INFERENCE IN HYBRID BAYESIAN NETWORKS

Helge Langseth, [a] Thomas D. Nielsen, [b] Rafael Rumí, [c] and Antonio Salmerón [c]

[a] *Department of Information and Computer Sciences, Norwegian University of Science and Technology, Norway*

[b] *Department of Computer Science, Aalborg University, Denmark*

[c] *Department of Statistics and Applied Mathematics, University of Almería, Spain*

Abstract

Since the 1980s, Bayesian Networks (BNs) have become increasingly popular for building statistical models of complex systems. This is particularly true for boolean systems, where BNs often prove to be a more efficient modelling framework than traditional reliability-techniques (like fault trees and reliability block diagrams). However, limitations in the BNs' calculation engine have prevented BNs from becoming equally popular for domains containing mixtures of both discrete and continuous variables (so-called *hybrid* domains). In this paper we focus on these difficulties, and summarize some of the last decade's research on inference in hybrid Bayesian networks. The discussions are linked to an example model for estimating human reliability.

1 Introduction

A reliability analyst will often find himself making decisions based on uncertain information. Examples of decisions he may need to take include defining a maintenance strategy or choosing between different system configurations for a safety system. These decisions are typically based on only limited knowledge about the failure mechanisms that are in play and the environment the system will be deployed in. This uncertainty, which can be both aleatory and

Email addresses: helgel@idi.ntnu.no (Helge Langseth,), tdn@cs.aau.dk (Thomas D. Nielsen,), rrumi@ual.es (Rafael Rumí,), antonio.salmeron@ual.es (Antonio Salmerón).

Preprint submitted to Reliability Engineering & System Safety 11 February 2009

epistemic, requires the analyst to use a statistical model representing the system in question. This model must be mathematically sound, and at the same time easy to understand for the reliability analyst and his team. To build the models, the analyst can employ different sources of information, e.g., historical data or expert judgement. Since both of these sources of information can have low quality, as well as come with a cost, one would like the modelling framework to use the available information as efficiently as possible. Finally, the model must be encoded such that the quantities we are interested in (e.g., the availability of a system) can be calculated efficiently.

All of these requirements have lead to a shift in focus, from traditional frameworks, like fault trees, to more flexible modeling frameworks. One such framework for building statistical models for complex systems is the Bayesian network (BN) framework [1–3]. BNs have gained popularity over the last decade [4], partly because a number of comparisons between BNs and the classical reliability formalisms have shown that BNs have significant advantages [5–10].

BNs consist of a qualitative part, an *acyclic directed graph*, where the nodes mirror stochastic variables and a quantitative part, a set of conditional probability functions. An example of the qualitative part of a BN is shown in Fig. 1. This BN models the risk of an explosion in a process system. An explosion (*Explosion?*) might occur if there is a leak (*Leak?*) of chemical substance that is not detected by the gas detection (GD) system. The GD system detects all leaks unless it is in its failed state (*GD Failed?*). The environment (*Environment?*) will influence the probability of a leak as well as the probability of a failure in the GD system. Finally, an explosion may lead to a number of casualties (*Casualties?*).

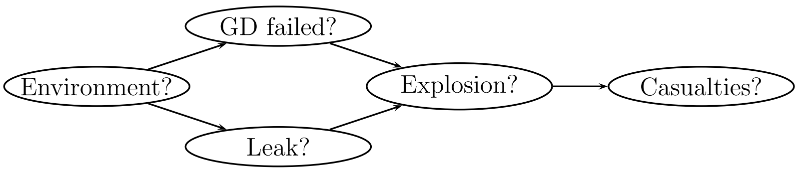


Fig. 1. An example BN describing a gas leak scenario. Only the qualitative part of the BN is shown.

The graphical structure has an intuitive interpretation as a model of causal influences. Although this interpretation is not necessarily entirely correct, it is helpful when the BN structure is to be elicited from experts. Furthermore, it can also be defended if some additional assumptions are made [11].

BNs originated as a robust and efficient framework for reasoning with uncertain knowledge. The history of BNs in reliability can (at least) be traced back to [12,13]; the first real attempt to use BNs in reliability analysis is probably the work of Almond [13], where he used the Graphical-Belief tool to calculate reliability measures concerning a low pressure coolant injection system for a nuclear reactor (a problem originally addressed by Martz [14]). BNs constitute a modelling framework which is particularly easy to use in interaction with domain experts, also in the reliability field [15]. BNs have found applications in, e.g., fault detection and identification, monitoring, software reliability, troubleshooting systems, and maintenance optimization. Common to these models are that all variables are *discrete*. As we shall see in Section 3, there is a purely technical reason why most BN models fall in this class. However, in Section 4 we introduce a model for human reliability analysis, where both discrete and continuous variables are in the same model. Attempts to handle such models are considered in Section 5 and we conclude in Section 6.

2 Preliminaries

Mathematically, a BN is a compact representation of a joint statistical distribution function. A BN encodes the probability density function governing a set of variables by specifying a set of conditional independence statements together with a set of conditional probability functions.

For notational convenience, we consider the variables $\{X_1, \ldots, X_n\}$ when we make general definitions about BNs in the following, and we use the corresponding lower-case letters when referring to instantiations of these variables. Now, we call the nodes with outgoing edges pointing into a specific node the *parents* of that node, and say that X_j is a *descendant* of X_i if and only if there exists a directed path from X_i to X_j in the graph. In Fig. 1, *Leak?* and *GD Failed?* are the parents of *Explosion?*, written $\mathrm{pa}\,(\textit{Explosion?}) = \{\textit{Leak?}, \textit{GD Failed?}\}$ for short. Furthermore, $\mathrm{pa}\,(\textit{Casualties?}) = \{\textit{Explosion?}\}$. Since there are no directed paths from *Casualties?* to any of the other nodes, the descendants of *Casualties?* are given by the empty set and, accordingly, its non-descendants are $\{\textit{Environment?}, \textit{GD Failed?}, \textit{Leak?}, \textit{Explosion?}\}$. The edges of the graph represent the assertion that a variable is conditionally independent of its non-descendants in the graph given its parents in the same graph. The graph in Fig. 1 does for instance assert that for all distributions compatible with it, we have that $\{\textit{Casualties?}\}$ is conditionally independent of $\{\textit{Environment?}, \textit{GD fails?}, \textit{Leak?}\}$ when conditioned on $\{\textit{Explosion?}\}$.

When it comes to the quantitative part, each variable is described by the conditional probability function (CPF) of that variable *given the parents* in the graph, i.e., the collection of CPFs $\{f(x_i|\mathrm{pa}\,(x_i))\}_{i=1}^n$ is required. The underlying assumptions of conditional independence encoded in the graph allow us to calculate the joint probability function as

$$f(x_1, \ldots, x_n) = \prod_{i=1}^{n} f(x_i|\mathrm{pa}\,(x_i)), \tag{1}$$

i.e., that the joint probability can be completely expressed as the product of a collection of local probability distributions. This is in fact the main point when working with BNs: Assume that a distribution function $f(x_1, \ldots, x_n)$ factorizes according to Equation (1). This defines the parent set of each X_i, which in turn defines the graph, and from the graph we can read off the conditional independence statements encoded in the model. As we have seen, this also works the other way around, as the graph defines that the joint distribution *must* factorize according to Equation (1). Thus, the graphical representation is bridging the gap between the (high level) conditional independence statements we want to encode in the model and the (low level) constraints this enforces on the joint distribution function. After having established the full joint distribution over $\{X_1, \ldots, X_n\}$ (using Equation (1)), any marginal distribution $f(x_i, x_j, x_k)$, as well as any conditional distribution $f(x_i, x_j|x_k, x_\ell)$, can in principle be calculated using extremely efficient algorithms. These will be considered in Section 3.

We will use $\mathbf{X} = \{X_1, \ldots, X_n\}$ to denote the set of variables in the BN. If we want to make explicit that some variables are observed, we use $\mathbf{E} \subset \mathbf{X}$ to denote the set of observed variables, and $\mathbf{e}$ will be used for the observed value of $\mathbf{E}$. We will use Ω_{X_i} to denote the possible values a variable X_i can take. If X_i is discrete, then Ω_{X_i} is a countable set of values, whereas when X_i is continuous, $\Omega_{X_i} \subseteq \mathbb{R}$. For $\mathbf{X} = \{X_1, \ldots, X_n\}$ we have $\Omega_{\mathbf{X}} = \times_{i=1}^{n} \Omega_{X_i}$.

3 Inference

In this paper we will only consider a special type of inference, namely the case of updating the marginal distributions of some variables of interest given that the values of some other variables are known, e.g., to compute the conditional density of $X_i \in \mathbf{X} \setminus \mathbf{E}$ given the observation $\mathbf{E} = \mathbf{e}$, denoted $f(x_i|\mathbf{e})$. Observe that

$$f(x_i|\mathbf{e}) = \frac{f(x_i, \mathbf{e})}{f(\mathbf{e})},$$

and since the denominator $f(\mathbf{e})$ does not depend on x_i, the inference task is therefore equivalent to obtaining $f(x_i, \mathbf{e})$ and normalizing afterwards. A brute force algorithm for carrying out this type of inference could be as follows:

(1) Obtain the joint distribution $f(x_1, \ldots, x_n)$ using Equation (1).
(2) Restrict $f(x_1, \ldots, x_n)$ to the value $\mathbf{e}$ of the observed variables $\mathbf{E}$, thereby obtaining $f(x_1, \ldots, x_n, \mathbf{e})$.
(3) Compute $f(x_i, \mathbf{e})$ from $f(x_1, \ldots, x_n, \mathbf{e})$ by marginalizing out every variable except X_i.

A problem with this naïve approach is that the joint distribution is usually unmanageably large. For instance, assume a simple case in which we deal with 10 discrete variables that have three states each. Specifying the joint distribution for those variables would be equivalent to defining a table with $3^{10} - 1 = 59\,048$ probability values, i.e., the size of the distribution grows exponentially with the number of variables. For instance, for 11 variables, the size of the corresponding table would increase to $3^{11} - 1 = 177\,146$, and so on. Models used in reliability domains commonly consist of hundreds or thousands of variables, and this naïve inference approach is simply not able to handle problems of this size.

The inference problem can be simplified by taking advantage of the factorization of the joint distribution encoded by the structure of the BN, which supports the design of efficient algorithms for this task. For instance, consider the network in Fig. 2, which is structurally equivalent to the model in Fig. 1; the variables are labelled $X_1, \ldots, X_5$ for notational convenience.

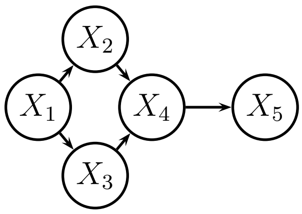


Fig. 2. An example of Bayesian network.

For this example, assume we are interested in X_5, that all variables are discrete, and that $\mathbf{E} = \emptyset$. By starting from the joint distribution, we find that

$$\begin{aligned}
f(x_5) &= \sum_{x_1,\ldots,x_4} f(x_1, x_2, x_3, x_4, x_5) \\
&= \sum_{x_1,\ldots,x_4} f(x_1) f(x_2|x_1) f(x_3|x_1) f(x_4|x_2, x_3) f(x_5|x_4) \\
&= \sum_{x_2,\ldots,x_4} \sum_{x_1} f(x_1) f(x_2|x_1) f(x_3|x_1) f(x_4|x_2, x_3) f(x_5|x_4) \\
&= \sum_{x_2,\ldots,x_4} f(x_4|x_2, x_3) f(x_5|x_4) \sum_{x_1} f(x_1) f(x_2|x_1) f(x_3|x_1) \\
&= \sum_{x_2,\ldots,x_4} f(x_4|x_2, x_3) f(x_5|x_4) h(x_2, x_3),
\end{aligned} \tag{2}$$

where $h(x_2, x_3) = \sum_{x_1} f(x_1) f(x_2|x_1) f(x_3|x_1)$. Therefore, we have reached a similar problem as initially, but with one variable less. Note that this operation, called *elimination* of X_1, only requires us to consider 3 variables at the same time (namely, X_1, X_2 and X_3), instead of all 5 variables. Repeating the same procedure for all variables except X_5 would lead us to the desired result. This procedure is known as the *variable elimination algorithm* [16–18]. Thus, the idea that distinguishes this approach from the naïve approach outlined above, is to organize the operations among the conditional distributions in the network, so that we do not manipulate distributions that are unnecessarily large. One limitation of the variable elimination algorithm, as formulated above, is that it has to be repeated for each variable of interest. This is overcome in other inference algorithms, e.g., [19].

Regardless of which algorithm that is used for implementing the inference process, there are three basic operations involved, which will be defined below. In the definitions we use for subset $\mathbf{Y} \subseteq \mathbf{X}$ the notation $\mathbf{x}^{\downarrow \Omega_{\mathbf{Y}}}$ to denote the sub-vector of $\mathbf{x}$, which is defined on $\Omega_{\mathbf{Y}}$ (i.e., dropping all coordinates not in $\boldsymbol{Y}$).

Restriction is used for inserting the values of the observed variables. Formally, the restriction of a function f to the values $\mathbf{x}' \subset \mathbf{x}$ is a new function defined on $\Omega_{\mathbf{X} \setminus \mathbf{X}'}$ s.t.:

$$f(\mathbf{w}) = f^{R(\mathbf{X}'=\mathbf{x}')}(\mathbf{x})$$

for all $\mathbf{w} \in \Omega_{\mathbf{X} \setminus \mathbf{X}'}$ such that $\mathbf{x} \in \Omega_{\mathbf{X}}$, $\mathbf{x}' = \mathbf{x}^{\downarrow \Omega_{\mathbf{X}'}}$ and $\mathbf{w} = \mathbf{x}^{\downarrow \Omega_{\mathbf{X} \setminus \mathbf{X}'}}$. Restriction is used to obtain a probability distribution over the variables $\mathbf{X}$ when $\mathbf{E} = \mathbf{e}$, which in this notation will be written as $f(\boldsymbol{w}) = f^{R(\mathbf{E}=\mathbf{e})}(\boldsymbol{x})$ for $\boldsymbol{w} \in \Omega_{\mathbf{X} \setminus \mathbf{E}}$.

Combination is the multiplication of two functions; this operation is used, e.g., when the conditional probability functions $\{f_{X_i}(x_i | \mathrm{pa}\,(x_i))\}_{i=1}^{n}$ are multiplied in Equation (1). More formally, let us consider two probability functions f_1 and f_2 defined for $\mathbf{X}_1$ and $\mathbf{X}_2$ respectively. The *combination* of f_1 and f_2 is a new function defined on $\mathbf{X} = \mathbf{X}_1 \cup \mathbf{X}_2$ s.t.

$$f(\mathbf{x}) = f_1(\mathbf{x}^{\downarrow \Omega_{\mathbf{X}_1}}) \cdot f_2(\mathbf{x}^{\downarrow \Omega_{\mathbf{X}_2}}) \quad \forall \mathbf{x} \in \Omega_{\mathbf{X}}.$$

Elimination is used to remove a variable from a function; an example of elimination is seen in Equation (2), where the variable X_5 is eliminated from the distribution $f(x_1, \ldots, x_5)$. Analogously to elimination, we also talk about *marginalization*. For example, $\{X_1, X_2, X_3, X_4\}$ are marginalized out of $f(x_1, \ldots, x_5)$ in Equation (2). Formally, we say that the *marginal* of f over a set of variables $\mathbf{X}' \subseteq \mathbf{X}$ is the function computed as

$$f(\mathbf{x}') = \sum_{\mathbf{x}:\mathbf{x}^{\downarrow \Omega_{\mathbf{X}'}} = \mathbf{x}'} f(\mathbf{x}),$$

Note that this function is defined on $\Omega_{\mathbf{X}'}$. If some of the variables in $\mathbf{X} \setminus \mathbf{X}'$ are continuous, the summation is replaced by an integration over those variables.

So far we have considered inference for discrete variables whose distribution can be represented by a table of probability values. This representation is very convenient from an operational point of view, as restriction, combination, and marginalization are closed for probability tables. It means that all the operations required during the inference process can be carried out using a single unique data structure. The problem becomes more complex when we face inference tasks that involve continuous variables.

Let us for instance consider the problem of calculating the reliability of a parallel system of three components. The components have life-lengths T_1, T_2 and T_3 respectively, and the system's life-length is thus given as $R = \max(T_1, T_2, T_3)$. We assume that each of the component's life lengths follow the exponential distribution with the parameter λ_i, so the survival probability at system level is $P(R \leq t) = \prod_{i=1}^{3} P(T_i \leq t) = \prod_{i=1}^{3} (1 - \exp(\lambda_i t))$, see Fig. 3 (a).

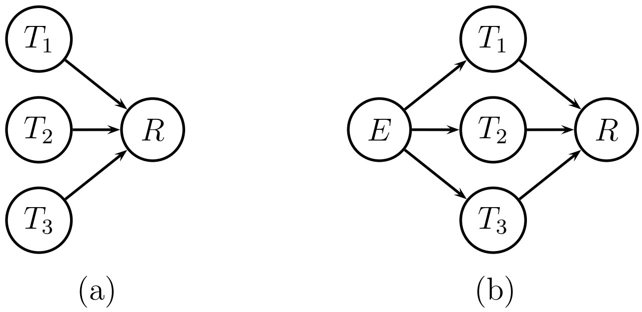


Fig. 3. Three components in a parallel system have life-lengths T_1, T_2 and T_3 respectively, giving the system a life-length of $R = \max(T_1, T_2, T_3)$.

A problem with this model is that the life lengths of the three components are considered independent, even if the components are exposed to the same environment. Obviously, a common environment introduces a correlation between T_1, T_2 and T_3: A rough environment will lead to reduced life-lengths for all components, whereas a gentle environment would imply an increase in the expected life-lengths of the components. Several researchers have been trying to overcome this defect by explicitly modelling the environment-induced correlation between components' life-lengths (see, e.g., [20–22]). We will now consider a candidate-solution to this problem, described by Lindley and Singpurwalla [21]. The authors assumed that when the components are operating in a controlled laboratory environment, their life-lengths T_i follow an exponential distribution with known parameter λ_i. To model the effect of the common environment, they introduced a random variable E affecting each T_i, see Fig. 3 (b). They assumed that E follows a Gamma distribution, and that $T_i|\{E = \xi\}$ is exponentially distributed with parameter $g_i(\xi; \lambda_i)$ for known functions $g_i(\xi; \lambda_i) = \lambda_i \xi$. These assumptions made them able to derive the marginal distribution of R when E is unobserved. However, it should be clear that we can make this problem analytically intractable, simply by choosing "difficult" functions $g_i(\xi; \lambda_i)$, for instance $g_1(\xi; \lambda_1) = \lambda_1\sqrt{\xi}$, $g_2(\xi; \lambda_1) = \lambda_2\xi$, and $g_3(\xi; \lambda_1) = \lambda_3\xi^2$. From an implementation point of view we now need to represent exponentials of more complex functions, but a more fundamental problem is that the results are no longer available analytically, so exact inference cannot be performed. Note that this is a consequence not of the modelling language, *but of the model itself*. Nevertheless, the simplicity of making Bayesian network models does not go well together with the difficulties of inference in the models, and restricting our attention to models containing only discrete variables seems very unsatisfactory in the domain of reliability analysis. This is why a lot of research is currently put into approximative methods for inference in hybrid Bayesian networks.

The most common approach to approximate inference among BN practitioners is *discretization*, i.e., to “translate” all continuous variables into discrete ones (we assume the reader has some familiarity with this concept; more detail is given in Section 5.1). The continuous variables are to be replaced by discrete variables, where the discrete variables are given a sufficient number of states to capture the true (continuous) variables sufficiently well. The problem with this approach is to balance the desire for high accuracy in the approximations with a reasonable calculation burden to obtain the results. Obviously, the accuracy of the approximations are particularly important in reliability applications, where the *tails* of the distributions receive a lot of attention. Say we are interested in calculating the survival function of the system, i.e. $P(R \leq t)$, and, in particular, we are concerned about the lower tail of the life length distribution of R. If we naïvely discretize each continuous variable into d states, then the operations for inference will need to handle d^4 numbers at once (c.f. Equation (2)). d must be chosen sufficiently large to convey enough information to find the (approximately) correct probability, and even refined discretzation techniques (like [23]) require $d \sim 30$ to obtain sufficiently accurate results, and thus need to perform sums over about 800.000 numbers to calculate $P(R > t_0)$ for a given t_0. If the parallel system had 10 components instead of 3, the sum would be over approximately $30^{11} = 2 \cdot 10^{16}$ numbers, which is intolerable in practice.

To conclude this section, exact inference in Bayesian networks require the three operations *restriction*, *combination*, and *elimination*. From a fundamental point of view we must make sure we can perform the operations analytically, and from a practical point of view it is beneficial if a single data structure can represent all intermediate results of these operations. It is not difficult to find examples were the requirements fail, particularly when some of the variables in the domain are continuous. In these cases, the most-used survival-strategy is to discretize the continuous variables, but as we just saw, this will typically either increase the computational complexity unbearable or give approximations with unacceptably poor quality. It is evident that models containing both discrete as well as continuous variables are of high interest to the reliability community, and we will therefore spend the remainder of the paper looking at the most powerful methods for approximate inference in Bayesian networks from the reliability analyst’s point of view (meaning that we are also interested in accurate approximation of the probability of infrequent events, like major accidents).

We proceed by discussing a model for human reliability in Section 4. Not only is this model of interest to us in its own right, but it is also quite simple, and it relies on only a few standard statistical distributions in its specification. This makes the model well-suited as a test–bed when we discuss the different methods for approximate inference in Section 5.

4 A model for human reliability

In this section we will consider a model used for explaining and predicting humans' ability to perform specific tasks in a given environment. The model is based on the THERP methodology [1] .

Consider the BN model in Fig. 4. T_i represents a person's ability to correctly perform task i, and T_i takes on the values "true" or "false". T_i is influenced by a set of explanatory variables, Z_j. The goal of the model is to quantify the effect the explanatory variables have on the observable ones, and to use this to predict a subject's ability to perform the tasks $T_1, \ldots, T_4$.

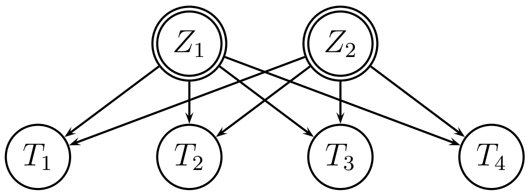


Fig. 4. A model for the analysis of human reliability. A subject's ability to perform four different tasks $T_1, \ldots, T_4$ is influenced by the two explanatory variables Z_1 and Z_2. The explanatory variables are drawn with double-line to signify that these variables are continuous.

Assume first that the explanatory variables are used to model the environment, that the environment can be considered constant between subjects, and that it can be disclosed in advance (that is, the variables are observed before inference is performed). An example of such a factor can for instance be *"Lack of lighting"*, with the assumption that the luminous flux can be measured in advance, and that it affects different people in the same way. Each T_i is modelled by logistic regression, meaning that we have

$$P(T_i = \text{true}|\boldsymbol{z}) = \frac{1}{1 + \exp\left(-\left(\boldsymbol{w}_i'\boldsymbol{z} + b_i\right)\right)}, \tag{3}$$

[1] THERP: Technique for Human Error Rate Prediction [24]

for a given set of weights $\boldsymbol{w}_i$ and offset b_i. As long as $\boldsymbol{Z} = \boldsymbol{z}$ is observed, this is a simple generalized linear model. Therefore, inference in this model can be handled; note that the $\boldsymbol{Z}$'s can be regarded simply as tools to fill in the probability tables for each T_i in this case.

Next, assume that some of the explanatory variables are used to model subject-specific properties, like a subject's likelihood for *"Omitting a step in a procedure"* (this is one of the explanatory variables in the THERP method). It seems natural to assume that these explanatory variables are unobserved, and for the case of simplicity, we give them Gaussian distributions a priori, $Z_j \sim \mathcal{N}(\mu_j, \sigma_j^2)$. To this end, the model is a *latent trait* model [25]; closely related to a factor analysis model, but with binary attributes.

Assume we have parameters $\boldsymbol{w}_i$ determining the strength of the influences the explanatory variables have on T_i, and that we are interested in calculating the likelihood of an observation $\{T_1 = 1, T_2 = 1, T_3 = 1, T_4 = 1\}$. (We will use the shorthand $\boldsymbol{T} = \mathbf{1}$ to denote the observation in the following.) The likelihood is given by

$$P(\boldsymbol{T} = \mathbf{1}) = \frac{1}{2\pi\sigma_1\sigma_2} \int_{\mathbb{R}^2} \frac{\exp\left(-\sum_{j=1}^{2} \frac{(z_j-\mu_j)^2}{2\sigma_j^2}\right)}{\prod_{i=1}^{4}\left\{1 + \exp(-\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{z} - b_i)\right\}}\, d\boldsymbol{z},$$

which unfortunately has no known analytic representation in general. Hence, we are confronted by a model where we cannot calculate the exact likelihood of the observation. We will now turn to some of the state-of-the-art techniques for approximate inference in Bayesian networks containing discrete and continuous variables, and use the model in Fig. 4 as our running example while doing so.

5 Approximative inference for hybrid Bayesian networks

As we saw in the previous section, exact inference is not tractable in the example model. The mathematical tools are simply not available for the calculations to be made. In this section we will therefore cover some of the more popular ways of approximating the inference procedure. In Section 5.1, Section 5.2, and Section 5.3, we consider three approaches that make explicit changes in the representations of the conditional distribution functions defined for each variable, then in Section 5.4 we will consider a scheme that leaves the underlying model definition unchanged, but uses sampling to approximate the inference procedure.

We will use the model in Fig. 4 as our running example, and for each approximative method we will calculate both the likelihood of the observation,

$P(\boldsymbol{T} = \mathbf{1})$, as well as the posterior distribution over the explanatory variables, $f(\boldsymbol{z}|\boldsymbol{T} = \mathbf{1})$. For simplicity, we assume that $\boldsymbol{w} = [1, 1]^{\mathrm{T}}$ and that $b_i = 0$, so Equation (3) can be simplified to $P(T_i = \text{true}|\boldsymbol{z}) = (1 + \exp(-z_1 - z_2))^{-1}$.

5.1 Discretization

The most common technique for handling inference in hybrid Bayesian networks is probably *discretization.* Discretization has been widely studied from both a general point of view [26,27], and aimed specifically at BNs [28,29] and classification problems [30,31]. Discretization amounts to replacing a continuous variable X in a model by its *discrete* counterpart X'. X' is obtained from X by separating Ω_X into disjoint intervals, that can formally be described as follows:

Definition 1 (Discretization) *A discretization of an interval $\Omega_X \subseteq \mathbb{R}$ is a partitioning of Ω_X into a finite set of disjoint connected regions $\{W_j : j = 1, \ldots, m\}$, where $W_i \cap W_j = \emptyset$ and $\cup_{j=1}^m W_j = \Omega_X$. Each W_j is labelled with a constant positive real value, $f_D(W_j)$, which denotes the* value *at the interval W_j.*

An example of discretization is shown in the left panel of Fig. 5. X follows the standard Gaussian distribution; the discretized version of X, X', has density function $f_D(x')$. Notice that $f_D(x')$ is a step-function, i.e., piece-wise constant.

After discretization, X' replaces X in the model, and can be handled as any other discrete variable. X' is defined such that its value is the same whenever X falls in the interval W_i. There are a number of different strategies for selecting the regions W_i, for example equal width, equal frequency, and even Bayesian approaches to mention a few.

Note that as long as m, the number of partitions, is "low" compared to the length of Ω_X, discretization may entail lack of accuracy, as $f_D(x')$ can be a poor approximation of $f(x)$. On the other hand, the distribution of X' can be made arbitrarily close to the one of X as $m \to \infty$. Unfortunately, though, introducing too many states in the variables may lead to an unfeasible problem from an inference point of view. In Equation (2) we saw that marginalizing amounts to summing over all states of the unobserved variables (which grows as m increases), and since the complexity depends on the size of the largest function handled during the variable elimination process, the computational burden may grow uncontrolled. Moreover, even though discretization can be a good choice to control the error in each local distribution $f(x_i|\text{pa}(x_i))$, it may not control the global error in the model, see, e.g., [23] for a discussion. This problem is in part due to the fact that many of the most common approaches discretize each variable independently, without considering the dependence relations in the graph. If one takes these relations into account, one would pay more attention to regions of the multivariate space where changes in the joint probability distribution is large, both a priori and also after inserting evidence [29].

Let us return to the example described in Section 4, and use discretization to be able to perform the inference in that model. Firstly, we need to discretize the distributions for Z_i, $i = 1, 2$. For the purpose of this example, we select 5 regions by means of equal length. We consider the domain $[-\frac{5}{2}, +\frac{5}{2}]$ in the following. This residual mass is allocated to the two extremes, and we get the following approximation:

W_j	$[-\frac{5}{2}, -\frac{3}{2}]$	$(-\frac{3}{2}, -\frac{1}{2}]$	$(-\frac{1}{2}, +\frac{1}{2}]$	$(\frac{1}{2}, \frac{3}{2}]$	$(\frac{3}{2}, \frac{5}{2}]$
$f_D(W_j)$	0.067	0.242	0.383	0.242	0.067

Fig. 5 (left panel) shows the original distribution of the latent variables together with the corresponding discretized version.

Recall that the conditional distribution $P(T_i = 1|Z_1, Z_2)$ is defined using logistic regression (Equation (3)). To use this definition, we need the discretized value to have a numerical representation. We obtain this by using the mid-points of each interval as the numerical interpretation, i.e., $\Omega_{Z_i'} = \{-2, -1, 0, 1, 2\}$.

Doing the calculations, we obtain that $P(\boldsymbol{T} = \mathbf{1}) = .176999$. The correct value is approximately $P(\boldsymbol{T} = \mathbf{1}) = .173865$, so the result is not too far off. However, the results are poorer if we are interested in the joint distribution $f(\boldsymbol{z}|\boldsymbol{T} = \mathbf{1})$, see Fig. 5, right panel. Note particularly the poor fit in the tail of the distribution.

Most of the software tools available for modelling Bayesian networks allow continuous variables to be discretized. This holds for instance, for Agena (`www.agena.co.uk`), Netica (`www.norsys.com`), Hugin (`www.hugin.com`), Elvira (`leo.ugr.es/elvira`), and Genie (`genie.sis.pitt.edu`).

5.2 Mixtures of truncated exponentials

The Mixtures of Truncated Exponentials (MTE) model [32] can be seen as a generalization of discretization, but instead of approximating the density function inside each region by a constant, MTEs approximate it by a linear combination of exponential functions; the benefit from this approach is the higher flexibility when it comes to approximating the distribution function.

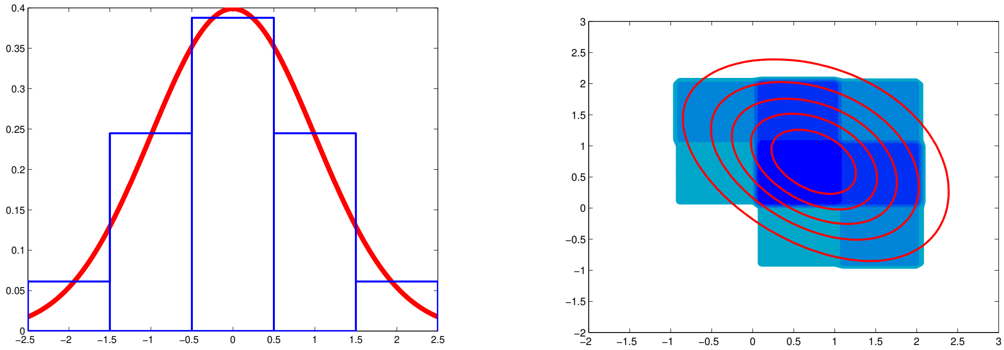


Fig. 5. Results of discretization in our running example. The left pane gives the approximation of the Gaussian distribution, whereas the right panel shows the approximation of $f(z_1, z_2|\boldsymbol{T} = \mathbf{1})$ together with the exact results (obtained by numerical integration). Note the poor approximation to the joint posterior distribution, particularly in the tail of the distribution.

So, in the MTE approach, the density functions (conditional or marginal) are represented by means of MTEs. A *potential* is a generalization of a density function, where it is not required that the integral equals one. A potential is an interesting structure because not every function involved in BN inference is a density. We therefore start by defining MTE potentials:

Definition 2 (MTE Potential) *Let* $\mathbf{X}$ *be a mixed* n*-dimensional random vector. Let* $\mathbf{Y} = (Y_1, \ldots, Y_d)$ *and* $\mathbf{Z} = (Z_1, \ldots, Z_c)$ *be the discrete and continuous parts of* $\mathbf{X}$*, respectively, with* $c + d = n$*. We say that a function* $f : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$ *is a* mixture of truncated exponentials potential (MTE potential) *if one of the following two conditions holds:*

(1) f *can be written as*

$$f(\mathbf{x}) = f(\mathbf{y}, \mathbf{z}) = a_{0,\mathbf{y}} + \sum_{i=1}^{k} a_{i,\mathbf{y}} \exp\left\{\sum_{j=1}^{c} b_{i,y}^{(j)} z_j\right\}, \tag{4}$$

for all $\mathbf{x} \in \Omega_{\mathbf{X}}$*, where* $a_{i,\mathbf{y}}$*,* $i = 0, \ldots, k$ *and* $b_{i,\mathbf{y}}^{(j)}$*,* $i = 1, \ldots, k$*,* $j = 1, \ldots, c$ *are real numbers.*

(2) There is a partition $\Omega_1, \ldots, \Omega_m$ *of* $\Omega_{\mathbf{X}}$ *for which the domain of the continuous variables,* $\Omega_{\mathbf{Z}}$*, is divided into hypercubes and such that* f *is defined as*

$$f(\mathbf{x}) = f_i(\mathbf{x}) \quad \text{if} \quad \mathbf{x} \in \Omega_i \ ,$$

where each f_i*,* $i = 1, \ldots, m$ *can be written in the form of Equation (4).*

An MTE potential ϕ is said to be an *MTE density* if $\sum_{\Omega_{\mathbf{y}}} \int_{\Omega_{\mathbf{z}}} \phi(\mathbf{y}, \mathbf{z}) d\mathbf{z} = 1$.

Example 3 *The function defined as*

$$\phi(z_1, z_2) = \begin{cases} 2 + e^{3z_1+z_2} + e^{z_1+z_2} & \text{if } \ 0 < z_1 \leq 1, \ 0 < z_2 < 2 \\ 1 + e^{z_1+z_2} & \text{if } \ 0 < z_1 \leq 1, \ 2 \leq z_2 < 3 \\ \frac{1}{4} + e^{2z_1+z_2} & \text{if } \ 1 < z_1 < 2, \ 0 < z_2 < 2 \\ \frac{1}{2} + 5e^{z_1+2z_2} & \text{if } \ 1 < z_1 < 2, \ 2 \leq z_2 < 3 \end{cases}$$

is an MTE potential since all its parts are MTE potentials. However, it is not an MTE density.

MTEs act as a general model, which can approximate any distribution arbitrarily well. As for discretization, the error of the approximation can be controlled by defining a finer partitioning (increasing m in Part 2 of the definition above), but for MTEs it is also possible to keep m fixed, and rather increase the number of exponential terms (k in Equation (4)) to improve the MTE approximation within each part. [2]

To model a hybrid domain we need to represent the distribution of all variables by means of a common structure, in this case MTE potentials. Therefore, also the conditional distributions have to be MTE potentials:

Definition 4 (Conditional MTE density) *Let* $\mathbf{X}_1 = (\mathbf{Y}_1, \mathbf{Z}_1)$ *and* $\mathbf{X}_2 = (\mathbf{Y}_2, \mathbf{Z}_2)$ *be two mixed random vectors. A potential* ϕ *defined over* $\Omega_{\mathbf{X}_1 \cup \mathbf{X}_2}$ *is said to be a* conditional MTE density *if for each* $\mathbf{x}_2 \in \Omega_{\mathbf{X}_2}$*, the restriction of potential* ϕ *to* $\mathbf{x}_2$*,* $\phi^{R(\mathbf{X}_2=\mathbf{x}_2)}$ *is an MTE density for* $\mathbf{X}_1$*.*

Finally, a Bayesian network is said to be an MTE network if the conditional and marginal distributions defined in the network are represented by MTE potentials.

The most important feature of MTE potentials is that they are closed under marginalization, combination and restriction [32]. It follows from Equation (1) that the joint probability distribution of an MTE network is a multivariate MTE density function. Since marginalization, combination and restriction are the only operations needed for inference in Bayesian networks, it follows that Bayesian networks with distributions represented by MTEs offer exact inference, see for example [35] for the adaption of the Shenoy and Shafer algorithm [36] to deal with MTE networks.

Going back to our example, we now proceed by building an MTE network for inference. First, we need to define MTE densities for the marginal distributions of Z_i and the conditional distributions of T_i. Accurate MTE approximations for the Gaussian distribution and the sigmoid function are given below [33].

[2] Empirical studies have concluded that $k = 2$ exponential terms are usually enough to get a very good approximation inside a limited interval [33,34].

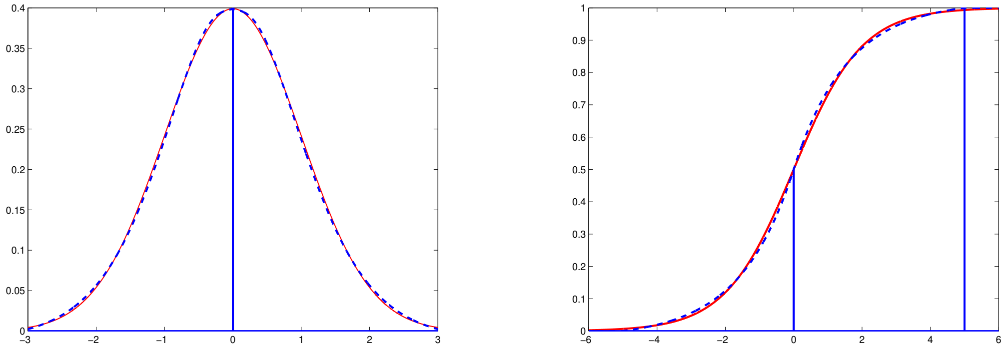


Fig. 6. Results of the MTE approximation in our running example: The approximation to each distribution is dashed, the true underlying density (Normal in the left panel and the logistic in the right) is given with solid line.

We use a '*' to indicate that a probability distribution is approximated using the MTE framework.

$$f^*(z_i) = \begin{cases} -0.017203 + 0.930964e^{1.27z_i} & \text{if } -3 \le z_i < -1 \\ 0.442208 - 0.038452e^{-1.64z_i} & \text{if } -1 \le z_i < 0 \\ 0.442208 - 0.038452e^{1.64z_i} & \text{if } 0 \le z_i < 1 \\ -0.017203 + 0.930964e^{-1.27z_i} & \text{if } 1 \le z_i < 3 \end{cases}$$

$$P^*(T_i = 1|z_1+z_2) = \begin{cases} 0 & \text{if } z_1 + z_2 < -5 \\ -0.021704 + 0.521704e^{0.635(z_1+z_2)} & \text{if } -5 \le z_1 + z_2 < 0 \\ 1.021704 - 0.521704e^{-0.635(z_1+z_2)} & \text{if } 0 \le z_1 + z_2 \le 5 \\ 1 & \text{if } z_1 + z_2 > 5 \end{cases}.$$

The computation of the likelihood $P(\boldsymbol{T} = \mathbf{1})$ is

$$P^*(\boldsymbol{T} = \mathbf{1}) = \int_{\mathbb{R}^2} f^*(z_1) f^*(z_2) \prod_{i=1}^{4} P^*(T_i = 1|\boldsymbol{z}) d\boldsymbol{z} = 0.176819.$$

The joint density for (Z_1, Z_2) given $\boldsymbol{T} = \mathbf{1}$ is

$$f^*(z_1, z_2|\boldsymbol{T}) = \frac{f^*(z_1) f^*(z_2) \prod_{i=1}^{4} P^*(T_i = 1|\boldsymbol{z})}{P^*(\boldsymbol{T} = \mathbf{1})}.$$

Since the combination of MTE potentials is again an MTE potential, the result will be an MTE potential (depicted in Fig. 7).

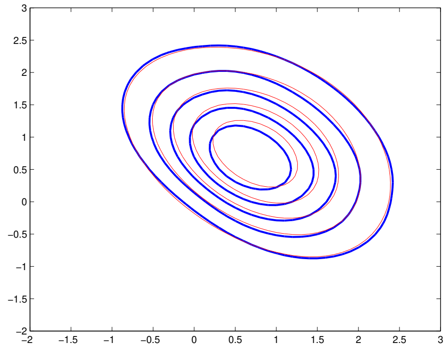


Fig. 7. The MTE approximation of $f(\boldsymbol{z}|\boldsymbol{T}=\mathbf{1})$. Note the very good approximation to the joint posterior distribution when compared to that of discretization (see Fig. 5).

The open-source project ELVIRA [37] implements the MTE approach outlined above. It is a research tool implemented in Java.

5.3 Variational approximations

As stated in Section 4, there is no analytical representation for calculating the likelihood of an observation

$$P(\boldsymbol{T}=\mathbf{1}) = \int_{\mathbb{R}^2} \left\{ \prod_{i=1}^{4} P(T_i = 1, \boldsymbol{z}) \right\} f(\boldsymbol{z}) d\boldsymbol{z}. \tag{5}$$

A variational approach [38–41] to handle this problem consists in defining a lower bound approximation to the logistic function. The approximation considered by the authors above is of a Gaussian shape, which (among other things) entails a closed form marginal likelihood approximation that also defines a lower bound for the face-value likelihood.

To put it more precisely, the logistic function $P(T_i = 1|\boldsymbol{z})$ can be approximated by

$$\tilde{P}(t_i|\boldsymbol{z}, \xi_i) = g(\xi_i) \exp((A_i - \xi_i)/2 + \lambda(\xi_i)(A_i^2 - \xi_i^2)), \tag{6}$$

where

$$A_i = (2t_i - 1)(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{z} + b_i) \quad \text{and} \quad \lambda(\xi_i) = \frac{\exp(-\xi_i) - 1}{4\xi_i(1 + \exp(-\xi_i))}.$$

The function $\tilde{P}(T_i = 1|\boldsymbol{z}, \xi_i)$ is a lower-bound variational approximation to $P(T_i = 1|\boldsymbol{z})$, which means that $\tilde{P}(T_i = 1|\boldsymbol{z}, \xi_i) \leq P(T_i = 1|\boldsymbol{z})$ for all values of the *variational parameter* ξ_i; equality is obtained when $\xi_i = (2t_i - 1)(\boldsymbol{w}_i^{\mathrm{T}} \boldsymbol{z} + b_i)$.

As an example, consider Fig. 8 which shows the logistic function $P(T_i = 1|z)$ as a function of $z := \boldsymbol{w}^{\mathrm{T}}\boldsymbol{z} + b = z_1 + z_2$ together with variational approximations for different values of ξ. Note that at e.g. $z = 1$ the approximation is exact if and only if $\xi = 1$, i.e., the approximation is exact only *point-wise*. The trick is now to find a value for ξ_i that is good "on average" (we shall return to this a bit later).

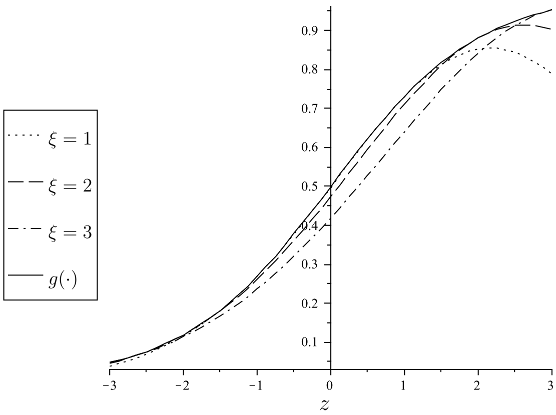


Fig. 8. The solid line shows the logistic function with $\boldsymbol{w} = [1, 1]^{\mathrm{T}}$ and $b = 0$ (as in our example). The three other functions correspond to the variational approximations defined by $\xi = 1$, $\xi = 2$,and $\xi = 3$, respectively; ξ is chosen so to maximize the expected lower bound of the data-complete marginal likelihood, hence it also depends on the prior distribution for the explanatory variables.

From Equation (6) we see that the variational approximation is Gaussian-shaped (quadratic in each z_j in the exponential), hence with a bit of pencil-pushing we can get a lower-bound approximation for the marginal likelihood in Equation (5) (for Equation (5) we have $d = 4$ and $q = 2$):

$$\begin{aligned} P(\boldsymbol{t}) &\geq \int_{\mathbb{R}^q} \left\{ \prod_{i=1}^{d} \tilde{P}(t_i|\boldsymbol{z}, \xi_i) \right\} f(\boldsymbol{z}) d\boldsymbol{z}. \\ &= \exp\left\{ -\frac{1}{2}\boldsymbol{\mu}^{\mathrm{T}}\boldsymbol{\Gamma}^{-1}\boldsymbol{\mu} + \frac{1}{2}(\boldsymbol{\mu}^p)^{\mathrm{T}}(\boldsymbol{\Gamma}^p)^{-1}\boldsymbol{\mu}^p + \frac{1}{2}\log\left(\frac{|\boldsymbol{\Gamma}^p|}{|\boldsymbol{\Gamma}|}\right) \right\} \cdot \\ &\quad \exp\left\{ \sum_{i=1}^{d} \left\{ \log(g(\xi_i)) - \xi_i/2 + \lambda_i(b_i^2 - \xi_i^2) + \frac{1}{2}(2t_i - 1)b_i \right\} \right\}. \end{aligned} \tag{7}$$

where $\boldsymbol{\Gamma}^p$ and $\boldsymbol{\mu}^p$ are the posterior covariance and expectation for $\boldsymbol{Z}$ given $\{\boldsymbol{t}, \boldsymbol{\xi}\}$:

$$\boldsymbol{\Gamma}^p = \left[\boldsymbol{\Gamma}^{-1} - 2\sum_{i=1}^{d} \lambda(\xi_i)\boldsymbol{w}_i\boldsymbol{w}_i^{\mathrm{T}} \right]^{-1} \tag{8}$$

$$\boldsymbol{\mu}^p = \boldsymbol{\Gamma}^p \left\{ \boldsymbol{\Gamma}^{-1}\boldsymbol{\mu} + \sum_{i=1}^{d} \left[t_i - \frac{1}{2} + 2\lambda(\xi_i)b_i \right] \boldsymbol{w}_i \right\}. \tag{9}$$

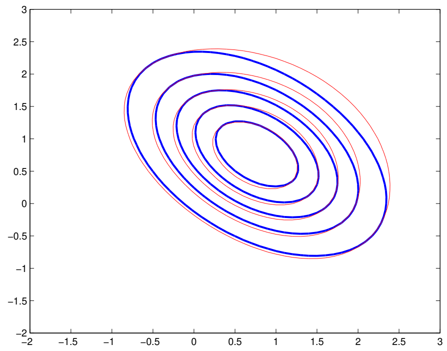


Fig. 9. The joint distribution of $f(z_1, z_2|\boldsymbol{T} = \mathbf{1})$ obtained by variational methods.

The approximations above all depend on ξ_i, but since $\boldsymbol{Z}$ is not observed we cannot directly calculate the value for ξ_i that maximizes the lower bound $\tilde{P}(\boldsymbol{t}|\boldsymbol{\xi})$; recall that the approximation is exact only if $\xi_i = (2t_i - 1)(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{z} + b_i)$. Instead we can maximize the expected lower bound $\mathbf{E}[\tilde{P}(\boldsymbol{t}, \boldsymbol{Z}|\boldsymbol{\xi})]$ by following an EM like approach [42]. For this, [43] showed that the expected lower bound is maximized by $\xi_i^2 = \mathbf{E}[(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{Z}+b_i)^2|\boldsymbol{T} = \boldsymbol{t}]$, but since this expectation depends on $\boldsymbol{\Gamma}^p$ and $\boldsymbol{\mu}^p$ an iterative scheme is required.

Algorithm 1 Variational inference

1: Start with initial guesses for $\boldsymbol{\Gamma}^p$ and $\boldsymbol{\mu}^p$.[3]
2: **repeat**
3: Update values for $\boldsymbol{\xi}$ by setting

$$\xi_i \leftarrow \sqrt{E\left[(\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{Z} + b_i)^2|\boldsymbol{T}\right]} = \sqrt{(\boldsymbol{\mu}^p)^{\mathrm{T}}\boldsymbol{\mu}^p + \boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{\Gamma}^p\boldsymbol{w}_i + 2b_i\boldsymbol{w}_i^{\mathrm{T}}\boldsymbol{\mu} + b_i^2}.$$

4: Calculate $\boldsymbol{\Gamma}^p$ and $\boldsymbol{\mu}^p$ based on the current $\boldsymbol{\xi}$ (Equations 8 and 9).
5: **until** Termination criterion

This iterative scheme is guaranteed to maximize the variational approximation $\tilde{P}(\boldsymbol{t}|\boldsymbol{\xi})$ and since the approximation defines a lower bound for $P(\boldsymbol{t})$ it is also guaranteed to maximize the actual likelihood of the observation $\boldsymbol{t}$.

Going back to our running example, we find the variational lower-bound of the likelihood to be 0.140329, a rather poor estimate. On the other hand, the *shape* of the joint distribution for Z_1 and Z_2 given $\boldsymbol{T} = \mathbf{1}$ is well approximated by the variational approximation (see Fig. 9).

From the above considerations we see that for the fixed structure given by our running example, there exists a variational approximation allowing us to answer the probabilistic queries of interest. However, applying the variational framework in domains with other probability distributions may require new variational approximations and coming up with such approximations may be a bit of an art-form. In mathematical terms, the general variational Bayes approach attempts to minimize the Kullback-Leibler divergence between the true posterior and a simpler, approximating distribution. Although it is not the case in our example, we usually see the approximating distribution made simpler than the true posterior by assuming that the parameters in the approximating distribution are independent (see, e.g., [44] for an overview).

VIBES (Variational Inference for Bayesian Networks) provides an inference engine for performing variational inference in Bayesian networks [45].

5.4 Markov Chain Monte Carlo Methods

Instead of using functional approximations to achieve tractable and analytical representations supporting probability updating, one may also estimate the required probabilities using sampling. As an example, consider calculating

$$\begin{aligned} P(T_1 = 1, \ldots, T_4 = 1) &= \int_{\mathbb{R}^2} \prod_{i=1}^{4} P(T_i = 1 | \boldsymbol{z}) f(\boldsymbol{z}) d\boldsymbol{z} \\ &= \mathbf{E}_{\boldsymbol{Z}}(P(\boldsymbol{T} = \mathbf{1} | \boldsymbol{Z})). \end{aligned} \tag{10}$$

This expectation can be estimated by drawing samples $\{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$ from $f(\boldsymbol{z})$ and then approximating

$$\mathbf{E}_{\boldsymbol{Z}}(P(\boldsymbol{T} = \mathbf{1} | \boldsymbol{Z})) \approx \frac{1}{N} \sum_{i=1}^{N} P(\boldsymbol{T} = \mathbf{1} | \boldsymbol{z}_i).$$

The law of large numbers guarantees that with a sufficiently large number of independent and identically distributed samples, we can obtain any desired degree of precision in the estimate. Fig. 10 shows how the precision improves as more samples are used; the shaded area gives the 5% and 95% quantiles (1000 repetitions) for the likelihood.

For the expectation above, we have that Z_1 and Z_2 are marginally independent, thus we only need to sample from a univariate normal distribution for which standard algorithms exist. Unfortunately, for distributions with no analytical representation (such as $f(\boldsymbol{z} | \boldsymbol{T} = \mathbf{1})$) it can be quite difficult to draw independent samples. Instead we may generate dependent samples and exploit that the independence assumption can be relaxed as long as we obtain samples throughout the support of the target distribution and in the correct proportions.

Markov chain Monte Carlo methods provide a general technique for drawing a sequence of dependent samples $\{\boldsymbol{z}_0, \boldsymbol{z}_1, \boldsymbol{z}_2, \ldots, \boldsymbol{z}_t\}$ from a target distribution

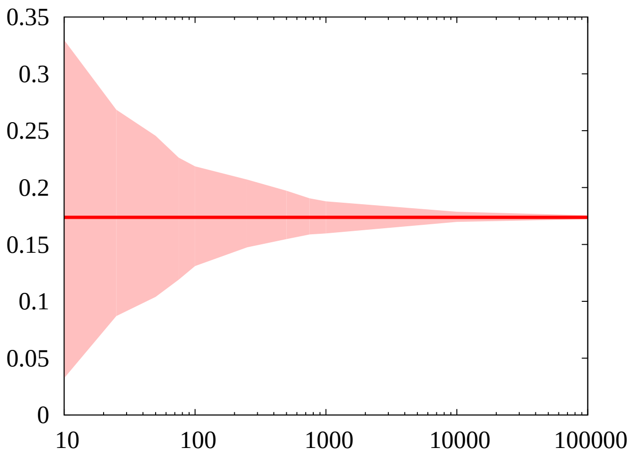


Fig. 10. Uncertainty of the estimate of $P(\boldsymbol{T} = \mathbf{1})$ as a function of the number of samples is displayed as the 5% and 95% quantiles (1000 repetitions).

that does not necessarily have an analytical representation. In this setting, a *transition function* $g(\boldsymbol{z}_{t+1}|\boldsymbol{z}_t)$ is used to sample the next state $\boldsymbol{z}_{t+1}$ given $\boldsymbol{z}_t$ and independently of $\boldsymbol{z}_i$, for $1 \leq i \leq t-1$; thus, the $\boldsymbol{z}_i$s form a Markov chain. Moreover, subject to certain regularity conditions [46], the distribution $f_t(\boldsymbol{z}|\boldsymbol{z}_0)$ from which the samples are drawn will eventually converge to a stationary distribution independent of the starting state $\boldsymbol{z}_0$.

One of the simpler instantiations of this general framework is the Metropolis-Hastings algorithm. In the Metropolis-Hastings algorithm, a candidate next state $\boldsymbol{c}$ is sampled from a so-called *proposal function* $q(\cdot|\cdot)$ (that may depend on the current state $\boldsymbol{z}_t$) and the proposed state is then acceptated with probability

$$\begin{aligned}
\mathrm{acc}(\boldsymbol{c}, \boldsymbol{z}_t) &= \min\left(1, \frac{P(\boldsymbol{c}|\boldsymbol{T} = \mathbf{1})q(\boldsymbol{z}_t|\boldsymbol{c})}{P(\boldsymbol{z}_t|\boldsymbol{T} = \mathbf{1})q(\boldsymbol{c}|\boldsymbol{z}_t)}\right) \\
&= \min\left(1, \frac{\prod_{i=1}^{4} P(T_i = 1|\boldsymbol{c})f(c_1)f(c_2)q(\boldsymbol{z}_t|\boldsymbol{c})}{\prod_{i=1}^{4} P(T_i = 1|\boldsymbol{z})f(z_1)f(z_2)q(\boldsymbol{c}|\boldsymbol{z}_t)}\right).
\end{aligned}$$

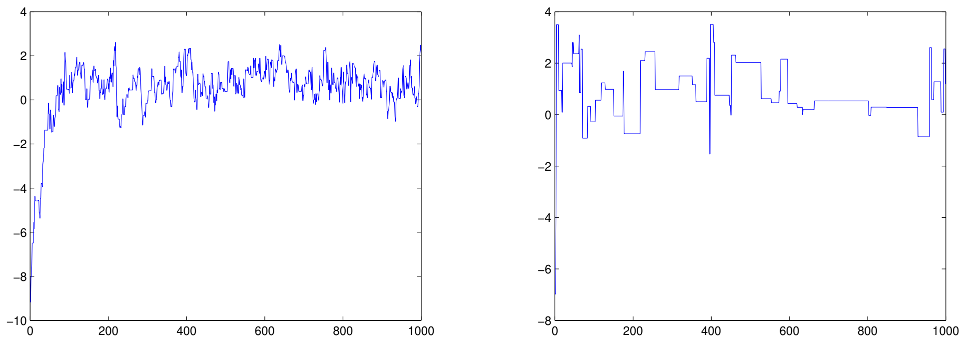
If the state is accepted, the chain moves to the proposed state ($\boldsymbol{z}_{t+1} \leftarrow \boldsymbol{c}$), otherwise it stays at its current state ($\boldsymbol{z}_{t+1} \leftarrow \boldsymbol{z}_t$). Algorithm 2 summarizes the description above.

It can be shown (see e.g. [46]) that under certain conditions, the stationary distribution is in fact the target distribution, irrespectively of the proposal function being used. Although convergence to the target distribution does not depend on the proposal function, it does, however, have an impact on the convergence speed and the mixing rate (the speed in which samples are drawn from the area with positive support under the target distribution). For example, Figure 11 shows the sample sequences obtained for $P(Z_1|\boldsymbol{T} = \mathbf{1})$ using $N(\boldsymbol{z}_t, \frac{1}{2}\boldsymbol{I})$ and $N(\boldsymbol{z}_t, 5\boldsymbol{I})$ as proposal functions; Z_1 and Z_2 are sampled

Algorithm 2 The Metropolis-Hastings algorithm

$t \leftarrow 0$
Initialize $\boldsymbol{z}_0$
repeat
    Sample a candidate state $\boldsymbol{c}$ form $q(\cdot|\boldsymbol{z}_t)$
    Sample a value a from a uniform distribution over the unit interval
    if $a \leq \text{acc}(\boldsymbol{z}_t, \boldsymbol{c})$ **then**
        $\boldsymbol{z}_{t+1} \leftarrow \boldsymbol{c}$
    else
        $\boldsymbol{z}_{t+1} \leftarrow \boldsymbol{z}_t$
    end if
    $t \leftarrow t + 1$
until Termination criterion

independently. In particular, for $q(\cdot|\boldsymbol{z}_t) = N(\boldsymbol{z}_t, 5\boldsymbol{I})$ we see that the chain mixes slowly (there are several regions where the chain does not move) and we will therefore require a longer running time to get a representative sample.



(a) (b)

Fig. 11. The figures show the samples for Z_1 using the proposals distributions $N(\boldsymbol{z}_t, \frac{1}{2}\boldsymbol{I})$ and $N(\boldsymbol{z}_t, 5\boldsymbol{I})$, respectively.

When using the sampled values to e.g. analyze the conditional distribution $P(\boldsymbol{z}|\boldsymbol{T} = \boldsymbol{1})$, the initial samples (called the *burn-in*) obtained prior to convergence are usually discarded. The question is now how to detect when the distribution $f_t(\boldsymbol{z}|\boldsymbol{z}_0)$ of the chain is sufficiently close to the target distribution and when a sufficient number of samples have been drawn. As examples, [47] and [48] discuss methods for analyzing the convergence properties by comparing several chains run in parallel, and [49] consider methods for analyzing a single chain.

Given a sample set, we can use the samples to analyze the target distribution by e.g. estimating the expectation and covariance of $\boldsymbol{Z}$. We can also estimate the distribution of $\boldsymbol{Z}$, say, by using kernel density estimation. The kernel density estimate for $\boldsymbol{Z}$ given $\boldsymbol{T} = \boldsymbol{1}$ is shown in Fig. 12. The calculation of the likelihood is, as we have seen (Fig. 10), a stochastic quantity. Using 1000 samples we obtained an estimated likelihood of 0.17466. It is interesting to see that although the MTEs are better at approximating the joint distribution $f(\boldsymbol{z}|\boldsymbol{T}=\mathbf{1})$ than MCMC with 1000 samples (compare Fig. 7 to the left panel of Fig. 12), the likelihood estimate of the MCMC approach outperforms that of the MTE. This is due to the nature of sampling: The law of large numbers ensures rapid convergence of sample-averages (like the likelihood, see Equation (10)), whereas low-probability events (like the probability $P(Z_1 > 1.5, Z_2 > 1.5|\boldsymbol{T}=\mathbf{1})$) are not as well approximated by moderately sized samples.

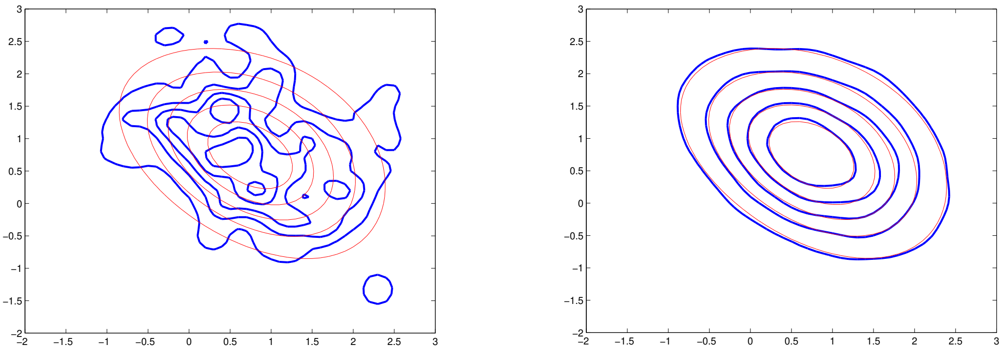


Fig. 12. The joint distribution of $f(z_1, z_2|\boldsymbol{T}=\mathbf{1})$ obtained by MCMC sampling. Left pane shows the results after 1000 samples, the right pane gives the results after 250.000 samples. In both cases, the first 100 observations were discarded as burn-in. The density estimates where smoothed using a Gaussian kernel.

BUGS [50] is a general purpose modelling language, which takes as its input a BN model and returns samples that can be used for estimating any (conditional) probability distribution. BUGS is accompanied by CODA [51], which is a tool for analyzing whether or not sufficient mixing has taken place.

5.5 Other approaches

There are other approaches for performing inference in hybrid Bayesian networks that we have not described here. One of the earliest ideas was based on the *conditional Gaussian* (CG) model [52], which assumes that the conditional distribution for the continuous variables given the discrete ones is multivariate Gaussian. A particular case is the *conditional linear Gaussian* (CLG) model, where the mean of the conditional distribution of each continuous variable is a linear function of its continuous parent variables in the network. There exist efficient algorithms for carrying out exact inference in Bayesian networks following a CLG model [53]. However, it is required that discrete variables only have discrete parents, and this imposes an important limitation to the problems that can be modeled following the CG-approach. For instance, our running example cannot be directly represented by CG or CLG models. A solution to this problem is proposed in [54], which consists in transforming a network containing discrete variables with continuous parents into another network in which the resulting distribution is a *mixture of Gaussians*.

Other approaches that are currently receiving some attention in the research community include *Expectation Propagation* [55] and techniques based on the *Laplace approximation* [56].

6 Conclusions

In this paper we have explored four approaches to inference in hybrid Bayesian networks: discretization, mixtures of truncated exponentials (MTEs), variational methods, and Markov chain Monte Carlo (MCMC). Each of them have their pros and cons, which we will briefly summarize here. We note that this paper is about *inference*, hence the specification of the models (either manually or by learning from data) is outside the scope of this discussion. Furthermore, we have considered the inference problem in the context of reliability analysis. This means that we are interested in obtaining good approximations for low probability events, and will therefore give the *tails* of the approximations some attention in the following.

The simplest approach to inference in hybrid domains is to use discretization. Discretization entails only a simple transformation of the continuous variables, it is implemented in almost all commercial BN tools, and the user only has to decide upon one parameter, namely m, the number of intervals the continuous variables are discretized into. Choosing a "good" value for m can be a bit of a problem, though, since a too high value leads to complexity problems and a too low value leads to poor approximations. We note that practitioners in reliability who use discretization without investigating this effect further are in danger of under-estimating the probability of unwanted events considerably.

MTEs are generalizations of standard discretization, with the aim of avoiding the complexity problems discretization are hampered by. MTEs benefit from the BNs' efficient inference engine. Furthermore, MTEs define a rather general framework, which can approximate any distribution accurately. In particular, MTEs are better at approximating the tail of the distribution of our running example than discretization (compare Fig. 5 and Fig. 6). MTEs are currently receiving a lot of attention from the research community; both refining the inference and exploring new applications are hot research topics. On the downside, the MTE framework is still in its infancy, and in particular methods for learning MTEs from data must be further explored.

The variational approximations provide satisfactory answers to the kind of queries associated with inference in hybrid Bayesian networks. However, the variational approximations are still rather *ad hoc*, and the formulae have to be rewritten depending on the underlying distribution used. It is also difficult to have a well-founded understanding of the error the variational approximation makes, and as we saw in the example, the error can be substantial.

MCMC is a very general inference technique, and it can take advantage of a BN's structure to speed up the simulation process. Together with standard discretization, MCMC is currently the most popular technique for inference in hybrid BNs. This is partly due to a strong mathematical foundation and well-known statistical properties of the generated estimates. From a practitioners point of view, one should however be vigilant when using MCMC to estimate the probability of rare events. If the probability of a gas leak, say, is $p = 10^{-4}$ one would on average need to generate $1/p = 10^4$ samples after burn-in to obtain a single sample of the event. It is also particularly important to consider the auto-correlation in the samples before conclusions regarding rare events are drawn. It is our experience that practitioners are not always aware of these facts, and sometimes abuse the methodology by underestimating the demands to obtain representative samples.

In our experience, the discretization method (with moderate number of regions) is the fastest technique, outperforming the MTE method (with the same number of regions) by a factor of about 2. On the other hand, MTEs are about four times faster than the variational approximation. This is not surprising, as the variational approximation requires a number of iterations to converge (refer to Algorithm 1). MCMC is comparably much slower than MTEs (a factor of about 10^3 to obtain results of comparable quality).

Among the four explored approaches, the MTE framework appears to be the one best suited for reliability applications: It balances the need for good approximations in the tail of the distributions with not-too-high computational complexity. MTEs are flexible from the modeling point of view, and there exist efficient methods for inference building on the classical BN inference scheme.

Acknowledgements

This work has been partially supported by the Spanish Ministry of Science and Innovation through project TIN2007-67418-C03-02, and by FEDER funds.

This paper is partly based on a talk given at the*The Fifth International Conference on Mathematical Models in Reliability (MMR'07)*. We would like to thank the participants there for interesting discussions. We also thank the anonymous reviewers for their constructive comments that helped improve the paper.

References

[1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, CA., 1988.

[2] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, D. J. Spiegelhalter, Probabilistic Networks and Expert Systems, Statistics for Engineering and Information Sciences, Springer-Verlag, New York, NY, 1999.

[3] F. V. Jensen, T. D. Nielsen, Bayesian Networks and Decision Graphs, Springer-Verlag, Berlin, Germany, 2007.

[4] H. Langseth, L. Portinale, Bayesian networks in reliability, Reliability Engineering and System Safety 92 (1) (2007) 92–108.

[5] J. G. Torres-Toledano, L. E. Sucar, Bayesian networks for reliability analysis of complex systems, in: Proceedings of the 6th Ibero-American Conference on AI (IBERAMIA 98), No. 1484 in Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, Germany, 1998, pp. 195–206.

[6] J. Solano-Soto, L. E. Sucar, A methodology for reliable system design, in: Proceedings of the 4th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Vol. 2070 of Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, Germany, 2001, pp. 734–745.

[7] L. Portinale, A. Bobbio, Bayesian networks for dependability analysis: an application to digital control reliability, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA., 1999, pp. 551–558.

[8] A. Bobbio, L. Portinale, M. Minichino, E. Ciancamerla, Improving the analysis of dependable systems by mapping fault trees into Bayesian networks, Reliability Engineering and System Safety 71 (3) (2001) 249–260.

[9] A. P. Tchangani, Reliability analysis using Bayesian networks, Studies in Informatics and Control 10 (3) (2001) 181–188.

[10] S. Montani, L. Portinale, A. Bobbio, Dynamic Bayesian networks for modeling advanced fault tree features in dependability analysis, in: Proceedings of the Sixteenth European Conference on Safety and Reliability, A. A. Balkema, Leiden, The Netherlands, 2005, pp. 1415–1422.

[11] J. Pearl, Causality – Models, Reasoning, and Inference, Cambridge University Press, Cambridge, UK, 2000.

[12] R. E. Barlow, Using influence diagrams, in: C. A. Clarotti, D. V. Lindley (Eds.), Accelerated life testing and experts' opinions in reliability, No. 102 in Enrico Fermi International School of Physics, Elsevier Science Publishers B. V., North-Holland, 1988, pp. 145–157.

[13] R. G. Almond, An extended example for testing GRAPHICAL-BELIEF, Technical Report 6, Statistical Sciences Inc., Seattle, WA (1992).

[14] H. F. Martz, R. A. Waller, Bayesian reliability analysis of complex series/parallel systems of binomial subsystems and components, Technometrics 32 (4) (1990) 407–416.

[15] J. Sigurdsson, L. Walls, J. Quigley, Bayesian belief nets for managing expert judgment and modeling reliability, Quality and Reliability Engineering International 17 (2001) 181–190.

[16] R. Dechter, Bucket elimination: A unifying framework for probabilistic inference, in: E. Horvitz, F. V. Jennsen (Eds.), Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA., 1996, pp. 211–219.

[17] Z. Li, B. D'Ambrosio, Efficient inference in Bayes networks as a combinatorial optimization problem, International Journal of Approximate Reasoning 11 (1994) 55–81.

[18] N. Zhang, D. Poole, Exploiting causal independence in Bayesian network inference, International Journal of Approximate Reasoning 5 (1996) 301–328.

[19] P. P. Shenoy, G. R. Shafer, Axioms for probability and belief function propagation, in: R. Shachter, T. Levitt, J. Lemmer, L. Kanal (Eds.), Proceedings of the Sixth Workshop on Uncertainty in Artificial Intelligence, North Holland, Amsterdam, 1990, pp. 169–198.

[20] M. Shaked, A concept of positive dependence for exchangeable random variables, The Annals of Statistics 5 (1977) 505–515.

[21] D. V. Lindley, N. D. Singpurwalla, Multivariate distributions for the lifelengths of components of a system sharing a common environment, Journal of Applied Probability 23 (1986) 418–431.

[22] P. A. Currit, N. D. Singpurwalla, On the reliability function for a system of components sharing a common environment, Journal of Applied Probability 26 (1988) 763–771.

[23] M. Neil, M. Tailor, D. Marquez, Inference in Bayesian networks using dynamic discretisation, Statistics and Computing 17 (3) (2007) 219–233.

[24] A. D. Swain, H. E. Guttman, Handbook of human reliability analysis with emphasis on nuclear power plant applications, NUREG/CR 1278, Nuclear Regulatory Commission, Washington, D.C. (1983).

[25] D. J. Bartholomew, Latent Variable Models and Factor Analysis, Charles Griffin & Co., London, UK, 1987.

[26] J. Dougherty, R. Kohavi, M. Sahami, Supervised and unsupervised discretization of continuous features, in: Proceedings of the Twelfth International Conference on Machine Learning, 1995, pp. 194–202.

[27] A. Christofides, B. Tanyi, S. Christofides, D. Whobrey, N. Christofides, The optimal discretization of probability density functions, Computuational Statistics and Data Analysis 31 (4) (1999) 475–486.

[28] N. Friedman, M. Goldszmidt, Discretizing continuous attributes while learning Bayesian networks, in: Proceedings of the Thirteenth International Conference on Machine Learning, Morgan Kaufmann Publishers, San Francisco, CA, 1996, pp. 157–165.

[29] A. V. Kozlov, D. Koller, Nonuniform dynamic discretization in hybrid networks, in: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, 1997, pp. 314–325.

[30] U. M. Fayyad, K. B. Irani, Multi-interval discretization of continuous-valued attributes for classification learning., in: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, Morgan Kaufmann Publishers, San Mateo, CA., 1993, pp. 1022–1027.

[31] M. J. Pazzani., An iterative approach for the discretization of numeric attributes in Bayesian classifiers, in: Proceedings of the First International Conference on Knowledge Discovery and Data Mining, 1995, pp. 228–233.

[32] S. Moral, R. Rumí, A. Salmerón, Mixtures of truncated exponentials in hybrid Bayesian networks, in: S. Benferhat, P. Besnard (Eds.), ECSQARU '01: Proceedings of the 6th European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Vol. 2143, Springer-Verlag, London, UK, 2001, pp. 156–167.

[33] B. Cobb, P. P. Shenoy, R. Rumí, Approximating probability density functions with mixtures of truncated exponentials, Statistics and Computing 16 (2006) 193–308.

[34] R. Rumí, A. Salmerón, S. Moral., Estimating mixtures of truncated exponentials in hybrid Bayesian networks, TEST: An Official Journal of the Spanish Society of Statistics and Operations Research 15 (2) (2006) 397–421.

[35] R. Rumí, A. Salmerón, Approximate probability propagation with mixtures of truncated exponentials, International Journal of Approximate Reasoning 45 (2) (2007) 191–210.

[36] G. R. Shafer, P. P. Shenoy, Probability propagation, Annals of Mathematics and Artificial Intelligence 2 (1990) 327–352.

[37] Elvira Consortium, Elvira: An environment for creating and using probabilistic graphical models, in: J. Gámez, A. Salmerón (Eds.), Proceedings of the First European Workshop on Probabilistic Graphical Models, 2002, pp. 222–230.

[38] T. S. Jaakkola, Variational methods for inference and estimation in graphical models, Ph.D. thesis, Dept. of Brain and Cognitive Sciences, Massachusetts Institute of Technology (1997).

[39] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, L. K. Saul, An introduction to variational methods for graphical models, Machine Learning 37 (1999) 183–233.

[40] T. Jaakkola, M. I. Jordan, Bayesian parameter estimation via variational methods, Statistics and Computing 10 (1999) 25–37.

[41] M. E. Tipping, Probabilistic visualisation of high-dimensional binary data, in: M. S. Kearns, S. A. Solla, D. A. Cohn (Eds.), Advances in Neural Information Processing Systems 11, The MIT Press, 1999, pp. 592– 598.

[42] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B 39 (1977) 1–38.

[43] K. P. Murphy, A variational approximation for Bayesian networks with discrete and continuous latent variables, in: K. B. Laskey, H. Prade (Eds.), Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, San Francisco, CA., 1999, pp. 467–475.

[44] V. Smídl, A. Quinn, The Variational Bayes Method in Signal Processing, Springer-Verlag, New York, 2006.

[45] C. M. Bishop, D. Spiegelhalter, J. Winn, Vibes: A variational inference engine for bayesian networks, in: S. T. S. Becker, K. Obermayer (Eds.), Advances in Neural Information Processing Systems 15, MIT Press, Cambridge, MA, 2003, pp. 777–784.

[46] W. Gilks, S. Richardson, D. J. Spiegelhalter, Markov Chain Monte Carlo in practice, Interdisciplinary Statistics, Chapmann & Hall, London, UK, 1996.

[47] A. Gelman, B. B. Rubin, Inference from iterative simulation using multiple sequences, Statistical Science 7 (1992) 457–472.

[48] S. Brooks, A. Gelman, General methods for monitoring convergence of iterative simulations, Journal of Computational and Graphical Statistics 7 (1998) 434–456.

[49] A. E. Raftery, S. M. Lewis, Markov Chain Monte Carlo in Practice, Chapmann & Hall, London, 1996, Ch. Implementing MCMC, pp. 115–130.

[50] W. Gilks, A. Thomas, D. J. Spiegelhalter, A language and program for complex Bayesian modelling, The Statistician 43 (1994) 169–178.

[51] N. Best, M. K. Cowles, K. Vines, CODA manual version 0.30, MRC Biostatistics Unit, Cambridge, UK (1995).

[52] S. L. Lauritzen, Propagation of probabilities, means and variances in mixed graphical association models, Journal of the American Statistical Association 87 (420) (1992) 1098–1108.

[53] A. L. Madsen, Belief update in CLG Bayesian networks with lazy propagation, International Journal of Approximate Reasoning 49 (2008) 503–521.

[54] P. P. Shenoy, Inference in hybrid Bayesian networks using mixtures of Gaussians, in: Proceedings of the Twenty-second Conference on Uncertainty in Artificial Intelligence, 2006, pp. 428–436.

[55] T. P. Minka, A family of algorithms for approximate Bayesian inference, Ph.D. thesis, MIT Media Lab (2001).

[56] D. M. Chickering, D. E. Heckerman, Efficient approximations for the marginal likelihood of Bayesian networks with hidden variables, in: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA., 1996, pp. 158–168.