Quadrature Rules in General Continuous Bayesian Networks : Discrete Inference without Discretization

Marvin Lasserre, Régis Lebrun, Pierre-Henri Wuillemin

Quadrature Rules in General Continuous Bayesian Networks : Discrete Inference without Discretization

Marvin Lasserre, Régis Lebrun, Pierre-Henri Wuillemin

Quadrature Rules in General Continuous Bayesian Networks : Discrete Inference without Discretization

Marvin Lasserre[1] **Régis Lebrun**[2] **Pierre-Henri Wuillemin**[3]

[1]Inria Centre, Bordeaux University, Bordeaux, France
[2]Airbus CRT, France
[3]LIP6, Sorbonne Université, 4 place de Jussieu, 75005 Paris, France,

Abstract

Probabilistic inference in high-dimensional continuous or hybrid domains poses significant challenges, commonly addressed through discretization, sampling, or reliance on parametric assumptions. The drawbacks of these methods are well-known: inaccuracy, slow computational speeds or overly constrained models.
This paper introduces a novel general inference algorithm designed for Bayesian networks featuring both discrete and continuous variables. The algorithm avoids the discretization of continuous densities into histograms by employing quadrature rules to compute continuous integrals and avoids the use of a parametric model by using orthogonal polynomials to represent the posterior density. Additionally, it preserves the computational efficiency of classical sum-product algorithms by using an auxiliary discrete Bayesian networks appropriately constructed to make continuous inference.
Numerous experiments are conducted using either the conditional linear Gaussian model as a benchmark, or non-Gaussian models for greater generality. Our algorithm demonstrates significant improvements both in speed and accuracy when compared with existing methods.

1 INTRODUCTION AND RELATED WORKS

Bayesian networks (BNs) [Pearl, 1988] are a class of probabilistic models useful for dealing with complex systems under uncertainty. Specifically, they exploit conditional independence between random variables to compactly encode their joint distribution as a product of lower-dimensional conditional probability distributions (CPDs). In addition, these conditional independence are represented through a Directed Acyclic Graph (DAG) giving an interpretability to the model which is valuable to assist domain experts.

Probabilistic inference, which consist of obtaining the marginal distribution of a set of unobserved variables given a set of observed variables, constitute one of the principal uses of BNs. Indeed, this task is at the very basis of many applications such as classification or prediction [Friedman et al., 1997]. While it is a NP-hard problem [Cooper, 1990a], classic algorithms [Koller and Friedman, 2009] take advantage of the factorisation of the joint distribution to make these calculations tractable. They are roughly divided into two categories: approximated inference algorithms that estimate the posterior distribution, for instance using sampling methods, and exact inference methods that compute exactly the posterior distribution. These algorithms perform effectively when all variables are discrete, but encounter difficulties when confronted to continuous variables. This is because the model used to represent the CPDs must be closed under product, marginalization and restriction [Shenoy and Shafer, 2008]. Discrete CPDs can be described using conditional probability tables (CPTs), which are closed under these operations, but a general model for continuous CPDs does not exist.

This paper focuses on exact inference methods for BNs containing both discrete and continuous variables, commonly refered as *hybrid* BN. While numerous attempts have been made to address this problem, they all comes with some sort of shortcomings.

Although discretization of continuous variables may appear as a simple solution, it comes with several drawbacks. Since the complexity of discrete inference algorithms is exponential with respect to the largest domain size of the variables, this results into a trade-off between the loss of accuracy and the increase in the computational cost. Furthermore, for a given number

of discretization points, their position is determined by minimizing a distance which can be computationally costly [Kozlov and Koller, 1997]. In general, numerous heuristics provide a diverse range of solutions, especially regarding the number of discretization points. This implies a certain level of arbitrariness in the selection of discretization methods, despite their substantial impact on the final outcome.

Mixtures of truncated basis functions (MoTBFs) [Langseth et al., 2012] are a set of models that approximate a continuous CPD using a piecewise function. Each piece corresponds to a function that is closed under the inference operations. This method generalizes discretization which is equivalent to approximating continuous CPDs with piecewise constant functions (histograms). However, despite being more expressive than histograms, the conversion of a CPD into a MoTBF can be difficult and the number of pieces can grow exponentially when multiplying CPDs [Shenoy, 2012].

Another common solution is to use a continuous model satisfying the closure properties such that inference algorithms can be applied with minor modifications. The most well-known example of this type is the conditional linear Gaussian (CLG) model [Lauritzen and Wermuth, 1989] but it has strong limitations, such as assuming normal distribution and linear relationships between variables. For a comprehensive overview of the methods discussed, see Salmeron et al. [2018].

This paper introduces a novel inference algorithm designed for hybrid Bayesian networks. Marginalization of continuous variables employs Gaussian quadrature methods, while the representation of the continuous posterior density utilizes orthogonal polynomials. This approach allows to avoid the discretization of the continuous CPDs or the use of a parametric assumptions. Additionally, inference is efficiently organized through an auxiliary discrete Bayesian network and classical sum-product algorithms. Consequently, this algorithm combines the efficiency of discrete inference methods with the expressive power of hybrid representation.

The remainder of the paper is organized as follows: Section 2 reviews classic algorithms to make inference in discrete Bayesian networks. Section 3 gives the basic notions about orthogonal polynomials and numerical integration using quadrature rules. Section 4 describes the construction of the discrete BN associated with the inference in the hybrid BN and how to retrieve the continuous model from the result of the discrete inference. Section 5 contains a simple example on generated Gaussian data. Finally, the last section illustrates the performance of our algorithm on classical Bayesian network structures, which we have equipped with continuous distributions.

2 INFERENCE IN BAYESIAN NETWORKS

Consider a random vector $\mathbf{X} = (X_1, \ldots, X_n)$ whose components X_i take values x_i from domains Ω_i. A BN structure G is a DAG whose nodes $\mathbf{X} = \{X_1, \ldots, X_n\}$ represent random variables and arcs represent direct dependencies between those variables. Let $\mathbf{Pa}_i$ denote the parents of X_i in G, the joint probability distribution $f_{\mathbf{X}}$ of the random vector $\mathbf{X}$ is said to factorize according to G if it can be written as

$$f_{\mathbf{X}}(x_1, \ldots, x_n) = \prod_{i=1}^{n} f_{X_i|\mathbf{Pa}_i}(x_i|\mathbf{pa}_i)$$

where $\mathbf{pa}_i$ denotes a realization of $\mathbf{Pa}_i$[1]. A BN is defined as a couple $(f_{\mathbf{X}}, G)$ where $f_{\mathbf{X}}$ factorizes over G and where each node X_i is associated with the CPD $f_{X_i|\mathbf{Pa}_i}$. For a visual representation, see figure (**??**).

Given a subvector of observed variables $\mathbf{X}_E \subset \mathbf{X}$, taking a known value $\boldsymbol{e} \in \Omega_E$, and a subvector of target variables $\mathbf{X}_T \subset \mathbf{X} \setminus \mathbf{X}_E$, a probabilistic inference consists of computing the posterior density $f_{\mathbf{X}_T|\mathbf{X}_E=\boldsymbol{e}}$ by marginalizing the variables $\mathbf{X}_M = \mathbf{X} \setminus \{\mathbf{X}_T \cup \mathbf{X}_E\}$. In the discrete case, where each domain Ω_i is countable, the posterior density is obtained by the formula:

$$f(\mathbf{x}_T|\boldsymbol{e}) = \frac{f(\mathbf{x}_T, \boldsymbol{e})}{f(\boldsymbol{e})} = \frac{\sum\limits_{\mathbf{x}_M \in \Omega_M} f(\mathbf{x}_T, \mathbf{x}_M, \boldsymbol{e})}{\sum\limits_{\mathbf{x}_T \in \Omega_T} f(\mathbf{x}_T, \boldsymbol{e})}. \quad (1)$$

Obtaining $f_{\mathbf{X}_i|\mathbf{X}_E=\mathbf{e}}$ is then equivalent to compute $f_{\mathbf{X}_i,\mathbf{X}_E=\mathbf{e}}$ and to normalize it afterwards. This task, called the inference, is difficult : exact and even approximate inference are NP-hard (Cooper [1990b], Dagum and Luby [1997]). However, in the discrete case, effective heuristics are used to perform these calculations with good efficiency (for instance Madsen and Jensen [1999]).

In the hybrid case, which is the focus of this paper, components of a random vector can be either discrete or continuous. To discriminate between discrete and continuous random vectors, they will be denoted respectively by $\mathbf{X}^d$ and $\mathbf{X}^c$, with dimensions n^d and n^c, and domains $\Omega^d_{\mathbf{X}}$ and $\Omega^c_{\mathbf{X}}$. If no exponent is specified, the hybrid case is implied. Equation (1) becomes:

$$f(\mathbf{x}_T|\boldsymbol{e}) = \frac{\sum\limits_{\mathbf{x}^d_M \in \Omega^d_M} \int_{\Omega^c_M} f(\mathbf{x}_T, \mathbf{x}^d_M, \mathbf{x}^c_M, \boldsymbol{e}) \mathrm{d}\mathbf{x}^c_M}{\sum\limits_{\mathbf{x}^d_T \in \Omega^d_I} \int_{\Omega^c_I} f(\mathbf{x}^d_T, \mathbf{x}^c_T, \boldsymbol{e}) \mathrm{d}\mathbf{x}^c_T} \quad (2)$$

[1] In the remainder, when it is clear from the arguments, the index of conditional densities will be dropped in order to alleviate the notations.

Note that no parametric assumptions are made for the distributions of the continuous variables. Several methods to make inferences in this context and their shortcomings were presented in the previous section. In the next, we propose to convert these integrals into sums using quadrature rules and to use discrete inference algorithms to efficiently compute them.

3 INTEGRATION USING QUADRATURE RULES

This section introduces the basic notions of integration using Gauss quadrature rules. It is mainly inspired from Gautschi [2011] and the interested reader can refer to this monograph for further details.

3.1 ORTHOGONAL POLYNOMIALS

Let $\mathbb{R}[X]$ denote the vector space of real polynomials in one variable and $\mathbb{R}_d[X]$ the subspace of polynomials of degree at most d. Let μ be a positive measure whose moments are all finite, the vector space $\mathbb{R}[X]$ is equipped with the following inner product:

$$\langle P, Q\rangle = \int_{\mathbb{R}} P(x)Q(x)\mathrm{d}\mu, \quad (P,Q) \in \mathbb{R}[X]\times\mathbb{R}[X] \quad (3)$$

and the resulting norm: $||P|| = \left(\int_{\mathbb{R}} P^2(x)\mathrm{d}\mu\right)^{1/2}$.

In this context, two polynomials P and Q of $\mathbb{R}[X]$ are said to be orthogonal if $\langle P, Q\rangle = 0$. If, in addition of being orthogonal, $||P|| = ||Q|| = 1$, then they are said to be orthonormal. Given that the inner product is positive definite on $\mathbb{R}[X]$, that is $||P|| > 0, \forall P \in \mathbb{R}[X]$ such that $P \not\equiv 0$, then there exists a unique family $\{P_i\}_{i\in\mathbb{N}}$, where i is the degree, of orthonormal polynomials (with a positive leading coefficient) associated with μ. For instance, the family associated with the uniform measure on $[-1, 1]$ and that will be used later, is called the Legendre polynomials. This family of orthonormal polynomials form a basis of $\mathbb{R}[X]$ and are therefore well suited for function approximation (Fig. **??**). In particular, we will now see that they play a crucial role in Gauss quadrature rules.

3.2 GAUSS QUADRATURE RULES

Let $f \in \mathcal{F}(\mathbb{R}, \mathbb{R})$, the set of integrable real function with respect to the measure μ. Given a set of p mutually distinct points $x[i]$ in the support of μ and a set of p real values $\{\omega_1, \ldots, \omega_p\}$, a p-point quadrature rule is a linear map L_p such that:

$$\forall f \in \mathcal{F}(\mathbb{R}, \mathbb{R}), L_p(f) = \sum_{i=1}^{p} \omega_i f(x[i]), \quad (4)$$

Then, $L_p(f)$ gives an approximation of the integral of f and $R_p(f)$ is the associated error :

$$R_p(f) = \int_{\mathbb{R}} f(x)\mathrm{d}\mu - L_p(f)$$

The points $x[i]$ are called the nodes and the values ω_i the weights of the quadrature. If for a given $d \in \mathbb{N}$, $\forall P \in \mathbb{R}_d[X], R_p(P) = 0$, the quadrature is said to have a degree of exactness d.

For a fixed number of nodes p, the maximum degree of exactness of such a quadrature rule is $d = 2p - 1$, and is obtained through the so-called Gaussian quadrature rules, where the points $x[i]$ are the zeros of P_p, the orthonormal polynomial of degree p associated to the measure μ. As for the weights, they are given by formulas depending on the orthogonality condition (Eq. 3). In the case of the uniform measure on $[-1, 1]$ and for a p-points quadrature rule, the nodes are the zeros of the p-th Legendre polynomial P_p and the weights are given by [Abramowitz et al., 1965]:

$$\omega_i = \frac{2p+1}{(1-x[i]^2)\left[P_p'(x[i])\right]^2} > 0.$$

4 INFERENCES USING GAUSS QUADRATURES

First, let us introduce some notations relative to discrete sets that will be used in this section. Discrete sets $\{a, a+1, \ldots, b-1, b\}$ where $a, b \in \mathbb{N}$ such that $a < b$ are denoted by $[\![a, b]\!]$. In the case where the bounds are vectors $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{N}^n$ such that $a_i < b_i, \forall i$, the notation $[\![\boldsymbol{a}, \boldsymbol{b}]\!]$ serves as a shorthand for the Cartesian product $\bigtimes_{i=1}^{n} [\![a_i, b_i]\!]$. Finally, when $a = 1$, the lower bound is implied and the notation is shortened as $[\![b]\!]$.

The numerical integration method discussed in the previous section can be easily extended to handle multivariate integrals by employing a quadrature rule for each dimension. This technique can then be applied for computing the integrals in equation (2) that becomes:

$$f(\mathbf{x}_T|\boldsymbol{e}) = \frac{\sum_{\mathbf{x}_M^d \in \Omega_M^d} \sum_{\boldsymbol{i}\in[\![\boldsymbol{p}_M]\!]} \omega_{\boldsymbol{i}} f(\mathbf{x}_T, \mathbf{x}_M^d, \mathbf{x}_M^c[\boldsymbol{i}], \boldsymbol{e})}{\sum_{\mathbf{x}_T^d \in \Omega_T^d} \sum_{\boldsymbol{i}\in[\![\boldsymbol{p}_T]\!]} \omega_{\boldsymbol{i}} f(\mathbf{x}_T^d.\mathbf{x}_T^c[\boldsymbol{i}], \boldsymbol{e})} \quad (5)$$

With the subscript "$*$" as a wildcard for T or M, $\boldsymbol{p}_* = (p_{*,1}, \ldots, p_{*,n_*^c})$ is the number of quadrature nodes (denoted by $\{x_*^c[\boldsymbol{i}]\}_{\boldsymbol{i}\in[\![\boldsymbol{p}_*]\!]}$) chosen for each component of $\mathbf{X}_*^c$ and $\omega_{\boldsymbol{i}} = \prod_{j\in[\![n_*^c]\!]} \omega_{i_j}$. Denoting $\mathbf{X}_*^\delta$ the discretized random vector obtained from $\mathbf{X}_*^c$ that takes values $\mathbf{x}_*^\delta$ in $\Omega_*^\delta = \{x_*^c[\boldsymbol{i}]\}$ with probability $f(\mathbf{x}_*^\delta) \propto \omega_{\boldsymbol{i}} f_{\mathbf{X}_*^c}(\mathbf{x}_*^c[\boldsymbol{i}])$, this last equation can be identified with the formula (1). Note that $\omega_{\boldsymbol{i}} f_{\mathbf{X}_*^c}$ is just an

approximation of a discrete probability distribution, since it does not sum to 1. However, the sum tends to 1 as the number of quadrature nodes increases.

Then, using Gauss quadratures to marginalize continuous variables leads to a formula that is equivalent to a discrete inference. Moreover, this inference is *exact* if the conditional densities $f_{X_i^c|\mathbf{Pa}_i}$ are polynomials of degree at most $d_i = 2p_i - 1$ of the variable x_i^c. By constructing a discrete BN judiciously from the hybrid one, the different operations can be done efficiently using the classic inference algorithms for BN that we have discussed in section 2.

4.1 CONSTRUCTION OF THE DISCRETE BN

The structure G^δ of the discrete BN remains the same as the structure G^h of the hybrid one. The transformation only lies in the continuous component of the conditional density. Given a topological order over this structure, let X_i^c be a continuous random variable in the hybrid BN, then this variable is replaced by the discrete variable X_i^δ in the discrete BN, and it has for CPD the table T_i containing:

$$\begin{aligned} T_i(\mathbf{x}_i^\delta = j|\mathbf{Pa}^d, \mathbf{Pa}^\delta = \boldsymbol{k}) =& (\omega_i)_j f(x_i^c[j]|\mathbf{Pa}_i^d, \mathbf{Pa}_i^c[\boldsymbol{k}]) \\ &\propto f(\mathbf{x}_i^\delta|\mathbf{Pa}_i^d, \mathbf{Pa}_i^\delta) \end{aligned}$$

where $\mathbf{Pa}_i^c[\boldsymbol{k}]$ are the nodes of the quadratures for the parents of i, determined by the multi-index $\boldsymbol{k}$ (following the definition of $x[i]$ in section 3.2).

Now, let X_i^d be a discrete random variable in the hybrid BN, then in the discrete BN, X_i^d has for CPD the table T_i containing:

$$T_i(\mathbf{x}_i^d|\mathbf{Pa}^d, \mathbf{Pa}^\delta = \boldsymbol{k}) = f(x_i^d|\mathbf{Pa}_i^d, \mathbf{Pa}_i^c[\boldsymbol{k}])$$

The parents of X_i are necessarily a set of discrete variables, since variables are replaced following a topological order.

Since the only difference between the hybrid BN and its discretized version comes from how the continuous variables are handled, in the remainder of this section and without a loss of generality, the case where $\mathbf{X}^d = \emptyset$ is considered in order to focus on the principal idea of our algorithm. In order to alleviate notations and since all variables will be continuous, the "c" exponent is dropped.

4.2 RECONSTRUCTION OF THE CONTINUOUS POSTERIOR

The previous construction allowed us to obtain a discrete BN that we can now use to perform inference. However, these inferences only give us access to a discrete table containing $f_{\mathbf{X}_T^\delta|\mathbf{X}_E^\delta}$, that is, an estimation of the conditional density evaluated at the Gauss nodes. Hence, a last step is needed to reconstruct a continuous function.

First, let us consider the case where $\mathbf{X}_E = \emptyset$ and we want to reconstruct the continuous joint density $f_{\mathbf{X}_T}$. In order to recover a continuous function from $f_{\mathbf{X}_T^\delta|\mathbf{X}_E^\delta}$, the assumption is made that the posterior density is a linear combination of multivariate orthogonal polynomials used for the Gauss quadrature rules. These multivariate orthogonal polynomials are constructed as the tensor product of univariate orthogonal polynomial :

$$f(\mathbf{x}_T) = \sum_{\boldsymbol{i} \in [\![\boldsymbol{p}_T]\!]} \alpha_{\boldsymbol{i}} P_{\boldsymbol{i}}(\mathbf{x}_T)$$

where $\boldsymbol{i}$ is a multi-index of dimension d_T and where $P_{\boldsymbol{i}} = \bigotimes_{k=1}^{d_T} P_{i_k}$. With this assumption, the value of coefficients $\alpha_{\boldsymbol{i}}$ can be easily estimated using the orthonormality property:

$$\alpha_{\boldsymbol{i}} = \langle f_{\mathbf{X}_T}, P_{\boldsymbol{i}} \rangle = \int_{\mathbb{R}^{d_T}} f(\mathbf{x}_T) P_{\boldsymbol{i}}(\mathbf{x}_T) \mathrm{d}\boldsymbol{\mu}$$

This multiple integral, once again, can be approximated using Gauss quadratures for each dimension:

$$\alpha_{\boldsymbol{i}} = \sum_{\boldsymbol{j} \in [\![\boldsymbol{p}_T]\!]} \omega_{\boldsymbol{j}} f(\mathbf{x}[\boldsymbol{j}]) P_{\boldsymbol{i}}(\mathbf{x}_T[\boldsymbol{j}]) \tag{6}$$

By construction of the discrete BN, the posterior discrete distribution for $\mathbf{X}_T^\delta$ is the tensor:

$$\phi_{\boldsymbol{j}} = \omega_{\boldsymbol{j}} f(\mathbf{x}_T[\boldsymbol{j}])$$

Consequently, the coefficients of the linear combination can be computed by the contraction of the tensor $\phi_{\boldsymbol{j}}$, and the tensor $\pi_{\boldsymbol{i},\boldsymbol{j}} = P_{\boldsymbol{i}}(\mathbf{x}_T[\boldsymbol{j}])$ whose elements corresponds to the evaluation of the multivariate orthogonal polynomials at the Gauss nodes:

$$\alpha_{\boldsymbol{i}} = \sum_{\boldsymbol{j} \in [\![\boldsymbol{p}_T]\!]} \pi_{\boldsymbol{i},\boldsymbol{j}} \phi_{\boldsymbol{j}} \tag{7}$$

In the case where $\mathbf{X}_E \neq \emptyset$, the posterior $f_{\mathbf{X}_T|\mathbf{X}_E}$ can be computed by making the ratio between $f_{\mathbf{X}_T,\mathbf{X}_E}$ and $f_{\mathbf{X}_E}$, that have been approximated by a linear combination whose coefficients are give by the previous equation:

$$\begin{aligned} f_{\mathbf{X}_T|\mathbf{X}_E}(\mathbf{x}_T|\mathbf{x}_E) &= \frac{\sum_{\boldsymbol{i} \in [\![\boldsymbol{p}_T]\!]} \sum_{\boldsymbol{j} \in [\![\boldsymbol{p}_E]\!]} \alpha_{\boldsymbol{i},\boldsymbol{j}} \left(P_{\boldsymbol{i}} \otimes P_{\boldsymbol{j}}\right)(\mathbf{x}_T, \mathbf{x}_E)}{\sum_{\boldsymbol{k} \in [\![\boldsymbol{p}_E]\!]} \alpha_{\boldsymbol{k}} P_{\boldsymbol{k}}(\mathbf{x}_E)} \\ &= \sum_{\boldsymbol{i} \in [\![\boldsymbol{p}_T]\!]} \left(\frac{\sum_{\boldsymbol{j} \in [\![\boldsymbol{p}_E]\!]} \alpha_{\boldsymbol{i},\boldsymbol{j}} P_{\boldsymbol{j}}(\mathbf{x}_E)}{\sum_{\boldsymbol{k} \in [\![\boldsymbol{p}_E]\!]} \alpha_{\boldsymbol{k}} P_{\boldsymbol{k}}(\mathbf{x}_E)} \right) P_{\boldsymbol{i}}(\mathbf{x}_T) \\ &= \sum_{\boldsymbol{i} \in [\![\boldsymbol{p}_T]\!]} \gamma_{\boldsymbol{i}}(\mathbf{x}_E) P_{\boldsymbol{i}}(\mathbf{x}_T) \end{aligned}$$

Hence, the posterior can also be written as a linear combination of multivariate orthogonal polynomials of $\mathbf{x}_T$ and whose coefficients are themselves ratio of multivariate orthogonal polynomials of $\mathbf{x}_E$. By allowing a further approximation and at the expense of $f_{X_T|\mathbf{X}_E}$ not being exactly the ratio of f_{X_T,X_E} and f_{X_E}, the reconstruction of the continuous posterior can be done using the result of an inference in the discrete BN. To do so, the coefficients $\gamma_{\boldsymbol{i}}(\mathbf{x}_E)$ has to be in a polynomial form and are then approximated by linear combinations of multivariate orthogonal polynomials of $\mathbf{x}_E$:

$$\begin{aligned} f_{\mathbf{X}_T|\mathbf{X}_E}(\mathbf{x}_T|\mathbf{x}_E) &= \sum_{\boldsymbol{i}\in[\![\boldsymbol{p}_T]\!]} \gamma_{\boldsymbol{i}}(\mathbf{x}_E) P_{\boldsymbol{i}}(\mathbf{x}_T) \\ &= \sum_{\boldsymbol{i}\in[\![\boldsymbol{p}_T]\!]} \left(\sum_{\boldsymbol{j}\in[\![\boldsymbol{p}_E]\!]} \gamma_{\boldsymbol{i},\boldsymbol{j}} P_{\boldsymbol{i}}(\mathbf{x}_T) \right) P_{\boldsymbol{j}}(\mathbf{x}_E) \\ &= \sum_{\boldsymbol{i}\in[\![\boldsymbol{p}_T]\!]} \sum_{\boldsymbol{j}\in[\![\boldsymbol{p}_E]\!]} \gamma_{\boldsymbol{i},\boldsymbol{j}} P_{\boldsymbol{i},\boldsymbol{j}}(\mathbf{x}_T, \mathbf{x}_E) \end{aligned}$$

where $P_{\boldsymbol{i},\boldsymbol{j}} = P_{\boldsymbol{i}} \otimes P_{\boldsymbol{j}}$. The coefficients $\gamma_{\boldsymbol{i},\boldsymbol{j}}$ can then be computed using a contraction between the potential $\phi_{\boldsymbol{k},\boldsymbol{l}} = f_{\mathbf{X}_T^\delta|\mathbf{X}_E^\delta}(\mathbf{x}_T^\delta|\mathbf{x}_E^\delta) = \omega_{\boldsymbol{i}} f(\mathbf{x}_T[\boldsymbol{k}]|\mathbf{x}_E[\boldsymbol{l}])$ obtained by inference in the discrete BN and the tensor $\pi_{\boldsymbol{i},\boldsymbol{j},\boldsymbol{k},\boldsymbol{l}} = P_{\boldsymbol{i},\boldsymbol{j}}(\mathbf{x}_T[\boldsymbol{k}], \mathbf{x}_E[\boldsymbol{l}])$:

$$\gamma_{\boldsymbol{i},\boldsymbol{j}} = \sum_{\boldsymbol{k}\in[\![\boldsymbol{p}_T]\!]} \sum_{\boldsymbol{l}\in[\![\boldsymbol{p}_T]\!]} \pi_{\boldsymbol{i},\boldsymbol{j},\boldsymbol{k},\boldsymbol{l}} \phi_{\boldsymbol{k},\boldsymbol{l}}$$

4.3 INFERENCE OF A UNIVARIATE MARGINAL USING GAUSS-LEGENDRE QUADRATURE

The numerical experiments made in the next section in order to test our algorithms are relying on the estimation of the univariate marginal of a variable X_i. This corresponds to the case where $\mathbf{X}_M^c = \{X_i\}$ and $\mathbf{X}_E^c = \emptyset$ and equation (7) reduces to:

$$\alpha_j = \sum_{j=1}^{p_i} \pi_{j,k}\phi_k = \sum_{j=1}^{p_i} P_j(x_i[k])\phi_k$$

which can be rewritten under matrix notation:

$$\boldsymbol{\alpha} = \Pi\boldsymbol{\phi}$$

where $\boldsymbol{\alpha}$ is the column vector whose components are the coefficients α_j, $\boldsymbol{\phi}$ is the potential obtained by inference in the discrete BN and Π is the matrix:

$$\Pi = \begin{pmatrix} P_1(\xi_i[1]) & \dots & P_1(\xi_i[p]) \\ \vdots & \ddots & \vdots \\ P_n(\xi_i[1]) & \dots & P_n(\xi_i[p]) \end{pmatrix}$$

Since we want to estimate the univariate marginal of X_i for any kind of probabilistic model for the conditional densities $f_{\mathbf{X}_j|\mathbf{Pa}_j}$, we take the family of orthogonal polynomials whose weight function is $\omega(x) = \frac{1}{2}$ on $[-1,1]$ and called the Legendre polynomials. Since the scalar product is defined on a bounded domain, we need to truncate conditional densities whose support is $\mathbb{R}$. To do so, we introduce a parameter ϵ that will control the probability mass that is ruled out by the truncation. The domain of a random variable X_i is the quantile at probability level ϵ. While this is possible for nodes without parents, this isn't the case for nodes with parents since their univariate marginal is unknown. To circumvent this problem, the quantile at probability level ϵ of $f_{X_i|\mathbf{Pa}[j]}$ is computed for each value of the discrete parent. The estimated lower and upper bound of the domain are then respectively given by the minimum and the maximum of the obtained bounds. Note that to estimate the domain of variable, we need that its parents are discretized and this task is done simultaneously with the discretization. Once the domain $[a_i, b_i]$ of a variable is known, the linear transformation:

$$\tilde{X}_i = \frac{2}{b_i - a_i} X - \frac{a_i + b_i}{2}$$

is applied and the new variable has domain $\Omega_i = [-1,1]$ and conditional density

$$f_{\tilde{X}_i|\widetilde{\mathbf{Pa}_i}}(\tilde{x}_i|\widetilde{\mathbf{pa}_i}) = \frac{b_i - a_i}{2} f_{X_i|\widetilde{\mathbf{Pa}_i}}(\frac{b_i - a_i}{2}\tilde{x}_i + \frac{a_i + b_i}{2}|\widetilde{\mathbf{Pa}_i})$$

The method described in the previous subsection is then applied to the discrete BN with renormalized random variables to obtain $f_{\tilde{X}_i}$. Finally, applying the inverse linear transformation to $\tilde{X}_i$, we obtain the marginal density defined on $[a_i, b_i]$:

$$f_{X_i}(x_i) = \frac{2}{b_i - a_i} f_{\tilde{X}_i}\left(\frac{2}{b_i - a_i} x_i - \frac{b_i + a_i}{b_i - a_i}\right)$$

Note that in addition of the parameter ϵ, our algorithm depends on the parameters p_i which are the number of Gauss nodes p_i taken for each variable X_i. They are all set to a same value p and consequently, the algorithm depends on the two parameters ϵ and p.

The method used to estimate the domains of the variable can overestimate the size of the true domain at level of probability ϵ. A solution to this downside is to initialize the domains using this rough estimation, discretize the nodes and to make inferences to estimate the univariate marginal of each node. Doing so, we can use the marginals to compute the domain corresponding to the quantile of level ϵ for each variable. By discretizing again the variables with these new domains we can improve the estimation and repeating the process we can improve the estimation. This add a new parameter r to our algorithm corresponding to the number of iteration done to estimate the domains.

Note however that in a learning setting, the domain can be estimated directly from the dataset and this problem doesn't appear.

5 NUMERICAL EXPERIMENTS

This section investigates the influence of hyperparameters on the performances of our new inference algorithm while also conducting a comparative analysis against classical methods. All the scripts necessary to reproduce the numerical experiments detailed here are available in the GitHub repository (anonymized). These scripts utilize the pyAgrum library [Ducamp et al., 2020] for probabilistic graphical models and the OpenTURNS library [Baudin et al., 2016] for dealing with continuous distributions.

5.1 EXPERIMENTAL SETUP

To establish a benchmark for our study and gain access to theoretical insights, the CLG model is employed [Lauritzen and Wermuth, 1989] for constructing continuous BNs. In this context, given a BN structure G, the CPDs have for expression:

$$f(x_i|\mathbf{pa}_i) = \mathcal{N}\left(x_i; \mu_i + \sum_{j=1}^{|\mathbf{Pa}_i|} b_{ij}\mathrm{pa}_{ij}, \sigma_i\right)$$

where pa_{ij} is the value of the j-th parent of X_i and $\mathcal{N}(x; \mu, \sigma)$ denotes the density of a normal distribution of mean μ and standard deviation σ. Without loss of generality, the parameters μ_i and σ_i are respectively set to 0 and 1 for all $i \in [\![1, n]\!]$. Doing so, the parameterization only depends on coefficients b_{ij} that quantify the strength of the arc between X_i and X_j.

The experiments conducted within this section focus on the estimation of the univariate marginals of the variables in the CLG. The performance of a method in estimating the marginal density of X_i is quantified through the normalized-root-mean-square-error (NRMSE) metric:

$$\mathrm{NRMSE}(i) = \left(\frac{\left\|\hat{f}_i - f_i\right\|_2}{\|f_i\|_2}\right)^{1/2}$$

between the theoretical marginal f_i and its estimated counterpart $\hat{f}_i$. The theoretical marginal is obtained using inference algorithms for CLG Lauritzen [1992].

Two inference methods are used for comparison with our new algorithm that will be labeled "Legendre" from now on. The first one, labeled "Classic", consists in the discretization of the random variables by partitioning their domains into p bins of equal size. Subsequently, the CPDs are converted into CPTs and a discrete BN is obtained, allowing to use the classic sum-product algorithms. Similarly to our algorithm, the domains of the variables are truncated and the probability mass that is ruled out is controlled by a parameter ϵ. Consequently, this method is characterized by two hyperparameters: ϵ and p, that are similar to the parameters of our algorithm. Indeed, ϵ has the same meaning for both algorithms and the number of bins and the number of Gauss nodes determine the size of the CPTs for the discrete BN. In other words, the complexity of (the memory used by) the discrete BNs is the same in both cases ("Legendre" or "classic").

The third method, labeled as "GKS", is a kernel smoothing method Wand and Jones [1994] used on a sample generated by forward sampling [Koller and Friedman, 2009, chap. 12] from the CLG. The kernel is chosen to be Gaussian and the bandwidth is fixed such that it minimizes the AMISE criteria. Consequently, the only hyperparameter for this method is the size of the sample denoted by s.

Discrete inference methods ("classical" and "legendre") require the construction of large discrete Bayesian networks. We have not taken this time into account in the benchmarks, as it is calculated off-line. Note also that all figures showing error curves are in semi-logarithmic scale.

5.2 CHAIN WITH TWO VARIABLES

In order to explore the impact of each hyperparameter of our algorithm on its performances, we consider the simplest non-trivial structure: the chain $X_1 \to X_2$. A CLG being equivalent to a multivariate normal distribution, the covariance of the latter can be expressed using parameters μ_i, σ_i and b_{ij} Neapolitan et al. [2004]. This allows to relate the coefficient b_{21} to the correlation ρ_{12} between X_1 and X_2:

$$b_{21} = \mathrm{sign}(\rho_{12})\left(\frac{1}{\rho_{12}^2} - 1\right)^{-1/2} \tag{8}$$

The evolution of the NRMSE for the marginal of X_2 is shown on figure 1 and is compared with the discretization method in the case where $\rho_{12} = 0.5$. The y-axis represents the number of discretization bins in the case of the classic method and the number of Gauss nodes in the case of our algorithm. The x-axis represents the value of ϵ used to truncate the domain of the random variables.

Our algorithm appears to be largely superior since it achieves a precision of order 10^{-8} for ($\epsilon = 10^{-8}, p = 51$) while the discrete method only attains a precision

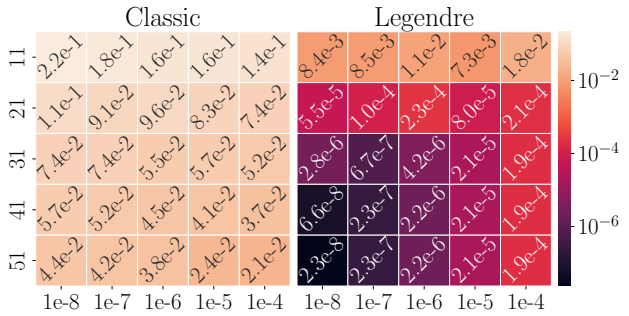


Figure 1: Heatmap visualization of the NRMSE of X_2 for the classic discretization (on left) and the Legendre transformation (on right).

of order 10^{-2} for the same values. As for the kernel method, it only attains a NRMSE of order 10^{-2} using a sample of size $s = 10^6$ (see Fig. 2).

Looking at its individual behaviour, we can observe that given a value of ϵ, the NRMSE seems to converge to a limit of the same order than ϵ as p increases. This is due to the fact that to obtain a true density function, a renormalization is done after the truncature of the domain. Hence the excluded probability mass of approximately 2ϵ is giving a bound to the precision that can be attained even if the number of Gauss nodes increases.

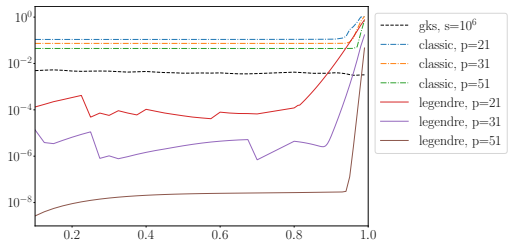
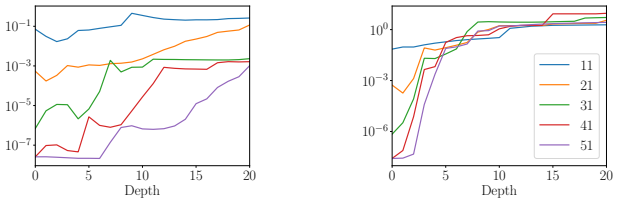


Figure 2: Evolution of the NRMSE of X_2 with respect to the correlation parameter ρ_{12} and with $\epsilon = 10^{-8}$ for *classic* and *legendre*.

Figure 2 shows the evolution of the NRMSE with respect to the value of ρ_{12} used to parameterized the CLG. The parameter ϵ for both "classic" and "legendre" is fixed to 10^{-8} and the sample size of "gks" is set to 10^6. Once again, we observe that our algorithm largely outperforms the other two methods for most of the value of ρ_{12}. However, if the correlation between the variables is too strong, the quality of the approximation is decreasing both for "classic" and "legendre". In the case of the discretization method, this is due to the fact that as the correlation approach 1, the parameter b_{21} increases. As a consequence, the mean of X_2 knowing $X_1 = x_1$ will be translated to large values compared to the domain at ϵ level. For this reason, the probability of each bin will be so small that it is rounded to 0, leading to bad results. Now, in the case of our algorithm, it can be explained by the fact that the zeros of Legendre polynomials accumulate at the bounds of the domain and most of the information will be concentrated in this area while the probability mass for Gaussian distributions is concentrated in the center of the domain.

5.3 CHAIN OF ARBITRARY SIZE

In order to visualize the loss of quality in the estimation of marginals due to the propagation of errors, we now consider a CLG with a chain structure $X_0 \to X_1 \to \cdots \to X_l$ of length $l + 1$.



(a) $\rho = 0.7$ (b) $\rho = 0.8$

Figure 3: Evolution of the NRMSE with respect to the depth of the node in the chain.

This CLG is parameterized such that the correlation $\rho_{i,i-1}$ between X_{i-1} and X_i is the same for all $i \in [\![1, l]\!]$. The marginal of each variable is estimated using our algorithm and figure 3 shows the evolution of the NRMSE with the depth of the variable in the chain. We observe that if the value of the correlation is not too high, the propagation of errors is manageable, giving enough Gauss nodes. However, in the case of a strong correlation, these errors are growing exponentially.

5.4 REFERENCE STRUCTURES

To evaluate our algorithm on larger structures, we use well-established BN structures derived from real-world applications, available from the *bnlearn* repository. Given that these BNs are discrete, we retain only the structure while reparameterizing the CPDs associated with each node using the Conditional Linear Gaussian (CLG) model. The coefficients b_{ij} are all set to the same value, leading to diverse correlations among the variables.

For each structure, the marginal of each node is estimated using the different methods and compared to the theoretical marginal via NRMSE. We estimate the marginal distribution of each node using the previous methods and compare them to the theoretical marginal distribution using the NRMSE. For our method and the "Classical" approach, we set $p = 51$ and $\epsilon = 10^{-8}$,

while the "GKS" method still uses a sample size of $s = 10^6$. Table 1 summarizes the results, presenting the maximum NRMSE for a node in the structure and the average computation time[2] across all nodes for inference tasks. Once again, our methods shows better results for NRMSE compared to the two others. The "Classic" method is the faster to make inference but it is at the expense of a poor accuracy. As for the "GKS" method, it is the slower but its accuracy is not sensitive to the structure.

NRMSE	Legendre	Classic	GKS
asia	**3.45e-07**	4.50e-02	5.28e-03
sachs	**2.58e-08**	4.49e-02	5.08e-03
child	**1.10e-04**	4.51e-02	5.18e-03
time (s)	Legendre	Classic	GKS
asia	9.42e-02	**3.30e-03**	3.17e+01
child	3.02e-01	**1.12e-01**	1.07e+02
sachs	3.67e-01	**2.13e-01**	4.58e+01

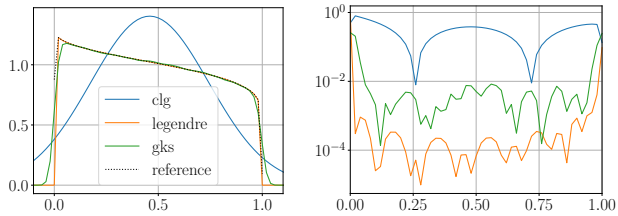
Table 1: NRMSE and computing time in seconds

5.5 NON-GAUSSIAN MODELS

In this subsection, we extend the application of our method to non-Gaussian models to demonstrate its versatility. Specifically, we construct a Copula Bayesian Network (CBN) [Elidan, 2010], which enables separate modeling of the dependency between variables and their marginal behavior. The dependency is established using a Gaussian copula [Nelsen, 2006], while the marginal behavior is modeled using beta distributions. As depicted in Figures 4a, the resulting marginals deviate significantly from Gaussian distributions. Given the absence of theoretical references in this context, reference marginals are computed using Gauss- Kronrod quadrature [Notaris, 2016] on the joint distribution. However, this method, though accurate, is prohibitively slow for high-dimensional settings, necessitating restriction to a few variables. Thus, we consider a BN structure comprising only four variables: A, B, C, and D, with arcs $A \to C, B \to C, C \to D, A \to D$. Discretization, having shown poor performance in previous sections, is omitted from this comparison. Instead, we employ a Conditional Linear Gaussian (CLG) model, constructed based on the closest CLG density in terms of Kullback-Leibler distance. Figure 4 illustrates the approximation of the marginal distribution of variable D using different techniques, along with the absolute difference between the estimated and reference values.

[2]Computations have been done using an Intel Core i7-8650U CPU @ 1.90GHz.

As expected, the CLG model exhibits considerable deviation from the reference density due to its Gaussian assumption. In contrast, our proposed model achieves closer approximation to the "true" density with significantly reduced computation time compared to kernel smoothing methods.



(a) Marginal of D (b) Absolute error

Figure 4: Estimation of the marginal of D in the CBN.

6 CONCLUSION AND FUTURE WORKS

In this paper, we introduce a novel general inference algorithm tailored for general hybrid BNs. Our method uses quadrature rules for continuous marginalization, circumventing the conventional discretization of continuous densities into histograms or the sampling. Additionally, we overcome the restrictive nature of the CLG model.Our algorithm maintains the computational efficiency of classical sum-product algorithms through an auxiliary discrete Bayesian network designed for continuous inference. Numerical experiments validate the efficacy of our approach, demonstrating its superiority over classical methods.

However, we observed limitations in our algorithm when correlations between variables approach deterministic relations. While increasing the number of quadrature points can mitigate this issue, it comes at the cost of slower inference. To address this, we can improve our algorithm with an adaptive method that dynamically adds points only as needed. In this context, utilizing Gauss-Kronrod quadratures, could offer advantages since quadrature points could be recycled. Moreover, exploring other Gauss quadratures based on orthogonal polynomials, like Hermite polynomials, may better suit for specific distributions.

Lastly, while our paper introduces a general algorithm for hybrid BNs, our experiments were confined to computing marginal densities without observations and with only continuous variables. Future implementations and research should remove these restrictions to conduct more comprehensive experiments.

References

Milton Abramowitz, Irene A. Stegun, and David Miller. Handbook of mathematical functions with formulas, graphs and mathematical tables (national bureau of standards applied mathematics series no. 55). *Journal of Applied Mechanics*, 32, 1965.

Michaël Baudin, Anne Dutfoy, Bertrand Iooss, and Anne-Laure Popelin. *OpenTURNS: An Industrial Software for Uncertainty Quantification in Simulation*, pages 1–38. Springer International Publishing, Cham, 2016. ISBN 978-3-319-11259-6. doi: 10.1007/978-3-319-11259-6_64-1. URL `https://doi.org/10.1007/978-3-319-11259-6_64-1`.

Gregory F Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial intelligence*, 42(2-3):393–405, 1990a.

Gregory F. Cooper. The computational complexity of probabilistic inference using bayesian belief networks. *Artificial Intelligence*, 1990b. ISSN 00043702. doi: 10.1016/0004-3702(90)90060-D.

Paul Dagum and Michael Luby. An optimal approximation algorithm for bayesian inference. *Artificial Intelligence*, 93(1):1–27, 1997. ISSN 0004-3702. doi: https://doi.org/10.1016/S0004-3702(97)00013-1. URL `https://www.sciencedirect.com/science/article/pii/S0004370297000131`.

Gaspard Ducamp, Christophe Gonzales, and Pierre-Henri Wuillemin. aGrUM/pyAgrum : a toolbox to build models and algorithms for Probabilistic Graphical Models in Python. In *10th International Conference on Probabilistic Graphical Models*, volume 138 of *Proceedings of Machine Learning Research*, pages 609–612, Skørping, Denmark, 2020.

Gal Elidan. Copula bayesian networks. *Advances in neural information processing systems*, 23, 2010.

Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29 (2):131–163, 1997.

Walter Gautschi. *Numerical analysis*. Springer Science & Business Media, 2011.

Daphne Koller and Nir Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.

Alexander V. Kozlov and Daphne Koller. Nonuniform dynamic discretization in hybrid networks. In Dan Geiger and Prakash P. Shenoy, editors, *UAI '97: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, Brown University, Providence, Rhode Island, USA, August 1-3, 1997*, pages 314–325. Morgan Kaufmann, 1997.

Helge Langseth, Thomas D Nielsen, Rafael Rumı, and Antonio Salmerón. Mixtures of truncated basis functions. *International Journal of Approximate Reasoning*, 53(2):212–227, 2012.

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

Steffen Lilholt Lauritzen and Nanny Wermuth. Graphical models for associations between variables, some of which are qualitative and some quantitative. *The annals of Statistics*, pages 31–57, 1989.

A. L. Madsen and Finn Verner Jensen. Lazy propagation: A junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, 113:203–245, 1999. ISSN 0004-3702.

Richard E Neapolitan et al. *Learning bayesian networks*, volume 38. Pearson Prentice Hall Upper Saddle River, 2004.

Roger B Nelsen. *An introduction to copulas*. Springer, 2006.

Sotirios E Notaris. Gauss-kronrod quadrature formulae–a survey of fifty years of research. *Electron. Trans. Numer. Anal*, 45:371–404, 2016.

Judea Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan kaufmann, 1988.

Antonio Salmeron, Rafael Rumi, Helge Langseth, Thomas D Nielsen, and Anders L Madsen. A review of inference algorithms for hybrid bayesian networks. *Journal of Artificial Intelligence Research*, 62: 799–828, 2018.

Prakash P. Shenoy. Two issues in using mixtures of polynomials for inference in hybrid bayesian networks. *Int. J. Approx. Reason.*, 53(5):847–866, 2012. doi: 10.1016/j.ijar.2012.01.008. URL `https://doi.org/10.1016/j.ijar.2012.01.008`.

Prakash P. Shenoy and Glenn Shafer. Axioms for probability and belief-function propagation. In Ronald R. Yager and Liping Liu, editors, *Classic Works of the Dempster-Shafer Theory of Belief Functions*, volume 219 of *Studies in Fuzziness and Soft Computing*, pages 499–528. Springer, 2008.

Matt P Wand and M Chris Jones. *Kernel smoothing*. CRC press, 1994.