Journal of Artificial Intelligence Research 62 (2018) 799-828 Submitted 11/17; published 08/18

A Review of Inference Algorithms for Hybrid Bayesian Networks

Antonio Salmerón ANTONIO.SALMERON@UAL.ES
Rafael Rumí RRUMI@UAL.ES
Dept. of Mathematics, University of Almería
04120 Almería, Spain

Helge Langseth HELGE.LANGSETH@NTNU.NO
Dept. of Computer and Information Science
Norwegian University of Science and Technology
7491 Trondheim, Norway

Thomas D. Nielsen TDN@CS.AAU.DK
Dept. of Computer Science, Aalborg University
9220 Aalborg, Denmark

Anders L. Madsen ANDERS@HUGIN.COM
Hugin Expert A/S
9000 Aalborg, Denmark and
Dept. of Computer Science, Aalborg University
9220 Aalborg, Denmark

Abstract

Hybrid Bayesian networks have received an increasing attention during the last years. The difference with respect to standard Bayesian networks is that they can host discrete and continuous variables simultaneously, which extends the applicability of the Bayesian network framework in general. However, this extra feature also comes at a cost: inference in these types of models is computationally more challenging and the underlying models and updating procedures may not even support closed-form solutions. In this paper we provide an overview of the main trends and principled approaches for performing inference in hybrid Bayesian networks. The methods covered in the paper are organized and discussed according to their methodological basis. We consider how the methods have been extended and adapted to also include (hybrid) dynamic Bayesian networks, and we end with an overview of established software systems supporting inference in these types of models.

1. Introduction

Probabilistic graphical models provide a well-founded and principled approach for performing inference in complex domains endowed with uncertainty. A probabilistic graphical model is a framework consisting of two parts: a qualitative component in the form of a graphical model encoding conditional independence assertions about the domain being modelled, and a quantitative component consisting of a collection of local probability distributions adhering to the independence properties specified in the graphical model. Collectively, the two

components provide a compact representation of the joint probability distribution over the domain.

Bayesian networks (BNs) (Pearl, 1988) are a particular type of probabilistic graphical model that has enjoyed widespread attention in the last three decades. Figure 1 shows the qualitative part of a BN representing the joint distribution of variables $\mathbf{X} = \{X_1, \ldots, X_5\}$. We will use lowercase letters to refer to values or configurations of values, so that x denotes a value of X and $\mathbf{x}$ is a configuration of the variables in $\mathbf{X}$. When referring to the graphical model, we will denote all nodes with an outgoing arc pointing into a particular node as the *parents* of that node, for example is $\{X_2, X_3\}$ the parent-set of X_4 in Figure 1; we will use the shorthand $\mathbf{pa}(X_i)$ to denote the parent-set of X_i. The *family* of X_i is defined as $\mathbf{fa}(X_i) = \mathbf{pa}(X_i) \cup \{X_i\}$. The *co-parents* of X_i wrt. some other variable X_j, $\mathbf{co}_{X_i}(X_j)$, is defined as X_i's parents except X_j, $\mathbf{co}_{X_i}(X_j) = \mathbf{pa}(X_i) \setminus \{X_j\}$. All nodes that a node X_i has outgoing arcs pointing into will be denoted the *children* of X_i, with the shorthand $\mathbf{ch}(X_i)$. As an example, $\mathbf{ch}(X_3) = \{X_4, X_5\}$ in Figure 1. Finally, the *Markov blanket* of a node X_i, $\mathbf{mb}(X_i)$, are the parents, children, and the children's co-parents wrt. X_i.

Attached to each node, there is a conditional probability distribution given its parents in the network, and so the joint distribution factorizes as

$$p(x_1, \ldots, x_5) = p(x_1)p(x_2|x_1)p(x_3|x_1)p(x_4|x_2, x_3)p(x_5|x_3).$$

In general, for a BN with N variables $\mathbf{X} = \{X_1, \ldots, X_N\}$, the joint distribution factorizes as

$$p(\mathbf{x}) = \prod_{i=1}^{N} p(x_i|\mathbf{pa}(x_i)), \tag{1}$$

where $\mathbf{pa}(x_i)$ with the lower-case argument now denoting the *configuration* of the set of parents of X_i in the network. Note how this factorization enforces several conditional independence statements. For instance any variable in a Bayesian network is conditionally independent of the other variables of the model given its Markov blanket, which we write $X_i \perp\!\!\!\perp \mathbf{X}|\mathbf{mb}(X_i)$ for short.

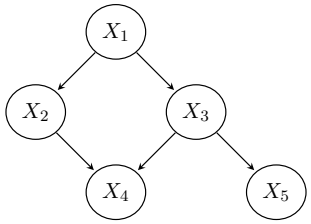

Figure 1: A Bayesian network with five variables.

BNs constitute a versatile tool with applications in a wide range of domains (Pourret, Naim, & Marcot, 2008). In practical applications, it is common to find scenarios that involve discrete and continuous variables simultaneously. Such problems can be modelled using so-called *hybrid* BNs, which are BNs where some of the variables are discrete and others are continuous.

The most common application of both Bayesian networks and hybrid Bayesian networks involves belief updating (or inference): the calculation of the posterior marginal distribution of a variable conditioned on the known values of a subset of the other variables in the model, e.g., $p(X_1|X_2, X_5)$ in Figure 1. Inference in discrete BNs is, however, already a challenging task, and additional problems arise when it comes to handling discrete and continuous variables simultaneously (Murphy, 1998; Koller & Friedman, 2009, ch. 14).

In order to assist both practitioners and researchers dealing with hybrid Bayesian network models, this paper gives an overview and illustration of the main methodological trends for performing inference in these types of models. We also extend the discussions to include dynamic domains (i.e., domains that evolve over discrete time) and provide an overview of established software systems facilitating inference in hybrid models.

As a starting-point we will look at the most prominent techniques for inference in *discrete* BNs in Section 2. In Section 3 we review the existing literature on inference in hybrid BNs related to static domains, that is, domains that do not evolve over time. Section 4 reports on dynamic BNs, where the time component is taken into account. A short overview of available software packages is given in Section 5 and the paper ends with conclusions in Section 6.

2. Inference in Discrete Bayesian Networks

Given a set of observed variables $\mathbf{X}_E \subset \mathbf{X}$ and a set of variables of interest $\mathbf{X}_I \subseteq \mathbf{X} \setminus \mathbf{X}_E$, the most common type of *probabilistic inference*, also called *probability propagation* or *belief updating*, consists of computing the posterior distribution $p(x_i|\mathbf{x}_E) = p(X_i = x_i|\mathbf{X}_E = \mathbf{x}_E)$ for each $i \in I$. In this section we shall present the most-used techniques for performing inference in *discrete* Bayesian networks, and return to the situation of hybrid BNs in the next section.

The goal of inference can now be formulated as computing

$$p(x_i|\mathbf{x}_E) = \frac{p(x_i, \mathbf{x}_E)}{p(\mathbf{x}_E)} = \frac{\sum_{\mathbf{x} \in \Omega_{\mathbf{X} \setminus \{\mathbf{X}_E \cup X_i\}}} p(\mathbf{x}, \mathbf{x}_E)}{\sum_{\mathbf{x} \in \Omega_{\mathbf{X} \setminus \mathbf{X}_E}} p(\mathbf{x}, \mathbf{x}_E)}, \tag{2}$$

where $\Omega_{\mathbf{X}}$ stands for the set of possible values of a set of variables $\mathbf{X}$ and $p(\mathbf{x}, \mathbf{x}_E)$ is the joint distribution over $\mathbf{X}$, where the observed variables $\mathbf{X}_E$ are instantiated to their observed values $\mathbf{x}_E$. Notice that since $\mathbf{X}$ is discrete, $\Omega_{\mathbf{X}}$ is countable, and the sums exist. A brute force algorithm for carrying out the inference could be as follows:

1. Obtain the joint distribution $p(\mathbf{x}) = p(x_1, \ldots, x_n)$ using Equation (1).
2. Restrict $p(\mathbf{x})$ to the value $\mathbf{x}_E$ of the observed variables $\mathbf{X}_E$. This results in $p(\mathbf{x}, \mathbf{x}_E)$.
3. Compute $p(x_i, \mathbf{x}_E)$ from $p(\mathbf{x}, \mathbf{x}_E)$ by marginalizing out every variable different from X_i using Equation (2).

The problem of this straight-forward method is that the joint distribution is usually unmanageably large. For instance, assume a simple case in which we deal with 10 qualitative variables, each having three possible states. Representing the joint distribution for

those variables would require a table with $3^{10} = 59,049$ probability values. The size of the distribution grows exponentially with the number of variables, hence the size of the corresponding table would soar up to $3^{11} = 177,147$ for 11 variables, and so on.

The inference problem can be simplified by taking advantage of the factorization of the joint distribution encoded by the structure of the BN, which enables the design of efficient algorithms for this task. For instance, consider the network in Figure 1, and assume our variable of interest is X_5, i.e. we want to compute $p(x_5)$. Further, we assume that $\mathbf{X}_E = \emptyset$ for this example. Starting off from the joint distribution, we find that

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

where $h(x_2,x_3) = \sum_{x_4} p(x_4|x_2,x_3)$. Therefore, we have reached a similar problem as initially, but with one variable less.[1] The operation illustrated in this example is called the *elimination* of variable X_4. Repeating the same procedure for each variable except X_5 would lead us to the desired result. This procedure is known as the *variable elimination algorithm* (Dechter, 1999; Li & D'Ambrosio, 1994; Zhang & Poole, 1996), where the key idea is to organize the operations among the conditional distributions in the network, so that we do not produce unnecessarily large intermediate distributions.

We will now briefly discuss some other main types of inference algorithms that work for discrete BNs, before moving on to inference in hybrid Bayesian networks in Section 3.

2.1 Inference Based on Message Passing Schemes

One limitation of the variable elimination algorithm as formulated above is that is has to be repeated once per variable we are interested in. This limitation can be overcome by using message passing schemes.

2.1.1 Join Tree Algorithms

Join tree algorithms organize the calculations over an auxiliary structure called a *join tree* or *junction tree* (Lauritzen & Spiegelhalter, 1988), which is an undirected tree constructed from the original BN, where the nodes are the *cliques* of a triangulated graph obtained from the BN. Figure 2 shows a join tree constructed from the BN in Figure 1. The join tree satisfies the following two properties (Shenoy & Shafer, 1990b):

1. As a side-comment, we note that since $h(x_2,x_3) = \sum_{x_4} p(x_4|x_2,x_3)$, we know without making the calculation that $h(x_2,x_3) = 1$ for all configurations (x_2,x_3). This is a consequence of X_4 being a *barren* node in this particular model. Utilizing such simplifications is also part of the *lazy propagation* inference scheme (Madsen & Jensen, 1999a).

1. Each family in the original Bayesian network is contained in at least one clique. This ensures that each conditional distribution can be attached to a clique, and thus the joint distribution over the join tree is the same as the joint distribution of the original network.

2. The tree meets the *running intersection property*, which means that if a variable is contained in two different cliques, it is also contained in each clique on the path connecting them. This property ensures that the relevant knowledge for each variable can be obtained by transferring information between adjacent cliques. More precisely, by transferring information relating to the variables shared between the two cliques.

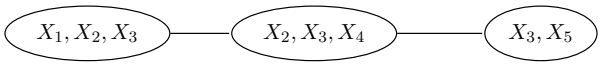

Figure 2: A join tree constructed from the BN in Figure 1.

One of the fundamental schemes for carrying out inference over trees of cliques is the algorithm developed by Shenoy and Shafer (1990b). This algorithm performs the inference by sending *messages* between neighboring nodes in the tree, in order to collect and distribute the information contained in the different parts of the tree. The message from a node V_i to one of its neighbours, V_j, is computed as

$$\phi_{V_i \to V_j} = \phi_{V_i} \prod_{V_k \in \mathbf{nb}(V_i) \setminus V_j} \phi_{V_k \to V_i} \ , \tag{3}$$

where ϕ_{V_i} is the initial probability potential on V_i, $\phi_{V_k \to V_i}$ are the messages from V_k to V_i and $\mathbf{nb}\,(V_i)$ are the neighboring nodes of V_i in the join-tree. The initial potential assigned to a node is the product of the conditional distributions that have been attached to it, and a function constantly equal to 1 if no conditional density has been assigned to it. In either case, it is defined as a function with domain equal to the combined state-space of the variables in the clique.

The complexity of this scheme is determined by the computation of the messages, which, as can be seen in Equation (3), involves the combination of the incoming messages with the function stored in the node that is sending the message. The result of this operation is a potential with size equal to the product of the number of states of the variables contained in the node which sends the message. The complexity is therefore related to the size of the largest node in the tree of cliques. In general, the computational complexity of both exact and approximate inference in BNs is NP-hard (Cooper, 1990; Dagum & Luby, 1993), but many real-world applications can nonetheless be solved efficiently by state-of-the-art inference algorithms (Shafer & Shenoy, 1990; Madsen & Jensen, 1999a; Darwiche, 2001, 2003).

Although one cannot in general hope to achieve polynomial time performance when designing and adapting inference algorithms, we can still aim for polynomial or even linear complexity that will allow for the evaluation of more complex domains. One approach for achieving this is to exploit parallel architectures, where several processors/cores can be

used to simultaneously solve different parts of the problem and afterwards obtain a global solution by combining the solutions of the parts (Díez & Mira, 1994; Kozlov & Singh, 1994, 1996; Pennock, 1998; Madsen & Jensen, 1999b; Namasivayam & Prasanna, 2006; Nikolova et al., 2009). However, the reduction in complexity is limited by the amount of computing resources allocated. It is also possible to adopt simplified models either by following modelling tricks (Díez & Druzdzel, 2006) or by using fixed structures, especially if the model is aimed at specific problems like classification (Friedman et al., 1997). Complexity reduction can also be attained by simplifying the structure of the network (Kjærulff, 1994; van Engelen, 1997) or even by using compact representations of the probability potentials in the network (Cano et al., 2000).

2.1.2 Loopy Belief Propagation

The computational cost of the junction tree algorithm is exponential in the number of nodes in the largest clique, and it may therefore not be feasible to employ this algorithm for large and/or densely connected networks. An alternative is the loopy belief propagation algorithm (Pearl, 1988; Peot & Shachter, 1991), for which messages are sent along the edges in the Bayesian network, and where the cost of inference is exponential only in the number of nodes of the largest *family*. A downwards message from a parent U_i to its child X is denoted $\pi_X(u_i)$ and from the child Y_j to X is $\lambda_{Y_j}(x)$. Their interpretations are that $\pi_{U_i}(x) = P(X = x, \mathbf{e}_{U_i})$ and $\lambda_{Y_j}(x) = P(\mathbf{e}_{Y_j}|X = x)$, where $\mathbf{e}_{U_i}$ (resp. $\mathbf{e}_{Y_j}$) denotes evidence that is upstream (resp. downstream) of X. Notice that if the network has a tree-structure, we can recover the posterior probability exactly; $P(X = x|\boldsymbol{e}) \propto \lambda(x)\pi(x)$, where $\lambda(x) = \lambda_X(x)\prod_j \lambda_{Y_j}(x)$, $\lambda_X(x)$ is the evidence on X and $\pi(x) = \sum_{\mathbf{u}} P(X = x|\mathbf{U} = \mathbf{u})\prod_{i=1}^{k} \pi_X(u_i)$ with $\mathbf{U} = \{U_1, \ldots, U_k\}$ being the parents of X. When the network is not tree-structured, an iterative scheme can nevertheless be attempted, and although guarantees are not given, it is often seen to converge in practice (Murphy, Weiss, & Jordan, 1999).

2.1.3 Variational Inference

Variational inference (Jordan, Ghahramani, Jaakkola, & Saul, 1999; Attias, 2000) is a deterministic approximate inference technique, where we seek to iteratively optimize a variational approximation to the posterior distribution of interest (Attias, 2000). Let $\mathcal{Q}$ be the set of possible approximations; then the variational approximation to a posterior distribution $p(\mathbf{x}|\mathbf{x}_E)$ is defined as

$$q^*_{\mathbf{x}_E}(\mathbf{x}) = \arg\min_{q\in\mathcal{Q}} D\left(\, q(\mathbf{x}) \,\|\, p(\mathbf{x}|\mathbf{x}_E)\,\right), \tag{4}$$

where $D\left(\, q \,\|\, p\,\right)$ is the Kullback-Leibler (KL) divergence from q to p.

A common approach is to employ a variational mean-field approximation of the posterior distribution, meaning that $\mathcal{Q}$ is constrained to functions that factors over the individual variables involved, i.e., $q^*_{\mathbf{x}_E}(\mathbf{x}) = \prod_{i\in I} q^*_{\mathbf{x}_E}(x_i)$. During the optimization of the variational mean-field one performs a coordinate ascent, where we iteratively update the individual variational distributions while holding the others fixed (Smídl & Quinn, 2006; Jaakkola & Qi, 2006). Updating a variational distribution involves calculating the variational expectation of the original log joint distribution function of the model, so for a variable $X_i \notin \mathbf{X}_E$

we have

$$q^*_{\mathbf{x}_E}(x_i) \propto \exp\left(\mathbb{E}_q[\ln p(\mathbf{x}, \mathbf{x}_E)]\right), \tag{5}$$

where the expectation is taken wrt. all $X_j \sim q^*_{\mathbf{x}_E}(\cdot)$, $j \neq i$ (Smídl & Quinn, 2006). Using the factorization of the variational distribution it follows that $q^*_{\mathbf{x}_E}(x_i)$ is determined by the variational moments[2] of the variables in $\mathbf{mb}(X_i)$, which invites an iterative solution strategy that is guaranteed to converge to a local optimum (Smídl & Quinn, 2006). Variational inference can thus be seen as a message passing scheme where a variable receives the messages from all variables in its Markov blanket, before updating its distribution according to Equation (5), and distribute its updated moments to all the variables in the Markov blanket. We will later return to variational inference to see how this algorithm can be implemented efficiently for a certain distributional family.

An alternative is to focus on minimizing $D(p \,\|\, q)$, which has the advantage of being convex with respect to q if q is a factorizing distribution (Haft, Hofmann, & Tresp, 1999), and therefore there is only one optimum approximating distribution q. However, the problem of $D(p \,\|\, q)$ is that the expectation is computed with respect to p instead of the less complex approximate distribution q.

A popular algorithm based on this approach is expectation propagation (Minka, 2001). It computes an approximate q by iteratively inserting factors of p into q minimizing $D(p \,\|\, q)$.

Even if the variational mean-field approximation shares many commonalities with expectation propagation, the two differ by the former having the interpretation that $q^*_{\mathbf{x}_E}(\mathbf{x})$ is chosen as to maximize a lower-bound of the data likelihood $p(\mathbf{x}_E)$ while the latter does not.

2.2 Monte Carlo-Based Methods

We next turn to sampling methods. The idea is to approximate the probability $p(x_i|\mathbf{x}_E)$ (or alternatively $p(x_i, \mathbf{x}_E)$) by generating random samples, and approximating the requested probability by a function over the samples. In case $p(\mathbf{x}|\mathbf{x}_E) > 0 \; \forall \mathbf{x} \in \Omega_{\mathbf{X}}$, and $p(x_i|\mathbf{pa}(x_i))$ is easy to sample from for all $X_i \in \mathbf{X} \setminus \mathbf{X}_E$, we can do *forward sampling*. For ease of notation assume that the variables X_i are labelled so that $\mathbf{pa}(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$. Then proceed to sample X_i in sample t in the following way: If X_i is observed, then $x_i^{(t)}$ is set to its observed value; otherwise it is drawn from the conditional distribution $p(X_i|X_1 = x_1^{(t)}, , \ldots, X_{i-1} = x_{i-1}^{(t)})$. When a sufficient number of samples have been obtained, the estimation of $p(x_i|\mathbf{x}_E)$ can be found from a sample mean estimator.

2.2.1 Importance Sampling

A limitation of forward sampling is that it can yield samples incompatible with the observed values $\mathbf{x}_E$. As an example, consider a network with two binary variables X_1 and X_2, where X_1 is parent of X_2, parameterized by $p(X_1 = 0) = 0.99$ and $p(X_2 = 0|X_1 = 0) = 0$. Assume that it has been observed that $X_2 = 0$. Then, 99% of the times forward sampling will obtain the configuration $(X_1 = 0, X_2 = 0)$, while $p(X_1 = 0, X_2 = 0) = p(X_1 = 0)p(X_2 = 0|X_1 = 0) = 0$ and therefore, should not be obtained. A solution to this limitation would be

2. The term "moment" is used loosely; the messages are determined by the distributional families in play, and one will typically see quantities like $\mathbb{E}_q[X_j]$, $\mathbb{E}_q[X_j^2]$, and $\mathbb{E}_q[\log(X_j)]$ being received.

to sample each variable from a distribution conditional on the observed values $\mathbf{x}_E$. In this example, that amounts to sample X_1 from $p(x_1|X_2 = 0)$. In a general setting, conditioning on the observations is equivalent to carrying out exact probabilistic inference.

Importance sampling (Hammersley & Handscomb, 1964) is a versatile simulation technique that in the case of inference in BNs amounts to transforming the numerator in Equation (2) by multiplying and dividing by a distribution p^*. The point of doing this is that sometimes it may be computationally expensive to generate samples from $p(\cdot)$, as illustrated in the example above. We instead utilize p^* as a proxy. We require that $p^*(\mathbf{x}) > 0$ whenever $p(\mathbf{x}) > 0$. Then, $p(x_i, \mathbf{x}_E)$ can be written as

$$\begin{aligned} p(x_i, \mathbf{x}_E) &= \sum_{\mathbf{x}\in\Omega_{\mathbf{X}}} p(\mathbf{x}, x_i, \mathbf{x}_E) = \sum_{\mathbf{x}\in\Omega_{\mathbf{X}}} p^*(\mathbf{x}, x_i, \mathbf{x}_E)\frac{p(\mathbf{x}, x_i, \mathbf{x}_E)}{p^*(\mathbf{x}, x_i, \mathbf{x}_E)} \\ &= \mathbb{E}_{p^*}\left[\frac{p(\mathbf{x}, x_i, \mathbf{x}_E)}{p^*(\mathbf{x}, x_i, \mathbf{x}_E)}\right], \end{aligned} \tag{6}$$

where the expected value is computed with respect to p^*. Therefore, the problem is again mapped to estimating an expected value, which can be naturally done using the sample mean estimator. Samples from p^* can be generated using the forward sampling technique. Let us denote by $\hat{p}(x_i, \mathbf{x}_E)$ the sample mean estimator of $p(x_i, \mathbf{x}_E)$. The convergence of $\hat{p}(x_i, \mathbf{x}_E)$ is determined by its variance, that for a sample of size T is

$$\mathrm{Var}(\hat{p}(x_i, \mathbf{x}_E)) = \frac{1}{T}\mathrm{Var}\left(\frac{p(\mathbf{x}, x_i, \mathbf{x}_E)}{p^*(\mathbf{x}, x_i, \mathbf{x}_E)}\right).$$

Therefore, the way to speed up convergence is to choose p^* so that $\frac{p(\mathbf{x},x_i,\mathbf{x}_E)}{p^*(\mathbf{x},x_i,\mathbf{x}_E)} \approx 1$ for all $\mathbf{x}$.

2.2.2 Markov Chain Monte Carlo

While importance sampling often has good convergence results in practice and has the benefit that it scales well because the sampling is embarrassingly parallel, it may require many samples to converge in certain cases. An alternative sampling-based approach is Markov Chain Monte Carlo (MCMC), where a Markov Chain is generated, so that it has $p(\mathbf{x}|\mathbf{x}_E)$ as its stationary distribution. A very popular approach in this class of algorithms is the Gibbs sampler (Geman & Geman, 1984; Hrycej, 1990). The key idea is that a variable X_i is independent of all other variables in the BN given its Markov blanket $\mathbf{mb}(X_i)$, thus one can sample each variable in turn conditional on the sampled variables in the Markov blanket. This can be efficient if $p(x_i|\mathbf{mb}(x_i))$ is easily determined and sampled from for all variables in the domain.

An alternative MCMC approach is the Metropolis-Hastings algorithm (Hastings, 1970). In this algorithm, a configuration over $\mathbf{X}$ in sample t, $\mathbf{x}^{(t)}$, is generated from a so-called *proposal function* $g(\cdot|\mathbf{x}^{(t-1)})$; note that the proposal distribution is conditional on the configuration of sample $t-1$. The sampled proposal is accepted with a given probability that is determined by $p(\cdot)$ and $g(\cdot|\cdot)$. If it is not accepted, $\mathbf{x}^{(t)}$ is set equal to $\mathbf{x}^{(t-1)}$. While the stationary distribution of this Markov Chain is guaranteed to be $p(\mathbf{x}|\mathbf{x}_E)$ under only mild regularity conditions, the choice of proposal function has an impact on the convergence

speed and the mixing rate, that is, the number of samples required before new samples approximately follow the the target distribution.

Since MCMC algorithms generate samples from a Markov chain, these samples are necessarily dependent, and auto-correlation of the samples should be monitored in practice.

3. Inference in Static Hybrid Bayesian Networks

Traditionally, Bayesian networks have been defined for domains where the entities of interest are modelled by discrete variables. However, this requirement imposes severe restrictions, as many domains contain entities that are more appropriately modelled by variables with continuous state spaces. To extend BNs with support for continuous variables, research has pursued various directions, which we review in the next sub-sections.

Let us consider again the domain $\mathbf{X}$, but now with both continuous and discrete variables; we again assume that $\mathbf{X}_E \subset \mathbf{X}$ has been observed. As before we are interested in calculating $p(x_i|\mathbf{x}_E)$ for each $X_i \in \mathbf{X}_I$, where $\mathbf{X}_I \subseteq \mathbf{X} \setminus \mathbf{X}_E$ is some subset of interest. We denote by $\mathbf{X}_C$ and $\mathbf{X}_D$ the set of continuous and discrete variables in $\mathbf{X} \setminus \mathbf{X}_E$, respectively. The goal of probabilistic inference in a hybrid domain can then be formulated as

$$p(x_i|\mathbf{x}_E) = \frac{p(x_i, \mathbf{x}_E)}{p(\mathbf{x}_E)} = \frac{\sum_{\mathbf{x}_D \in \Omega_{\mathbf{X}_D \setminus \{X_i\}}} \int_{\mathbf{x}_C \in \Omega_{\mathbf{X}_C \setminus \{X_i\}}} p(\mathbf{x}, \mathbf{x}_E) \mathrm{d}\mathbf{x}_C}{\sum_{\mathbf{x}_D \in \Omega_{\mathbf{X}_D}} \int_{\mathbf{x}_C \in \Omega_{\mathbf{X}_C}} p(\mathbf{x}, \mathbf{x}_E) \mathrm{d}\mathbf{x}_C}. \tag{7}$$

Notice the relation to Equation (2). In principle the only complication when one works in hybrid domains is that the continuous variables that are not in $\mathbf{X}_E$ need be integrated out, whereas we previously marginalized variables out using summation.

So far considerations about inference complexity have been made assuming discrete variables whose distribution can be represented by a table of probability values. This representation is very convenient from an operational point of view, as the operations required for inference (restriction, combination, and marginalization) are closed for probability tables. It means that all the operations required during the inference process can be carried out using only one data structure. The problem may turn more complex when we perform inference involving continuous variables. For instance, consider a BN with two continuous variables X_1 and X_2 where X_1 is the parent of X_2. Assume X_1 is a standard normal variable, i.e., its density is

$$p(x_1) = \frac{1}{\sqrt{2\pi}} e^{-x_1^2/2}, \quad -\infty < x_1 < \infty \ ,$$

and X_2 has the following density:

$$p(x_2|x_1) = \begin{cases} 3x_2^2 & \text{if } x_1 < 0, \quad 0 < x_2 < 1, \\ 2x_2 & \text{if } x_1 \geq 0, \quad 0 < x_2 < 1. \end{cases}$$

These two densities require different data structures to be handled: The first one requires a data structure able to deal with exponentials of quadratic functions, while the second requires the use of polynomials. Moreover, the combination of the two would require the

manipulation of functions that are the product of a polynomial and an exponential of a quadratic function. The problem can be even more complex if we consider hybrid networks, in which densities that are defined for discrete and continuous variables simultaneously are required (Langseth, Nielsen, Rumí, & Salmerón, 2009).

We will now consider how the algorithmic families presented in Section 2 have been adapted to hybrid domains. We will give the presentation in increasing order of model complexity: We start with models containing only Gaussian and discrete variables. Thereafter, we turn to models supporting distributions belonging to the conjugate exponential family, before we end up discussing models containing general (unconstrained) distributions.

3.1 Algorithms Based on the Gaussian Assumption

One may choose to carefully construct the hybrid model such that the exact inference algorithms developed for discrete models can be applied after only simple modifications. For instance, exact inference based on join trees is possible whenever the joint distribution over the variables of the domain is from a distribution-class that is closed under the operations required for inference (Shenoy & Shafer, 1990a). A class of models constructed in this way is the set of *Conditional Linear Gaussian* (CLG) distribution (Lauritzen & Wermuth, 1989) models.

3.1.1 Conditional Linear Gaussian Models

In the CLG model, discrete variables are not allowed to have continuous parents. The conditional distribution of each discrete variable $X \in \mathbf{X}_D$ given its parents is a multinomial, whilst the conditional distribution of each continuous variable $Z \in \mathbf{X}_C$ with discrete parents $\boldsymbol{Z}_D \subseteq \mathbf{X}_D$ and continuous parents $\boldsymbol{Z}_C \subseteq \mathbf{X}_C$, is given by

$$f(z|\mathbf{pa}(z) = \{\boldsymbol{z}_D, \boldsymbol{z}_C\}) = \mathcal{N}(z; \alpha(\boldsymbol{z}_D) + \boldsymbol{\beta}(\boldsymbol{z}_D)^{\mathrm{T}}\boldsymbol{z}_C, \sigma^2(\boldsymbol{z}_D))\,, \tag{8}$$

for all $\boldsymbol{z}_D \in \Omega_{\boldsymbol{Z}_D}$ and $\boldsymbol{z}_C \in \Omega_{\boldsymbol{Z}_C}$, where α and $\boldsymbol{\beta}$ are the coefficients of a linear regression model of Z given its continuous parents; this model can differ for each configuration of the discrete variables $\boldsymbol{Z}_D$. After fixing any configuration of the discrete variables $\mathbf{X}_D$, the joint distribution of any subset of the continuous variables $\mathbf{X}_C$ is a multivariate Gaussian.

The first attempt at developing an exact method over join trees following this approach was introduced by Lauritzen (1992) and later revised by Lauritzen and Jensen (2001). The algorithm is able to carry out exact inference in hybrid BNs, as long as the joint distribution is a CLG. Instead of join trees, a more restrictive structure, called *strong junction trees* (Cowel, Dawid, Lauritzen, & Spiegelhalter, 1999), is used. A strong junction tree is a join tree that has at least one distinguished clique R, called a *strong root*, such that for each pair (C_1, C_2) of adjacent cliques in the tree, with C_1 closer to R than C_2, it holds that $C_2 \setminus C_1$ only contains continuous variables or $C_2 \cap C_1$ only contains discrete variables. Consider again the model in Figure 1, and define $\mathbf{X}_D = \{X_1, X_3\}$, $\mathbf{X}_C = \{X_2, X_4, X_5\}$. Now the junction tree in Figure 2 is a strong junction tree with strong root $\{X_1, X_2, X_3\}$. A numerically more stable algorithm that operates under the CLG framework is given by Cowel (2005). It operates with univariate regressions on the continuous variables, avoiding matrix manipulations and the possible numerical errors entailed by these manipulations.

A more efficient algorithm for exact inference in CLG networks is given by Madsen (2008). It is based on lazy propagation (Madsen & Jensen, 1999a), that uses factorized potentials in the cliques as well as in the separators of the join tree. Semantic knowledge is used to produce a refined version of the algorithm, that takes advantage of the nature of the potentials (conditional densities or not) involved in the calculations (Zhu et al., 2012).

3.1.2 General Models of Gaussian and Discrete Variables

The CLG model does impose certain limitations on the domain being modelled: discrete variables cannot directly depend on a continuous variable and each continuous variable must follow a conditional linear Gaussian distribution. Furthermore, inference in CLG models is in general harder than in models that are fully discrete or fully Gaussian. Even in simple models like poly-trees, where exact inference is linear in the size of the model in the pure Gaussian case, approximate inference is NP-hard for CLGs (Lerner & Parr, 2001).

In order to overcome the structural restrictions mentioned above, Lerner, Segal, and Koller (2001) introduced the so-called *augmented CLG networks*, in which conditional densities of discrete nodes with continuous parents are modelled using the soft-max function. The drawback is that the algorithm is not able to provide exact inference results, as the product of Gaussian and soft-max functions is only approximated by a Gaussian. The scheme is similar to the method previously proposed by Murphy (1999), where augmented CLG networks are also used, and the product of a logistic and Gaussian function is approximated by a Gaussian.

An approximation scheme of a different nature is followed by Heskes and Zoeter (2003), where *general belief propagation* over augmented CLG networks is utilized. The algorithm is based on loopy belief propagation (cf. Section 2.1.2) with weak marginalization, which means than only the first two moments are kept after marginalization, instead of the full density. The posterior density for the variables of interest is computed by minimizing the Kikuchi energy (Yedidia, Freeman, & Weiss, 2001).

Approximate inference in CLG networks has also been approached using Monte Carlo methods. An algorithm based on adaptive importance sampling is proposed by Sun and Chang (2005), cf. Equation (6). The idea is that each variable is sampled from a conditional density given its parents in the network that is dynamically updated as the sampling process progresses. The sampling order is therefore from parents to children. The sampling conditional densities are modelled using CLG functions.

Rao-Blackwellization is used by Paskin (2004) as the core of an algorithm that combines Gibbs sampling with exact computations in a join tree. The idea is that in CLGs, inference over the continuous variables is tractable when the discrete variables are observed. Then, the posterior densities are estimated by sampling on the discrete variables and manipulating the conditional densities given the sample. Rao-Blackwellization combined with importance sampling is used by Gogate and Dechter (2005) to carry out inference on hybrid Bayesian networks with CLG distribution and discrete constraints.

Scalability is addressed by Sun, Chang, and Laskey (2010). The algorithm only provides exact results in CLG networks with poly-tree structure. Otherwise, mixture of Gaussian approximations are used. Recently, a parallel algorithm based on importance sampling has

been proposed by Salmerón et al. (2015). It is based on an adaptation of the Evidence Weighting algorithm (Fung & Chang, 1990) to CLG networks.

3.2 The Conjugate Exponential Family

We will now turn to the *conjugate exponential family* models, which have received a lot of interest lately. Under this model, each conditional distribution in the model belongs to the exponential family. Recall that if a univariate variable X is in the exponential family, its distribution can be written as

$$\ln p(x) = \ln h(x) + \boldsymbol{\eta}^T \boldsymbol{s}(x) - A(\boldsymbol{\eta}),$$

where the scalar functions $h(x)$ and $A(\boldsymbol{\eta})$ are the base measure and the log-normalizer, respectively; the vectors $\boldsymbol{\eta}$ and $\boldsymbol{s}_X(\cdot)$ are the *natural parameters* and the *sufficient statistics* vectors, respectively. The exponential family of distributions include many commonly used distributions, including both the Gaussian and the multinomial as special cases.

For the *conjugate* exponential family models we require that all distributions are of the following form:

$$\ln p(x_i|\mathbf{pa}(x_i)) = \ln h_{X_i} + \boldsymbol{\eta}_{X_i}(\mathbf{pa}(x_i))^T \boldsymbol{s}_{X_i}(x_i) - A_{X_i}(\boldsymbol{\eta}_{X_i}(\mathbf{pa}(x_i))), \tag{9}$$

where the natural parameters depend on the configuration of the parents $\mathbf{pa}(x_i)$. The subscript X_i means that the associated functional forms may be different for the different factors of the model. Since the distributions are conjugate, the posterior distribution for each variable in the model has the same functional form as its prior distribution, and probabilistic inference only changes the values of the parameters of the model rather than the functional form of the distributions.

Inference can be done efficiently and in closed form when the distributions involved are conjugate-exponential (Beal, 2003). In particular, this scheme is useful for the mean-field approximation (see Section 2.1.3). For this situation, a general architecture for supporting variational message passing in graphical models was presented by Winn and Bishop (2005), highlighting how distributions that are conjugate-exponential families (Attias, 2000; Beal, 2003; Winn & Bishop, 2005) can be utilized to efficiently represent the messages by the expected natural statistics. The key insight is to express the functional form of $p(x|\mathbf{pa}(x))$ in terms of the sufficient statistics $\boldsymbol{s}_Z(z)$ of any of the parents $Z \in \mathbf{pa}(X)$:

$$\ln p(x|\mathbf{pa}(x)) = \ln h_Z + \boldsymbol{\eta}_{XZ}(X, \mathbf{co}_Z(X))^T \boldsymbol{s}_Z(x) - A_Z(\boldsymbol{\eta}_{XZ}(x, \mathbf{co}_Z(X))),$$

where $\mathbf{co}_Z(X)$ denotes the coparents of Z with respect to X. This leads to the relationship

$$\boldsymbol{\eta}_X = \mathbb{E}_q(\boldsymbol{\eta}_X(\mathbf{pa}(X))) + \sum_{Y \in \mathbf{ch}(X)} \mathbb{E}_q(\boldsymbol{\eta}_{XY}(Y, \mathbf{co}_X(Y))),$$

and, since $\boldsymbol{\eta}_X(\cdot)$ and $\boldsymbol{\eta}_{XY}(\cdot,\cdot)$ are multi-linear in their arguments (Winn & Bishop, 2005), an efficient message passing scheme can be constructed.

Efficient implementations of variational message passing under the mean-field assumption is currently a hot research topic. For instance, a scalable solution called *distributed*

variational message passing has been developed by Masegosa et al. (2016, 2017), based on approaching variational message passing as a projected natural gradient ascent algorithm. The method is designed following the Map-Reduce paradigm.

An extension of the mean-field approximation is the so-called structured variational Bayes (Jordan et al., 1999). It relaxes the mean-field approximation by assuming a factorization of the variational approximation involving factors containing more than single variables, thereby increasing the ability to represent the complexity of the posterior belief state. Unfortunately, structured variational Bayes solutions must be tailor-made to the problem at hand, as, e.g., the solution given by Ghahramani and Jordan (1997).

Similar schemes can also be deployed for expectation propagation (Minka, 2001), but there relying on a transformation between the exponential family representation's expected sufficient statistics and the distribution's moments. The method introduced by Heskes, Opper, Wiegerinck, Winther, and Zoeter (2005) is an approximate algorithm that works for models within the exponential family. It is based on expectation propagation and is formulated for the general context of *factor graphs*, which contain Bayesian networks as special cases.

3.3 General Distribution Families

The most common idea for inference in general distribution families is to "translate" the original model into an approximate model, for which efficient exact or approximate inference algorithms can be easily applied. Other solutions consists of adopting general algorithms of approximate nature, or use different model representations.

3.3.1 Translation Procedures

A model can for instance be translated by *discretizing* the continuous variables. Inference can then be carried out using techniques developed for discrete models.

For a continuous variable X_k without parents, discretization is in essence to approximate the density function $p(x_k)$ by a piecewise constant function. The discretization process amounts to selecting an appropriate set of pieces in the approximating distribution, observing the tradeoff between loss of accuracy and increase in computational cost. For a continuous variable X_i with continuous parents $\mathbf{pa}(X_i) \subseteq \mathbf{X}_C$, the domain of the parents, $\Omega_{\mathbf{pa}(X_i)}$, is split into disjoint hypercubes $\mathcal{P}_1, \ldots, \mathcal{P}_{r_i}$. For each hypercube $\mathcal{P}_\ell$ the conditional distribution is approximated by a function $q_\ell(\cdot)$ so that $p(x_i|\mathbf{pa}(x_i)) \approx q_\ell(x_i)$ for all $\mathbf{pa}(x_i) \in \mathcal{P}_\ell$. Thereafter, each $q_\ell(x_i)$ is discretized as if it was an unconditional distribution, while choosing the same split-points for all the hypercubes $\ell = 1, \ldots, r_i$. For computational reasons, the separating of $\Omega_{\mathbf{pa}(X_i)}$ into hypercubes is defined using the same split-points as was used when $X_j \in \mathbf{pa}(X_i)$ was discretized as the head of *its* conditional distribution.

The method proposed by Kozlov and Koller (1997) performs the discretization by minimizing an upper bound of the Kullback-Leibler divergence between the true and the discretized density. The minimization is done dynamically on the join tree. This is, however, a computationally costly method that even requires a specific data structure, the so-called BSP-trees. A more efficient approach is proposed by Neil, Tailor, and Marquez (2007). The key idea here is that it is the individual posterior densities that are discretized, in-

stead of dynamically discretizing the densities in the join tree. This approach to dynamic discretization can therefore be implemented on top of any discrete inference scheme.

More sophisticated approaches use translations with higher expressive power than discretisation. These include mixtures of truncated exponentials (MTEs) (Moral, Rumí, & Salmerón, 2001) and mixtures of polynomials (MoPs) (Shenoy & West, 2011a). The mixtures of truncated basis functions (MoTBFs) framework (Langseth, Nielsen, Rumí, & Salmerón, 2012b) aims to combine these. The key idea for the MoTBF framework is that $q_\ell(x_i)$ is now approximated by a sum of *basis-functions*, i.e., a set of functions that are closed under the operations needed for probabilistic inference. That is, $q_\ell(x_i) = \sum_{r=0}^{k} \theta_{\ell,r}\psi_r(x_i)$, where $\psi_r(\cdot)$ are the pre-defined set of basis functions and $\theta_{\ell,r}$ the parameters. While piecewise constant basis-functions lead to discretization, more interesting examples include polynomials (leading to the MoP framework) and exponentials (leading to the MTE framework). There is no general rule for determining which kind of basis function (e.g., exponential or polynomial) is preferable, and successful applications of both polynomials and exponentials can be found in the literature. A procedure for efficiently finding an MoTBF approximation to any density function was described by Langseth, Nielsen, Rumí, and Salmerón (2012a).

The first inference algorithm specifically designed for MTEs was proposed by Cobb and Shenoy (2006a). It is based on the Shenoy-Shafer scheme (Shenoy & Shafer, 1990a). An approximate inference scheme was proposed by Rumí and Salmerón (2005), where the *Penniless propagation algorithm* (Cano et al., 2000) is adapted to MTEs. A more recent proposal for inference in hybrid BNs with MTEs (Fernández, Rumí, & Salmerón, 2012) is based on importance sampling with approximate pre-computation (Salmerón, Cano, & Moral, 2000), where the sampling distributions are obtained following a variable elimination scheme (Zhang & Poole, 1996) and the distributions are represented using *mixed trees* (Moral, Rumí, & Salmerón, 2003). The instantiation of the Shenoy-Shafer architecture to MoPs was reported by Shenoy and West (2011a). The most general proposal for exact inference within this framework is given by Langseth et al. (2012a). The scheme is based on postponing the combination and addition of potentials as much as possible, adopting a lazy scheme (Madsen & Jensen, 1999a). It is valid for MoTBFs in general.

The main drawback of translation procedures is that conditional densities are defined by splitting the domain of the parents into hypercubes, which means that the approximating distributions are piecewise defined functions. Hence, their combination may result in an increase of the size of the result similar to the one produced when combining probability tables. In addition to that, the combination of mixture distributions usually results in an increase in the number of terms. This fact limits its efficiency compared to exact inference in CLG models or variational methods for exponential family models.

A recent proposal (Cortijo & Gonzales, 2017) succeeds in keeping the complexity of inference comparable to that for discrete networks. The idea is to discretize the continuous variables and add an artificial continuous variable for each one of them. The conditional distribution of each extra variable is modeled using a conditionally truncated density. Since these variables do not have children, inference cam be carried out efficiently.

A related scheme has been recently introduced by Mori and Mahalec (2016). Inference is carried out over a representation of the probability distributions based on decision trees. It is equivalent to discretizing the continuous variables whenever they appear in internal nodes

of the tree. The distributions stored in the leaves are either Gaussian (for the continuous variables) or multinomial (for the discrete variables).

Another approach consists of approximating the conditional densities by kernels (Wand & Jones, 1995). However, given the complexity of kernels, the idea has only been applied to restricted BN structures used in Bayesian classifiers (Pérez, Larrañaga, & Inza, 2009).

The use of multivariate mixtures of Gaussians as a proxy has been explored by Hanea, Morales-Napoles, and Ababei (2015) and Ickstadt et al. (2010). Both works are based on the fact that mixtures of multivariate Gaussians can approximate any density arbitrarily well in $\mathbb{R}^k$ using a sufficiently large number of components (Wu & Ghosal, 2008). The resulting models are called *nonparametric Bayesian networks*. In comparison with CLGs, nonparametric BNs sidestep both the Gaussian assumption and the linearity assumption, as the relation between a node and its parents is, in this case, a mixture of linear models rather than a single linear model. The underlying idea is very similar to the procedure for learning mixtures of Bayesian networks given by Thiesson, Meek, Chickering, and Heckerman (1998) and a similar idea was also pursued by Shenoy (2006). Shenoy works with augmented CLG networks, and the structure of the network is modified using arc reversals in order to make it compatible with the exact algorithm of Lauritzen and Jensen (2001). The result of the inference is approximate, as the conditional densities resulting from the arc reversal process are approximated by mixtures of Gaussians. A general algorithm for inference in hybrid BNs was also proposed by Cinicioglu and Shenoy (2009). The fundamental idea is based on arc reversal and approximation by mixture of Gaussians of the densities in the network, incorporating the ability of dealing with deterministic conditionals.

3.3.2 General Algorithms

The next group of algorithms is also able to operate over general densities. All of them are, however, of approximate nature. Examples of general approximation schemes include the Gibbs sampler (Geman & Geman, 1984; Hrycej, 1990), variational inference (Attias, 2000; Jordan et al., 1999), and expectation propagation (Minka, 2001).

The scheme proposed by Koller, Lerner, and Anguelov (1999) is based on the Shenoy-Shafer architecture, where the approximation is two-fold. First, messages over the join tree are approximated in order to reduce their complexity, and afterwards sampling is performed in each clique. The posterior densities are estimated from the samples.

Importance sampling is the methodology underlying the proposal presented by Yuan and Druzdzel (2007b). It is valid for any density representation that allows sampling. The posterior densities are given in terms of mixtures of Gaussians. A similar approach enhanced with the ability to deal with deterministic conditionals is given by Yuan and Druzdzel (2007a). Mixtures of Gaussians are the underlying model in the so-called mix-nets (Davis & Moore, 2000), where the authors propose to carry out inference using sampling or variational approximations, given that exact inference is not possible.

The so-called *nonparametric belief propagation* (Sudderth, Ihler, Isard, Freeman, & Willsky, 2010) also operates over general hybrid BNs. It is based on approximating densities by sampling and then fitting a kernel to the sample. The drawback is that kernels are not efficient representations for large samples or in complex models.

When using variational inference (cf. Section 3.2) for distributions that fall outside the family of conjugate-exponential models, one has for instance resorted to developing tight lower bounds to commonly occurring functions (Jordan et al., 1999; Jaakkola & Jordan, 2000) or apply stochastic optimization for calculating the required expectations (Paisley, Blei, & Jordan, 2012; Hoffman, Blei, Wang, & Paisley, 2013). Recently, stochastic approximation methods have been devised for direct optimization of the variational bound. First, assume the variational distribution $q \in \mathcal{Q}$ is parameterized by $\boldsymbol{\nu}$. Then, observe that the optimization in Equation (4) can equivalently be written as the maximization of $\mathcal{L}(q_{\mathbf{x}_E}(\boldsymbol{x}_I|\boldsymbol{\nu}))$, where (Jordan et al., 1999)

$$\mathcal{L}(q_{\mathbf{x}_E}(\boldsymbol{x}_I|\boldsymbol{\nu})) = \mathbb{E}_q\left[\log(p(\mathbf{x})) - \log(q_{\mathbf{x}_E}(\boldsymbol{x}_I|\boldsymbol{\nu}))\right]. \tag{10}$$

This leads to an iterative gradient-based updating scheme over $\boldsymbol{\nu}$, where, at iteration t, $\boldsymbol{\nu}$ is updated as

$$\boldsymbol{\nu}_t \leftarrow \boldsymbol{\nu}_{t-1} + \rho_t \nabla_{\boldsymbol{\nu}} \mathcal{L}(q(\boldsymbol{x}|\boldsymbol{\nu}_{t-1})),$$

where ρ_t is a sequence of learning rates fulfilling the Robbins-Monro conditions (Robbins & Monro, 1951). For nonconjugate-exponential models, the expectation in Equation (10) cannot be calculated in closed form. Instead, Ranganath, Gerrish, and Blei (2014) exploit that $\nabla_{\boldsymbol{\nu}} \mathcal{L}(q(\boldsymbol{x}|\boldsymbol{\nu}))$ can be calculated as

$$\nabla_{\boldsymbol{\nu}} \mathcal{L}(q(\boldsymbol{x}|\boldsymbol{\nu})) = \mathbb{E}_q\left[\nabla_{\boldsymbol{\nu}} \log(q_{\mathbf{x}_E}(\boldsymbol{x}_I|\boldsymbol{\nu}))\left[\log(p(\mathbf{x})) - \log(q_{\mathbf{x}_E}(\boldsymbol{x}_I|\boldsymbol{\nu}))\right]\right], \tag{11}$$

which leads to *black-box* variational inference, where the gradient in Equation (11) is approximated by sampling from $q(\boldsymbol{x}|\boldsymbol{\nu}_{t-1})$. Hence, black-box variational inference is flexible in the sense that the only constraints that should be satisfied is that one needs to be able to sample from the variational distribution and be able to calculate $\nabla_{\boldsymbol{\nu}} \log(q_{\mathbf{x}_E}(\boldsymbol{x}_I|\boldsymbol{\nu}))$ as well as $\log(p(\mathbf{x}))$ and $\log(q_{\mathbf{x}_E}(\boldsymbol{x}_I|\boldsymbol{\nu}))$. Due to the sampling component, the efficiency of the method is influenced by the variance of the estimated gradient. Measures to reduce this variance include the use of Rao-Blackwellization and control variates.

3.3.3 Other Representations

Probabilistic inference in hybrid models has also been studied from the perspective of other graphical models related to Bayesian networks. Of special interest are the *probabilistic decision graphs* (PDGs) (Jaeger, 2004), which are particularly designed to support efficient inference. A convenient feature of PDGs is that any discrete BN can be transformed into a PDG (Jaeger, 2004). It is also known that CLG models can be represented as PDGs (Nielsen, Gámez, & Salmerón, 2012) and therefore inference on such hybrid models can be efficiently performed over them. Recently, it has been shown that MoTBFs can also be represented as PDGs, though so far they have only been tested in practice in supervised classification problems (Fernández, Rumí, del Sagrado, & Salmerón, 2014).

4. Inference in Dynamic Bayesian Networks

Many domains have strong internal structure that can often be appropriately described using an object oriented language, either due to repetitive substructures or substructures that can naturally be ordered in a superclass–subclass hierarchy (Koller & Pfeffer, 1997;

Neil, Fenton, & Nielson, 2000). For instance, in temporally evolving (or dynamic) domains we often see the same type of object repeated over time (for an unbounded number of steps). Bayesian networks extended to such dynamic domains are called *Dynamic Bayesian networks* (Dean & Kanazawa, 1989; Kjærulff, 1992; Murphy, 2002) and encode the temporal dynamics of the system explicitly in the model (these models can therefore also be seen as special cases of OOBNs).

For regular types of dynamic Bayesian networks we usually assume that observations are made at a fixed time-frequency and that the probability distribution underlying the transition model and the observation model are invariant over time. Most dynamic Bayesian networks used in practice are also assumed to follow a first order Markov process, adhere to the sensor Markov assumption, and be stationary. A special type of dynamic Bayesian network is the hidden Markov model (HMM) (Rabiner, 1989). In these models, a latent (vector) variable $\mathbf{X}_t$ is used to represent the *belief state* at time t and the (vector) variable $\mathbf{E}_t$ the observations. The dynamic aspect of the model is conveyed only through the connection of the belief states over time. Smyth (1994) presents a simple dynamic model of this type for detecting both known and unknown failure states of a system, and he also demonstrates the necessity of modelling temporal dynamics in order to achieve high prediction accuracy. The *Input-Output* HMMs (Bengio, 1999) extend the HMMs by explicitly modelling "input" variables $\mathbf{U}_t$ that are controlled externally, resulting in an effect on the "output" variables $\mathbf{E}_t$. An example of a three time slice input-output HMM is shown in Figure 3. The assumption of having a fixed observation-frequency underlying the above model classes is dropped in the continuous time Bayesian networks, which can be seen as the limit model when the length of the time-interval encoded in the DBN approaches zero (Nodelman, Shelton, & Koller, 2002).

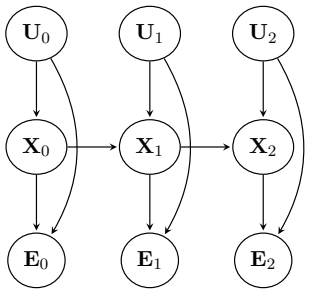

Figure 3: A three time slice input-output HMM.

Inference in dynamic Bayesian networks obviously share the computational difficulties of regular Bayesian networks, but in the dynamic case we are also often faced with additional problems. One is the entanglement problem, where after a certain time step, all variables $\mathbf{x}_t$ describing the belief state have become dependent after observing $\{\mathbf{u}_{1:t}, \mathbf{e}_{1:t}\}$. Consequently, we cannot represent the exact belief state $p(\mathbf{x}_t|\mathbf{e}_{1:t}, \mathbf{u}_{1:t})$ in factorized form (Boyen & Koller, 1998, 1999). As an example, Figure 4 illustrates a dynamic Bayesian network, where all variables used to encode the belief state at time $t = 2$, i.e., X_2^1, X_2^2, and X_2^3, have become dependent after observing the evidence $\{e_0, e_1, e_2\}$. Rather than dealing with this fully correlated belief state one often employs approximate methods including approximate

factorizations of the joint probability distribution describing the system state (Boyen & Koller, 1998) as well as sampling based techniques in the form of particle filtering (Doucet, De Freitas, Murphy, & Russell, 2000). An important specialization of the HMMs, which enforces internal structure on the belief state during specification, is the factorial HMMs (Ghahramani & Jordan, 1997), and its later extension to infinite factorial HMMs (Van Gael, Teh, & Ghahramani, 2009) that is further generalized by Doshi, Wingate, Tenenbaum, and Roy (2011). Inference in the two latter model classes is performed using sampling.

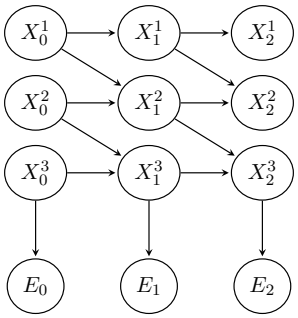

Figure 4: A three time slice dynamic Bayesian network. All variables used to describe the belief state (i.e. X_2^1, X_2^2, and X_2^3) have become correlated at time $t = 2$.

Similar to the extension of the static Bayesian network model to hybrid domains, dynamic Bayesian network models have likewise been extended to continuous and hybrid domains (Lerner, Moses, Scott, McIlraith, & Koller, 2002). In purely continuous domains, where the continuous variables follow linear Gaussian distributions, the dynamic Bayesian network corresponds to (a factorized version of) a Kalman filter (1960). In these types of models, the dynamics of the process are assumed to be linear, and exact inference can be performed efficiently. When modeling non-linear domains, the dynamics and observational distributions are often approximated through, e.g., the extended Kalman filter or the unscented Kalman filter (Julier & Uhlmann, 1997).

In hybrid domains where the continuous variables follow a conditional linear Gaussian distribution, the model is also known as a switching dynamical linear system (SDLS), where changes in the discrete variables cause the continuous linear dynamics to switch. A particular problem with an SDLS model is that the marginal distribution over the continuous variables (at a certain point in time) is a mixture of multivariate Gaussian distributions, and the number of mixture components grow exponentially and unboundedly over time. In order to ensure that the mixture of Gaussians does not grow too large several approximate solution techniques have been proposed. For example, Barber (2006) collapses a mixture of Gaussians into a single Gaussian in order to avoid the exponential blow-up in the number of mixture components. Alternatively, one could also apply discretization approaches (Straub, 2009; Vinh, Chetty, Coppel, & Wangikar, 2012; Zhu & Collette, 2015) or restrict the model to only cover the k most recent observations, thereby effectively limiting the number of mixture components. Solutions based on Markov Chain Monte Carlo have also been explored (Iamsumang, Mosleh, & Modarres, 2014, 2015).

Variational approximations have also been used in this setting (see, e.g., the work by Ghahramani & Hinton, 2000), but as noted by, e.g., Turner and Sahani (2011), the compactness-property of variational inference (which is a consequence of the calculation scheme's built-in preference for approximations $q(\mathbf{x})$ that are strictly positive only when the true distribution $p(\mathbf{x})$ is strictly positive) can lead to a failure to propagate uncertainty in time, thus limiting the usefulness of the calculated belief states. Alternatives include Expectation propagation (Zoeter & Heskes, 2011) and structured variational Bayes (Jordan et al., 1999), where the model complex factors defined by the latter increases the ability to represent the complexity of the posterior belief state.

A Bayesian formulation of a stationary dynamic Bayesian network complicates the inference. Each model parameter, modelled as an (unobserved) latent variable, will have a child in each time-step that effectively introduces new correlations over time. These variables, denoted *global hidden variables* (Hoffman et al., 2013), require special attention using the variational Bayes formulation (Ghahramani & Beal, 2000; Hoffman et al., 2013; Broderick, Boyd, Wibisono, Wilson, & Jordan, 2013; Luttinen, 2013).

5. Software for Inference in Hybrid Bayesian Networks

Table 1 contains a list of available software products implementing inference in hybrid Bayesian networks. Column "Models" refers to the type of distribution used. In that sense, unconstrained means that they handle different types of distributions, but of course inference in such cases is only carried out by means of Monte Carlo methods. The column "Inference" indicates the type of inference algorithm used. The keys used in this column are VE for variable elimination, VMP for variational message passing, EP for expectation propagation, MCMC for Markov Chain Monte Carlo, and VB for variational Bayes (see Section 3). Column "Open Source / Language" indicates if it is an open source product, in which case its programming language is given. In the last column, API stands for application programming interface and GUI for graphical user interface.

6. Conclusions

This document gives an overview of state-of-the art of inference in hybrid Bayesian networks. While inference in static hybrid BNs has been widely studied, scalability has only received limited attention. Given the complexity of the inference task, approximate algorithms are of special interest.

The complexity of inference in dynamic models not only resembles the difficulties of the static case, but the inclusion of the time component increases the magnitude of the problem and emphasizes the need of approximate scalable algorithms. The problem is even more complex if a Bayesian formulation of the inference problem is adopted.

As a general rule, exact inference is only feasible for models of moderate size. In such case, if the normality assumption holds, CLG-related algorithms stand out as the right choice. If the underlying process is not Gaussian, or the structure restriction imposed by CLG models does not fit the problem under study, then the use of translation procedures like discretization or MoTBFs constitute a clear alternative. In Table 2 we give an overview of how the referenced inference methods support different distributional assumptions.

Name	Website	Models	Inference	Free?	Open Source / Language	Platform(s)
AgenaRisk	http://www.agenarisk.com	Discretised	Join Tree	No	No	All
AMIDST toolbox	http://amidsttoolbox.com	Conjugate exponential	VMP, importance sampling	Yes	Java 8	All
Analytica	http://www.lumina.com	Unconstrained	Sampling	No	No	Windows
Bayes Server	http://www.bayesserver.com	CG	VE, Sampling	No	No	API: all, GUI: Windows
BayesiaLab	http://www.bayesia.com	Discretised	Join Tree	No	No	All
BNT	https://github.com/bayesnet/bnt	CG	Several	Yes	Matlab / C	All
Dimple	https://github.com/AnalogDevicesLyricLabs/dimple	Discretised	Several	Yes	Matlab / Java	All
Edward	http://edwardlib.org/	Unconstrained	MCMC, VB	Yes	Python	All
Elvira	http://leo.ugr.es/elvira/	MTE	VE	Yes	Java	All
gR	https://cran.r-project.org/web/views/gR.html	CG	Join Tree	Yes	R	All
Hugin Expert	http://www.hugin.com	CG	Join Tree	No	No	All
Infer.NET	http://research.microsoft.com/en-us/um/cambridge/projects/infernet/	Conjugate exponential	VMP, EP, Gibbs sampling	Yes	C#	All
JAGS	http://mcmc-jags.sourceforge.net/	Unconstrained	MCMC	Yes	Java	All
Stan	http://mc-stan.org/	Unconstrained	MCMC, VB	Yes	C++	All
Vibes	http://vibes.sourceforge.net	Conjugate exponential	VMP	Yes	Java	All

Table 1: Software supporting inference in hybrid BNs.

	Methodological basis			
	Message passing	Monte Carlo	Variational inference	Translation based
Discrete	Lauritzen and Spiegelhalter (1988), Shenoy and Shafer (1990b), Madsen and Jensen (1999a), Murphy et al. (1999), Shafer and Shenoy (1990), Darwiche (2001, 2003), Namasivayam and Prasanna (2006), Nikolova et al. (2009)		Attias (2000), Smídl and Quinn (2006), Jaakkola and Qi (2006)	Kjærulff (1994), van Engelen (1997), Friedman et al. (1997), Díez and Druzdzel (2006), Cano et al. (2000)
GGM	Lauritzen (1992), Cowel et al. (1999), Cowel (2005), Madsen (2008), Lauritzen and Jensen (2001), Nielsen et al. (2012), Zhu et al. (2012), Lerner and Parr (2001), Lerner et al. (2001), Yedidia et al. (2001), Heskes and Zoeter (2003), Sun et al. (2010)	Sun and Chang (2005), Paskin (2004), Gogate and Dechter (2005), Salmerón et al. (2015)	Ghahramani and Jordan (1997), Murphy (1999)	
CEF	Winn and Bishop (2005)		Attias (2000), Minka (2001), Beal (2003), Winn and Bishop (2005), Heskes et al. (2005), Paisley et al. (2012), Hoffman et al. (2013), Masegosa et al. (2016, 2017)	
GDF	Koller et al. (1999)	Hrycej (1990), Davis and Moore (2000), Yuan and Druzdzel (2007b, 2007a), Sudderth et al. (2010)	Jordan et al. (1999), Minka (2001), Ranganath et al. (2014)	Kozlov and Koller (1997), Neil et al. (2007), Moral et al. (2001), Shenoy and West (2011a, 2011b), Cobb and Shenoy (2006a, 2006b), Rumí and Salmerón (2005, 2007), Shenoy (2006), Fernández et al. (2012), Langseth et al. (2009, 2012a), Cinicioglu and Shenoy (2009), Ickstadt et al. (2010), Shenoy et al. (2015), Hanea et al. (2015), Mori and Mahalec (2016), Cortijo and Gonzales (2017)

Table 2: Classification of inference methods relative to distributional assumptions and inference principles. For the distributional assumptions we distinguish between purely discrete distributions (discrete), general Gaussian models (GGMs) with discrete and Gaussian distributed variables (this includes CLG models as a special case), the conjugate exponential family (CEF), and general distribution families (GDF). Note that methods listed for e.g., the CDF model class are also applicable to any specialized model-subset within that class such as CLG models.

When facing large problems, scalability turns out to be crucial. In such scenarios, variational inference algorithms are the preferred alternative in the literature, for handling models within the conjugate exponential family. Outside that family of distributions, one has to resort to Monte Carlo algorithms or methods like black box variational inference. This group of algorithms is receiving considerable attention in recent literature, and will probably be the focus of future research.

Acknowledgments

This work was performed as part of the AMIDST project. AMIDST has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no 619209. This research has been partly funded by the Spanish Ministry of Economy and Competitiveness, through project TIN2016-77902-C3-3-P and by ERDF funds.

References

Attias, H. (2000). A variational Bayesian framework for graphical models. *Advances in neural information processing systems*, *13*, 209–215.

Barber, D. (2006). Expectation correction for smoothed inference in switching linear dynamical systems. *Journal of Machine Learning Research*, *7*, 2515–2540.

Beal, M. J. (2003). *Variational algorithms for approximate Bayesian inference*. Ph.D. thesis, Gatsby Computational Neuroscience Unit, University College London.

Bengio, Y. (1999). Markovian models for sequential data. *Neural Computing Surveys*, *2*, 129–162.

Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. In *Proceedings of the Fourteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pp. 33–42.

Boyen, X., & Koller, D. (1999). Exploiting the architecture of dynamic systems. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence (AAAI-99)*, pp. 313–320.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., & Jordan, M. (2013). Streaming variational Bayes. In *Advances in Neural Information Processing Systems 26*, pp. 1727–1735. Neural Information Processing Systems.

Cano, A., Moral, S., & Salmerón, A. (2000). Penniless propagation in join trees. *International Journal of Intelligent Systems*, *15*, 1027–1059.

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

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

Cobb, B., & Shenoy, P. (2006b). Operations for inference in continuous Bayesian networks with linear deterministic variables. *International Journal of Approximate Reasoning*, *42*, 21–36.

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

Cortijo, S., & Gonzales, C. (2017). On conditional truncated densities Bayesian networks. *International Journal of Approximate Reasoning*, *In Press*.

Cowel, R. (2005). Local propagation in conditional Gaussian Bayesian networks. *Journal of Machine Learning Research*, *6*, 1517–1550.

Cowel, R., Dawid, A., Lauritzen, S., & Spiegelhalter, D. (1999). *Probabilistic networks and decision graphs*. Springer.

Dagum, P., & Luby, M. (1993). Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, *60*(1), 141–153.

Darwiche, A. (2001). Recursive conditioning. *Artificial Intelligence*, *125*(1–2), 5–41.

Darwiche, A. (2003). A differential approach to inference in Bayesian networks. *Journal of the Association for Computing Machinery*, *50*(3), 280–305.

Davis, S., & Moore, A. (2000). Mix-nets: Factored mixtures of Gaussians in Bayesian networks with mixed continuous and discrete variables. In Boutilier, C., & Goldszmidt, M. (Eds.), *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pp. 168–175.

Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, *5*(3), 142–150.

Dechter, R. (1999). Bucket elimination: a unifiying framework for reasoning. *Artificial Intelligence*, *113*, 41–85.

Díez, F., & Druzdzel, M. (2006). Canonical probabilistic models for knowledge engineering. Tech. rep. 06-01, CISIAD.

Díez, F., & Mira, J. (1994). Distributed inference in Bayesian networks. *Cybernetics and Systems*, *25*, 39–61.

Doshi, F., Wingate, D., Tenenbaum, J. B., & Roy, N. (2011). Infinite dynamic Bayesian networks. In *International Conference on Machine Learning (ICML-11)*, pp. 913–920.

Doucet, A., De Freitas, N., Murphy, K., & Russell, S. (2000). Rao-Blackwellised particle filtering for dynamic Bayesian networks. In *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, pp. 176–183.

Fernández, A., Rumí, R., del Sagrado, J., & Salmerón, A. (2014). Supervised classification using hybrid probabilistic decision graphs. In van der Gaag, L., & Feelders, A. (Eds.), *PGM 2014*, Vol. 8754 of *Lecture Notes in Artificial Intelligence*, pp. 206–221. Springer.

Fernández, A., Rumí, R., & Salmerón, A. (2012). Answering queries in hybrid Bayesian networks using importance sampling. *Decision Support Systems*, *53*, 580–590.

Friedman, N., Geiger, D., & Goldszmidt, M. (1997). Bayesian network classifiers. *Machine Learning*, *29*, 131–163.

Fung, R., & Chang, K. (1990). Weighting and integrating evidence for stochastic simulation in Bayesian networks. In Henrion, M., Shachter, R., Kanal, L., & Lemmer, J. (Eds.), *Uncertainty in Artificial Intelligence*, Vol. 5, pp. 209–220. North-Holland (Amsterdam).

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *6*, 721–741.

Ghahramani, Z., & Beal, M. J. (2000). Propagation algorithms for variational Bayesian learning. In Leen, T. K., Dietterich, T. G., & Tresp, V. (Eds.), *Advances in Neural Information Processing Systems 12*, pp. 507–513. MIT Press.

Ghahramani, Z., & Hinton, G. E. (2000). Variational learning for switching state-space models. *Neural Computation*, *12*(4), 831–864.

Ghahramani, Z., & Jordan, M. I. (1997). Factorial hidden Markov models. *Machine Learning*, *29*(2-3), 245–273.

Gogate, V., & Dechter, R. (2005). Approximate inference algorithms for hybrid Bayesian networks with discrete constraints. In Bacchus, F., & Jaakkola, T. (Eds.), *Proceedings of the 21st Conference on Uncertainty in Artificial Intelligence*, pp. 209–216.

Haft, M., Hofmann, R., & Tresp, V. (1999). Model-independent mean-field theory as a local method for approximate propagation of information.. *Network*, *10*, 93–105.

Hammersley, J., & Handscomb, D. (1964). *Monte Carlo Methods*. Chapman & Hall.

Hanea, A., Morales-Napoles, O., & Ababei, D. (2015). Non-parametric Bayesian networks: Improving theory and reviewing applications. *Reliability Engineering and System Safety*, *144*, 265–284.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, *57*, 97–109.

Heskes, T., Opper, M., Wiegerinck, W., Winther, O., & Zoeter, O. (2005). Approximate inference techniques with expectation constraints. *Journal of Statistical Mechanics: Theory and Experiment*, *2005*, 1105.

Heskes, T., & Zoeter, O. (2003). Generalized belief propagation for approximate inference in hybrid Bayesian networks. In *Proceedings of the 9th Workshop on A.I. and Statistics*.

Hoffman, M., Blei, D., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, *14*, 1303–1347.

Hrycej, T. (1990). Gibbs sampling in Bayesian networks (research note). *Artificial Intelligence*, *46*, 351–363.

Iamsumang, C., Mosleh, A., & Modarres, M. (2014). Computational algorithm for dynamic hybrid Bayesian network in on-line system health management applications. In *Prognostics and Health Management (PHM), 2014 IEEE Conference on*, pp. 1–8.

Iamsumang, C., Mosleh, A., & Modarres, M. (2015). Hybrid DBN monitoring and anomaly detection algorithms for on-line shm. In *Reliability and Maintainability Symposium (RAMS), 2015 Annual*, pp. 1–7.

Ickstadt, K., Bornkamp, B., Grzegorczyk, M., Wieczorek, J., Sheriff, M. R., Grecco, H., & Zamir, E. (2010). Nonparametric Bayesian networks. In Bernardo, J., Bayarri, M., Berger, J., Heckerman, A. D. D., Smith, A., & West, M. (Eds.), *Bayesian Statistics*, Vol. 9. Oxford University Press.

Jaakkola, T. S., & Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, *10*(1), 25–37.

Jaakkola, T. S., & Qi, Y. (2006). Parameter expanded variational Bayesian methods. In *Advances in Neural Information Processing Systems*, pp. 1097–1104.

Jaeger, M. (2004). Probabilistic decision graphs – combining verification and AI techniques for probabilistic inference. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *12*, 19–42.

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

Julier, S. J., & Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems. In *SPIE AeroSense Symposium*, pp. 182–193.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, *82*, 35–45.

Kjærulff, U. (1994). Reduction of computational complexity in Bayesian networks through removal of weak dependencies. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pp. 374–382. Morgan Kaufmann.

Kjærulff, U. (1992). A computational scheme for reasoning in dynamic probabilistic networks. In *Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence*, pp. 121–129, San Francisco, California. Morgan Kaufmann Publishers.

Koller, D., & Friedman, N. (2009). *Probabilistic graphical models: Principles and techniques*. MIT Press.

Koller, D., Lerner, U., & Anguelov, D. (1999). A general algorithm for approximate inference and its application to hybrid Bayes nets. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pp. 324–333.

Koller, D., & Pfeffer, A. (1997). Object-oriented Bayesian networks. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pp. 302–313. Morgan Kaufmann.

Kozlov, A. V., & Koller, D. (1997). Nonuniform dynamic discretization in hybrid networks. In *Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence*, pp. 314–325.

Kozlov, A. V., & Singh, J. P. (1996). Parallel implementations of probabilistic inference. *Computer*, *29*(12), 33–40.

Kozlov, A. V., & Singh, J. P. (1994). A parallel Lauritzen-Spiegelhalter algorithm for probabilistic inference. In *Proceedings of the 1994 ACM/IEEE conference on Supercomputing - Supercomputing '94*, pp. 320–329. ACM Press.

Langseth, H., Nielsen, T., Rumí, R., & Salmerón, A. (2009). Inference in hybrid Bayesian networks. *Reliability Engineering and System Safety*, *94*, 1499–1509.

Langseth, H., Nielsen, T., Rumí, R., & Salmerón, A. (2012a). Inference in hybrid Bayesian networks with mixtures of truncated basis functions. In *Proceedings of the Sixth European Workshop on Probabilistic Graphical Models (PGM'2012*, pp. 171–178.

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

Lauritzen, S. L., & Jensen, F. (2001). Stable local computation with conditional Gaussian distributions. *Statistics and Computing*, *11*(2), 191–203.

Lauritzen, S., & Spiegelhalter, D. (1988). Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B*, *50*, 157–224.

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

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

Lerner, U., & Parr, R. (2001). Inference in hybrid networks: Theoretical limits and practical algorithms. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pp. 310–318. Morgan Kaufmann.

Lerner, U., Segal, E., & Koller, D. (2001). Exact inference in networks with discrete children of continuous parents. In *Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence*, pp. 319–328.

Lerner, U., Moses, B., Scott, M., McIlraith, S., & Koller, D. (2002). Monitoring a complex physical system using a hybrid dynamic Bayes net. In *Proceedings of the 18th Annual Conference on Uncertainty in Artificial Intelligence (UAI-02)*.

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

Luttinen, J. (2013). Fast variational Bayesian linear state-space model. In *Machine Learning and Knowledge Discovery in Databases*, Vol. 8188 of *Lecture Notes in Computer Science*, pp. 305–320. Springer Berlin Heidelberg.

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

Madsen, A. L., & Jensen, F. V. (1999a). Lazy propagation: A junction tree inference algorithm based on lazy evaluation. *Artificial Intelligence*, *113*, 203–245.

Madsen, A. L., & Jensen, F. V. (1999b). Parallelization of inference in Bayesian networks. Technical report R-99-5002, Department of Computer Science, Aalborg University.

Masegosa, A. R., Martínez, A. M., Langseth, H., Nielsen, T. D., Salmerón, A., Ramos-López, D., & Madsen, A. L. (2017). Scaling up Bayesian variational inference using distributed computing clusters. *International Journal of Approximate Reasoning*, *88*, 435–451.

Masegosa, A., Martínez, A., Langseth, H., Nielsen, T., Salmerón, A., Ramos-López, D., Rumí, R., & Madsen, A. (2016). d-VMP: Distributed variational message passing. In *PGM'2016. JMLR: Workshop and Conference Proceedings*, Vol. 52, pp. 321–332.

Minka, T. (2001). Expectation propagation for approximate Bayesian inference. In *Proceedings of the Seventeenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-01)*, pp. 362–369.

Moral, S., Rumí, R., & Salmerón, A. (2003). Approximating conditional MTE distributions by means of mixed trees. *ECSQARU'03. Lecture Notes in Artificial Intelligence*, *2711*, 173–183.

Moral, S., Rumí, R., & Salmerón, A. (2001). Mixtures of truncated exponentials in hybrid Bayesian networks. In *EQSCARU'2001*, Vol. 2143 of *Lecture Notes in Artificial Intelligence*, pp. 145–167. Springer, Berlin, Germany.

Mori, J., & Mahalec, V. (2016). Inference in hybrid Bayesian networks with large discrete and continuous domains. *Expert Systems with Applications*, *49*, 1–19.

Murphy, K. (1999). A variational approximation for Bayesian networks with discrete and continuous latent variables. In *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*, pp. 457–466.

Murphy, K. P. (1998). Inference and learning in hybrid Bayesian networks. Tech. rep. UCB/CSD-98-990, EECS Department, University of California, Berkeley.

Murphy, K. P. (2002). *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. thesis, UC Berkeley, Computer Science Division.

Murphy, K. P., Weiss, Y., & Jordan, M. I. (1999). Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, UAI'99, pp. 467–475, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Namasivayam, V., & Prasanna, V. (2006). Scalable parallel implementation of exact inference in Bayesian networks. In *12th International Conference on Parallel and Distributed Systems - (ICPADS'06)*, Vol. 1, p. 8 pp. IEEE.

Neil, M., Fenton, N., & Nielson, L. (2000). Building large-scale Bayesian networks. *The Knowledge Engineering Review*, *15*(3), 257–284.

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

Nielsen, J., Gámez, J., & Salmerón, A. (2012). Modelling and inference with conditional Gaussian probabilistic decision graphs. *International Journal of Approximate Reasoning*, *53*, 929–945.

Nikolova, O., Zola, J., & Aluru, S. (2009). A parallel algorithm for exact Bayesian network inference. In *2009 International Conference on High Performance Computing (HiPC)*, pp. 342–349. IEEE.

Nodelman, U., Shelton, C., & Koller, D. (2002). Continuous time Bayesian networks. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 378–387.

Paisley, J. W., Blei, D. M., & Jordan, M. I. (2012). Variational Bayesian inference with stochastic search. In *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pp. 1367–1374.

Paskin, M. (2004). Sample propagation. In Thrun, S., Saul, L., & Schölkopf, B. (Eds.), *Advances in Neural Information Processing Systems 16*, pp. 425–432. MIT Press.

Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Mateo, CA.

Pennock, D. M. (1998). Logarithmic time parallel Bayesian inference. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 431–438.

Peot, M. A., & Shachter, R. D. (1991). Fusion and propagation with multiple observations in belief networks. *Artificial Intelligence*, *48*(3), 299–318.

Pérez, A., Larrañaga, P., & Inza, I. (2009). Bayesian classifiers based on kernel density estimation: Flexible classifiers. *International Journal of Approximate Reasoning*, *50*, 341–362.

Pourret, O., Naim, P., & Marcot, B. (2008). *Bayesian networks. A practical guide to applications*. Statistics in Practice. Wiley.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, *77*(2), 257–286.

Ranganath, R., Gerrish, S., & Blei, D. M. (2014). Black box variational inference.. In *AISTATS*, pp. 814–822.

Robbins, H., & Monro, S. (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, *22*(3), 400–407.

Rumí, R., & Salmerón, A. (2005). Penniless propagation with mixtures of truncated exponentials. *Lecture Notes in Computer Science*, *3571*, 39–50.

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

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

Salmerón, A., Ramos-López, D., Borchani, H., Masegosa, A., Fernández, A., Langseth, H., Madsen, A., & Nielsen, T. (2015). Parallel importance sampling in conditional linear Gaussian networks. *CAEPIA'2015. Lecture Notes in Artificial Intelligence*, *9422*, 36–46.

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

Shenoy, P., & Shafer, G. (1990a). Axioms for probability and belief-function propagation. In *Proceedings of the Sixth Workshop on Uncertainty in Artificial Intelligence*, pp. 169–198.

Shenoy, P. P., & Shafer, G. (1990b). Axioms for probability and belief functions propagation. In Shachter, R., Levitt, T., Lemmer, J., & Kanal, L. (Eds.), *Uncertainty in Artificial Intelligence, 4*, pp. 169–198. North Holland, Amsterdam.

Shenoy, P. (2006). Inference in hybrid Bayesian networks with mixtures of Gaussians. In *Proceedings of the 22nd Conference on Uncertainty in Artificial Intelligence*, pp. 428–436.

Shenoy, P., Rumí, R., & Salmerón, A. (2015). Practical aspects of solving hybrid Bayesian networks containing deterministic conditionals. *International Journal of Intelligent Systems*, *30*, 265–291.

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

Shenoy, P. P., & West, J. C. (2011b). Extended Shenoy-Shafer architecture for inference in hybrid Bayesian networks with deterministic conditionals. *International Journal of Approximate Reasoning*, *52*(6), 805–818.

Smídl, V., & Quinn, A. (2006). *The Variational Bayes Method in Signal Processing*. Springer, New York.

Smyth, P. (1994). Markov monitoring with unknown states. *IEEE Journal of Selected Areas in Communications, Special Issue on Intelligent Signal Processing for Communications*, *12*(9), 1600–1612.

Straub, D. (2009). Stochastic modeling of deterioration processes through dynamic Bayesian networks. *Journal of Engineering Mechanics*, *135*, 1089–1099.

Sudderth, E., Ihler, A., Isard, M., Freeman, W., & Willsky, A. (2010). Nonparametric belief propagation. *Communications of the ACM*, *53*(10), 95–103.

Sun, W., & Chang, K. (2005). Probabilistic inference using linear Gaussian importance sampling for hybrid Bayesian networks. In *Signal Processing, Sensor Fusion, and Target Recognition XIV. Proc. of SPIE*, Vol. 5809, pp. 322–329.

Sun, W., Chang, K., & Laskey, K. (2010). Scalable inference for hybrid Bayesian networks with full density estimations. In *Proceedings of the 13th Conference on Information Fusion*, pp. 1–8.

Thiesson, B., Meek, C., Chickering, D., & Heckerman, D. (1998). Learning mixtures of DAG models. In Cooper, G., & Moral, S. (Eds.), *Proceedings of the Fourteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pp. 504–513, San Francisco, CA. Morgan Kaufmann.

Turner, R. E., & Sahani, M. (2011). Two problems with variational expectation maximisation for time-series models. In Barber, D., Cemgil, T., & Chiappa, S. (Eds.), *Bayesian Time series models*, chap. 5, pp. 109–130. Cambridge University Press.

van Engelen, R. A. (1997). Approximating Bayesian belief networks by arc removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(8), 916–920.

Van Gael, J., Teh, Y. W., & Ghahramani, Z. (2009). The infinite factorial hidden Markov model. In *Advances in Neural Information Processing Systems*, Vol. 21.

Vinh, N., Chetty, M., Coppel, R., & Wangikar, P. (2012). Data discretization for dynamic Bayesian network based modeling of genetic networks. In Huang, T., Zeng, Z., Li, C., & Leung, C. (Eds.), *Neural Information Processing*, Vol. 7664 of *Lecture Notes in Computer Science*, pp. 298–306. Springer Berlin Heidelberg.

Wand, M., & Jones, M. (1995). *Kernel smoothing*. Chapman & Hall, London.

Winn, J., & Bishop, C. (2005). Variational message passing. *Journal of Machine Learning Research*, *6*, 661–694.

Wu, Y., & Ghosal, S. (2008). Kullback Leibler property of kerner mixture priors in Bayesian density estimation. *Electronic Journal of Statistics*, *2*, 298–331.

Yedidia, J., Freeman, W., & Weiss, Y. (2001). Bethe free energy, Kikuchi approximations and belief propagation algorithms. In *Advances in Neural Information Processing Systems*, Vol. 13.

Yuan, C., & Druzdzel, M. (2007a). Generalized evidence pre-propagated importance sampling for hybrid Bayesian networks. In *AAAI'07 Proceedings of the 22nd national conference on Artificial intelligence*, Vol. 2, pp. 1296–1302.

Yuan, C., & Druzdzel, M. (2007b). Importance sampling for general hybrid Bayesian networks. In *Proceedings of the 11th International Conference on Artificial Intelligence and Statistics*, pp. 652–659.

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

Zhu, J., & Collette, M. (2015). A dynamic discretization method for reliability inference in dynamic Bayesian networks. *Reliability Engineering and System Safety*, *138*, 242–252.

Zhu, M., Liu, S., & Yang, Y. (2012). Propagation in CLG Bayesian networks based on semantic modeling. *Artificial Intelligence Review*, *38*, 149–162.

Zoeter, O., & Heskes, T. (2011). Expectation propagation and generalised EP methods for inference in switching Kalman filter models. In Barber, D., Cemgil, A. T., & Chiappa, S. (Eds.), *Probabilistic Methods for Time-Series Analysis*, pp. 181–207. Cambridge University Press.