Reliability Engineering and System Safety 169 (2018) 269–280

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Efficient approximate inference in Bayesian networks with continuous variables

Chenzhao Li, Sankaran Mahadevan*

Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN 37235, United States

ARTICLE INFO

Keywords:
Unscented Kalman filter
Bayesian network
Inference
Continuous variable
Auxiliary variable

ABSTRACT

Inference is one key objective in a Bayesian network (BN), and it aims to estimate the posterior distributions of state variables based on evidence (observations). While efficient analytical inference algorithms (either approximate or exact) for BN with discrete variables have been well-established in the literature, the inference in BN with continuous variables is still challenging if the BN is non-linear and/or non-Gaussian. In this case we can either discretize the continuous variable and utilize the inference approaches for discrete BN; or we have to use sampling-based methods such as MCMC for static BN and particle filter for dynamic BN. This paper proposes a network collapsing technique based on the concept of probability integral transform to convert a multi-layer BN to an equivalent simple two-layer BN, so that the unscented Kalman filter can be applied to the collapsed BN and the posterior distributions of state variables can be obtained analytically. For dynamic BN, the proposed method is also able to propagate the state variables to the next time step analytically using the unscented transform, based on the assumption that the posterior distributions of state variables are Gaussian. Thus the proposed method achieves a very fast approximate solution, making it particularly suitable for dynamic BN where inference and uncertainty propagation are required over many time steps.

1. Introduction

During the past 30 years, the Bayesian network (BN) has become a key method for representation and reasoning under uncertainty in the fields of engineering [1,2], machine learning [3,4], artificial intelligence [5,6], etc. BN is a directed acyclic graph (DAG) model that represents the joint distribution of a set of random variables. In a BN, random variables are denoted by nodes (vertices) and their dependence relationships are denoted by directed edges (arcs). An edge indicates the conditional dependence of the down-stream child node on the up-stream parent node(s). This dependence is described mathematically by a conditional probability distribution (CPD), which can be a small discrete conditional probability table (CPT) where both the child and parent nodes have a limited number of possible values, or a continuous distribution where the child and/or parent nodes occupy a continuous sampling space of an infinite number of possible values. A BN may have different types of random variables as nodes, including discrete and continuous variables of different distribution types. The BN is also able to incorporate heterogeneous information, such as operational data, laboratory data, reliability data, expert opinion, and mathematical models (physics-based as well as empirical) [7]. Based on the earlier discussion, a BN can be denoted as $\langle\langle \boldsymbol{V}, \boldsymbol{E}, \rangle \boldsymbol{P}\rangle$, where $\boldsymbol{V} = \{\boldsymbol{X}, \boldsymbol{Y}\}$ is the vector of nodes (random variables); $\boldsymbol{X}$ denotes the state variables to be inferred and $\boldsymbol{Y}$ denotes the observable variables; $\boldsymbol{E}$ represents the directed edges; and $\boldsymbol{P}$ denotes the CPDs for the edges in $\boldsymbol{E}$.

The methodology of BN has two main components: inference and learning. Inference aims to estimate the posterior distribution of the state variables based on evidence. Usually this evidence is the observation $\boldsymbol{y}$ of nodes $\boldsymbol{Y}$, thus the inference is to calculate the posterior probability distribution $P(\boldsymbol{X}|\boldsymbol{Y} = \boldsymbol{y})$. Learning aims to construct the DAG and estimate the CPD for each edge based on the data of the random variables; thus learning calculates $\boldsymbol{E}$ and $\boldsymbol{P}$. This paper focuses on inference, i.e., calculating $P(\boldsymbol{X}|\boldsymbol{Y} = \boldsymbol{y})$. The inference is based on Bayes' theorem:

$$P(\boldsymbol{X}|\boldsymbol{Y} = \boldsymbol{y}) \propto P(\boldsymbol{X})P(\boldsymbol{y}|\boldsymbol{X}) \quad (1)$$

where $P(\boldsymbol{X})$ and $P(\boldsymbol{X}|\boldsymbol{Y} = \boldsymbol{y})$ are the prior and posterior distributions of state variables $\boldsymbol{X}$, and $P(\boldsymbol{y}|\boldsymbol{X})$ is the likelihood function of $\boldsymbol{X}$.

The BN explained above refers to a "static" Bayesian network for a time-independent system. To track a time-dependent system whose states evolve over time, the concept of BN is extended to a dynamic Bayesian network (DBN), which can be considered as a series of static BNs, one for each time instant, with additional edges connecting the state variables in adjacent time instants. Based on the first-order Markov assumption, the state variables of the BN at time t depend only on the

* Corresponding author.
E-mail address: sankaran.mahadevan@vanderbilt.edu (S. Mahadevan).

http://dx.doi.org/10.1016/j.ress.2017.08.017
Received 12 February 2017; Received in revised form 2 July 2017; Accepted 29 August 2017
Available online 14 September 2017
0951-8320/

state variables of the BN at time $t-1$, and this dependence and the underlying CPDs are generally assumed to be time-invariant [8]. In addition, the observable variable $\boldsymbol{Y}_t$ at time t only depends on the state variable $\boldsymbol{X}_t$ at the same time instant. The following expressions and equations can be derived from this first-order Markov assumption:

$$\begin{aligned} &\boldsymbol{X}_t \perp \boldsymbol{y}_{1:t-1} | \boldsymbol{X}_{t-1} \Rightarrow P(\boldsymbol{X}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{X}_{t-1}) = P(\boldsymbol{X}_t | \boldsymbol{X}_{t-1}) \\ &\boldsymbol{y}_t \perp \boldsymbol{y}_{1:t-1} | \boldsymbol{X}_t \Rightarrow P(\boldsymbol{y}_t | \boldsymbol{X}_t, \boldsymbol{y}_{1:t-1}) = P(\boldsymbol{y}_t | \boldsymbol{X}_t) \end{aligned} \tag{2}$$

In Eq. (2), the symbol "⊥" means "independent of"; thus the first formula in Eq. (2) denotes that $\boldsymbol{X}_t$ is independent of $\boldsymbol{y}_{1:t-1}$ at a given value of $\boldsymbol{X}_{t-1}$; and the second formula denotes that $\boldsymbol{y}_t$ is independent of $\boldsymbol{y}_{1:t-1}$ at a given value of $\boldsymbol{X}_t$. The inference in DBN is to estimate the probability $P(\boldsymbol{X}_t|\boldsymbol{y}_{1:t})$, i.e., the posterior distribution of the state variables in the current time instant given observations in the past and current time instants. This paper considers inference in both static and dynamic Bayesian networks.

The inference in a DBN is a recursive process across time instants. Using Bayes' theorem and Eq. (2), if $\boldsymbol{X}_t$ are continuous variables we have

$$\begin{aligned} &P(\boldsymbol{X}_t | \boldsymbol{y}_{1:t}) \propto P(\boldsymbol{X}_t | \boldsymbol{y}_{1:t-1}) P(\boldsymbol{y}_t | \boldsymbol{X}_t,\ \boldsymbol{y}_{1:t-1}) \\ &= \left[\int P(\boldsymbol{X}_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{X}_{t-1}) P(\boldsymbol{X}_{t-1} | \boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{X}_{t-1}\right] P(\boldsymbol{y}_t | \boldsymbol{X}_t) \\ &= \left[\int P(\boldsymbol{X}_t | \boldsymbol{X}_{t-1}) P(\boldsymbol{X}_{t-1} | \boldsymbol{y}_{1:t-1}) \mathrm{d}\boldsymbol{X}_{t-1}\right] P(\boldsymbol{y}_t | \boldsymbol{X}_t) \end{aligned} \tag{3}$$

In Eq. (3), $P(\boldsymbol{y}_t|\boldsymbol{X}_t,\ \boldsymbol{y}_{1:t-1})$ is replaced by $P(\boldsymbol{y}_t|\boldsymbol{X}_t)$ based on the second formula of Eq. (2); and $P(\boldsymbol{X}_t|\boldsymbol{y}_{1:t-1}, \boldsymbol{X}_{t-1})$ is replaced by $P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})$ based on the first formula of Eq. (2). Then Eq. (3) can be rewritten as $P(\boldsymbol{X}_t|\boldsymbol{y}_{1:t}) \propto [\int P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})P(\boldsymbol{X}_{t-1}|\boldsymbol{y}_{1:t-1})\mathrm{d}\boldsymbol{X}_{t-1}]P(\boldsymbol{y}_t|\boldsymbol{X}_t)$, where the terms on the right-hand side indicate two components in estimating $P(\boldsymbol{X}_t|\boldsymbol{y}_{1:t})$:

1. Propagate the posterior distribution $P(\boldsymbol{X}_{t-1}|\boldsymbol{y}_{1:t-1})$ obtained at time $t-1$ through the transient CPD $P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})$ and marginalize over $\boldsymbol{X}_{t-1}$ to construct the prior distribution $P(\boldsymbol{X}_t|\boldsymbol{y}_{1:t-1})$ at time t;
2. Calculate the likelihood function $P(\boldsymbol{y}_t|\boldsymbol{X}_t)$ which only utilizes the observation at time t. These two components also imply that the state variables and observations at earlier time instants can be neglected once the prior distribution $P(\boldsymbol{X}_t|\boldsymbol{y}_{1:t-1})$ at time t is constructed. This process is repeated for the BN in each time instant in order to track the evolution of the state variables over time.

Note that if $\boldsymbol{X}_t$ are discrete variables, Eq. (3) will be re-derived as $P(\boldsymbol{X}_t|\boldsymbol{y}_{1:t}) \propto [\sum_{\boldsymbol{X}_{t-1}} P(\boldsymbol{X}_t|\boldsymbol{X}_{t-1})P(\boldsymbol{X}_{t-1}|\boldsymbol{y}_{1:t-1})]P(\boldsymbol{y}_t|\boldsymbol{X}_t)$. The implication of the two components in the previous paragraph is still valid. Both cases (discrete $\boldsymbol{X}_t$ vs. continuous $\boldsymbol{X}_t$) will be discussed in this paper, but the proposed algorithm focuses on continuous state variables.

In Eq. (2) for static BN and Eq. (3) for DBN, the product of the prior distribution and the likelihood function is only proportional to but not equal to the posterior distribution. Thus a specific inference algorithm, either exact or approximate, is required to calculate the PDF/PMF value of the posterior distribution or generate random samples representing the posterior distribution. Fast, analytical inference algorithms for static/dynamic BN with *discrete variables* have been well-developed in the literature, but current algorithms for static/dynamic BN with *continuous variables* are either time-consuming or restricted to specific CPDs and/or BN topology.

This paper aims to develop a more general approximate inference algorithm for static/dynamic BN with continuous variables. Of course, continuous variables can be discretized in order to make use of the methods for discrete BN such as junction tree method, but that also introduces approximations and subjectivity, especially in deciding the number of discrete variables. We can weaken this approximation by increasing the number of discretization levels but this would bring intensive computation so that the scalability would be affected. In this paper, we are interested in developing a fast inference method by directly considering continuous variables without considering discretization. The main idea of the proposed algorithm is to utilize an auxiliary variable method based on the probability integral transform [9,10] to collapse a complex BN of arbitrary topology to a two-layered BN so that the unscented Kalman filter (UKF) can be used for inference. The proposed algorithm is analytical and fast, and is applicable to static/dynamic BNs of any topology and CPDs as long as the assumption of Gaussian posterior distribution is acceptable.

The rest of the paper is organized as follows. Section 2 gives an overview of Bayesian network inference algorithms in the literature. Section 3 gives a brief introduction of the unscented Kalman filter, which is used in the proposed method. Section 4 develops the proposed method and Section 5 provides two numerical examples.

2. Overview of inference algorithms for static/dynamic BN

2.1. Static BN

Exact and approximate inference algorithms for static BNs have been developed in the literature, as shown in Fig. 1. For a static BN with *discrete variables*, exact inference is always possible and available algorithms include the popular junction tree algorithm [11], the variable elimination algorithm [12], the arc reversal method [13], and the differential approach [14]. However, exact inference is computationally prohibitive for large networks, thus approximate inference algorithms such as loopy belief propagation [15] have been developed to improve the computational efficiency.

For a static BN with *continuous variables*, if all the root nodes (i.e., nodes without parents) have Gaussian distributions and all the edges from parent nodes $\boldsymbol{U} \in \mathbb{R}^{N_U}$ to child node $V \in \boldsymbol{V}$ are linear Gaussian CPDs such that $P(V|\boldsymbol{U}) \sim N(\boldsymbol{W}_V U + \boldsymbol{\mu}_V, \sigma_V^2)$ where matrix $\boldsymbol{W}_V \in \mathbb{R}^{N_U \times N_U}$ and vector $\boldsymbol{\mu}_V \in \mathbb{R}^{N_U}$ and variance $\sigma_V^2 \in \mathbb{R}$ have been predefined, then the joint distribution of $\boldsymbol{V}$ is multivariate Gaussian. Inference $P(\boldsymbol{X}|\boldsymbol{Y}=\boldsymbol{y})$ for this static BN is simply a conditional Gaussian distribution and the exact analytical solution can be found in [16].

A more general static BN will have non-Gaussian variables, thus a sampling-based approximate inference algorithm is needed. The sampling algorithms can be categorized into importance sampling (IS) and Markov Chain Monte Carlo (MCMC) methods. The major difference between these two categories is that the IS generates samples independently from an importance function in one shot, while the MCMC methods generate samples sequentially thus the next sample depends on the current sample. IS has several variants including: 1) the logic sampling algorithm [17] where the importance function is the prior distribution of BN; and 2) the adaptive importance sampling algorithm [18,19] where the importance function is optimized adaptively. Note that these sampling algorithms are also applicable for static BN with discrete variables.

As shown in Fig. 1, usually the stochastic simulation algorithms are the only choice for static BN with continuous non-Gaussian variables. These sample-based methods are computationally expensive for large networks; therefore this paper aims to develop an approximate but analytical algorithm to improve the efficiency.

2.2. DBN

Exact and approximate inference algorithms for the DBN have been developed in the literature, as shown in Fig. 2. For a DBN with *discrete variables*, exact inference is always possible and available algorithms include the forward-backward algorithm [20] and the frontier algorithm [21], etc. As shown in Eq. (3), the inference at time t of the DBN is not related to earlier state variables and observations once the prior distribution of $\boldsymbol{X}_t$ are constructed, and the subsequent step is the inference for the BN at time t, which is static. Thus the exact inference algorithms for static BN can be extended to DBN. Murphy [22] proposed the interface algorithm by extending the junction tree algorithm to inference in DBNs with discrete *variables*. Approximate inference algorithms for