Bayesian Network Learning for Data-Driven Design

Zhen Hu
Department of Industrial and Manufacturing Systems Engineering,
University of Michigan-Dearborn,
2340 Heinz Prechter Engineering Complex (HPEC),
Dearborn, MI 48128
e-mail: zhennhu@umich.edu

Sankaran Mahadevan[1]
Professor
Department of Civil and Environmental Engineering,
Vanderbilt University,
272 Jacobs Hall VU, PMB 351831,
Nashville, TN 37235
e-mail: sankaran.mahadevan@vanderbilt.edu

Bayesian networks (BNs) are being studied in recent years for system diagnosis, reliability analysis, and design of complex engineered systems. In several practical applications, BNs need to be learned from available data before being used for design or other purposes. Current BN learning algorithms are mainly developed for networks with only discrete variables. Engineering design problems often consist of both discrete and continuous variables. This paper develops a framework to handle continuous variables in BN learning by integrating learning algorithms of discrete BNs with Gaussian mixture models (GMMs). We first make the topology learning more robust by optimizing the number of Gaussian components in the univariate GMMs currently available in the literature. Based on the BN topology learning, a new multivariate Gaussian mixture (MGM) strategy is developed to improve the accuracy of conditional probability learning in the BN. A method is proposed to address this difficulty of MGM modeling with data of mixed discrete and continuous variables by mapping the data for discrete variables into data for a standard normal variable. The proposed framework is capable of learning BNs without discretizing the continuous variables or making assumptions about their conditional probability densities (CPDs). The applications of the learned BN to uncertainty quantification and model calibration are also investigated. The results of a mathematical example and an engineering application example demonstrate the effectiveness of the proposed framework. [DOI: 10.1115/1.4039149]

Keywords: Bayesian network, Gaussian mixture model, uncertainty quantification

1 Introduction

Data-driven design requires the use of machine learning algorithms to learn the system characteristics from available data. Among many machine learning algorithms, Bayesian networks (BNs) have been receiving increasing attention in recent years and are being studied for uncertainty quantification, reliability analysis, and model calibration, which are essential elements in engineering design under uncertainty [1–6]. For example, Shahan and Seepersad applied BN for the set-based collaborative design [7]; Khakzad et al. developed a dynamic safety analysis approach by mapping bow-tie into BN [8,9]; Yuan et al. evaluated the risk of dust explosion scenarios using BNs [10]; Gradowska and Cooke developed a method to estimate the expected information using BNs and applied it to fish consumption advisory [11]; Liang and Mahadevan proposed a Bayesian Network-based reliability-based design optimization method [12]; Bartram and Mahadevan used BNs to integrate heterogeneous information in structural health monitoring [13]; Groth et al. developed a method for human reliability analysis using BNs [14,15]; Sankararaman and Mahadevan proposed a methodology to integrate model verification, validation, and calibration using BNs [16]. All of the aforementioned applications of BNs are based on known physics models or modeling of BNs based on known causal dependence relations. In these problems, we usually assume that we know the conditional probability densities (CPDs) or conditional probability tables (CPTs), and the BN topology.

In practical applications, often we may not have mathematical models, but only observed data. In that case, the BN needs to be constructed (learned) purely based on data before being used for design or other purposes mentioned previously [17]. For instance, connections between social factors and parameters of green vehicles are sought to be discovered from data to guide the design of vehicles [18]. Learning of a BN refers to discovering the underlying BN topology and/or CPTs or CPDs from data. During the past decades, BN learning approaches and algorithms have been studied in many fields including bioinformatics [19], biological engineering [20], and machine learning [21].

Most of the current algorithms focus on learning of BNs with only discrete variables. In many engineering applications, however, we often have both discrete and continuous variables. Learning of a BN with both discrete and continuous variables is more challenging than learning with only discrete variables. To overcome this difficulty, approaches have been proposed based on discretization of continuous variables [22], vine-copula models [23,24], mixture of truncated exponential polynomials [25], Gaussian mixture models (GMMs) [26], and assumption of linear Gaussian CPDs [27]. Even if these approaches can perform BN learning for some specific problems, the accuracy and efficiency of these methods need to be further improved for application to generalized engineering problems with both discrete and continuous variables.

In this paper, we propose a new framework for the learning of BNs based on the integration of score functions (defined for BNs with only discrete variables) and GMMs. The two main contributions of this paper are: (1) the development of an optimal univariate Gaussian mixture (OUGM) approach for BN structure learning and (2) the development of a new multivariate Gaussian mixture (MGM) approach to improve the accuracy of CPD learning. Two kinds of data mapping approaches are implemented in order to realize the above-mentioned two contributions respectively, namely: (1) mapping of data for continuous variables to data for hidden discrete variables and (2) mapping of data for discrete variables to data for a standard normal variable. The applications of the learned BN to uncertainty quantification and model calibration are also studied.

The remainder of this paper is organized as follows. Section 2 reviews background concepts of BN learning and GMMs. Section 3 develops the proposed method. Section 4 considers two

[1]Corresponding author.

Manuscript received May 16, 2017; final manuscript received January 15, 2018; published online April 18, 2018. Assoc. Editor: Faisal Khan.

numerical examples to illustrate the proposed method, and Sec. 5 gives concluding remarks.

2 Background

2.1 Bayesian Networks. A Bayesian network is a probabilistic graphical model that represents the joint probability distribution of a set of variables through a directed acyclic graph [28,29]. It is capable of representing the causal dependence and flow of information in complex systems. For a Bayesian network with variables (nodes), $V_1, V_2, \ldots, V_n$, the joint probability is given by

$$P(V_1, V_2, \ldots, V_n) = \prod_{i=1}^{n} P(V_i|\boldsymbol{\pi}_i) \quad (1)$$

in which $P(\cdot)$ stands for probability, $\boldsymbol{\pi}_i$ are the parent nodes of node V_i, and $P(V_i|\boldsymbol{\pi}_i)$ is defined by CPTs for discrete nodes and by CPDs for continuous nodes.

In general, there are three elements in a BN: (a) the directed acyclic graph, $\mathbb{G}$, which defines the BN structure, (b) the conditional probabilities $P(V_i|\boldsymbol{\pi}_i)$, $i = 1, 2, \ldots, n$, which give the CPTs and CPDs between parent nodes and child nodes, and (c) marginal probabilities $P(V_j)$, $j = 1, 2, \ldots, r$, where V_j are the root nodes (no parents) and r is the number of root nodes.

2.2 Learning of Bayesian Network From Data. Learning of BN from data mainly consists of two parts: structure learning (topology) and parameter learning. The problem can be summarized as

Given: data, $\mathbb{D}$.

Find: topology $\mathbb{G}$, and marginal and conditional probabilities (CPTs and/or CPDs) given $\mathbb{G}$.

Algorithms for BN structure learning can be roughly classified into *constraint-based* methods and *score-based* methods [30]. The basis of constraint-based methods is the inductive causation algorithm proposed in Ref. [31]. The score-based methods are more widely used than constraint-based methods. Three commonly used score functions include the minimal description length (MDL), Bayesian–Dirichlet equivalence (BDe), and mutual information test (MIT). The MDL score function is given by [30]

$$\mathrm{MDL}(\mathbb{G}, \mathbb{D}) = -\log P(\mathbb{D}|\mathbb{G}, \boldsymbol{\theta}) + (r(\mathbb{G})\log N)/2 \quad (2)$$

where N is the number of training data, $\boldsymbol{\theta}$ are parameters estimated using maximum likelihood (ML) for given $\mathbb{G}$, and $r(\mathbb{G})$ is the number of free parameters in $\mathbb{G}$.

The BDe criterion maximizes the following posterior conditional probability [32]

$$\mathrm{P}(\mathbb{G}|\mathbb{D}) \propto \mathrm{P}(\mathbb{G})\mathrm{P}(\mathbb{D}|\mathbb{G}) = \mathrm{P}\Big(\mathbb{G}\Big)\int \mathrm{P}(\mathbb{D}|\mathbb{G}, \boldsymbol{\theta})\mathrm{P}(\boldsymbol{\theta}|\mathbb{G})d\theta \quad (3)$$

After derivations, the BDe score function is given by [32]

$$\begin{aligned}\mathrm{BDe}(\mathbb{G}, \mathbb{D}) = \prod_{j=1}^{n}\prod_{\mathbf{b}}(\Gamma(\alpha_{j\mathbf{b}})/\Gamma(\alpha_{j\mathbf{b}}\\ + \alpha'_{j\mathbf{b}})\prod_{a}(\Gamma(\alpha_{ja\mathbf{b}} + \alpha'_{ja\mathbf{b}})/\alpha'_{ja\mathbf{b}}))\end{aligned} \quad (4)$$

where $\Gamma(\cdot)$ is Gamma function, $\alpha_{ja\mathbf{b}}$ is the number of instances in the data, and $\alpha'_{ja\mathbf{b}}$ are the Dirichlet distribution parameters.

The MIT score function is defined as [32]

$$\mathrm{MIT}(\mathbb{G}, \mathbb{D}) = \sum_{i=0;\,\boldsymbol{\pi}_i\neq\varnothing}^{n}\left\{2N \cdot I(X_i, \boldsymbol{\pi}_i) - \sum_{j=1}^{s_i}\chi_{\alpha l_{i\sigma_i(j)}}\right\} \quad (5)$$

where $I(X_i, \boldsymbol{\pi}_i)$ is the mutual information between X_i and its parents $\boldsymbol{\pi}_i$, s_i is the number of parent nodes, and $\chi_{\alpha l_{i\sigma_i(j)}}$ is the chi-square distribution at confidence level $1-\alpha$.

After the BN topology is learned from the data, the CPTs are estimated using the ML approach, or expectation maximization (EM) method [33]. The above-mentioned score functions, however, have been developed for problems with only discrete variables and cannot be directly applied to BN with both discrete and continuous variables.

2.3 Gaussian Mixture Model. Gaussian mixture model represents an arbitrary probability distribution using mixtures of Gaussian components. For a random variable X with probability density function (PDF) $f_X(x)$, its PDF is approximated in GMM using K component Gaussian distributions as follows:

$$f_X(x) = \sum_{i=1}^{K}\lambda_i f_i(x) \quad (6)$$

where $f_i(x)$ is the PDF of the ith Gaussian component, λ_i is the weight of the ith Gaussian component, and $\sum_{i=1}^{K}\lambda_i = 1$.

The EM method is commonly used to estimate the parameters of $f_i(x)$ and weights λ_i [33]. The optimal number of components is selected using the Bayesian information criterion (BIC) and Akaike's information criterion (AIC) scores. The BIC score is given by [33]

$$\mathrm{BIC}(G_{\mathrm{GMM}}) = \log P(\mathbb{D}|G_{\mathrm{GMM}}, \boldsymbol{\theta}) - (r(G_{\mathrm{GMM}})\log N)/2 \quad (7)$$

where $\boldsymbol{\theta}$ are the parameters of G_{GMM} and $r(G_{\mathrm{GMM}})$ is the number of parameters in G_{GMM}.

The AIC score is given by

$$\mathrm{AIC}(G_{\mathrm{GMM}}) = \log P(\mathbb{D}|G_{\mathrm{GMM}}, \boldsymbol{\theta}) - r(G_{\mathrm{GMM}}) \quad (8)$$

The AIC and BIC scores are used throughout this paper to select the optimal number of components in GMM. In addition, samples can be generated from a GMM model in two steps:

(1) Generate a discrete number (component index i, $i = 1, 2, \ldots, K$) based on the values of λ_i;
(2) Generate a sample from the distribution associated with the discrete number.

In Sec. 3, we will develop the method to integrate BN learning with GMMs for BNs with both discrete and continuous variables.

3 Learning of Bayesian Networks Using Gaussian Mixture Model

In this section, we first give a brief review of related work in engineering applications and methodologies. Following that, we will present the proposed new learning method.

3.1 A Brief Review of Related Work

3.1.1 Related Engineering Applications. During the past decades, learning BN based on data for Bayesian inference has been widely used in various engineering applications. For example, the air quality of Washington, DC and its relationship between the power plant emissions near the city has been modeled using BN based on data from observation stations [23]. The causality between the faults of a hydraulic actuator system and the system parameters needs to be mined from the operational data of the system [13]. Sahin et al. performed fault diagnosis for airplane engines using BN learned from data sets obtained from airplane engines during actual flights [34]. Yang and Lee [35] developed a diagnostics and prognostics method of semiconductor manufacturing systems using BN learned from data collected from sensors. Regarding the application of GMM in Bayesian inference,

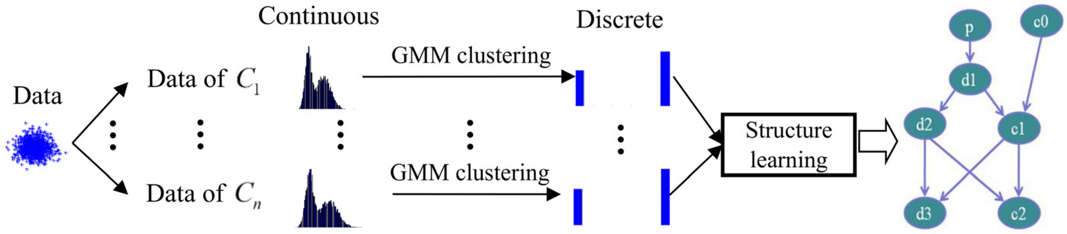


Fig. 1 Concept of the BNFINDER method

Rodriguez-Zas and Ko [36] applied the GMM to the inference of gene pathways, Sun et al. [37] constructed BN using GMM to predict traffic flow, and Zhang et al. [38] used hierarchical GMMs to perform probabilistic community discovery.

3.1.2 Related Methodologies and Software Packages. As discussed in the introduction, various methods have been investigated for the learning of BN, such as the software of Hugin which discretizes continuous variables into discrete variables in learning [22], the development of vine-copula models [23,39], and the employment of mixture of truncated exponential polynomials [25], and assumption of linear Gaussian CPDs [27]. Several methods have also been studied for the learning of BNs using GMMs during the past decades. For instance, Davies and Moore [40] developed a method called mix-nets, which uses a factored mixture of Gaussians for BN parameter learning with discrete and continuous variables, and Wilczynski and coworkers [26,32] developed Python packages called BNFINDER and BNFINDER2 for the hybrid BN structure learning by using GMMs to handle continuous variables. Among the above reviewed methods, most of them only focus on the CPD modeling of continuous variables. BNFINDER is the one which focuses on both structure learning and parameter learning. Since BNFINDER is also the one that is the most related to the method developed in this paper, we therefore briefly review this method first, before developing the proposed method.

3.1.3 BNfinder. Figure 1 gives the main principle of BNFINDER. Three kinds of score functions (MDL, BDe, and MIT) as reviewed in Sec. 2.2 are implemented in BNFINDER. All these score functions were originally developed for the learning of BNs with only discrete variables. To apply these score functions to structure learning with both discrete and continuous variables, BNFINDER treats the distribution of a continuous node C_i as a mixture of two Gaussian components without arbitrary discretization of the continuous variables. Based on this treatment, a hidden discrete variable W_i is introduced to represent the Gaussian component corresponding to a given value of C_i. The conditional distribution of C_i is computed by

$$P(C_i|\boldsymbol{\pi}) = \sum_{d\in\{\text{low, high}\}} \sum_{\mathbf{d}\in\{\text{low, high}\}|\boldsymbol{\pi}} P(C_i|W_i = d) \times P(W_i = d|\boldsymbol{\pi}' = \mathbf{d})P(\boldsymbol{\pi}' = \mathbf{d}|\boldsymbol{\pi}) \tag{9}$$

where d is a realization of the hidden discrete node W_i.

Analyses show that this treatment (Eq. (9)) does not work well for the BN learning of some problems for two reasons. First, only a two-component Gaussian mixture model is used in BNFINDER to represent a continuous node. In many practical situations, two components are not enough when variables follow multimodal distributions. Second, BNFINDER is developed for structure learning but not for model calibration. When it is applied to model calibration with continuous variables, the CPD modeling of a continuous variable will have large error due to the two-component univarite Gaussian mixture model used in Eq. (9). The fundamental reasons for the large errors are explained in Sec. 3.2.2.

Based on all the previously-mentioned observations, we develop a new method below for the learning of BNs with both discrete and continuous variables.

3.2 Proposed Bayesian Network Learning Framework. In this section, we first provide an overview of the proposed framework. Following that, details of the new methods are explained.

3.2.1 Overview of the Proposed Framework. The proposed framework consists of three elements: the OUGM, score functions for learning, and a new MGM. The OUGM model is first built for each continuous variable. Based on the OUGM, data for continuous variables are mapped into data for latent discrete variables using GMM clustering. Once the mapping of data is completed, we apply available score functions for BN learning with only discrete variables to learning of the BN topology and CPTs between discrete nodes. After that, we develop a new MGM model to improve the accuracy of CPD learning of continuous variables. Figure 2 shows the overall framework for BN learning with both discrete and continuous variables. There are two stages:

- *Stage 1*—Learning of the BN topology and CPTs (of discrete variables and latent discrete variables of continuous variables) using available learning algorithms for BNs with only discrete variables. The learning is based on the mapping of data for continuous variables to data for latent discrete variables through GMM clustering based on OUGM.
- *Stage 2*—Continuous nodes with parents are identified from the learned BN topology, and the joint probability density functions of continuous nodes and their parents are modeled using MGMs.

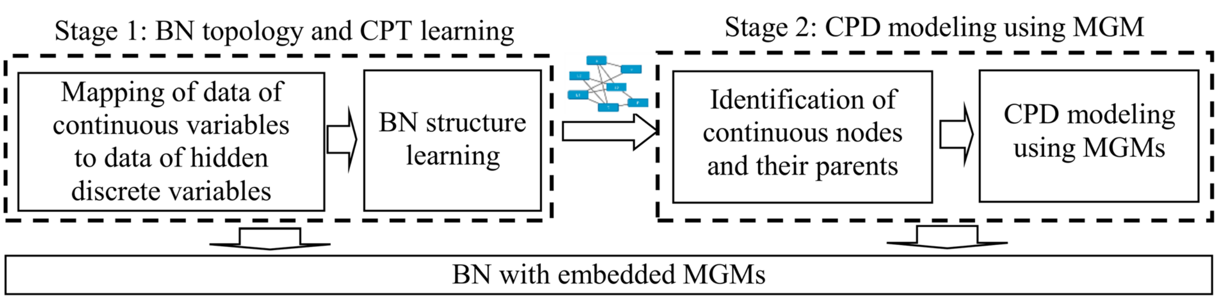


Fig. 2 Overview of the proposed method

Note that Stage 1 uses univariate Gaussian mixtures whereas Stage 2 uses multivariate Gaussian mixtures. The final learned BN will be a BN with MGMs embedded as CPD functions.

3.2.2 Optimal Univariate Gaussian Mixture for Structure Learning. For BN structure learning (topology), in this paper, we extend the current approach of using two Gaussian components to using an optimal number of components according to the actual distribution of available data. The optimal number is determined based on the BIC scores of the GMMs. For each continuous variable C_i, $i = 1, 2, \ldots, n$, its marginal distribution is approximated using a multiple-component univariate GMM as follows:

$$f(C_i) = \sum_{j=1}^{K_i} \lambda_j N(C_i|\mu_j, \sigma_j^2) \tag{10}$$

where K_i is the number of components for variable C_i, $N(C_i|\mu_j, \sigma_j^2)$ is the PDF of the jth Gaussian component, μ_j and σ_j are the mean and standard deviation of jth Gaussian component, and λ_j is the weight of jth component. After extending the two—component GMM to multiple number of Gaussian components, Eq. (9) becomes

$$P(C_i|\boldsymbol{\pi}) = \sum_{d\in\{1,2,\ldots,K_i\}} \sum_{\mathbf{d}\in\{\mathbf{K}\}|\boldsymbol{\pi}} P(C_i|W_i = d) \times P(W_i = d|\boldsymbol{\pi}' = \mathbf{d})P(\boldsymbol{\pi}' = \mathbf{d}|\boldsymbol{\pi}) \tag{11}$$

where $\mathbf{K}$ stands for all possible realizations of parents of C_i.

Based on this extension, we map the data for continuous variables into discrete data for hidden discrete variables W_i and then use score functions reviewed in Sec. 2.2 for the structure learning. From the learned structure, we then estimate the CPTs of discrete variables using the ML method. As indicated in the numerical examples in Sec. 4, this strategy is adequate for the structure (topology) and CPT learning. However, the use of univariate Gaussian mixtures (with either two components or an optimal number of components) may have large error in conditional probability learning. The reason is explained as follows.

Based on Eq. (11), the CPD of a continuous variable, C_i is written as

$$P(C_i|\boldsymbol{\pi}) = \sum_{d\in\{1,2,\ldots,K_i\}} \lambda(d)P(C_i|W_i = d) \tag{12}$$

where $\lambda(d) = \sum_{\mathbf{d}\in\{1,2,\ldots,K_i\}|\boldsymbol{\pi}} P(W_i = d|\boldsymbol{\pi}' = \mathbf{d})P(\boldsymbol{\pi}' = \mathbf{d}|\boldsymbol{\pi})$.

The mean $\mu_{C_i|\boldsymbol{\pi}}$ and variance $\sigma^2_{C_i|\boldsymbol{\pi}}$ of $C_i|\boldsymbol{\pi}$ are given by

$$\mu_{C_i|\boldsymbol{\pi}} = \int (c_i \sum_{d\in\{1,2,\ldots,K_i\}} \lambda(d)P(C_i|W_i = d))dc_i = \sum_{d\in\{1,2,\ldots,K_i\}} \lambda(d)\mu(d) \tag{13}$$

$$\sigma^2_{C_i|\boldsymbol{\pi}} = \int (c_i^2 \sum_{d\in\{1,2,\ldots,K_i\}} \lambda(d)P(C_i|W = d))dc_i - \mu^2_{C_i|\boldsymbol{\pi}} = \sum_{d\in\{1,2,\ldots,K_i\}} \lambda(d)(\sigma^2(d) + \mu^2(d)) - \mu^2_{C_i|\boldsymbol{\pi}} \tag{14}$$

where $\mu(d)$ and $\sigma(d)$ are the mean and standard deviation of component d, respectively.

Equations (13) and (14) show that $\mu_{C_i|\boldsymbol{\pi}}$ is bounded by the means $\mu(d)$ of Gaussian components of C_i and $\sigma^2_{C_i|\boldsymbol{\pi}}$ is also bounded by the variances of the Gaussian components. No matter how the parents $\boldsymbol{\pi}$ change, the mean and variance of the conditional distribution will not go beyond those bounds. In Bayesian inference, however, it is quite often that the mean and variance of CPD fall outside of those bounds. This implies that there may be a large error in the conditional probability by treating it as in Eq. (11). Note that the original BNFINDER (reviewed in Sec. 3.1.1) using two-component Gaussian mixture model is a special case of the OUGM. The above-mentioned limitation therefore exists in both BNFINDER and OUGM, i.e., in the use of univariate Gaussian mixtures. To improve the accuracy of CPD learning and make the learned BN applicable for model calibration, we introduce the multivariate Gaussian mixture approach in Sec. 3.2.3.

3.2.3 Multivariate Gaussian Mixture for Conditional Probability Density Learning. Multivariate GMMs are developed in this section for the purpose of CPD modeling of continuous nodes. They are used to model the joint PDF of multiple variables instead of marginal distributions of individual variables. The GMM for n continuous variables, $\mathbf{C} = [C_1, C_2, \ldots, C_n]$, $i = 1, 2, \ldots, n$, is given by

$$f(\mathbf{C}) = \sum_{j=1}^{K_m} \lambda_j N(\mathbf{C}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \tag{15}$$

where K_m is the number of components in the multivariate GMM, $N(\mathbf{C}|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)$ is the PDF of the jth multivariate normal distribution, $\boldsymbol{\mu}_j$ and $\boldsymbol{\Sigma}_j$ are the mean and covariance matrix of the jth multivariate normal distribution, and λ_j is the weight of the jth multivariate normal distribution. Based on the MGM, as indicated in Eqs. (17)–(19), the conditional probability distributions will change with the values of parents instead of fixing at several given distributions presented in Eqs. (13) and (14). Thus, the accuracy of CPD modeling is improved.

The application of multivariate GMMs to the modeling of CPDs is not straightforward. In this section, we will study how the CPDs can be approximated using multivariate GMMs. This is the critical part of stage 2 in the proposed framework. There are two cases of CPD modeling using multivariate GMMs. We discuss these two cases separately.

3.2.3.1 Case 1—Continuous node with only continuous parents. When both the parent and child nodes are continuous variables, the CPD modeling using GMMs has been studied and reported in the literature [37]. For a given continuous variable C_i with continuous parents, $\mathbf{C}_{\mathrm{Pa}}$, the CPD of C_i is given by

$$f(C_i|\mathbf{C}_{\mathrm{Pa}}) = f(C_i, \mathbf{C}_{\mathrm{Pa}})/f(\mathbf{C}_{\mathrm{Pa}}) = f(\mathbf{C})/f(\mathbf{C}_{\mathrm{Pa}}) \tag{16}$$

where $f(\mathbf{C}_{\mathrm{Pa}})$ stands for the joint PDF of continuous parent nodes and $f(\mathbf{C})$ is the joint PDF of C_i and its continuous parents. If $f(\mathbf{C})$ is modeled using a multivariate GMM as given in Eq. (15), then the CPD of C_i is also a GMM given by

$$f(C_i|\mathbf{C}_{\mathrm{Pa}}) = \sum_{j=1}^{K_m} w_j N(\mu_{j,C_i|\mathbf{C}_{\mathrm{Pa}}}, \boldsymbol{\Sigma}_{j,C_i|\mathbf{C}_{\mathrm{Pa}}}) \tag{17}$$

where $\mu_{j,C_i|\mathbf{C}_{\mathrm{Pa}}}$ and $\boldsymbol{\Sigma}_{j,C_i|\mathbf{C}_{\mathrm{Pa}}}$ are the conditional mean and variance, respectively, and

$$\mu_{j,C_i|\mathbf{C}_{\mathrm{Pa}}} = \mu_{j,C_i} + \boldsymbol{\Sigma}_{j,C_i\mathbf{C}_{\mathrm{Pa}}}\boldsymbol{\Sigma}^{-1}_{j,\mathbf{C}_{\mathrm{Pa}}}(\mathbf{C}_{\mathrm{Pa}} - \boldsymbol{\mu}_{j,\mathbf{C}_{\mathrm{Pa}}}) \tag{18}$$

$$\boldsymbol{\Sigma}_{j,C_i|\mathbf{C}_{\mathrm{Pa}}} = \boldsymbol{\Sigma}_{j,C_iC_i} - \boldsymbol{\Sigma}_{j,C_i\mathbf{C}_{\mathrm{Pa}}}\boldsymbol{\Sigma}^{-1}_{j,\mathbf{C}_{\mathrm{Pa}}}\boldsymbol{\Sigma}^{T}_{j,C_i\mathbf{C}_{\mathrm{Pa}}} \tag{19}$$

$$w_j = \lambda_j f_j(\mathbf{C}_{\mathrm{Pa}}) \Big/ \sum_{j=1}^{K_m} \lambda_j f_j(\mathbf{C}_{\mathrm{Pa}}) \tag{20}$$

In Eq. (17), w_j can also be treated as the posterior probability of weights for $\mathbf{C}_{\mathrm{Pa}}$.

3.2.3.2 Case 2—Continuous node with both discrete and continuous parents. If a continuous node has both discrete and

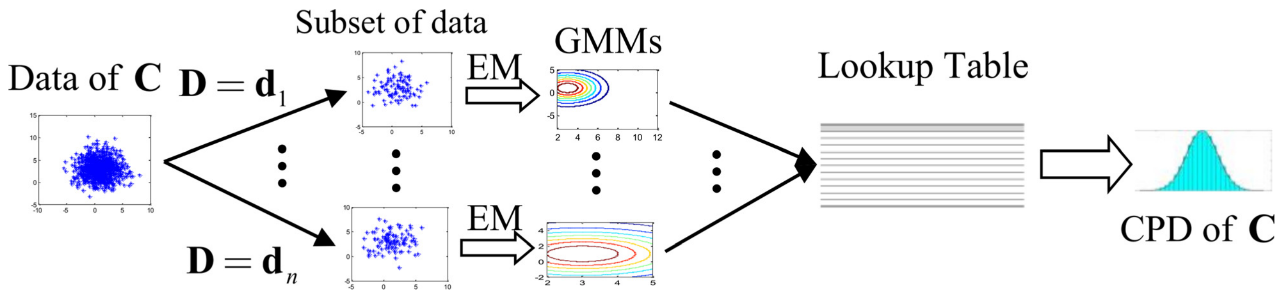


Fig. 3 Concept of the mix-nets method

continuous nodes, the GMM method in case 1 is not applicable anymore. A possible way of solving this problem is using the mix-nets method [40]. Figure 3 shows the main principle of the mix-nets method. The main idea of handling discrete variables in mix-nets is to use a lookup table. Suppose a set of nodes $\mathbf{V}$ includes both discrete nodes $\mathbf{D} = [D_i]_{i=1,2,\ldots,n}$ and continuous nodes $\mathbf{C} = [C_j]_{j=1,2,\ldots,m}$. For each possible realization $\mathbf{d}$ of $\mathbf{D}$, there are two elements in the lookup table: the marginal probability $P(\mathbf{d})$ of $\mathbf{d}$, and a Gaussian mixture model, $P(\mathbf{C}|\mathbf{d})$, conditioned on $\mathbf{d}$. The Gaussian mixture model, $P(\mathbf{C}|\mathbf{d})$, is learned using the EM algorithm based on a subset of data of $\mathbf{C}$ associated with the values of $\mathbf{d}$. Based on the lookup table, algorithms [40] are developed to compute the following conditional probability:

$$P(V_i|\boldsymbol{\pi}) = P(\boldsymbol{\pi}|V_i)/P(\boldsymbol{\pi}) \tag{21}$$

where V_i is a node in $\mathbf{V}$ and $\boldsymbol{\pi}$ is a vector of its parents which may have both discrete and continuous nodes.

It can be found that the data set of continuous nodes is partitioned in the mix-nets method based on the values of discrete nodes. When the data are abundant, the mix-nets method works well. When the number of discrete nodes or possible values of the discrete node increases, as being pointed out [40], the data of continuous variables will be shattered into many separate GMMs and the data support for each of the GMMs will be very little, thus affecting the accuracy. Inspired by the latent variable approach developed by Morlini [41] for data clustering of mixed binary and continuous variables, a new MGM method is developed in this section to overcome this challenge.

In the latent variable approach, a continuous latent variable with threshold is introduced to represent binary variables. The thresholds and correlations of the latent continuous variables are determined according to the statistical properties of the binary variables. Suppose the following CPD needs to be evaluated, where $\mathbf{C}_{\mathrm{Pa}}$ are continuous parents and $\mathbf{D}_{\mathrm{Pa}} = [D_1, D_2, \ldots, D_m]$ are discrete parents

$$P(C_i|\mathbf{C}_{\mathrm{Pa}} = \mathbf{c}_{\mathrm{Pa}}, \mathbf{D}_{\mathrm{Pa}} = \mathbf{d}_{\mathrm{Pa}}) = P(C_i, \mathbf{c}_{\mathrm{Pa}}, \mathbf{d}_{\mathrm{Pa}})/P(\mathbf{c}_{\mathrm{Pa}}, \mathbf{d}_{\mathrm{Pa}}) \tag{22}$$

where C_i is the child node, $P(C_i, \mathbf{c}_{\mathrm{Pa}}, \mathbf{d}_{\mathrm{Pa}})$ is the joint probability density function of C_i, $\mathbf{c}_{\mathrm{Pa}}$, and $\mathbf{d}_{\mathrm{Pa}}$.

In the threshold method presented by Morlini [41], a latent variable is introduced for each binary variable. For a given binary variable D_i, its associated latent variable X_i (Gaussian variable) has the following property:

$$D_i = \begin{cases} 1, \text{ if } X_i \geq \xi_i \\ 0, \text{ otherwise} \end{cases} \tag{23}$$

in which ξ_i is a threshold determined according to the marginal probability of D_i.

In order to account for the correlations among $D_1, D_2, \ldots,$ and D_m, the covariance matrix $\boldsymbol{\Sigma}$ of all the latent variables X_i, $i = 1, 2, \ldots, m$ is computed based on the following probability equivalency:

$$P(\mathbf{D}_{\mathrm{Pa}}) = \int \ldots \int \left(1/\sqrt{(2\pi)^m|\boldsymbol{\Sigma}|}\right)\exp(-(\mathbf{x}-\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\mathbf{x}-\boldsymbol{\mu})/2)\mathbf{dx} \tag{24}$$

where $P(\mathbf{D}_{\mathrm{Pa}})$ is the joint probability of discrete variables obtained from data.

Based on Eq. (24), the covariance matrix of the GMMs of all the continuous nodes and latent variables is estimated in Morlini's method [41] through the principle of *maximum a posteriori estimation*. It is found that this method has the following drawbacks:

(1) The number of latent variables and the dimensionality of integration are high.
(2) Solving the high-dimensional covariance matrix based on the joint probability equivalency is computationally intensive and often has singularity problems.
(3) Analytically estimating the global optimal covariance matrix for all continuous variables and latent variables is very difficult.

To overcome the above-mentioned difficulties, we develop a new method by using only one latent variable U to represent all the discrete parents of a continuous node. Before discussing the proposed method, we define a new discrete variable D_a with the following characteristics

$$P(D_a) = P(\mathbf{D}_{\mathrm{Pa}}) \tag{25}$$

Suppose each D_i has n_i states, D_a is a discrete variable with the number of states given by

$$N = \prod_{i=1}^{m} n_i \tag{26}$$

Based on the definition of D_a, we have

$$\begin{aligned} P(C_i|\mathbf{C}_{\mathrm{Pa}} = \mathbf{c}_{\mathrm{Pa}}, \mathbf{D}_{\mathrm{Pa}} = \mathbf{d}_{\mathrm{Pa}}) &= P(C_i|\mathbf{C}_{\mathrm{Pa}} = \mathbf{c}_{\mathrm{Pa}}, D_a = d_a) \\ &= P(C_i, \mathbf{c}_{\mathrm{Pa}}, d_a)/P(\mathbf{c}_{\mathrm{Pa}}, d_a) \end{aligned} \tag{27}$$

and

$$P(\mathbf{D}_{\mathrm{Pa}} = \mathbf{d}_{\mathrm{Pa}}) = P(D_a = d_a^i) = P_i, \; i = 1, 2, \ldots, N \tag{28}$$

In order to model the CPD of C_i with parents $\mathbf{C}_{\mathrm{Pa}}$ and D_a using GMM, thresholds of the latent variable U are defined as follows:

$$D_a = \begin{cases} d_a^1, \text{ if } u \leq u_1 \\ d_a^i \text{ if } u_{i-1} < u \leq u_i, \forall\, 1 < i < N \\ d_a^N \text{ if } u_{N-1} \leq u \end{cases} \tag{29}$$

Table 1 Pseudo code for converting discrete data into continuous data

Step	Description
1	Identify all possible states d_a^i, $i = 1, 2, \ldots, N$, of D_a, the probability, $P(D_a = d_i)$, and indices $\mathbf{In}_i$ of each state d_a^i in the data set
2	Define an initial sampling matrix $\mathbf{s}_{\text{new}} = 0_{(n_{it}\times n_s)\times(1+n_c)}$, where n_{it} is a parameter used to repeat the data of continuous variables to account for the fact that each data of $d_a^{(i)}$ corresponding to an interval $[u_L,\ u_U]$.
3	**For** $i = 1{:}N$
4	Generate a sampling matrix $\mathbf{u}_{\text{temp}}$ with dimension of $n_{it} \times \text{length}(\mathbf{In}_i)$ using the Latin Hypercube sampling (LHS) approach.
5	Convert $\mathbf{u}_{\text{temp}}$ into the truncated standard normal probability interval according to $P(D_a = d_a^i)$.
6	Convert $\bar{\mathbf{u}}_{\text{temp}}$ into data of continuous variable, $\mathbf{u}$, in the standard normal space using the inverse CDF function
7	**For** $it=1{:}n_{it}$
8	$\mathbf{s}_{\text{new}}((it-1)n_s+1 : it\times n_s,\ 2:\text{end}) = \mathbf{s}(:,\ 2:\text{end})$
9	$\mathbf{s}_{\text{new}}((it-1)n_s+\mathbf{I}_i,\ 1) = \mathbf{u}(it,\ :)^T$
10	**End**
11	**End**

in which u_i, $i = 1, 2, \ldots, N-1$ are thresholds computed by

$$u_i = \begin{cases} \Phi^{-1}(P_i), \text{ if } i = 1 \\ \Phi^{-1}\left(\sum_{j=1}^{i} P_j\right), \text{ otherwise} \end{cases} \forall i = 1, 2, \ldots, N-1 \tag{30}$$

Note that in this paper, the standard normal variable is used as the latent variable U. Other variables such as Lognormal, Uniform, or Weibull can be used as the latent variable as well.

After introducing the latent variable into Eq. (27), we have

$$P(C_i|\mathbf{C}_{\text{Pa}} = \mathbf{c}_{\text{Pa}},\ D_a = d_a^i) = \begin{cases} \int_{-\infty}^{u_1} P(C_i|\mathbf{C}_{\text{Pa}} = \mathbf{c}_{\text{Pa}},\ u) P(U = u|D_a = d_a^1) du \\ \int_{u_{i-1}}^{u_i} P(C_i|\mathbf{C}_{\text{Pa}} = \mathbf{c}_{\text{Pa}},\ u) P(U = u|D_a = d_a^i) du, \text{ if } 1 < i < N \\ \int_{u_i}^{\infty} P(C_i|\mathbf{C}_{\text{Pa}} = \mathbf{c}_{\text{Pa}},\ u) P(U = u|D_a = d_a^N) du \end{cases} \tag{31}$$

where $P(U = u|D_a = d_a^i)$ is the PDF of U at u given $D_a = d_a^i$.

For a given $D_a = d_a^i$, the corresponding distribution of U is a truncated normal distribution. We therefore have

$$P(U = u|D_a = d_a^i) = \phi(u)/P(D_a = d_a^i) \tag{32}$$

where $\phi(u)$ is the PDF of the standard normal variable.

In Eq. (31), $P(C_i|\mathbf{C}_{\text{Pa}} = \mathbf{c}_{\text{Pa}},\ u)$ is a continuous CPD given by

$$P(C_i|\mathbf{C}_{\text{Pa}} = \mathbf{c}_{\text{Pa}},\ u) = P(C_i,\ \mathbf{c}_{\text{Pa}},\ u)/P(\mathbf{c}_{\text{Pa}},\ u) \tag{33}$$

Suppose that the joint PDF of C_i, $\mathbf{C}_{\text{Pa}}$, and U is modeled using the multivariate GMM. We then have

$$P(C_i|\mathbf{C}_{\text{Pa}} = \mathbf{c}_{\text{Pa}},\ u) = \sum_{j=1}^{K_m} w_j N(\boldsymbol{\mu}_{j,\ C_i|\mathbf{c}_{\text{Pa}},u},\ \boldsymbol{\Sigma}_{j,\ C_i|\mathbf{c}_{\text{Pa}},u}) \tag{34}$$

$$\mu_{j,\ C_i|\mathbf{c}_{\text{Pa}},u} = \mu_{j,C_i} + \boldsymbol{\Sigma}_{j,C_i\mathbf{c}_{\text{Pa}},u}\boldsymbol{\Sigma}^{-1}_{j,\mathbf{c}_{\text{Pa}},u}([\mathbf{c}_{\text{Pa}},\ u] - \boldsymbol{\mu}_{j,\mathbf{c}_{\text{Pa}},u}) \tag{35}$$

$$\boldsymbol{\Sigma}_{j,\ C_i|\mathbf{C}_{\text{Pa}}} = \boldsymbol{\Sigma}_{j,C_iC_i} - \boldsymbol{\Sigma}_{j,C_i\mathbf{C}_{\text{Pa}}}\boldsymbol{\Sigma}^{-1}_{j,\mathbf{C}_{\text{Pa}}}\boldsymbol{\Sigma}^{T}_{j,C_i\mathbf{C}_{\text{Pa}}} \tag{36}$$

and

$$w_j = \lambda_j f_j(\mathbf{c}_{\text{Pa}},\ u) \bigg/ \left(\sum_{j=1}^{K_m} \lambda_j f_j(\mathbf{c}_{\text{Pa}},\ u)\right) \tag{37}$$

Substituting Eqs. (32)–(37) into Eq. (31), we have

$$P\left(C_i|\mathbf{C}_{\text{Pa}} = \mathbf{c}_{\text{Pa}},\ D_a = d_a^i\right) = \int_{u_L}^{u_U} \frac{\sum_{j=1}^{K_m} \lambda_j f_j(\mathbf{c}_{\text{Pa}},\ u) N\left(\boldsymbol{\mu}_{j,\ C_i|\mathbf{c}_{\text{Pa}},u},\ \boldsymbol{\Sigma}_{j,\ C_i|\mathbf{c}_{\text{Pa}},u}\right)\phi(u)}{P\left(D_a = d_a^i\right)\left(\sum_{j=1}^{K_m} \lambda_j f_j(\mathbf{c}_{\text{Pa}},\ u)\right)} du \tag{38}$$

where

$$[u_L,\ u_U] = \begin{cases} [-\infty,\ u_1], \text{ if } i = 1 \\ [u_{i-1},\ u_i], \text{ if } 1 < i < N \\ [u_{N-1},\ \infty], \text{ if } i = N \end{cases} \tag{39}$$

in which u_i, $i = 1, 2, \ldots, N-1$ are obtained from Eq. (30).

3.2.3.3 Constructing multivariate Gaussian mixture for discrete and continuous data. The earlier-mentioned derivations are based on the assumption that we know the multivariate GMMs for the joint PDF of C_i, $\mathbf{C}_{\text{Pa}}$, and U. In the CPD learning, however, we need to construct this MGM with available data. Suppose that n_s observation samples have been collected for BN learning of a problem with m discrete nodes and n_c continuous nodes, we denote the sample matrix as

$$\mathbf{s} = \begin{bmatrix} d_1^{(1)} & \ldots & d_m^{(1)} & \mathbf{c}^{(1)} \\ d_1^{(2)} & \ldots & d_m^{(2)} & \mathbf{c}^{(2)} \\ \vdots & \ddots & \vdots & \vdots \\ d_1^{(n_s)} & \ldots & d_m^{(n_s)} & \mathbf{c}^{(n_s)} \end{bmatrix}_{n_s\times(m+n_c)} \tag{40}$$

where $d_i^{(j)}$ is the jth sample of ith variable d_i.

Based on the definition in Eq. (25), the sample matrix is rewritten as

$$\mathbf{s} = \begin{bmatrix} d_a^{(1)} & d_a^{(2)} & \ldots & d_a^{(n_s)} \\ \mathbf{c}^{(1)} & \mathbf{c}^{(2)} & \ldots & \mathbf{c}^{(n_s)} \end{bmatrix}^T_{(1+n_c)\times n_s} \tag{41}$$

Since we need to learn the joint PDF of C_i, $\mathbf{C}_{Pa}$, and U using the MGM, the data of d_a in Eq. (41) need to be converted into data of U. Any given data, $d_a^{(i)}$, $i = 1, 2, \ldots, n_s$ corresponds to an interval $[u_L,\ u_U]$ given in Eq. (39). In order to preserve this property and maintain the correlation between D_a and all other continuous variables as given in Eq. (40), we present an algorithm in Table 1.

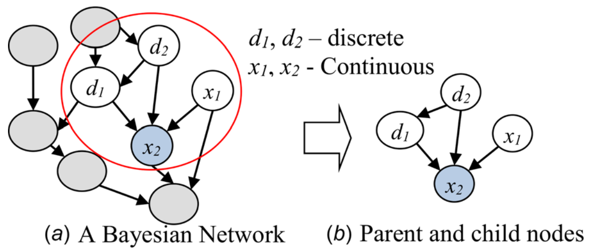


Fig. 4 A continuous node with discrete and continuous parents

Table 2 CPT and CPD of nodes

Node		CPT and CPD		
d_1		$d_1 = 1$ 0.1	$d_1 = 2$ 0.1	$d_1 = 3$ 0.8
d_2	$d_1 = 1$	$d_2 = 1$ 0.2	$d_2 = 2$ 0.7	$d_2 = 3$ 0.1
	$d_1 = 2$	$d_2 = 1$ 0.1	$d_2 = 2$ 0.1	$d_2 = 3$ 0.8
	$d_1 = 3$	$d_2 = 1$ 0.3	$d_2 = 2$ 0.1	$d_2 = 3$ 0.6
x_1			$N(15, 5^2)$	
x_2			$N((x_1 + 5)^2 + 100d_1 + 50d_2, 40^2)$	

Based on the algorithm in Table 1, we convert data of mixed discrete and continuous variable given in Eq. (41) into continuous data. With the continuous data, an MGM is constructed for $P(C_i, \mathbf{C_{Pa}}, U)$ using the EM method reviewed in Sec. 2.3. Once the MGM is available, the CPD of continuous nodes are obtained using the method presented in Sec. 3.2.3.2.

The proposed method overcomes the aforementioned three drawbacks of Morlini's method [41] since only one latent variable U is used to represent the discrete variables. Due to the introduction of only one latent variable U and a new discrete variable D_a, we do not need to solve the high-dimensional covariance matrix and find the optimal covariance matrix. The correlations between the different discrete variables are automatically accounted for in the new discrete variable D_a. An example is given below to illustrate the effectiveness of the proposed CPD modeling for continuous nodes with both discrete and continuous parents. As indicated in Fig. 4, a continuous node x_2 and its parents (Fig. 4(*b*)) are identified from the overall BN (Fig. 4(*a*)). The exact CPTs and CPDs of x_2 and its parents are given in Table 2. The CPD of x_2 is learned using the proposed method and the mix-nets method based on 500 collected samples. Figure 5 shows the comparison between the learned CPDs of x_2 from different methods.

The results show that for this example, overall the learned CPDs from the proposed method are better than their counterparts from the mix-nets approach. For some situations, the mix-nets approach fails to learn the CPDs (i.e., $(d_1, d_2) = (1, 3)$ and $(d_1, d_2) = (2, 2)$). The proposed method, however, always works well for all kinds of scenarios.

3.3 Summary of Bayesian Network Learning Scenarios.

In this section, we summarize the CPT and CPD learning of different *parent to child* scenarios.

(a) *Discrete to discrete*: Discrete CPTs $P(d|\mathbf{D}_{Pa} = \mathbf{d}_{Pa})$ are learned using the OUGM (Sec. 3.2.2) based on the score functions given in Sec. 2.2. Here, OUGM means the optimal univariate Gaussian mixture model instead of two – component GMM.

(b) *Continuous to discrete*: For this scenario also, we use the method presented in OUGM. The CPT for $P(d|\mathbf{C}_{\mathrm{Pa}} = \mathbf{c}_{\mathrm{Pa}})$ is expressed as

$$P(d|\mathbf{C}_{\mathrm{Pa}} = \mathbf{c}_{\mathrm{Pa}}) = \sum_{\mathbf{d} \in \{\mathbf{K}\}} P(d|\boldsymbol{\pi}' = \mathbf{d}) P(\boldsymbol{\pi}' = \mathbf{d}|\mathbf{c}_{\mathrm{Pa}}) \tag{42}$$

(c) *Mixed discrete and continuous to discrete*: This scenario is a combination of scenarios (a) and (b). The CPT, $P(d|\mathbf{D}_{\mathrm{Pa}} = \mathbf{d}_{\mathrm{Pa}}, \mathbf{C}_{\mathrm{Pa}} = \mathbf{c}_{\mathrm{Pa}})$ is modeled as

$$P(d|\mathbf{D}_{\mathrm{Pa}} = \mathbf{d}_{\mathrm{Pa}}, \mathbf{C}_{\mathrm{Pa}} = \mathbf{c}_{\mathrm{Pa}}) = \sum_{\mathbf{d} \in \{\mathbf{K}\}} P(d|\boldsymbol{\pi}' = \mathbf{d}) P(\boldsymbol{\pi}' = \mathbf{d}|\mathbf{c}_{\mathrm{Pa}}, \mathbf{d}_{\mathrm{Pa}}) \tag{43}$$

(d) *Continuous to continuous*: The CPDs are modeled using the GMM method presented in Sec. 3.2.3.1.

(e) *Mixed discrete and continuous to continuous*: The CPDs are modeled using the new method developed in Secs. 3.2.3.2 and 3.2.3.3.

Once the BN topology, CPTs, and CPDs are learned, model calibration can be performed based on the learned BN when new observation data are available.

3.4 Implementation Procedure.

Table 3 summarizes the overall implementation procedure of the proposed framework for learning BNs with both discrete and continuous variables.

4 Illustrative Examples

In this section, a comprehensive mathematical example and an engineering example are used to illustrate the proposed BN

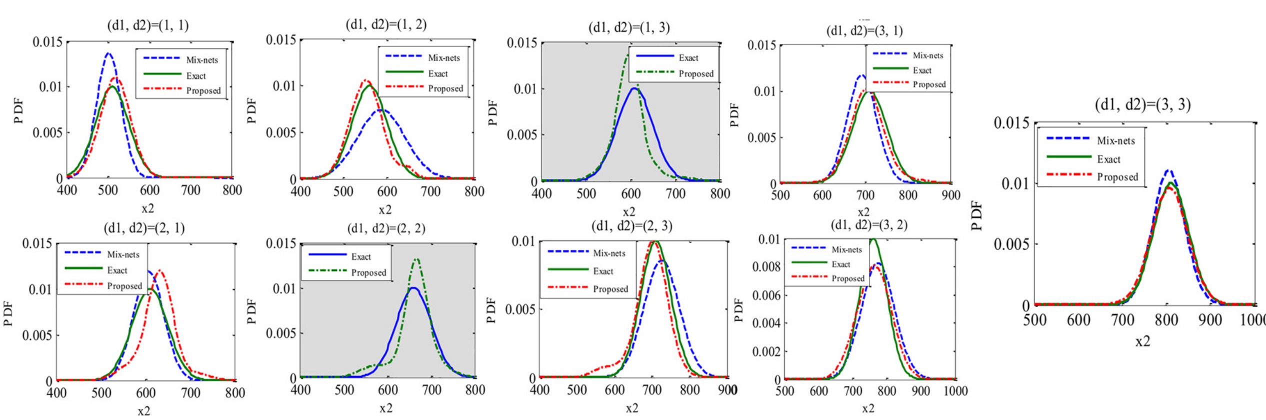


Fig. 5 Results comparison of learned CPDs, given $x_1 = 14$ (Note: Gray background implies mix-nets failed to model the CPD)

Table 3 Summary of BN learning implementation procedure

Step	Description
Data organization	
1	Map the data on continuous variables into data of hidden discrete variables based on the clustering of optimal univariate GMMs (Sec. 3.2.2).
Learning of topology and CPTs	
2	Perform BN structure learning using score functions which are originally defined for discrete BNs. (Sec. 2.2).
3	Obtain the BN structure topology and CPTs of discrete variables from the learned BN structure
4	Identify continuous variables and their parents
Learning of CPDs	
5	Construct multivariate GMM (Sec. 3.2.3.1) based on original data for the continuous variables with continuous parents identified in Step 4.
6	Construct multivariate GMM (Sec. 3.2.3.3) for continuous variables with mixed continuous and discrete parents identified in Step 4.
7	Model the CPDs of continuous variables with mixed continuous and discrete parents (Sec. 3.2.3.2).
8	Obtain the learned BN with embedded MGMs for further analyses

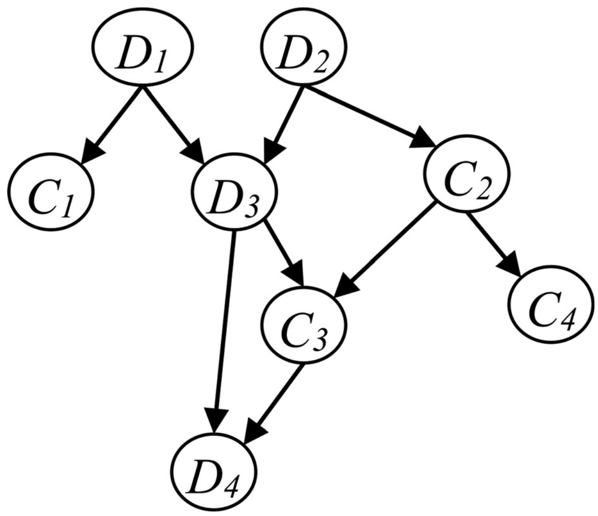


Fig. 6 Bayesian network of Example 1

Table 4 CPT of D_1

Value of D_1	$D_1 = 0$	$D_1 = 1$
Probability	0.3	0.7

Table 5 CPT of D_2

Value of D_2	$D_2 = 0$	$D_2 = 1$	$D_2 = 2$
Probability	0.6	0.3	0.1

learning methodology (OUGM and MGM). Model calibration is also investigated with the learned BN.

4.1 A Mathematical Example

4.1.1 Problem Statement. A BN as shown in Fig. 6 is used as our first example. In this BN, there are four discrete nodes (D_1, D_2, D_3, D_4) and four continuous nodes (C_1, C_2, C_3, C_4). All possible scenarios of parent–child combinations between discrete and continuous nodes are included in this BN. The exact CPTs and CPDs of this example are given in Tables 4–8 and Eqs. (44)–(46).

The CPDs of C_1, C_2, and C_4 are assumed to be as follows:

$$C_1|D_1 \sim N(10 + 4D_1,\ 2) \tag{44}$$

Table 6 CPT of D_3

Value of (D_1, D_2)		(0, 0)	(0, 1)	(0, 2)	(1, 0)	(1, 1)	(1, 2)
Probability	$D_3 = 0$	0.1	0.3	0.4	0.6	0.8	0.9
	$D_3 = 1$	0.9	0.7	0.6	0.4	0.2	0.1

Table 7 CPD of C_3

Value of D_3	$D_3 = 0$	$D_3 = 1$
$C_3\|C_2, D_3$	$N(0.15C_2^2,\ 2)$	$N(1.5C_2,\ 1)$

$$C_2|D_2 \sim N(6 + 2D_2^2,\ 1) \tag{45}$$

$$C_4|C_2 \sim N(0.1C_2^2 + 0.6C_2 + 1,\ 2) \tag{46}$$

4.1.2 Bayesian Network Structure Learning. We assume that we do not have any prior information about the topology, CPTs, or CPDs of the BN. We learn all of the above information purely based on data. There are many possible structures for the BN. Following the procedure given in Table 3, we first map the data for each continuous variable into data for corresponding hidden discrete variables using two-component GMM and optimal univariate GMM, respectively. In OUGM, the optimal number of Gaussian mixture used for variables C_1, C_2, C_3, and C_4 are all three with an optimization interval of three to ten. Based on that, we performed structure learning using score functions reviewed in Sec. 2.2. Figures 7 and 8 give the learned BN topologies with different numbers of samples from the two-component GMM (i.e., BNFINDER) and OUGM, respectively.

Figures 7 and 8 show that both the structure learning algorithms based on two-component GMM and OUGM successfully recovered most of connections in the network. For this example, the learned BN structures from OUGM with 1000–3000 samples are exactly the same as the true BN structure given in Sec. 4.1.1. The learning based on only two-component GMM cannot exactly learn the true structure (Connection $C_3 \rightarrow D_4$ cannot be learned). It implies that the effectiveness of structure learning has been improved by extending the two-component GMM to OUGM. Besides, with more and more samples available, some additional spurious connections ($D_2 \rightarrow C_3$, $D_2 \rightarrow C_4$) are learned for both BNFINDER and OUGM, which are different from the true structure. The reason for this phenomenon is that there are some weak correlations between D_2 and C_3, and between D_2 and C_4, over fitting occurs as the sample size becomes large.

The previously-mentioned study is to investigate the effectiveness of the BN structure learning algorithm using OUGM to handle continuous variables. In the following study, the CPD learning

Table 8 CPT of D_4

Value of (D_3, C_3)		$D_3 = 0$, $C_3 < 9$	$D_3 = 0$, $9 \le C_3 < 11$	$D_3 = 0$, $C_3 \ge 11$	$D_3 = 1$, $C_3 < 9$	$D_3 = 1$, $9 \le C_3 < 11$	$D_3 = 1$, $C_3 \ge 11$
Probability	$D_4 = 0$	0.4	0.3	0.6	0.4	0.1	0.3
	$D_4 = 1$	0.6	0.7	0.4	0.6	0.9	0.7

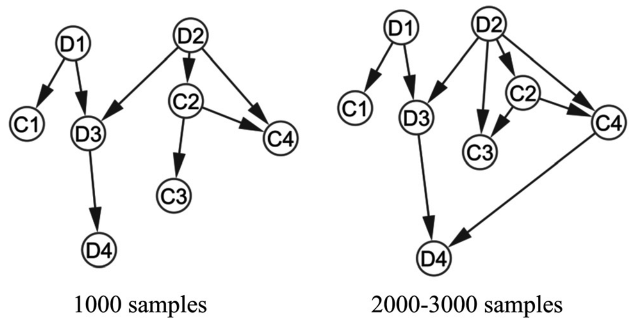


Fig. 7 Learned BN topology from BNFINDER with different numbers of samples

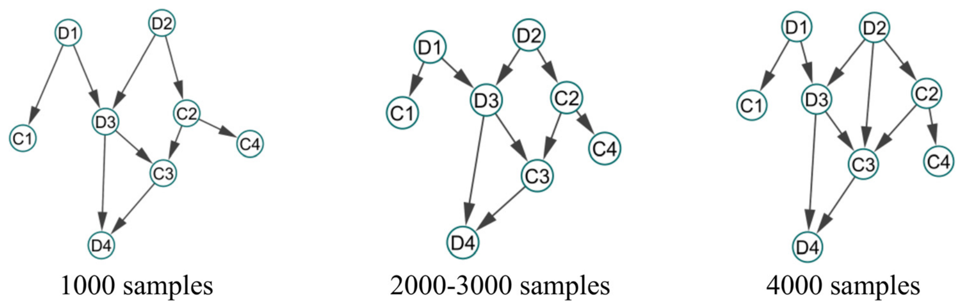


Fig. 8 Learned BN topology from OUGM with different numbers of samples

is based on the topology learning results obtained from 3000 data points using OUGM. The parents of C_1, C_2, C_3, C_4 are first identified based on the learned BN topology. The CPDs are learned next using the MGM method proposed in Sec. 3.2.3.

4.1.3 Conditional Probability Densities Learning. In order to verify the effectiveness of the learned CPDs, we give realizations of child nodes and investigate the variations in the outputs. Figure 9 shows the CPDs of C_3 obtained from different methods by fixing its parent nodes at different values. The results indicate that the proposed MGM method performs much better than the method just using univariate GMM (OUGM and BNFINDER) in the CPD learning. Here, only the OUGM is used for comparison because BNFINDER is just a special case of OUGM with two Gaussian components.

4.1.4 Model Calibration Based on the Learned Bayesian Network. We also perform model calibration based on the learned BN. Assume that we have some observations collected for C_3; we calibrate the values of the parent nodes, D_3 and C_2, based on the observations. Table 9 and Fig. 10 give the calibration results from different methods.

Table 9 and Fig. 10 show that the learned BN from the proposed method can be effectively applied to both CPD learning and later model calibration.

4.2 A Cantilever Beam With Possible Crack and Support Damage

4.2.1 Problem Statement. A cantilever beam as shown in Fig. 11 is employed as our second example. This example is modified from Ref. [42]. The exact BN of this problem is given in Fig. 11(*b*). There are two discrete nodes (C and D) and five continuous nodes (A,B_1,B_2,P,F).

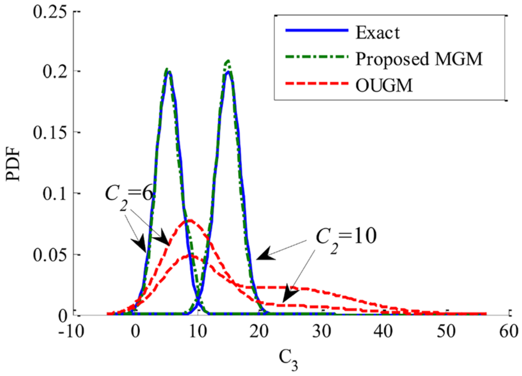
C is a crack indicator. $C = 1$, if there is a crack, otherwise, $C = 0$. D is the bolt damage indicator. $D = 1$, if the bolt is damaged, otherwise $D = 0$. A is the crack length. P is the applied load. B_1 and B_2 are two parameters (obtained from finite element simulations) relating the deflection, F, of the beam to the load P. Tables 10 shows the true CPTs of C and D.

If there is a crack, suppose the crack length is given by

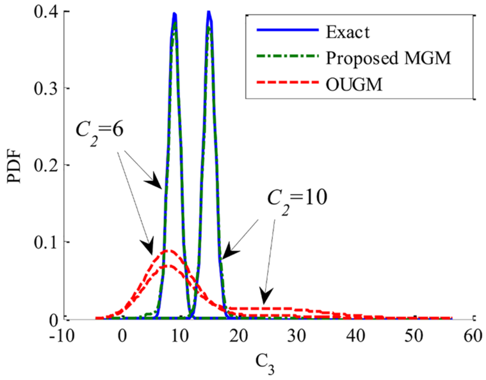
$$A = A_1/100000 \qquad (47)$$

where $A_1 \sim LN(8 + 3C,\ 0.1 + 0.4C)$ and $LN(\cdot,\ \cdot)$ stands for log-normal distribution. Note that, when $C = 0$, A_1 represents the initial crack length which is inherent in the beam structure.

The load P is assumed to follow a normal distribution, $P \sim N(25,\ 2.5^2)$ Newton. Distributions of B_1 and B_2 are given by $B_1 \sim N(\mu_{B_1},\ (0.1|\mu_{B_1}|)^2)$ and $B_2 \sim N(\mu_{B_2},\ |0.1\mu_{B_2}|^2)$, where μ_{B_1}



(a) D_3=0



(b) D_3=1

Fig. 9 Learned CPDs of C_3 with different values of parent nodes: (a) $D_3 = 0$ and (b) $D_3 = 1$

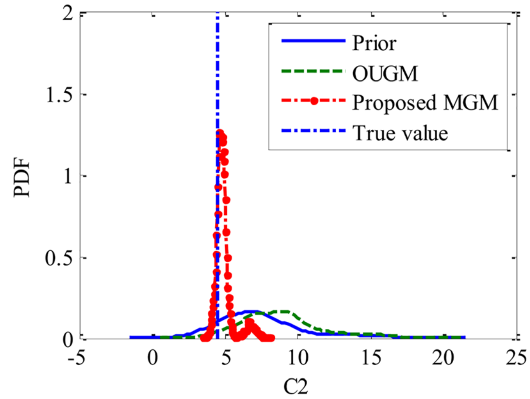
and μ_{B_2} are obtained from finite element analysis simulations and are given by

$$[\mu_{B_1}, \mu_{B_2}] = \begin{cases} [-0.0025,\ 5.5933 \times 10^{-4}] & \text{if } C = 1 \text{ and } D = 1 \\ [-2.3332 \times 10^{-4},\ 5.3143 \times 10^{-4}] & \text{if } C = 1 \text{ and } D = 0 \\ [-1.8231 \times 10^{-4},\ 5.5932 \times 10^{-4}] & \text{if } C = 0 \text{ and } D = 1 \\ [-1.2488 \times 10^{-5},\ 5.3141 \times 10^{-4}] & \text{if } C = 0 \text{ and } D = 0 \end{cases} \quad (48)$$

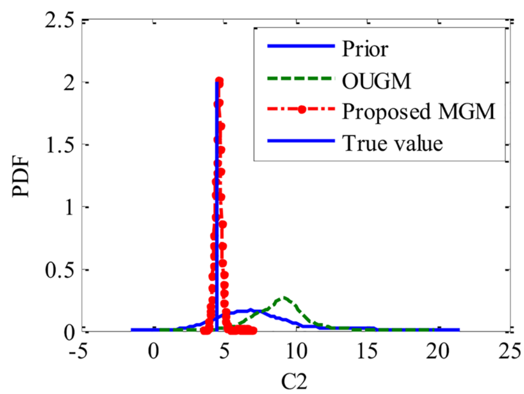
The CPD of F is given by

$$F \sim N(B_1 + B_2 P + A(0.0013 + 0.008D),\ (3 \times 10^{-4})^2)\, m \quad (49)$$

4.2.2 Bayesian Network Structure Learning. Similar to Example 1, we assume that we have no knowledge of the exact BN and CPDs. We learn the structure and parameters from synthetic data generated based on the information in Sec. 4.2.1. Figure 12 gives the learned structures with different amount of data from different

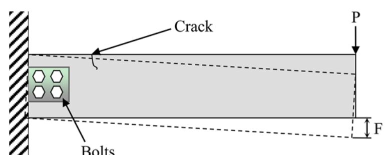


(a) 5 observations

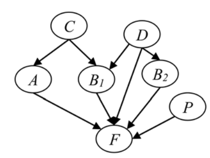


(b) 10 observations

Fig. 10 Model calibration results based on different methods: (a) five observations and (b) ten observations



(a) A beam with crack



(b) Exact BN

Fig. 11 A beam with possible crack and support damage: (a) a beam with crack and (b) exact BN

Table 10 CPTs of C and D

Value of C	$C = 0$	$C = 1$
Probability	0.25	0.75
Value of D	$D = 0$	$D = 1$
Probability	0.4	0.6

Table 9 Calibration results of D_3 from different methods

			Five observations		Ten observations	
	True value	Prior	Proposed MGM	OUGM	Proposed MGM	OUGM
$P(D_3 = 1)$	1	0.45	0.9203	0.6887	0.9810	0.8317
$P(D_3 = 0)$	0	0.55	0.0797	0.3113	0.0190	0.1683

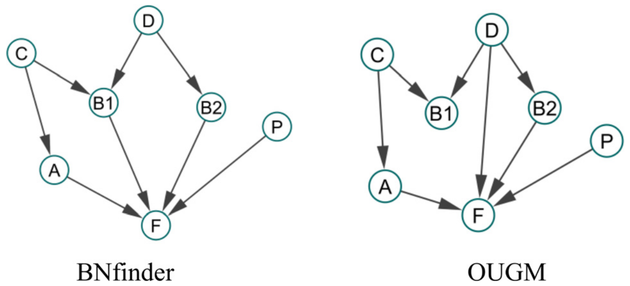


Fig. 12 Learned BN from different methods with 1000–5000 samples

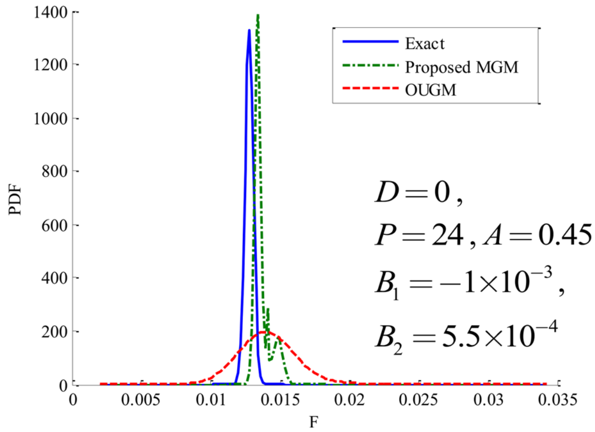
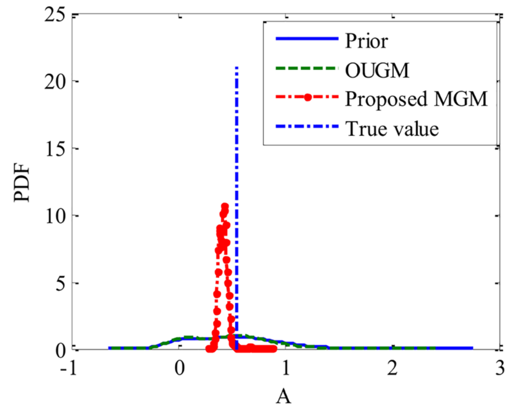
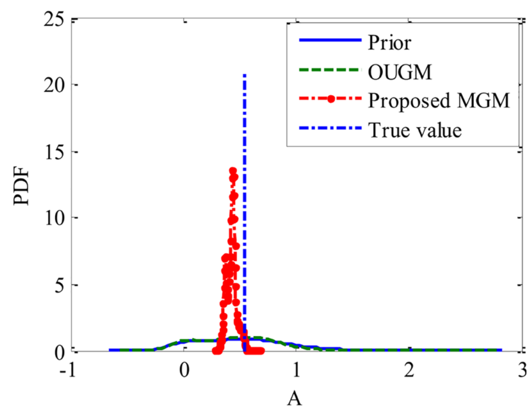


Fig. 13 Learned CPD of *F* under given values of its parents



(*a*) 4 observations



(*b*) 8 observations

Fig. 14 Model calibration results using different methods: (*a*) four observations and (*b*) eight observations

methods (BNFINDER with two-component GMM and OUGM). In OUGM, the optimal number of Gaussian mixture used for variables A,B_1,B_2,P, and F are 3, 5, 2, 2, and 2, respectively.

Figure 12 shows that the BN structure learning algorithms can effectively learn most of the BN topology based on available data. Some connections, however, was not learned due to the identifiability problems of the particular nodes. For the BNFINDER with two-component GMM, the connection $D \rightarrow F$ is not learned. For OUGM, the connection $B1 \rightarrow F$ is not learned. Note that the true model given in Sec. 4.2.1 indicates that the connection $D \rightarrow F$ is more important than $B1 \rightarrow F$. This implies that OUGM learns the structure better than the two-component GMM.

4.2.3 Conditional Probability Densities Learning and Model Calibration. Based on the learned BN topology obtained from 1000 samples, we perform CPD learning and model calibration using the learned BN. Figure 13 shows the CPD of F under given values of its parents. Table 11 and Fig. 14 present the calibration results of D and A based on different numbers of observations of P, B_2, and F.

The results show that the proposed method (MGM) gives results much closer to the true value than the method using univariate GMM for both CPD learning and backward uncertainty quantification problems.

5 Conclusion

In practical engineering applications of data-driven design, it is quite common to have both discrete and continuous variables. This paper pursued the Bayesian network methodology for machine learning, and investigated the Gaussian mixture approach for topology learning and parameter learning. It is observed that existing approaches in the literature employ two-component univariate GMM, which may not perform well for some problems. By extending the two-component univariate GMM to optimal univariate GMM, the OUGM is adequate for topology learning and also for parameter learning in BNs that have only discrete variables. For BNs with both discrete and continuous variables, a new MGM method is proposed in this paper to map the data of discrete variables into data of a standard normal variable. Two numerical examples illustrate the effectiveness of the proposed framework and its application in model calibration.

Conditional probabilities obtained from GMMs have constant variances; therefore, the proposed method is affected by this limitation of GMMs. Besides, the proposed method to transform discrete data into continuous data may not be the best way of

Table 11 Calibration results of *D* from different methods

			Four observations		Eight observations	
	True value	Prior	Proposed MGM	OUGM	Proposed MGM	OUGM
$P(D=1)$	0	0.6080	0.1620	0.5240	0.0470	0.4730
$P(D=0)$	1	0.3920	0.8380	0.4760	0.9530	0.5270

handling this problem. For some problems, it may cause difficulties in the modeling of multivariate GMMs. Future work needs to explore other ways of addressing this issue.

Funding Data

- National Institute of Standards and Technology under the Smart Manufacturing Data Analytics Project (Cooperative Agreement No. 70 NANB14H036).

References

[1] Telenko, C., and Seepersad, C. C., 2014, "Probabilistic Graphical Modeling of Use Stage Energy Consumption: A Lightweight Vehicle Example," ASME J. Mech. Des., **136**(10), p. 101403.

[2] Liang, C., and Mahadevan, S., 2017, "Pareto Surface Construction for Multi-Objective Optimization Under Uncertainty," Struct. Multidiscip. Optim., **55**(5), pp. 1865–1882.

[3] Du, X., and Chen, W., 2004, "Sequential Optimization and Reliability Assessment Method for Efficient Probabilistic Design," ASME J. Mech. Des., **126**(2), pp. 225–233.

[4] Cheng, Y., and Du, X., 2016, "System Reliability Analysis With Dependent Component Failures During Early Design Stage—A Feasibility Study," ASME J. Mech. Des., **138**(5), p. 051405.

[5] Hu, Z., and Mahadevan, S., 2016, "Resilience Assessment Based on Time-Dependent System Reliability Analysis," ASME J. Mech. Des., **138**(11), p. 111404.

[6] Hu, Z., Mahadevan, S., and Du, X., 2016, "Uncertainty Quantification of Time-Dependent Reliability Analysis in the Presence of Parametric Uncertainty," ASCE-ASME J. Risk Uncertainty Eng. Syst., Part B: Mech. Eng., **2**(3), p. 031005.

[7] Shahan, D. W., and Seepersad, C. C., 2012, "Bayesian Network Classifiers for Set-Based Collaborative Design," ASME J. Mech. Des., **134**(7), p. 071001.

[8] Khakzad, N., Khan, F., and Amyotte, P., 2013, "Dynamic Safety Analysis of Process Systems by Mapping Bow-Tie Into Bayesian Network," Process Saf. Environ. Prot., **91**(1–2), pp. 46–53.

[9] Khakzad, N., Khan, F., and Amyotte, P., 2012, "Dynamic Risk Analysis Using Bow-Tie Approach," Reliab. Eng. Syst. Saf., **104**, pp. 36–44.

[10] Yuan, Z., Khakzad, N., Khan, F., and Amyotte, P., 2015, "Risk Analysis of Dust Explosion Scenarios Using Bayesian Networks," Risk Anal., **35**(2), pp. 278–291.

[11] Gradowska, P. L., and Cooke, R. M., 2014, "Estimating Expected Value of Information Using Bayesian Belief Networks: A Case Study in Fish Consumption Advisory," Environ. Syst. Decisions, **34**(1), pp. 88–97.

[12] Liang, C., and Mahadevan, S., 2016, "Multidisciplinary Optimization Under Uncertainty Using Bayesian Network," SAE Int. J. Mater. Manf., **9**(2), pp. 419–429.

[13] Bartram, G., and Mahadevan, S., 2014, "Integration of Heterogeneous Information in SHM Models," Struct. Control Health Monit., **21**(3), pp. 403–422.

[14] Groth, K. M., and Swiler, L. P., 2013, "Bridging the Gap Between HRA Research and HRA Practice: A Bayesian Network Version of SPAR-H," Reliab. Eng. Syst. Saf., **115**, pp. 33–42.

[15] Groth, K. M., and Mosleh, A., 2012, "A Data-Informed PIF Hierarchy for Model-Based Human Reliability Analysis," Reliab. Eng. Syst. Saf., **108**, pp. 154–174.

[16] Sankararaman, S., and Mahadevan, S., 2015, "Integration of Model Verification, Validation, and Calibration for Uncertainty Quantification in Engineering Systems," Reliab. Eng. Syst. Saf., **138**, pp. 194–209.

[17] Hu, Z., and Mahadevan, S., 2017, "Bayesian Network Learning for Uncertainty Quantification," ASME Paper No. DETC2017-68187.

[18] He, L., Wang, M., Chen, W., and Conzelmann, G., 2014, "Incorporating Social Impact on New Product Adoption in Choice Modeling: A Case Study in Green Vehicles," Transp. Res. Part D: Transp. Environ., **32**, pp. 421–434.

[19] Vinh, N. X., Chetty, M., Coppel, R., and Wangikar, P. P., 2011, "GlobalMIT: Learning Globally Optimal Dynamic Bayesian Network With the Mutual Information Test Criterion," Bioinformatics, **27**(19), pp. 2765–2766.

[20] Ziebarth, J. D., Bhattacharya, A., and Cui, Y., 2013, "Bayesian Network Webserver: A Comprehensive Tool for Biological Network Modeling," Bioinformatics, **29**(21), pp. 2801–2803.

[21] Murphy, K. P., 2002, "Dynamic Bayesian Networks: Representation, Inference and Learning," Ph.D. dissertation, University of California, Berkeley, CA.

[22] Karkera, K. R., 2014, *Building Probabilistic Graphical Models With Python*, Packt Publishing, Birmingham, UK.

[23] Hanea, A., Kurowicka, D., Cooke, R. M., and Ababei, D., 2010, "Mining and Visualising Ordinal Data With Non-Parametric Continuous BBNs," Comput. Stat. Data Anal., **54**(3), pp. 668–687.

[24] Bedford, T., and Cooke, R. M., 2002, "Vines: A New Graphical Model for Dependent Random Variables," Ann. Stat., **30**(4), pp. 1031–1068.

[25] Shenoy, P. P., and West, J. C., 2011, "Inference in Hybrid Bayesian Networks Using Mixtures of Polynomials," Int. J. Approximate Reasoning, **52**(5), pp. 641–657.

[26] Dojer, N., Bednarz, P., Podsiadło, A., and Wilczyński, B., 2013, "BNFinder2: Faster Bayesian Network Learning and Bayesian Classification," Bioinformatics, **29**(16), pp. 2068–2070.

[27] McGeachie, M. J., Chang, H.-H., and Weiss, S. T., 2014, "CGBayesNets: Conditional Gaussian Bayesian Network Learning and Inference With Mixed Discrete and Continuous Data," PLoS Comput. Biol., **10**(6), p. e1003676.

[28] Hu, Z., and Mahadevan, S., 2017, "Uncertainty Quantification and Management in Additive Manufacturing: Current Status, Needs, and Opportunities," Int. J. Adv. Manuf. Technol., **93**(5–8), pp. 2855–2874.

[29] Hu, Z., Mahadevan, S., and Ao, D., 2017, "Uncertainty Aggregation and Reduction in Structure–Material Performance Prediction," Comput. Mech., epub.

[30] Scutari, M., 2009, "Learning Bayesian Networks With the bnlearn R Package," preprint arXiv: 0908.3817.

[31] Bonissone, P., Henrion, M., Kanal, L., and Lemmer, J., "Equivalence and Synthesis of Causal Models," *Uncertainty Artificial Intelligence*, Elsevier, Amsterdam, The Netherlands, pp. 255–270.

[32] Wilczyński, B., and Dojer, N., 2009, "BNFinder: Exact and Efficient Method for Learning Bayesian Networks," Bioinformatics, **25**(2), pp. 286–287.

[33] Bilmes, J. A., 1998, "A Gentle Tutorial of the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," Int. Comput. Sci. Inst., **4**(510), p. 126.

[34] Sahin, F., Yavuz, M. Ç., Arnavut, Z., and Uluyol, Ö., 2007, "Fault Diagnosis for Airplane Engines Using Bayesian Networks and Distributed Particle Swarm Optimization," Parallel Comput., **33**(2), pp. 124–143.

[35] Yang, L., and Lee, J., 2012, "Bayesian Belief Network-Based Approach for Diagnostics and Prognostics of Semiconductor Manufacturing Systems," Rob. Comput.-Integr. Manuf., **28**(1), pp. 66–74.

[36] Rodriguez-Zas, S., and Ko, Y., 2011, "Elucidation of General and Condition-Dependent Gene Pathways Using Mixture Models and Bayesian Networks," *Applied Statistics for Network Biology: Methods in Systems Biology*, Wiley-Blackwell, Weinheim, Germany, pp. 91–103.

[37] Sun, S., Zhang, C., and Yu, G., 2006, "A Bayesian Network Approach to Traffic Flow Forecasting," IEEE Trans. Intell. Transp. Syst., **7**(1), pp. 124–132.

[38] Zhang, H., Giles, C. L., Foley, H. C., and Yen, J., 2017, "Probabilistic Community Discovery Using Hierarchical Latent Gaussian Mixture Model," 22nd National conference on Artificial Intelligence (AAAI'07), Vancouver, BC, Canada, July 22–26, pp. 663–668.

[39] Hu, Z., and Mahadevan, S., 2017, "Time-Dependent Reliability Analysis Using a Vine-ARMA Load Model," ASCE-ASME J. Risk Uncertainty Eng. Syst., Part B: Mech. Eng., **3**(1), p. 011007.

[40] Davies, S., and Moore, A., "Mix-Nets: Factored Mixtures of Gaussians in Bayesian Networks With Mixed Continuous and Discrete Variables," Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI), Stanford, CA, June 30–July 3, pp. 168–175.

[41] Morlini, I., 2012, "A Latent Variables Approach for Clustering Mixed Binary and Continuous Variables Within a Gaussian Mixture Model," Adv. Data Anal. Classif., **6**(1), pp. 5–28.

[42] Bartram, G. W., 2013, "System Health Diagnosis and Prognosis Using Dynamic Bayesian Networks," Ph.D. thesis, Vanderbilt University, Nashville, TN.