Applications of Signatures to Expert Systems Modelling

Claudiu Pozna\textsuperscript{1,2}, Radu-Emil Precup\textsuperscript{3}

\textsuperscript{1}Department of Informatics, Széchenyi István University, Egyetem tér 1, H-9026 Győr, Hungary
\textsuperscript{2}Department of Automation and Information Technology, Transilvania University of Brasov, Str. Mihai Viteazu 5, RO-500174 Brasov, Romania
E-mail: cp@unitbv.ro
\textsuperscript{3}Department 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
\[ S^{(n)} = \prod_{i=1}^{n} S_i, \quad S_i = \mathbb{R} \text{ or } S_i = S^{(m)}, \quad m \geq 1, \quad i = 1 \ldots n, \] (1)

where \( \mathbb{R} \) 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 \( A : X \to S^{(n)} \), 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

\[
A(x) = \begin{bmatrix}
\ldots \\
 a_i \\
 a_{i+1,1} \\
 a_{i+1,2} \\
 a_{i+2,1} \\
 a_{i+2,2,1} \\
 a_{i+2,2,2} \\
 \ldots 
\end{bmatrix},
\]

\[
A^T(x) = \begin{bmatrix}
\ldots \\
 a_i [a_{i+1,1} \ a_{i+1,2}] [a_{i+2,1} \ a_{i+2,2,1} \ a_{i+2,2,2}] \ldots 
\end{bmatrix}.
\]

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

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

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

\[
S_i = \prod_{i=1}^{m} \mathbb{R} = \mathbb{R}^m.
\]
\( f @ : S \to S, \quad f @ (a^{1,2,\ldots,n}) = a^1 = [a], \)
\( f @ : S \to S, \quad f @ (a^{1,\ldots,m}) = a^{1,\ldots,n}, \quad \text{if } i \leq n, \)
\( f @ (a^{-}) = a^{-}, \quad \text{otherwise}, \)
\( f @ : S \to S, \quad f @ (a^{1,\ldots,m}) = a^{1,\ldots,n}, \quad \text{if } i \leq n, j \leq m, k \leq r, \)
\( f @ (a^{-}) = a^{-}, \quad \text{otherwise}, \)
\( f @ : S \to S, \quad f @ (a^{1,\ldots,m}) = a^{1,\ldots,n}, \quad \text{if } i \leq n, \quad \text{otherwise}, \)

where \( a = f(a_1, a_2, \ldots, a_n), \quad f : \mathbb{R}^n \to \mathbb{R}, \quad a_i = f(a_{i1}, \ldots, a_{im}), \quad f : \mathbb{R}^m \to \mathbb{R}, \) and \( a_{i,j,k} = f(a_{i,j,k1}, \ldots, a_{i,j,ikq}), \quad f : \mathbb{R}^q \to \mathbb{R}. \) 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):
\[ |f @ (a^{1,2,\ldots,n})| = a. \]

**Definition 3.** The extension of a signature is defined as one of the functions
\( g \overline{\quad} @ ^{p} : S \to S, \)
\( g \overline{\quad} @ ^{q} (a^{1,2,\ldots,n}) = a^{1,2,\ldots,\{1,\ldots,q\},\ldots,n}, \quad \text{if } i \leq n, \)
\( g \overline{\quad} @ ^{q} (a^{-}) = a^{-}, \quad \text{otherwise}, \)
\( g \overline{\quad} @ ^{p} : S \to S, \)

where two forms of the function \( g \) are used, \( g : \mathbb{R} \to \mathbb{R}^p, \quad g(a_i) = [a_{i1}, \ldots, a_{ip}], \)
\( g : \mathbb{R} \to \mathbb{R}^q, \quad g(a_{i,j}) = [a_{i,j,k1}, \ldots, a_{i,j,ikq}], \) 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
\( g \overline{\quad} @ ^{p} : S \to S, \)
\( g \overline{\quad} @ (a^{1,2,\ldots,n}) = g \overline{\quad} @ ^{p} (g \overline{\quad} @ ^{q} (g \overline{\quad} @ ^{p} (g \overline{\quad} @ ^{q} (a^{1,2,\ldots,n})))) = a^{1,\ldots,p}[1,\ldots,p][1,\ldots,p], \)

where \( g : \mathbb{R} \to \mathbb{R}^p, \quad g(a_j) = [a_{j1}, \ldots, a_{jp}], \quad j = 1 \ldots n. \)
Definition 4. The pruning of a signature is defined as one of the three functions
\[ \emptyset_i : S \rightarrow S, \]
\[ \emptyset_i(a_{1,2,...,i,...,n}) = a_{1,2,...,i-1,i+1,...,n}, \quad \text{if } i \leq n, \]
\[ \emptyset_i(a^-) = a^-, \quad \text{otherwise}, \]
\[ \emptyset_j : S \rightarrow S, \]
\[ \emptyset_j(a_{1,...,m},...n) = a_{1,...,i-1,i+1,...,n}, \quad \text{if } i \leq n, \]
\[ \emptyset_j(a^-) = a^-, \quad \text{otherwise}, \]
\[ \emptyset_{i,j,k} : S \rightarrow S, \]
\[ \emptyset_{i,j,k}(a_{1,...,l,...,r},...m) = a_{1,...,i-1,i+1,...,n}, \quad \text{if } i \leq n, j \leq m, k \leq r, \]
\[ \emptyset_{i,j,k}(a^-) = a^-, \quad \text{otherwise}. \]

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

Definition 5. The addition of two signatures is defined as the function
\[ f \oplus_i : S \times S \rightarrow S, \]
\[ f \oplus_i(a_{1,2,...,l,...,r},b_{1,2,...,m}) = c_{1,2,...,l,...,r}, \]
where \( c_k = a_k \quad \forall \ k \neq i, \ c_j = f(a_{j},b_{j}) \quad \forall \ j = 1..m, \ f : \mathbb{R}^2 \rightarrow \mathbb{R}. \]
If \( a^{1,2,...,n}, b^{1,2,...,m} \in S \), then
\[ a^{1,2,...,n} \oplus b^{1,2,...,m} = \emptyset_i(a_{1,2,...,n}) \oplus b_{1,2,...,m}, \]
\[ = c_{1,2,...,l,...,r}, \quad \text{if } i \leq n, \]
\[ c^- \oplus_i b^- = a^-, \quad \text{otherwise}, \]
where \( c_{ij} = f(a_{ij},b_{ij}) \), \( [a_{1},...,a_{m}] = g(a_{j}), \ j = 1..m, \ c_k = a_k \quad \forall \ k \neq i. \)

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.

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.

![Figure 1](image_url)

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

\[
\text{Rule 1: If } Y \text{ Then } Z, \\
\text{Rule 2: If } X \text{ Then } Y, \\
\]

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:

\[
\text{If (If } X \text{ Then } Y \text{) Then } Z, \\
\]

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 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 modeled by signatures in one of the three forms

Rule \( r_1 : \text{If } A \text{ Then } Z \) is equivalent to \( r_1 = [A]^T \) and \( @ (r_1) = [Z]^T \),

Rule \( r_1 : \text{If } A \land B \land C \text{ Then } Z \) is equivalent to

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

Rule \( r_1 : \text{If } A \text{ Then } Z \lor \text{Rule } r_2 : \text{If } B \text{ Then } Z \) is equivalent to

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

where \( \land \) stands for the conjunction, \( \lor \) stands for the disjunction, and \( f \) and \( g \) are functions related to the conjunction and to the disjunction, respectively:

\[
f : \{0,1\}^n \to \{0,1\}, \quad f(x_1, \ldots, x_n) = \min_{i=1 \ldots n} x_i,
\]

\[
g : \{0,1\}^n \to \{0,1\}, \quad g(x_1, \ldots, x_n) = \max_{i=1 \ldots n} x_i.
\]

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 \( r_1 : \text{If } X \land Y \text{ Then } Z \) is equivalent to

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

Rule \( r_2 : \text{If } V \land W \text{ Then } X \) is equivalent to

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

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

\[
r = r_1 \oplus r_2 = [(V, W), Y]^T,
\]

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

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

Rule \( r_1 : \text{If } F_1 \lor F_2 \text{ Then } Z \),

Rule \( r_2 : \text{If } C \land D \land E \text{ Then } F_1 \),

Rule \( r_3 : \text{If } A \land B \text{ Then } C \),

Rule \( r_4 : \text{If } H \land I \text{ Then } F_2 \),

Rule \( r_5 : \text{If } J \land K \land L \text{ Then } I \).

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

$$
\begin{align*}
  r_1 &= [F_1, F_2]^T, & \gamma @ (r_1) &= [Z]^T, & r_2 &= [C, D, E]^T, & \mathcal{I} @ (r_2) &= [F_1]^T, \\
  r_3 &= [A, B]^T, & \mathcal{I} @ (r_3) &= [C]^T, & r_4 &= [I, H]^T, & \mathcal{I} @ (r_4) &= [F_2]^T, \\
  r_5 &= [J, K, L]^T, & \mathcal{I} @ (r_5) &= [I]^T, & r &= r_1 \oplus_1 r_2 \oplus_1 r_3 \oplus_2 r_4 \oplus_2 r_5 \\
  &= [[[A, B], D, E], [[J, K, L], H]]^T.
\end{align*}
$$

Inferring the expert system output means

$$
[Z]^T = @ (r) = \mathcal{S} @ (\mathcal{I} @ (\mathcal{I} @ (\mathcal{I} @ (\mathcal{I} @ (\mathcal{I} @ r))))),
$$

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 $\Phi$), the following function is proposed:

$$
g : \{0,1\}^2 \rightarrow \{F_1, F_2, \Phi\},
g(x_1, x_2) = \begin{cases} 
  F_i, & \text{if } x_i = 1, \ x_j = 0, \\
  \Phi, & \text{if } x_i = 0, \ x_j = 0, \\
  F_i, & \text{if } x_1 = 1, \ x_2 = 1, \ p(i) > p(j),
\end{cases}
$$

where $p(i)$ is the priority of rule $r_i$, and $\Phi$ 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 \( \land \) links all facts, we can erase all branches until we reach a disjunction \( \lor \). Figure 2 (b) points out that \( A, C \) and \( F_1 \) are not observed, and this situation is modelled by the pruning operation:

\[
\mathbf{r} = \emptyset_1(\mathbf{r}) = [[J, K, L], H]]^T. \tag{20}
\]

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

\[
\mathbf{r} = \mathbf{r}_1 \oplus_1 \mathbf{r}_2 \oplus_2 \mathbf{r}_4 \oplus_2 \mathbf{r}_5 = [[C, D, E], [[J, K, L], H]]^T. \tag{21}
\]

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

\[ \text{Rule 1: If } A \{ \text{LS} \equiv x, \text{LN} \equiv y \} \text{ Then } B \{ \text{prior} u \}, \]  

(22)

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

\[ r_i = [A]^T, \quad h@(r_i) = [B]^T, \]  

(23)

where the definition of the function \( h \) is

\[
h(A) = \begin{cases} 
\frac{O(B \mid A)}{1 + O(B \mid A) + O(B \mid \neg A)} & \text{if } A, \\
\frac{O(B \mid \neg A)}{1 + O(B \mid \neg A)} & \text{if } \neg A, 
\end{cases}
\]  

(24)

\[ O(B) = \frac{u}{1-u}, \]

\[ O(B \mid A) = x O(B), \quad O(B \mid \neg A) = y O(B), \]

\[ O(B) \] is the prior evidence of fact \( B \), \( O(B \mid A) \) 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.

\[ \text{Step 1. The signature is} \]

\[ r = [T_{wD}, T_{wR}]^T, \]  

(25)

and the algorithm memorizes

\[ i \circ @(r) = [T_{w}]^T, \]

\[ T_w = g(T_{wD}, T_{wR}) = \begin{cases} 
T_{wD}, & \text{if } T_{wD} > T_{wR}, \\
T_{wR}, & \text{if } T_{wR} > T_{wD}. 
\end{cases} \]  

(26)
Table 1
Rules, symbols and signatures of the Bayesian rule-based expert system

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

Step 2. The signature is

$$r = \circ_2 (\circ_1 (r)) = [[TwD_2, TwD_3, TwD_4], [TwR_1, TwR_3, TwR_6]]^T,$$  \hspace{1cm} (27)

and the algorithm memorizes
\[ e_1(r) = [TwD_1, TwR_1, TwR_5, TwR_6]^T (r) = [Tw]^T, \]
\[ e_2(r) = [(TwD_2, TwD_3, TwD_4), TwR]^T. \]

**Step 3.** The results are:
\[
\begin{align*}
  r &= r \oplus r_2 \oplus r_3 \oplus r_4 \oplus r_5 \oplus r_6 \\
  &= [[TyD], [TyR, RaL], [TyR, RaL, TeC]], [[TyR], [TyD, TeW]], \\
  &\quad [TyD, TeW, [TyD, TeW, SyO]]^T, \quad h_{e-f} @_{1,1}(r) = [[TwD_2], \\
  &\quad [TyR, RaL], [TyR, RaL, TeC]], [[TyR], [TyD, TeW]], \\
  &\quad [TyD, TeW, SyO]]^T, \quad h_{e-f} @_{1,2}(r) = [[[TyD], TwD_3], \\
  &\quad [TyR, RaL, TeC]], [[TyR], [TyD, TeW]], \\
  &\quad [TyD, TeW, SyO]]^T, \ldots, \quad h_{e-f} @_{2,3}(r) = [[[TyD], \\
  &\quad [TyR, RaL], [TyR, RaL, TeC]], [[TyR], [TyD, TeW], TwR_6]]^T.
\end{align*}
\]

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:
\[
\begin{align*}
  r &= [[TyD], [TyR, RaL], [TyR, RaL, TeC]], [[TyR], [TyD, TeW], \\
  &\quad [TyD, TeW, SyO]]^T = [[[0], [1,1], [1,1,1]], [[1,0,0], [0,0,1]]]^T, \\
  r &= h @_{1,1} (h @_{1,2} (h @_{1,3} (h @_{2,1} (h @_{2,2} (h @_{2,3} (r)))))) = [[[0], [1], [1]], \\
  &\quad [[1], [0], [0]]]^T, \quad r = h @_{1,1} (h @_{1,2} (h @_{1,3} (h @_{2,1} (h @_{2,2} (h @_{2,3} (r)))))) \\
  &= [[0.285, 0.909, 0.6], [0.714, 0.473, 0.5]]^T, \\
  r &= e @_{1} (e @_{2} (r)) = [0.909, 0.714]^T.
\end{align*}
\]

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


