O P E R A T I O N S R E S E A R C H A N D D E C I S I O N S
No. 3 2015
DOI: 10.5277/ord150304

Andrew SCHUMANN[1]
Jan WOLEŃSKI[2]

DECISIONS INVOLVING DATABASES, FUZZY DATABASES AND CODATABASES

The authors consider the following three ways of decision making: (i) decisions involving databases by means of standard tools of sequential logic and universal algebra; (ii) decisions involving fuzzy databases by means of fuzzy logic; (iii) decision involving continuously growing databases (codatabases) using the tools of Bayesian networks.

Keywords: *rough sets, fuzzy sets, decision making, labelled transition systems, coalgebras, Bayesian networks*

1. Introduction

A typical model of a decision process **Π** (see for example [4, p. 1–2], [11, p. 6–8], [1, p. 2], [9, p. 17–40]) identifies it using the ordered quintuple

(*) (**P**, **S**, **D**, **R, F**)

where **P** refers to the decision agent, **S** is the set of possible states of the world, **D** is the set of possible (alternative) actions (undertaken by **P**) on **S**, $\mathbf{R} \subset \mathbf{S}$ is the set of possible results following from the actions **D**, **F** is a utility function taking values in [0, 1] and whose argument is in **S**. Consequently, the subject **P** undertakes a decision (solves the decision problem) using a mapping **D** from a subset X of **S** into the set of results **R**

[1]University of Information Technology and Management in Rzeszów, ul. Sucharskiego 2, 35-225 Rzeszów, Poland, e-mail: andrew.schumann@gmail.com
[2]Jagiellonian University in Kraków, ul. Gołębia 24, 31-007 Kraków, Poland,
e-mail: wolenski@if.uj.edu.pl

taking into account the utility function **F** on **S**. In other words, **P** wants to achieve a result from a repertoire of possible outcomes using the action **D** based on the utility function **F**. Although each element of **Π** deserves further elaboration, we, following current decision theory, take (*) as the standard decomposition of a decision process. Clearly, the actual undertaking of decisions does not conform exactly to **Π**. For instance, (*) suggests that the decision process is perfectly discrete. However, particular states of Π may be difficult to precisely separate and the entire state space appears to be continuous. On the other hand, it usually happens that decision makers discretize their action space, for instance, to calculate, intuitively or mathematically, values of the utility function.

The set of possible states of the world **S** is represented as a database containing data and some ideas of how to process this data based on the set of available actions **D** to achieve the desired results **R**. As a consequence, the decision process **Π** proceeds by performing a finite comprehensible series of actions **D** on **S** in a way that the solution **R** can be reached by completing an appropriate algorithm, for instance, by implementing a sequential logic structure (IF/THEN/ELSE instructions) to apply a set of instructions in sequence from the top to the bottom of the algorithm.

Typically, decisions are divided into three groups: decisions under certainty (**P** knows what will happen in the world and **S** is well-structured) and decisions under uncertainty (**P** does not know what will happen and **S** contains uncertain items) corresponding to the reasoning of fuzzy logic [2, 5]. The latter category is particularly important, because most decisions in daily life and economics (perhaps the most important areas of practical human activities) involve decisions under uncertainty.

In this paper, we propose another possible model of decision making, which assumes that **P** does not know what will happen, because **S** grows rapidly. In this case, IF/THEN/ELSE instructions have no sense. For instance, labeled transition systems are continuously growing structures which can only be presented as databases in a static form. So, we could only apply IF/THEN/ELSE instructions in this static form. But these data sets are expanding. Notice that continuously growing structures (trees, graphs, sets) are mathematically understood as non-well-founded sets [14] or coalgebras [15]. We introduce the concept of Bayesian networks on growing structures.

Let us consider an example of a growing structure that is called the game of two brokers. Two brokers at a stock exchange have appropriate expert systems which are used to support decision making. The network administrator illegally copied both expert systems and sold to each broker the expert system of his opponent. Then he tries to sell each of them the following information: *Your opponent has your expert system*. Then the administrator tries to sell the information: *Your opponent knows that you have his expert system*, etc. How should brokers use the information received from the administrator and what information at a given iteration is essential? Thus, we must make a decision based on an infinite hierarchy of decisions.

In Section 2, we show how we use logical rules to make decisions involving databases and fuzzy databases and the way we can make decisions involving a growing structure describing the hierarchy of (fuzzy) decisions. In Section 3, we show how we can make decisions involving data regarding transition systems using the tools of Bayesian networks.

2. Hierarchies of decisions involving databases and fuzzy databases

A sequential logic structure is based on IF/THEN/ELSE instructions, according to which if a condition X is true (in **S**), then we execute a set of instructions $\mathbf{D}_1$, or else another set of instructions $\mathbf{D}_2$ are processed (for example, we execute the false instructions when the resultant of the condition is false). Conditions involving the state of the world **S** and resultants **R** in a sequential logic structure may only have the following forms: superpositions of logical operators (AND, OR, and NOT); expressions using relational operators (such as greater than or less than); variables which have the values true or false; combinations of logical, relational, and mathematical operators.

Due to the sequential logic structure modelling $\mathbf{\Pi}$, we can present sets of instructions **D** as decision trees, where nodes are conditions or resultants and edges are implications between conditions and resultants. Implications are treated in the Boolean way: (i) when 'if A, then B' is true, then A is a subset of B; (ii) when 'if A, then B' is false, then 'if non-B, then non-A' is true.

A decision will thus be a choice from multiple alternatives made with a fair degree of rationality. Let A be a finite set of possible alternatives $\mathbf{A} = \{a_1, a_2, a_3, \ldots, a_n\}$ from a database **S** and $\{g_1(\cdot), g_2(\cdot), g_3(\cdot), \ldots, g_n(\cdot)\}$ be a set of evaluation criteria for **F**. Value patterns used to compare alternatives such as 'better than', 'worse than', 'equally good', 'equal in value to', 'at least as good as', etc., are represented as binary relations which are called preference relations. So we can introduce the notion 'better than' $(>_{g_i})$ for each g_i, where $i = 1, \ldots, n$, to denote a strong preference according to g_i, the notion 'equal in value to' $(\equiv_{g_i})$ for each g_i, to denote indifference according to g_i, and the notion 'at least as good as' $(\geq_{g_i})$ for each g_i, to denote a weak preference according to g_i.

Let us notice that strong preference $>_{g_i}$, indifference $\equiv_{g_i}$, and weak preference $\geq_{g_i}$ are transitive:

$$\text{if } A >_{g_i} B \text{ and } B >_{g_i} C, \text{ then } A >_{g_i} C$$

$$\text{if } A \equiv_{g_i} B \text{ and } B \equiv_{g_i} C, \text{ then } A \equiv_{g_i} C$$

$$\text{if } A \geq_{g_i} B \text{ and } B \geq_{g_i} C, \text{ then } A \geq_{g_i} C$$

Weak preference is acyclic:

$$A \equiv_{g_i} B \text{ if and only if } A \geq_{g_i} B \text{ and } B \geq_{g_i} A$$

Strong preference is also acyclic:

$$\text{if neither } A >_{g_i} B \text{ nor } B >_{g_i} A, \text{ then } A \equiv_{g_i} B$$

$$A >_{g_i} B \text{ if and only if } A \geq_{g_i} B \text{ and not } B \geq_{g_i} A$$

Weak preference is reflexive:

$$A \geq_{g_i} A$$

Indifference is symmetric:

$$\text{if } A \equiv_{g_i} B, \text{ then } B \equiv_{g_i} A$$

Strong preference is asymmetric:

$$\text{if } A >_{g_i} B, \text{ then not } B >_{g_i} A$$

In conventional decision theory, it is assumed as well that each weak preference relation $\geq_{g_i}$ satisfies the formal property of completeness: the relation $\geq_{g_i}$ is complete if and only if for any elements A and B, at least one of $A \geq_{g_i} B$ or $B \geq_{g_i} A$ holds. Note that the binary relations 'better than' and 'worse than' are not quite symmetrical from the psychological point of view: 'A is better than B' is not exactly the same in our perceptions as 'B is worse than A'. For example, suppose a manager discusses the abilities of two employees. If he says 'the second employee is better than the first employee', he may be satisfied with both of them, but if he says ' the second employee is worse than the first employee', then he probably wants to dismiss them both from their jobs.

The preference relations $>_{g_i}$, $\equiv_{g_i}$, $\geq_{g_i}$ are a good basis for ordering a database **S**. Binary relations by which we can order entities within a database can be also understood in terms of utility relations ('gives more profit than', 'of equal profit', etc.), loss rela-

tions ('causes more loss than', 'of equal loss', etc.), and so on. A database ordered according to preference relations, utility relations, etc., may be presented as an algebraic system [10].

If a database can be represented as an algebraic system, then we can use conventional logics (e.g. classical logic) to make decisions: consider an ordered set (e.g. inductive set), then we can interpret logical operations as follows:

- the implication $b \Rightarrow a$ is true if and only if $a \geq b$ (e.g. $a \geq_{g_i} b$);
- the negation $\neg a$ is true if and only if $a \Rightarrow 0$ is true, where 0 is a minimal member of the database;
- the disjunction $a \vee b = c$ is true if and only if c is a minimal member such that $c \geq a$ and $c \geq b$;
- the conjunction $a \wedge b = c$ is true if and only if c is a maximal member such that $a \geq c$ and $b \geq c$.

Note that $a \vee b = \neg a \Rightarrow b$ and $a \wedge b = \neg(\neg a \vee \neg b)$.

Nevertheless, in most cases we deal with uncertainty in data and cannot define sets precisely. But representing databases in the form of algebraic systems is still possible. On the one hand, in the case of uncertain entities, we appeal to bounded rationality that captures the fact that rational choices are constrained by the limits of knowledge and cognitive capability. On the other hand, we can generalize logical (algebraic) operations to operations on uncertain entities as well.

There exist a lot of notions which are uncertain (imprecisely defined), such as 'being young', 'being tall', 'being healthy', 'being bald', etc. Fuzzy set theory and fuzzy logic reflect the fuzzy concepts and reasoning in which such items occur. A fuzzy set (sometimes called 'rough set' [8]) A such as 'young people' is defined by its membership function μ_A that takes values in the interval of real numbers [0, 1] which indicate the degree of membership as to how imprecise elements x belong to A. If $\mu_A(x) = 1$, then it means that x certainly belongs to A. If $\mu_A(x) = 0$, then x certainly does not belong to A, and if $0 < \mu_A(x) < 1$, then x only partially belongs to A. We can define the following logical operations on a fuzzy set (see [5] for an elementary account of fuzzy logic, [3] for an advanced treatment including metalogical studies and [2] for a broad overall view):

A is a subset of $B := \mu_A(x) \leq \mu_B(x)$;

A and $B := \min(\mu_A(x), \mu_B(x))$;

A or $B := \max(\mu_A(x), \mu_B(x))$;

non-$A := 1 - \mu_A(x)$.

Note that fuzzy sets differ from probabilistic sets. For instance, assume that there are two water bottles A and B. Let bottle A belong to the set of water for drinking with the membership function of 0.9 and bottle B belong to the set of water for drinking with probability equal to 90%. Which bottle is preferable for drinking? A is certainly not a good choice, because it is not for drinking at 90%. B may be a good choice or not at 90%.

Hence, in everyday decisions we very often refer to fuzzy IF/THEN/ELSE reasoning such as:

If a client's profit is 'big', then his credit rating is 'good'

The terms 'big' and 'good' are fuzzy. For example, we can suppose that a 'big profit' means a profit of more than 5%. Let us consider another example. Assume that a trader defines a trading rule by means of a long position when the slope defining a trend is greater than or equal to a certain value x and volatility is less than or equal to a certain value y. Such a rule can be interpreted as follows: If the slope $\geq x$ and the volatility $\leq y$, then the long position should be equal to z, where x, y, z are the parameters defined by the trader. The following is a fuzzy version of the same rule: If the slope is large and positive and the volatility is low, then the trading position is long.

Fuzzy databases can be modelled by the ordered set (α, T, X, G, M), where $\alpha$is a linguistic variable, T is a set of its meanings (terms) representing all the names of the fuzzy variable α which are defined on the set X, G is a set of syntactic rules on T, allowing, in particular, to generate new terms (meanings of α); M is a set of semantic rules, allowing to refer each new term to meanings of the fuzzy variable α. For example, to define the meaning of income, we can introduce the notions 'small', 'average', and 'big' income. Let the minimal income be equal to \$2000 and maximal to \$10,000. Now we can define a fuzzy database for the linguistic variable 'income' within the ordered system $\langle \alpha, \mathrm{T}, \mathrm{X}, \mathrm{G}, \mathrm{M} \rangle$, where α is income; T = {'small income', 'average income', 'big income'}; X = [\$2000, \$10,000]; G is a set of syntactic rules for generating new terms by means of the connectives 'and', 'or', 'not', 'very', etc., e.g. 'small *or* average income', '*very* big income', etc.; M is a set of semantic rules mapping the fuzzy subsets 'small income', 'average income', 'big income', as well as their logical superpositions into the set X = [\$2000; \$10,000]. Note that the ordered set $\langle \alpha, \mathrm{T}, \mathrm{X}, \mathrm{G}, \mathrm{M} \rangle$ is a kind of database whose data can also be presented as an inductive set. This means that we can develop a conventional logic, which is called fuzzy logic, for imprecise data. Using fuzzy logic we can also make deductive decisions.

Thus, the IF/THEN/ELSE instructions for decisions in **S** can cover (i) precisely defined items, where we can deal with **S** represented as an algebraic system, or (ii) imprecisely defined items, where we can deal with **S** represented as a fuzzy database.

Now, let us construct an infinite hierarchy of (fuzzy) decisions in accordance with labels $l = 0, 1, 2, \ldots$ which mean that a decision with label i is more important than a decision with label j if and only if $i > j$.

Let us consider **S** as a sequence of ensembles S_l labelled by l (importance in the hierarchy) and having volumes card(S_l), $l = 0, 1, 2, \ldots$ Let $S = \prod_{j=0}^{\infty} S_j$. We may imagine the ensemble S as being the population of a tower $T = T_S$ which has an infinite number

of floors with the population of the j-th floor being S_j. Set $T_k = S = \prod_{j=0}^{\infty} S_j \times \prod_{m=k+1}^{\infty} \varnothing_m$. This is the population of the first $k + 1$ floors.

The cardinality of T_k is defined as follows: card(T_k) := (card(T_1), card(T_2), card(T_3), …), i.e. it is a stream of cardinal numbers. The cardinality of S is defined thus: card(S) := $\lim_{k\to\infty} \text{card}(T_k)$. Arithmetic operations involving the numbers card(A), card(B), such that $A \subseteq S$ and $B \subseteq S$, are calculated digit by digit: card(A) ♣ card(B) := (card(A_1) ♣ card(B_1), card(A_2) ♣ card(B_2), card(A_3) ♣ card(B_3), …), where ♣ ∈ {+, –, ·, /, inf, sup}.

Let $A \subset S$. We define the probability of A by the standard proportional relation:

$$P(A) := P_S(A) = \text{card}(A \cap S)/\text{card}(S)$$

So $P(S) = \mathbf{1}$ and $P(\varnothing) = \mathbf{0}$, where $\mathbf{1}$ = (1, 1, 1, …), $\mathbf{0}$ = (0, 0, 0, …). If $A \subseteq S$ and $B \subseteq S$ are disjoint, i.e. inf($P(A)$, $P(B)$) = $\mathbf{0}$, then $P(A \cup B) = P(A) + P(B)$. Otherwise, $P(A \cup B) = \sup(P(A), P(B))$. $P(\neg A) = \mathbf{1} - P(A)$ for all $A \subseteq S$, where $\neg A = S - A$.

Relative probability functions $P(A|B)$ are defined as follows:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

where $P(B) \neq 0$ and $P(A \cap B) = \inf(P(A), P(B))$.

Let At be a finite nonempty set of attributes which express the properties of $s \in S_l$, V_a is a nonempty set of values $v \in V_a$ for $a \in At$, $I_a : S_l \to V_a$ is an information function that maps an object in S_l to a value of $v \in V_a$ for an attribute $a \in At$. Now we consider all the Boolean compositions of atomic formulas $(a, v)_l$. The meaning $\| \Phi_l \|_S$ of formulas Φ_l in our language is defined in the following way:

$$\| (a,v)_l \|_S = \{s \in S_l : I_a(s) = v\}, \quad a \in At,\ v \in V_a$$

$$\| \Phi_l \vee \Psi_l \|_S = \| \Phi_l \|_S \cup \| \Psi_l \|_S$$

$$\| \Phi_l \wedge \Psi_l \|_S = \| \Phi_l \|_S \cap \| \Psi_l \|_S$$

$$\| \neg \Phi_l \|_S = S_l - \| \Phi_l \|_S$$

A finite hierarchy of (fuzzy) decisions is understood as follows:

$$\Phi^k := (\Phi_1, \Phi_2, ..., \Phi_k)$$

with meaning:

$$\| \Phi^k \| := (\| \Phi_1 \|, \| \Phi_2 \|, ..., \| \Phi_k \|)$$

An infinite hierarchy of (fuzzy) decisions is understood as follows:

$$\Phi_\infty := \lim_{l \to \infty} (\Phi_1, \Phi_2, ..., \Phi_l, ...)$$

with meaning:

$$\| \Phi_\infty \| := \lim_{l \to \infty} (\| \Phi_1 \|, \| \Phi_2 \|, ..., \| \Phi_l \|, ...)$$

A decision rule over decisions of different hierarchy levels is a graph $\Phi^m \to \Psi^n$, where Φ^m is a parent and Ψ^n is a child, which can be interpreted as the following appropriately defined conditional probability:

$$\pi_S(\Psi^n \mid \Phi^m) = P(\| \Psi^n \|_S \mid \| \Phi^m \|_S) = \frac{\text{card}(\| \Psi^n \|_S \cap \| \Phi^m \|_S)}{\text{card}(\| \Phi^m \|_S)}$$

where $\| \Phi^m \|_S \neq \varnothing$ and

$$\text{card}(\| \Phi^m \|_S) = \prod_{j=0}^{m} \text{card}(\| \Phi^m \|_S) \times \prod_{n=m+1}^{\infty} \text{card}(\varnothing)$$

In this way. we can construct Bayesian networks using Bayes' formula:

$$\pi_S(\Psi^n \mid \Phi^m) = \frac{\pi_S(\Psi^n)\pi_S(\Phi^m \mid \Psi^n)}{\pi_S(\Phi^m \mid \Psi^n)\pi_S(\Psi^n) + \pi_S(\Phi^m \mid \neg \Psi^n)\pi_S(\neg \Psi^m)}$$

where $\pi_S(\Psi^n \mid \Phi^m)$ is the *a posteriori* probability of Ψ^n given Φ^m, $\pi_S(\Psi^n)$ is the *a priori* probability of Ψ^n, and $\pi_S(\Phi^m \mid \Psi^n)$ is the likelihood of Φ^m given Ψ^n.

Hence, we can build Bayesian networks on **S**, represented as a hierarchy of ensembles, to make a decision over an (infinite) hierarchy of different decisions.

3. Decisions involving codatabases and Bayesian networks

Precise or fuzzy data have to be fixed, i.e. they are limited by inductions (least fixed points), so that the number of their members does not change. Such data satisfy the set-theoretic axiom of foundation (e.g. this means that they are inductive sets) and hence such data are called well-founded. Nevertheless, in the case of decisions over a hierarchy of other decisions, we have dealt with hierarchies that change continuously, e.g. grow rapidly. Such an infinite hierarchy does not satisfy the foundation axiom and therefore involve so called non-well-founded data or codata [14]. Making decisions involving codata is much more sophisticated than making decisions involving well-founded data, because we cannot formulate algorithmic decision rules.

The most natural examples of codata result from processes. A process is a state-based system transforming an input sequence into an output sequence. It obtains step-by-step an input value and, depending on its current state, it produces an output value and changes its state. According to this definition, no process can be fixed as an inductive set. It is always changing, i.e. it flows, as Heraclitus of Ephesus would say. Mathematically, a process is given as a coalgebra by the following entities [12, 13, 15]: a set S of states, a set I of inputs, a set O of outputs, a set R of results, and a function $f : S \times I \to \mathrm{C}(R + S \times O)$, for some functor C. The function f describes one step of the process: in the state $s \in S$ and the input $i \in I$, f chooses a possible continuation that consists of either terminating with a result $r \in R$, or continuing in a state $s' \in S$ and producing an output value $o \in O$.

A coalgebra can be regarded as a greatest fixed point of the choice functor C. This functor C determines which kind of process we have. Important examples for choice functors are as follows:

1. The deterministic choice functor $\mathrm{C}_{\mathrm{det}}$ represented by the identity functor which is used if the input uniquely determines what happens next.

2. A non-deterministic choice functor $\mathrm{C}_{n\mathrm{det}}$ represented by a finite power-set functor which is used if for a given input there may be various possible continuations of the process.

3. A probabilistic choice functor $\mathrm{C}_{\mathrm{prob}}$ represented by a finite probability functor which is used if the continuation of the process is random.

Thus, codata are a process defined coalgebraically, where the choice functor can be understood in various ways. We transform the data **S** into an appropriate *codatabase* when we know that an unconventional logic is suited to making deductive decisions on codata of **S**. So it is possible to deal not only with (fuzzy) databases, but also with co-databases.

Graphically, coalgebras (e.g. processes or games) can be represented as infinite trees. Trees defined coalgebraically are called non-well-founded. They are the simplest graphic examples of codata.

Now let us try to define fuzzy reasoning on codatabases. Each codatabase can be considered as a transition system $TS = (S, E, T, I)$, where:

- S is a non-empty set of states,
- E is the set of actions,
- $T \subseteq S \times E \times S$ is the transition relation,
- $I \subseteq S$ is the set of initial states.

In transition systems, transitions are performed by labeled actions in the following way: if $(s, e, s') \in T$, then the system goes from s to s'. The element $(s, e, s') \in T$ is called a transition.

Any transition system $TS = (S, E, T, I)$ can be represented in the form of a labeled graph with nodes corresponding to states of S, edges representing the transition relation T, and labels of edges corresponding to events of E.

Let us extend the transition system TS to the following information system:

$$\mathsf{S} = (S, At, \{V_a : a \in At\}, \{I_a : a \in At\})$$

where: S is a finite nonempty set of objects called the universe associated with the set of all states of TS, At is a finite nonempty set of attributes which express the properties of $s \in S$, V_a is a nonempty set of values $v \in V_a$ for $a \in At$, $I_a : S \to V_a$ is an information function that maps an object in S to a value of $v \in V_a$ for an attribute $a \in At$.

Now let us build a standard logical language L_S closed over Boolean compositions of atomic formulas (a, v). The meaning $\| \Phi \|_S$ of the formula $\Phi \in \mathrm{L}_S$ is defined by induction:

$$\| (a, v) \|_S = \{s \in S : I_a(s) = v\}, \quad a \in At, \quad v \in V_a$$

$$\| \Phi \vee \Psi \|_S = \| \Phi \|_S \cup \| \Psi \|_S$$

$$\| \Phi \wedge \Psi \|_S = \| \Phi \|_S \cap \| \Psi \|_S$$

$$\| \neg \Phi \|_S = S - \| \Phi \|_S$$

Using the language L_S, we can define decision rules in S as follows: Assume each formula $\Phi \in \mathrm{L}_S$ is considered as a node of a directed, acyclic graph. Then a decision rule in S is a graph $\Phi \to \Psi$, where Φ is a parent and Ψ is a child, interpreted as an appropriately defined conditional probability, cf. [6]:

$$\pi_S(\Psi \mid \Phi) = P(\| \Psi \|_S \mid \| \Phi \|_S) = \frac{\text{card}(\| \Psi \|_S \cap \| \Phi \|_S)}{\text{card}(\| \Phi \|_S)}$$

where $\| \Phi \|_S \neq \varnothing$.

In this way, the direct causal link $\{\Phi \to \Psi\}$ corresponding to a decision rule is expressed by $\pi_S(\Psi \mid \Phi)$, the indirect causal link $\{\Phi \to \Psi, \Psi \to \Theta\}$ by $\pi_S(\Psi \mid \Phi)\pi_S(\Theta \mid \Psi)$, the common causal link $\{\Phi \to \Psi, \Theta \to \Psi\}$ by $\pi_S(\Psi \mid \Phi, \Theta)$, the common effect $\{\Phi \to \Psi, \Phi \to \Theta\}$ by $\pi_S(\Psi \mid \Phi)\pi_S(\Theta \mid \Phi)$. For each formula $\Phi \in \mathrm{L}_S$ with k atomic parents, we have 2^k rows for the combinations of parental values $v \in V_a$. Each row gives a number $p \in [0, 1]$ if Φ is true, and it gives the number $1 - p$ if Φ is false. If each formula has no more than k parents, then the complete network requires $O(2^k n)$ numbers.

So we can construct Bayesian networks in L_S by using Bayes' formula as follows:

$$\pi_S(\Psi \mid \Phi) = \frac{\pi_S(\Psi)\pi_S(\Phi \mid \Psi)}{\pi_S(\Phi \mid \Psi)\pi_S(\Psi) + \pi_S(\Phi \mid \neg \Psi)\pi_S(\neg \Psi)}$$

where $\pi_S(\Psi \mid \Phi)$ is the *a posteriori* probability of Ψ given Φ, $\pi_S(\Psi)$ is the *a priori* probability of Ψ, and $\pi_S(\Phi \mid \Psi)$ is the likelihood of Φ given Ψ. Hence, Bayes' formula allows us to infer the *a posteriori* probability $\pi_S(\Psi \mid \Phi)$ from the *a priori* probability $\pi_S(\Psi)$ via the likelihood $\pi_S(\Phi \mid \Psi)$.

Our main assumption in building a Bayesian network is as follows: for all formulas $\Phi_1, \Phi_2, \ldots, \Phi_n$ involved in constructing the decision graph, the full joint distribution can be defined as the product of the local conditional distributions: $\pi_S(\Phi_1, \ldots, \Phi_n) = \prod_{i=1}^{n} \pi_S(\Phi_i \mid \text{parents}(\Phi_i))$, where parents (Φ_i) are all the parents of Φ_i.

In defining Bayesian networks, we use the following probabilities:

$$P(s \in S) = \frac{1}{\text{card}(S)}$$

$$P(A \subseteq S) = \frac{\text{card}(A)}{\text{card}(S)}$$

$$\pi_S(\Psi) = P(\| \Psi \|_S)$$

Furthermore, we can also appeal to fuzzy probabilities defined on rough sets:

$$p_{S*}(A \subseteq S) = \frac{\text{card}(\text{Pre}_*(A))}{\text{card}(S)}$$

$$p^*_S(A \subseteq S) = \frac{\text{card}(\text{Pre}^*(A))}{\text{card}(S)}$$

where $\text{Pre}_*(X) = \{s \in S : \text{Post}(s) \neq \emptyset \text{ and } \text{Post}(s) \subseteq X\}$, $\text{Pre}_*(X) = \{s \in S : \text{Post}(s) \cap X \neq \emptyset\}$, $\text{Post}(s)$ is the set of all direct successors of s in *TS*. Thus, if the anticipation of states from A is exact, then $p_{S*}(A) = p^*_S(A) = \text{Pre}(A)$. Some other useful properties of probabilities on rough sets given $A, B \subseteq S$ are as follows (see [7] for more details):

$$p_{S*}(A) \leq \text{Pre}(A) \leq p^*_S(A)$$

$$p_{S*}(\emptyset) = p^*_S(\emptyset) = 0$$

$$p_{S*}(S) = p^*_S(S) = 1$$

$$p_{S*}(S - A) = 1 - p^*_S(A)$$

$$p^*_S(S - A) = 1 - p_{S*}(A)$$

$$p_{S*}(A \cup B) \geq p_{S*}(A) + p_{S*}(B) - p_{S*}(A \cap B);$$

$$p^*_S(A \cup B) \leq p^*_S(A) + p^*_S(B) - p^*_S(A \cap B)$$

We can also define Bayesian networks based on these probabilities in the way proposed by Yao and Zhou [17]. Instead of two-valued decisions where Φ is true or false, they proposed to use three regions corresponding to a three-way decision between acceptance, deferment, and rejection. For these networks, we use rough probabilities:

$$P_S(A) = [p_{S*}(A), p^*_S(A)]$$

Obviously, $p_{S*}(A) = p_S^*(A)$ implies $P_S(A) = \mathrm{Pre}(A)$. If $P_S(\| \Phi \|_S) = 1$, then we are dealing with acceptance, if $P_S(\| \Phi \|_S) = 0$, we are dealing with rejection, and if $0 < P_S(\| \Phi \|_S) < 1$, we are dealing with deferment.

Thus, we can construct Bayesean networks based on the probabilities corresponding to transition systems.

4. Conclusion

Bayesian networks have been defined on two kinds of coalgebra: (i) an infinite hierarchy of decisions and (ii) decisions involving transition systems. Hence, we have extended the notion of precise and imprecise IF/THEN/ELSE instructions to Bayesean networks on codata.

Acknowledgment

This research has been supported by the Narodowe Centrum Nauki (Poland) on the basis of decision No. DEC-2012/07/B/HS1/00263.

References

[1] BINMORE K., *Rational Decisions*, Princeton University Press, Princeton 2009.
[2] *Handbook of Mathematical Fuzzy Logic*, Vol. 1, 2, P. Cintula, P. Hájek, C. Noguerra (Eds.), College Publications, London 2011.
[3] HÁJEK P., *Metamathematics of Fuzzy Logic*, Kluwer Academic Publishers, Dordrecht 1998.
[4] JEFFREY R., *The Logic of Decisions*, University of Chicago Press, Chicago 1983.
[5] MUKAIDONO M., *Fuzzy Logic for Beginners*, World Scientific, Singapore 2001.
[6] PAWLAK Z., *Rough probability*, Bulletin of the Polish Academy of Sciences, Mathematical Sciences, 32 (9–10), 2002, 607.
[7] PAWLAK Z., *Rough sets, decision algorithms and Bayes theorem*, European Journal of Operational Research, 136 (1), 2002, 181.
[8] PAWLAK Z., *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht 1991.
[9] PETERSON M., *An Introduction to Decision Theory*, Cambridge University Press, Cambridge 2009.
[10] RASIOWA H., SIKORSKI R., *The Mathematics of Metamathematics*, Państwowe Wydawnictwo Naukowe, Warszawa 1963.
[11] RESNIK M., *Choices. An Introduction to Decision Theory*, University of Minnesota Press, Minneapolis 1993.

[12] RUTTEN J.J.M.M., *A coinductive calculus of streams*, Mathematical Structures in Computer Science, 15 (1), 2005, 93.
[13] RUTTEN J.J.M.M., *Behavioural differential equations a coinductive calculus of streams, automata, and power series*, Theoretical Computer Science, 308, 2003, 1.
[14] RUTTEN J.J.M.M., *Processes as terms non-well-founded models for bisimulation*, Mathematical Structures in Computer Science, 2 (3), 1992, 257.
[15] RUTTEN J.J.M.M., *Universal coalgebra. A theory of systems*, Theoretical Computer Science, 249 (1), 2000, 3.
[16] VERCELLIS C., *Business intelligence data mining and optimization for decision making*, Wiley, 2009.
[17] YAO Y., ZHOU B., *Naive Bayesian rough sets*, [in:] *Rough Set and Knowledge Technology*, Vol. 6401 of Lecture Notes in Computer Science, J. Yu, S. Greco, P. Lingras, G. Wang, A. Skowron (Eds.), Springer, Berlin 2010, 719.

Received 16 February 2015
Accepted 13 July 2015