

Applications of Signatures to Expert Systems Modelling

Claudiu Pozna^{1,2}, Radu-Emil Precup³

¹Department of Informatics, Széchenyi István University, Egyetem tér 1, H-9026 Győr, Hungary

²Department of Automation and Information Technology, Transilvania University of Brasov, Str. Mihai Viteazu 5, RO-500174 Brasov, Romania
E-mail: cp@unitbv.ro

³Department of Automation and Applied Informatics, Politehnica University of Timisoara, Bd. V. Parvan 2, RO-300223 Timisoara, Romania
E-mail: radu.precup@aut.upt.ro

Abstract: This paper offers a new approach that applies the signatures to expert systems modelling. Signatures and their operators, viewed as a generalization of fuzzy signatures, represent a convenient framework for the symbolic representation of data. The models are derived by a three-step algorithm that maps the signatures onto expert systems. An expert systems modelling algorithm is given. Our algorithm has two inputs, the knowledge base, i.e., the rules, and the data base, i.e., the facts, and it constructs the signatures which represent models of expert systems. The algorithm is advantageous because of its systematic and general formulation allowing for the modelling of uncertain expert systems. The theoretical results are exemplified by a case study which produces models of a Bayesian expert system with mechatronics applications.

Keywords: expert systems; mechatronics; modelling; operators; rules; signatures

1 Introduction

Expert systems are products of artificial intelligence that aim the goal of problem solving. The problems belong to various domains, and expert level knowledge is used with this regard. Successful implementations of expert systems include evolving classification and fuzzy systems [1, 2], multi-agent systems [3, 4], data mining [5, 6], and meta-heuristic engines [7-9]. As mentioned in [10, 11], the better understanding of how natural cognitive processes can effectively co-evolve with processes in artificially cognitive systems in the framework of Cognitive Infocommunications (CogInfoCom) [12-14]. Rule-based expert systems will be addressed in this paper as a step in the context of cognitive processes modelling.

Several approaches to expert systems modelling are currently reported in the literature. Multi-expert models are discussed in [15]. The combination of fuzzy logic, neural networks, genetic algorithms and statistic analysis is analysed in [16]. Reliable rule-based systems with uncertainty are obtained in [17] on the basis of semantic data integration. Model-driven engineering techniques for the development of multi-agent systems are proposed in [18]. A comparison of structure learning approaches to structure learning for belief rule base expert system is conducted in [19].

The new idea of this paper in the context of the above analysed literature is a new approach to expert systems modelling based on signatures. Signatures and their operators are defined in [20] as a convenient framework for the symbolic representation of data as a generalization of fuzzy signatures [21, 22]. Applications of signatures are given in [10, 23, 24].

The new modelling approach is formulated as an original three-step algorithm that maps the signatures onto expert systems. The algorithm has two inputs represented by the knowledge base which are the rules, and by the data base which represents the facts. The algorithm constructs the signatures which stand for a class of expert systems models.

Our expert systems modelling approach is important and advantageous with respect to the state-of-the-art because of the following reasons:

- The systematic formulation in terms of an algorithm offers transparency and relatively simple modelling.
- The formulation of the algorithm is general and applicable to both certain and uncertain expert systems.

This paper is organized as follows: a short overview on signatures and on their operators is presented in the next section. Section 3 is dedicated to the new modelling approach. An illustrative example is included and the modelling algorithm is given. Section 4 validates the theoretical approach by a case study focused on the construction of signature-based models of a Bayesian expert system defined in [25]. The conclusions are finally outlined.

2 Overview on Signatures and on Their Operators

Signatures and operators on signatures are defined and analysed in [20]. A part of the definitions which enable the modelling of expert systems is presented in this section, and the reader is invited to address [20] for examples to understand the concept of signatures. Let $S^{(n)}$ be a set defined recursively as

$$S^{(n)} = \prod_{i=1}^n S_i, \quad S_i = \mathbf{R} \text{ or } S_i = S^{(m)}, \quad m \geq 1, \quad i = 1 \dots n, \quad (1)$$

where \mathbf{R} is the set of real numbers, and \prod is the Cartesian product.

Definition 1. Let X be a nonempty set. The *collection of signatures* is defined as the function $A: X \rightarrow S^{(n)}$, the *signature* of the element $x \in X$ is $A(x) \in S^{(n)}$ and the *transposition* of the signature $A(x)$ is represented by $A^T(x)$ given as

$$A(x) = \left[\begin{array}{c} \dots \\ a_i \\ \left[\begin{array}{c} a_{i+1,1} \\ a_{i+1,2} \end{array} \right] \\ a_{i+2,1} \\ \left[\begin{array}{c} a_{i+2,2,1} \\ a_{i+2,2,2} \end{array} \right] \\ \dots \end{array} \right], \quad (2)$$

$$A^T(x) = [\dots \quad a_i \quad [a_{i+1,1} \quad a_{i+1,2}] \quad [a_{i+2,1} \quad [a_{i+2,2,1} \quad a_{i+2,2,2}]] \quad \dots].$$

The following notations are introduced in [20] to simplify the characterization of signatures:

- A signature $A(x)$ with values $a_1, a_2, \dots, a_n, a_{i,1}, a_{i,2}, \dots, a_{i,m}, \dots, a_{j,k,l}, \dots$, is indicated by a^{\dots} .
- $A(x) = a^{1 \dots n}$ is used if $\exists x \in X$ and $A^T(x) = [a_1 \quad \dots \quad a_n]$.
- If $\exists y \in Y$ and $A^T(y) = [a_1 \quad \dots \quad a_{i-1} \quad [a_{i,1} \quad \dots \quad a_{i,m}] \quad a_{i+1} \quad \dots \quad a_n]$, then we will use the notation $A(y) = a^{1 \dots [1, \dots, m]_i \dots n}$. The sets are defined here as $S_1 = S_2 = \dots = S_{i-1} = S_{i+1} = \dots = S_n = \mathbf{R}$, and their Cartesian product is expressed as $S_i = \prod_{i=1}^m \mathbf{R} = \mathbf{R}^m$.
- A signature of type $[\dots \quad [[a_1]] \quad \dots]$ is equivalent to the signature $[a_1]$, where $a_1 \in \mathbf{R}$.

As shown in [20], signatures can be used in complex data representation. Some definitions of operators on signatures will be exemplified in the sequel.

Definition 2. The *contraction* of a signature is defined as one of the three functions

$$\begin{aligned}
& f @ : \mathbf{S} \rightarrow \mathbf{S}, \quad f @(a^{1,2,\dots,n}) = a^1 = [a], \\
& f @_i : \mathbf{S} \rightarrow \mathbf{S}, \\
& \begin{cases} f @_i (a^{1,\dots,[1,\dots,m]_i,\dots,n}) = a^{1,\dots,i,\dots,n}, & \text{if } i \leq n, \\ f @_i (a^{\dots}) = a^{\dots}, & \text{otherwise,} \end{cases} \quad (3) \\
& f @_{i,j,k} : \mathbf{S} \rightarrow \mathbf{S}, \\
& \begin{cases} f @_{i,j,k} (a^{1,\dots,[1,\dots,[1,\dots,[1,\dots,q]_k,\dots,r]_j,\dots,m]_i,\dots,n}) \\ = a^{1,\dots,[1,\dots,[1,\dots,k,\dots,r]_j,\dots,m]_i,\dots,n}, & \text{if } i \leq n, j \leq m, k \leq r, \\ f @_{i,j,k} (a^{\dots}) = a^{\dots}, & \text{otherwise,} \end{cases}
\end{aligned}$$

where $a = f(a_1, a_2, \dots, a_n)$, $f : \mathbf{R}^n \rightarrow \mathbf{R}$, $a_i = f(a_{i1}, \dots, a_{im})$, $f : \mathbf{R}^m \rightarrow \mathbf{R}$, and $a_{i,j,k} = f(a_{i,j,k1}, \dots, a_{i,j,kq})$, $f : \mathbf{R}^q \rightarrow \mathbf{R}$. Extra indices can be inserted after i , j and k , to generalize this definition. We use the following notation for the absolute value of a contraction if a has the first form in (3):

$$|f @(a^{1,2,\dots,n})| = a. \quad (4)$$

Definition 3. The *extension* of a signature is defined as one of the functions

$$\begin{aligned}
& {}^g \overline{@}_i^p : \mathbf{S} \rightarrow \mathbf{S}, \\
& \begin{cases} {}^g \overline{@}_i^p (a^{1,2,\dots,i,\dots,n}) = a^{1,2,\dots,[1,\dots,p]_i,\dots,n}, & \text{if } i \leq n, \\ {}^g \overline{@}_i^p (a^{\dots}) = a^{\dots}, & \text{otherwise,} \end{cases} \\
& {}^g \overline{@}_{i,j,k}^q : \mathbf{S} \rightarrow \mathbf{S}, \quad (5) \\
& \begin{cases} {}^g \overline{@}_{i,j,k}^q (a^{1,\dots,[1,\dots,[1,\dots,k,\dots,r]_j,\dots,m]_i,\dots,n}) \\ = a^{1,\dots,[1,\dots,[1,\dots,[1,\dots,q]_k,\dots,r]_j,\dots,m]_i,\dots,n}, & \text{if } i \leq n, j \leq m, k \leq r, \\ {}^g \overline{@}_{i,j,k}^q (a^{\dots}) = a^{\dots}, & \text{otherwise,} \end{cases}
\end{aligned}$$

where two forms of the function g are used, $g : \mathbf{R} \rightarrow \mathbf{R}^p$, $g(a_i) = [a_{i1}, \dots, a_{ip}]$, $g : \mathbf{R} \rightarrow \mathbf{R}^q$, $g(a_{i,j,k}) = [a_{i,j,k1}, \dots, a_{i,j,kq}]$, and we can continue to generalize it by adding extra indices after i , j and k . The *zero-step extension* of a signature is defined as the function

$$\begin{aligned}
& {}^g \overline{@}^p : \mathbf{S} \rightarrow \mathbf{S}, \\
& {}^g \overline{@}^p (a^{1,2,\dots,n}) = {}^g \overline{@}_1^p ({}^g \overline{@}_2^p (\dots ({}^g \overline{@}_n^p (a^{1,2,\dots,n})))) = a^{[1,\dots,p],[1,\dots,p],\dots,[1,\dots,p]}, \quad (6)
\end{aligned}$$

where $g : \mathbf{R} \rightarrow \mathbf{R}^p$, $g(a_j) = [a_{j1}, \dots, a_{jp}]$, $j = 1 \dots n$.

Definition 4. The *pruning* of a signature is defined as one of the three functions

$$\begin{aligned}
 & \emptyset_i : \mathbf{S} \rightarrow \mathbf{S}, \\
 & \begin{cases} \emptyset_i(a^{1,2,\dots,j,\dots,n}) = a^{1,2,\dots,j-1,i+1,\dots,n}, & \text{if } i \leq n, \\ \emptyset_i(a^{\dots}) = a^{\dots}, & \text{otherwise,} \end{cases} \\
 & \emptyset_i : \mathbf{S} \rightarrow \mathbf{S}, \\
 & \begin{cases} \emptyset_i(a^{1,\dots,[1,\dots,m]_i,\dots,n}) = a^{1,\dots,j-1,i+1,\dots,n}, & \text{if } i \leq n, \\ \emptyset_i(a^{\dots}) = a^{\dots}, & \text{otherwise,} \end{cases} \quad (7) \\
 & \emptyset_{i,j,k} : \mathbf{S} \rightarrow \mathbf{S}, \\
 & \begin{cases} \emptyset_{i,j,k}(a^{1,\dots,[1,\dots,[1,\dots,[1,\dots,q]_k,\dots,r]_j,\dots,m]_i,\dots,n}) & \text{if } i \leq n, j \leq m, k \leq r, \\ = a^{1,\dots,[1,\dots,[1,\dots,k-1,k+1,\dots,r]_j,\dots,m]_i,\dots,n}, & \\ \emptyset_{i,j,k}(a^{\dots}) = a^{\dots}, & \text{otherwise.} \end{cases}
 \end{aligned}$$

This definition can be generalized as well by adding extra indices after i, j and k , where $a^{\dots}\emptyset_i = b^{\dots}$ is the notation for the pruning of a signature.

Definition 5. The *addition* of two signatures is defined as the function

$$\begin{aligned}
 & {}^f\oplus_i : \mathbf{S} \times \mathbf{S} \rightarrow \mathbf{S}, \\
 & {}^f\oplus_i(a^{1,2,\dots,[1,\dots,m]_i,\dots,n}, b^{1,2,\dots,m}) = c^{1,2,\dots,[1,\dots,m]_i,\dots,n}, \quad (8)
 \end{aligned}$$

where $c_k = a_k \quad \forall k \neq i$, $c_{ij} = f(a_{ij}, b_j) \quad \forall j = 1..m$, $f : \mathbf{R}^2 \rightarrow \mathbf{R}$. If $a^{1,2,\dots,n}, b^{1,2,\dots,m} \in \mathbf{S}$, then

$$\begin{aligned}
 & \begin{cases} a^{1,2,\dots,n} {}^f\oplus_i b^{1,2,\dots,m} = {}^g\overline{\oplus}_i^m(a^{1,2,\dots,n}) {}^f\oplus_i b^{1,2,\dots,m} & \text{if } i \leq n, \\ = c^{1,2,\dots,[1,2,\dots,m]_i,\dots,n}, & \\ a^{\dots} {}^f\oplus_i b^{\dots} = a^{\dots}, & \text{otherwise,} \end{cases} \quad (9)
 \end{aligned}$$

where $c_{ij} = f(a_{ij}, b_j)$, $[a_{i1}, \dots, a_{im}] = g(a_i)$, $j = 1..m$, $c_k = a_k \quad \forall k \neq i$. We can generalize this definition by adding extra indices after i , where $a^{\dots} {}^f\oplus_i b^{\dots} = c^{\dots}$ is the notation for the addition of two signatures.

All these operators presented above refer to structural transformations and data transformations. Several examples concerning the application of these operators and of other operators (e.g., grafting and multiplication) are presented in [20]. These operators can be conveniently implemented as software objects. The extension to inequalities is also of interest [26-28] with focus on mechatronics applications [29-36].

3 Modelling Approach

The structure of a rule-based expert system is presented in Figure 1 (a) which points out the following subsystems: the knowledge base which contains the “If...Then...” rules, the database which contains the facts, the inference engine where the goal of the expert system is computed, and the user interface where the user interacts with the expert system. Several internal elements can be added to this structure; they include explanation facilities where the results are explained systematically, and the developer interface where the expert system interacts with the developer. External elements can be included as well like external databases or programs which support the inference engine.

The core of the expert system is the inference engine, where the rules are fired using the known facts. After firing a rule a new fact is inferred; this can fire in turn a new rule. This process is cyclic, and it can be represented by the schema illustrated in Figure 1 (b). The end of the cycle is obtained when no more rules can be fired and the knowledge on the goal is obtained.

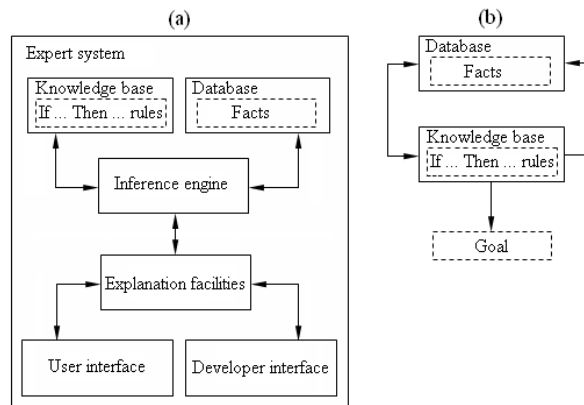


Figure 1

Structure of rule-based expert system (a), and cycles of inference engine (b)

The cycle presented in Figure 1 (b) suggests that the rules are interconnected in an inference chain which starts with facts and ends with the problem goal. Two ways are rules to execute the rules, forward chaining and backward chaining. The forward chaining is data-driven reasoning expressed as: the reasoning starts with the known data, each time only the topmost rule is executed, when fired the rule adds a new fact in the database, and the cycle stops when no further rules can be fired. Contrarily, the backward chaining is goal-driven reasoning, where: the knowledge base is first searched to find the rules that might have the desired solution, and if this rule contains facts which are not observed then these facts are next replaced with related rules until all rules from the decision chain contain observable facts.

Signatures will be used as follows as data structures to model the processes specific to expert systems. Our idea is to transform an inference chain into a signature and thus to map signatures onto expert systems. The inference chains can be different because of the observed facts. This means that we can divide the operating processes of expert systems in two steps:

- I. First, carry out the backward construction of the signature starting with the goal and replacing the unobserved facts with rules until all rules contains observed facts.
- II. Second, apply certain operators to the already constructed signature and compute the goal of the expert system.

Each “*If...Then...*” rule contains a relation between two types of facts. The first part of rule, “*If...*”, concerns the antecedent facts, and the second part of the rule, “*Then...*”, is related to the consequent facts. The terms “observed / unobserved facts” mentioned several times in the previous section refer to the antecedent facts. If such a fact is not observed the rule cannot be fired. In this case the only solution is to find another rule (from the knowledge base) which contains the unobserved fact like a consequent fact.

The construction of the inference chain is exemplified by means of the following example. If the goal involves Z and the knowledge base contain the rule

$$\begin{aligned} \text{Rule 1: } & \textit{If } Y \textit{ Then } Z, \\ \text{Rule 2: } & \textit{If } X \textit{ Then } Y, \end{aligned} \tag{10}$$

we will start (backward) with the rule 1. If Y is observed, then rule 1 will be fired and the knowledge on Z will be obtained. If not, Y will be replaced by rule 2:

$$\textit{If (If } X \textit{ Then } Y) \textit{ Then } Z, \tag{11}$$

and we will ask about the observability of the fact X .

We propose the following typology of facts: observable facts: the facts which can be inputted to the expert system, unobservable facts: facts which are not observable, inferable facts: facts which are established by rules, and uninferable facts: facts which are not inferable. An uninferable fact must be observable and that the inferable facts can be observable or unobservable. Observable facts are not observed in certain situations. An observed or inferred fact can be modelled with the value 1; contrarily, if the fact is not observed or inferred, it can be modelled with the value 0.

Since a rule can be modelled by a signature, the construction of the inference chains can be modelled by operators on signatures as well. In this regard, two definitions are suggested as follows. The following definition concerns a rule which is a dependency between two types of facts, viz. the antecedents and the consequences.

Definition 8. A rule is modelled by signatures in one of the three forms

Rule r_1 : *If A Then Z* is equivalent to $r_1 = [A]^T$ and $@(r_1) = [Z]^T$,

Rule r_1 : *If A ∧ B ∧ C Then Z* is equivalent to

$$r_1 = [A, B, C]^T \text{ and } {}^f @ (r_1) = [Z]^T, \quad (12)$$

Rule r_1 : *If A Then Z* ∨ Rule r_2 : *If B Then Z* is equivalent to

$$r = [A, B]^T \text{ and } {}^g @ (r) = [Z]^T,$$

where \wedge stands for the conjunction, \vee stands for the disjunction, and f and g are functions related to the conjunction and to the disjunction, respectively:

$$\begin{aligned} f : \{0,1\}^n &\rightarrow \{0,1\}, f(x_1, \dots, x_n) = \min_{i=1..n} x_i, \\ g : \{0,1\}^n &\rightarrow \{0,1\}, g(x_1, \dots, x_n) = \max_{i=1..n} x_i. \end{aligned} \quad (13)$$

The addition of signatures replaces an antecedent fact with a rule in terms of the following definition.

Definition 7. Let the two rules be

Rule r_1 : *If X ∧ Y Then Z* is equivalent to

$$r_1 = [X, Y]^T \text{ and } {}^f @ (r_1) = [Z]^T, \quad (14)$$

Rule r_2 : *If V ∧ W Then X* is equivalent to

$$r_2 = [V, W]^T \text{ and } {}^f @ (r_2) = [X]^T.$$

The replacement of an antecedent fact with these two rules is modelled as

$$r = r_1 \oplus_1 r_2 = [[V, W], Y]^T, \quad (15)$$

where r is obtained as an inference of r_1 and r_2 .

The application of Definitions 6 and 7 to expert systems modelling is exemplified as follows by a short example. Let us suppose that five rules are defined as

Rule r_1 : *If F₁ ∨ F₂ Then Z*,

Rule r_2 : *If C ∧ D ∧ E Then F₁*,

Rule r_3 : *If A ∧ B Then C*, (16)

Rule r_4 : *If H ∧ I Then F₂*,

Rule r_5 : *If J ∧ K ∧ L Then I*.

These rules can be represented as the inference chain presented in Figure 2 (a).

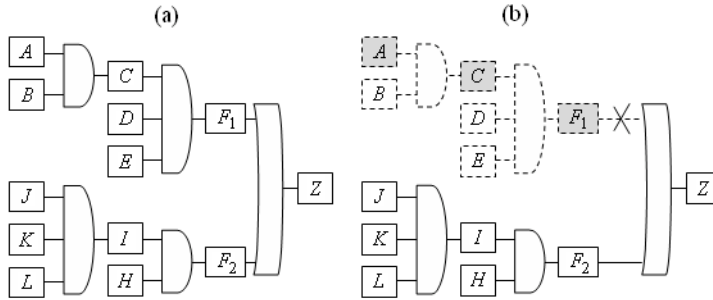


Figure 2

Inference diagram (a), which illustrates the inference chain corresponding to (30), and pruning the diagram (b)

Using the previous proposed typology it can be concluded that A, B, J, K, L, D, E and H are not inferable facts so they are observable, and that C, I, H, F_1 and F_2 are inferable facts. This aspect does not indicate that C, I, H, F_1 or F_2 are not observable. Definitions 6 and 7 lead to the following step-by-step construction of a signature:

$$\begin{aligned}
 r_1 &= [F_1, F_2]^T, \quad {}^s @ (r_1) = [Z]^T, \quad r_2 = [C, D, E]^T, \quad {}^f @ (r_2) = [F_1]^T, \\
 r_3 &= [A, B]^T, \quad {}^f @ (r_3) = [C]^T, \quad r_4 = [I, H]^T, \quad {}^f @ (r_4) = [F_2]^T, \\
 r_5 &= [J, K, L]^T, \quad {}^f @ (r_5) = [I]^T, \quad r = r_1 \oplus_1 r_2 \oplus_{1,1} r_3 \oplus_2 r_4 \oplus_{2,1} r_5 \\
 &= [[[A, B], D, E], [[J, K, L], H]]^T.
 \end{aligned} \tag{17}$$

Inferring the expert system output means

$$[Z]^T = @ (r) = {}^s @ ({}^f @_1 ({}^f @_{1,1} ({}^f @_2 ({}^f @_{2,1} (r))))), \tag{18}$$

where f and g indicate the conjunction and the disjunction, respectively.

Since in this case the function g must output not only the degree of truth but also the rule to be applied (F_1, F_2 or Φ), the following function is proposed:

$$\begin{aligned}
 g : \{0,1\}^2 &\rightarrow \{F_1, F_2, \Phi\}, \\
 g(x_i, x_j) &= \begin{cases} F_i, & \text{if } x_i = 1, x_j = 0, \\ \Phi, & \text{if } x_i = 0, x_j = 0, \\ F_i, & \text{if } x_i = 1, x_j = 1, p(i) > p(j), \end{cases} \tag{19}
 \end{aligned}$$

where $p(i)$ is the priority of rule r_i , and Φ indicates no rules to apply.

The signature presented in (17) is constructed accepting the assumption that A, B, J, K, L, D, E and H are observed. If this assumption is not accepted, i.e., at least

one of the observable facts is not observed, the signature is simplified. The pruning operation can be used in order to simplify the signatures. In such situations since the conjunction operator \wedge links all facts, we can erase all branches until we reach a disjunction \vee . Figure 2 (b) points out that A , C and F_1 are not observed, and this situation is modelled by the pruning operation:

$$r = \varnothing_1(r) = [[J, K, L], H]^T. \quad (20)$$

If C is observable and it is observed the rule r_3 is no more needed, the signature is transformed into

$$r = r_1 \oplus_1 r_2 \oplus_2 r_4 \oplus_{2,1} r_5 = [[C, D, E], [[J, K, L], H]]^T. \quad (21)$$

A conflict management strategy should be considered if F_1 , F_2 , and F_3 are inferred (observed) and each one is a possible output of the expert system: The conflict management strategy should be focused on establishing a priority order. Our approach to expert systems modelling based on signatures can manage easily this situation by including the priority order in the contraction law g , where g is defined in (19).

Our modelling approach is supported by a three-step algorithm. The first step concerns the construction of the signature of the inference engine, the second step computes the inference result represented by the expert system output, and the third step ensures the iteration. As suggested in Figure 1 (a), the algorithm uses two inputs, the knowledge base (the rules) and the database (the facts). The expert systems modelling algorithm consists of the following steps:

Step 1. Select from the knowledge base those rules which are related to the expert system goal, use equation (12) to construct the signature, and memorize the contraction law of the signature.

Step 2. Develop the signature by the one-by-one investigation of the facts contained in the signature:

- If the fact is unobservable, select all rules from the knowledge base which refer this fact as a consequence, replace them using equation (15), and memorize the contraction law,
- If the fact is observable, search the database to find out if the fact has actually been observed:
 - If yes, replace it with the observed value,
 - If not, search the database to find the rules which refer this fact as a consequence:
 - If a rule is found, replace the fact with the rule using equation (15), and memorize the contraction law,
 - If a rule is not found, prune the signature from this fact (leaf) to the first branch which supposes a disjunctive contraction.

Step 3. Continue with step 2 until all facts of the signature are replaced with data.

The flowchart of the algorithm is presented in Figure 3. This algorithm can be simplified if the rules which contain unobservable facts are identified. The idea is to compute a priori composed rules. Equation (14) is employed in such cases in order to generate a signature which can be used directly at step 2. The application of our algorithm is exemplified in the next section.

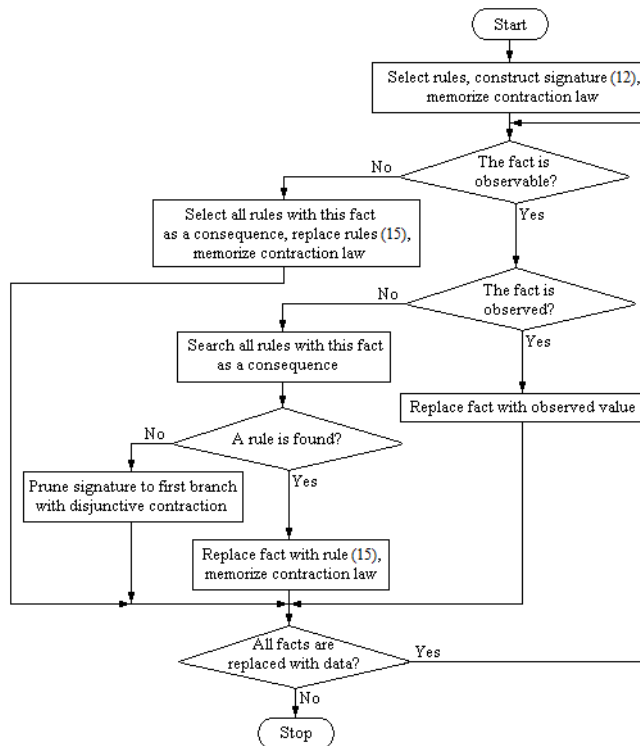


Figure 3

Flowchart of expert systems modelling algorithm

4 Case Study

This case study will apply the results presented in the previous section to derive models of a Bayesian rule-based expert system. The new models are expressed as signatures defined in Section 2.

The case study considers the signature-based modelling of an uncertain rule-based expert system represented by a Bayesian expert system. This rule-based expert

system is also taken from [25], and it predicts the weather for the next day, viz. the “tomorrow” weather (tomorrow will rain or not). Two signatures will be constructed as follows using the algorithm defined in Section 3; the first one will ignore the accumulation of evidence and the second one will consider that.

Prior to the presentation of the application of our expert systems modelling algorithm, some details on the firing of Bayesian rules are given as follows. Let us consider the rule

$$\text{Rule 1: } \textit{If } A \{LS \equiv x, LN \equiv y\} \textit{ Then } B \{\text{prior } u\}, \quad (22)$$

where $LS \equiv x$ is the likelihood of sufficiency of fact A , $LN \equiv y$ is the likelihood of necessity of fact A , and u is the prior probability of fact B . The application of Definition 6 leads to

$$r_1 = [A]^T, \quad {}^h @ (r_1) = [B]^T, \quad (23)$$

where the definition of the function h is

$$h(A) = \begin{cases} \frac{O(B|A)}{1+O(B|A)} & \textit{if } A, \\ \frac{O(B|\neg A)}{1+O(B|\neg A)} & \textit{if } \neg A, \end{cases} \quad (24)$$

$$\begin{cases} O(B|A) = x O(B), \\ O(B|\neg A) = y O(B), \end{cases} \quad O(B) = \frac{u}{1-u},$$

$O(B)$ is the prior evidence of fact B , $O(B|A)$ is the posterior evidence of fact B given the fact A (true); *and*; $h(A)$ is the posterior probability of fact B given the fact A .

The rules and signatures of this expert system are synthesized in Table 1, f is defined in (12), and the general notation $h \circ f = h(f)$ is used.

If the accumulation of evidence is ignored the application of the three steps of our expert systems modelling algorithm are first presented as follows.

Step 1. The signature is

$$r = [TwD, TwR]^T, \quad (25)$$

and the algorithm memorizes

$${}^s @ (r) = [Tw]^T, \quad (26)$$

$$Tw = g(TwD, TwR) = \begin{cases} TwD, & \textit{if } TwD > TwR, \\ TwR, & \textit{if } TwR > TwD. \end{cases}$$

Table 1
Rules, symbols and signatures of the Bayesian rule-based expert system

Nr.	Rule	Symbol	Signature
1	If [Today is rain] {LS=2.5; LN=0.6} Then [Tomorrow is rain] {prior=0.5}	[Today is rain] = TyR [Tomorrow is rain]= TwR	$r_1 = [TyR]^T$, $h_1^{of} @ (r_1) = [TwR]^T$
2	If [Today is dray] {LS=1.6; LN=0.4} Then [Tomorrow is dray] {prior=0.5}	[Today is dray] = TyD [Tomorrow is dray] = TwD	$r_2 = [TyD]^T$, $h_2^{of} @ (r_2) = [TwD]^T$
3	If [Today is rain] and [Rainfall is low]{LS=10; LN=1} Then [Tomorrow is dray] {prior=0.5}	[Today is rain] = TyR [Rainfall is low] = RaL [Tomorrow is dray] = TwD	$r_3 = [TyR, RaL]^T$, $h_3^{of} @ (r_3) = [TwD]^T$
4	If [Today is rain] and [Rainfall is low] and [Temperature is cold] {LS=1.5; LN=1} Then [Tomorrow is dray] {prior=0.5}	[Today is rain] = TyR [Rainfall is low] = RaL [Temperature is cold] = TeC [Tomorrow is dray] = TwD	$r_4 = [TyR, RaL, TeC]^T$, $h_4^{of} @ (r_4) = [TwD]^T$
5	If [Today is dray] and [Temperature is warm]{LS=2; LN=0.9} Then [Tomorrow is rain] {prior=0.5}	[Today is dray] = TyD [Temperature is warm] = TeW [Tomorrow is rain]= TwR	$r_5 = [TyD, TeW]^T$, $h_5^{of} @ (r_5) = [TwR]^T$
6	If [Today is dray] and [Temperature is warm] and [sky is overcast]{LS=5; LN=1} Then [Tomorrow is rain] {prior=0.5}	[Today is dray] = TyD [Temperature is warm] = TeW [Sky is overcast] = SyO [Tomorrow is rain] = TwR	$r_6 = [TyD, TeW, SyO]^T$, $h_6^{of} @ (r_6) = [TwR]^T$

Step 2. The signature is

$$r = \overline{\textcircled{2}}^3(\overline{\textcircled{1}}^3(r)) = [[TwD_2, TwD_3, TwD_4], [TwR_1, TwR_5, TwR_6]]^T, \quad (27)$$

and the algorithm memorizes

$$\begin{aligned} {}^g @_1(r) &= [TwD, [TwR_1, TwR_5, TwR_6]]^T(r) = [Tw]^T, \\ {}^g @_2(r) &= [[TwD_2, TwD_3, TwD_4], TwR]^T. \end{aligned} \quad (28)$$

Step 3. The results are:

$$\begin{aligned} r &= r \oplus_{1,1} r_2 \oplus_{1,2} r_3 \oplus_{1,3} r_4 \oplus_{2,1} r_1 \oplus_{2,2} r_5 \oplus_{2,3} r_6 \\ &= [[[TyD], [TyR, RaL], [TyR, RaL, TeC]], [TyR], \\ & [TyD, TeW], [TyD, TeW, SyO]]^T, \quad {}^{h_2^{\circ f}} @_{1,1}(r) = [[TwD_2, \\ & [TyR, RaL], [TyR, RaL, TeC]], [TyR], [TyD, TeW], \\ & [TyD, TeW, SyO]]^T, \quad {}^{h_3^{\circ f}} @_{1,2}(r) = [[[TyD], TwD_3, \\ & [TyR, RaL, TeC]], [TyR], [TyD, TeW], \\ & [TyD, TeW, SyO]]^T, \dots, \quad {}^{h_6^{\circ f}} @_{2,3}(r) = [[[TyD], \\ & [TyR, RaL], [TyR, RaL, TeC]], [TyR], [TyD, TeW], TwR_6]]^T. \end{aligned} \quad (29)$$

For the observations [Today is rain], [Rainfall is low], [Temperature is cold], [Sky is overcast] ($TyR=1$, $RaL=1$, $TeC=1$, $SyO=1$), the expert system output is computed in terms of:

$$\begin{aligned} r &= [[[TyD], [TyR, RaL], [TyR, RaL, TeC]], [TyR], [TyD, TeW], \\ & [TyD, TeW, SyO]]^T = [[[0], [1,1], [1,1,1]], [[1], [0,0], [0,0,1]]]^T, \\ r &= {}^f @_{1,1}({}^f @_{1,2}({}^f @_{1,3}({}^f @_{2,1}({}^f @_{2,2}({}^f @_{2,3}(r)))))) = [[[0], [1], [1]], \\ & [[1], [0], [0]]]^T, \quad r = {}^{h_2} @_{1,1}({}^{h_3} @_{1,2}({}^{h_4} @_{1,3}({}^{h_1} @_{2,1}({}^{h_5} @_{2,2}({}^{h_6} @_{2,3}(r)))))) \\ & = [[0.285, 0.909, 0.6], [0.714, 0.473, 0.5]]^T, \\ r &= {}^g @_1({}^g @_2(r)) = [0.909, 0.714]^T. \end{aligned} \quad (30)$$

This expert system response shows that the probability for “Tomorrow is Dray” is 0.909 and the probability for “Tomorrow is Rain” is 0.714. Prior to observations, the probabilities have been equal, i.e., 0.5 to 0.5.

Conclusions

This paper has given an application of signatures to expert systems modelling. Our modelling approach is backed up by a systematic modelling algorithm. The proposed approach has proved to be effective in accounting for certain observations, and the results have been generalized to uncertain observations.

The theoretical framework has been exemplified by a case study concerning a Bayesian model. We have shown how to elicit a rule base from the case study. The quality of the rules is not evaluated quantitatively, and this aspect will be considered as a future research direction of rule evaluation for confirming the effectiveness of the proposed modelling approach.

The future research will be focused on the reduction of the number of iterations which correspond to the step 2 of the algorithm. Since this case study is related to mechatronics applications but not to industrial problems, the proposed modelling results will be applied to other illustrative case studies such as those discussed in [37-46] to prove its effectiveness. More convincing mechatronics applications will be mapped onto our expert systems modelling approach.

Acknowledgement

This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS – UEFISCDI, project number PN-II-ID-PCE-2011-3-0109, and by the Romanian Ministry of Education, Research and Innovation through the PNII Idei project 842/2008.

References

- [1] P. Angelov: Fuzzily Connected Multi-Model Systems Evolving Autonomously From Data Streams, *IEEE Transactions on Systems, Man, and Cybernetics – Part B, Cybernetics*, Vol. 41, No. 4, 2011, pp. 898-910
- [2] P. Angelov, R. Yager: Density-Based averaging – A New Operator for Data Fusion, *Information Sciences*, Vol. 222, 2013, pp. 163-174
- [3] Y. Dimopoulos, M. A. Hashmi, P. Moraitis: μ -SATPLAN: Multi-Agent Planning as Satisfiability, *Knowledge-based Systems*, Vol. 29, 2012, pp. 54-62
- [4] H. H. L. C. Monte-Alto, A. B. Biasão, L. O. Teixeira, E. H. M. Huzita: Multi-Agent and Context-Aware Solutions for a Global Software Development Environment, *International Journal of Artificial Intelligence*, Vol. 11, No. A13, 2013, pp. 115-129
- [5] D. Pal, K. M. Mandana, S. Pal, D. Sarkar, C. Chakraborty: Fuzzy Expert System Approach for Coronary Artery Disease Screening Using Clinical Parameters, *Knowledge-Based Systems*, Vol. 26, 2012, pp. 162-174
- [6] D. Yazdani, B. Saman, A. Sepas-Moghaddam, F. Mohammad-Kazemi, M. Reza Meybodi: A New Algorithm Based on Improved Artificial Fish Swarm Algorithm for Data Clustering, *International Journal of Artificial Intelligence*, Vol. 11, No. A13, 2013, pp. 193-221
- [7] A. Azarian, A. Siadat, P. Martin: A New Strategy for Automotive Off-Board Diagnosis Based on a Meta-Heuristic Engine, *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 5, 2011, pp. 733-747
- [8] W. Caicedo Torres, M. Quintana, H. Pinzón: Differential Diagnosis of Hemorrhagic Fevers Using ARTMAP and an Artificial Immune System, *International Journal of Artificial Intelligence*, Vol. 11, No. A13, 2013, pp. 150-169

-
- [9] S. Szénási, Z. Vámosy: Evolutionary Algorithm for Optimizing Parameters of GPGPU-based Image Segmentation, *Acta Polytechnica Hungarica*, Vol. 10, No. 5, 2013, pp. 7-28
- [10] C. Pozna, R.-E. Precup: Aspects Concerning the Observation Process Modeling in the Framework of Cognition Process, *Acta Polytechnica Hungarica*, Vol. 9, No. 1, 2012, pp. 203-223
- [11] C. Pozna, R.-E. Precup: A General Formulation of Abduction Algorithms, *Proceedings of IEEE 3rd International Conference on Cognitive Infocommunications (CogInfoCom 2012)*, Kosice, Slovakia, 2012, pp. 573-578
- [12] P. Baranyi, A. Csapó: Cognitive Infocommunications: CogInfoCom, *Proceedings of 11th IEEE International Symposium on Computational Intelligence and Informatics (CINTI 2010)*, Budapest, Hungary, 2010, pp. 141-146
- [13] A. Csapó, P. Baranyi: A Unified Terminology for CogInfoCom Applications, *Proceedings of 2nd International Conference on Cognitive Infocommunications (CogInfoCom 2011)*, Budapest, Hungary, 2011, 6 pp.
- [14] A. Csapó, P. Baranyi: A Unified Terminology for the Structure and Semantics of CogInfoCom Channels, *Acta Polytechnica Hungarica*, Vol. 9, No. 1, 2012, pp. 85-105
- [15] M. Nakano, Y. Hasegawa, K. Funakoshi, J. Takeuchi, T. Torii, K. Nakadai, N. Kanda, K. Komatani, H. G. Okuno, H. Tsujino: A Multi-Expert Model for Dialogue and Behavior Control of Conversational Robots and Agents, *Knowledge-Based Systems*, Vol. 24, No. 2, 2011, pp. 248-256
- [16] C. A. Laurentys, C. H. M. Bomfim, B. R. Menezes, W. M. Caminhas: Design of a Pipeline Leakage Detection Using Expert System: A Novel Approach, *Applied Soft Computing*, Vol. 11, No. 1, 2011, pp. 1057-1066
- [17] B. Jankowska, Using Semantic Data Integration to Create Reliable Rule-Based Systems With Uncertainty, *Engineering Applications of Artificial Intelligence*, Vol. 24, No. 8, 2011, pp. 1499-1509
- [18] J. M. Gascueña, E. Navarro, A. Fernández-Caballero: Model-Driven Engineering Techniques for the Development of Multi-Agent Systems, *Engineering Applications of Artificial Intelligence*, Vol. 25, No. 1, 2012, pp. 159-173
- [19] L. Chang, Y. Zhou, J. Jiang, M. Li, X. Zhang: Structure Learning for Belief Rule Base Expert System: A Comparative Study, *Knowledge-Based Systems*, Vol. 39, 2012, pp. 159-172
- [20] C. Pozna, N. Minculete, R.-E. Precup, L. T. Kóczy, Á. Ballagi: Signatures: Definitions, Operators and Applications to Fuzzy Modeling, *Fuzzy Sets and Systems*, Vol. 201, 2012, pp. 86-104

-
- [21] Á. Ballagi, L. T. Kóczy, T. D. Gedeon: Robot Cooperator Without Explicit Communication by Fuzzy Signatures and Decision trees, Proceedings of Joint 2009 International Fuzzy Systems Association World Congress and 2009 European Society of Fuzzy Logic and Technology Conference (IFSA-EUSFLAT 2009), Lisbon, Portugal, 2009, pp. 1468-1473
- [22] L.T. Kóczy, T. Vámos, G. Biró: Fuzzy Signatures, Proceedings of 4th Meeting of the Euro Working Group on Fuzzy Sets and 2nd International Conference on Soft and Intelligent Computing (EUROPUSE-SIC'99), Budapest, Hungary, 1999, pp. 210-217
- [23] C. Pozna, R.-E. Precup: Results Concerning a New Pattern of Human Knowledge, Proceedings of 2nd International Conference on Cognitive Infocommunications (CogInfoCom 2011), Budapest, Hungary, 2011, 18 pp.
- [24] C. Pozna, R.-E. Precup: Novel Design of Cognitive System Strategies, Proceedings of 4th IEEE International Symposium on Logistics and Industrial Informatics (LINDI 2012), Smolenice, Slovakia, 2012, pp. 205-214
- [25] M. Negnevitsky, Artificial Intelligence, Person Education Limited, Harlow, 2011
- [26] N. Minculete: On Certain Inequalities About Arithmetic Functions Which Use Exponential Divisors, International Journal of Number Theory, Vol. 8, No. 6, 2012, pp. 1527-1535
- [27] N. Minculete: A Note About Properties of Exponential Divisors, Applied Mathematics & Information Sciences, Vol. 7, No. 1, 2013, pp. 319-322
- [28] F. C. Mitroi, N. Minculete: Mathematical Inequalities for Biparametric Extended Information Measures, Journal of Mathematical Inequalities, Vol. 7, No. 1, 2013, pp. 63-71
- [29] S. Preitl, R.-E. Precup: On the Algorithmic Design of a Class of Control Systems Based on Providing the Symmetry of Open-Loop Bode Plots, Scientific Bulletin of "Politehnica" University of Timisoara, Romania, Transactions on Automatic Control and Computer Science, Vol. 41 (55), No. 1-2, 1996, pp. 47-55
- [30] S. Preitl, R.-E. Precup: Introducere în conducerea fuzzy a proceselor, Editura Tehnica, Bucharest, 1997
- [31] Z. C. Johanyák, S. Kovács: Polar-Cut Based Fuzzy Model for Petrophysical Properties Prediction, Scientific Bulletin of "Politehnica" University of Timisoara, Romania, Transactions on Automatic Control and Computer Science, Vol. 57 (67), No. 4, 2008, pp. 195-200
- [32] R.-E. Precup, S. Doboli, S. Preitl: Stability Analysis and Development of a Class of Fuzzy Control Systems, Engineering Applications of Artificial Intelligence, Vol. 13, No. 3, 2000, pp. 237-247

-
- [33] J. Vaščák, M. Rutrich: Path Planning in Dynamic Environment Using Fuzzy Cognitive Maps, Proceedings of 6th International Symposium on Applied Machine Intelligence and Informatics (SAMI 2008), Herľany, Slovakia, 2008, pp. 5-9
- [34] E. S. Nicoară, F.-G. Filip, N. Paraschiv: Simulation-based Optimization Using Genetic Algorithms for Multi-objective Flexible JSSP, Studies in Informatics and Control, Vol. 20, No. 4, 2011, pp. 333-344
- [35] A. Widyotriatmo, A. K. Pamosoaji, K.-S. Hong: Control Architecture of an Autonomous Material Handling Vehicle, International Journal of Artificial Intelligence, Vol. 10, No. S13, 2013, pp. 139-153
- [36] P. Melin, L. Astudillo, O. Castillo, F. Valdez, M. Garcia: Optimal Design of Type-2 and Type-1 Fuzzy Tracking Controllers for Autonomous Mobile Robots Under Perturbed Torques Using a New Chemical Optimization Paradigm, Expert Systems with Applications, Vol. 40, No. 8, 2013, pp. 3185-3195
- [37] R.-E. Precup, S. Preitl: Popov-Type Stability Analysis Method for Fuzzy Control Systems, Proceedings of Fifth EUFIT'97 European Congress, Aachen, Germany, 1997, pp. 1306-1310
- [38] F. Previdi, T. Schauer, S. M. Savaresi, K. J. Hunt: Data-Driven Control Design for Neuroprotheses: A Virtual Reference Feedback Tuning (VRFT) Approach, IEEE Transactions on Control Systems Technology, Vol. 12, No. 1, 2004, pp. 176-182
- [39] S. Preitl, R.-E. Precup, J. Fodor, B. Bede: Iterative Feedback Tuning in Fuzzy Control Systems. Theory and applications, Acta Polytechnica Hungarica, Vol. 3, No. 3, 2006, pp. 81-96
- [40] R.-E. Precup, M. L. Tomescu, S. Preitl: Lorenz System Stabilization Using Fuzzy Controllers, International Journal of Computers, Communications & Control, Vol. II, No. 3, 2007, pp. 279-287
- [41] G. Hermann, G. Benkó, A. Nagy: Retrofitting a Length Measuring Machine for Linear Scale Calibration, Proceedings of 13th International Conference on Intelligent Engineering Systems (INES 2009), Barbados, 2009, pp. 79-82
- [42] G. Klančar, D. Matko, S. Blažič: A Control Strategy for Platoons of Differential-Drive Wheeled Mobile Robot, Robotics and Autonomous Systems, Vol. 59, No. 2, 2011, pp. 57-64
- [43] M. Oprea: INTELEnvQ-Air: An Intelligent System for Air Quality Analysis in Urban Regions, International Journal of Artificial Intelligence, Vol. 9, No. A12, 2012, pp. 106-122

- [44] L. Horváth, I. J. Rudas: Active Knowledge for the Situation-driven Control of Product Definition, Acta Polytechnica Hungarica, Vol. 10, No. 2, 2013, pp. 217-234
- [45] K. Lamár, J. Neszveda: Average Probability of Failure of Aperiodically Operated Devices, Acta Polytechnica Hungarica, Vol. 10. No. 8, 2013, pp. 153-167
- [46] I. Mundi, M. M. E. Alemany, A. Boza, R. Póler: A Model-Driven Decision Support System for the Master Planning of Ceramic Supply Chains with Non-uniformity of Finished Goods, Studies in Informatics and Control, Vol. 22, No. 2, 2013, pp. 153-162