* For the reference see the last page

LEARNING RULES FROM EXAMPLES IN RULE-BASED SYSTEMS VIA AN INTEGER PROGRAMMING APPROACH

Evangelos Triantaphyllou and Soundar R.T. Kumara
Department of Industrial and Management Systems Engineering
The Pennsylvania State University

Abstract. An increasingly interesting issue in Rule-Based System (RBS) development is the problem of inferring rules from examples. As RBSs attempt to solve larger and more complex problems, automatic ways for determining the knowledge bases of these systems become critical in Knowledge Engineering. The present paper proposes the use of an Integer Programming approach in deriving rule structures from a set of examples. This Integer Programming approach provides a flexible and powerful way for utilizing information that very often is difficult to handle, mainly due to the complexity of the problem.

Key Words. Rule-Based Systems, machine learning, learning from examples, Integer Programming.

1.0 Introduction

In this section we deal with some of the fundamentals of Rule-Based Systems (RBSs). Edward Feigenbaum in [Harmon and King, 1985] has defined Expert Systems or Rule Based Systems as: *"... an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution "*. From the above definition it follows that any RBS is associated with a knowledge base and of inference procedures. The inference procedures guide the usage of the knowledge base in order to solve a problem under question. In RBSs the knowledge base is organized in terms of procedural rules and declarative facts. A rule is divided in two parts: the antecedent and the consequent.

The rules used in a knowledge base can be unreliable and changeable or reliable and static. A system that belongs in the first case is MYCIN. Such systems require considerable attention on such issues as: probabilities, fuzzy logic, uncertainty etc. Usually there is no single widely accepted way of handling data of this nature [Stefik, et al., 1982]. The systems that we will deal with are the ones that use reliable and static knowledge.

While the inference process takes place it is possible for the system to require additional information. Usual sources of additional information are data bases (for example: files with numbers), human experts, or the working memory. The working memory is used to store findings that are established during a consultation session. The working memory might expand or shrink during a consultation session. To summarize, an expert system has the following parts:

1) The knowledge base

2) The inference procedures

3) Additional sources of information (data bases, humans, sensors, etc.)

4) A dynamically changing working memory

We deal with deterministic RBSs (reliable and static) for three critical reasons:

1) These are simpler than systems of the first kind (ie: the ones that use unreliable and changeable knowledge bases). We do not need to consider probabilities or fuzzy logic. The knowledge is much better structured.

2) Even if we needed to study systems of the first kind, we still need to examine first the systems with reliable and static data (since they are of simpler nature).

3) Systems of the second kind are more easily formulated by means of automata theory or Integer Programming formulations. That is, they are more promising for study.

A typical rule in the RBSs under consideration has the following structure:

$$\text{IF } (P_1 \text{ and } P_2 \text{ and } \ldots P_n) \text{ THEN } (P_{i1}, P_{i2}, \ldots P_{in}) \tag{1.1}$$

That is, if premises $P_1, P_2, \ldots, P_n$ are true then, premises P_{i1}, $P_{i2}, \ldots P_{in}$ claimed to be true as well.

As an example, rules in the VM system reported by Fagan [Fagan, 1980] follow the above structure (1.1). For instance, the VM rule:

IF the current context is 'Assist' AND
respiration rate has been stable for 20 minutes AND
I/E ratio has been stable for 20 minutes,
THEN the patient is on 'CMV' (Controlled Mandatory Ventilation).

follows the structure depicted in (1.1). Here the premises are:

P_1 = " the current context is 'Assist' "

P_2 = " respiration rate has been stable for 20 minutes "

P_3 = " I/E ratio has been stable for 20 minutes "

P_{i1} = " then the patient is on 'CMV'

In the following section the RBSs that are under consideration in this study are described in a formal manner. This kind of treatment is necessary for a number of reasons. First, the structure of these RBSs is established in a clear and rigorous fashion. Secondly, some elementary properties of the RBSs are revealed. In this way, it is possible to understand some of the combinatorial difficulties that emerge in many RBS related problems in a better manner.

2.0 Some Characteristics of RBSs

Using the previous notation and structure of rules, it can be derived easily that a typical rule in the RBSs under consideration can be represented as an ordered pair of two sets. More specifically, the rule given in the previous section as (1.1), becomes:

$$(\{P_1, P_2, P_3, \ldots P_n\}, \quad \{P_{i1}, P_{i2}, P_{i3}, \ldots P_{in}\}) \tag{2.1}$$

The consequent part of the rule shown in (2.1) involves <in> indicator variables (or <in> premises). For reasons of ease of modeling it is assumed that the consequent part of any rule involves **only one** indicator variable. This assumption is **without loss of generality** because a rule given as in (2.1) can equivalently be represented by the set of <in> rules given as (2.2):

$$(\{P_1, P_2, P_3, \ldots P_n\}, \quad \{P_{i1}\}) \tag{2.2}$$

$$(\{P_1, P_2, P_3, \ldots P_n\}, \quad \{P_{i2}\})$$

$$(\{P_1, P_2, P_3, \ldots P_n\}, \quad \{P_{i3}\})$$

.

.

.

$$(\{P_1, P_2, P_3, \ldots P_n\}, \quad \{P_{in}\})$$

Previously it was assumed that the premises in the antecedent parts are connected only with ANDs. This is without loss of generality because if ORs are present they can be eliminated by substituting them with equivalent expressions that have only ANDs. For example, the rule:

$$(\{\{P_1 \text{ AND } P_2\} \text{ OR } \{P_3 \text{ AND } P_4\}\} \quad \{P_5\})$$

is equivalent to the following set of rules:

$$(\{P_1 \text{ AND } P_2\} \quad \{P_5\})$$

$$(\{P_3 \text{ AND } P_4\} \quad \{P_5\})$$

From the previous discussion it follows that the knowledge base of any RBS can be represented by a set of ordered pairs that have the form:

$$(\{P_1, P_2, P_3, \ldots P_n\}, \quad \{P_k\}) \tag{2.3}$$

If it is assumed that a RBS has a knowledge base with M rules with n_i premises in the antecedent part of the i^{th} rule (for $i=1,2,3,\ldots,M$) then, the knowledge base of that RBS can be represented as:

$$(\{P_{11}, P_{12}, P_{13}, \ldots P_{1n_1}\}, \quad \{P_{k1}\}) \tag{2.4}$$

$$(\{P_{21}, P_{22}, P_{23}, \ldots P_{2n_2}\}, \quad \{P_{k2}\})$$

$$(\{P_{31}, P_{32}, P_{33}, \ldots P_{3n_3}\}, \quad \{P_{k3}\})$$

$(\{P_{M1}, P_{M2}, P_{M3},\dots P_{Mn_M}\}, \quad \{P_{kM}\})$

Suppose that a RBS has N premises. From the N premises suppose that N_1, $(0 < N_1 < N)$ are in the antecedent parts of the rules. These N_1 premises are called **input premises.** Let N_2 to be the number of premises in the consequent parts. In reality some of the premises that appear in the consequent parts might also appear in the antecedent parts of some rules. However, it is assumed that these premises are substituted with premises that appear only in the antecedent parts. This is without loss of generality. In this way rules that might cause infinite loops are avoided. For example, the following rules create such a loop:

RULE 1: $(\{P_1, P_2, P_3\}, \{P_4\})$

RULE 2: $(\{P_4, P_5\}, \{P_1\})$

However, if it is assumed that consequent premises do not appear in antecedent parts, the previous two rules become:

RULE 1′: $(\{P_1, P_2, P_3\}, \{P_4\})$

RULE 2′: $(\{P_1, P_2, P_3, P_5\}, \{P_1\})$

That is, the second rule has the consequent premise in the antecedent part. This might make the use of the second rule impossible since P_1 may never be proved. In order to avoid this problem, it is assumed that the consequent premises do not appear in the antecedent part in any rule. Therefore, these N_2 consequent premises will be called **output premises** (since their value is the output of the system). From the above discussion, relation (2.5) follows:

$$N_1 + N_2 = N \tag{2.5}$$

The above concepts are illustrated through the following example:

Example 1.

The following rules deal with the problem (adopted from [Harmon anf King, 1985]) about getting to a theater on time. The knowledge base is depicted in Table I. The premises involved in these rules can be encoded as shown in Table II.

From the 16 premises shown in Table II the first 9 are input premises while the remaining 7 are output premises. That is, in this example: $N = 16$, $N_1 = 9$, and $N_2 = 7$. From the definition of the above premises the rules in the knowledge base can be written in terms of ordered pairs as follows:

From Rule #1:

$(\{P_1\} \quad \{P_{10}\})$

$(\{P_5, P_3\} \quad \{P_{10}\})$

From Rule #2:

$(\{ P_2, P_3 \} \quad \{ P_{10} \})$

$(\{ P_2, P_3 \} \quad \{ P_{16} \})$

From Rule #3:

$(\{ P_2, P_4 \} \quad \{ P_{10} \})$

From Rule #4 (and Rule #1):

$(\{ P_1, P_5 \} \quad \{ P_{12} \})$

From Rule #5 (and Rule #1):

$(\{ P_1, P_6 \} \quad \{ P_{13} \})$

From Rule #6:

$(\{ P_9 \} \quad \{ P_{10} \})$

From Rule #7 (and Rule #6):

$(\{ P_9, P_7 \} \quad \{ P_{14} \})$

From Rule #8 (and Rule #6):

$(\{ P_9, P_8 \} \quad \{ P_{15} \})$

That is, the knowledge base given in Table I, by using the conventions established in Table II, can be represented by the ordered pairs depicted in Table III.

Table I. Rules about getting to the theater

Rule	IF:	THEN:
1	Distance is greater than 5 miles OR (Location is Downtown AND Time is less than 15 minutes)	Means is drive
2	Distance is greater than 1 mile AND Time is less than 15 minutes	Means is drive AND Advise is hurry up
3	Distance is greater than 1 mile AND Time is greater than 15 minutes	Means is drive
4	Means is drive AND Location is downtown	Action is take a cab
5	Means is drive AND Location is not downtown	Action is drive your car
6	Distance is less than 1 mile	Means is walk
7	Means is walk AND Weather is bad	Action is take a coat and walk
8	Means is walk AND Weather is good	Action is walk

Table II. Set of premises

Code	Premise	Code	Premise
P_1	Distance is greater than 5 miles	P_9	Distance is less than 1 mile
P_2	Distance is greater than 1 mile	P_{10}	Means is drive
P_3	Time is less than 15 minutes	P_{11}	Means is walk
P_4	Time is greater than 15 minutes	P_{12}	Action is take a cab
P_5	Location is downtown	P_{13}	Action is drive your car
P_6	Location is not downtown	P_{14}	Action is take a coat and walk
P_7	Weather is bad	P_{15}	Action is walk
P_8	Weather is good	P_{16}	Advise is hurry up

Table III. Set of ordered pairs

Derived from:	Ordered pair	
Rule #1:	$(\{P_1\}$	$\{P_{10}\})$
	$(\{P_5, P_3\}$	$\{P_{10}\})$
Rule #2:	$(\{P_2, P_3\}$	$\{P_{10}\})$
	$(\{P_2, P_3\}$	$\{P_{16}\})$
Rule #3:	$(\{P_2, P_4\}$	$\{P_{10}\})$
Rule #4:	$(\{P_1, P_5\}$	$\{P_{12}\})$
Rule #5:	$(\{P_1, P_6\}$	$\{P_{13}\})$
Rule #6:	$(\{P_9\}$	$\{P_{10}\})$
Rule #7:	$(\{P_9, P_7\}$	$\{P_{14}\})$
Rule #8:	$(\{P_9, P_8\}$	$\{P_{15}\})$

3.0 Description of the RBS Inference Problem

The current paper primarily deals with the RBS inference problem. That is, to infer a RBS from a set of example consultations. An example consultation is a set of input premises that have to be true in order for certain output premises to be true, or false.

For example, suppose that a system under question involves the following five premises as input ones:

P_1, P_2, P_3, P_4, P_5

Suppose also that the following three are the output premises:

P_6, P_7, P_8

Then, a example consultation might be:

$(\{P_1, P_3\} \ \{P_6, P_8\})$

This example can be interpreted that if premises P_1 and P_3 are true, then the output premises P_6, P_8 should also be true, while the output premise P_7 must be false. As with the case of the rule format, example consultations are assumed to be ordered pairs of two sets. The first set contains the input premises that are true in the current example. The second set simply contains the output premise that is true when the input premises are true. If the second set is empty, then no output premise becomes true in that example. This is without loss of generality. For example: $(\{P_1, P_3\} \ \{P_6, P_8\})$ becomes:

$(\{P_1, P_3\} \ \{P_6\})$

$(\{P_1, P_3\} \ \{P_8\})$

The inference problem then is to utilize examples of the above form and to determine a RBS that can derive these examples. As a second demonstration, suppose that a system under question has five input premises (ie: P_1, P_2, P_3, P_4, P_5) and only one output premise (ie: P_6). Suppose also that the following example consultations (interpretable as previously) are the only available data:

$(\{P_1, P_2, P_3\} \ \{P_6\})$

$(\{P_1, P_2, P_4\} \ \{P_6\})$

$(\{P_2, P_3\} \ \{P_6\})$

$(\{P_2, P_4\} \ \{ \ \})$

The last example simply indicates that when premises P_2 and P_4 are true then, the output premise P_6 cannot be true.

A RBS that can satisfy the above data might have **three rules** with the following trivial structure:

$$IF\ (P_1\ AND\ P_2\ AND\ P_3)\quad THEN\ (P_6)$$
$$IF\ (P_1\ AND\ P_2\ AND\ P_4)\quad THEN\ (P_6)$$
$$IF\ (P_2\ AND\ P_3)\quad THEN\ (P_6)$$

Apparently, these three rules can derive all the above examples. However, it is desirable to design a method that determines a RBS, with the above requirements, that has the **minimum** number of rules. For the current example, such a RBS might be:

$$IF\ (P_1\ AND\ P_2)\quad THEN\ (P_6)$$
$$IF\ (P_2\ AND\ P_3)\quad THEN\ (P_6)$$

The importance of the RBS inference problem was recognized from the first steps of the AI development. McCarthy expressed the ultimate objective of research on AI by stating that *"Our ultimate objective is to make programs that learn from their experience as effectively as humans do"* [McCarthy, 1958]. This problem is also known as learning from examples. An early treatment of this problem is due to Fu [Fu, 1970]. In this approach the unknown system is considered as finite-state automaton. A comparative review of selected methods for learning from examples, on a RBS context, can be found in [Dietterich and Michalski, 1983]. More recent developments are included in [Michalski, 1986], [Mozetic, 1986], [Fisher, 1987], [Tallis, 1988], [Gross, 1988], [Helft, 1988], [Kadie, 1988].

Research on learning from examples does not provide a unified formalization of RBSs. In order for a methodology on this subject to have pervasive power, a formalization of the majority of RBSs is needed. In the literature the RBS inference problem has not been examined from a **minimum number of rules** point of view. This is one of the reasons why Mathematical Programming, and especially Integer Programming, is a reasonable approach in treating machine learning issues. Furthermore, the highly successful application of Mathematical Programming techniques on the logical inference problem (see, for example, [Williams, 1986], [Jeroslow, 1988], [Hooker, 1988a, 1988b]) provide a strong motivation for examining learning from examples from an Integer Programming point of view.

4.0 Formalization of RBSs

4.1 Some Preliminary Definitions

It is assumed that initially there are N premises involved in the RBS that is to be determined. These premises are denoted as: $P_1,\ P_2,\ P_3,\ \ldots\ P_N$. It is also assumed, without loss of generality, that the first N_1 $(1 \le N_1 \le N)$ premises are **input premises**, that is, their value (true or false) is given as input to the system. Similarly, the remaining N_2 $(N_2 = N - N_1)$ premises are **output premises**, that is, their value (true or false) is given as the output of the system. After establishing the concepts of input and output premises, the following definitions are introduced:

Definition 4.1:

Define S_{N_1} to be the set of **all** the possible input conditions. Since the number of input premises is N_1, it follows that there are $(2^{N_1} - 1)$ different input conditions (since the empty set is not considered).

Definition 4.2:

Define S_i^T ($i = N_1+1, N_1+2, N_1+3, \dots N$) to be the set of **all** input conditions that result in the premise P_i to be true.

Definition 4.3:

Similarly, define S^F_i ($i = N_1+1, N_1+2, N_1+3, \dots N$) to be the set of all input conditions that result in the premise P_i to be false.

Definition 4.4:

Define E_m^T ($N_1 < m \leq N$) as the set of all the members $a_j \in S_{N_1}$ that are already **known** to satisfy the relation: $a_j \in S_m^T$.

Definition 4.5:

Define E_m^F ($N_1 < m \leq N$) as the set of all the members $a_j \in S_{N_1}$ that are already **known** to satisfy the relation: $a_j \in S_m^F$.

4.2 The Definition of the $\phi(S)$, $t(a)$, and $f(a)$ Sets

Let $a \in S_{N_1}$ then, a can be written as:

$$a = [\, b_1\, b_2\, b_3 \dots b_{N_1} \,]$$

In other words, a can be represented by a sequence of N_1 characters that are either zero or one according to the following rule:

$$b_i = \begin{cases} 1 & \textit{if input premise } P_i \textit{ is true} \\ 0, & \textit{otherwise} \end{cases} \qquad (\textit{for any } i = 1, 2, 3, \dots N_1)$$

That is, the members of the S_{N_1}, S_i^T and S_i^F sets are strings of size N_1 with characters from the alphabet: $\{0, 1\}$.

Definition 4.6:

Define the set $\phi(\{P_{n1}, P_{n2}, P_{n3}, \dots P_{nL}\}$ to be the set with members from the set $\{P_{n1}, P_{n2}, P_{n3}, \dots P_{nL}\}$ plus all its **non-empty** subsets.

For example: $\phi(\{P_1, P_2, P_3\}) =$

$$\{\{P_1, P_2, P_3\}, \{P_1, P_2\}, \{P_1, P_3\}, \{P_2, P_3\}, \{P_1\}, \{P_2\}, \{P_3\}\}$$

For reasons of notational simplicity the expression: $\phi(\{P_{n1}, P_{n2}, P_{n3}, \dots P_{nL}\}$ will be denoted as: $\phi(\{n_1, n_2, n_3, \dots n_L\})$

Let $a \in S_{N_1}$, where: $a = [\, b_1\, b_2\, b_3 \dots b_{N_1} \,]$

Suppose also that the following relation (4.1) is true:

$$\sum_{j=1}^{N_1} b_j = \sum_{j=1}^{K} b_{i_j} = i_K \tag{4.1}$$

That is, the $b_{i_j}'s$ are the only characters in string a to be equal to 1 or equivalently, the $P_{i_1}, P_{i_2}, P_{i_3}, \dots P_{i_K}$ are the only premises to be true among all the input premises. The following definition formalizes the previous concept:

Definition 4.7:

Let $a \in S_{N_1}$. Then, $t(a)$ is defined to be the set of the input premises that are stated as **true** in string a.

Definition 4.8:

Similarly, let $a \in S_{N_1}$. Then, $f(a)$ is defined to be the set of the input premises that are stated as **false** in string a.

The following example illustrates the above issues:

Example 2:

Let $N = 5$ and $N_1 = 4$. Suppose also that:

$$a_1 = (1\,1\,1\,0)$$

$$a_2 = (0\,0\,0\,1)$$

$$a_3 = (1\,1\,1\,1)$$

Then, the following relations are derived by using the appropriate definitions (the P's are dropped from the notation for easier representation. For example, {1, 2, 3} stands for: $\{P_1, P_2, P_3\}$):

$t(a_1) = \{1, 2, 3\}$

$t(a_2) = \{4\}$

$t(a_3) = \{1, 2, 3, 4\}$

$f(a_1) = \{4\}$

$f(a_2) = \{1, 2, 3\}$

$f(a_3) = \{\ \}$

and consequently:

$\phi(t(a_1)) = \{\{1, 2, 3\}, \{1, 2\}, \{2, 3\}, \{1, 3\}, \{1\}, \{2\}, \{3\}\}$

$\phi(t(a_2)) = \{\{4\}\}$

$\phi(t(a_3)) = \{\{1, 2, 3, 4\}, \{1, 2, 3\}, \{1, 2\}, \{2, 3\}, \{1, 3\}, \{1, 2, 4\}, \{1, 4\}, \{2, 4\}, \{3, 4\}, \{2, 3, 4\}, \{1, 3, 4\}, \{1\}, \{2\}, \{3\}, \{4\}\}$

4.3 Inference Properties of Members of the E_m^T and E_m^F Sets

Suppose that it is given that:

$$a_1 = (1\,1\,1\,0) \in E_5^T$$

The previous relation indicates that when premises P_1, P_2, and P_3 are **true** then the output premise P_5 should also be true. That is, there is a rule that has in its antecedent part a member of the set:

$$\{\{1,2,3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1\}, \{2\}, \{3\}\} = \phi(t(a_1))$$

and has $\{P_5\}$ as a consequent. Similarly, suppose that:

$$a_2 = (0\,1\,1\,0) \in E_5^F$$

In this case the above relation indicates that any rule that has as antecedent a member from the set:

$$\{\{2,3\}, \{2\}, \{3\}\} = \phi(t(a_2))$$

cannot have the consequent: $\{P_5\}$.

The previous two observations can be generalized as follows:

Statement 4.1:

Let $a \in E_m^T$ $(N_1 < m \leq N)$. Then, a was **derived** by a rule that has in its antecedent part a member of the set: $\phi(t(a))$ and as consequent: $\{P_m\}$.

Statement 4.2:

Let $a \in E_m^F$ $(N_1 < m \leq N)$. Then, a **cannot be derived** by a rule that has in its antecedent part a member of the set: $\phi(t(a))$ and as consequent: $\{P_m\}$.

The following example elaborates the previous two statements.

Example 3:

As in example 2, let $N = 5$ and $N_1 = 4$. Suppose also:

$$a_1 = (1\,1\,1\,0) \in S_5^T$$

$$a_2 = (0\,0\,0\,1) \in S_5^T$$

$$a_3 = (1\,1\,1\,1) \in S_5^T$$

and $$a_4 = (0\,1\,1\,0) \in S_5^F$$

Then, from Statement 4.1, the string a_1 was derived by a rule that has in its antecedent part a member of the set $\phi((t(a_1))$ and as a consequent $\{P_5\}$. Similarly, a_2 and a_3 were derived from rules that have antecedent parts from the sets $\phi((t(a_2))$ and $\phi((t(a_3))$, respectively. Statement 4.2 and the fact that $a_4 \in E_5^F$ indicate that rules that have consequent part: $\{P_5\}$ **cannot** have as antecedents a member of the set $\phi((t(a_4)))$. From the previous discussion it follows that:

The antecedent part of the rule that derived a_1 is a member of f_1

The antecedent part of the rule that derived a_2 is a member of f_2

The antecedent part of the rule that derived a_3 is a member of f_3

where:

$$f_1 = \phi(((t(a_1))) - \phi(((t(a_4))) = \{\{1,2,3\}, \{1,2\}, \{1,3\}, \{1\}\}$$

$$f_2 = \phi(((t(a_2))) - \phi(((t(a_4))) = \{\{4\}\}$$

$$f_3 = \phi(((t(a_3))) - \phi(((t(a_4))) = \{\{1,2,3,4\}, \{1,2,3\}, \{1,2\}, \{1,3\}, \{1,2,4\}, \{1,4\}, \{2,4\}, \{3,4\}, \{2,3,4\}, \{1,3,4\}, \{1\}, \{4\}\}$$

It can be also seen that:

$$f_1 \cap f_2 \cap f_3 = \phi$$

From the above relation it follows that given:

$E_5^T = \{a_1, a_2, a_3\}$ and

$E_5^F = \{a_4\}$

it is **impossible** for a_1, a_2, a_3 to be derived by a **single** rule. However, the previous data suggest that it is possible that a_2, a_3 were derived by the same rule that has antecedent: { 4 } and consequent: { 5 }. This can be seen easily from the following Venn diagrams of the sets f_1, f_2, f_3 (Figure 1).

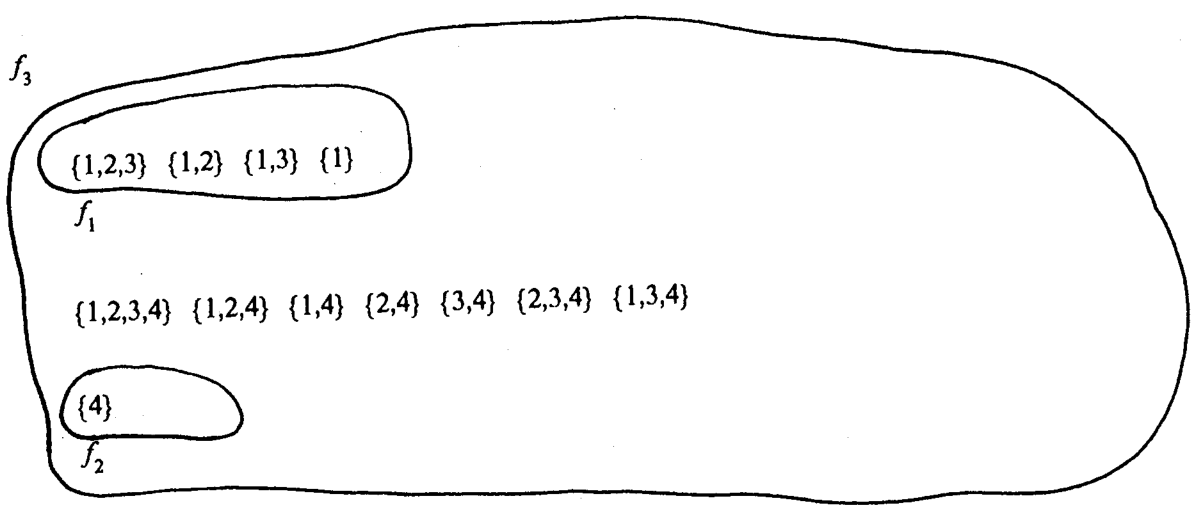

Figure 1. Venn diagrams

From Figure 1 it can be seen that all the RBSs that have one rule that derives a_2, a_3 and a second rule that derives a_1 must have the first rule with antecedent { 4 } and the second rule with antecedent a member of the set { {1,2,}, {1,2}, {1,3}, {1} }. That is, there are: 1x4 = 4 such systems. Similarly, it is possible to derive a_1, a_3 from one rule and a_2 from a second rule. In this case the first rule has as antecedent a member from the set { {1,2,}, {1,2}, {1,3}, {1} } and the second rule has antecedent { 4 }. Apparently it is impossible to derive a_1 and a_2 from the same rule (since $f_1 \bigcap f_2 = \phi$).

From the above considerations and under the requirement of the **minimum** number of rules it follows that given the data of example 3 there are **4 candidate** RBSs that can result a_1, a_2, $a_3 \in E_5^T$ and $a_4 \in E_5^F$.

From the way RBS are modeled in the present treatment it is a good idea to consider **one output premise at a time.** This is true because if we know the rules that correspond to each output premise then, the rules in the final RBS can be found by simply taking together all the above sets of rules. For this reason, from now on, it is assumed that there is **only one output premise** in each rule. This consideration indicates that the RBS inference problem is reduced to the problem of inferring a RBS given only one pair of sets E_m^T and E_m^F. That is, the inference problem under investigation can be formulated as follows:

Problem:
Given a pair of two E_m^T and E_m^F $(N \geq m > N_1)$ sets derive a RBS that has the **minimum** number of rules and can derive the above pair of E_i^T and E_i^F sets.

5.0 Proposed Algorithm

5.1 A Preliminary Discussion.

The problems presented in the previous sections are examined from an Integer Programming point of view. IP has long been used in RBS related problems (see, for example, [Williams, 1986], [Jeroslow, 1988], [Hooker, 1988a, 1988b]). These treatments deal mainly with the problem of inferring facts given a set of conditions (rules) and assertions (logical inference problem). The proposed approach uses two IP models. The first IP model determines the minimum number of rules that can be derived from a pair of E_m^T and E_m^F sets. The second IP model uses the results of the previous model and determines a representative set of rules that can derive the members of the E_m^T set, while they do not derive the members of the E_m^F set. In order the motivation of this formulation to become clear, an example is examined first. Then, a formal formulation follows as generalization of this example.

Example 4:

Let $N = 6$ and $N_1 = 5$. Suppose that the following data are also available:

$a_1 = (1\ 1\ 1\ 0\ 0) \in E_6^T$

$a_2 = (1\ 1\ 0\ 1\ 0) \in E_6^T$

$a_3 = (0\ 1\ 1\ 0\ 0) \in E_6^T$

and $a_4 = (0\ 1\ 0\ 1\ 0) \in E_6^F$

That is:

$E_6^T = \{ a_1, a_2, a_3 \}, \quad E_6^F = \{ a_4 \}$

Variable Definitions

Let $M_1 = | E_m^T | = | E_6^T |$ and $M_2 = | E_m^F | = | E_6^F |$ (where: $| S |$ means the **cardinality** of set S). Since there are three strings in E_6^T, it is reasonable to assume that at **most** three rules (ie: M_1) will be derived (one for each string). Then, the following variables are being introduced:

[Variable Set #1]

Let

$X_{ij} = 1$ *if rule i has premise* P_j *on its antecedent part*

$X_{ij} = 0$ *otherwise*

where:

$1 \leq i \leq M_1 (= 3)$

$1 \leq j \leq N_1 (= 5)$

[Variable Set #2]

Let

$W_{ij} = 1$ *if string* a_j ($a_j \in E_6^T$) *was derived by rule i*

$W_{ij} = 0$ *otherwise*

where:

$1 \leq i \leq M_1 (= 3)$

$1 \leq j \leq M_1 (= 3)$

[Variable Set #3]

Let

$R_i = 1$ *if rule i is NOT nil*

$R_i = 0$ *if rule i is nil* (*ie: there are NO premises in its antecedent part*)

where:

$1 \leq i \leq M_1 (= 3)$

Using the above variables the following constraints are being introduced:

[Constraint Set #1]

The first family of constraints requires each member of E_6^T to be derived by at least **one** rule. For this example we have:

$a_1 \in E_6^T$ which requires that:

$$(X_{11}+X_{21}+X_{31}) + (X_{12}+X_{22}+X_{32}) + (X_{13}+X_{23}+X_{33}) \geq 1$$

Also $a_2 \in E_6^T$ requires that:

$$(X_{11}+X_{21}+X_{32}) + (X_{12}+X_{22}+X_{32}) + (X_{14}+X_{24}+X_{34}) \geq 1$$

Similarly, $a_3 \in E_6^T$ requires that:

$$(X_{12}+X_{22}+X_{32}) + (X_{13}+X_{23}+X_{33}) \geq 1$$

[Constraint Set #2]

If a_j ($a_j \in E_6^T$) was derived by rule i then, by the definition of the second class of variables it should be: $W_{ij} = 1$. This statement indicates that if $W_{ij} = 1$ then, all the X variables of the i[th] rule that correspond to premises that are **not true** in string a_j should be equal to zero. Therefore we have:

$a_1 \in E_6^T$ which requires that:

$$2 \times W_{11} + (X_{14}+X_{15}) \leq 2$$
$$2 \times W_{21} + (X_{24}+X_{25}) \leq 2$$
$$2 \times W_{31} + (X_{34}+X_{35}) \leq 2$$

Similarly, $a_2 \in E_6^T$ requires that:

$$2 \times W_{12} + (X_{13}+X_{15}) \leq 2$$
$$2 \times W_{22} + (X_{23}+X_{25}) \leq 2$$
$$2 \times W_{32} + (X_{33}+X_{35}) \leq 2$$

[Constraint Set #3]

Since each $a_j \in E_6^T$ was derived by at **least** one rule, the following constraints should be satisfied:

$$W_{11}+W_{21}+W_{31} \geq 1$$
$$W_{12}+W_{22}+W_{32} \geq 1$$
$$W_{13}+W_{23}+W_{33} \geq 1$$

[Constraint Set #4]

If $a_L \in E_6^F$ then, the premises that are true in a_L cannot be present **alone** in the antecedent part of any rule. For this example, since $a_4 \in E_6^F$ it follows that:

$$if\ (X_{i2}+X_{i4}) \geq 1 \ \ then,\ (X_{i1}+X_{i3}+X_{i5}) \geq 1$$
$$for\ any\ i=1,2,3\ (=M_1)$$

or:

$$2\times(X_{11}+X_{13}+X_{15}) \geq (X_{12}+X_{14})$$
$$2\times(X_{21}+X_{23}+X_{25}) \geq (X_{22}+X_{24})$$
$$2\times(X_{31}+X_{33}+X_{35}) \geq (X_{32}+X_{34})$$

[Constraint Set #5]

This family of constraints simply requires that if at least one premise in the antecedent part of a rule is true then, the corresponding R variable has to be equal to 1. For this example this is achieved by the following three constraints:

$$5\times R_1 \geq X_{11}+X_{12}+X_{13}+X_{14}+X_{15}$$
$$5\times R_2 \geq X_{21}+X_{22}+X_{23}+X_{24}+X_{25}$$
$$5\times R_3 \geq X_{31}+X_{32}+X_{33}+X_{34}+X_{35}$$

[Constraint Set #6]

This family of constraints requires that if all the premises in the antecedent part of a rule is true then, the corresponding R variable has to be equal to 0. For this example this is achieved by the following three constraints:

$$X_{11}+X_{12}+X_{13}+X_{14}+X_{15} \geq R_1$$
$$X_{21}+X_{22}+X_{23}+X_{24}+X_{25} \geq R_2$$
$$X_{31}+X_{32}+X_{33}+X_{34}+X_{35} \geq R_3$$

[Constraint Set #7]

The last family of constraints requires that if an R_i variable is zero then all the corresponding W_{ij} should be equal to 0 for any value of j. For this example this is achieved by the following three constraints:

$$W_{11}+W_{12}+W_{13} \leq 3\times R_1$$
$$W_{21}+W_{22}+W_{23} \leq 3\times R_2$$
$$W_{31}+W_{32}+W_{33} \leq 3\times R_3$$

This IP approach has **two phases.** In the first phase **the minimum number of rules** is determined. That is, the following objective function is used.

$$MIN \sum_{i=1}^{M_1} R_i$$

In the second phase, the body of constraints is first simplified. That is, the variables that correspond to rules that were derived to be nil in the first phase are dropped. This step reduces the complexity of the current formulation. After the simplification step, a maximization function is used. This function is as follows:

$$MAX \sum_{i=1}^{MM} \sum_{j=1}^{N_1} X_{ij}$$

Where MM is the value of the objective function of the first phase. Also in the second phase the indices are modified to express the compaction of the current formulation.

Phase two results in each rule (ie: non nil rule) to have the **maximum** number of premises. Suppose that rule z has the following premises in its antecedent part:

$$S_z = \{ P_{z_1}, P_{z_2}, P_{z_3}, \ldots, P_{z_n} \}$$

Then, this rule is allowed to have as antecedent parts members of the following set f_z :

$$f_z = \phi(S_z) - \bigcup_{a_j \in E_m^F} \phi(t(a_j))$$

The above discussion can be summarized in the following IP formulation:

$$MIN \ (R_1 + R_2 + R_3) \qquad (\textit{for phase one})$$

subject to:

$$(X_{11} + X_{21} + X_{31}) + (X_{12} + X_{22} + X_{32}) + (X_{13} + X_{23} + X_{33}) \geq 1$$
$$(X_{11} + X_{21} + X_{32}) + (X_{12} + X_{22} + X_{32}) + (X_{14} + X_{24} + X_{34}) \geq 1$$
$$(X_{12} + X_{22} + X_{32}) + (X_{13} + X_{23} + X_{33}) \geq 1$$

$$2 \times W_{11} + (X_{14} + X_{15}) \leq 2$$
$$2 \times W_{21} + (X_{24} + X_{25}) \leq 2$$
$$2 \times W_{31} + (X_{34} + X_{35}) \leq 2$$
$$2 \times W_{12} + (X_{13} + X_{15}) \leq 2$$
$$2 \times W_{22} + (X_{23} + X_{25}) \leq 2$$
$$2 \times W_{32} + (X_{33} + X_{35}) \leq 2$$

$$W_{11} + W_{21} + W_{31} \geq 1$$
$$W_{12} + W_{22} + W_{32} \geq 1$$
$$W_{13} + W_{23} + W_{33} \geq 1$$

$$2 \times (X_{11} + X_{13} + X_{15}) \geq (X_{12} + X_{14})$$

$$2\times(X_{21}+X_{23}+X_{25}) \geq (X_{22}+X_{24})$$
$$2\times(X_{31}+X_{33}+X_{35}) \geq (X_{32}+X_{34})$$

$$5\times R_1 \geq X_{11}+X_{12}+X_{13}+X_{14}+X_{15}$$
$$5\times R_2 \geq X_{21}+X_{22}+X_{23}+X_{24}+X_{25}$$
$$5\times R_3 \geq X_{31}+X_{32}+X_{33}+X_{34}+X_{35}$$

$$X_{11}+X_{12}+X_{13}+X_{14}+X_{15} \geq R_1$$
$$X_{21}+X_{22}+X_{23}+X_{24}+X_{25} \geq R_2$$
$$X_{31}+X_{32}+X_{33}+X_{34}+X_{35} \geq R_3$$

$$W_{11}+W_{12}+W_{13} \leq 3\times R_1$$
$$W_{21}+W_{22}+W_{23} \leq 3\times R_2$$
$$W_{31}+W_{32}+W_{33} \leq 3\times R_3$$

where the R, X, and W, are 0-1 variables.

When the phase one model is solved, the solution indicates that:

$X_{11}=0 \quad X_{21}=1 \quad X_{31}=0$

$X_{12}=0 \quad X_{22}=1 \quad X_{32}=0$

$X_{13}=0 \quad X_{23}=0 \quad X_{33}=1$

$X_{14}=0 \quad X_{24}=0 \quad X_{34}=0$

$X_{15}=0 \quad X_{25}=0 \quad X_{35}=0$

$W_{11}=0 \quad W_{21}=1 \quad W_{31}=0$

$W_{12}=0 \quad W_{22}=1 \quad W_{32}=0$

$W_{13}=0 \quad W_{23}=0 \quad W_{33}=1$

$R_1=0 \quad R_2=1 \quad R_3=1$

That is, two is the minimum number of rules for this example! Since three rules were allowed as the maximum number of rules and the result of phase one indicates two rules, **a simplification** of the set of constraints is possible. The (simplified) model for the second phase is as follows:

$$MAX \sum_{i=2}^{3}\sum_{j=1}^{5} X_{ij} \qquad \textit{(for phase two)}$$

subject to:

$$(X_{21}+X_{31}) + (X_{22}+X_{32}) + (X_{23}+X_{33}) \geq 1$$
$$(X_{21}+X_{32}) + (X_{22}+X_{32}) + (X_{24}+X_{34}) \geq 1$$
$$(X_{22}+X_{32}) + (X_{23}+X_{33}) \geq 1$$

$$2\times W_{21} + (X_{24}+X_{25}) \leq 2$$
$$2\times W_{31} + (X_{34}+X_{35}) \leq 2$$
$$2\times W_{22} + (X_{23}+X_{25}) \leq 2$$
$$2\times W_{32} + (X_{33}+X_{35}) \leq 2$$

$$W_{21}+W_{31} \geq 1$$
$$W_{22}+W_{32} \geq 1$$
$$W_{23}+W_{33} \geq 1$$

$$2\times(X_{21}+X_{23}+X_{25}) \geq (X_{22}+X_{24})$$
$$2\times(X_{31}+X_{33}+X_{35}) \geq (X_{32}+X_{34})$$

$$5\times R_2 \geq X_{21}+X_{22}+X_{23}+X_{24}+X_{25}$$
$$5\times R_3 \geq X_{31}+X_{32}+X_{33}+X_{34}+X_{35}$$

$$X_{21}+X_{22}+X_{23}+X_{24}+X_{25} \geq R_2$$
$$X_{31}+X_{32}+X_{33}+X_{34}+X_{35} \geq R_3$$

$$W_{21}+W_{22}+W_{23} \leq 3\times R_2$$
$$W_{31}+W_{32}+W_{33} \leq 3\times R_3$$

where the R, X, and W, are 0-1 variables.

When the phase two model runs, the following solution is derived:

$X_{21}=1 \quad X_{31}=0$

$X_{22}=1 \quad X_{32}=1$

$X_{23}=0 \quad X_{33}=1$

$X_{24}=1 \quad X_{34}=0$

$X_{25}=0 \quad X_{35}=0$

$W_{21}=0 \quad W_{31}=1$

$$W_{22} = 1 \quad W_{32} = 0$$
$$W_{23} = 0 \quad W_{33} = 1$$
$$R_2 = 1 \quad R_3 = 1$$

That is, the proposed RBS is:

$$\{\{1, 2, 4\}, \quad \{6\}\}$$
$$\{\{2, 3 \quad\}, \quad \{6\}\}$$

or:

$$IF\ (P_1\ AND\ P_2\ AND\ P_4)\ \ THEN\ (P_6)$$
$$IF\ (P_2\ AND\ P_3)\ \ \ \ \ \ \ \ THEN\ (P_6)$$

5.2 A Generalization of the IP Formulation

The above formulation can easily be generalized. The general model for phase one is as follows (the model for phase two is obvious and hence omitted):

$$MIN \sum_{i=1}^{M_1} R_i \qquad \textit{(for phase one)}$$

subject to:

[Constraint Set #1]

$$\sum_{i=1}^{M_1} \sum_{\text{all } j \text{ such that: } P_j \in t(a_k)} X_{ij} \geq 1 \qquad \text{for any } a_k \in E_m^T$$

[Constraint Set #2]

$$n \times W_{ik} + \sum_{\text{all } j \text{ such that: } P_j \in f(a_k)} X_{ij} \leq n \qquad \text{for any } a_k \in E_m^T \text{ and } i = 1,2,3,\ldots,M_1$$

where: $n = |f(a_k)|$

[Constraint Set #3]

$$\sum_{i=1}^{M_1} W_{ij} \geq 1 \qquad \text{for any } j \text{ such that}: a_j \in E_m^T$$

[Constraint Set #4]

$$n \times \sum_{\text{all } j \text{ such that: } P_j \in f(a_k)} X_{ij} \leq \sum_{\text{all } j \text{ such that: } P_j \in t(a_k)} X_{ij} \qquad \text{for any } a_k \in E_m^F \text{ and } i = 1,2,3,\ldots,M_1$$

where: $n = |t(a_k)|$

[Constraint Set #5]

$$N_1 R_i \geq \sum_{j=1}^{M_1} X_{ij} \qquad \textit{for any } i = 1,2,3,\ldots,M_1$$

[Constraint Set #6]

$$\sum_{j=1}^{N_1} X_{ij} \geq R_i \qquad \textit{for any } i = 1,2,3,\ldots,M_1$$

[Constraint Set #7]

$$\sum_{j=1}^{M_1} W_{ij} \leq M_1 \times R_i \qquad \textit{for any } i = 1,2,3,\ldots,M_1$$

where the R, X, and W, are 0-1 variables.

From the above formulations it follows that the number of the X_{ij} variables is $M_1 \times N_1$, the number of the W_{ij} variables is $M_1 \times M_1$, and the number of the R_i variables is M_1. That is, there are $M_1 \times (N_1 + M_1 + 1)$ 0-1 variables in this IP formulation. Similarly, the constraint sets 1, 2, 3, ..., 7 have M_1, $M_1 \times M_1$, M_1, $M_2 \times M_1$, M_1, M_1, and M_1 constraints, respectively. That is, there are $M_1 \times (M_1 + M_2 + 5)$ linear constraints.

This analysis of the size of the IP formulation reveals that the derived approach is linearly dependent on the number of input premises (*ie*: N_1), the number of members of the set E_m^F (*ie*: M_2), and on the square of the number of members of the set E_m^T (*ie*: M_1). Thus, the small size of the models, makes the application of the present IP approach to be a flexible one.

6.0 Concluding Remarks

In this paper we considered the problem of learning structures of rules in Rule-Based Systems from selections of examples. This problem is one of the major issues in machine learning. Learning from examples has been realized as a critical issue in Knowledge Engineering. Most often the structure of the rules in the knowledge base of a RBS is not known and has to be derived from examples. As the problems that RBSs are called to solve today become increasingly complex, the problem of machine learning becomes more vital and difficult at the same time.

A formalization of RBSs was introduced in order to highlight the fundamental issues of the learning problem that this paper is dealing with. This formalization leads to an IP approach that effectively treats the problem of inferring rules from examples. The sizes of the resulting IP models make this approach to be particularly powerful.

Integer Programming has long been used in logical inference from a set of facts and rules. In that context IP turned out to be a very powerful approach, since many difficult logical inference problems can be solved this way. However, the RBS inference problem (ie: inferring rules from examples) has not been intensively examined from an IP point of view. The present work is a step towards this direction. The successful treatments presented in this paper indicate that IP can also be a powerful tool in dealing with many complex machine learning problems.

References

DIETTERICH, T.C., and R.S. MICHALSKI, *A Comparative Review of Selected Methods for Learning from Examples,* Machine Learning: An Artificial Intelligence Approach, R.S. Michalski, J.G. Carbonell, and T.M. Mitchell (Eds.), Tioga Publishing Company, Palo Alto, CA, 1983, pp. 41-81.

FAGAN, J.M., *VM: Representing Time-Depedent Relations in a Medical Setting,* Doctoral Dissertation, Computer Science Department, Stanford University, Stanford, CA, June 1980.

FISHER, D.H., *Conceptual Clustering, Learning from Examples, and Inference* Proceedings of the Fourth International Workshop on Machine Learning, P. Langley (Ed.), June 22-25, 1987, University of California, Irvine, pp. 38-49.

FU, K.S., *Grammatical Inference: Introduction and Survey,* IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-5, No. 1, 1975, pp. 409-423.

HELFT, N., *Learning Systems of First-Order Rules,* Proceedings of the Fifth International Conference on Machine Learning, John Laird (Ed.), June 12-14, 1988, University of Michigan, Ann Arbor, pp. 395-401.

HOOKER, J.N., *Generalized Resolution and Cutting Planes,* Annals of Operations Research, R.G. Jeroslow (Ed.), Vol. 12, No. 1-4, 1988a, pp. 217-239.

HOOKER, J.N., *A Quantitative Approach to Logical Inference,* Decision Support Systems, North-Holland, No. 4, 1988b, pp. 45-69.

KADIE, C.M., *Diffy-S: Learning Robot Operator Schemata from Examples,* Proceedings of the Fifth International Conference on Machine Learning, John Laird (Ed.), June 12-14, 1988, University of Michigan, Ann Arbor, pp. 430-436.

JEROSLOW, R.G., *Computation-Oriented Reductions of Predicate to Propositional Logic,* Decision Support Systems, North-Holland, No. 4, 1988, pp. 183-197.

McCARTHY, J., *Programs with Common Sense,* Proceedings of the Symposium on the Mechanization of Thought Processes, National Physical Laboratory I:77-84 (Reprinted in M.L. Minsky (Ed.), 1968, Semantic Information Processing, Cambridge, Mass, MIT Press, pp. 403-409.

MICHALSKI, R.S., *Machine Learning Research in the Artificial Intelligence Laboratory at Illinois,* Machine Learning: A Guide to Current Research, T.M. Mitchell, J.G. Carbonell, R.S. Michalski (Eds.), Kluwer Academic Publishers, Boston, Mass, 1986. pp. 193-198.

MOZETIC, I., *Knowledge Extraction Through Learning from Examples,* Machine Learning: A Guide to Current Research, T.M. Mitchell, J.G. Carbonell, R.S. Michalski (Eds.), Kluwer Academic Publishers, Boston, Mass, 1986. pp. 227-232.

STEFIK, M., J. AIKINS, R. BALZER, J. BENOIT, L. BIRNBAUM, F. HAYES-ROTH, and E. SACERDOTI, *The Organization of Expert Systems, A Tutorial,* Artificial Intelligence, No. 18, 1982, pp. 135-173.

TALLIS, H., Tuning Rule-Based Systems to Their Environments, Proceedings of the Fifth International Conference on Machine Learning, John Laird (Ed.), June 12-14, 1988, University of Michigan, Ann Arbor, pp. 8-14.

WILLIAMS, H.P., *Linear and Integer Programming Applied to Artificial Intelligence,* preprint series, University of Southampton, Faculty of Mathematical Studies, July 1986, pp. 1-33.

Evangelos Triantaphyllou

KNOWLEDGE ENGINEERING TODAY'S MARKETPLACE

IAKE '89

University of Maryland
University College
College Park, Maryland

June 26 - 28, 1989

Proceedings
Annual Conference of the
International Association of Knowledge Engineers

Cosponsors:

University of Maryland/University College
U.S. Air Force Logistics Command
Artificial Intelligence Division