Data-Based Construction of Multidimensional Probabilistic Models with MUDIM

RADIM JIROUŠEK, *Institute of Information Theory and Automation, Academy of Sciences, Pod vodárenskou věží 4, Praha 8, Czech Republic. E-mail: radim@utia.cas.cz*

Abstract

The goal of the paper is to introduce a program system, MUDIM, and to show how it can be used for multidimensional probabilistic model construction. The system is being developed with the goal to gain a tool for experimental computations with compositional models which are, in a way, an alternative to Bayesian networks. These models are based on the idea of composing a multidimensional distribution from a great number of low-dimensional ones. When considering knowledge-based systems, this approach quite naturally cope with the difficulty of expressing global knowledge about a field of practise. We have only to work with a system of pieces of local knowledge from which the global knowledge must be assembled.

Keywords: Compositional models, maximum entropy, software package, Bayesian network.

1 Introduction

In this text we shall deal with a relatively new class of models built within the framework of probability theory to be used for representing multidimensional distributions. However, it should be stressed that these models can also be developed equally efficiently within possibility theory [18].

When pieces of local knowledge are represented by oligodimensional distributions, the global knowledge should be represented by a multidimensional probability distribution. In artificial intelligence, application of the whole class of methods based on knowledge modelling by multidimensional probability distributions – and here we have in mind distributions of hundreds rather than tens of variables – was catalyzed by the success achieved during the last twenty years in the field often called "graphical Markov modelling [16]. Perhaps the most famous representative of these models are Bayesian networks [7, 8], but there are several others, such as influence diagrams, decomposable and graphical models, chain graph models, and others.

The approach presented herein dispenses with the necessity to describe the dependence structure of a modelled distribution in a graph. In contrast to this, the presented technique of compositional models directly describes how the multidimensional distribution is computed – composed – from a system of oligodimensional distributions, and therefore need not represent the dependence structure explicitly. Thus, we start describing the model with an assumption that there are a (usually great) number of pieces of local knowledge represented by a system of oligodimensional distributions. The task we will address in this text resembles a jig-saw puzzle that has a great

 © Oxford University Press

number of parts, each bearing a local piece of a picture, and our goal is to find how to assemble them in such a way that the global picture makes sense, reflecting all the individual small parts. In other words, we will look for an ordering of oligodimensional distributions in such a way that, when composed together, the resulting multidimensional distribution optimally reflects all the local knowledge carried by the oligodimensional distributions.

To help the reader familiar with Bayesian networks, we include Section 2.4, where the relationship between compositional models and Bayesian networks is explained (mainly) by an example. It is shown that, although these two types of models are fully equivalent in the sense that any distribution representable by one of the models can be represented also by the other one with (approximately) the same number of parameters, they also manifest some differences. We will explain how it is possible that some of the computational procedures are algorithmically simpler for perfect sequence models than for Bayesian networks.

In this paper we do not present new theoretical results (the reader will always be referred to original sources). Its goal is to briefly recollect all the notions necessary to introduce the MUDIM program system, which is intended for experimental computations and construction of compositional models, and which is under development in the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic. For this reason we will also describe a heuristic approach for data-based model construction, which will be used to illustrate the system.

2 Compositional Models

2.1 Basic notions and notation

In this text, we will deal with a finite system of finite-valued random variables, each of which describes one feature, symptom, sign, etc. Let N be an index set, $N = \{1, 2, \ldots, |N|\}$. Each variable from $\{X_i\}_{i\in N}$ is assumed to have a finite (non-empty) set of values (therefore continuous quantities are supposed to be discretized). Distributions of these variables will be denoted by Greek letters (usually π, κ); thus for $K \subset N$, we can consider a distribution $\pi((x_i)_{i\in K})$, which is defined only for variables $X_K = \{X_i\}_{i\in K}$. To make the formulae more lucid, this distribution will be denoted just by symbol $\pi(x_K)$ (when several distributions will be considered, we shall distinguish them by indices). For a probability distribution $\pi(x_K)$ and $J \subset K$ we will often consider a *marginal distribution* $\pi(x_J)$.

2.2 Operator of Composition

To be able to compose low-dimensional distributions to get a distribution of a higher dimension, we will introduce an *operator of composition*.

To make it clear from the very beginning, let us stress that it is just a generalization of the idea of computing the 3-dimensional distribution from two 2-dimensional ones introducing the conditional independence:

$$\pi(x_1, x_2) \rhd \kappa(x_2, x_3) = \frac{\pi(x_1, x_2)\kappa(x_2, x_3)}{\kappa(x_2)} = \pi(x_1, x_2)\kappa(x_3|x_2),$$

where $\kappa(x_3|x_2)$ is a respective conditional distribution.

DEFINITION 2.1
For two arbitrary distributions $\pi(x_K)$ and $\kappa(x_L)$ their *composition* is given by the formula:

$$\pi(x_K) \triangleright \kappa(x_L) = \begin{cases} \frac{\pi(x_K)\kappa(x_L)}{\kappa(x_{K\cap L})} & \text{if } \pi(x_{K\cap L}) \ll \kappa(x_{K\cap L}), \\ \text{undefined} & \text{otherwise,} \end{cases}$$

where $\pi(x_{K\cap L}) \ll \kappa(x_{K\cap L})$ means that $\kappa(x_{K\cap L})$ dominates $\pi(x_{K\cap L})$. In the finite case, which is considered in the paper, this simplifies to the condition that for all $x_{K\cap L}$

$$\kappa(x_{K\cap L}) = 0 \Longrightarrow \pi(x_{K\cap L}) = 0.$$

REMARK 2.2
Notice that we do not impose any conditions on the relationship of the two sets of variables: X_K and X_L. Nevertheless, if these sets are not disjoint, it may happen that the composition $\pi \triangleright \kappa$ remains undefined (in this case the formula need not define a probability distribution). It may happen only when for some $x_{K\cap L}$, $\pi(x_{K\cap L}) > 0$ and simultaneously $\kappa(x_{K\cap L}) = 0$. This situation is rather interesting from a theoretical point of view. Since we do not go into theoretical details in this paper, we assume that whenever a composition of two (or more) distributions is mentioned, the result will always be defined. Moreover, when the oligodimensional distributions in question are received as estimations from one data file then, as a rule, all these distributions are pairwise consistent, i.e. $\pi(x_{K\cap L}) = \kappa(x_{K\cap L})$. In this case, if for any $x_{K\cap L}$ $\pi(x_{K\cap L}) = 0$ (and therefore also $\kappa(x_{K\cap L}) = 0$) then there is a product of two zeros in the numerator and we take, quite naturally,

$$\frac{0 \cdot 0}{0} = 0.$$

If K and L are disjoint then we put $\kappa(x_{K\cap L}) = 1$ and the formula $\pi \triangleright \kappa$ degenerates to a simple product $\pi \cdot \kappa$.

The following simple assertion, which summarizes two lemmata proven in [9], answers the question: what is the result of composition of two distributions?

THEOREM 2.3
If $\pi(x_K) \triangleright \kappa(x_L)$ is defined then it is a probability distribution of variables $X_{K\cup L}$ and its marginal distribution for variables X_K equals π:

$$(\pi \triangleright \kappa)(x_K) = \pi(x_K).$$

Moreover, if π and κ are consistent (i.e., $\pi(x_{K\cap L}) = \kappa(x_{K\cap L})$) then $\pi \triangleright \kappa = \kappa \triangleright \pi$ and this composition is the maximum entropy extension of π and κ:

$$\pi \triangleright \kappa = \arg \max_{\nu \in \Pi^{(K\cup L)}(\pi,\kappa)} H(\nu),$$

where

$$\Pi^{(K\cup L)}(\pi, \kappa) = \{\nu(x_{K\cup L}) : \nu(x_K) = \pi(x_K) \ \ \& \ \ \nu(x_L) = \kappa(x_L)\}.$$

Remark 2.4
Since in this paper we are mainly interested in practical applications of operators of composition, it is sufficient just to remember that composing two distributions we receive a distribution whose dimension is larger than the dimensionality of either input distributions. In addition to this, π is always a marginal distribution of $\pi \triangleright \kappa$. The second part of the theorem says that if they are consistent then $\pi \triangleright \kappa$ comprises all the information contained in π and κ. Moreover, the composition achieves the maximum entropy among all the distributions having this property. Such a distribution is considered by many authors to be the best representative of knowledge contained in π and κ [1, 2].

2.3 Generating Sequences

The main significance of the operator of composition is in the fact that it can, when applied iteratively, form multidimensional distributions from a system of oligodimensional ones.

It is important to point out that the operator $\triangleright$ is neither commutative nor associative. For example, generally

$$\pi_1 \triangleright \pi_2 \triangleright \pi_3 \neq \pi_2 \triangleright \pi_1 \triangleright \pi_3,$$
$$\pi_1 \triangleright \pi_2 \triangleright \pi_3 \neq \pi_1 \triangleright \pi_3 \triangleright \pi_2.$$

Therefore, let us stress that, unless specified otherwise by brackets, operator $\triangleright$ is always applied from left to right. This means that

$$\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_{n-1} \triangleright \pi_n = (\ldots(\pi_1 \triangleright \pi_2) \triangleright \ldots \triangleright \pi_{n-1}) \triangleright \pi_n.$$

Thus, in order to construct a multidimensional distribution it is sufficient to determine a sequence – we will call it a *generating sequence* – of oligodimensional distributions; the resulting distribution is received by applying the operator of composition in the corresponding ordering. However, not all generating sequences are equally efficient in their representation of multidimensional distributions. Among them, the so-called perfect sequences hold an important position.

Definition 2.5
A generating sequence of probability distributions is called perfect if

$$(\pi_1 \triangleright \ldots \triangleright \pi_{k-1}) \triangleright \pi_k = \pi_k \triangleright (\pi_1 \triangleright \ldots \triangleright \pi_{k-1})$$

holds for all $k = 2, \ldots, n$.

From this definition one can hardly see the importance of perfect sequences. This importance becomes clearer from the following characterization theorem (proven in [11]).

Theorem 2.6
A sequence of distributions $\pi_1, \pi_2, \ldots, \pi_n$ is perfect *iff* all the distributions from this sequence are marginals of the distribution $\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n$.

The above-presented theorem claims that a model defined by a generating sequence preserves all the given marginals only when the model is defined by a perfect sequence.

In addition to this, it is not difficult to show that for a perfect sequence $\pi_1, \pi_2, \ldots, \pi_n$ distribution $\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n$ is the maximum entropy extension of all the distributions[1] $\pi_1, \pi_2, \ldots, \pi_n$. If the considered generating sequence is not perfect then some of the marginal distributions computed from the resulting model differ from the given ones. It is possible to show that a non-perfect generating sequence need not preserve even one-dimensional marginal distributions.

All in all, this is the main message conveyed by the above-presented characterization theorem: Considering that oligodimensional distributions from a perfect sequence are carriers of local information, the constructed multidimensional distribution $\pi_1 \triangleright \ldots \triangleright \pi_n$ represents global information, faithfully reflecting all of the local input. This is why we will be so interested in perfect sequence models.

Remark 2.7
At this point the reader can realize one of the differences between Bayesian networks and perfect sequence representation of multidimensional probability distributions. When defining a Bayesian network, one can take an acyclic directed graph (which determines the structure of the model) and any respective system of conditional distributions. When defining these distributions, one need not care about their consistency, because *any* system of distributions (corresponding to the graph) defines a multidimensional distribution.

However, defining a perfect sequence model is more difficult. After selecting a structure of the model (it is defined by the selected sets of variables – more precisely their indices – for which the distributions are to be defined: $K_1, \ldots, K_n$) one cannot take an arbitrary system of distributions $\pi_1(x_{K_1}), \ldots, \pi_n(x_{K_n})$; it has to meet a rather strong condition of Definition 2.5. Though it seems to be just a complication in comparison with Bayesian network construction, it brings the following two advantages:

1. Generally, the more conditions on model construction, the higher chance that the resulting distribution manifests the required properties.
2. As it will be shown in Section 2.4, there are (explicitly expressed) probabilities in a perfect sequence model, which can be computed from the corresponding Bayesian network; however, such computation may appear to be computationally expensive.

Example 2.8
Consider the generating sequence

$$\pi_1(x_1, x_2), \pi_2(x_1, x_3), \pi_3(x_2, x_4), \pi_4(x_3, x_4, x_5),$$

with values given in Table 1. To check that this sequence is perfect we reformulate the condition of Definition 2.5 into an equivalent one (the equivalence will be obvious from Theorem 2.9): a generating sequence $\pi_1(x_{K_1}), \ldots, \pi_n(x_{K_n})$ is perfect if and only if

$$\pi_i(x_{K_i \cap (K_1 \cup \ldots \cup K_{i-1})}) = (\pi_1 \triangleright \ldots \triangleright \pi_{i-1})(x_{K_i \cap (K_1 \cup \ldots \cup K_{i-1})})$$

[1] It follows from the fact that due to the definition of perfect sequence

$$\pi_1 \triangleright \pi_2 \triangleright \pi_3 \triangleright \ldots \triangleright \pi_n = \pi_n \triangleright (\pi_{n-1} \triangleright \ldots \triangleright (\pi_3 \triangleright (\pi_2 \triangleright \pi_1)) \ldots),$$

which is exactly the distribution computed by application of the *Iterative Proportional Fitting Procedure* to $\pi_1, \pi_2, \ldots, \pi_n$ starting with the uniform distribution.

TABLE 1. Distributions $\pi_1, \pi_2, \pi_3, \pi_4$

π_1	x_1	
x_2	$\frac{1}{2}$	$\frac{1}{4}$
	0	$\frac{1}{4}$

π_2	x_1	
x_3	$\frac{3}{8}$	$\frac{1}{8}$
	$\frac{1}{8}$	$\frac{3}{8}$

π_3	x_4	
x_2	$\frac{3}{8}$	$\frac{3}{8}$
	$\frac{1}{8}$	$\frac{1}{8}$

π_4	x_4			
	x_5		x_5	
x_3	$\frac{1}{4}$	0	0	$\frac{1}{4}$
	0	$\frac{1}{4}$	$\frac{1}{4}$	0

TABLE 2. Distribution $\pi_1 \triangleright \pi_2$

$\pi_1 \triangleright \pi_2$	x_1				
	x_3		x_3		
x_2	$\frac{3}{8}$	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{3}{4}$
	0	0	$\frac{1}{16}$	$\frac{3}{16}$	$\frac{1}{4}$

for all $i = 2, \ldots, n$. Using this criterion it is not difficult to check that this sequence is perfect. We have just to show that

$$\pi_1(x_1) = \pi_2(x_1), \tag{2.1}$$
$$(\pi_1 \triangleright \pi_2)(x_2) = \pi_3(x_2), \tag{2.2}$$
$$(\pi_1 \triangleright \pi_2 \triangleright \pi_3)(x_3, x_4) = \pi_4(x_3, x_4). \tag{2.3}$$

Equality (2.1) is trivial: both distributions $\pi_1(x_1)$ and $\pi_2(x_1)$ are uniform. Applying the formula from Definition 2.1 to π_1 and π_2 we get distribution $\pi_1 \triangleright \pi_2$, which is in Table 2. From this, one immediately sees that distributions $(\pi_1 \triangleright \pi_2)(x_2)$ and $\pi_3(x_2)$ are equivalent, too. So it remains to show that equivalence (2.3) also holds true.

First, using Definition 2.1 compute[2]

$$(\pi_1 \triangleright \pi_2 \triangleright \pi_3)(x_2, x_3, x_4) = (\pi_1 \triangleright \pi_2)(x_2, x_3) \triangleright \pi_3(x_2, x_4),$$

which is recorded in Table 3. Now, we immediately see that although neither of the distributions $\pi_1 \triangleright \pi_2 \triangleright \pi_3$ and π_4 is uniform, their marginals $(\pi_1 \triangleright \pi_2 \triangleright \pi_3)(x_3, x_4)$ and $\pi_4(x_3, x_4)$ are uniform. Thus, as all equations (2.1), (2.2) and (2.3) hold true, $\pi_1, \pi_2, \pi_3, \pi_4$ form a perfect sequence.

Since π_1 and π_2 are consistent, we see immediately from Theorem 2.6 that the sequence

$$\pi_2(x_1, x_3), \pi_1(x_1, x_2), \pi_3(x_2, x_4), \pi_4(x_3, x_4, x_5)$$

[2]Here we are also using the following assertion, which was proved in [11]

LEMMA

Consider $\pi(x_K)$ and $\kappa(x_L)$ for which $\pi \triangleright \kappa$ is defined. If $K \cap L \subseteq M \subseteq K \cup L$ then

$$(\pi \triangleright \kappa)(x_M) = \pi(x_{K \cap M}) \triangleright \kappa(x_{L \cap M}).$$

TABLE 3. Distribution $(\pi_1 \triangleright \pi_2 \triangleright \pi_3)(x_2, x_3, x_4)$

$\pi_1 \triangleright \pi_2 \triangleright \pi_3$	x_3			
	x_4		x_4	
x_2	$\frac{7}{32}$	$\frac{7}{32}$	$\frac{5}{32}$	$\frac{5}{32}$
	$\frac{1}{32}$	$\frac{1}{32}$	$\frac{3}{32}$	$\frac{3}{32}$

is also perfect. We leave to the reader to show that sequences $\pi_1, \pi_3, \pi_2, \pi_4$ and $\pi_3, \pi_1, \pi_2, \pi_4$ are perfect, too, and that all these sequences define the same distribution (hint: realize that all of them are maximum entropy extensions of the same system of distributions). Notice, however, that neither from the remaining 20 permutations of the considered distributions is a perfect sequence. Let us show it, for example, for

$$\pi_2(x_1, x_3), \pi_3(x_2, x_4), \pi_1(x_1, x_2), \pi_4(x_3, x_4, x_5).$$

Since both π_2 and π_3 are strictly positive, it is an immediate consequence of the formula from Definition 2.1 that $\pi_2 \triangleright \pi_3$ must also be strictly positive, which is not true for π_1. (In addition to this,

$$(\pi_2 \triangleright \pi_3)(x_1, x_2, x_3, x_4) = \pi_2(x_1, x_3) \cdot \pi_3(x_2, x_4)$$

and therefore X_1 and X_2 are independent under distribution $\pi_2 \triangleright \pi_3$, which is also in contradiction with distribution π_1.)

The following assertion shows that each generating sequence can be transformed into a perfect sequence (see [9, 10]). This property (among others) is of great importance for the model construction process presented below.

THEOREM 2.9
For an arbitrary generating sequence $\pi_1(x_{K_1}), \ldots, \pi_n(x_{K_n})$, the sequence $\kappa_1, \ldots, \kappa_n$ computed by the following process

$$\begin{aligned}
\kappa_1(x_{K_1}) &= \pi_1(x_{K_1}) \\
\kappa_2(x_{K_2}) &= \kappa_1(x_{K_2 \cap K_1}) \triangleright \pi_2(x_{K_2}) \\
\kappa_3(x_{K_3}) &= (\kappa_1 \triangleright \kappa_2)(x_{K_3 \cap (K_1 \cup K_2)}) \triangleright \pi_3(x_{K_3}) \\
&\vdots \\
\kappa_n(x_{K_n}) &= (\kappa_1 \triangleright \ldots \triangleright \kappa_{n-1})(x_{K_n \cap (K_1 \cup \ldots \cup K_{n-1})}) \triangleright \pi_n(x_{K_n}),
\end{aligned}$$

is perfect and

$$\kappa_1 \triangleright \kappa_2 \triangleright \ldots \triangleright \kappa_n = \pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n.$$

This result gives us not only the instructions for testing whether a generating sequence is perfect or not, but from this we can see that the perfectness is not a *structural* property of a sequence (it does not depend only on the system of sets, which form arguments of individual distributions) but a property speaking about all individual probabilities. As a trivial example, let us realize that *any* sequence of uniform distributions form a perfect sequence.

2.4 Relation to Bayesian networks

Before we start discussing the process of compositional model construction, let us answer an important question: what type of distributions can be represented by perfect sequences? It is not too difficult to prove that these are exactly the distributions representable by Bayesian networks. In fact, the class of Bayesian networks is equivalent to the class of perfect sequence models in the following sense:

1. If $\pi_1(x_{K_1}), \pi_2(x_{K_2}), \ldots, \pi_n(x_{K_n})$ is perfect, then there exists a Bayesian network with an acyclic directed graph $G = (N, E)$, such that the distribution $\pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n$ is a distribution represented by the Bayesian network, and for each node $i \in N$ of graph G with parents $pa(i)$ there exists (at least one) $k \in N$ such that $\{i\} \cup pa(i) \subseteq K_k$. (This means that each conditional distribution appearing in the definition of the Bayesian network can be computed directly from the distribution π_k.)
2. For each Bayesian network representing a distribution κ one can construct a perfect sequence $\pi_1(x_{K_1}), \pi_2(x_{K_2}), \ldots, \pi_n(x_{K_n})$, such that $\kappa = \pi_1 \triangleright \pi_2 \triangleright \ldots \triangleright \pi_n$ and for each distribution $\pi_k(x_{K_k})$ set K_k consists of a node and its parents from the graph of the considered Bayesian network; i.e., there is $i \in K_k$, for which $K_k = \{i\} \cup pa(i)$.

Yet there is a rather important difference between these two types of multidimensional distribution representations, which comes to light when one starts considering the processes transforming these models into each other. Let us illustrate it using a simple example.

Example 2.10
Consider the perfect sequence from Example 2.8

$$\pi_1(x_1, x_2), \pi_2(x_1, x_3), \pi_3(x_2, x_4).\pi_4(x_3, x_4, x_5).$$

It represents the distribution

$$\begin{aligned} \rho(x_1, x_2, x_3, x_4, x_5) &= \pi_1(x_1, x_2) \triangleright \pi_2(x_1, x_3) \triangleright \pi_3(x_2, x_4) \triangleright \pi_4(x_3, x_4, x_5) \\ &= \pi_1(x_1, x_2)\pi_2(x_3|x_1)\pi_3(x_4|x_2), \pi_4(x_5|x_3, x_4). \end{aligned}$$

Thus, we immediately see that this distribution is also represented by a Bayesian network with a graph in Figure 1 and the corresponding system of conditional distributions

$$\begin{aligned} &\nu_1(x_1) = \pi_1(x_1), && \nu_2(x_2|x_1) = \pi_1(x_2|x_1), \\ &\nu_3(x_3|x_1) = \pi_2(x_3|x_1), && \nu_4(x_4|x_2) = \pi_3(x_4|x_2), \\ &&& \nu_5(x_5|x_3, x_4) = \pi_4(x_5|x_3, x_4). \end{aligned}$$

Notice that the distributions ν_i are easily computed from the respective (unconditional) distributions π_i in a *local* way; one does not need any auxiliary memory and the computational time depends linearly on the size of the distributions π_i.

Let us turn our attention to the opposite task. Consider the Bayesian network with graph from Figure 1 and with the corresponding 5 conditional distributions $\nu_1, \ldots, \nu_5$. It represents the distribution

$$\rho(x_1, x_2, x_3, x_4, x_5) = \nu_1(x_1)\nu_2(x_2|x_1)\nu_3(x_3|x_1)\nu_4(x_4|x_2), \nu_5(x_5|x_3, x_4).$$

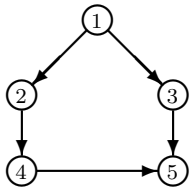

FIG. 1. Acyclic graph of a Bayesian network

Our goal is now to get the corresponding perfect sequence model. We know that any generating sequence $\mu_1, \ldots, \mu_5$ for which

$$\begin{array}{ll} \mu_1(x_1) = \nu_1(x_1), & \mu_2(x_2|x_1) = \nu_2(x_2|x_1), \\ \mu_3(x_3|x_1) = \nu_3(x_3|x_1), & \mu_4(x_4|x_2) = \nu_4(x_4|x_2), \\ \multicolumn{2}{c}{\mu_5(x_5|x_3,x_4) = \nu_5(x_5|x_3,x_4)} \end{array}$$

defines the required 5-dimensional distribution ρ (providing that $\mu_1 \triangleright \ldots \triangleright \mu_5$ is defined). Such a sequence can easily be obtained in many ways. For example, taking a uniform distribution λ, distributions μ_i computed according to the formulae

$$\begin{array}{ll} \mu_1(x_1) = \nu_1(x_1), & \mu_2(x_1,x_2) = \lambda(x_1)\nu_2(x_2|x_1), \\ \mu_3(x_3|x_1) = \lambda(x_1)\nu_3(x_3|x_1), & \mu_4(x_4|x_2) = \lambda(x_2)\nu_4(x_4|x_2), \\ \multicolumn{2}{c}{\mu_5(x_5|x_3,x_4) = \lambda(x_3,x_4)\nu_5(x_5|x_3,x_4)} \end{array}$$

form a generating sequence defining the same 5-dimensional distribution as the initial Bayesian network. Notice that these computations can again be performed locally. This sequence, however, is not perfect unless the Bayesian network is to some extent degenerated (its 4-dimensional marginal distribution $\rho(x_1,x_2,x_3,x_4)$ is uniform), which is not the case for the distributions $\nu_1, \ldots, \nu_5$ computed from the considered distributions $\pi_1, \ldots, \pi_4$. Nevertheless, let us recall that, due to Theorem 2.9, this sequence can be transformed into a perfect sequence

$$\begin{aligned} \kappa_1(x_1) &= \mu_1(x_1) \\ \kappa_2(x_1,x_2) &= \kappa_1(x_1) \triangleright \mu_2(x_1,x_2) \\ \kappa_3(x_1,x_3) &= (\kappa_1 \triangleright \kappa_2)(x_1) \triangleright \mu_3(x_1,x_3) \\ \kappa_4(x_2,x_4) &= (\kappa_1 \triangleright \kappa_2 \triangleright \kappa_3)(x_2) \triangleright \mu_4(x_2,x_4) \\ \kappa_5(x_3,x_4,x_5) &= (\kappa_1 \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_4)(x_3,x_4) \triangleright \mu_5(x_3,x_4,x_5). \end{aligned}$$

And it is the very computation of

$$(\kappa_1 \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_4)(x_3,x_4)$$

that cannot be performed locally. We will not analyse its computational complexity in detail but it depends, in a way, on the size of the 4-dimensional space of all combinations of values x_1, x_2, x_3, x_4. (This does not mean that we have to compute $(\kappa_1 \triangleright \kappa_2 \triangleright \kappa_3 \triangleright \kappa_4)(x_3,x_4)$ by summing up over 4-dimensional space. We can, for example, compute first $(\kappa_1 \triangleright \kappa_2 \triangleright \kappa_3)(x_2,x_3)$ and then compute the required distribution marginalizing from 3-dimensional distribution $(\kappa_1 \triangleright \kappa_2 \triangleright \kappa_3)(x_2,x_3) \triangleright \kappa_4(x_2,x_4)$. So it means that we can get the required distribution by marginalizing twice over 3-dimensional space.)

From what has been said in the previous example we see a difference in representation of multidimensional distributions by Bayesian networks and perfect sequence models. Though both types of models represent the same probability distributions, in perfect sequence models some of the marginal distributions are directly expressed. These can be computed from the Bayesian network too; however, their computation may be rather computationally expensive. This fact also explains how it is possible that some of computational procedures (especially those requiring enumeration of marginal distributions) for perfect sequence models are simpler (i.e., more computationally efficient) than the respective procedures for Bayesian networks.

3 Data-Based Process for Model Construction

The reader interested in other theoretical issues concerning the operator of composition and perfect sequence models is referred to [14] and the papers cited there. At this point, we want to briefly introduce a possible way to create a perfect sequence model from a data file – see Figure 2. This scheme describes an expert-driven process whose individual steps can be realized with the help of MUDIM; nevertheless, all the decisions within the process are made by an expert:

1. Selection of oligodimensional distributions at the beginning of the whole process.
2. Decision which type of "refinement" procedure should be chosen (detailed explanation is given below).
3. Stopping rule.

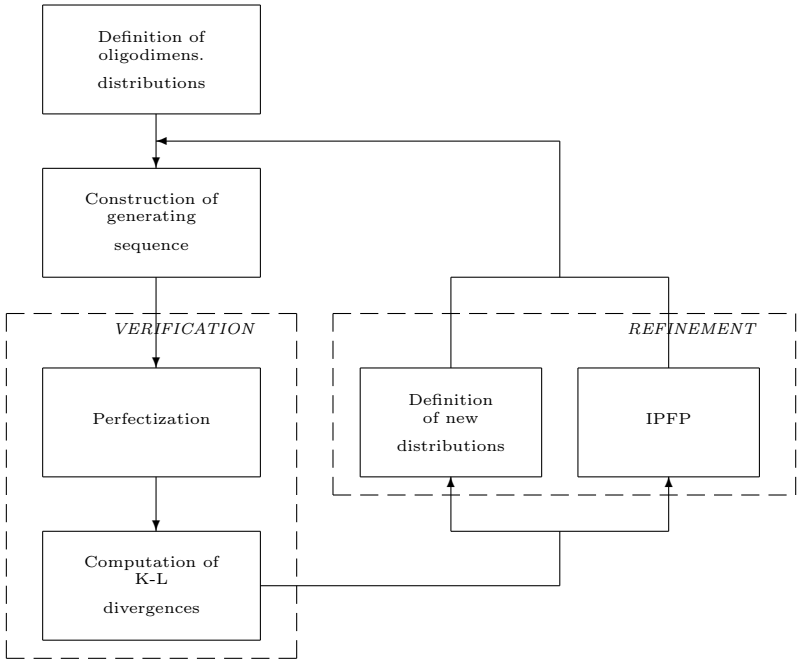

FIG. 2. Process of model construction

As it can be seen from the diagram, the process is initiated with definition of a system of oligodimensional distributions. Regarding the fact that the process cyclically employs steps of *verification* and *refinement*, during which this initial system is gradually changed, the result is fairly independent of the initial selection. For example, starting with all 2-dimensional distributions may be quite reasonable (another possibility is used in Example 3.3). Generally, we propose to select distributions carrying a greater amount of information. This idea is supported by the following theorem, proved in [14] (Corollary 1.). It claims that the higher information content of a perfect sequence, the better approximation of the unknown distribution. Before presenting the respective assertion we have to introduce two concepts: Kullback-Leibler divergence (also called cross-entropy, or relative entropy) and information content.

Definition 3.1
Klullback-Leibler divergence of distributions $\pi(x_N)$ and $\kappa(x_N)$ is given by the following formula

$$Div(\pi\|\kappa) = \begin{cases} \sum\limits_{x_N} \pi(x_N) \log \frac{\pi(x_N)}{\kappa(x_N)} & \text{if } \pi \ll \kappa, \\ +\infty & \text{otherwise.} \end{cases}$$

Information content of a distribution $\pi(x)$ is the Kullback-Leibler divergence of π and a product distribution of its 1-dimensional marginal distributions:

$$I(\pi) = Div(\pi\| \prod_{i\in N} \pi(x_i)) = \sum_{x_N} \pi(x_N) \log \frac{\pi(x_N)}{\prod\limits_{i\in N} \pi(x_i)}.$$

Let us recall that the information content is just a natural generalization of a Shannon mutual information, which will be used in the algorithm further in this text, and which is for 2-dimensional distribution $\pi(x_1, x_2)$ defined

$$MI_\pi(x_1; x_2) = \sum_{(x_1,x_2)} \pi(x_1, x_2) \log \frac{\pi(x_1, x_2)}{\pi(x_1)\pi(x_2)},$$

and in a general case of $\pi(x_N)$ and two disjoint (nonempty) $K_1, K_2 \subset N$ it is expressed by an analogous formula

$$MI_\pi(x_{K_1}; x_{K_2}) = \sum_{x_{K_1\cup K_2}} \pi(x_{K_1\cup K_2}) \log \frac{\pi(x_{K_1\cup K_2})}{\pi(x_{K_1})\pi(x_{K_2})}.$$

The reader most likely noticed that, since π is always dominated by the product distribution $\prod_{i\in N} \pi(x_i)$, the information content is always finite. Moreover, recall that it is always nonnegative and equals 0 if and only if π is a product distribution.

Now, we are able to present the previously-announced assertion, which provides crucial theoretical support of the algorithm for generating sequence construction.

Theorem 3.2
If for a distribution κ a generating sequence of its marginals $\kappa(x_{K_1})$, $\kappa(x_{K_2}), \ldots, \kappa(x_{K_n})$ is perfect, then

$$Div(\kappa\|\kappa(x_{K_1}) \triangleright \ldots \triangleright \kappa(x_{K_n})) = I(\kappa) - I(\kappa(x_{K_1}) \triangleright \ldots \triangleright \kappa(x_{K_n})).$$

Therefore, if we want to construct a perfect sequence model approximating an unknown distribution κ, we have to aim at getting the model with the highest possible information content (under the assumption that the oligodimensional distributions, which the perfect sequence consists of, are marginals of the approximated distribution). In [14] we have published the following heuristic algorithm producing a sub-optimal generating sequence from a system of oligodimensional distributions.

Algorithm

Input: System of low-dimensional distributions $\pi_1(x_{K_1}), \ldots \pi_n(x_{K_n})$.

Initialization: Select a variable X_m and a distribution π_j such that $m \in K_j$. Set $\kappa_1 := \pi_j(x_m)$, $L := \{m\}$ and $k := 1$.

Computational Cycle: While $K_1 \cup \ldots \cup K_n \setminus L \neq \emptyset$ perform the following 3 steps:

1. for all $j = 1, \ldots, n$ and all $m \in K_j \setminus L$ compute the mutual information
$$MI_{\pi_j}(X_m; X_{K_j \cap L}).$$
2. Fix j and m for which $MI_{\pi_j}(X_m; X_{K_j \cap L})$ achieved its maximal value.
3. Increase k by 1. Set $\kappa_k := \pi_j(x_{(K_j \cap L) \cup \{m\}})$ and $L := L \cup \{m\}$.

Output: Generating sequence $\kappa_1, \kappa_2, \ldots, \kappa_k$.

What can be said about the resulting generating sequence $\kappa_1, \kappa_2, \ldots, \kappa_k$? Distribution $\rho = \kappa_1 \triangleright \kappa_2 \triangleright \ldots \triangleright \kappa_k$ is a probability distribution of variables $X_{K_1 \cup K_2 \cup \ldots \cup K_n}$. The algorithm realizes a greedy (therefore very efficient) process, which seeks to find a sequence utilizing the information content of individual oligodimensional distributions in a maximal possible way. The result is a generating sequence which, unfortunately, need not be perfect. It means that some of the input distributions are not marginals of the resulting multidimensional model. As a rule, the expert (the model constructor) has to accept some deviations of the model marginals from the input oligodimensional distributions. To decide whether the obtained deviations are acceptable, i.e., whether the whole model construction process depicted in Figure 2 should be terminated, the expert must be provided with some additional information. To get it, the process employs the perfectization procedure described in Theorem 2.9. Then it is possible to compare original oligodimensional distributions with the corresponding marginals defined by the model. The comparison may be done with the help of Kullback-Leibler divergence; as already said above, its value equals 0 *iff* $\pi = \kappa$, otherwise it is always positive. Therefore, the lower this value, the closer κ to π. The goal of this step is to find all the marginal distributions which are unacceptably distorted by the model. If there is no such a marginal distribution, the process is terminated. In the opposite case, the expert proposes to perform another cycle of the whole process with a modified system of oligodimensional distributions. The described process then proceeds so that several original distributions are substituted with one *a-little-bit-more*-dimensional one in the *refinement* step.

As the reader can see from Figure 2, there are two possibilities to get these new distributions. If it is possible (i.e., the data file is large enough) the expert can decide to get them as estimations from the given data file (going along the left branch of the *refinement* box in Figure 2). However, if the data file is too small to get reliable esti-

mations (which may happen easily if one needs to substitute several distributions with a distribution whose dimensionality is high – let us say, 6 or more), then one can take advantage of the well-known Iterative Proportional Fitting Procedure (IPFP) (see [4]; for its effective implementation, which makes it possible to compute distributions of pretty high dimensions, see e.g. [6]). This is a procedure computing a maximum entropy distribution with given marginals. Using the operator of composition, this procedure consists in iterative computation of distributions

$$\kappa_j = \pi_i \rhd \kappa_{j-1},$$

where κ_0 is a uniform distribution, $i = ((j-1) \bmod s) + 1$ and $\pi_1, \ldots, \pi_s$ are the selected marginal distributions. It is known (for the proof see [3]) that, if there exists a distribution for which all $\pi_1, \ldots, \pi_s$ are marginals, then the sequence $\kappa_0, \kappa_1, \kappa_2, \ldots$ converges and its limit distribution $\lim_{j\to\infty} \kappa_j$ maximizes Shannon entropy among all the distributions with the given marginals.

In this way, when all the desired substitutions are realized, a new system of oligodimensional distributions is set up, to which the heuristic algorithm for generating sequence construction is again applied. The described cycle is repeated until the expert decides that a suitable multidimensional model representing (approximating) all the required oligodimensional distributions has been achieved.

Let us stress that the process shown in Figure 2 is fully controlled by the expert. The more cycles of the process are performed, the higher dimensions of the input distributions are considered. If the expert had continued ad absurdum, the process would have, in fact, finished with an application of IPFP to all of the initial oligodimensional distributions (which is, as a rule, computationally intractable in practical situations).

Example 3.3

Let us illustrate the behaviour of the above-described process when applied to a small medical data file. The goal was to find a model corresponding to a probability distribution describing relationships among the 11 cardiological risk factors (variables) listed in Table 4.

From the data file we computed all 165 3-dimensional distributions and selected those whose informational content $IC \geq 0.025$. Thus we got 10 (oligodimensional) distributions $\pi_1, \ldots, \pi_{10}$ presented in Table 5.

When initialized with variable L and distribution π_1, the algorithm produced the sequence

$$\pi_1(L,W,Y), \pi_2(T,W,Y), \pi_4(C,W,Y), \pi_7(C,H,Y), \pi_5(H,S,Y), \\ \pi_3(E,H,S), \pi_8(D,H,Y), \pi_{10}(A,D,H), \pi_9(A,B,D). \quad (3.1)$$

This sequence was perfect and therefore we did not need to perform the process of perfectization. Nevertheless, because distribution $\pi_6(A,B,H)$ was not included in the model, we computed the respective Kullback-Leibler divergence, getting

$$Div(\pi_6(A,B,H) \| \kappa(A,B,H)) = 0.023,$$

for

$$\kappa = \pi_1 \rhd \pi_2 \rhd \pi_4 \rhd \pi_7 \rhd \pi_5 \rhd \pi_3 \rhd \pi_8 \rhd \pi_{10} \rhd \pi_9.$$

TABLE 4. List of risk factors - variables

variable	**denotation**	**n. of values**
age	A	4
BMI	B	4
cholesterol	C	4
diabetes	D	3
beer consumption	E	4
hypertension	H	3
liquor consumpt.	L	4
smoking	S	4
triglycerides	T	4
wine consumption	W	4
hyperlipidemia	Y	3

TABLE 5. Triplets of variables ordered by IC values

n. of distrib.	**variables**	**IC**
1	LWY	0.726
2	TWY	0.102
3	EHS	0.088
4	CWY	0.080
5	HSY	0.059
6	ABH	0.046
7	CHY	0.045
8	DHY	0.040
9	ABD	0.030
10	ADH	0.029

At this stage we had to decide, based on the value of $Div(\pi_6(A,B,H)\|\kappa(A,B,H))$, whether to employ the step of *refinement*, or to finish the process. We decided to continue.

The refinement process led to substituting three distributions $\pi_6(A,B,H)$, $\pi_9(A,B,D)$ and $\pi_{10}(A,D,H)$ by a 4-dimensional distribution $\pi_{11}(A,B,D,H)$. Since the number of cases in the data file at our disposal did not enable us to compute estimates of 4-dimensional distributions (we only had 530 cases) we chose the other possibility: we computed distribution π_{11} from π_6, π_9, π_{10} by application of IPFP.

The second step of the process, when the heuristic algorithm was applied to distributions $\pi_1, \pi_2, \pi_3, \pi_4, \pi_5, \pi_7, \pi_8, \pi_{11}$, eventually yielded a perfect sequence

$$\pi_1(L,W,Y), \pi_2(T,W,Y), \pi_4(C,W,Y), \pi_7(C,H,Y), \pi_5(H,S,Y), \\ \pi_3(E,H,S), \pi_8(D,H,Y), \pi_{11}(A,B,D,H).$$

4 MUDIM system

Let us conclude the paper with a brief demonstration of how the heuristic algorithm introduced in the previous section can be realized in the MUDIM system [3]. Naturally, no computational system can be described on a few pages, so the reader is kindly asked to take this section rather as a MUDIM advertisement.

The system, which is still under development, is built in the environment of **R** language ([19]) and is intended to help experimentalists to build models describing collected data as well as researchers to analyse properties of the compositional models. In agreement with the philosophy of **R** language, MUDIM is available in source code; it is free software and comes with absolutely no warranty. Everybody is welcome to extend it by her/his own statements and redistribute it under certain conditions (those given by the group developing **R** language).

The first version of MUDIM, which is currently available[4], realizes only a rather limited number of special MUDIM statements, but a vast abundance of original **R** functions makes it possible that the user can code rather complex computational algorithms, such as those for automatic model construction, even at the current stage of MUDIM development.

All computations with the MUDIM system are performed within the framework of a "problem". This is (usually) specified by one data file, system of variables, each of which is given by its name and a (finite) list of its values. Thus each session must start either with problem definition or by loading a problem saved in previous sessions. So, for example, if one needs recollection of the active problem description, it is enough just to type

```
mudim.get.info(ProblemA)
```

where `ProblemA` is either a name or a handle of the processed problem.

It is the philosophy of **R** language that all subjects can be identified either by their names (as the name `ProblemA` used in previous statement) or by their "handles", unique numbers assigned to all objects by **R**. This, as it will be seen from the next MUDIM statements, makes processing of lists of objects easier. For example, the following statement assigns handles of all variables into the object named `Handlevars`.

```
Handlevars <- mudim.get.all.variables(ProblemA)
```

In the previous section, we mentioned that it is not a bad idea to start the heuristic method of model construction with all the 2-dimensional marginal distributions. To get them from a data file representing a distribution with the name `Datadistrib`, one can use the following statements (handles of the resulting 2-dimensional marginal distributions are placed into `Handledistrib2`)[5]:

```
Handledistrib2 <- NULL
for (i in Handlevars)
     {for (j in Handlevars) if (j > i)
          Handledistrib2 <- c(Handledistrib2,
```

[3] MUDIM stands for MUltiDImensional Models. The presented examples are taken from [15].

[4] The system is currently available from <http://mtr.utia.cas.cz/mudim>.

[5] `c(... , ...)` is a standard function for concatenation in **R** language.

```
            mudim.marginalize(ProblemA, Datadistrib)
    }
```

Now, it is easy to compute informational content of all these 2-dimensional marginal distributions and to order them according to these values:

```
Informs <- NULL
for (i in Handledistrib2)
      Informs <- c(Informs, mudim.get.informationvalue(i))
Ordering <- order(Informs,decreasing=TRUE)
```

In practical situations, processing of all 2-dimensional distributions may be rather time consuming. So it may be advantageous to find out how many out of these 2-dimensional distributions (respecting the ordering according to their informational content) are necessary to build a model (distribution), which is defined for all the variables. In other words, assuming that $\pi_{i_1}, \pi_{i_2}, \ldots, \pi_{i_n}$ is a permutation of all the considered 2-dimensional distributions for which $IC(\pi_{i_1}) \geq IC(\pi_{i_2}) \geq \ldots \geq IC(\pi_{i_n})$, we want to find the smallest m such that

$$\pi_{i_1} \triangleright \pi_{i_2} \triangleright \ldots \triangleright \pi_{i_m}$$

is a distribution of all the variables appearing in the problem in question. It means that we are looking for the smallest m such that $K_{i_1} \cup K_{i_2} \cup \ldots \cup K_{i_m} = N$. This can be realized by the following statements.

```
Numvariables <- length(mudim.get.all.variables(ProblemA))
Nummarginals <-length(Handledistrib2)
Numselectmarginals <- 0; Selectedvariables <- NULL
for (i in 1:Nummarginals)
     { if (length(Selectvariables) < Numvariables)
        { Selectvariables <- union(Selectvariables,
             mudim.get.all.variables(Handledistrib2[Ordering[i]]))
          Numselectmarginals <- Numselectmarginals + 1
        }
     }
```

The required number m is in `Numselectmarginals`.

In conclusion of these examples, let us present commented **R**-code of a function realizing the heuristic algorithm for model construction introduced in the previous section.

```
       ################################################
       #                                              #
       # HEURISTIC ALGORITHM FOR MODEL CONSTRUCTION #
       #                                              #
       ################################################
# Description of parameters:
#        1.  aListofDistribution - list of handles of distributions
```

```
#        2.  StartingVariable - handle of a selected variable

mudim.algorithm <- function(aListofDistribution,StartingVariable)
{ selectvariables <- StartingVariable
  selectdistributions <- NULL; allvariables <- NULL
  for(dist in aListofDistribution)
   allvariables <- union(allvariables,mudim.get.all.variables(dist))
# end of INITIALIZATION step
# 'allvariables' contains handles of all the variables, for which
# the model should be constructed

# beginning of COMPUTATIONAL CYCLE
# here 'setdiff' (difference of sets) and 'length' are standard R
# functions
# the following cycle is performed until the sets
# 'selectvariable' and 'allvariables' are the same
  while(length(setdiff(allvariables,selectvariables)))
    { infmemory <- -1
      for (dist in aListofDistribution)
        { variables <- mudim.get.all.variables(dist)
# to speed up the cycle, the following tests exclude computation of
# mutual information between groups of variables from which at least
# one is empty
          if (length(intersect(selectvariables,variables)) > 0
                & length(setdiff(variables,selectvariables)) > 0)
# mutual information is here computed as a difference of two values
# of informational content
            { infselect <- mudim.get.informationvalue(
                 mudim.marginal(,dist,selectvariables))
              for (v in setdiff(variables,selectvariables))
                { x <- c(selectvariables, v)
                  inf <- (mudim.get.informationvalue(
                       mudim.marginal(,dist,v)) + infselect
                       - mudim.get.informationvalue(
                       mudim.marginal(,dist,x)))
                  if (inf > infmemory)
                    { infmemory <- inf; varmemory <- v
                      distmemory <- dist
                    }
                }
            }
        }
      selectvariables <- c(selectvariables,varmemory)
      selectdistributions <- c(selectdistributions,distmemory)
    }      # end of COMPUTATIONAL CYCLE
  return(selectdistributions)
}
```

Let us conclude this section by summarizing what are the most important special MUDIM statements realized in its first version. Not speaking about the basic statements necessary for problem definition and data organization, most important are those defining oligodimensional distributions, computing their marginals and creating compositional models from generating sequences of (oligodimensional) distributions. Model construction processes are supported by functions computing entropy and information content of distributions. To be able to evaluate different models, we can use a function initiating computation of the Kullback-Leibler divergence of two distributions.

5 Conclusions

In this paper we presented a minimum of theory of compositional models necessary to comprehend that this approach is an advantageous alternative to Bayesian networks. Advantages of compositional models are mainly manifested when the respective computational procedures are considered, especially procedures computing marginal distributions.

The last part of the paper was devoted to an (necessarily brief and informal) introduction to the system MUDIM, which is intended for experimental computations with the introduced multidimensional models. The design of the system, which has been developed under **R** language, enables all users to enlarge the scope of its applications according to their requirements. The author's intention is to develop the system along the following two lines. First, new MUDIM statements extending the application scope of this system will be introduced. The most urgent are those introducing computational procedures: marginalization, computation of conditional probabilities from multidimensional models, and effective implementation of IPFP. Model verification will be supported by possibility to display a *persegram* of a model ([12]), which enables detection of all conditional independence relations holding true for the model.

The other line of the system development can be aimed at generalization of all the procedures in such a way that the system would also be able to construct and process possibilistic multidimensional models, which would be parameterized by a selected t-norm (defining a t-independence in possibility theory). Let us stress, however, that we do not intend to develop a user-friendly software system to enable a lay user to construct probabilistic models just by clicking the mouse.

6 Acknowledgement

The research was partially supported by Grant Agency of the Academy of Sciences of ČR under grant A2075302 and Austrian-Czech grant AKTION KONTAKT 2004/19. The development of a program system MUDIM is currently supported by the grant of the Ministry of Education of the Czech Republic no. 1M6798555601.

References

[1] Cheeseman, P.: A method of computing generalized Bayesian probability values for expert systems. In: *Proc. 6th Int. Conf. on AI (IJCAI 83)*, Karlsruhe, Germany, 1983, pp. 198-202

[2] Cheeseman, P.: In defence of probability. In: Proc. 8th Int. Conf. on AI (IJCAI 85), Los Angeles,

CA, USA, 1985, pp. 1002-1007

[3] Csiszár, I.: I-divergence geometry of probability distributions and minimization problems. *Ann. Probab.*, **3** (1975), pp.146-158

[4] Deming, W.E., Stephan, F.F.: On a least square adjustment of a sampled frequency table when the expected marginal totals are known, *Ann. Math. Stat.* **11** (1940), pp. 427-444

[5] Dubois, D., Prade, H.: *Possibility theory.* Plenum Press, New York, 1988

[6] Hájek, P., Havránek, T., Jiroušek, R.: *Uncertain Information Processing in Expert Systems*. CRC Press, Inc., Boca Raton, 1992

[7] Jensen, F., V.: *Introduction to Bayesian Network*. UCL Press, London, 1996

[8] Jensen, F., V.: *Bayesian Networks and Decision Graphs*. Springer Verlag, 2001

[9] Jiroušek, R.: Composition of probability measures on finite spaces, in: *Proc. of the 13th Conf. Uncertainty in Artificial Intelligence UAI'97*, eds. D. Geiger and P. P. Shenoy, Morgan Kaufmann Publ., San Francisco, California, 1997, pp. 274-281

[10] Jiroušek, R. : Graph Modelling without Graphs. In *Proc. of the 17th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge-based Systems IPMU'98*, Paris (B. Bouchon-Meunier, R.R. Yager, eds.). Editions E.D.K. Paris, 1988, pp. 809-816

[11] Jiroušek, R.: Marginalization in composed probabilistic models. In: *Proc. of the 16th Conf. Uncertainty in Artificial Intelligence UAI'00* (C. Boutilier and M. Goldszmidt eds.), Morgan Kaufmann Publ., San Francisco, California, 2000, pp. 301-308

[12] Jiroušek, R.: Detection of independence relations from persegrams. In: *Proceedings of the 9th Int. Conf. on Information Processing and Management of Uncertainty in Knowledge Based Systems* (Bernadette Bouchon-Meunier, Ronald R. Yager (ed)), ESIA, Annecy, France, 2002, pp. 1261-1267

[13] Jiroušek, R.: Decomposition of multidimensional distributions represented by perfect sequences. *Annals of Mathematics and Artificial Intelligence*, **35**(2002), pp. 215-226

[14] Jiroušek, R.: On Approximating Multidimensional Probability Distributions by Compositional Models. In: *Proceedings of the 3rd Internationa Symposium on Imprecise Probabilities and Their Application*, (J.M. Bernard, T. Seidenfeld, M. Zeffalon, eds.), Carleton Scientific, Ottawa, Canada, 2003, pp. 305-320

[15] Jiroušek, R.: On experimental system for multidimensional model development MUDIM. *Neural Network World*, **13**(2003), pp. 513-520

[16] Lauritzen, S.L.: *Graphical Models*. Clarendon Press, Oxford, 1996

[17] Šimeček, P., Zvárová, J., Tomečková, M., Jiroušek, R.: Application of a compositional model to cardiology. Poster presented at MEDINFO 2004

[18] Vejnarová, J.: Possibilistic independence and operators of composition of possibility measures. In: *Prague Stochastics'98* (M. Hušková, J. Á. Víšek, P. Lachout eds.) JČMF, 1998, pp. 575–580

[19] R Development Core Team: *R Language Definition (Draft).* http://cran.r-project.org/doc/manuals/R-lang.pdf

Received 2 October 2005.