Proceedings of the Twenty-Ninth International
Florida Artificial Intelligence Research Society Conference

# Bayesian Networks with Conditional Truncated Densities

**Santiago Cortijo  Christophe Gonzales**
Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6, UMR 7606, Paris, France.
firstname.lastname@lip6.fr

## Abstract

The majority of Bayesian networks learning and inference algorithms rely on the assumption that all random variables are discrete, which is not necessarily the case in real-world problems. In situations where some variables are continuous, a trade-off between the expressive power of the model and the computational complexity of inference has to be done: on one hand, conditional Gaussian models are computationally efficient but they lack expressive power; on the other hand, mixtures of exponentials (MTE), bases or polynomials are expressive but this comes at the expense of tractability. In this paper, we propose an alternative model that lies in between. It is composed of a "discrete" Bayesian network (BN) combined with a set of monodimensional conditional truncated densities modeling the uncertainty over the continuous random variables given their discrete counterpart resulting from a discretization process. We show that inference computation times in this new model are close to those in discrete BNs. Experiments confirm the tractability of the model and highlight its expressive power by comparing it with MTE.

## Introduction

For several decades, Bayesian networks (BN) (Pearl 1988) have been successfully exploited for dealing with uncertainties. Their popularity has stimulated the development of many efficient learning and inference algorithms (Shafer 1996; Dechter 1999; Madsen and Jensen 1999; Bacchus, Dalmao, and Pitassi 2003; Chavira and Darwiche 2008). While these algorithms are relatively well understood when they involve only discrete variables, their ability to cope with continuous variables is often unsatisfactory. Dealing with continuous random variables is much more complicated than dealing with discrete ones and one actually has to trade-off between the expressive power of the uncertainty model and the computational complexity of its learning and inference mechanisms. Conditional Gaussian models and their mixing with discrete variables (Lauritzen and Wermuth 1989; Lauritzen 1992; Lerner, Segal, and Koller 2001) lie on one side of the spectrum. They compactly represent multivariate Gaussian distributions. As such, they lack expressive power but their inference mechanisms are computationally very efficient. On the other side of the spectrum, there are expressive models like mixtures of exponentials (MTE) (Moral, Rumí, and Salmerón 2001; Cobb, Shenoy, and Rumí 2006; Rumíand Salmerón 2007), mixtures of truncated basis functions (Langseth et al. 2012b; 2012a) and mixtures of polynomials (Shenoy 2011; Shenoy and West 2011; Shenoy 2012). Those can approximate very well arbitrary density functions but this comes at the expense of tractability: their exact inference computation times tend to grow exponentially with the number of continuous variables, which makes them unusable when they contain hundreds of random variables.

In this paper, we propose an alternative model that lies in between these two extremes. The key idea is to discretize the random variables, thereby mapping each (continuous) value of their domain into an interval within a *finite* set of intervals. Of course, whenever some discretization is performed, some information about the continuous random variables is lost. But this can be significantly alleviated by modeling the distribution of the continuous values within an interval by a density function which is not necessarily a uniform distribution (which is the implicit assumption when using a classical discretization). The set of density functions over all the intervals of a continuous variable constitutes its "*conditional truncated density*" given its discretized counterpart. Now, our uncertainty model is a (discrete) BN over the set of discrete and discretized random variables combined with the set of conditional truncated densities assigned to the continuous random variables that were discretized. This model is derived from the result of an algorithm for learning BNs from datasets containing both discrete and continuous random variables (Mabrouk et al. 2015).

By assigning conditional truncated densities to continuous variables, our model gains expressive power over a BN in which all continuous variables are discretized. For inference, the density functions need only be included in the BN as discrete evidence (computed by integrations) over the discretized variables and, then, only a classical inference over discrete variables is needed to complete the process. As the density functions are monodimensional, integrations can be performed quickly. So the inference computational complexity in our model is very close that of an inference in a classical BN, which makes it tractable. Experiments highlight this point but also the expressive power of the model.

The paper is organized as follows. In the next section, we recall related works. Then we present our model and its inference mechanism. Next, its efficiency and effectiveness are highlighted through experiments. Finally, a conclusion and some perspectives are provided in the last section.

## Related Works

In the rest of the paper, capital letters (possibly subscripted) refer to random variables and boldface capital letters to sets of variables. To distinguish continuous random variables from discrete ones, we denote by $\mathring{X}_i$ a continuous variable and by $X_i$ a discrete one. Without loss of generality, for any $\mathring{X}_i$, variable $X_i$ represents its discretized counterpart. Throughout the paper, let $\mathbf{X_D} = \{X_1, \ldots, X_d\}$ and $\mathring{\mathbf{X}}_\mathbf{C} = \{\mathring{X}_{d+1}, \ldots, \mathring{X}_n\}$ denote the set of discrete and continuous random variables respectively. We denote by $\mathcal{X} = \mathbf{X_D} \cup \mathring{\mathbf{X}}_\mathbf{C}$ the set of all random variables. Finally, for any variable $X$ or set of random variables $\mathbf{Y}$ or $\mathring{\mathbf{Y}}$, let $\Omega_X$ (resp. $\Omega_\mathbf{Y}$ or $\Omega_{\mathring{\mathbf{Y}}}$) denote the domain of $X$ (resp. $\mathbf{Y}$ or $\mathring{\mathbf{Y}}$).

The closest works related to our model are MTE, MOP and MTBF. In MTE (Moral, Rumí, and Salmerón 2001), the distribution over the set of all random variables $\mathcal{X}$ is specified by a density function $f$ such that:

- $\sum_{\mathbf{x_D} \in \Omega_{\mathbf{X_D}}} \int_{\Omega_{\mathring{\mathbf{X}}_\mathbf{C}}} f(\mathbf{x_D}, \mathring{\mathbf{x}}_\mathbf{C})\, d\mathring{\mathbf{x}}_\mathbf{C} = 1$,
- $f$ is an MTE potential over $\mathcal{X}$, i.e.:

**Definition 1 (MTE potential)** *Let $\mathbf{Y} = \{X_{r_1}, \ldots, X_{r_p}\}$ and $\mathring{\mathbf{Z}} = \{\mathring{X}_{s_1}, \ldots, \mathring{X}_{s_q}\}$ be sets of discrete and continuous variables respectively. A function $\phi : \Omega_{\mathbf{Y} \cup \mathring{\mathbf{Z}}} \mapsto \mathbb{R}_0^+$ is a MTE potential if one of the two following conditions holds:*

1. *$\phi$ can be written as:*

$$\phi(\mathbf{y}, \mathring{\mathbf{z}}) = a_0 + \sum_{i=1}^{m} a_i \exp\left\{ \sum_{j=1}^{p} b_i^{(j)} x_{r_j} + \sum_{k=1}^{q} b_i^{(p+k)} \mathring{x}_{s_k} \right\} \quad (1)$$

   *for all $(x_{r_1}, \ldots, x_{r_p}) \in \mathbf{Y}$, $(\mathring{x}_{s_1}, \ldots, \mathring{x}_{s_q}) \in \mathring{\mathbf{Z}}$, where $a_i$, $i = 0, \ldots, m$ and $b_i^{(j)}$, $i = 1, \ldots, m$, $j = 1, \ldots, p+q$, are real numbers.*
2. *There exists a partition $\Omega_1, \ldots, \Omega_k$ of $\Omega_{\mathbf{Y} \cup \mathring{\mathbf{Z}}}$ such that the domain of the continuous variables, $\Omega_{\mathring{\mathbf{Z}}}$, is divided into hypercubes, the domain $\Omega_\mathbf{Y}$ of the discrete variables is divided into arbitrary sets, and such that $\phi$ is defined as:*

$$\phi(\mathbf{y}, \mathring{\mathbf{z}}) = \phi_i(\mathbf{y}, \mathring{\mathbf{z}}) \quad \text{if } (\mathbf{y}, \mathring{\mathbf{z}}) \in \Omega_i,$$

   *where each $\phi_i$, $i = 1, \ldots, k$, can be written in the form of Eq. (1), i.e., it is a MTE potential on $\Omega_i$.*

MTEs present attractive features. First, they are expressive in the sense that they can approximate (w.r.t. the Kullback-Leibler distance) any continuous density function (Cobb, Shenoy, and Rumí 2006; Cobb and Shenoy 2006). Second, they are easy to learn from datasets (Moral, Rumí, and Salmerón 2002; Romero, Rumí, and Salmerón 2004). Finally, they satisfy Shafer-Shenoy's propagation axioms (Shenoy and Shafer 1990) and inference can thus be performed using a junction tree-based algorithm (Moral, Rumí, and Salmerón 2001; Cobb and Shenoy 2006).

This algorithm can be described as follows. An undirected graph called a Markov network is first created: its nodes correspond to the variables of $\mathcal{X}$ and its edges are such that, for every MTE potential $\phi_i$, all the nodes involved in $\phi_i$ are linked together. This graph is then triangulated by eliminating sequentially all the nodes. A node elimination consists i) in adding edges to the Markov network in order to create a clique (a complete subgraph) containing the eliminated node and all its neighbors; and ii) in removing the eliminated node and its adjacent edges from the Markov network. The cliques created during this process constitute the nodes of the junction tree. They are linked in order to satisfy a "running intersection" property (Madsen and Jensen 1999). Finally, each MTE potential $\phi_i$ is inserted into a clique containing all its variables.

A collect-distribute message-passing algorithm can then be performed in this junction tree, hence enabling to compute *a posteriori* marginal distributions of all the random variables. As usual, the message passed from one clique $\mathcal{C}_i$ to a neighbor $\mathcal{C}_j$ is the projection onto the variables in $\mathcal{C}_i \cap \mathcal{C}_j$ of the combination of the MTE potentials stored in $\mathcal{C}_i$ with the messages received by $\mathcal{C}_i$ from all its neighbors except $\mathcal{C}_j$. By Eq. (1), combinations and projections are Algebraic operations over sums of exponentials. Unfortunately, these operations have a serious shortcoming: when propagating messages from one clique to another, the number of $a_i$/exp terms in Eq. (1) tends to grow exponentially, hence limiting the use of this exact inference mechanism to problems with only a small number of cliques.

To overcome this issue, approximate algorithms based on MCMC are provided in the literature (Moral, Rumí, and Salmerón 2001; Rumíand Salmerón 2007).

Mixtures of polynomials (MOP) are similar to MTE except that functions $\phi : \Omega_{\mathbf{Y} \cup \mathring{\mathbf{Z}}} \mapsto \mathbb{R}_0^+$ of Eq. (1) are substituted by polynomials over the variables in $\mathbf{Y} \cup \mathring{\mathbf{Z}}$ (Shenoy 2011; Shenoy and West 2011). MOPs have several advantages over MTEs: their parameters for approximating density functions are easier to determine than those of MTEs. They are also applicable to a larger class of deterministic functions in hybrid BNs. As MTE, the MOP model satisfies Shafer-Shenoy's propagation axioms and inference can thus be performed by message-passing in a junction tree. But, similarly to Eq. (1), the number of terms these messages involve tends to grow exponentially with the number of cliques in the junction tree, thereby limiting the use the message-passing algorithm to junction trees with a small number of cliques/random variables.

Finally, mixtures of truncated basis functions (MTBF) generalize both MTEs and MOPs (Langseth et al. 2012b). The definition of an MTBF is the same as Definition 1 except that Eq. (1) is substituted by:

$$\phi(\mathbf{y}, \mathring{\mathbf{z}}) = \sum_{i=0}^{m} \prod_{k=1}^{q} a_{i,\mathbf{y}}^{(k)} \psi_i(\mathring{x}_{s_k}), \quad (2)$$

where potentials $\psi_i : \mathbb{R} \mapsto \mathbb{R}$ are basis functions. MTBFs

are defined so that the potentials are closed under combination and projection which, again, ensures that inference can be performed by message-passing in a junction tree. By exploiting cleverly factorizations of terms in Eq. (2), inference in MTBFs can be more efficient than in MTEs (Langseth et al. 2012a). But, like all the other aforementioned models, the sizes of the messages tend to grow with the number of cliques in the junction tree.

In the next section, we propose an alternative model that overcomes this issue while still being expressive.

## BNs with Conditional Truncated Densities

As mentioned in the introduction, our model combines a discrete BN $\mathcal{B}$ with a set of conditional truncated densities assigned to each continuous random variable.

**Definition 2 (Bayesian network)** *A (discrete) BN $\mathcal{B}$ is a pair $(\mathcal{G}, \boldsymbol{\theta})$ where $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ is a directed acyclic graph (DAG), $\mathbf{X} = \{X_1, ..., X_n\}$ represents a set of discrete and/or discretized random variables*[1]*, $\mathbf{A}$ is a set of arcs, and $\boldsymbol{\theta} = \{P(X_i|\mathbf{Pa}(X_i))\}_{i=1}^n$ is the set of the conditional probability tables/distributions (CPT) of the variables $X_i$ in $\mathcal{G}$ given their parents $\mathbf{Pa}(X_i)$ in $\mathcal{G}$. The BN encodes the joint probability over $\mathbf{X}$ as $P(\mathbf{X}) = \prod_{i=1}^n P(X_i|\mathbf{Pa}(X_i))$.*

A discretization of a continuous variable $\mathring{X}$ is a function $d_{\mathring{X}} : \Omega_{\mathring{X}} \to \{0, \ldots, g\}$ defined by an increasing sequence of $g$ cut points $\{t_1, t_2, ..., t_g\}$ such that:

$$d_{\mathring{X}}(\mathring{x}) = \begin{cases} 0 \text{ if } \mathring{x} < t_1, \\ k \text{ if } t_k \leq \mathring{x} < t_{k+1}, \text{ for all } k \in \{1, \ldots, g-1\} \\ g \text{ if } \mathring{x} \geq t_g. \end{cases}$$

Thus the discretized variable $X$ corresponding to $\mathring{X}$ has a finite domain of $\{0, \ldots, g\}$. Therefore, after discretizing all the continuous random variables, the uncertainty over all the discrete and discretized random variables $\mathbf{X} = \{X_1, ..., X_n\}$ can be represented by a classical BN in which very efficient message-passing inference mechanisms can be used, notably junction tree-based algorithms (Shafer 1996; Madsen and Jensen 1999).

However, discretizing continuous random variables raises two issues: i) which discretization function shall be used to minimize the loss of information? and ii) will the loss of information affect significantly the results of inference? A possible answer to the first question is to exploit "*conditional truncated densities*" (Mabrouk et al. 2015). The answer to the second question of course strongly depends on the discretization performed but, as we shall see, conditional truncated densities can limit the discrepancy between the exact *a posteriori* marginal density functions of the continuous random variables and the approximation they provide.

**Definition 3 (Conditional Truncated Density)** *Let $\mathring{X}$ be a continuous random variable. Let $d_{\mathring{X}}$ be a discretization of $\mathring{X}$ with set of cutpoints $\{t_1, t_2, ..., t_g\}$. Finally, let $X$ be a discrete random variable with domain $\Omega_X = \{0, \ldots, g\}$. A conditional truncated density is a function $f(\mathring{X}|X) : \Omega_{\mathring{X}} \times \Omega_X \mapsto \mathbb{R}_0^+$ satisfying the following properties:*

1. *$f(\mathring{x}|x) = 0$ for all $x \in \Omega_X$ and $\mathring{x} \notin [t_x, t_{x+1}]$ with, by abuse of notation $t_0 = \inf \Omega_{\mathring{X}}$ and $t_{g+1} = \sup \Omega_{\mathring{X}}$;*
2. *the following equation holds:*

$$\int_{t_x}^{t_{x+1}} f(\mathring{x}|x)\, d\mathring{x} = 1, \quad \textit{for all } x \in \Omega_X. \tag{3}$$

In other words, $f(\mathring{x}|x)$ represents the truncated density function of random variable $\mathring{X}$ over the interval of discretization $[t_x, t_{x+1}]$.

**Lemma 1** *Let $P(X)$ be any probability distribution over the discrete random variable $X$ of Definition 3. Then $f(\mathring{X}|X)P(X)$ is a density function over $\Omega_{\mathring{X}} \times \Omega_X$.*

**Proof.** By definition, $f(\mathring{X}|X)$ and $P(X)$ are non-negative real-valued functions, hence $f(\mathring{X}|X)P(X)$ is also a non-negative real-valued function. So, to prove that it is a density function, it is sufficient to show that:

$$\sum_{x \in \Omega_X} \int_{\Omega_{\mathring{X}}} f(\mathring{x}|x)P(x)\, d\mathring{x} = 1.$$

By Property 1., $f(\mathring{X} = \mathring{x}|X = x)P(X = x) = 0$ for all $x \in \Omega_X$ and $\mathring{x} \notin [t_x, t_{x+1}]$. So, the above equation is equivalent to:

$$\sum_{x \in \Omega_X} \int_{t_x}^{t_{x+1}} f(\mathring{x}|x)P(x)\, d\mathring{x} = 1,$$

which, by the fact that $x$ is a constant inside the integral and by Eq. (3), is also equivalent to:

$$\sum_{x \in \Omega_X} P(x) \int_{t_x}^{t_{x+1}} f(\mathring{x}|x)\, d\mathring{x} = \sum_{x \in \Omega_X} P(x) = 1.$$

As a consequence, $f(\mathring{X}|X)P(X)$ is a density function. ■

Let us introduce "*Bayesian networks with conditional truncated densities*" (ctdBN): let $\mathbf{X_D} = \{X_1, \ldots, X_d\}$ and $\mathring{\mathbf{X}}_\mathbf{C} = \{\mathring{X}_{d+1}, ..., \mathring{X}_n\}$ be sets of discrete and continuous random variables respectively. Let $\mathbf{X_C} = \{X_{d+1}, \ldots, X_n\}$ be a set of discretized variables resulting from the discretization of the variables in $\mathring{\mathbf{X}}_\mathbf{C}$. Then, a BN with conditional truncated densities is a pair $(\mathcal{G}, \boldsymbol{\theta})$ where $\mathcal{G} = (\mathbf{X}, \mathbf{A})$ is a directed acyclic graph, $\mathbf{X} = \mathbf{X_D} \cup \mathbf{X_C} \cup \mathring{\mathbf{X}}_\mathbf{C}$ and $\mathbf{A}$ is a set of arcs such that nodes $\mathring{X}_i \in \mathring{\mathbf{X}}_\mathbf{C}$ *have no children and exactly one parent equal to $X_i$*. This condition is the key to guarantee that inference in a ctdBN is as fast as that in a classical BN. Finally, $\boldsymbol{\theta} = \boldsymbol{\theta}_\mathbf{D} \cup \boldsymbol{\theta}_\mathbf{C}$, where $\boldsymbol{\theta}_\mathbf{D} = \{P(X_i|\mathbf{Pa}(X_i))\}_{i=1}^n$ is the set of the conditional probability tables of the discrete and discretized variables $X_i$ in $\mathcal{G}$ given their parents $\mathbf{Pa}(X_i)$ in $\mathcal{G}$, and $\boldsymbol{\theta}_\mathbf{C} = \{f(\mathring{X}_i|X_i)\}_{i=d+1}^n$ is the set of the conditional truncated densities of the continuous random variables of $\mathring{\mathbf{X}}_\mathbf{C}$. Note that $\boldsymbol{\theta}_\mathbf{C}$ needs a very limited amount of memory compared to $\boldsymbol{\theta}_\mathbf{D}$ since truncated densities are monodimensional (e.g., a truncated normal distribution $f(\mathring{X}_i|X_i)$ is specified by only $2|\Omega_{X_i}|$ parameters).

An example of ctdBN is given in Figure 1. The model contains 3 continuous variables, $\mathring{\mathbf{X}}_\mathbf{C} = \{\mathring{X}_1, \mathring{X}_3, \mathring{X}_5\}$ represented by dotted circles, which are discretized into $\mathbf{X_C} =$

[1] By abuse of notation, we use interchangeably $X_i \in \mathbf{X}$ to denote a node in the BN and its corresponding random variable.

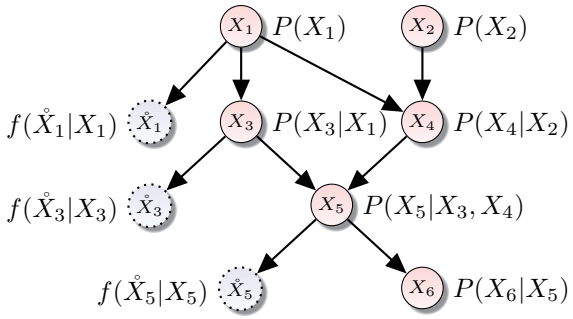


Figure 1: A BN with conditional truncated densities.

$\{X_1, X_3, X_5\}$. Nodes in solid circles $\mathbf{X_C}$ and $\mathbf{X_D}$ form a classical BN. Finally, all the continuous nodes $\mathring{X}_i \in \mathring{\mathbf{X}}_\mathbf{C}$ are children of their discretized counterpart $X_i$ and none has any child. The key idea of ctdBNs is thus to extend BNs by specifying the uncertainties over continuous random variables $\mathring{X}_i$ as 2-level functions: a "rough" probability distribution for discrete variable $X_i$ and a finer-grain conditional density $f(\mathring{X}_i|X_i)$ for $\mathring{X}_i$. This idea can be somewhat related to second order probabilities (Baron 1987).

**Proposition 1** *In a BN with conditional truncated densities defined over* $\mathbf{X} = \mathbf{X_D} \cup \mathbf{X_C} \cup \mathring{\mathbf{X}}_\mathbf{C}$*, where* $\mathbf{X_D} = \{X_1, \ldots, X_d\}$*,* $\mathbf{X_C} = \{\mathring{X}_{d+1}, \ldots, \mathring{X}_n\}$ *and* $\mathring{\mathbf{X}}_\mathbf{C} = \{\mathring{X}_{d+1}, \ldots, \mathring{X}_n\}$*, function* $h : \mathbf{X} \mapsto \mathbb{R}_0^+$ *defined as:*

$$h(\mathbf{X}) = \prod_{i=1}^{n} P(X_i|\mathbf{Pa}(X_i)) \prod_{i=d+1}^{n} f(\mathring{X}_i|X_i) \qquad (4)$$

*is a density function over* $\mathbf{X}$*.*

**Proof.** First, note that all the terms in the product are non-negative real-valued functions, hence $h$ is also a non-negative real-valued function. Let

$$\begin{aligned}\alpha = &\sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} \int_{\mathring{X}_{d+1}} \cdots \int_{\mathring{X}_n} \prod_{i=1}^{n} P(x_i|\mathbf{Pa}(x_i)) \\ &\prod_{i=d+1}^{n} f(\mathring{x}_i|x_i) d_{\mathring{x}_n} \cdots d_{\mathring{x}_{d+1}} \\ = &\sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} \prod_{i=1}^{n} P(x_i|\mathbf{Pa}(x_i)) \times \\ &\left(\int_{\mathring{X}_{d+1}} f(\mathring{x}_{d+1}|x_{d+1}) d_{\mathring{x}_{d+1}}\right) \cdots \left(\int_{\mathring{X}_n} f(\mathring{x}_n|x_n) d_{\mathring{x}_n}\right).\end{aligned}$$

By Property 2 of Definition 3, each integral of a conditional truncated density is equal to 1, hence:

$$\alpha = \sum_{x_1 \in X_1} \cdots \sum_{x_n \in X_n} \prod_{i=1}^{n} P(x_i|\mathbf{Pa}(x_i)).$$

This formula is also equal to 1 since its terms constitute a discrete BN. Therefore, $h(\mathbf{X})$ is a density function. ■

## Inference in ctdBNs

Let us now investigate how to perform inference in ctdBNs. The terms in Equation (4) satisfy Shafer-Shenoy's propagation axioms (Shenoy and Shafer 1990), so we can rely on a message-passing algorithm in a junction tree. The latter is constructed by node eliminations from the Markov network, as described previously. It was proved that first eliminating all simplicial nodes, i.e., nodes that, together with their neighbors in the Markov network, constitute a clique (a complete maximal subgraph), cannot prevent obtaining a junction tree that is optimal w.r.t. inference (van den Eijkhof and Bodlaender 2002). By the definition of ctdBNs, all the continuous nodes $\mathring{X}_i \in \mathring{\mathbf{X}}_\mathbf{C}$ constitute a clique with their parent $X_i$ (for instance, in Figure 1, $\{X_3, \mathring{X}_3\}$ is a complete maximal subgraph and is thus a clique). As a consequence, the junction tree of a ctdBN is simply the junction tree of its discrete BN part defined over $\mathbf{X_C} \cup \mathbf{X_D}$ to which cliques $\{X_i, \mathring{X}_i\}$, for $\mathring{X}_i \in \mathring{\mathbf{X}}_\mathbf{C}$, have been added (linked to a clique containing $X_i$). Figure 2 shows a junction tree related to the ctdBN of Figure 1. All the CPTs $P(X_i|\mathbf{Pa}(X_i))$, $i = 1, \ldots, n$, are inserted into cliques not containing any continuous node of $\mathring{\mathbf{X}}_\mathbf{C}$. Of course, conditional truncated densities are inserted into cliques $\{X_i, \mathring{X}_i\}$, $\mathring{X}_i \in \mathring{\mathbf{X}}_\mathbf{C}$.

The inference process can now be performed by message passing within the junction tree, for instance using a collect-distribute algorithm in a Shafer-Shenoy-like architecture, selecting a root clique $\mathcal{C}_R$ *not containing a continuous node of* $\mathring{\mathbf{X}}_\mathbf{C}$. Let $\mathcal{C}_i = \{X_k, \mathring{X}_k\}$ be a clique containing a continuous variable $\mathring{X}_k$ and let $X_k$ be the discretized variable corresponding to $\mathring{X}_k$. Assume that $\{t_1, \ldots, t_g\}$ are the cut-points of the discretization function applied to $\mathring{X}_k$. By construction, clique $\mathcal{C}_i$ has only one neighbor clique, say $\mathcal{C}_j$, and the separator between $\mathcal{C}_i$ and $\mathcal{C}_j$ is necessarily $S_{ij} = \{X_k\}$. During the collect phase, as clique $\mathcal{C}_i$ contains only conditional truncated density $f(\mathring{X}_k|X_k)$, the message from clique $\mathcal{C}_i$ to clique $\mathcal{C}_j$ is a vector of size $g + 1$ corresponding to:

$$\mathcal{M}_{\mathcal{C}_i \to \mathcal{C}_j}(x_k) = \int_{\Omega_{\mathring{X}_k}} f(\mathring{x}_k|x_k)\, d\mathring{x}_k = 1, \text{ for all } x_k \in \Omega_{X_k}.$$

If evidence $e_{\mathring{X}_k}$ of the type "$\mathring{X}_k$ belongs to interval $[a, b]$" is received, this can be entered into clique $\mathcal{C}_i$ as a function $f(e_{\mathring{X}_k}|\mathring{X}_k) : \Omega_{\mathring{X}_k} \mapsto [0, 1]$ equal to 1 when $\mathring{X}_k \in [a, b]$ and

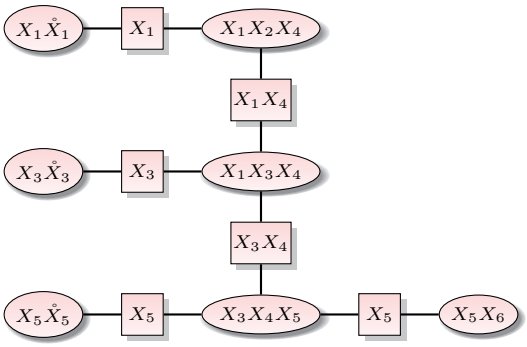


Figure 2: A junction tree for the ctdBN of Figure 1.

0 otherwise. More generally, beliefs about $\mathring{X}_k$ can be entered as any $[0,1]$-valued function $f(e_{\mathring{X}_k}|\mathring{X}_k)$. In this case, message $\mathcal{M}_{\mathcal{C}_i\to\mathcal{C}_j}(x_k)$ needs simply be updated as:

$$\mathcal{M}_{\mathcal{C}_i\to\mathcal{C}_j}(x_k) = \int_{\Omega_{\mathring{X}_k}} f(\mathring{x}_k|x_k) f(e_{\mathring{X}_k}|\mathring{x}_k)\, d\mathring{x}_k.$$

In all cases, note that message $\mathcal{M}_{\mathcal{C}_i\to\mathcal{C}_j}$ is computed by integrating a *monodimensional* function, which, in practice, is not time consuming (it can be done either exactly in closed-form formula or approximately using a MCMC algorithm). The resulting message $\mathcal{M}_{\mathcal{C}_i\to\mathcal{C}_j}$ is a real-valued vector of size $|\Omega_{X_k}| = g+1$. All the other messages sent during the collect phase are computed exactly as in a classical (discrete) junction tree. As a consequence, for the collect, the messages sent are precisely the same as those that would have been sent in the junction tree, had clique $\mathcal{C}_i$ not existed and evidence on $X_k$ been entered into clique $\mathcal{C}_j$ in the form of belief vector $\mathcal{M}_{\mathcal{C}_i\to\mathcal{C}_j}$. So, for the collect phase, except for the monodimensional integrations, the computational complexity of inference is the same as in the discrete case.

For the distribute phase, it is easy to see that all the messages are computed w.r.t. discrete random variables and are thus similar to messages computed in a classical (discrete) junction tree. Consequently, this property holds for the computations of all the messages in a Shafer-Shenoy-like architecture. To complete inference, there remains to compute the marginal *a posteriori* distributions of all the nodes. For the discrete ones, as usual, it is sufficient to select any clique containing the node and to multiply the CPTs stored in it with the messages it received, to marginalize out all the other variables and then to normalize the result. Here, there is no overhead compared to an inference w.r.t. only discrete nodes. For continuous node $\mathring{X}_k$ of clique $\{X_k, \mathring{X}_k\}$, the process is similar: message $\mathcal{M}_{\mathcal{C}_j\to\mathcal{C}_i}$ must be multiplied by the conditional truncated density stored in clique $\mathcal{C}_i$ and, then, $X_k$ must be marginalized out, which amounts to compute:

$$\sum_{x_k=0}^{g} \mathcal{M}_{\mathcal{C}_j\to\mathcal{C}_i}(x_k) f(\mathring{X}_k|X_k) f(e_{\mathring{X}_k}|\mathring{x}_k).$$

This corresponds to a mixture of conditional truncated densities since $\mathcal{M}_{\mathcal{C}_j\to\mathcal{C}_i}(x_k)$ is just a real number. Finally, the normalization amounts to integrate again a monodimensional function. Overall, except for the monodimensional functions integrations, inference in ctdBNs is thus of the same complexity as that in standard BNs.

## Experiments

For evaluating the expressive power of our model as well as the efficiency of the above inference algorithm, a set of hybrid Bayesian Networks (HBN) of different sizes are randomly generated (Moral, Rumí, and Salmerón 2002). In every HBN, half of the random variables are discrete; densities of the continuous variables are MTE potentials. The domain size of each discrete variable is randomly chosen between 2 and 4. The domain size $\Omega_{\mathring{X}_i}$ of each continuous random variable $\mathring{X}_i$ is randomly partitioned into 1, 2, or 3 subdomains $\Omega_k$ with probabilities of occurrence of 20%, 40% and 40% respectively. In each subdomain $\Omega_k$, the number of exponential function terms in the density functions is chosen randomly among 0 (uniform distribution), 1 and 2, with respective probabilities 5%, 75% and 20%. In addition, the number of parents of all the nodes follow a Poisson distribution with a mean varying w.r.t. the number of nodes in the HBN as described in Table 1. For each number of nodes, 500 different HBNs are generated. Finally, the DAG structures are constrained to contain only one connected component. To construct these HBNs, we use the ELVIRA library (`http://leo.ugr.es/elvira`). Then, from each HBN, a dataset is generated, from which a ctdBN is learnt using the learning algorithm described in (Mabrouk et al. 2015), constraining it to use the same set of arcs as the HBN. As a consequence, the ctdBN can be considered as an approximation of an exact multivariate density function specified by the HBN.

Table 1: Average number of parents per node.

| $\#nodes$ | $Mean(\#parents)$ |
|---|---|
| 4 | 2 |
| 8 | 1.83 |
| 16 | 1.66 |
| 32 | 1.5 |
| 64 | 1.33 |
| 128 | 1.16 |
| 256 | 1.1 |

In each HBN, we perform an exact inference using the ELVIRA library in order to compute the marginal probabilities of all the random variables in the HBN. Similarly, for the ctdBN, we execute the inference algorithm of the preceding section, using a non-parallel implementation in C++ and the aGrUM library (`http://agrum.lip6.fr`). All the experiments are performed on a Linux box with an Intel Xeon at 2.40GHz and 128GB of RAM. For comparing MTEs and ctdBNs, we use two criteria: i) the Jensen-Shannon Divergence (JSD) between the marginal distributions in the ctdBNs and those in the corresponding HBN; and ii) the times for computing these marginal distributions. The first criterion allows to assess the expressive power of ctdBNs whereas the second one allows to assess the efficiency of our inference algorithm.

Tables 2 and 3 report for each network size, specified by its number of nodes, the average over the 500 networks generated for this size of the JSD between the marginal distributions of the nodes obtained in the ctdBN model and those obtained in the MTE model. The tables display the average

Table 2: JSD for discrete variables.

| $\#nodes$ | $\mu_{JSD}$ | $\sigma_{JSD}$ | $min_{JSD}$ | $max_{JSD}$ |
|---|---|---|---|---|
| 4 | 0.0065 | 0.0040 | 0.0024 | 0.0105 |
| 8 | 0.0076 | 0.0078 | 0.0007 | 0.0201 |
| 16 | 0.0078 | 0.0099 | 0.0003 | 0.0303 |
| 32 | 0.0122 | 0.0137 | 0.0004 | 0.0510 |
| 64 | 0.0354 | 0.0269 | 0.0016 | 0.1111 |
| 128 | 0.0831 | 0.0527 | 0.0028 | 0.2326 |
| 256 | 0.1329 | 0.0805 | 0.0033 | 0.3752 |

Table 3: JSD for continuous variables.

| $\#nodes$ | $\mu_{JSD}$ | $\sigma_{JSD}$ | $min_{JSD}$ | $max_{JSD}$ |
|---|---|---|---|---|
| 4 | 0.0059 | 0.0017 | 0.0042 | 0.0077 |
| 8 | 0.0064 | 0.0024 | 0.0036 | 0.0099 |
| 16 | 0.0064 | 0.0026 | 0.0030 | 0.0114 |
| 32 | 0.0071 | 0.0032 | 0.0028 | 0.0146 |
| 64 | 0.0101 | 0.0044 | 0.0033 | 0.0215 |
| 128 | 0.0161 | 0.0072 | 0.0045 | 0.0349 |
| 256 | 0.0222 | 0.0098 | 0.0054 | 0.0476 |

Table 4: Inference computation times (in milliseconds).

| $\#nodes$ | $T_{MTE}$ | $T_{ctdBN}$ | $T_{MTE}/T_{ctdBN}$ |
|---|---|---|---|
| 4 | 40.92 | 0.27 | 151.56 |
| 8 | 279.16 | 0.84 | 332.33 |
| 16 | 7416.75 | 1.96 | 3784.06 |
| 32 | 42304.21 | 4.51 | 9380.09 |
| 64 | 88738.92 | 10.26 | 8649.02 |
| 128 | 94307.49 | 28.48 | 3311.36 |
| 256 | 122185.62 | 71.52 | 1708.41 |

of these JSDs over the nodes of the networks ($\mu_{JSD}$) but also their standard deviations $\sigma_{JSD}$ and their min ($min_{JSD}$) and max ($max_{JSD}$) values. As can be observed from these tables, the JSDs always remain small (remind that, for any distributions $P, Q$, $JSD(P\|Q) \in [0, 1]$). This shows that our model is expressive and faithful since its approximation of the true (MTE) densities is accurate. As shown in Table 4, which reports the inference execution times, this accuracy is not at the cost of inference performance: our algorithm significantly outperforms MTE inference and, the larger the network, the higher the discrepancy between the computation times of the two inference algorithms.

## Conclusion

In this paper, a new graphical model for handling uncertainty over sets of continuous and discrete variables is introduced. As shown by experiments, this model is both expressive and tractable for inference. For future works, we plan to enrich it by allowing the conditional truncated densities to depend not only on the discretized nodes but also on their parents. This shall increase the expressive power of the model. In addition, keeping the conditional truncated densities of the same form as the CPTs of the discretized nodes shall ensure tractability of inference. Of course, new algorithms will be needed for learning this model from data.

**Acknowledgments.** This work was supported by European project H2020-ICT-2014-1 #644425 Scissor.

## References

Bacchus, F.; Dalmao, S.; and Pitassi, T. 2003. Algorithms and complexity results for #sat and Bayesian inference. In *Proceedings of FOCS'03*, 340–351.

Baron, J. 1987. Second-order probabilities and belief functions. *Theory and Decision* 23(1):25–36.

Chavira, M., and Darwiche, A. 2008. On probabilistic inference by weighted model counting. *Artificial Intelligence* 172(6-7):772–799.

Cobb, B., and Shenoy, P. 2006. Inference in hybrid Bayesian networks with mixtures of truncated exponentials. *International Journal of Approximate Reasoning* 41(3):257–286.

Cobb, B.; Shenoy, P.; and Rumí, R. 2006. Approximating probability density functions in hybrid Bayesian networks with mixtures of truncated exponentials. *Statistics and Computing* 16(3):293–308.

Dechter, R. 1999. Bucket elimination: A unifying framework for reasoning. *Artificial Intelligence* 113:41–85.

Langseth, H.; Nielsen, T.; Rumí, R.; and Salmerón, A. 2012a. Inference in hybrid Bayesian networks with mixtures of truncated basis functions. In *Proceedings of PGM'12*, 171–178.

Langseth, H.; Nielsen, T.; Rumí, R.; and Salmerón, A. 2012b. Mixtures of truncated basis functions. *International Journal of Approximate Reasoning* 53(2):212–227.

Lauritzen, S., and Wermuth, N. 1989. Graphical models for associations between variables, some of which are qualitative and some quantitative. *Annals of Statistics* 17(1):31–57.

Lauritzen, S. 1992. Propagation of probabilities, means and variances in mixed graphical association models. *Journal of the American Statistical Association* 87:1098–1108.

Lerner, U.; Segal, E.; and Koller, D. 2001. Exact inference in networks with discrete children of continuous parents. In *Proceedings of UAI'01*, 319–328.

Mabrouk, A.; Gonzales, C.; Jabet-Chevalier, K.; and Chojnaki, E. 2015. Multivariate cluster-based discretization for Bayesian network structure learning. In *Proceedings of SUM'15*.

Madsen, A., and Jensen, F. 1999. LAZY propagation: A junction tree inference algorithm based on lazy inference. *Artificial Intelligence* 113(1–2):203–245.

Moral, S.; Rumí, R.; and Salmerón, A. 2001. Mixtures of truncated exponentials in hybrid Bayesian networks. In *Proceedings of ECSQARU'01*, volume 2143 of *LNAI*, 156–167.

Moral, S.; Rumí, R.; and Salmerón, A. 2002. Estimating mixtures of truncated exponentials from data. In *Proceedings of PGM'02*, 135–143.

Pearl, J. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kauffmann.

Romero, R.; Rumí, R.; and Salmerón, A. 2004. Structural learning of Bayesian networks with mixtures of truncated exponentials. In *Proceedings of PGM'04*, 177–184.

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

Shafer, G. 1996. *Probabilistic expert systems*. Society for Industrial and Applied Mathematics.

Shenoy, P., and Shafer, G. 1990. Axioms for probability and belief function propagation. In *Proceedings of UAI'90*, 169–198.

Shenoy, P., and West, J. 2011. Inference in hybrid Bayesian networks using mixtures of polynomials. *International Journal of Approximate Reasoning* 52(5):641–657.

Shenoy, P. 2011. A re-definition of mixtures of polynomials for inference in hybrid Bayesian networks. In *Proceedings of ECSQARU'11*, volume 6717 of *LNCS*, 98–109.

Shenoy, P. 2012. Two issues in using mixtures of polynomials for inference in hybrid Bayesian networks. *International Journal of Approximate Reasoning* 53(5):847–866.

van den Eijkhof, F., and Bodlaender, H. L. 2002. Safe reduction rules for weighted treewidth. In *Proceedings of WG'02*, volume 2573 of *LNCS*, 176–185.