Answering queries in hybrid Bayesian networks using importance sampling

Antonio Fernández*, Rafael Rumí, Antonio Salmerón

Dept. Statistics and Applied Mathematics, University of Almería, La Cañada de San Urbano s/n, 04120 Almería, Spain

Abstract

In this paper we propose an algorithm for answering queries in hybrid Bayesian networks where the underlying probability distribution is of class MTE (mixture of truncated exponentials). The algorithm is based on importance sampling simulation. We show how, like existing importance sampling algorithms for discrete networks, it is able to provide answers to multiple queries simultaneously using a single sample. The behaviour of the new algorithm is experimentally tested and compared with previous methods existing in the literature.

Keywords: Bayesian networks, probabilistic reasoning, importance sampling, Mixtures of Truncated Exponentials

2000 MSC: 6505, 68T37

1. Introduction

Bayesian networks [17, 36] have become a popular tool for representing uncertainty in decision support systems. A review of recent literature shows

*Corresponding author. Tel.: +34 950214650; fax: +34 950015167.

Email addresses: afalvarez@ual.es (Antonio Fernández), rrumi@ual.es (Rafael Rumí), antonio.salmeron@ual.es (Antonio Salmerón)

the variety of applications in which they have been successfully used [1, 10, 24, 35, 45]. One of the main reasons for using them as the inference engine in a decision support system is that efficient reasoning algorithms can be designed, taking advantage of their structure [2, 3, 16, 44, 43, 30, 29].

Most of the methodological development around Bayesian networks has concentrated on the case in which all the variables involved are qualitative or discrete. However, decision support systems usually have to operate in domains described in terms of both discrete and continuous variables simultaneously. In such scenarios, there is always the possibility of discretising the continuous variables [20, 34], in order to be able to use methods designed for discrete variables. But such a solution in general conveys a loss of information.

Continuous and discrete variables can be handled simultaneously, with no need to discretise, in the so-called *hybrid Bayesian networks*. The first advances in this field came along with the definition of the Conditional Gaussian (CG) model [25, 26, 28]. The limitations of this approach are the assumption of normality over the continuous variables, and also the fact that dependencies of discrete variables conditional on continuous ones, are not allowed. This structural restriction is overcome in the *augmented Conditional Linear Gaussian (CLG) networks*, where discrete nodes are allowed to have continuous parents, by representing their conditional distributions as *softmax* functions [27]. However this model also relies on the normality assumption. Furthermore, exact inference is not possible in augmented CLG networks, and the solution proposed in [27] is based on a Gaussian approximation of the product of the Gaussian and softmax functions, which provides exact marginals for the discrete variables and also is able to obtain exact values only for the first and second order moments of the distribution

of the continuous variables.

A more general proposal is based on the use of mixtures of truncated exponentials (MTEs), which do not impose any restriction and also do not rely on the normality assumption [31]. This model has been successfully applied to decision problems [6]. An important feature of MTEs is that they are compatible with efficient exact inference algorithms like, for instance, the Shenoy-Shafer architecture [44] and the variable elimination scheme [49]. As MTEs are able to approximate a wide variety of probability distributions [7], they can be used as a general framework for carrying out inference in hybrid Bayesian networks, just by approximating each conditional distribution in the network by an MTE and then using an exact inference algorithm. This approach has been analysed in [22], by solving a network involving Logistic and Gaussian distributions using MTEs, variational approximations [18], discretisation [33] and Markov Chain Monte Carlo [12].

A recent approach, similar in essence to MTEs, is based on representing the distribution in a hybrid Bayesian network as a *Mixture of Polynomials (MOPs)* [42]. Both MTEs and MOPs have been generalised in a global framework for representing hybrid Bayesian networks, called *Mixtures of Truncated Basis Functions (MoTBFs)* [23]. However, even though MOPs have some advantages over MTEs, specially the ability of dealing with a wider class of deterministic relationships, so far they lack of an algorithm for learning the models from data, while this issue has been solved for MTEs [38]. Hence, MTEs can be used as an exact model and not only as an approximation of other distributions. In that sense, MTEs behave as a nonparametric model, where no assumption is made about the underlying distribution.

Even though Bayesian networks allow efficient inference algorithms to

operate over them, it is known that exact probabilistic inference is an NP-hard problem [8]. Furthermore, approximate probabilistic inference is also an NP-hard problem if a given precision is required [9]. For that reason, approximate algorithms that tradeoff complexity for accuracy have been developed for discrete Bayesian networks. An important class of such approximate algorithms are based on the importance sampling technique, that provides a flexible approach to construct anytime reasoning algorithms [4, 13, 32, 46, 47, 48].

Inference in hybrid Bayesian networks with MTEs does not escape from the above mentioned complexity. If the model is learnt from a database using the algorithm in [38], it can be too complex if the number of variables is high. But even using the approximations in [7], inference may become unfeasible if the model is complex enough.

With this motivation, in this paper we propose an approximate algorithm for computing fast and accurate answers to precise queries in hybrid Bayesian networks with MTEs. The algorithm is based on importance sampling, and therefore it is an *anytime* algorithm [37] in the sense that the accuracy of its results is proportional to the time it is allowed to use for computing the answer. We show how our proposal outperforms the previous state-of-the-art method for approximate inference with MTEs, introduced in [40].

The rest of the paper is organised as follows. We establish the notation and define some preliminary concepts in Sec. 2. The problem addressed here is formally posted in Sec. 3. The core of the methodological contributions is in Sec. 4, and the details of the algorithm can be found in Sec. 5. The experimental analysis carried out to test the performance of the algorithm is reported in Sec. 6. The concluding remarks are given in Sec. 7.

2. Notation and preliminaries

Formally, a *Bayesian network* is a directed acyclic graph where each node represents a random variable, and the topology of the graph encodes the independence relations among the variables, according to the d-separation criterion [36]. Given the independences attached to the graph, the joint distribution is determined giving a probability distribution for each node conditioned on its parents, so that for a Bayesian network with variables $X_1, \ldots, X_n$, the joint distribution factorises as

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

where $pa(x_i)$ denotes the parents of variable X_i in the network.

We will use uppercase letters to denote random variables, and boldfaced uppercase letters to denote random vectors, e.g. $\mathbf{X} = \{X_1, \ldots, X_n\}$, and its domain will be written as $\Omega_{\mathbf{X}}$. By lowercase letters x (or $\mathbf{x}$) we denote some element of Ω_X (or $\Omega_{\mathbf{X}}$).

We are interested in *hybrid* Bayesian networks, which are defined for a set of variables $\mathbf{X}$ that contains discrete and continuous variables. Throughout this paper we will assume that $\mathbf{X} = \mathbf{Y} \cup \mathbf{Z}$, being $\mathbf{Y}$ and $\mathbf{Z}$ sets containing only discrete and only continuous variables respectively. We will follow the approach based on mixtures of truncated exponentials [31], in which all the conditional distributions in Eq. (1) are represented as MTE potentials, which are formally defined as follows.

Definition 1. (MTE potential) *Let* $\mathbf{X}$ *be a mixed* n*-dimensional random vector. Let* $\mathbf{Y} = (Y_1, \ldots, Y_d)^{\mathsf{T}}$ *and* $\mathbf{Z} = (Z_1, \ldots, Z_c)^{\mathsf{T}}$ *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 (MTE) potential *if*

for each fixed value $\mathbf{y} \in \Omega_{\mathbf{Y}}$ *of the discrete variables* $\mathbf{Y}$*, the potential over the continuous variables* $\mathbf{Z}$ *is defined as:*

$$f(\mathbf{z}) = a_0 + \sum_{i=1}^{m} a_i \exp\left\{\mathbf{b}_i^{\mathsf{T}} \mathbf{z}\right\}, \tag{2}$$

for all $\mathbf{z} \in \Omega_{\mathbf{Z}}$*, where* $a_i \in \mathbb{R}$ *and* $\mathbf{b}_i \in \mathbb{R}^c$*,* $i = 1, \ldots, m$*. We also say that* f *is an MTE potential if there is a partition* $D_1, \ldots, D_k$ *of* $\Omega_{\mathbf{Z}}$ *into hypercubes and in each one of them,* f *is defined as in Eq.* (2)*. An MTE potential is an* MTE density *if it integrates to 1.*

A *conditional MTE density* can be specified by dividing the domain of the conditioning variables and specifying an MTE density for the conditioned variable for each configuration of splits of the conditioning variables. The next is an example of a conditional MTE density.

$$f(y|x) = \begin{cases} 1.26 - 1.15e^{0.006y} & \text{if } 0.4 \leq x < 5,\ 0 \leq y < 13\ , \\ 1.18 - 1.16e^{0.0002y} & \text{if } 0.4 \leq x < 5,\ 13 \leq y < 43\ , \\ 0.07 - 0.03e^{-0.4y} + 0.0001e^{0.0004y} & \text{if } 5 \leq x < 19,\ 0 \leq y < 5\ , \\ -0.99 + 1.03e^{0.001y} & \text{if } 5 \leq x < 19,\ 5 \leq y < 43\ . \end{cases}$$

Since MTEs are defined into hypercubes, they admit a tree-structured representation in a natural way. Each entire branch in the tree determines one hypercube where the potential is defined, and the function stored in the leaf of a branch is the definition of the potential on it. An example of a tree-structured representation of an MTE potential is shown in Fig. 1.

We use the term *mixed tree* [31] to refer to a tree-structure representation of an MTE potential. A tree $\mathcal{T}$ is a *mixed tree* if: (i) every internal node represents a random variable, (ii) every arc outgoing from a continuous variable Z is labeled with an interval of values of Z, so that the domain of Z

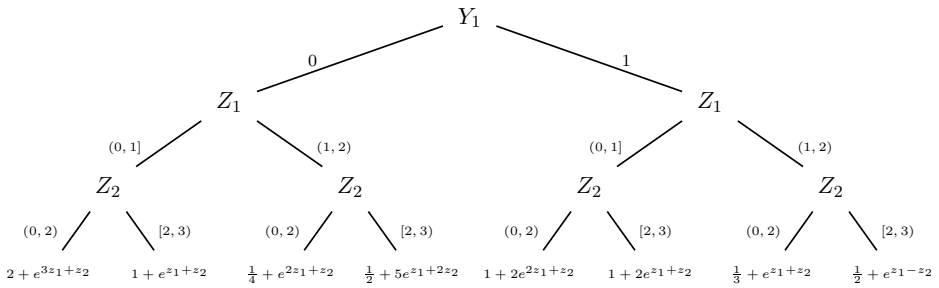

Figure 1: A mixed tree representing an MTE potential.

is the union of the intervals corresponding to the arcs Z-outgoing, (iii) every discrete variable has a number of outgoing arcs equal to its number of states and (iv) each leaf node contains an MTE potential defined on variables in the path from the root to that leaf.

3. Problem formulation

The goal of this paper is to introduce a method for answering queries in hybrid Bayesian networks with MTEs. We consider a hybrid Bayesian network defined for a set of variables $\mathbf{X}$. A *query* is a question about a probability value for a target variable $W \in \mathbf{X}$ given that the values of some variables $\mathbf{E} \subset \mathbf{X}$ are known. Thus, if we write $\mathbf{X} = (W, \mathbf{Y}^\mathsf{T}, \mathbf{Z}^\mathsf{T}, \mathbf{E}^\mathsf{T})^\mathsf{T}$, where $\mathbf{Y} = (Y_1, \ldots, Y_d)^\mathsf{T}$ represents the non-observed discrete variables and $\mathbf{Z} = (Z_1, \ldots, Z_c)^\mathsf{T}$ represents the non-observed continuous variables and $\mathbf{E} = (E_1, \ldots, E_k)^\mathsf{T}$, then a query about W given that $\mathbf{E} = \mathbf{e}$ is

$$P(a < W < b | \mathbf{E} = \mathbf{e}) = \frac{\displaystyle\int_a^b \left(\sum_{\mathbf{y} \in \mathbf{Y}} \int_{\Omega_\mathbf{Z}} \phi(w, \mathbf{y}, \mathbf{z}, \mathbf{e}) d\mathbf{z} \right) dw}{\phi_\mathbf{E}(\mathbf{e})} \tag{3}$$

if W is a continuous variable. The function ϕ in Eq. (3) is the joint distribution in the network and $\phi_{\mathbf{E}}$ is its marginal over variables $\mathbf{E}$. Let ϕ_X denote the conditional distribution of any variable X in the network. Then, the joint distribution is defined as

$$\phi(w, \mathbf{y}, \mathbf{z}, \mathbf{e}) = \\ \phi_W(w|pa(w)) \prod_{i=1}^{d} \phi_{Y_i}(y_i|pa(y_i)) \prod_{j=1}^{c} \phi_{Z_j}(z_j|pa(z_j)) \prod_{l=1}^{k} \phi_{E_l}(e_l|pa(e_l)). \quad (4)$$

Since our goal is to answer a query given a fixed value $\mathbf{e}$ of variables $\mathbf{E}$, we will rather be interested in the restriction of the joint distribution to the knowledge that $\mathbf{E} = \mathbf{e}$. We will replace any symbol ϕ in Eq. (4) by ψ, where the new symbols means the former function restricted to $\mathbf{e}$. With this notation, the joint distribution restricted to $\mathbf{e}$ can be written as

$$\psi(w, \mathbf{y}, \mathbf{z}) = \\ \psi_W(w|pa(w)) \prod_{i=1}^{d} \psi_{Y_i}(y_i|pa(y_i)) \prod_{j=1}^{c} \psi_{Z_j}(z_j|pa(z_j)) \prod_{l=1}^{k} \psi_{E_l}(e_l|pa(e_l)). \quad (5)$$

So, the numerator in Eq. (3) can be obtained as

$$\begin{aligned} \int_a^b \left(\sum_{\mathbf{y} \in \mathbf{Y}} \int_{\Omega_{\mathbf{Z}}} \phi(w, \mathbf{y}, \mathbf{z}, \mathbf{e}) d\mathbf{z} \right) dw &= \int_a^b \left(\sum_{\mathbf{y} \in \mathbf{Y}} \int_{\Omega_{\mathbf{Z}}} \psi(w, \mathbf{y}, \mathbf{z}) d\mathbf{z} \right) dw \\ &= \int_a^b h(w) dw, \end{aligned} \quad (6)$$

where $h(w) = \sum_{\mathbf{y} \in \mathbf{Y}} \int_{\Omega_{\mathbf{Z}}} \psi(w, \mathbf{y}, \mathbf{z}) d\mathbf{z}$. To finally answer the query expressed in Eq. (3), we still have to compute $\phi_{\mathbf{E}}(\mathbf{e})$. This is obtained as

$$\phi_{\mathbf{E}}(\mathbf{e}) = \int_{\Omega_W} \left(\sum_{\mathbf{y} \in \mathbf{Y}} \int_{\Omega_{\mathbf{Z}}} \psi(w, \mathbf{y}, \mathbf{z}) d\mathbf{z} \right) dw = \int_{\Omega_W} h(w) dw. \quad (7)$$

On the other hand, if W is discrete, a query is formulated as

$$P(W = w|\mathbf{E} = \mathbf{e}) = \frac{\sum_{\mathbf{y}\in\mathbf{Y}} \int_{\Omega_\mathbf{Z}} \phi(w, \mathbf{y}, \mathbf{z}, \mathbf{e}) d\mathbf{z}}{\phi_\mathbf{E}(\mathbf{e})}, \tag{8}$$

where $w \in \Omega_W$. The numerator of Eq. (8) can be expressed as

$$\sum_{\mathbf{y}\in\mathbf{Y}} \int_{\Omega_\mathbf{Z}} \phi(w, \mathbf{y}, \mathbf{z}, \mathbf{e}) d\mathbf{z} = \sum_{\mathbf{y}\in\mathbf{Y}} \int_{\Omega_\mathbf{Z}} \psi(w, \mathbf{y}, \mathbf{z}) d\mathbf{z} = h(w). \tag{9}$$

A similar procedure is carried out to compute the denominator of Eq. (8):

$$\phi_\mathbf{E}(\mathbf{e}) = \sum_{w\in\Omega_W} \sum_{\mathbf{y}\in\mathbf{Y}} \int_{\Omega_\mathbf{Z}} \psi(w, \mathbf{y}, \mathbf{z}) d\mathbf{z} = \sum_{w\in\Omega_W} h(w). \tag{10}$$

Hence, answering the queries formulated in Eqs. (3) and (8), requires the computation of the expressions in Eqs. (6), (7), (9) and (10). The problem is that in all the cases, the calculations are carried out over the joint distribution, which size is exponential in the number of variables in the network. Therefore, if the number of variables is high, it can be difficult or even impossible to represent such a joint distribution in a decision support system, specially if memory resources are limited. In the next section we propose a solution for approximating the quantities required to answer the queries, keeping the complexity bounded. The solution is based on the use of the *importance sampling* technique [39].

4. Answering queries using importance sampling

4.1. Continuous target variable

We will start off by considering the case in which the target variable, W, is continuous. Let us denote by θ the numerator of Eq. (3). We can write θ as

$$\theta = \int_a^b h(w) dw = \int_a^b \frac{h(w)}{f^*(w)} f^*(w) dw = E_{f^*}\left[\frac{h(W^*)}{f^*(W^*)}\right], \tag{11}$$

where f^* is a probability density function on (a, b) called *sampling distribution*, and W^* is a random variable with density f^*. Let $W_1^*, \ldots, W_m^*$ be a sample drawn from f^*. Then it is easy to prove that

$$\hat{\theta}_1 = \frac{1}{m} \sum_{i=1}^{m} \frac{h(W_i^*)}{f^*(W_i^*)} \tag{12}$$

is an unbiased estimator of θ. This procedure is called *importance sampling*.

As $\hat{\theta}_1$ is unbiased, the error of the estimation is determined by its variance, which is

$$\mathrm{Var}(\hat{\theta}_1) = \mathrm{Var}\left(\frac{1}{m} \sum_{i=1}^{m} \frac{h(W_i^*)}{f^*(W_i^*)} \right) = \frac{1}{m} \mathrm{Var}\left(\frac{h(W^*)}{f^*(W^*)} \right). \tag{13}$$

In order to minimise the variance in the expression above, f^* must be selected in such a way that the ratio between h and f^* be as constant as possible within interval (a, b). Actually, the minimum variance is reached when f^* is proportional to h in that interval, but that is of no practical value, as we are assuming that h, which is equivalent to the joint distribution, is difficult to handle. Later on we will show in detail a way to obtain an approximation to h, but keeping the complexity bounded. Let h^* be such an approximation. Then it holds that

$$f^*(w) = \frac{h^*(w)}{\int_a^b h^*(w) dw}, \quad a < w < b, \tag{14}$$

is a probability density function within interval (a, b). Therefore, in order to apply importance sampling to answer our target query, we have to find an approximation, h^*, of h and then obtain a sampling distribution from it, according to Eq. (14). Finally, we can estimate θ using Eq. (12).

On the other hand, $\phi_{\mathbf{E}}(\mathbf{e})$ can be estimated using importance sampling as well. In principle, a new sample should be generated, since the integral

range in this case is the entire domain of W, and not only interval (a, b). To avoid generating two different samples, we can consider the following density:

$$f_2^*(w) = \frac{h^*(w)}{\int_{\Omega_W} h^*(w)dw}, \tag{15}$$

which is a density for Ω_W. From this, we can generate a sample $W_1^*, \ldots, W_m^*$. Then, it holds that

$$\hat{\delta} = \frac{1}{m} \sum_{i=1}^{m} \frac{h(W_i^*)}{f_2^*(W_i^*)} \tag{16}$$

is an unbiased estimator of $\phi_{\mathbf{E}}(\mathbf{e})$.

Now, if we write $W_{(1)}^*, \ldots, W_{(k)}^*$ for the elements from sample $W_1^*, \ldots, W_m^*$ that fall inside interval (a, b), then it can be shown that

$$\hat{\theta}_2 = \frac{1}{k} \sum_{i=1}^{k} \frac{h(W_{(i)}^*)}{f_2^*(W_{(i)}^*)} \tag{17}$$

is an unbiased estimator of θ. Next proposition establishes the impact of using the same sample on the accuracy of the estimation.

Proposition 1. *Let $m, k, \hat{\theta}_2$ and $\hat{\delta}$ be as in Eqs.* (16) *and* (17)*. Then,*

$$\mathrm{Var}(\hat{\theta}_2) \leq \frac{m}{k}\mathrm{Var}(\hat{\delta}) + \frac{\phi_{\mathbf{E}}(e)^2}{2k} \ . \tag{18}$$

Proof. Let functions h and f_2^* be as in Eqs. (16) and (17). We define ξ, ξ_1 and ξ_2 as $\xi(w) = \frac{h(w)}{f_2^*(w)}$, $\xi_1(w) = \frac{h(w)I_{(a,b)}(w)}{f_2^*(w)}$ and $\xi_2(w) = \frac{h(w)I_{\mathbb{R}\setminus(a,b)}(w)}{f_2^*(w)}$, $w \in \mathbb{R}$, where $a, b \in \mathbb{R}$, $I_{(a,b)}(w) = 1$ if $w \in (a, b)$ and 0 otherwise, and $I_{\mathbb{R}\setminus(a,b)}(w) = 0$ if $w \in (a, b)$ and 1 otherwise.

It is clear that $\xi = \xi_1 + \xi_2$ and $\xi_1 \times \xi_2 = 0$. Also, notice that the expected values of ξ_1 and ξ_2 can be written, respectively, as $E[\xi_1] = P(a < W < b|\mathbf{E} = \mathbf{e})\phi_{\mathbf{E}}(e)$ and $E[\xi_2] = P(W \notin (a, b)|\mathbf{E} = \mathbf{e})\phi_{\mathbf{E}}(e)$.

Then,

$$\begin{aligned}
\mathrm{Var}(\xi) &= \mathrm{Var}(\xi_1+\xi_2) = \mathrm{Var}(\xi_1)+\mathrm{Var}(\xi_2)+2\mathrm{Cov}(\xi_1,\xi_2)\\
&= \mathrm{Var}(\xi_1)+\mathrm{Var}(\xi_2)+2(E[\xi_1\xi_2]-E[\xi_1]E[\xi_2])\\
&= \mathrm{Var}(\xi_1)+\mathrm{Var}(\xi_2)-2P(a<W<b|\mathbf{E}=\mathbf{e})\phi_{\mathbf{E}}(e)P(W\notin(a,b)|\mathbf{E}=\mathbf{e})\phi_{\mathbf{E}}(e)\\
&= \mathrm{Var}(\xi_1)+\mathrm{Var}(\xi_2)-2\phi_{\mathbf{E}}(e)^2P(a<W<b|\mathbf{E}=\mathbf{e})(1-P(a<W<b|\mathbf{E}=\mathbf{e}))
\end{aligned}$$

Hence,

$$\begin{aligned}
&\mathrm{Var}(\xi_1) =\\
&\mathrm{Var}(\xi)-\mathrm{Var}(\xi_2)+2\phi_{\mathbf{E}}(e)^2P(a<W<b|\mathbf{E}=\mathbf{e})(1-P(a<W<b|\mathbf{E}=\mathbf{e}))\\
&\qquad\qquad \leq \mathrm{Var}(\xi)+\frac{1}{2}\phi_{\mathbf{E}}(e)^2,
\end{aligned}$$

since $\mathrm{Var}(\xi_2)\geq 0$ and $P(a<W<b|\mathbf{E}=\mathbf{e})(1-P(a<W<b|\mathbf{E}=\mathbf{e}))\leq\frac{1}{4}$. Thus,

$$\begin{aligned}
\frac{1}{m}\mathrm{Var}(\xi_1) &\leq \frac{1}{m}\mathrm{Var}(\xi)+\frac{\phi_{\mathbf{E}}(e)^2}{2m} \Rightarrow \frac{k}{m}\frac{1}{k}\mathrm{Var}(\xi_1)\leq\frac{1}{m}\mathrm{Var}(\xi)+\frac{\phi_{\mathbf{E}}(e)^2}{2m}\Rightarrow\\
\frac{k}{m}\mathrm{Var}(\hat{\theta}_2) &\leq \mathrm{Var}(\hat{\delta})+\frac{\phi_{\mathbf{E}}(e)^2}{2m}\Rightarrow \mathrm{Var}(\hat{\theta}_2)\leq\frac{m}{k}\mathrm{Var}(\hat{\delta})+\frac{\phi_{\mathbf{E}}(e)^2}{2k}.
\end{aligned}$$

□

Proposition 1 establishes that the variance of $\hat{\theta}_2$ is related to the variance of $\hat{\delta}$ by the inverse of the proportion of elements in the sample that fall within interval (a,b). It means that using a single sample does not increase the error of the estimation dramatically. Actually, if all the elements in the sample are inside the target interval, then the variance of both estimators is asymptotically the same, as the term $\phi_{\mathbf{E}}(e)^2/2k$ tends to 0 as k increases. Therefore, for large samples, the ratio between the variances of both estimators verify that $\frac{\mathrm{Var}(\hat{\theta}_2)}{\mathrm{Var}(\hat{\delta})}\leq\frac{m}{k}$.

Notice that, if we used two samples instead of one (i.e., we used $\hat{\theta}_1$ instead of $\hat{\theta}_2$), of size m for $\hat{\delta}$ and size k for $\hat{\theta}_1$, the ratio would be

$$\frac{\mathrm{Var}(\hat{\theta}_1)}{\mathrm{Var}(\hat{\delta})} = \frac{\frac{1}{k}\mathrm{Var}\left(\frac{h(W^*)}{f^*(W^*)}\right)}{\frac{1}{m}\mathrm{Var}\left(\frac{h(W^*)}{f_2^*(W^*)}\right)},$$

and according to Eqs. (14) and (15), it follows that

$$\frac{\mathrm{Var}(\hat{\theta}_1)}{\mathrm{Var}(\hat{\delta})} = \frac{\frac{(\int_a^b h^*(w)dw)^2}{k}\mathrm{Var}\left(\frac{h(W^*)}{h^*(W^*)}\right)}{\frac{(\int_{\Omega_W} h^*(w)dw)^2}{m}\mathrm{Var}\left(\frac{h(W^*)}{h^*(W^*)}\right)} = \frac{m}{k}\frac{(\int_a^b h^*(w)dw)^2}{(\int_{\Omega_W} h^*(w)dw)^2} \leq \frac{m}{k}.$$

The conclusion is that for large sample sizes, the variances of $\hat{\theta}_1$ and $\hat{\theta}_2$ are equally related to the variance of $\hat{\delta}$. Therefore, for large samples, the use of a single sample is worth it.

4.2. Discrete target variable

If the target variable is discrete, the procedure is analogous. More precisely, if W is discrete then from Eq. (9) it follows that

$$\begin{aligned}\sum_{\mathbf{y}\in\mathbf{Y}}\int_{\Omega_\mathbf{Z}}\phi(w,\mathbf{y},\mathbf{z},\mathbf{e})d\mathbf{z} &= \sum_{w'\in\Omega_W} h(w')I_w(w') = \sum_{w'\in\Omega_W}\frac{h(w')I_w(w')}{p^*(w')}p^*(w') \\ &= E_{p^*}\left[\frac{h(W^*)I_w(W^*)}{p^*(W^*)}\right],\end{aligned}$$

where p^* is any probability mass function defined on Ω_W, W^* is a discrete random variable with distribution p^*, and $I_w(x) = 1$ if $w = x$ and 0 otherwise. The rest of the procedure is analogous to the continuous case, that is, a sample $W_1^*, \ldots, W_m^*$ is generated from p^* and $\theta_d = \sum_{\mathbf{y}\in\mathbf{Y}}\int_{\Omega_\mathbf{Z}}\phi(w,\mathbf{y},\mathbf{z},\mathbf{e})d\mathbf{z}$ is estimated as

$$\hat{\theta}_d = \frac{1}{m}\sum_{i=1}^{m}\frac{h(W_i^*)I_w(W_i^*)}{p^*(W_i^*)}, \tag{19}$$

where subscript d indicates that this estimator is for the discrete case.

4.3. Obtaining a sampling distribution

The error in the estimation procedure above described, depends on the variance of the ratio h/f^*. Therefore the best behaviour is obtained if the sampling distribution is close to h, as we mentioned before. In [41] a method for computing an accurate sampling distribution for discrete Bayesian networks was developed. It is based on computing the sampling distribution for a given variable through a process of eliminating the other variables from the set of all the conditional distributions in the network, $H = \{p(\mathbf{x}_i|pa(\mathbf{x}_i)), i = 1, \ldots, n\}$. The procedure can be adapted to the case of a hybrid Bayesian network as follows. Let $\{X_1, \ldots, X_l\}$ be the set of all the variables in the network, except the target W and the observations $\mathbf{E}$. An elimination order σ is considered and variables are deleted according to such order: $X_{\sigma(1)}, \ldots, X_{\sigma(l)}$.

The deletion of a variable $X_{\sigma(i)}$ consists of marginalising it out from the combination of all the functions in H which are defined for that variable. More precisely, the steps are as follows:

- Let $\mathrm{dom}(f)$ denote the set of variables for which function f is defined.
- Let $H_{\sigma(i)} = \{f \in H | X_{\sigma(i)} \in \mathrm{dom}(f)\}$.
- Calculate

$$f_{\sigma(i)} = \prod_{f \in H_{\sigma(i)}} f \tag{20}$$

  and $f'_{\sigma(i)}$ defined on $\mathrm{dom}(f_{\sigma(i)}) \setminus \{X_{\sigma(i)}\}$, by

$$f'_{\sigma(i)}(\mathbf{y}) = \int_{x_{\sigma(i)} \in \Omega_{X_{\sigma(i)}}} f_{\sigma(i)}(\mathbf{y}, x_{\sigma(i)}) dx_{\sigma(i)} \quad \forall \mathbf{y} \in \Omega_{\mathrm{dom}(f_{\sigma(i)}) \setminus \{X_{\sigma(i)}\}}. \tag{21}$$

- Transform H into $H \setminus H_{\sigma(i)} \cup \{f'_{\sigma(i)}\}$.

Note that the integral in Eq.(21) would be a summatory if W were discrete. After deleting all the variables $X_{\sigma(1)}, \ldots, X_{\sigma(l)}$ from the set of distributions $H = \{p(\mathbf{x}_i|pa(\mathbf{x}_i)), i = 1, \ldots, n\}$, the remaining functions will depend only on W. If all the computations are exact, it was proved in [14] that the remaining function is actually the optimal sampling distribution.

However, the result of the products (see Eq. (20)) in the process of obtaining the sampling distribution may require a large amount of space to be stored, and therefore the algorithm in [41] approximates the result of the combinations by pruning the probability trees (in our case, mixed trees) used to represent the potentials. The price to pay is that the sampling distribution is not the optimal one and the accuracy of the estimations will depend on the quality of the approximations. Here we propose a strategy for approximating the MTE potentials resulting from the products in Eq. (20). We will explain the idea by considering an MTE potential defined for a set of continuous variables $\mathbf{Z} = (Z_1, \ldots, Z_t)^{\mathsf{T}}$ as $\phi(\mathbf{z}) = a_0 + \sum_{i=1}^{t} a_i e^{\mathbf{b}_i^{\mathsf{T}}\mathbf{z}}$.

The goal is to detect those exponential terms in $\phi(\mathbf{z})$ that are almost constant and remove them. The rationale behind this strategy is that, from the point of view of simulation, a flat or constant term does not provide any useful information to the entire density, as there is already a constant term, namely a_0.

Thus, we consider a threshold $\alpha \in (0, 1)$ and then, for each term $g_j(\mathbf{z}) = a_j e^{\mathbf{b}_j^{\mathsf{T}}\mathbf{z}}$, $j = 1, \ldots, t$, in the mixture, if the condition $\frac{\min(g_j(\mathbf{z}))}{\max(g_j(\mathbf{z}))} > \alpha$ is satisfied, then $g_j(\mathbf{z})$ is replaced by $k_j = \int_{\mathbf{z}} g_j(\mathbf{z}) d\mathbf{z}$.

The closer to 1 α is, the more accurate the approximation. Note that the previous statements can be made taking into account that the exponential function by nature is strictly increasing or decreasing on its whole domain, and therefore its maximum and minimum are always located at the borders

of the domain. In this way, the shape of the function can be controlled.

Summing up, if the j-th term of the mixture is replaced by constant k_j, except in the cases where the resulting density could have negative values. To avoid the presence of negative values, we correct the value of k_j by making $k_j = \max\{\min_{\mathbf{z}}\{g_j(\mathbf{z})\}, \int_{\mathbf{z}} g_j(\mathbf{z})d\mathbf{z}\}$. Thus, the resulting potential is

$$\hat{\phi}(\mathbf{z}) = k + k_j + \sum_{\substack{i\in\{1,\ldots,t\}\\ i\neq j}} a_i e^{\mathbf{b}_i^{\mathsf{T}}\mathbf{z}} \ ,$$

But in fact, MTE potentials are defined into hypercubes. Therefore, rather than approximating a single potential, after each product the whole mixed tree representing the resulting potential should be approximated following this strategy. The detailed procedure can be found in Alg. 1.

4.4. Answering multiple queries simultaneously

The procedure described so far is designed to answer queries concerning a single variable at a time. We will show in this section that it can be extended to allow the possibility of answering multiples queries about different variables at the same time. The idea is based on the elimination procedure described in Sec. 4.3.

It is possible to carry out a simulation in an order contrary to the one in which variables are deleted. To obtain a value for $X_{\sigma(i)}$, the function $f_{\sigma(i)}$ obtained in the deletion of this variable is used. This function is defined for the values of variable $X_{\sigma(i)}$ and other variables already sampled. Function $f_{\sigma(i)}$ is restricted to the already obtained values of variables in $\mathrm{dom}(f_{\sigma(i)}) \setminus \{X_{\sigma(i)}\}$, giving rise to a density function which depends only on $X_{\sigma(i)}$. Finally, a value for this variable is drawn from this density. If all the computations are exact, it was proved in [14] that the simulation is actually

Algorithm 1: **PruneMTEPotential**($\mathcal{T}$,α)

Input: An mixed tree $\mathcal{T}$ and a threshold α for pruning terms.

Output: Tree $\mathcal{T}$ with terms pruned according to α.

1 Let $\mathbf{Z}$ be the set of continuous variables of tree $\mathcal{T}$.

2 **foreach** *leaf in* $\mathcal{T}$ **do**

3 Let $\phi(\mathbf{z}) = k + \sum_{i=1}^{t} a_i e^{\mathbf{b}_i^{\mathsf{T}}\mathbf{z}}$ be the MTE stored in the current leaf.

4 **for** $j := 1$ **to** t **do**

5 Let $a_j e^{\mathbf{b}_j^{\mathsf{T}}\mathbf{z}}$ be the j-term of $\phi(\mathbf{z})$.

6 **if** $\frac{min(a_j e^{\mathbf{b}_j^{\mathsf{T}}\mathbf{z}})}{max(a_j e^{\mathbf{b}_j^{\mathsf{T}}\mathbf{z}})} > \alpha$ **then**

7 $k_j := \max\{\min_{\mathbf{z}}\{a_j e^{\mathbf{b}_j^{\mathsf{T}}\mathbf{z}}\}, \int_{\mathbf{z}} a_j e^{\mathbf{b}_j^{\mathsf{T}}\mathbf{z}} d\mathbf{z}\}$.

8 Remove $a_j e^{\mathbf{b}_j^{\mathsf{T}}\mathbf{z}}$ from $\phi(\mathbf{z})$

9 Update the independent term k of $\phi(\mathbf{z})$ to $k + k_j$.

10 **return** $\mathcal{T}$.

carried out using the optimal density, and we obtain a sample from the joint distribution of $X_{\sigma(1)}, \ldots, X_{\sigma(l)}$.

The details of this procedure are given in Alg. 2, which computes a sampling distribution for each unobserved variable in a hybrid Bayesian network. Later on we will study how to determine the order of the variables in Step. 4. Now let us denote by $W_1, \ldots, W_n$ the unobserved variables in the network, and by $E_1, \ldots, E_k$ the observed ones. Note that after applying Alg. 2, if we set $\alpha = 1$ in Step. 7, then it holds that the true joint probability function is $f(w_1, \ldots, w_n, e_1, \ldots, e_k) = \prod_{i=1}^{l} f^*_{X_i}$. That is, if we simulate each variable X_i using $f^*_{X_i}$, we would actually be obtaining a sample of random vectors $\mathbf{w}_1, \ldots, \mathbf{w}_n, \mathbf{e}_1, \ldots, \mathbf{e}_k$ from the true distribution.

Algorithm 2: **SamplingDistributions**($\mathcal{B}$,**e**)

Input: A hybrid BN, $\mathcal{B}$, and an observation **e**.

Output: A sampling distribution for each variable in the network.

Let $H := \{\psi_{X_1}, \ldots, \psi_{X_l}\}$ be all the potentials in $\mathcal{B}$ restricted to the evidence **e**, represented as mixed trees.

$S := \emptyset$.

for $i := 1$ **to** l **do**

    Select the next variable to remove, X_i.

    $H_{X_i} := \{\psi \in H | X_i \in \text{dom}(\psi)\}$.

    $f_{X_i} := \prod_{\psi \in H_{X_i}} \psi$.

    $f^*_{X_i} :=$ **PruneMTEPotential**(f_{X_i}, α).

    $S := S \cup \{f^*_{X_i}\}$; $H := H \setminus H_{X_i}$.

    if X_i *is continuous* **then**

        $H := H \cup \{\int_{X_i} f^*_{X_i} dx_i\}$.

    else

        $H := H \cup \{\sum_{X_i} f^*_{X_i}\}$.

return S.

Our goal in this section is to answer a set of queries about the unobserved variables expressed as $P(W_i = w_i | \mathbf{E} = \mathbf{e})$ or $P(a_i < W_i < b_i | \mathbf{E} = \mathbf{e})$, $i = 1, \ldots, n$, if W_i is discrete or continuous, respectively. It can be shown that we can use the joint sample to estimate the different probabilities separately, since each individual sample is itself a sufficient statistic for the probability of a precise variable.

Let $W_1^{(j)}, \ldots, W_n^{(j)}$, $j = 1, \ldots, m$ be a sample of size m drawn from the

sampling distribution in the set S returned by Alg. 2. Then

$$\hat{\delta}_2 = \frac{1}{m} \sum_{j=1}^{m} \frac{\psi(W_1^{(j)}, \ldots, W_n^{(j)})}{\prod_{i=1}^{n} f_{W_i}^*(W_i^{(j)})} \tag{22}$$

is an unbiased estimator of $\phi_{\mathbf{E}}(\mathbf{e})$.

Let $W_1^{(j)*}, \ldots, W_n^{(j)*}$, $j = 1, \ldots, r$ be the elements from the sample above that fall into interval (a_i, b_i) (or for which $W_i^{(j)} = w_i$ in the discrete case), $i = 1, \ldots, n$. Then

$$\hat{\theta}_{W_i} = \frac{1}{r} \sum_{j=1}^{r} \frac{\psi(W_1^{(j)*}, \ldots, W_n^{(j)*})}{\prod_{i=1}^{n} f_{W_i}^*(W_i^{(j)*})} \tag{23}$$

is an unbiased estimator of $\int_a^b \left(\sum_{\mathbf{y} \in \mathbf{Y}} \int_{\Omega_{\mathbf{Z}}} \phi(w_i, \mathbf{y}, \mathbf{z}, \mathbf{e}) d\mathbf{z} \right) dw_i$, $i = 1, \ldots, n$ (see Eq. (3)). A similar result can be derived immediately in the case that W_i is discrete, and therefore the quantity to estimate is $\sum_{\mathbf{y} \in \mathbf{Y}} \int_{\Omega_{\mathbf{Z}}} \phi(w_i, \mathbf{y}, \mathbf{z}, \mathbf{e}) d\mathbf{z}$ (see Eq. (8)). In Eqs. (22) and (23), function ψ in the numerator is defined in a similar way as in Eq. (5), i.e. the product of conditionals restricted to the observations.

5. The algorithm

In this section we give the details of the algorithm that implements our proposal for answering multiples queries in hybrid Bayesian networks with MTEs using importance sampling. First of all it should be emphasised that Alg. 2 makes a decision about which variable to remove in each iteration (see Step 4). The decision there influences the complexity of the product in Step 6, since it determines the set of potentials that will be multiplied. We propose to use a one-step look-ahead heuristic based on selecting the variable

that results in a potential of lowest size[1] after the product in Step 6.

Though it is not possible to know beforehand the exact size of a potential resulting from a product, an upper bound is given in [40]. This is the bound actually used for deciding the elimination order in Alg. 2. In this point, we have all the tools necessary for establishing our proposal for answering multiple queries, which is described in Alg. 3.

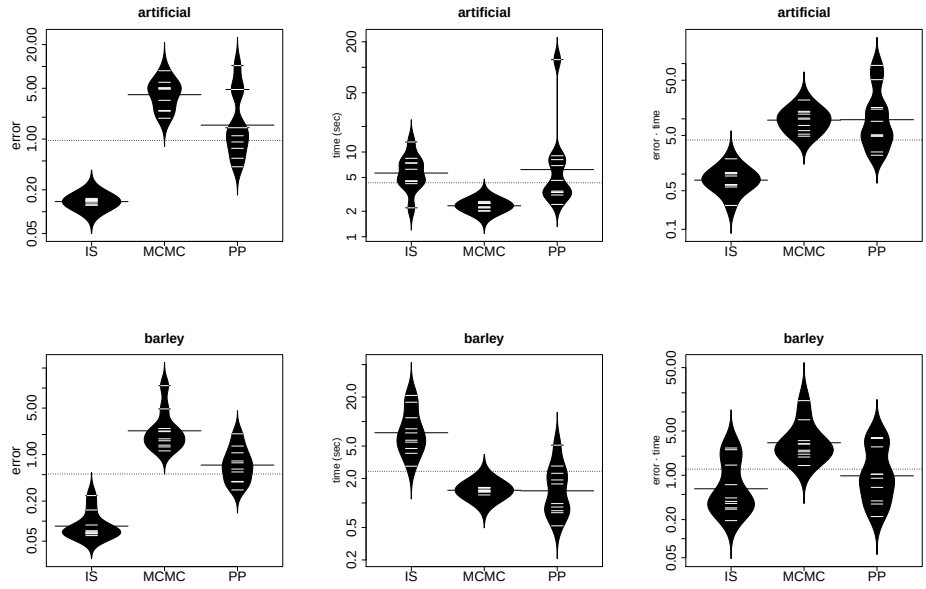

Figure 2: Beanplots of the χ^2 error, execution time and the rate $error \times time$ for the queries in networks `artificial` and `barley`.

[1]The size of an MTE potential is defined as its number of exponential terms, including the independent term.

Algorithm 3: **AnswerQueries**($\mathcal{B}$,e,Q)

Input: A hybrid BN $\mathcal{B}$ with variables $\mathbf{X}$. An observation $\mathbf{e}$ about a set of variables $\mathbf{E}$. A list of queries Q of the form $P(a_i < W_i < b_i \mid \mathbf{e})$ if W_i is continuous and $P(W_i = w_i \mid \mathbf{e})$ otherwise.

Output: Estimations $\hat{P}(a_i < W_i < b_i \mid \mathbf{e})$ or $\hat{P}(W_i = w_i \mid \mathbf{e})$.

Let $W_1, \ldots, W_n$ be the variables in $\mathbf{X} \setminus \mathbf{E}$.
$S :=$ **SamplingDistributions**($\mathcal{B}$,e)
Initialise $r_i := 0$ and $\hat{P}_i := 0$, $i = 1, \ldots, n$, and $\hat{\phi}(\mathbf{e}) := 0$.
for $j := 1$ **to** m **do**
    Generate a sample $w_1^*, \ldots, w_n^*$ for variables $W_1^{(j)}, \ldots, W_n^{(j)}$ by simulating in reverse order to the one used in Alg. 2, using the sampling distributions in S (see [40]).
    for $i := 1$ **to** n **do**
        if W_i *is continuous* **then**
            if $w_i^* \in (a_i, b_i)$ **then**
                $\hat{P}_i := \hat{P}_i + \frac{\psi(w_1^*,\ldots,w_n^*)}{\prod_{k=1}^n f^*_{W_k}(w_k^*)}$.
                $r_i := r_i + 1$.
        else
            if $w_i^* = w_i$ **then**
                $\hat{P}_i := \hat{P}_i + \frac{\psi(w_1^*,\ldots,w_n^*)}{\prod_{k=1}^n f^*_{W_k}(w_k^*)}$.
                $r_i := r_i + 1$.
    $\hat{\phi}(\mathbf{e}) := \hat{\phi}(\mathbf{e}) + \frac{\psi(w_1^*,\ldots,w_n^*)}{\prod_{k=1}^n f^*_{W_k}(w_k^*)}$.
$\hat{\phi}(\mathbf{e}) := \frac{\hat{\phi}(\mathbf{e})}{m}$.
$\hat{P}_i := \frac{\hat{P}_i}{r_i \times \hat{\phi}(\mathbf{e})}$, $i = 1, \ldots, n$.
return $\hat{P}_1, \ldots, \hat{P}_n$.

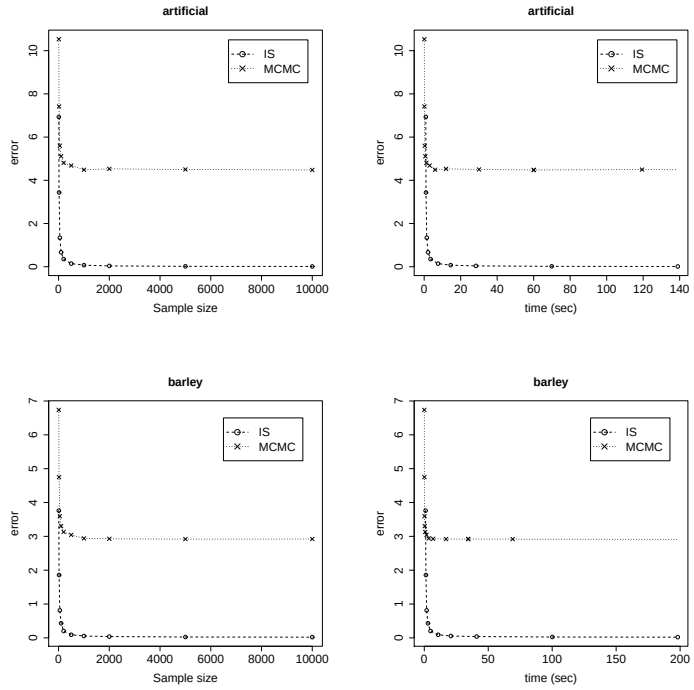

Figure 3: χ^2 error for methods IS and MCMC as a function of the sample size and execution time. Results for networks `artificial` and `barley`.

6. Experimental evaluation

A series of experiments was carried out with the aim of analysing the performance of the proposed methodology. We have used two hybrid Bayesian networks. The first one, denoted as `artificial`, is an artificial network with 97 variables, whose structure and parameters were generated at random, in the same way as the networks used in [40].

The second one has been created taking the structure from the `barley` network [21], which is originally fully discrete, and making some assumptions about the kind of the variables. Out of the 48 variables in the network, 10

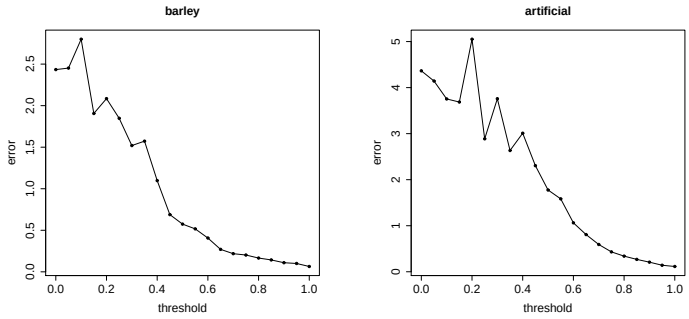

Figure 4: χ^2 error for different levels of pruning. The higher the α threshold, the less pruning is actually carried out. Results for networks `barley` and `artificial`.

of them were considered as discrete with two states, and the remaining were considered continuous with support in the interval $[0, 1]$. The domain of each continuous variable was split into two pieces. The MTE densities associated with each split were defined using 2 exponential terms, with parameters generated at random as in [40]. For each network, 20% of the variables were observed at random, considering as goal variables the remaining 80%. For each network, we considered 10 different observations. The queries were also selected at random, with uniform probability for each value of the discrete variables, and considering an interval of width of a 10% of its support for each continuous target variable.

6.1. Experiment 1

In this experiment we compared the performance of the Importance Sampling (IS) algorithm versus the other two approximate propagation methods existing in the literature for MTE networks: Markov Chain Monte Carlo (MCMC) and Penniless Propagation (PP) [40]. The version of the MCMC algorithm used in this paper is the adaptation for MTEs described in [40].

For each set of observations, the execution time and the error in the estimations were computed. The error was calculated using the χ^2 divergence, which is defined as

$$\chi^2 = \frac{1}{n}\sum_{i=1}^{n}\frac{(\hat{p}_i - p_i)^2}{p_i},$$

where p_i, $i = 1, \ldots, n$ are the true probabilities for each query, and $\hat{p}_i$, $i = 1, \ldots, n$ are their estimations. The true probabilities have been computed using the Variable Elimination algorithm [49]. Notice that, using that algorithm it is possible to obtain the exact probabilities, but the time required is too long compared with the three approximate methods analysed here.

Fig. 2 shows the results of the experiment for networks `artificial` and `barley`, respectively, represented as beanplots [19], which are extended versions of the well known box-plots where the empirical distribution of the data is also shown. The three beanplots correspond to the χ^2 error, execution time and the rate $error \times time$ obtained for a set of 10 observations. Each execution of the simulation algorithms (IS and MCMC) was repeated 10 times, using in both cases a sample of size 500. The results shown correspond to the average over the 10 executions. In order to simplify the potentials during the propagation, we have set a threshold $\alpha = 0.95$ for the mixed trees in the IS algorithm (see Sec. 4.3) and for algorithm PP we chose the following parameters, taken from [40]: $\epsilon_{Join} = 0.05$, $\epsilon_{Disc} = 0.05$. We refer the readers to the original reference for a detailed explanation of the meaning of those parameters. We limited the maximum number of exponential terms in the PP algorithm to 2.

The experimental results show how the IS algorithm clearly outperforms the other two in terms of accuracy, speed and rate $error \times time$ for network

`artificial`. For network `barley`, the error is again lower for IS, but in exchange the running time is the worse. This is due to the higher complexity of the potentials involved in this network, which makes the algorithm invest much time on obtaining the sampling distributions. However, the time invested is worth it, as can be seen looking at the plot corresponding to the rate $error \times time$, which is better for IS. Therefore, we conclude that this experiments suggest that IS offers the best way for dealing with the tradeoff between complexity and accuracy when answering multiple queries.

6.2. Experiment 2

The second experiment is devoted to analyse the impact of the sample size as well as the execution time in the behaviour of the simulation algorithms, that is IS and MCMC. Fig. 3 shows the χ^2 divergence as a function of sample size and time, for the two networks considered. It can be seen that IS converges more quickly than MCMC, and also converges to a more accurate solution. The results are consistent with the known tendency of MCMC in Bayesian networks, to fall in regions of the sample space conformed by configurations of low probability [15].

6.3. Experiment 3

The third experiment was aimed at testing the impact of using the pruning method proposed in Sec. 4.3. More precisely, we performed a test consisting of running the algorithm with different α thresholds and measuring the χ^2 error of the predictions. As in previous experiments, for each of the 10 observations, the algorithm was run 10 times. The results displayed in Fig. 4 show the average of the errors obtained. As expected, the error decreases as we increase the threshold, which means that we are being more strict with the pruning criterion.

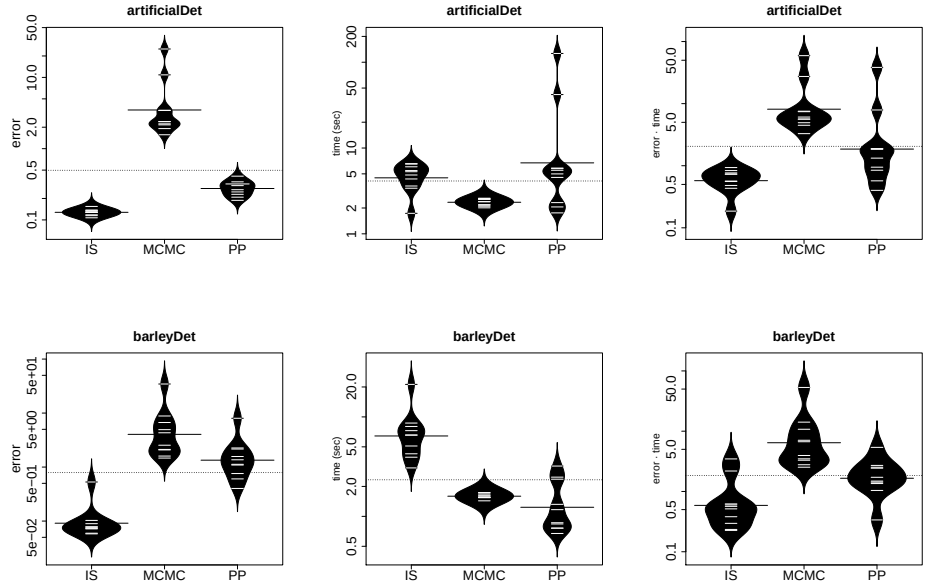

Figure 5: Beanplots of the χ^2 error, execution time and the rate $error \times time$ for the queries in networks `artificial` and `barley` with deterministic relations.

6.4. Experiment 4

Finally, we replicated the three experiments described above including deterministic relations in the used networks. We only considered deterministic conditionals for discrete variables, as the MTE model does not support this kind of relations among continuous variables beyond linear dependencies involving a single variable [5].

In order to include deterministic conditionals, we selected at random 80% of the discrete variables and then set to 1 the probability of one of its possible values, and to 0 the remaining probabilities. The results are displayed in Figs. 5 to 7. It can be seen that the performance of the algorithm in the presence of deterministic relations is similar to the general case.

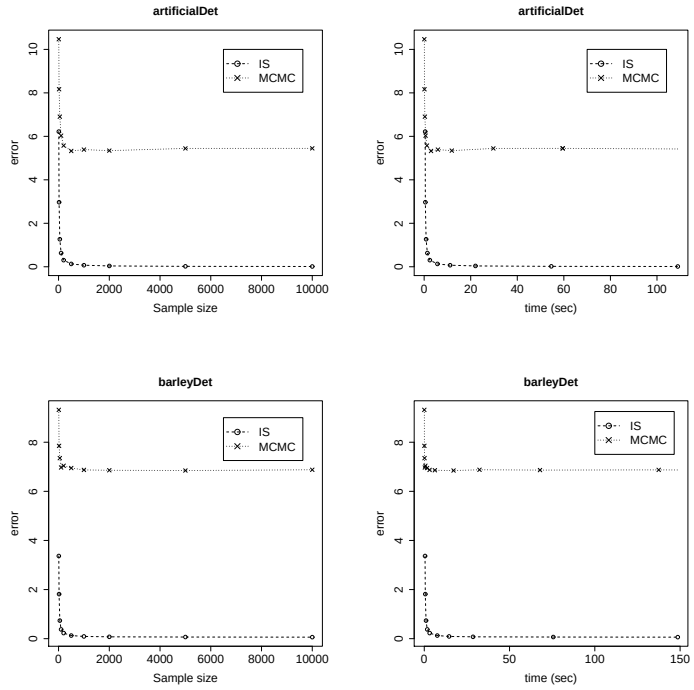

Figure 6: χ^2 error for methods IS and MCMC as a function of the sample size and execution time. Results for networks `artificial` and `barley` with deterministic relations.

7. Conclusions

We have introduced a method for solving multiples queries in hybrid Bayesian networks with MTEs. The method is based on importance sampling, which makes it an anytime algorithm. The algorithm is able to compute answers to multiple questions using a unique sample. We have shown that the variance remains bounded if the same sample is also used to compute the numerator and denominator in each query.

The experiments conducted illustrate the behaviour of the proposed algorithm, and they support the idea that the IS algorithm outperforms the

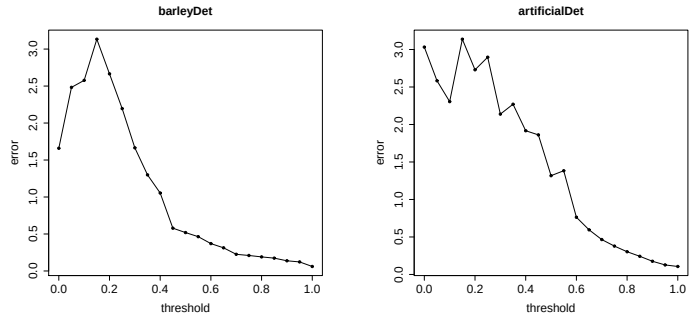

Figure 7: χ^2 error for different levels of pruning. The higher the α threshold, the less pruning is actually carried out. Results for networks `barley` and `artificial` with deterministic relations.

two algorithms previously used for carrying out probabilistic reasoning in hybrid Bayesian networks with MTEs. Therefore, the methodology introduced here expands the class of problems that can be handled using hybrid Bayesian networks, and more precisely, it provides versatility to the MTE model, by increasing the efficiency in solving probabilistic inference tasks.

We expect to continue this research line by developing methods for answering more complex queries. For instance, a query consisting on finding the most probable explanation to an observed fact in terms of a set of target variables, which is called *abductive inference* [11]. We also plan to study the application of the proposed algorithm to MOPs [42]. The main difference would be in Alg. 1, as in the case of MOPs, each term may oscillate within an interval, while MTEs are smoother.

Acknowledgements

Work supported by the Spanish Ministry of Science and Innovation, projects TIN2007-67418-C03-02, TIN2010-20900-C04-02 and ERDF funds.

References

[1] X. Bai, Predicting consumer sentiments from online text, Decision Support Systems 50 (2011) 732–742.

[2] C. Butz, S. Hua, K. Konkel, H. Yao, Join tree propagation with prioritized messages, Networks 55 (2010) 350–359.

[3] C. Butz, K. Konkel, P. Lingras, Join tree propagation utilizing both arc reversal and variable elimination, International Journal of Approximate Reasoning 52 (2010) 948–959.

[4] J. Cheng, M. J. Druzdzel, AIS-BN: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks, Journal of Artificial Intelligence Research 13 (2000) 155–188.

[5] E. Cinicioglu, P. Shenoy, Arc reversals in hybrid Bayesian networks with deterministic variables, International Journal of Approximate Reasoning 50 (2009) 763–777.

[6] B. R. Cobb, Efficiency of influence diagram models with continuous decision variables, Decision Support Systems 48 (2009) 257–266.

[7] B. R. Cobb, P. P. Shenoy, R. Rumí, Approximating probability density functions with mixtures of truncated exponentials, Statistics and Computing 16 (2006) 293–308.

[8] G. F. Cooper, The computational complexity of probabilistic inference using Bayesian belief networks, Artificial Intelligence 42 (1990) 393–405.

[9] P. Dagum, M. Luby, Approximating probabilistic inference in Bayesian belief networks is NP-hard, Artificial Intelligence 60 (1993) 141–153.

[10] A. Fernández, M. Morales, C. Rodríguez, A. Salmerón, A system for relevance analysis of performance indicators in higher education using Bayesian networks, Knowledge and Information Systems 27 (2011) 327–344.

[11] J. Gámez, Abductive inference in Bayesian networks: A review, in: J. Gámez, S. Moral, A. Salmerón (eds.), Advances in Bayesian Networks, Springer Verlag, 2004, pp. 101–120.

[12] W. R. Gilks, S. Richardson, D. J. Spiegelhalter, Markov chain Monte Carlo in practice, Chapman and Hall, London, UK, 1996.

[13] V. Gogate, R. Dechter, SampleSearch: Importance sampling in presence of determinism, Artificial Intelligence 175 (2011) 694–729.

[14] L. D. Hernández, S. Moral, A. Salmerón, A Monte Carlo algorithm for probabilistic propagation in belief networks based on importance sampling and stratified simulation techniques, International Journal of Approximate Reasoning 18 (1998) 53–91.

[15] C. S. Jensen, A. Kong, U. Kjærulff, Blocking Gibbs sampling in very large probabilistic expert systems, International Journal of Human-Computer Studies 42 (1995) 647–666.

[16] F. V. Jensen, S. L. Lauritzen, K. G. Olesen, Bayesian updating in causal probabilistic networks by local computation, Computational Statistics Quarterly 4 (1990) 269–282.

[17] F. V. Jensen, T. D. Nielsen, Bayesian Networks and Decision Graphs, Springer, 2007.

[18] M. Jordan, An introduction to variational methods for graphical models, Machine Learning 37 (1999) 183–233.

[19] P. Kampstra, Beanplot: A boxplot alternative for visual comparison of distributions, Journal of Statistical Software 28 (2008) 1–9.

[20] D. Kozlov, D. Koller, Nonuniform dynamic discretization in hybrid networks, in: D. Geiger, P. P. Shenoy (eds.), Proceedings of the 13th Conference on Uncertainty in Artificial Intelligence, Morgan & Kaufmann, 1997, pp. 302–313.

[21] K. Kristensen, I. A. Rasmussen, The use of a Bayesian network in the design of a decision support system for growing malting barley with- out use of pesticides, Computers and Electronics in Agriculture 33 (2002) 197–217.

[22] H. Langseth, T. D. Nielsen, R. Rumí, A. Salmerón, Inference in hybrid Bayesian networks, Reliability Engineering and Systems Safety 94 (2009) 1499–1509.

[23] H. Langseth, T. D. Nielsen, R. Rumí, A. Salmerón, Mixtures of truncated basis functions, International Journal of Approximate Reasoning 53 (2012) 212–227.

[24] P. Larrañaga, S. Moral, Probabilistic graphical models in artificial intelligence, Applied Soft Computing 11 (2011) 1511–1528.

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

[26] S. L. Lauritzen, F. Jensen, Stable local computation with conditional Gaussian distributions, Statistics and Computing 11 (2001) 191–203.

[27] U. Lerner, Exact inference in networks with discrete children of continuous parents, in: in: J. Breese, D. Koller (Eds.), Uncertainty in Artificial Intelligence, Morgan Kaufmann, 2001, pp. 319–328.

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

[29] A. Madsen, Improvements to message computation in lazy propagation, International Journal of Approximate Reasoning 51 (2010) 499–514.

[30] A. Madsen, F. Jensen, Lazy propagation: a junction tree inference algorithm based on lazy evaluation, Artificial Intelligence 113 (1999) 203–245.

[31] S. Moral, R. Rumí, A. Salmerón, Mixtures of truncated exponentials in hybrid Bayesian networks, Lecture Notes in Artificial Intelligence 2143 (2001) 135–143.

[32] S. Moral, A. Salmerón, Dynamic importance sampling in Bayesian networks based on probability trees, International Journal of Approximate Reasoning 38 (2005) 245 – 261.

[33] M. Neil, M. Tailor, D. Marquez, Inference in bayesian networks using dynamic discretisation, Statistics and Computing 17 (1999) 219–233.

[34] M. Neil, M. Tailor, D. Marquez, N. Fenton, P. Hearty, Modelling dependable systems using hybrid Bayesian networks, Reliability Engineering and System Safety 93 (2008) 933–939.

[35] E. Ngai, Y. Hu, Y. Wong, Y. Chen, X. Sun, The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature, Decision Support Systems 50 (2011) 559–569.

[36] J. Pearl, Probabilistic reasoning in intelligent systems, Morgan-Kaufmann (San Mateo), 1988.

[37] F. Ramos, F. Cozman, Anytime anyspace probabilistic inference, International Journal of Approximate Reasoning 38 (2005) 53 – 80.

[38] V. Romero, R. Rumí, A. Salmerón, Learning hybrid Bayesian networks using mixtures of truncated exponentials, International Journal of Approximate Reasoning 42 (2006) 54–68.

[39] R. Y. Rubinstein, Simulation and the Monte Carlo Method, Wiley (New York), 1981.

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

[41] A. Salmerón, A. Cano, S. Moral, Importance sampling in Bayesian networks using probability trees, Computational Statistics and Data Analysis 34 (2000) 387–413.

[42] P. Shenoy, J. West, Inference in hybrid Bayesian networks using mixtures of polynomials, International Journal of Approximate Reasoning 52 (2011) 641–657.

[43] P. P. Shenoy, Binary join trees for computing marginals in the Shenoy-Shafer architecture, International Journal of Approximate Reasoning 17 (1997) 239–263.

[44] P. P. Shenoy, G. Shafer, Axioms for probability and belief function propagation, in: R. D. Shachter, T. S. Levitt, J. F. Lemmer, L. N. Kanal (eds.), Uncertainty in Artificial Intelligence 4, North Holland, Amsterdam, 1990, pp. 169–198.

[45] K. Xu, S. Liao, J. Li, Y. Song, Mining comparative opinions from customer reviews for competitive intelligence, Decision Support Systems 50 (2011) 743–754.

[46] H. Yu, R. van Engelen, Arc refractor methods for adaptive importance sampling on large Bayesian networks under evidential reasoning, International Journal of Approximate Reasoning 51 (2010) 800–819.

[47] C. Yuan, M. Druzdzel, Importance sampling algorithms for Bayesian networks: Principles and performance, Mathematical and Computer Modeling 43 (2005) 1189–1207.

[48] C. Yuan, M. Druzdzel, Theoretical analysis and practical insights into importance sampling for Bayesian networks, International Journal of Approximate Reasoning 46 (2007) 320–333.

[49] N. L. Zhang, D. Poole, Exploiting causal independence in Bayesian network inference, Journal of Artificial Intelligence Research 5 (1996) 301–328.