Constructing probabilistic models*

Radim Jiroušek[a,b] and Nicholas Kushmerick[b,c]
[a]Faculty of Informatics and Statistics, VŠE, Prague, Czech Republic
[b]Institute of Information Theory and Automation, AV ČR, Prague, Czech Republic
[c]Department of Computer Science, University of Washington, Seattle, USA
electronic mail: radim@vse.cz, nick@cs.washington.edu

March 6, 1996

Abstract

Bayesian networks have become one of the most popular probabilistic techniques in AI, largely due to the development of several efficient inference algorithms. In this paper we describe a heuristic method for constructing Bayesian networks. Our construction method relies on the relationship between Bayesian networks and decomposable models, a special kind of graphical model. We explain this relationship and then show how it can be used to facilitate model construction. Finally, we describe an implemented computer program that illustrates these ideas.

1 Introduction

Relationships among symptoms, evidence and diseases are often so complex that they can be described only by very general models. Such models can easily be constructed within the framework of probability theory. Several papers in defense of applying probability theory to AI appeared in the early 1980s (*e.g.* [2, 18]), and since then probability theory has been widely accepted as a framework for representing and reasoning about uncertain knowledge. Early objections to the use of probability theory were directed mainly at its high computational complexity. But recently many highly effective representations and algorithms have been developed which, together with the continuously increasing power of computers, has led to numerous commercial applications of probability theory. Two examples serve to demonstrate the popularity of probabilistic models: probability theory is used in the on-line help system in Microsoft's Windows–95 operating system [8]; and Finn Jensen's book *Introduc-*

*This research was supported by grants GA ČR 201/93/0781, GA ČR 201/94/0471, and CEC CIPA3511CT930053.

tion to Bayesian Networks [10] was the top-selling Expert Systems book in May 1995[1].

Bayesian networks, a particular kind of probabilistic model, have become one of the most popular probabilistic technique used in AI for several reasons. Being based on probability theory, they inherit many of the efficient methods and strong results of mathematical statistics. Moreover, numerous efficient methods for their application to inference have been developed [7].

Although in this paper we briefly discuss these inference techniques, we focus mainly on a method for constructing Bayesian networks utilizing both the knowledge of experts and information from data bases.

2 Notation

We consider only finite domains. Let $\vec{X} = \{X_1, \ldots, X_n\}$ denote a vector of n variables taking their values from finite sets $\mathbf{X}_1, \ldots, \mathbf{X}_n$, respectively. In this model we do not distinguish between variables representing symptoms and evidence from those representing diagnoses; from this point of view the model is completely symmetric.

A Bayesian network for the system of variables $\vec{X}$ is defined to be an acyclic directed graph whose nodes are the indices $\{1, \ldots, n\}$, and a system of conditional probability distributions $\{Q_i(X_i|\{X_j\}_{j \in \mathrm{pa}(i)})\}_{i=1}^n$, where $\mathrm{pa}(i)$ denotes the parents of node i: $\mathrm{pa}(i)$ is the set of graph nodes from which there is an edge to node i.

A Bayesian network represents the joint n–dimensional probability distribution $Q(\vec{X})$ which is the product of the conditional distributions assigned to the nodes:

$$Q(\vec{X}) = \prod_{i=1}^{n} Q_i(X_i \mid \{X_j\}_{j \in \mathrm{pa}(i)}).$$

It can be shown that this joint distribution is the only one that is consistent with the given set of conditional probability distributions and simultaneously whose dependence structure reflects all the conditional independence relations given by the underlying graph[2].

Let us consider a simple example, which resembles the well-known example discussed in [17]. The five variables in the domain are described in Table 1, the Bayesian network is defined by the graph in Figure 1, and the conditional distributions are listed in Table 2. This Bayesian network represents the 5–dimensional distribution

$$Q(X_1, \ldots, X_5) = Q_1(X_1)Q_2(X_2)Q_3(X_3 \mid X_1)Q_4(X_4 \mid X_1, X_2)Q_5(X_5 \mid X_3, X_4)$$

[1] Information provided by the Internet Book Shop Mailing Services.

[2] *Consistency* means that for all $i = 1, \ldots, n$, $Q(X_i \mid \{X_j\}_{j \in \mathrm{pa}(i)}) = Q_i(X_i \mid \{X_j\}_{j \in \mathrm{pa}(i)})$. The reader not acquainted with the necessary theory can take as definitional that a Bayesian network represents the joint distribution given by the product of the given conditional distributions.

whose values are listed in Table 3.

variable	*meaning*	*values*
X_1	patient is smoker	0 =no, 1 =yes
X_2	family history of cancer	0 =no, 1 =yes
X_3	patient has bronchitis	0 =no, 1 =yes
X_4	patient has lung cancer	0 =no, 1 =yes
X_5	X-ray indicates disease	0 =no, 1 =yes

Table 1: Variables and their values.

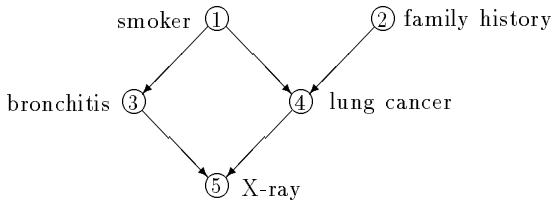

Figure 1: Directed graph of a Bayesian network.

$Q_1(X_1=1)=0.25$	$Q_2(X_2=1)=0.2$
$Q_3(X_3=1 \mid X_1=0)=0.1$	$Q_3(X_3=1 \mid X_1=1)=0.25$
$Q_4(X_4=1 \mid X_1=0, X_2=0)=0.05$	$Q_4(X_4=1 \mid X_1=1, X_2=0)=0.25$
$Q_4(X_4=1 \mid X_1=0, X_2=1)=0.2$	$Q_4(X_4=1 \mid X_1=1, X_2=1)=0.4$
$Q_5(X_5=1 \mid X_3=0, X_4=0)=0.02$	$Q_5(X_5=1 \mid X_3=1, X_4=0)=0.2$
$Q_5(X_5=1 \mid X_3=0, X_4=1)=0.6$	$Q_5(X_5=1 \mid X_3=1, X_4=1)=0.8$

Table 2: Conditional probability distributions.

As just mentioned, $Q(X_1,\ldots,X_5)$ is unique, and it is consistent with the given conditional distributions and the requirements on the dependence structure of the represented distribution. In the example, this fact implies that $Q(X_1,\ldots,X_5)$ encodes the following conditional independences:

- X_2 is independent from X_1 and X_3 (written $[X_2 \perp_Q X_1, X_3]$), meaning that we assume that family history of cancer influences neither whether

			$X_1 = 0$		$X_1 = 1$	
			$X_2 = 0$	$X_2 = 1$	$X_2 = 0$	$X_2 = 1$
$X_3 = 0$	$X_4 = 0$	$X_5 = 0$	0.50274	0.10584	0.1029	0.02058
		$X_5 = 1$	0.01026	0.00216	0.0021	0.00042
	$X_4 = 1$	$X_5 = 0$	0.0108	0.0108	0.014	0.0056
		$X_5 = 1$	0.0162	0.0162	0.021	0.0084
$X_3 = 1$	$X_4 = 0$	$X_5 = 0$	0.0456	0.0096	0.036	0.0072
		$X_5 = 1$	0.0114	0.0024	0.009	0.0018
	$X_4 = 1$	$X_5 = 0$	0.0006	0.0006	0.003	0.0012
		$X_5 = 1$	0.0024	0.0024	0.012	0.0048

Table 3: Joint 5–dimensional probability distribution.

a person is a smoker or suffers from bronchitis. Of course in reality these factors may well not be independent. For example, it may be that one's tendency to smoke is influenced by the fact that one's relatives suffered from cancer. But $Q(X_1, \ldots, X_5)$ contains this independence relationship because it is encoded in the Bayesian network.

- $[X_3 \perp_Q X_2, X_4 | X_1]$: in determining whether a patient suffers from bronchitis, her family history of cancer supplies no information, and neither does the fact that she has lung cancer if it is already known whether she smokes. This independence is conditional because the analysis of a person's chance of getting lung cancer depends on the analysis of whether she smokes which in turn depends on whether she has bronchitis.

- $[X_5 \perp_Q X_1, X_2 | X_3, X_4]$

Naturally, several more independence relations follow from these: $[X_1 \perp_Q X_2]$, $[X_4 \perp_Q X_3 | X_1, X_2]$, and so forth.

Finally, note that the 5–dimensional distribution $Q(X_1, \ldots, X_5)$ represented by the Bayesian network of this example can be stored in computer memory with 12 probabilities (namely, the values shown in Table 2). In contrast, Table 3 illustrates that a general 5–dimensional distribution of binary variables is determined by 31 probabilities; in the general case, an n–dimensional distribution requires storage that grows exponentially in n. Thus a major advantage of Bayesian networks is that they compactly represent distributions over large state spaces.[3]

[3]Note that we take advantage of the fact that the probabilities of a distribution sum to one. Although a polynomial amount of additional storage is needed to represent the graph itself, this requirement is negligible as n becomes large.

3 Inference in Bayesian networks

In probabilistic applications, one typically uses a distribution to represent knowledge of some area of interest, and the problem is to determine what can be concluded from this distribution. Thus methods for using such distributions for inference have been developed. In terms of probability theory, we need effective procedures for computing arbitrary conditional probability distributions. Due to their popularity, several such techniques have been developed for Bayesian networks.

For example, Shachter has developed two transformations on a Bayesian network (*node deletion* and *edge reversal*) that produce a smaller network representing a marginal of the original distribution, thus yielding the required conditional probabilities almost directly [19, 20, 21]. Another approach is based on Lauritzen and Spiegelhalter's well-known method of *local computation* that transforms the original Bayesian network into a *decomposable model* that is more suitable for computations than Bayesian networks [17].

Although these and similar techniques are applicable in realistic problems, their performance varies when applied to different Bayesian networks. It is difficult to give general principles for determining which methods are most suitable for a particular situation. Nevertheless, they all operate on the same underlying probability distribution and therefore the results—the required conditional probabilities—are the same.

In this paper we will not analyze these aspects of Bayesian network theory; the reader is referred to the literature cited above and to [7, 10]. Rather, we will discuss a technique for constructing probabilistic models. Nevertheless, because decomposable models play an important role in the process of constructing Bayesian networks, we now describe Lauritzen and Spiegelhalter's approach in more detail.

4 Decomposable models

In contrast to the directed graphs that define a Bayesian networks, in the sequel we model the domain with undirected graphs. Moreover, graphs with loops or multiple edges are are not allowed.

A *clique* of a graph G= (V, E) is a maximal subset of nodes such that every pair of clique nodes are connected by an graph edge. A graph is *triangulated* if every cycle with length greater than three has a *chord*, a pair of non-consecutive nodes is connected by an edge.

From our point of view, an important property of triangulated graphs is the fact that the cliques of a triangulated graph can be enumerated in such a way

that the ordering $C_1, \ldots, C_m$ meets the *running intersection property* [7]:

$$\forall\ 1 < i \leq m \ \ \exists\ 1 \leq j < i \quad C_i \cap \bigcup_{k=1}^{i-1} C_k \ \subseteq\ C_j .$$

Given such an ordering, we define $B_i = C_i \cap \bigcup_{k=1}^{i-1} C_k$ for $1 < i \leq m$.

We can now define the concept of a probability distribution being decomposable with respect to a triangulated graph. Assume a triangulated graph $G = (V, E)$ with $V = \{1, \ldots, n\}$. Let $C_1, \ldots, C_m$ be the cliques of G and (without loss of generality) assume that these cliques are ordered to meet the running intersection property. We say that the distribution $Q(\vec{X})$ is *decomposable with respect to* G if

$$Q(\vec{X}) = \prod_{i=1}^{m} Q(\{X_j\}_{j \in C_i \setminus B_i} | \{X_j\}_{j \in B_i}).$$

The importance of the concept of decomposability lies in two facts. First, a decomposable distribution is *uniquely* defined by its marginals for the sets of variables corresponding to the cliques of the respective graph. The second fact is that the joint distribution can be expressed as a *product* of its marginals

$$Q(\vec{X}) = \prod_{i=1}^{m} Q(\{X_j\}_{j \in C_i}) \prod_{i=2}^{m} Q^{-1}(\{X_j\}_{j \in B_i}) = \prod_{i=1}^{m} \frac{Q(\{X_j\}_{j \in C_i})}{Q(\{X_j\}_{j \in B_i})}.$$

Note that the marginal $Q(\{X_j\}_{j \in B_i})$ for $B_i = \emptyset$ equals *constant* 1, and we have

$$Q(\{X_j\}_{j \in C_i \setminus B_i} | \{X_j\}_{j \in B_i}) = Q(\{X_j\}_{j \in C_i \setminus B_i})$$

for $B_i = \emptyset$ as well.

To find an undirected graph with respect to which the distribution represented by a Bayesian network is decomposable, Lauritzen and Spiegelhalter's method transforms the original directed graph first into an undirected *moral* graph and then into a triangulated graph. *Moralization* involves connecting all nodes having a common child by undirected edges and then unorienting the original edges. *Triangularization* of a graph is the process of adding new edges to an undirected graph to get a triangulated graph; Tarjan and Yannakakis' algorithm [22] has been widely used.

The goal of this paper is not to teach the reader the method of local computations; the reader is referred to [7, 17]. We do not therefore describe the equations according to which the original conditional probabilities are transformed into a set of distributions on the cliques $C_1, \ldots, C_m$. Nevertheless, let us illustrate the process for the simple example introduced in Section 2.

Moralization of the graph from Figure 1 leads to the graph in Figure 2. This graph does not contain a chordless cycle of length greater than three (there are two cycles of length four: $(1, 3, 5, 4)$ and $(1, 3, 4, 2)$, with chords $(3, 4)$ and $(1, 4)$

respectively; and one cycle of length five, $(1,3,5,4,2)$ having two chords, $(1,4)$ and $(3,4)$) and thus the graph is also triangulated. The respective distributions are uniquely defined by the three 3–dimensional distributions listed in Table 4.

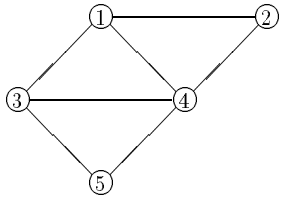

Figure 2: Moral graph of the Bayesian network from Figure 1.

$Q(X_1=0, X_2=0, X_4=0) = 0.57$	$Q(X_1=0, X_2=0, X_4=1) = 0.03$
$Q(X_1=0, X_2=1, X_4=0) = 0.12$	$Q(X_1=0, X_2=1, X_4=1) = 0.03$
$Q(X_1=1, X_2=0, X_4=0) = 0.15$	$Q(X_1=1, X_2=0, X_4=1) = 0.05$
$Q(X_1=1, X_2=1, X_4=0) = 0.03$	$Q(X_1=1, X_2=1, X_4=1) = 0.02$
$Q(X_1=0, X_3=0, X_4=0) = 0.621$	$Q(X_1=0, X_3=0, X_4=1) = 0.054$
$Q(X_1=0, X_3=1, X_4=0) = 0.069$	$Q(X_1=0, X_3=1, X_4=1) = 0.006$
$Q(X_1=1, X_3=0, X_4=0) = 0.126$	$Q(X_1=1, X_3=0, X_4=1) = 0.049$
$Q(X_1=1, X_3=1, X_4=0) = 0.054$	$Q(X_1=1, X_3=1, X_4=1) = 0.021$
$Q(X_3=0, X_4=0, X_5=0) = 0.73206$	$Q(X_3=0, X_4=0, X_5=1) = 0.01494$
$Q(X_3=0, X_4=1, X_5=0) = 0.0412$	$Q(X_3=0, X_4=1, X_5=1) = 0.0618$
$Q(X_3=1, X_4=0, X_5=0) = 0.0984$	$Q(X_3=1, X_4=0, X_5=1) = 0.0246$
$Q(X_3=1, X_4=1, X_5=0) = 0.0054$	$Q(X_3=1, X_4=1, X_5=1) = 0.0216$

Table 4: 3–dimensional distributions defining the decomposable model.

Just as every Bayesian network can be converted into a decomposable model, a decomposable model can always be expressed in the form of a Bayesian network. To do so, the nodes of the decomposable model (*i.e.* the indices of the variables) are ordered so that the corresponding ordering of the cliques meets the running intersection property:

$$\underbrace{1,\ldots,k}_{C_1},\underbrace{k+1,\ldots,\ell}_{C_2\setminus B_2},\underbrace{\ell+1,\ldots}_{C_3\setminus B_3},\ldots,\underbrace{\ldots,n}_{C_m\setminus B_m}\ .$$

The parents of node i are then those nodes which are adjacent to i in the original decomposable model and which are before i in the constructed ordering. The

conditional distributions are directly computed from the marginal distributions assigned to the decomposable model.

5 Iterative proportional fitting procedure

We now use this ability to transform a decomposable model into a Bayesian network as the basis of a simple data-based method for constructing Bayesian networks. Our method requires only knowledge regarding which variables are highly dependent on each other; this can be measured by, for example, information-theoretic measures such as *mutual information* or *informational content*.

Assume a system of *oligodimensional* distributions $\{P_\ell(\{X_j\}_{j\in D_\ell})\}_{\ell=1}^r$ over the sets of variables $D_1,\ldots,D_r$. An oligodimensional distribution is a distribution over a "reasonably" small set of highly mutually-dependent variables. Assume further that the D_ℓ collectively cover the variables: $\bigcup_{\ell=1}^r D_\ell = \{1,\ldots,n\}$.

To find an n–dimensional probability distribution $Q(\vec{X})$ having the given distributions for its marginals—*i.e.* a distribution such that

$$Q(\{X_j\}_{j\in D_\ell}) = P_\ell(\{X_j\}_{j\in D_\ell})$$

for each ℓ—one can use the well known *Iterative Proportional Fitting Procedure* (IPFP), proposed as early as 1940 by W. E. Deming and F. F. Stephan [6].

The mathematical description of this procedure is simple. IPFP is an iterative process, computing from an initial uniform n–dimensional probability distribution R_0 a sequence of distributions $R_1, R_2, \ldots$ of the same dimensions, where the sequence converges to the distribution R^* that has the required marginals. The reader is referred to [5] for a description of the theoretical properties of IPFP.

Each iterative step corresponds to an adjustment of one marginal distribution. The order in which marginals are adjusted is regularly rotated, as expressed in the following recurrent formula

$$R_i(\vec{X}) = P_\ell(\{X_j\}_{j\in D_\ell}) R_{i-1}(\{X_j\}_{j\notin D_\ell} | \{X_j\}_{j\in D_\ell})$$

for $i = 1, 2, \ldots$ and $\ell = 1 + ((i-1) \bmod r)$.

Note that as described so far IPFP computes the values of an n–dimensional distributions, which in general has severe computational complexity. Fortunately, an effective implementation of this procedure has been designed which represents all the distributions—the R_i as well as the resulting limit distribution R^*—as decomposable models [11, 14]. IPFP can thus be applied even in domains involving tens (and in special cases even hundreds) of variables. Moreover, we can take advantage of the fact that the result of this process is a decomposable model.

6 Constructing Bayesian networks

In the previous two sections we described how IPFP can be used to construct a decomposable model from a set of oligodimensional distributions, and how the resulting decomposable model can then be translated to a Bayesian network. To complete the task of constructing a model, one must determine the dependence structure of the model. In this section we describe a method for doing so.

We deal here only with the situation in which one starts with a set of empirically-gathered data, such as a list of symptoms and diseases observed among some population of patients. The goal of the process is to build a Bayesian network that describes this data and compare alternative networks. It is important to note that we are *not* proposing to entirely automate this procedure. Although such techniques have been studied, we focus here on the more modest goal of building a system that can *partially* automate the process.

The first step involves collecting raw data from the domain. This data is in the form of a series of data vectors, each describing one patient's diagnoses and symptoms as well as the results of various laboratory tests. In our example, the vectors indicate whether the respective patients smoke, suffer from bronchitis and lung cancer, have a family history of cancer, and the results of X-ray examinations.

These data are then used to compute probability distributions. As a simple approach, we can assume that the empirical frequencies can be used to estimate the underlying probabilities. Although this approach is often used, we must point out that there are numerous difficulties concerning missing data values [?]. In particular, if missing values are treated improperly, then the computed system of oligodimensional distributions might be such that there does not exist a joint distribution having the given distributions as its marginals.

In some situations, the designer knows the structure of the desired Bayesian network. Then the only problem is to compute estimates of the respective conditional probability distributions. In this case, it is reasonable to use existing statistical techniques to handle missing values and similar problems. However, this situation is very rare. Often the designer must first decide among plausible alternative structures.

6.1 Choosing a structure

It can be very difficult to select the best among several alternative structures. We propose as a criterion for evaluating different models the Kullback-Leibler divergence (cross-entropy) between $P(\cdot)$, the empirical distribution defined by the data, and $Q(\cdot)$, the distribution represented by the model [16]. This divergence is defined

$$C(P,Q) = \sum_{\vec{x}} P(\vec{x}) \log_2 \frac{P(\vec{x})}{Q(\vec{x})}.$$

It is not difficult to show that $C(P,Q)$ can be expressed as follows:

$$\begin{aligned} C(P,Q) &= \sum_{\vec{x}} P(\vec{x}) \log_2 P(\vec{x}) - \sum_{i=1}^{n} \sum_{\vec{x}} P(\vec{x}) \log_2 Q_i(x_i|\{x_j\}_{j \in \mathrm{pa}(i)})) \\ &= H(P(X_1,\ldots,X_n)) \\ &\quad - \sum_{i=1}^{n} \sum_{x_i,\{x_j\}_{j\in\mathrm{pa}(i)}} P(x_i,\{x_j\}_{j\in\mathrm{pa}(i)}) \log_2 \frac{Q_i(x_i|\{x_j\}_{j\in\mathrm{pa}(i)})}{P(x_i)} \\ &\quad + \sum_{i=1}^{n} H(P(X_i)), \end{aligned}$$

where $H(P(\cdot))$ denotes the Shannon entropy of the distribution $P(\cdot)$. It has been shown [16, Theorem 7] that the Kullback-Leibler divergence $C(P,Q)$ is a monotonically decreasing function of

$$\sum_{i=1}^{n} W_P(Q_i(X_i,\{X_j\}_{j\in\mathrm{pa}(i)})),$$

where

$$\begin{aligned} &W_P(Q_i(X_i,\{X_j\}_{j\in\mathrm{pa}(i)})) \\ &\qquad = \sum_{x_i,\{x_j\}_{j\in\mathrm{pa}(i)}} P(x_i,\{x_j\}_{j\in\mathrm{pa}(i)}) \log_2 \frac{Q_i(x_i|\{x_j\}_{j\in\mathrm{pa}(i)})}{P(x_i)}. \end{aligned}$$

It is worth mentioning that this formula can be used directly in spite of the fact that it contains the unknown distribution P because only its oligodimensional marginals $P(x_i,\{x_j\}_{j\in pa(i)})$ are actually used, and they can be reasonably-well estimated from the data.

Finally, note that the larger the sum

$$W(Q) = \sum_{i=1}^{n} W_P(Q_i(X_i,\{X_j\}_{j\in\mathrm{pa}(i)})),$$

the smaller the Kullback-Leibler divergence $C(P,Q)$ and therefore the closer the distribution Q is to the distribution P which generated the data. For this reason, the value $W(Q)$ is used to evaluate suitability of the corresponding Bayesian network as an approximation of the (unknown) distribution P.

6.2 Manual construction of decomposable models

Finally, the designer chooses the best Bayesian network as follows:

1. Select small groups D_ℓ of variables which appear to be highly dependent on each other. The groups should be small so that the data vectors yield accurate estimates of probability distributions for these groups; statistical theory can be consulted to determine how small the distributions need to be in order to achieve a desired level of accuracy.

2. Compute oligodimensional probability distributions P_ℓ for the given sets of variables D_ℓ. For each this distribution $P_\ell(\{X_j\}_{j \in D_\ell})$ compute its *informational content*:

$$I(P_\ell(\{X_j\}_{j \in D_\ell})) = \sum_{\{x_j\}_{j \in D_\ell}} P_\ell(\{x_j\}_{j \in D_\ell}) \log_2 \frac{P_\ell(\{x_j\}_{j \in D_\ell})}{\prod_{j \in D_\ell} P_\ell(x_j)}.$$

3. Now, the experimental nature of the process begins. Repeatedly select a subset of the oligodimensional distributions (preferring those with high informational content), and apply the Iterative Proportional Fitting procedure to them. The resulting decomposable models can then be transformed into Bayesian networks.

4. Having constructed several networks, choose one according to the criterion described in the previous section.

7 Probabilistic Model Editor

The Probabilistic Model Editor (PME) [15] is a computer system designed to realize the ideas developed in this paper. PME allows one to import data vectors from a domain, build and manipulate oligodimensional distributions, and construct and compare decomposable models.

Oligodimensional distributions can be imported directly or they can be estimated from a set of data vectors. PME can directly convert a set of oligodimensional distributions into a decomposable model if they already constitute such a model. Alternatively, IPFP can be run to convert a set of oligodimensional distributions into a decomposable model. Finally, decomposable models can be compared by comparing the respective $W(Q)$ values.

PME can convert a decomposable model into a Bayesian network. Although PME provides no facilities for evaluating Bayesian networks, PME can save a Bayesian network in the format that is used by HUGIN system [10, Chapter 10]. In this sense PME and HUGIN offer complementary services: HUGIN offers a range of tools for evaluating Bayesian networks, while PME facilitates their comparison and construction.

PME provides several extensions to IPFP that facilitate research on model construction. For example, rather than starting with a uniform distribution, PME's implementation of IPFP can be instructed to start with an arbitrary decomposable model that has the same underlying structure. A second extension

permits manual modification of the graph in order to force a desired triangularization. Several extensions have also been made concerning IPFP's termination condition, permitting the system to detect some cases in which IPFP will provably never converge.

A number of further extensions have been implemented. Although not necessarily useful in real applications, these extensions are likely to be helpful to the model-construction research community. For example, PME allows one to use a decomposable model as a simulator to generate data vectors, or to explicitly compute the (potentially very large) joint distribution encoded by a decomposable model.

The Probabilistic Model Editor is fully implemented in C++, and runs in the Microsoft Windows environment. Executables, complete source code, data files for the example described in this paper, and a detailed user manual are all available on request.

References

[1] R. R. Bouckaert: Probabilistic network construction using the minimum description length principle. In: Transactions of ESQARU'93, Granada, 1993, pp 41–48.

[2] P. Cheeseman: In defense of probability. In: Proc. 8th Int. Conf. on AI (IJCAI'85), Los Angeles, CA, 1985, pp 1002–1007.

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

[4] P. Cheeseman: In defense of probability. In: Proc. 8th Int. Conf. on AI (IJCAI 85), Los Angeles, CA, 1985, pp 1002-1007

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

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

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

[8] D. Heckerman, J. Breese, J. Rommelse: Decision Theoretic Troubleshooting. *Communication of the ACM*, March 1995.

[9] D. Heckerman, D. Geiger, D. M. Chickering: Learning Bayesian networks: the combination of knowledge and statistical data. Techn. Rep. MSR-TR-94-09, Microsoft Research, Advanced Technology Division, Microsoft Co., Redmond, WA, 1994.

[10] F. Jensen: Introduction to Bayesian Network. UCL Press, 1995.

[11] R. Jiroušek: Solution of the marginal problem and decomposable distributions. Kybernetika, 27, pp 403-412 (1991)

[12] R. Jiroušek: From learning Bayesian networks to a decomposable model construction. In: Papers and Abstracts from the Fifth Annual Workshop on Normative Systems. George Mason University, Fairfax, Virginia, 1995, pp 8–10.

[13] R. Jiroušek, G. D. Kleiter: A note on learning Bayesian networks. In: 8th European Conf. on Machine Learning. Workshop notes: Statistics, Machine Learning and Knowledge Discovery in Databases (Y. Kodratoff, G. Nakhaeizadeh, C. Taylor, eds.), Heraklion, Greece, 1995, pp 148–153.

[14] R. Jiroušek, Stanislav Přeučil: On the effective implementation of the iterative proportional fitting procedure. *Comput. Stat. and Data Analysis*, 19, pp 177–189 (1995).

[15] N. Kushmerick: The Probabilistic Model Editor User's Manual. Technical Report, Institute of Information Theory and Automation, Academy of Science of the Czech Republic, Prague, 1995.

[16] W. Lam, F. Bacchus: Learning Bayesian belief networks: an approach based on the MDL principle. *Computational Intelligence*, 10, pp 269–293 (1994).

[17] S. L. Lauritzen, D. Spiegelhalter: Local computations with probabilities on graphical structures and their applications to expert systems. J. Roy. Stat. Soc. Ser. B., 50, pp 157–189 (1988).

[18] A. Perez: A probabilistic approach to the integration of partial knowledge for medical decision-making. (in Czech) In: Proc. 1st Czechoslovak Congress of Biomed. Eng. (BMI 83), Mariánské Lázně, 1983, pp 221–226.

[19] R. D. Shachter: Evaluating influence diagrams. Oper. Res., 34, pp 871–882 (1986).

[20] R. D. Shachter: Intelligent probabilistic inference. In: L. N. Kanal, J. F. Lemmer (eds.): Uncertainty in Artificial Intelligence. North Holland, Amsterdam, pp 371–382, 1986.

[21] R. D. Shachter: Probabilistic inference and influence diagrams. Oper. Res., 36, pp 589–604 (1988).

[22] R. E. Tarjan, M. Yannakakis: Simple linear-time algorithms to test chordality of graphs, test acyclidity of hypergraphs, and selectively reduce acyclic hypergraphs. SIAM J. Comput, 13, pp 566-591 (1984)