalized to the least general generalization of the rule in Table 6 and of the abduced ISA relationship (17):

$\forall$ x4, soil(x4, red-soil) ∅ soil(x4, fertile-soil).

The generalization of the implications from Figure 3 form the explanation structure S_1 from the top part of Figure 4. To transform this explanation structure into a general justification tree one has to determine the most general unification of the connection patterns which, in this case is (x1=x2=x3=x4=x). By making these unifications one obtains the tree from the bottom of Figure 4 which represents the most general plausible generalization of the justification tree from Figure 3.

The explanation structure S_1:



The plausible generalization T_u of T_1:

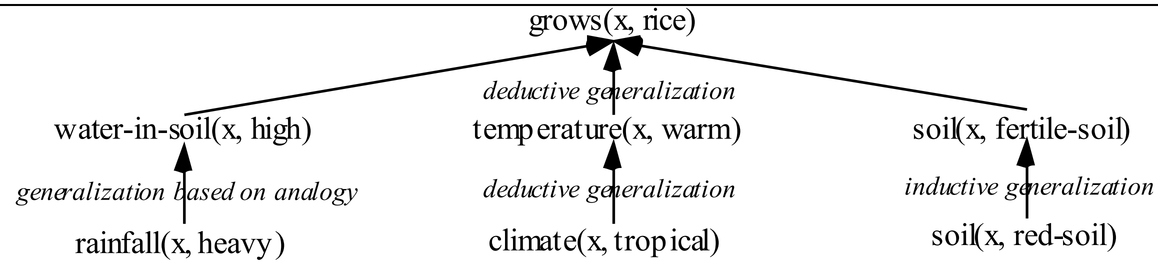


Figure 4: Building the plausible generalization T_u of the justification tree T_1 in Figure 3

As mentioned in section 2, an interesting research direction suggested by the generalization of the plausible justification trees is to investigate different forms of generalizations, not only deductive and inductive, but also analogical, abductive, etc.

4.7 Generalization of the KB

As indicated in Table 1, the system will generalize the KB so as to entail the tree T_u in Figure 4. In this case it learned a new fact (by analogy)

water-in-soil(Thailand, high),

positive examples of the determination "rainfall(x, y) --> water-in-soil(x, z)"

(x<-Philippines, y<-heavy, z<-high),

(x<-Thailand, y<-heavy, z<-high)

and a rule (by empirical generalization)

∀ x, soil(x, red-soil) ∅ soil(x, fertile-soil)

with the positive examples: (x<-Greece), (x<-Egypt), (x<-Thailand).

5 LEARNING FROM A NEW POSITIVE EXAMPLE

5.1 Generalization of the plausible justification tree

Let us now consider that the system receives Example 2 in Table 3. As indicated in Table 1, the system tries to generalize the current justification tree T_u from Figure 4 so that to cover a justification of the new positive example. This process is illustrated in Figure 5.

First of all, the system determines the instance of the general tree T_u in Figure 4, corresponding to Example 2 in Table 3. Then it analyzes the leaf predicates and the inference steps from this tree. If the leaf predicates are true and the inference steps are plausible, then this tree is a plausible justification of the new positive example that is already covered by the general justification tree T_u in Figure 4. This ends the processing of the current example. However, this tree is not a correct justification of Example 2 because the leaf predicates "climate(Pakistan, tropical)" and "soil(Pakistan, red-soil)" are not true. Therefore, the system uses the deductive rules

"∀ x, climate(x, subtropical) □ temperature(x, warm)" and

"∀ x, soil(x, loamy) □ soil(x, fertile-soil)"

from Table 2, and builds a correct plausible justification tree T_2.

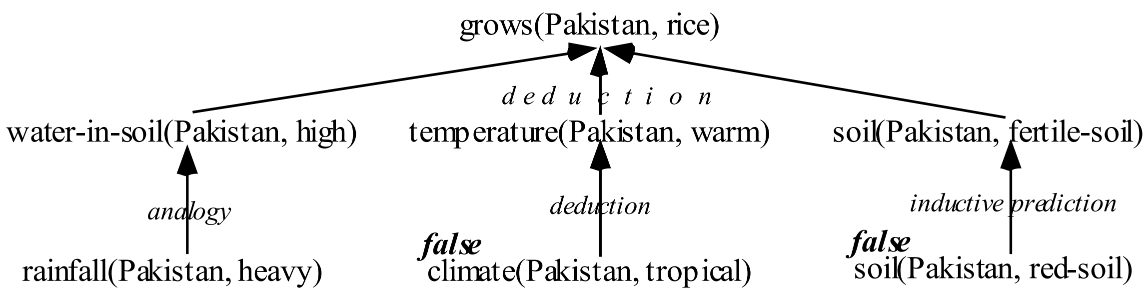
It is important to notice that this plausible justification tree of Example 2 has been built by using the plausible justification tree of the previous example (Example 1). This not only facilitates the process of building the justification tree, but also the process of generalizing the general tree T_u, as will be shown in the following. Moreover, it shows some similarities between this method and human learning which involves the use of the explanations of previous examples in the process of building an explanation for a new example [Wisniewski and Medin, 1991].

The next step of the learning process is to build the explanation structure S_2 that has two general components to be unified:

- the tree T_u in Figure 4,
- the generalization of the part of the tree T_2 in Figure 5 that is specific to it (this generalization being made according to the procedures described in section 4.6).

As the result of the unification of the connection patterns from the explanation structure S_2 one obtains the general justification tree T_u from the bottom of Figure 5. This general tree covers both the justification tree of Example 1 and that of Example 2.

Instance of the current T_u corresponding to Example 2:



Plausible justification tree T_2 of Example 2:

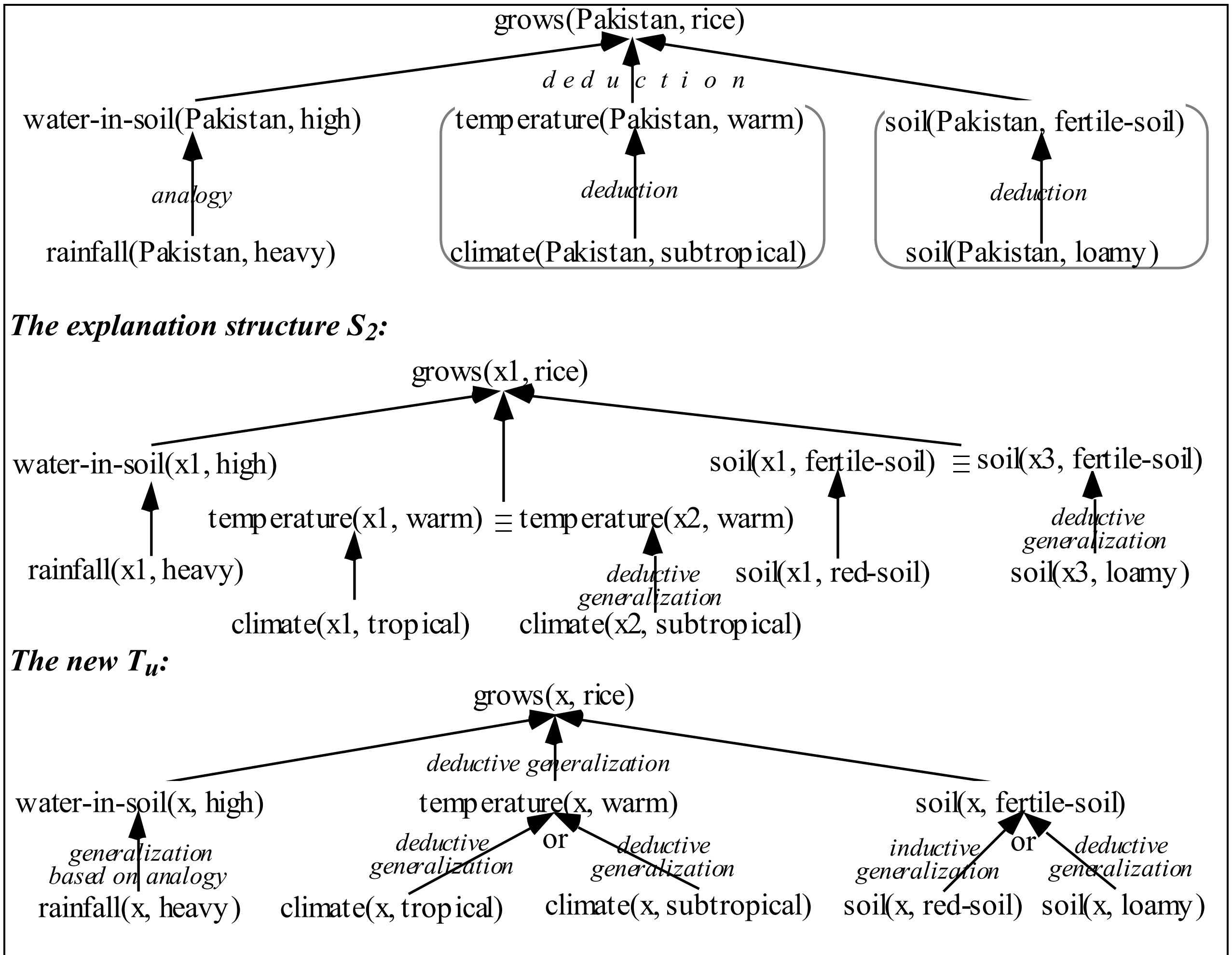


Figure 5: Generalization of T_u so that to cover the justification tree T_2 of Example 2

It should be noticed that, although the justification trees of individual positive examples are AND trees, the generalization of these trees is, in general, an *AND/OR tree.* This is also the case with the tree T_u in Figure 5. Indeed, "grows(x, rice)" is an AND node, "climate(x, warm)" is an OR nodes, and "soil(x, fertile-soil)" is also an OR node.

5.2 Generalization of the KB

Besides generalizing the tree T_u, another result of learning from Example 2 consists of extending the KB with a new fact

water-in-soil(Pakistan, high)

and a new positive example of the determination "rainfall(x, y) --> water-in-soil(x, z)"

(x<-Pakistan, y<-heavy, z<-high).

6 LEARNING FROM NEGATIVE EXAMPLES

6.1 Specialization of the general justification tree

Let us now consider that the system receives the Negative Example 3 from Table 3. As indicated in Table 1, the system tries to specialize the general justification tree T_u so as no longer to cover any justification of the negative example. In the same time, it may need to specialize the KB so as to entail the new T_u without entailing the previous T_u.

Again the system builds the instance of the general justification tree T_u in Figure 5, instance corresponding to this new example (see Figure 6). This tree would lead to the wrong conclusion that the current input is a positive example of "grows(x, rice)". Therefore, the tree must contain some false leaf facts or false implications. These have to be detected, and both the general justification tree T_u in Figure 5 and the KB should be specialized, so as no longer to contain them. One should notice that this is a limited specialization of the KB. Further specializing the KB so that no longer to entail *any* plausible justification of the negative example does not seem to be an obvious goal for a plausible reasoner that, by definition, may also reach some false conclusions.

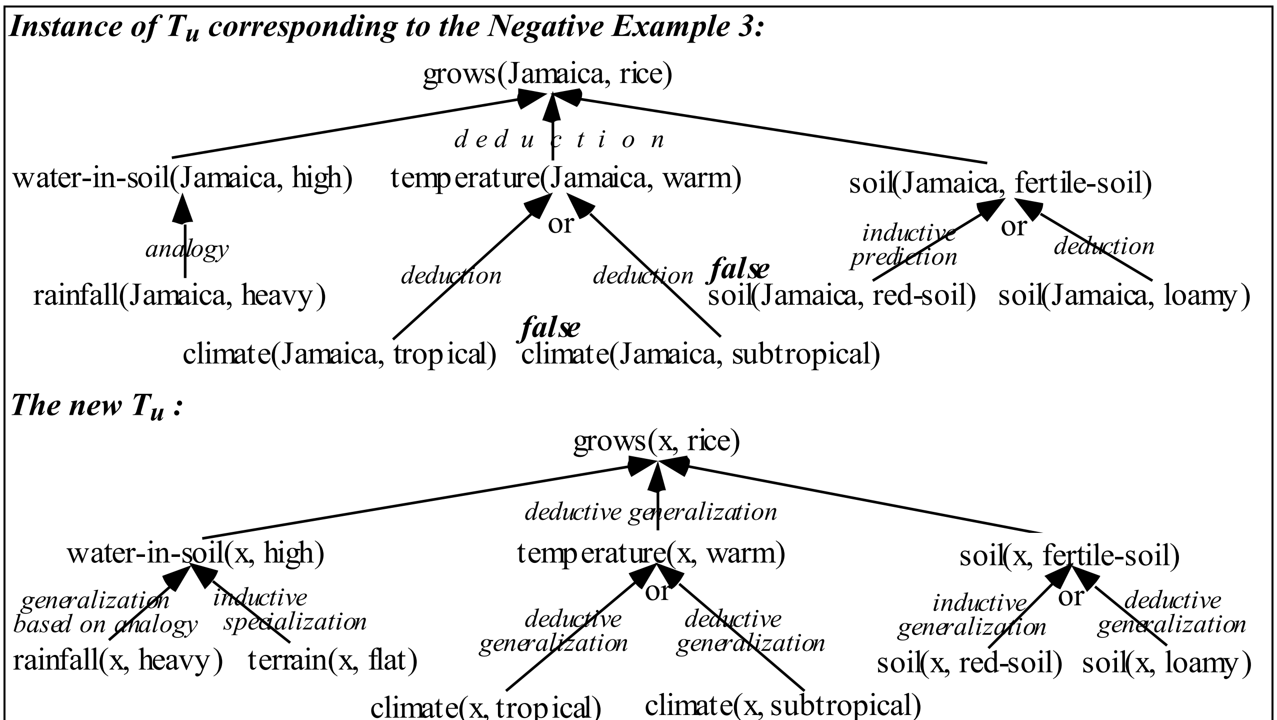


Figure 6: Specialization of T_u so that no longer to cover a justification of the Negative Example 3

Because the instance of T_u from the top of Figure 6 is an AND/OR tree, one should make sure to prove that enough of the leaf facts and implications are false. For instance, both "climate(Jamaica, subtropical)" and "soil(Jamaica, red-soil)" are false facts. However, because the nodes "temperature(x, warm)" and "soil(Jamaica, fertile-soil)" are OR nodes, the tree may still entail "grows(Jamaica, rice)". Therefore, one should show that an implication is false.

Deciding which is the false implication is a difficult problem. In the current version of MTL-JT, the implications hypothesized to be false are selected according to the following criteria:

- select the weakest implications (first abduction, then inductive prediction, then analogy, and lastly deduction);
- among the selected implications select those for which the corrections of the KB and of the general justification tree cause the minimum loss of coverage of the known instances;
- among the selected ones, select those for which the corrections produce a minimum increase in the complexity of the modified knowledge pieces;
- choose arbitrarily from the remaining hypotheses.

In the considered example, hypothesizing which is the false implication was simple because the justification tree from the top of Figure 6 contains one analogical implication and three deductive implications. Therefore, the analogical implication was considered to be the false one:

rainfall(Jamaica, heavy) -/-> water-in-soil(Jamaica, high) 18)

The corresponding implication from the current general justification tree is

rainfall(x, heavy) ∅ water-in-soil(x, high) 19)

which was derived from the determination

rainfall(x, y) --> water-in-soil(x, z) 20)

Consequently, the system will try to specialize the rule (20) so as no longer to cover (18), by taking into account the known instances of (19) and (20):

rainfall(Philippines, heavy) ∅ water-in-soil(Philippines, high)
rainfall(Thailand, heavy) ∅ water-in-soil(Thailand, high)
rainfall(Pakistan, heavy) ∅ water-in-soil(Pakistan, high)

together with the known properties of the involved objects (Jamaica, Philippines, Thailand, and Pakistan).

The inductive learner of MTL-JT will suggest, in this case, to specialize the determination (20) by adding the left-hand side predicate "terrain(x, flat)":

rainfall(x, y) & terrain(x, flat) --> water-in-soil(x, z)

The same specialization is applied to the implication (19). Thus, the general justification tree is specialized as indicated at the bottom of Figure 6.

6.2 Specialization of the KB

As a result of learning from the Negative Example 3, the system discovered a negative example of the plausible determination rule in Table 2, and specialized it by conjunctively adding the left hand side predicate "terrain(x, flat)", as shown in Table 4.

Because Negative Example 3 is the last input example, the system extracts from the tree T_u in Figure 6 the operational and abstract definitions indicated in Table 4.

7 BASIC CASES

An important feature of the presented method is that it behaves as a single-strategy learning method whenever the learning task of MTL-JT is specialized to the learning task of the single-strategy method. This feature is important because it shows that the MTL-JT method is a generalization of the integrated learning strategies which not only takes advantage of the complementarity of the integrated strategies (as has been shown in the previous sections), but also inherits the features of these strategies.

The next sections show that the MTL-JT method may behave as:

- explanation-based learning, learning by abduction, or learning by analogy, when the input consists of only one positive example;
- multiple-example explanation-based generalization, when the input consists of a sequence of positive examples;
- empirical or constructive inductive generalization when the input consists of a sequence of positive and negative examples.

7.1 Explanation-based learning

Let us suppose that, in addition to the rules in Table 2, the KB also contains the following deductive rules:

∀x, rainfall(x, heavy) □ water-in-soil(x, high)
∀x, soil(x, red-soil) □ soil(x, fertile-soil)

In such a case, the justification trees in Figure 3 and 4 become logical proofs, and the result of learning from Example 1 is an operational definition of "grows(x, rice)". Thus, the MTL-JT method reduces to explanation-based learning [DeJong & Mooney, 1986; Mitchell, Keller & Kedar-Cabelli, 1986].

7.2 Learning by abduction

Let us now suppose that the relationship between "rainfall" and "water-in-soil" is not a plausible determination, but a deductive implication

∀x, rainfall(x, heavy) □ water-in-soil(x, high)

and the KB does not contain examples of the predicate "soil". In this case, in order to build the justification tree of Example 1, the system needs only to create the explanatory hypothesis

soil(Thailand, red-soil) ∅ soil(Thailand, fertile-soil)

Therefore, the result of learning is the created explanatory hypothesis, and the MTL-JT method reduces to abductive learning.

7.3 Learning by analogy

Let us suppose that the KB contains only the following knowledge that is related to Example 1:

Facts:

rainfall(Philippines, heavy), water-in-soil(Philippines, high)

Determination:

rainfall(x, y) --> water-in-soil(x, z)

Then the system can only infer that "water-in-soil(Thailand, high)", by analogy with "water-in-soil(Philippines, high)", as shown in section 4.3. Thus, in this case, the MTL-JT method reduces to analogical learning.

7.4 Multiple-example explanation-based generalization

If the input of the system consists only of positive examples, that are deductively entailed by the KB, then the presented MTL-JT method behaves as the multiple example explanation-based generalization, or mEBG, that was developed, among others, by [Kedar-Cabelli, 1985; Pazzani, 1988; Hirsh, 1989].

7.5 Empirical and constructive inductive generalization

Let us assume that the KB does not contain the knowledge from Table 3, and the input consists of all the examples from Table 4. In this case, the input is new, neither confirming nor contradicting the KB. Therefore, each example is interpreted as representing a single implication that define a tree, as shown in the top part of Figure 7.

The MTL-JT method will compute the least general generalization of the trees corresponding to the positive examples, generalization that does not cover the tree corresponding to the negative example. The result of learning is therefore an empirical generalization that represents an operational definition of "grows(x, rice)" (see Figure 7). Thus, in this case, the MTL-JT method behaves as empirical inductive generalization.

If, however, the KB contains the deductive rules

$$\forall\ x,\ \text{climate}(x,\ \text{subtropical}) \rightarrow \text{temperature}(x,\ \text{warm})$$

$$\forall\ x,\ \text{climate}(x,\ \text{tropical}) \rightarrow \text{temperature}(x,\ \text{warm})$$

then the result of learning is the generalization from the bottom of Figure 7. This is a constructive generalization because it contains the descriptor "temperature" which does not belong to the description space of the examples. Therefore, in this case, the MTL-JT method behaves as constructive inductive generalization.

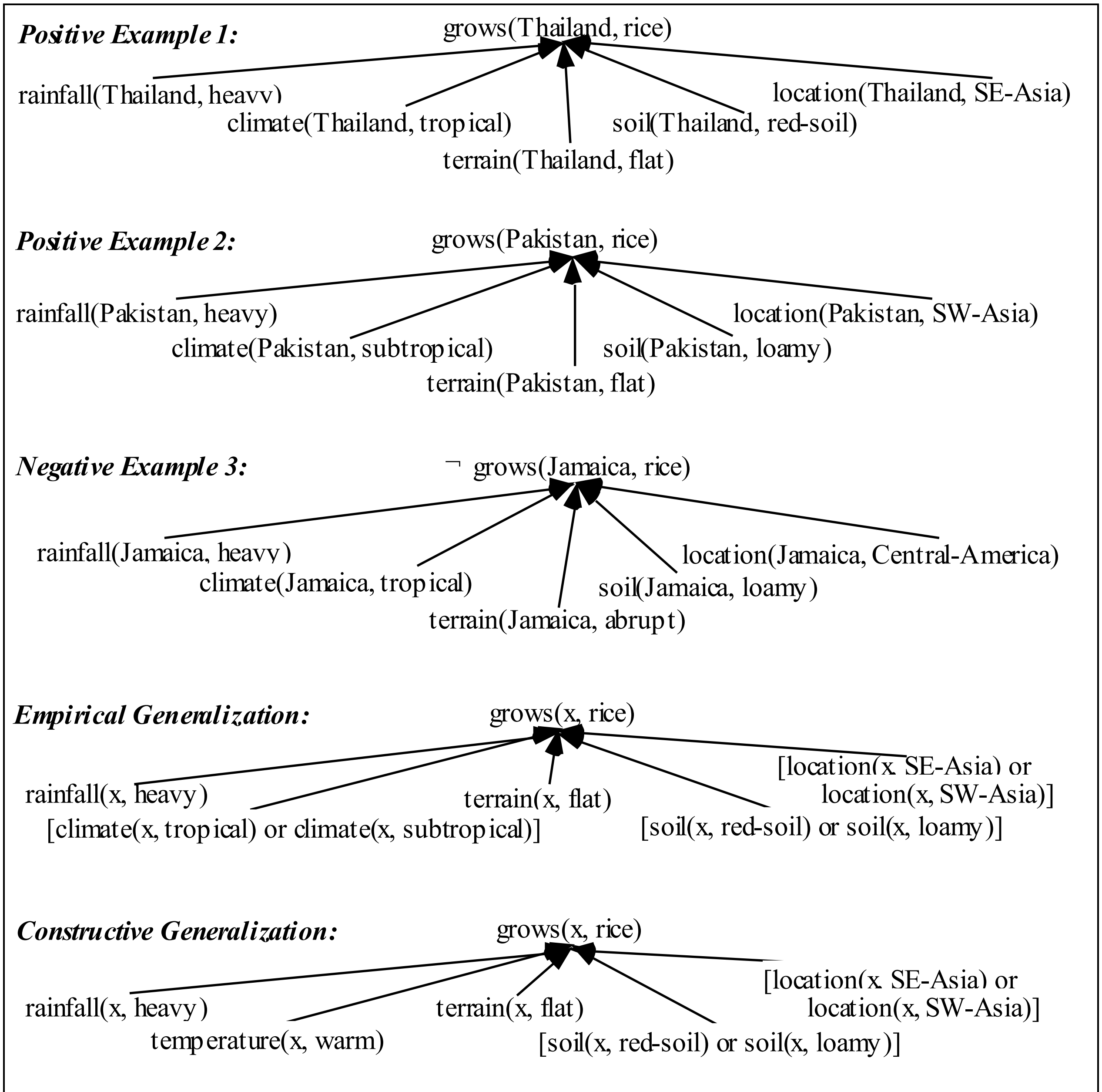


Figure 7: Empirical and constructive inductive generalizations of the input examples

8 DISCUSSION AND CONCLUSION

This chapter presented an inference-based framework for multistrategy learning and illustrated it with a specific multistrategy task-adaptive learning method called MTL-JT. The framework and the method evolved as a generalization and extension of Disciple [Tecuci, 1988; Tecuci and Kodratoff, 1990], an early interactive multistrategy learning system which learns from a specific problem solving episode by building a plausible explanation of it, overgeneralizing the explanation, and updating it as a result of an analysis of its instances.

This work is also closely related to the inferential theory of learning [Michalski, 1992] in that the learning strategies described in this theory could naturally be integrated into the presented framework.

Other researchers have also investigated the use of different types of inference in building explanations as, for instance, determinations [Mahadevan 1989; Widmer, 1989] or qualitative reasoning [DeJong, 1989; Widmer, 1992], which suggests that this idea is very appealing.

There are several dimensions of generality of the presented framework:

- it allows learning from different types of input as, for instance, one or several facts, examples, or problem solving episodes;

- it allows the KB to contain a variety of knowledge pieces that support different types of inference;

- it solves a general learning problem which includes theory revision and learning different types of concept definitions;

- it is extensible in that new types of inference, and therefore learning strategies, could naturally be integrated into it;

- it allows the use of different search strategies in the process of building plausible justification trees. The strategy employed in the MTL-JT method is a uniform-cost search of an AND-OR tree. However, one could employ any other search strategy (not only exhaustive but also heuristic).

- has certain similarities with human learning as, for instance, the building of the justification tree of an example by using the justification trees of the previous examples [Wisniewski and Medin, 1991], and the use of multiple lines of reasoning in the justification of a plausible inference step [Collins and Michalski, 1989].

The multistrategy task-adaptive learning method MTL-JT is only one way in which such a framework could be instantiated. It points, however, to the potential of the presented framework.

An important feature of MTL-JT is that it behaves as a single-strategy learning method whenever the learning task of MTL-JT is specialized to the learning task of the respective single-strategy method. This shows that MTL-JT is a generalization of the integrated learning strategies which not only takes advantage of their complementarity, but also inherits their features.

The presented framework and method have also revealed an new research direction in the theory of generalization by suggesting that with each type of inference may be associated a certain type of generalization. Consequently, one could perform not only deductive and inductive generalizations, but also generalizations based on analogy, on abduction, etc.

There are also several limitations and necessary developments of MTL-JT and of the general framework that need to be addressed by the future research.

One limitation has already been mentioned in section 6.1: during learning from a negative example the KB is not specialized enough so that to guarantee that it no longer entails any justification tree that would prove that the example is positive.

Also, MTL-JT does not deal with noisy input. This is an intrinsically difficult problem for a plausible reasoner that may itself make wrong inferences. However, because MTL-JT is a generalization of methods that could deal with noisy input, it inherits these capabilities. For instance, as in EBL, one may reject as noisy a negative example if one could build a deductive proof tree showing that the example is positive. Or, one may reject the negative example if the required specializations of the KB would determine a significant loss of coverage of instances of the knowledge pieces to be specialized.

The learning strategies integrated into MTL-JT (especially learning by analogy) are simple and should be replaced by more powerful ones.

Also new symbolic and even subsymbolic methods (as, for instance, reinforcement learning or neural network learning) should be integrated into MTL-JT. This will require, of course, elaboration of generalization techniques specific to each new strategy.

MTL-JT may also be extended so that to learn from other types of input (like general pieces of knowledge, or input already known).

Another important research direction regards the extension and the application of MTL-JT to knowledge acquisition from a human expert. In this case, the method would need to be extended with an important interactive component that would allow the system to ask different questions to the human expert, in order to decide on the best learning actions to take [Tecuci, 1988, 1992b, 1992c]. In general, the human expert would be asked to solve the problems that are intrinsically difficult for a learning system as, for instance, *the credit/blame assignment problem* (i.e. assigning credit or blame to the individual decisions that led to some overall result) or *the new terms problem* (i.e. extending the representation language with new terms when this cannot represent the concept or the rule to be learned).

ACKNOWLEDGMENTS

This research was done in the GMU Center for Artificial Intelligence. Research of the Center is supported in part by the National Science Foundation Grant No.IRI-9020266, in part by the Office of Naval Research grant No.N00014-91-J-1351, and in part by the Defense Advanced Research Projects Agency grant No.N00014-91-J-1854, administered by the Office of Naval Research.

References

Bergadano F., and Giordana A., "Guiding Induction with Domain Theories," in *Machine Learning: An Artificial Intelligence Approach, Vol. 3,* Y. Kodratoff and R.S. Michalski (Eds.), San Mateo, CA, Morgan Kaufmann, 1990.

Cohen W., "The Generality of Overgenerality," in *Machine Learning: Proceedings of the Eighth International Workshop*, L. Birnbaum and G. Collins (Eds.), Chicago, IL, Morgan Kaufmann, pp. 490-494, 1991.

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 the International Joint Conference on Artificial Intelligence*, Milan, Italy, Morgan Kaufmann, pp. 274-276, 1987.

Davies T.R. and Russell S.J., "A Logical Approach to Reasoning by Analogy," *Proceedings of the International Joint Conference on Artificial Intelligence*, Milan, Italy, Morgan Kaufmann, pp. 264-270, 1987.

DeJong, G. and Mooney, R., "Explanation-Based Learning: An Alternative View," *Machine Learning*, Vol. 1, pp. 145-176, 1986.

Flann, N., and Dietterich, T., "A Study of Explanation-based Methods for Inductive Learning," *Machine Learning*, Vol. 4, pp. 187-266, 1989.

Genest, J., Matwin, S., and Plante, B., "Explanation-based Learning with Incomplete Theories: A Three-step Approach," in *Machine Learning: Proceedings of the Eighth International Workshop,* B. Porter and R. Mooney (Eds.), Texas, Morgan Kaufmann, 1990.

Gordon, D.F., "An Enhancer for Reactive Plans," in *Machine Learning: Proceedings of the Eighth International Workshop,* L. Birnbaum, and G. Collins (Eds.), Chicago, IL, Morgan Kaufmann, pp. 505-508, 1991

Hirsh, H., "Incremental Version-space Merging: A General Framework for Concept Learning," *Doctoral dissertation,* Stanford University, 1989.

Josephson J., "Abduction: Conceptual Analysis of a Fundamental Pattern of Inference," *Technical Research Report 91-JJ,* Laboratory for Artificial Intelligence Research, The Ohio State University, 1991.

Lebowitz, M., "Integrated Learning: Controlling Explanation," *Cognitive Science,* Vol. 10, pp. 219-240, 1986.

Mahadevan S., "Using Determinations in Explanation-based Learning: A Solution to Incomplete Theory Problem," in *Proceedings of the Sixth International Workshop on Machine Learning,* A. Segre (Ed.), Ithaca, NY, Morgan Kaufman, 1989.

Michalski, R.S., "Toward a Unified Theory of Learning: Multistrategy Task-Adaptive Learning," *Reports of Machine Learning and Inference Laboratory*, MLI 90-1, Center for Artificial Intelligence, George Mason University, 1990.

Michalski, R.S., "Inferential Learning Theory: Developing Theoretical Foundations for Multistrategy Learning," in *Machine Learning: An Multistrategy Approach, Vol. 4,* R.S. Michalski and G. Tecuci (Eds.), San Mateo, CA, Morgan Kaufmann, 1992.

Minton, S., Carbonell, J.G., "Strategies for Learning Search Control Rules: An Explanation-based Approach," *Proceedings of the International Joint Conference on Artificial Intelligence,* Milan, Italy, Morgan Kaufmann, pp. 228-235, 1987.

Mitchell, T.M., Keller, T., and Kedar-Cabelli, S., "Explanation-Based Generalization: A Unifying View," *Machine Learning,* Vol. 1, pp. 47-80, 1986.

Mooney R., Bennet S., "A Domain Independent Explanation Based Generalizer," *Proceedings of the Fifth National Conference on Artificial Intelligence,* Philadelphia, PA, Morgan Kaufmann, pp. 551-555, 1986.

Mooney R. and Ourston D., "A Multistrategy Approach to Theory Refinement," in *Machine Learning: An Multistrategy Approach, Vol. 4,* R.S. Michalski and G. Tecuci (Eds.), San Mateo, CA, Morgan Kaufmann, 1992.

Morik K., "Balanced Cooperative Modeling," in *Machine Learning: An Multistrategy Approach, Vol. 4,* R.S. Michalski and G. Tecuci (Eds.), San Mateo, CA, Morgan Kaufmann, 1992.

Nilsson, N., *Problem Solving Methods in Artificial Intelligence,* McGraw-Hill, 1971.

Pazzani, M.J., "Integrating Explanation-Based and Empirical Learning Methods in OCCAM," *Proceedings of the Third European Working Session on Learning,* Glasgow, Scotland, pp. 147-166, 1988.

De Raedt L., Interactive Concept Learning. *Ph.D. Thesis*, Catholic University of Leuven, 1991.

Ram, A. and Cox, M., "Introspective Reasoning using Meta-explanations for Multistrategy Learning," in *Machine Learning: An Multistrategy Approach, Vol. 4,* R.S. Michalski and G. Tecuci (Eds.), San Mateo, CA, Morgan Kaufmann, 1992.

Reich, Y., "Macro and Micro Perspectives of Multistrategy Learning," in *Machine Learning: An Multistrategy Approach, Vol. 4,* R.S. Michalski and G. Tecuci (Eds.), San Mateo, CA, Morgan Kaufmann, 1992.

Saitta L. and Botta M., "Multistrategy Learning and Theory Revision," *Machine Learning Journal*, special issue on Multistrategy Learning, 1992.

Shavlik J.W., and Towell G.G., "An Approach to Combining Explanation-based and Neural Learning Algorithms," in *Readings in Machine Learning,* J. W. Shavlik and T. Dietterich (Eds.), Morgan Kaufmann, 1990.

Tecuci G., DISCIPLE: A Theory, Methodology, and System for Learning Expert Knowledge. *Ph.D. Thesis*, University of Paris-South, 1988.

Tecuci G., "Plausible Justification Trees: a Framework for the Deep and Dynamic Integration of Learning Strategies," *Machine Learning Journal*, special issue on Multistrategy Learning, 1992a.

Tecuci G., "Automating Knowledge Acquisition as Extending, Updating, and Improving a Knowledge Base," *IEEE Transactions on Systems, Man and Cybernetics,* Vol. 22, No. 6, 1992b.

Tecuci G., "Cooperation in Knowledge Base Refinement," in *Machine Learning: Proceedings of the Ninth International Conference (ML92),* D. Sleeman and P. Edwards (Eds.), Morgan Kaufmann, San Mateo, CA., 1992c.

Tecuci G. and Kodratoff Y., "Apprenticeship Learning in Imperfect Theory Domains," in *Machine Learning: An Artificial Intelligence Approach, Vol. 3,* Y. Kodratoff and R.S. Michalski (Eds.), San Mateo, CA, Morgan Kaufmann, 1990.

Tecuci G. and Michalski R.S., "A Method for Multistrategy Task-adaptive Learning Based on Plausible Justifications," in *Machine Learning: Proceedings of the Eighth International Workshop,* L. Birnbaum, and G. Collins (Eds.), Chicago, IL, Morgan Kaufmann, pp. 549-553, 1991a.

Tecuci G. and Michalski R.S., "Input Understanding as a Basis for Multistrategy Task-adaptive Learning," in *Proceedings of the International Symposium on Methodologies for Intelligent Systems,* Charlotte, NC, Springer-Verlag, pp. 419-428, 1991b.

Whitehall, B.L., "Knowledge-based Learning: Integration of Deductive and Inductive Leaning for Knowledge Base Completion," *PhD Thesis*, University of Illinois at Champaign-Urbana, 1990.

Widmer, G., "A Tight Integration of Deductive and Inductive Learning," *Proceedings of the Sixth International Workshop on Machine Learnin,* A. Segre (Ed.), Ithaca, NY, Morgan Kaufmann, 1989.

Widmer, G., "Learning with a Qualitative Domain Theory by Means of Plausible Explanations," in *Machine Learning: An Multistrategy Approach, Vol. 4,* R.S. Michalski and G. Tecuci (Eds.), San Mateo, CA, Morgan Kaufmann, 1992.

Wilkins, D.C., "Knowledge Base Refinement as Improving an Incorrect and Incomplete Domain Theory," in *Machine Learning: An Artificial Intelligence Approach, Vol. 3,* Y. Kodratoff and R.S. Michalski (Eds.), San Mateo, CA, Morgan Kaufmann, 1990.

Wisniewski E.J., and Medin D.J., "Is it a Pocket or a Purse ? Tightly Coupled Theory and Data Driven Learning," in *Machine Learning: Proceedings of the Eighth International Workshop*, L. Birnbaum and G. Collins (Eds.), Chicago, IL, Morgan Kaufmann, pp. 564-568, 1991.