




Being Bayesian about learning Bayesian networks from hybrid data

Marco Grzegorzcyk 

Bernoulli Institute, Groningen University, Nijenborgh 9, 9747AG, Groningen, Netherlands

ARTICLE INFO

Keywords:

Hybrid Bayesian networks
Mean-adjusted BGe (mBGe) score
Score-equivalence
Bayesian regression
Markov Chain Monte Carlo

ABSTRACT

We develop a new Bayesian model to infer the structure of Bayesian networks from hybrid data, that is, data containing a mix of continuous (Gaussian) and discrete (categorical) variables. In line with state-of-the-art hybrid Bayesian network models, we do not allow discrete variables to have continuous parents. However, our new model differs from existing approaches by incorporating discrete variables through multivariate linear regression rather than mixture modeling. In our model, the continuous variables follow a multivariate Gaussian distribution with a shared covariance matrix, while the mean vector varies across different configurations.

As with all Bayesian network models, we use directed acyclic graphs (DAGs) to represent conditional dependency relations among the continuous variables. For our Gaussian distribution, this requires the covariance matrix to be consistent with the structure of the DAG. Our key idea is to apply multivariate linear regression, using the discrete variables as potential covariates to adjust the mean vector of the multivariate Gaussian distribution. Each continuous variable is associated with its own regression model and discrete parent set. Since the values of the discrete variables vary across observations, the mean vector becomes observation-specific.

This enables mean-adjustment of the continuous variables for their discrete parents while simultaneously inferring a Gaussian Bayesian network among them. In simulation studies, we compare our new model against two state-of-the-art hybrid Bayesian network models and demonstrate that both existing models have conceptual shortcomings, positioning our new hybrid Bayesian network model as a strong alternative.

1. Introduction

Bayesian networks (BNs) are a flexible and powerful tool for learning the probabilistic dependencies among random variables. The variables are represented as nodes of a directed acyclic graph (DAG), and the directed edges of the DAG represent conditional (in-)dependence relations among them. Thorough mathematical introductions to BNs can be found in several classic textbooks, such as Pearl [1] and Neapolitan [2], as well as in the more recent textbook by Koller and Friedman [3]. For a more applied introduction to BNs, we recommend the textbook by Scutari and Denis [4]. BN structure learning aims to learn DAGs from data. Since the number of possible DAGs grows super-exponentially with the number of nodes [5], structure learning is computationally challenging. For introductory readings on BN structure learning, we recommend Kuipers and Moffa [6], Scutari et al. [7] and Scutari [8]. An impressive exhaustive review of more than sixty BN structure learning algorithms has been recently provided by Kitson et al. [9]. In this paper, we focus on score-based structure learning methods. These methods evaluate (i.e., score) DAGs based on the data. The most commonly

E-mail address: m.a.grzegorzcyk@rug.nl.

<https://doi.org/10.1016/j.ijar.2025.109549>

Received 7 April 2025; Received in revised form 3 July 2025; Accepted 10 August 2025

used scoring metrics are marginal likelihoods [10] and BIC scores [3]. Frequentist approaches typically seek the highest-scoring DAG, while Bayesian approaches focus on sampling DAGs from the posterior distribution and averaging over the sampled DAGs (i.e., Bayesian model averaging). Different data types require different scoring metrics. For example, there are scoring metrics for Gaussian data [11–13], for discrete data [14,15], for ordinal data [16–18] as well as for a mix of discrete and Gaussian data [19,20,8]. In this paper, we refer to data containing both discrete (categorical) and continuous (Gaussian) variables as hybrid data.

The focus of this paper is on learning Bayesian networks from hybrid data. A review of the literature reveals many works on hybrid Bayesian networks. However, almost all of these works assume that the network structure (DAG) is known and focus on probability propagation inference algorithms. Different graph types (model classes) can be used to describe the probabilistic relationships among variables, and there are tailored algorithms for each important model class. A traditional and still widely used model class is the class of conditional linear Gaussian (clG) models. The clG models do not allow discrete variables to have continuous parents. For clG models, efficient exact inference algorithms have been developed by Lauritzen and Wermuth [20], Lauritzen [21] and Lauritzen and Jensen [22]. In particular, Cowell [23] developed a local propagation scheme for conditional Gaussian distributions, allowing all computations involving continuous variables to be carried out via univariate regression. Other researchers have relaxed this restriction and proposed inference algorithms for more general graph structures. For example, the approximate clique tree algorithm from Koller et al. [24] and the algorithm from Moral et al. [25], which employs mixtures of exponential distributions to represent the domain variables. For overviews of inference algorithms, we refer to the works by Romero et al. [26], Langseth et al. [27] and Salmerón et al. [28].

There is limited literature on learning network structures (DAGs) from hybrid data. One straightforward approach is to discretize the Gaussian data. Then all variables are discrete, allowing the use of scoring metrics for discrete data. Information-conserving discretization algorithms have, for example, been proposed by Friedman and Goldszmidt [29] and Monti and Cooper [30]. To our knowledge, Heckerman and Geiger [10] were the first to propose a structure learning approach for hybrid data. Geiger and Heckerman propose inferring two separate DAGs: one for the discrete variables and another for the continuous variables. While the DAG among the discrete variables can be scored using any metric for discrete data, the DAG among the continuous variables is scored using a Gaussian mixture model with known labels. The continuous data are divided into mixture components, with each possible value combination of the discrete nodes corresponding to a separate mixture component. There is only one DAG across all components, so that the component-specific Gaussian distributions are subject to the same conditional dependencies. Conditional on the shared DAG, the component-specific network parameters are independent. Effectively, for any given DAG, there is then for each mixture component a separate Gaussian Bayesian network with the so-called *BGe score* [11]. The product of the component-specific BGe scores gives the overall score of the DAG among the continuous variables. We refer to the model that uses this product as its score as the *hybrid Bayesian model having score-equivalence (hBe)* model. A detailed description of the hBe model is provided in Section 2.

A similar model for hybrid Bayesian networks is implemented in the popular *bnlearn* R package [31,32,4]. The model is adapted from the conditional linear Gaussian (clG) model of Lauritzen and Wermuth [20], Lauritzen [21], Lauritzen and Jensen [22]. The clG model is similar to the model from Heckerman and Geiger [10], but it differs in key points. It infers a DAG among all variables but imposes the restriction that discrete variables are not allowed to have continuous variables as parent nodes. The clG model employs mixtures of linear regression to describe the relationships between parent and child nodes. For a continuous variable with both discrete and continuous parent nodes, the clG model assigns a specific mixture component of the linear regression model to each value combination of the discrete parent nodes. The linear regression model describes the component-specific relationship between the continuous variable and its continuous parent nodes, with the regression model parameters (coefficients and noise variances) being specific to each component. The regression mixtures collectively specify the joint distribution of all continuous variables. We propose a new model for hybrid Bayesian networks. We follow earlier works [10,20,8,33,4] and do not allow discrete variables to have continuous parent nodes. Our model differs from earlier approaches [10,20] in that it does not use mixture modeling. Instead, it employs the covariance matrix of a Gaussian Bayesian network to describe the relationships among the continuous variables, while allowing the discrete parent variables to influence the mean vector of the Gaussian distribution through multivariate linear regression. In other words, our model adjusts the observation-specific means of the continuous variables for the effects of the discrete variables, while simultaneously learning a Gaussian Bayesian network (DAG) among them. We believe that this results in an intuitive statistical model, as in our approach both parent types (discrete and continuous) have independent additive effects on the child node. The earlier models from Heckerman and Geiger [10] and Lauritzen and Wermuth [20] use mixture models, which lead to interaction effects between the continuous and the discrete parent nodes. The key idea behind our new model for hybrid Bayesian networks closely aligns with the modeling approach of Rijmen [34], who proposed using logistic regression models to describe conditional dependencies in discrete Bayesian networks. As in our model, the exclusion of interaction effects in the logistic regression models ensures that the effects of discrete variables remain additive, significantly reducing the number of network parameters.

In practical applications, selecting the most suitable hybrid Bayesian network model for a given dataset can be challenging. One advantage of our new model is that it has fewer parameters (see Table 2). Due to its sparsity, we expect our model to perform better on small datasets with limited observations. In addition to model sparsity, our new model, which we refer to as the mean-adjusted BGe (mBGe) model, also circumvents the following shortcomings of the two competing models:

- The clG model [8,33], as implemented in the *bnlearn* R package, has an overlooked conceptual issue. It enforces a graph structure in which all edges between discrete and continuous variables must be directed towards the continuous variables. As a result, some of these edges may contradict their natural directions, potentially disrupting equivalence classes and violating the principle of score equivalence. In Section 4.1, we illustrate this issue with a toy example.

- The hBe model [10] infers a single DAG but applies separate Gaussian BGe models for all value combinations of the discrete nodes. As a result, all continuous variables depend on all discrete variables, preventing the hBe model from learning which specific continuous variables are affected by which discrete variables and rendering the continuous variables mutually pairwise (marginally) dependent. In Section 4.2 we use a toy example to illustrate this.

In Section S1 of the supplementary material, we review linear regression with a mix of continuous and discrete covariates to motivate the mBGe model.

2. Statistical methods

Sections 2.1 and 2.2 provide an overview of Bayesian networks (BNs) and Gaussian Bayesian networks (GBNs), respectively. We introduce our new mBGe model for hybrid BNs in Section 2.3 and describe our MCMC algorithm for model inference in Section 2.4. The competing models for hybrid BNs are reviewed in Section 2.5, while technical details and simulation specific details are presented in Section 3.

2.1. Bayesian networks

Bayesian networks (BNs) use *directed acyclic graphs* (DAGs) to impose conditional dependencies among random variables X_1, \dots, X_n . The variables become the nodes of the DAG, and the directed edges between the nodes represent conditional dependencies among them. DAGs must be acyclic, meaning they do not contain any directed paths of the form $X_i \rightarrow \dots \rightarrow X_i$. Node X_j is called a parent of X_i if there is a directed edge from X_j to X_i . We introduce the symbol Π_i to represent the set of all parent nodes of X_i . A DAG \mathcal{G} , with parent sets Π_i for $i = 1, \dots, n$, symbolically $\mathcal{G} = \{\Pi_1, \dots, \Pi_n\}$, implies conditional dependencies among the variables, such that:

$$p(X_1, \dots, X_n | \mathcal{G}) = \prod_{i=1}^n p(X_i | \Pi_i). \quad (1)$$

When two DAGs \mathcal{G}_r ($r = 1, 2$) imply the same conditional dependencies, the factorizations are equivalent, meaning that:

$$p(X_1, \dots, X_n | \mathcal{G}_1) = p(X_1, \dots, X_n | \mathcal{G}_2). \quad (2)$$

The two DAGs \mathcal{G}_1 and \mathcal{G}_2 are then said to be *equivalent*. More generally, all possible DAGs can be grouped into equivalence classes, and *completed partially directed acyclic graphs* (CPDAGs) can be used to characterize these DAG equivalence classes [35]. Chickering [35] also shows that two DAGs are equivalent if and only if they have the same skeleton and the same v-structures.¹ The DAGs within one equivalence class and the corresponding CPDAG share the same skeleton. However, unlike DAGs, which consist only of directed edges, CPDAGs can contain both directed and undirected edges. In a CPDAG, a directed edge $X_j \rightarrow X_i$ indicates that all DAGs in the equivalence class include this directed edge. An undirected edge $X_j - X_i$ indicates that all DAGs in the equivalence class have X_j and X_i connected, but there is at least one DAG with the edge $X_j \rightarrow X_i$ and at least one DAG with the edge $X_j \leftarrow X_i$.

A dataset \mathcal{D} with N realizations of the random variables X_1, \dots, X_n can be represented as an n -by- N matrix. We denote the value of X_i in the v -th observation by $x_{i,v}$. To simplify notation, we introduce the vector notations:

$$X := \begin{pmatrix} X_1 \\ \vdots \\ X_n \end{pmatrix} \quad D := \begin{pmatrix} x_{1,1} & x_{1,2} & \dots & x_{1,N} \\ x_{2,1} & x_{2,2} & \dots & x_{2,N} \\ \vdots & \vdots & & \vdots \\ x_{n,1} & x_{n,2} & \dots & x_{n,N} \end{pmatrix} \quad \mathbf{x}_v := \begin{pmatrix} x_{1,v} \\ \vdots \\ x_{n,v} \end{pmatrix}.$$

Thus, X is a random vector of the domain variables, D is an n -by- N data matrix, and \mathbf{x}_v is the v -th observation of \mathbf{x} or the v -th column of D , respectively.

Given a prior distribution over the DAGs, $p(\mathcal{G})$, and the marginal likelihood, $p(D | \mathcal{G})$, the DAG posterior distribution is defined as:

$$p(\mathcal{G} | D) := \frac{p(D | \mathcal{G})p(\mathcal{G})}{p(D)} \propto p(D | \mathcal{G})p(\mathcal{G}), \quad (3)$$

where $p(D) := \sum_{\mathcal{G}} p(D | \mathcal{G})p(\mathcal{G})$ is a normalization constant.

Geiger and Heckerman developed the *BGe scoring metric* for Gaussian variables [12,13,10,11], and Geiger, Heckerman and Chickering introduced the *BDe scoring metric* for discrete variables [15,19,10].² Both metrics (BGe and BDe) allow the marginal likelihood, $p(D | \mathcal{G})$, to be computed analytically and ensure that equivalent graphs yield the same marginal likelihood value. This property, known as *score-equivalence*, is crucial in BNs because equivalent graphs \mathcal{G}_r ($r = 1, 2$) describe the same conditional dependencies and, therefore, should not be distinguishable based on data. Mathematically, this means that two equivalent graphs \mathcal{G}_1 and \mathcal{G}_2 must have the same marginal likelihood value, so $p(D | \mathcal{G}_1) = p(D | \mathcal{G}_2)$.

¹ The skeleton of the DAG \mathcal{G} is obtained by replacing all directed edges with undirected edges. A v-structure is an edge constellation, where two edges converge on a node X_i as $X_k \rightarrow X_i \leftarrow X_j$, with no edge between the parent nodes X_k and X_j .

² BGe/BDe stand for: *Bayesian scoring metric for Gaussian/Discrete networks having score equivalence*.

Markov Chain Monte Carlo (MCMC) sampling can be used to generate samples from the DAG posterior distribution $p(\mathcal{G}|\mathcal{D})$ in Eq. (3). Given the DAG posterior sample $\mathcal{G}_1, \dots, \mathcal{G}_R$, marginal edge posterior probabilities can be computed. Taking equivalence classes into account, we first replace each sampled DAG \mathcal{G}_i with its CPDAG and interpret each undirected CPDAG edge $X_j - X_i$ as a superposition of two oppositely oriented edges, $X_j \leftrightarrow X_i$. We then estimate the marginal posterior probability $e_{j,i} \in [0, 1]$ of the edge $X_j \rightarrow X_i$ as the proportion of CPDAGs that contain either the directed edge $X_j \rightarrow X_i$ or the bidirectional edge $X_j \leftrightarrow X_i$.

2.2. Gaussian Bayesian networks

In Gaussian Bayesian networks (GBNs) the vector X follows an n -dimensional Gaussian distribution, and the columns of \mathcal{D} are a random sample:

$$\mathbf{x}_1, \dots, \mathbf{x}_N | (\boldsymbol{\mu}, \boldsymbol{\Sigma}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}). \quad (4)$$

The BGe score from [12,13,10,11] imposes a Normal-Wishart prior on the mean vector $\boldsymbol{\mu}$ and precision matrix $\mathbf{W} = \boldsymbol{\Sigma}^{-1}$. In Grzegorzczak [17], the BGe score was simplified by assuming a known mean vector $\boldsymbol{\mu} = \mathbf{0}$ and imposing a Wishart prior on \mathbf{W} , $\mathbf{W} \sim \mathcal{W}(\alpha_w, \mathbf{R})$, with $\alpha_w > n + 1$ degrees of freedom and parametric matrix \mathbf{R} .

In Section 2.3, we consider hybrid BNs comprising a mix of Gaussian variables, X_1, \dots, X_n , and discrete variables, Z_1, \dots, Z_m . We then employ multivariate linear regression to model the relationship between the discrete variables Z_1, \dots, Z_m (as covariates) and the mean vector $\boldsymbol{\mu}$ (as the response vector). This regression model defines a linear transformation $\mathbf{T} : \mathbb{R}^m \rightarrow \mathbb{R}^n$, such that each realization \mathbf{z}_v of $(Z_1, \dots, Z_m)^\top$ corresponds to a specific mean vector $\boldsymbol{\mu}_v = \mathbf{T}(\mathbf{z}_v)$. Loosely speaking, our new approach adjusts the continuous variables to account for the effects of the discrete variables, and the covariance matrix $\boldsymbol{\Sigma}$ captures the dependencies among the mean-adjusted variables. This leads to the relation:

$$\mathbf{x}_v | (\boldsymbol{\Sigma}, \mathbf{T}, \mathbf{z}_v) \sim \mathcal{N}(\mathbf{T}(\mathbf{z}_v), \boldsymbol{\Sigma}), \quad (v = 1, \dots, N). \quad (5)$$

The posterior distribution of \mathbf{W} is given by [12,13,17]:

$$\mathbf{W} | (\mathcal{D}, \boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_N) \sim \mathcal{W}(\alpha_w + N, \mathbf{R} + \mathbf{S}) \text{ with } \mathbf{S} := \sum_{v=1}^N (\mathbf{x}_v - \boldsymbol{\mu}_v)(\mathbf{x}_v - \boldsymbol{\mu}_v)^\top,$$

where α_w and \mathbf{R} are the parameters of the Wishart prior, and $\boldsymbol{\mu}_v := \mathbf{T}(\mathbf{z}_v)$. Also the marginal likelihood, $p(\mathcal{D}|\mathcal{G})$, can be computed analytically, as shown in Section S2 of the supplementary material.

Covariance matrices, $\boldsymbol{\Sigma} = \mathbf{W}^{-1}$, sampled from the posterior distribution above do not imply conditional independencies. To sample a covariance matrix that is consistent with a given DAG \mathcal{G} , we use the algorithm from Grzegorzczak [36]. In a GBN with $\mathbf{x}_v \sim \mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\Sigma})$, the conditional distributions $X_i | \boldsymbol{\Pi}_i$ are univariate Gaussian distributions of the form [36]:

$$x_{i,v} | (\boldsymbol{\Pi}_i, \mathbf{x}_{-i,v}) \sim \mathcal{N} \left(\mu_{i,v} + \sum_{j: (X_j \in \boldsymbol{\Pi}_i)} b_{i,j} (x_{j,v} - \mu_{j,v}), \sigma_i^2 \right) \quad (i = 1, \dots, n),$$

where $\mathbf{x}_{-i,v}$ is the vector \mathbf{x}_v without its i -th element, $b_{i,j}$ is a linear coefficient reflecting the strength of the relationship between X_i and X_j , $\mu_{j,v}$ is the j -th element of $\boldsymbol{\mu}_v$, and $x_{j,v}$ is the v -th value of X_j . A covariance matrix $\boldsymbol{\Sigma}$ is consistent with a DAG \mathcal{G} , denoted $\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\mathcal{G})$, if each Gaussian $\mathcal{N}(\boldsymbol{\mu}_v, \boldsymbol{\Sigma})$ can be factorized according to Eq. (1):

$$p(\mathbf{x}_v | \boldsymbol{\mu}_v, \boldsymbol{\Sigma}) = \prod_{i=1}^n p(x_{i,v} | \boldsymbol{\Pi}_i, \mathbf{x}_{-i,v}). \quad (6)$$

2.3. A new modeling approach for hybrid Bayesian networks

We extend a GBN with n continuous variables, X_1, \dots, X_n , by incorporating m discrete variables, Z_1, \dots, Z_m . Each discrete variable Z_q ($1 \leq q \leq m$) can take values from the set $\{1, \dots, k_q\}$, with each value representing a different category. Our goal is to learn a hybrid BN among the $n + m$ variables. We follow earlier works [20,10,4,8,33] and restrict the model such that discrete variables cannot have continuous parents.

We recall that $X := (X_1, \dots, X_n)^\top$, \mathbf{x}_v is the v -th observation of X , $x_{i,v}$ represents the value of X_i in \mathbf{x}_v , and that we use $\boldsymbol{\Pi}_i \subset \{X_1, \dots, X_n\}$ to denote the continuous parent nodes of X_i . In a similar manner, we define:

$$Z := \begin{pmatrix} Z_1 \\ \vdots \\ Z_m \end{pmatrix} \quad D_z := \begin{pmatrix} z_{1,1} & z_{1,2} & \dots & z_{1,N} \\ z_{2,1} & z_{2,2} & \dots & z_{2,N} \\ \vdots & \vdots & & \vdots \\ z_{m,1} & z_{m,2} & \dots & z_{m,N} \end{pmatrix} \quad \mathbf{z}_v := \begin{pmatrix} z_{1,v} \\ \vdots \\ z_{m,v} \end{pmatrix},$$

so that Z is the random vector of the discrete variables, \mathbf{z}_v is the v -th observation of Z , and D_z represents the discrete data. Furthermore, let $\Delta_i \subset \{Z_1, \dots, Z_m\}$ denote the set of discrete parent nodes of X_i .

We use linear regression to model the relationships between the discrete variables (covariates) and the continuous variables (responses). For the mean $\mu_{i,v}$ of X_i in observation v , we assume the following relationship:

$$\mu_{i,v} | (\Delta_i, \mathbf{z}_v, \Theta_i) = \sum_{q: (Z_q \in \Delta_i)} \theta_{i,q,z_{q,v}}, \quad (7)$$

where $\theta_{i,q,z_{q,v}}$ represents the effect of $Z_q = z_{q,v}$ on $\mu_{i,v}$, and Θ_i is the vector of all model parameters for response variable X_i . Eq. (7) can be rewritten as:

$$\mu_{i,v} | (\Delta_i, \mathbf{z}_v, \Theta_i) = \beta_{i,0} + \sum_{q: (Z_q \in \Delta_i)} \sum_{k=2}^{k_q} \theta_{i,q,k} I_k(z_{q,v}) \quad (v = 1, \dots, N), \quad (8)$$

where $I_k(z_{q,v}) = 1$ if $z_{q,v} = k$, and $I_k(z_{q,v}) = 0$ otherwise, and $\beta_{i,0} + \theta_{i,q,k}$ represents the effect of $z_q = k$ on μ_i ($k > 1$).³ Eq. (8) can be compactly written as a scalar product:

$$\mu_{i,v} | (\Delta_i, \mathbf{z}_v, \beta_i) = \tilde{\mathbf{z}}_{i,v}^\top \beta_i,$$

where $\tilde{\mathbf{z}}_{i,v}$ is a binary indicator vector, encoding the values of the discrete parent nodes Δ_i of X_i in \mathbf{z}_v , with an initial element corresponding to the intercept term $\beta_{i,0}$. The vector β_i arranges the parameters $\beta_{i,0}$ and $\theta_{i,q,k}$ ($q: (Z_q \in \Delta_i)$ and $k = 2, \dots, k_q$) in such a way that they correspond to the elements of the binary indicator vector $\tilde{\mathbf{z}}_{i,v}$.

By stacking the mean parameters and the regression coefficient vectors, and constructing a block design matrix, we can express the relationship compactly as:

$$\boldsymbol{\mu}_v | (\tilde{\Delta}, \mathbf{z}_v, \tilde{\beta}) = \mathbf{Z}_v \tilde{\beta}, \quad (9)$$

where $\tilde{\Delta} := \{\Delta_1, \dots, \Delta_n\}$ denotes the system of discrete parent sets, and:

$$\boldsymbol{\mu}_v := \begin{pmatrix} \mu_{1,v} \\ \vdots \\ \mu_{n,v} \end{pmatrix} \in \mathbb{R}^n, \mathbf{Z}_v := \begin{pmatrix} \tilde{\mathbf{z}}_{1,v}^\top & \mathbf{0} \\ \vdots & \ddots \\ \mathbf{0} & \tilde{\mathbf{z}}_{n,v}^\top \end{pmatrix} \in \mathbb{R}^{n,\kappa}, \text{ and } \tilde{\beta} := \begin{pmatrix} \beta_1 \\ \vdots \\ \beta_n \end{pmatrix} \in \mathbb{R}^\kappa,$$

where $\kappa := \sum_{i=1}^n l_i$, with l_i denoting the length of $\tilde{\mathbf{z}}_{i,v}$ ($\tilde{\mathbf{z}}_{i,v} \in \mathbb{R}^{l_i}$). The vector $\tilde{\mathbf{z}}_{i,v}$ depends on the discrete parent set Δ_i and the observation \mathbf{z}_v . Henceforth, the matrix \mathbf{Z}_v depends on $\tilde{\Delta}$ and \mathbf{z}_v , symbolically $\mathbf{Z}_v = \mathbf{Z}_v(\mathbf{z}_v, \tilde{\Delta})$.

Replacing the mapping $\mathbf{T}(\mathbf{z}_v)$ in Eq. (5) with $\mathbf{Z}_v \tilde{\beta}$ from Eq. (9), we obtain:

$$\mathbf{x}_v | (\Sigma, \tilde{\Delta}, \mathbf{z}_v, \tilde{\beta}) \sim \mathcal{N}(\mathbf{Z}_v \tilde{\beta}, \Sigma) \quad (v = 1, \dots, N). \quad (10)$$

By stacking the observation vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$, we get:

$$\begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} | (\Sigma, \tilde{\Delta}, \mathbf{D}_z, \tilde{\beta}) \sim \mathcal{N} \left(\begin{pmatrix} \mathbf{Z}_1 \\ \vdots \\ \mathbf{Z}_N \end{pmatrix} \tilde{\beta}, \begin{pmatrix} \Sigma & \mathbf{0} \\ \mathbf{0} & \Sigma \end{pmatrix} \right).$$

On the regression vector $\tilde{\beta}$, we impose a Gaussian prior: $\tilde{\beta} | \lambda^2 \sim \mathcal{N}(\mathbf{0}, \lambda^2 \mathbf{I})$, where $\lambda^2 > 0$ is a variance hyperparameter. With the definitions

$$\mathbf{x} := \text{vec}(\mathbf{D}) = \begin{pmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_N \end{pmatrix} \text{ and } \mathbf{Z} := \begin{pmatrix} \mathbf{Z}_1 \\ \vdots \\ \mathbf{Z}_N \end{pmatrix} \in \mathbb{R}^{Nn,\kappa}, \quad (11)$$

where $\text{vec}(\cdot)$ is the vectorization operator, we can compactly write:

$$\mathbf{x} | (\Sigma, \tilde{\Delta}, \mathbf{D}_z, \tilde{\beta}) \sim \mathcal{N}(\mathbf{Z} \tilde{\beta}, \mathbf{I} \otimes \Sigma), \quad (12)$$

$$\tilde{\beta} | \lambda^2 \sim \mathcal{N}(\mathbf{0}, \lambda^2 \mathbf{I}). \quad (13)$$

From Eqs. (12)-(13) it follows from standard rules (see, for example, Section 2.3.3 in [37]) that the marginal likelihood and full conditional distribution are given by:

$$\mathbf{x} | (\Sigma, \tilde{\Delta}, \mathbf{D}_z, \lambda^2) \sim \mathcal{N}(\mathbf{0}, \mathbf{I} \otimes \Sigma + \lambda^2 \mathbf{Z} \mathbf{Z}^\top), \quad (14)$$

$$\tilde{\beta} | (\Sigma, \tilde{\Delta}, \mathbf{D}_z, \lambda^2, \mathbf{x}) \sim \mathcal{N}(\Sigma^* \mathbf{Z}^\top (\mathbf{I} \otimes \Sigma)^{-1} \mathbf{x}, \Sigma^*), \quad (15)$$

where $\Sigma^* := (\lambda^{-2} \mathbf{I} + \mathbf{Z}^\top (\mathbf{I} \otimes \Sigma)^{-1} \mathbf{Z})^{-1}$.

We impose a Gamma prior distribution on λ^{-2} with shape parameter $a > 0$ and rate parameter $b > 0$. This implies the full conditional distribution:

³ Each variable Z_q with categories $\{1, \dots, k_q\}$ is represented by $k_q - 1$ indicator variables I_2, \dots, I_{k_q} indicating the presence or absence of the values $2, \dots, k_q$. For each Z_q , the value $z_q = 1$ is the reference level, which is captured by the intercept $\beta_{i,0}$.

$$\lambda^{-2} | (\Sigma, \tilde{\Delta}, D_z, \tilde{\beta}) \sim \text{Ga} \left(a + \frac{\kappa}{2}, b + \frac{1}{2} \tilde{\beta}^\top \tilde{\beta} \right). \quad (16)$$

In Section 2.4, we use the results from Eqs. (14)-(16), to develop a Markov Chain Monte Carlo (MCMC) sampling algorithm. The algorithm infers the discrete parent sets $\tilde{\Delta} = \{\Delta_1, \dots, \Delta_n\}$ and the corresponding regression vector $\tilde{\beta}$ by sampling them from the posterior distribution. Section S3 of the supplementary material provides a compact graphical representation of the new mBGe model.

2.4. Model inference

The new mBGe model for hybrid BNs consists of two components (i) a DAG $\mathcal{G} = \{\Pi_1, \dots, \Pi_n\}$ among the continuous variables and (ii) n parent sets of discrete variables $\tilde{\Delta} = \{\Delta_1, \dots, \Delta_n\}$. Each combination of \mathcal{G} and $\tilde{\Delta}$ defines a distinct model, and our goal is to sample these models from the posterior distribution:

$$p(\mathcal{G}, \tilde{\Delta} | D, D_z).$$

The model parameters consist of a covariance matrix Σ and a regression parameter vector $\tilde{\beta}$, where Σ must be consistent with \mathcal{G} , i.e. $\Sigma = \Sigma(\mathcal{G})$. This means that Σ and \mathcal{G} must imply the same factorization (see Eq (6)). The design matrix Z in Eq. (11) can be constructed from the system of discrete parent sets $\tilde{\Delta}$ and the discrete data $D_z = \{z_1, \dots, z_N\}$. The regression vector $\tilde{\beta}$ must be consistent with the stacked design matrices $Z = (Z_1^\top, \dots, Z_N^\top)^\top$.

We design a Markov Chain Monte Carlo (MCMC) sampling algorithm to generate posterior samples. Our MCMC scheme combines (M1) structure MCMC moves for the DAGs of BNs [38,39] with (M2) standard MCMC moves for Bayesian linear regression (see, e.g., Chapter 9 in Hoff [40]). The MCMC algorithm alternates between these two moves, M1 and M2, until convergence is reached.

- M1** Given the discrete parent sets $\tilde{\Delta}$ and the regression vector $\tilde{\beta}$, we can compute the observation-specific mean vectors $\mu_v = Z_v \tilde{\beta}$. With μ_1, \dots, μ_N known, the simplified BGe score [17] of any DAG \mathcal{G} can be computed analytically. This allows us to update the DAG \mathcal{G} using the structure MCMC sampler [38,39]. For each \mathcal{G} , we can then sample a consistent covariance matrix $\Sigma = \Sigma(\mathcal{G})$ using the algorithm from [36].
- M2** Given Σ , the marginal likelihood for each system of discrete parent sets $\tilde{\Delta}$ can be computed using Eq. (14). Hence, we can use a Metropolis-Hastings (MH) MCMC step to update $\tilde{\Delta}$. Given the current Σ and $\tilde{\Delta}$, we sample a regression vector $\tilde{\beta}$ from its full conditional distribution, as specified in Eq. (15). Finally, given $\tilde{\beta}$, we update the hyperparameter λ^2 by sampling it from its full conditional distribution, as specified in Eq. (16).

A more detailed description of the MCMC sampling algorithm for our new mBGe model is presented in Table 1. Since it is known that the structure MCMC sampler mixes rather slowly [3], we considered using more computationally efficient MCMC samplers for BNs [41,42]. However, the advanced samplers require pre-computation and storage of the BGe scores in look-up tables, which is computationally expensive. Unlike the BGe model, the mBGe model works with mean-adjusted BGe (mBGe) scores. Each change in the regression models affects the mBGe scores, meaning that **all** mBGe scores would need to be re-computed after each M2 move.

2.5. Competing hybrid Bayesian network models

In Section 5 we compare four models for hybrid BNs.

- **BGe**: A naive approach is to ignore the discrete variables and to simply infer DAGs among the continuous variables X_1, \dots, X_n . We implement the BGe model [12,13] and employ structure MCMC [38,39] to sample DAGs from the posterior distribution.
- **hBe**: Heckerman and Geiger [10] proposed a score-equivalent model for hybrid BNs, which we refer to as the *hybrid Bayesian model for networks having score-equivalence* (**hBe**) model. A DAG captures the dependencies among the continuous variables X_1, \dots, X_n , while the network parameters (regression parameters and error variances) depend on the value combination of the discrete variables $Z = (Z_1, \dots, Z_m)^\top$. The model can be viewed as a multivariate Gaussian mixture model with known labels. It has a mixture component for each possible value combination $z^{(1)}, \dots, z^{(K)}$ of the discrete variables in Z . Effectively, the hBe model computes component-specific BGe scores, and their product is the hBe score. We use structure MCMC [38,39] to sample DAGs from the posterior distribution.
- **clG**: The *conditional linear Gaussian* (**clG**) model describes the relationship between each X_i and its continuous parents in Π_i through a mixture of linear regressions. For each X_i it learns a separate set of discrete parents $\Delta_i \subset \{Z_1, \dots, Z_m\}$. Since the Δ_i are variable-specific, the linear regression mixture models for different variables have distinct components. The clG model was introduced by Lauritzen and Wermuth [20], and it is implemented in the *bnlearn* R package [31,32]. We use hill climbing with multiple restarts and perturbations from *bnlearn* to identify the DAG with the highest clG score.
- **mBGe**: In Section 2.3 we have introduced the *mean-adjusted BGe* (**mBGe**) model. The idea is to describe the relationship between each X_i and its parents in Π_i and Δ_i by linear regression, where both types of parents (continuous vs. discrete) have additive effects. We use the MCMC algorithm from Table 1 for model inference.

Table 1

Algorithmic Description of the MCMC Sampler for the mBGe Model.

Initialize the discrete parent sets $\tilde{\Delta} = \{\Delta_1, \dots, \Delta_n\}$, the regression vector $\tilde{\beta}$, and the DAG $\mathcal{G} = \{\Pi_1, \dots, \Pi_n\}$. For example, initialize with: $\Delta_i = \emptyset \forall i$ and $\tilde{\beta} = \mathbf{0}$, and choose a DAG \mathcal{G} with no edges, i.e. $\Pi_i = \emptyset \forall i$.

Iterations: Loop through the following five sampling steps:

M1a Apply Eq. (9) to compute the mean vectors $\mu_v = \mathbf{Z}_v \tilde{\beta}$, where \mathbf{Z}_v depends on the current $\tilde{\Delta}$ and D_x . Use structure MCMC moves [38,39] to update the current DAG \mathcal{G} . Given \mathcal{G} , sample a neighboring DAG \mathcal{G}_* , where neighbors are defined as DAGs that can be obtained from \mathcal{G} by a single edge addition, deletion, or reversal. Accept the new DAG \mathcal{G}_* with the MH acceptance probability

$$A = \min \left\{ 1, \frac{p(D|\mathcal{G}_*)}{p(D|\mathcal{G})} \cdot \frac{p(\mathcal{G}_*)}{p(\mathcal{G})} \cdot \frac{N(\mathcal{G})}{N(\mathcal{G}_*)} \right\},$$

where the marginal likelihoods $p(D|\dots)$ can be computed analytically, and $N(\mathcal{G})$ and $N(\mathcal{G}_*)$ denote the number of neighbor DAGs of \mathcal{G} and \mathcal{G}_* , respectively. If the move is accepted, replace \mathcal{G} with \mathcal{G}_* .

M1b Use the algorithm from Section 3.2 in Grzegorzczak [36] to sample a new covariance matrix Σ that is consistent with the current DAG \mathcal{G} .

M2a Given Σ , perform a Metropolis-Hastings MCMC step to sample the discrete parent sets in $\tilde{\Delta}$. Select one Δ_i at random. Each of the m discrete variables can either be in or out. Choose one random Z_q . Changing the status of Z_q either by adding it or removing it from Δ_i , yields the new system $\tilde{\Delta}_*$. Accept $\tilde{\Delta}_*$ with the acceptance probability

$$A = \min \left\{ 1, \frac{p(\mathbf{x}|\Sigma, \tilde{\Delta}_*, D_x, \lambda^2)}{p(\mathbf{x}|\Sigma, \tilde{\Delta}, D_x, \lambda^2)} \cdot \frac{p(\tilde{\Delta}_*)}{p(\tilde{\Delta})} \cdot 1 \right\},$$

where the marginal likelihoods $p(\mathbf{x}|\dots)$ can be computed with Eq. (14). If the move is accepted, replace $\tilde{\Delta}$ with $\tilde{\Delta}_*$.

M2b Sample a new regression parameter vector $\tilde{\beta}$ from the full conditional distribution $\tilde{\beta}|\Sigma, \tilde{\Delta}, D_x, \lambda^2, \mathbf{x}$, as given in Eq. (15).

M2c Sample λ^{-2} from its full conditional distribution, given in Eq. (16), and invert it to obtain λ^2 .

Table 2

Number of Parameters for Each Model. When $\Delta_i = \emptyset$ for all i , both the mBGe and the clG models reduce to a form similar to the BGe model. Conversely, when $\Delta_i = \{Z_1, \dots, Z_m\}$ for all i , the clG model reduces to the hBe model.

Hybrid BN model	Number of model parameters
mBGe	$\sum_{i=1}^n \left((\Pi_i + 2) + \sum_{q: (Z_q \in \Delta_i)} (k_q - 1) \right)$
BGe	$\sum_{i=1}^n (\Pi_i + 2)$
hBe	$\left(\prod_{q=1}^m k_q \right) \cdot \sum_{i=1}^n (\Pi_i + 2)$
clG	$\sum_{i=1}^n \left((\Pi_i + 2) \cdot \left\{ \prod_{q: (Z_q \in \Delta_i)} k_q \right\} \right)$

Fig. 1 summarizes the main differences between the four models, and in Section S4 of the supplementary material we compare them in terms of the underlying regression equations. The numbers of model parameters are listed in Table 2. For domains with many discrete variables (large m) and/or discrete variables with many categories (large k_q), the hBe model generally has substantially more parameters than the mBGe model. The number of parameters of the clG model varies, but it is typically larger than for the mBGe model.

3. Simulation details and performance measures

Model priors and hyperparameters: We assume all DAGs \mathcal{G} and all configurations of discrete parent sets $\tilde{\Delta}$ are equally likely, resulting in uniform prior distributions across models. Specifically, we have $p(\mathcal{G}) = c$ for all \mathcal{G} and $p(\tilde{\Delta}) = c$ for all $\tilde{\Delta}$. For the simplified BGe score, we adopt a weak uninformative prior distribution by setting $\mathbf{R} = \mathbf{I}$, where \mathbf{I} is the identity matrix, and using the minimal value for α_w (i.e., $\alpha_w = n + 2$). Under this choice, the covariance matrix Σ in Eq. (4) follows an inverse Wishart prior, $\Sigma^{-1} \sim \mathcal{W}(n + 2, \mathbf{I})$. For the Gamma prior on the variance parameter λ^2 (see Eq. (16)), a common choice is to set $a = b = \alpha$ for a small value of α [43]. Here, we set $\alpha = 0.1$. Additionally, for the BGe and hBe model, we also specify an uninformative prior for the mean vector μ in Eq. (4). We use the prior $\mu \sim \mathcal{N}(\mathbf{0}, \alpha_\mu^{-1} \Sigma)$ with the uninformative setting $\alpha_\mu = 1$.

bnlearn: For the clG model, we use the *bnlearn* package in R and evaluate DAGs using the Bayesian Information Criterion (BIC). For each dataset, we search for the best-scoring model (i.e., the one with the lowest BIC value). We use hill climbing with 200 random restarts at each of five perturbation levels (5, 10, 20, 40, 80), yielding a total of 1,000 restarts per dataset.

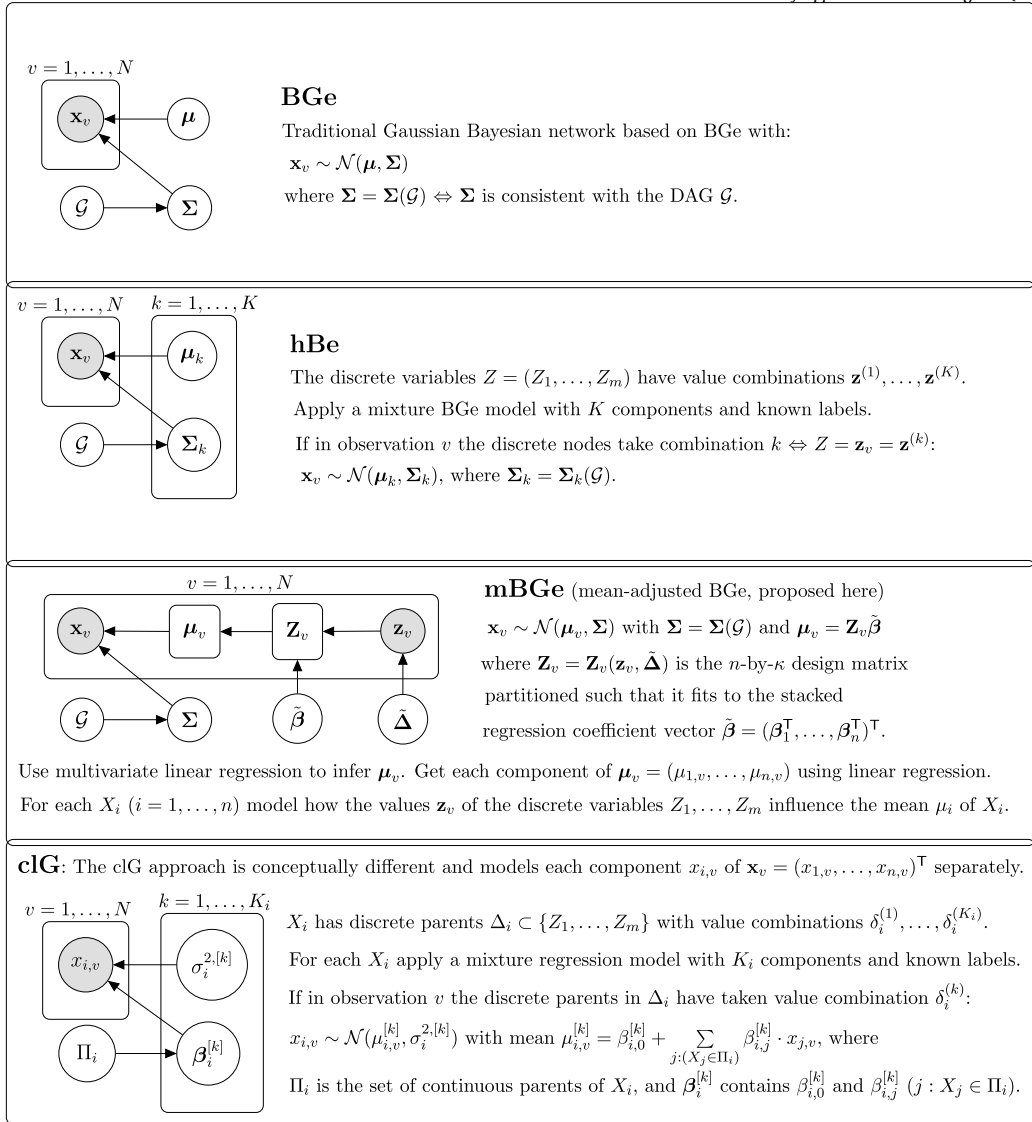


Fig. 1. Graphical Summary of the Four Hybrid BN Models Under Comparison. Each model (BGe, hBe, mBGe, cIG) is presented in a separate panel, including a brief description and a graphical model highlighting the most important relationships. Further details are provided in Sections 2.3 (mBGe) and 2.5 (BGe, hBe, cIG). Section S4 of the supplementary material compares the four models in terms of their corresponding regression equations.

MCMC simulation lengths and edge scores: We run all MCMC simulations for 500k (500,000) iterations, setting the burn-in to 250k (50%) and applying a thinning factor of 250 during the sampling phase, resulting in $R = 1000$ posterior samples. From the graphs, $\mathcal{G}_1, \dots, \mathcal{G}_R$, we compute the marginal posterior probability (or edge score) $e_{i,j} \in [0, 1]$ for each possible edge $X_i \rightarrow X_j$, as described in Section 2.1. In the same vein, we can estimate the marginal posterior probabilities of edges pointing from discrete to continuous variables. We have the posterior sample of discrete parent sets $\tilde{\Delta}_1, \dots, \tilde{\Delta}_R$, where $\tilde{\Delta}_r = \{\Delta_{r,1}, \dots, \Delta_{r,n}\}$. The score $d_{q,i} \in [0, 1]$ of the edge $Z_q \rightarrow X_i$ is the proportion of discrete parent sets $\tilde{\Delta}_r$ in which $Z_q \in \Delta_{r,i}$.

Convergence diagnoses: In preliminary studies, we selected a few datasets and ran independent MCMC simulations to assess convergence. We compared the results using standard MCMC convergence diagnostics, such as trace plots and scatter plots of the estimated edge scores [36,17]. For an MCMC run length of 500k iterations, the trace plots consistently reached the same plateau levels, and the scatter plots of edge scores showed that nearly all points lay very close to the diagonal, indicating that independent MCMC simulations produced nearly identical and thus reliable edge scores.

Edge types in hybrid BNs: There are three types of edges:

(I) edges among the discrete variables,

- (II) edges pointing from the discrete to the continuous variables,
- (III) edges among the continuous variables.

Since edges from continuous to discrete variables are excluded, Type I edges do not interact with edges of Types II and III. This implies that inferring the edges among the discrete variables can be treated as a separate, independent step. For each of the models under comparison, this separate analysis can be carried out using any Bayesian network structure learning method for discrete variables. Since comparing methods for discrete Bayesian networks is beyond the scope of this paper, we focus on the continuous variables and do not infer the edges among the discrete variables. When quantifying network reconstruction accuracy using relative structural Hamming distances (rSHDs), we therefore focus exclusively on edges of Type III. However, we note that continuous variables can have both continuous and discrete variables as parents, leading to interdependencies. As a result, the presence or absence of Type II edges may influence Type III edges, and vice versa.

Structural Hamming distances: Tsamardinos et al. [44] define the structural Hamming distance (SHD) between two CPDAGs as the minimum number of *single edge operations* required to transform from one CPDAG into the other.⁴ If one of the two CPDAGs represents the true graph and the other is a predicted CPDAG, the relative structural Hamming distance (rSHD) is defined as the SHD divided by the number of edges in the true CPDAG. A lower rSHD indicates higher accuracy of the predicted CPDAG. While the cIG model learns a DAG whose CPDAG can be computed and compared with the CPDAG of the true DAG, the Bayesian approaches yield edge scores $e_{i,j}$. We impose the threshold $\psi = 0.5$ and extract only those edges whose scores exceed ψ . The extracted edges $\{X_i \rightarrow X_j : e_{i,j} > 0.5\}$ define a network. For $e_{i,j} > 0.5$ and $e_{j,i} > 0.5$ the network possesses the bi-directional edges $X_i \leftrightarrow X_j$. We follow [17] and interpret $X_i \leftrightarrow X_j$ as an undirected edge $X_i - X_j$. This renders the predicted network a CPDAG, which can be compared with the true CPDAG in terms of the rSHD score.

Predictive probabilities: The Bayesian models (BGe, hBe, and mBGe) can be compared in terms of their posterior predictive probabilities, while for the cIG model, we compute predictive likelihoods, which are comparable to (posterior) predictive probabilities [45].

For the Bayesian models, MCMC simulations generate posterior samples given the data \mathcal{D} . Each sample consists of a model \mathcal{M}_r and a corresponding parameter set Θ_r that is consistent with \mathcal{M}_r ($r = 1, \dots, R$). The predictive probability for a new dataset with N^* observations of the continuous and discrete variables, denoted by $\mathcal{D}^* := \{\mathbf{x}_1^*, \dots, \mathbf{x}_{N^*}^*\}$ and $\mathcal{D}_z^* := \{\mathbf{z}_1^*, \dots, \mathbf{z}_{N^*}^*\}$, can be approximated using Monte Carlo methods. The likelihood of the new validation data \mathcal{D}^* is averaged over the models and parameters obtained from the posterior distribution based on the old data \mathcal{D} :

$$\hat{p}(\mathcal{D}^* | \mathcal{D}) := \frac{1}{R} \sum_{r=1}^R p(\mathcal{D}^* | \mathcal{M}_r, \Theta_r) = \frac{1}{R} \sum_{r=1}^R \left(\prod_{v=1}^{N^*} p(\mathbf{x}_v^* | \mathcal{M}_r, \Theta_r) \right).$$

To make predictive probabilities for validation datasets with different sample sizes N^* comparable, we compute the geometric mean predictive probabilities by raising the predictive probabilities (or predictive likelihoods) to the power of $1/N^*$:

$$\hat{p}_*(\mathcal{D}^* | \mathcal{D}) := \hat{p}(\mathcal{D}^* | \mathcal{D})^{\frac{1}{N^*}}.$$

For a more detailed description of how we compute predictive probabilities and likelihoods, we refer to Section S5 of the supplementary material.

Software: Our MATLAB implementation of the mBGe model is available at: <https://github.com/MarcoAndreas/mBGe>

4. Examples illustrating limitations of the models

In this section, we use toy examples to illustrate specific shortcomings of the cIG and hBe models. The accompanying empirical results are provided in Sections S6 and S7 of the supplementary material.

4.1. Shortcoming of the cIG model

The cIG model is characterized by two key assumptions (see, e.g., [8,33]). First, it allows continuous nodes to have discrete parent nodes. Second, it models the effect of discrete parent nodes on continuous child nodes using mixtures of linear regressions. To our knowledge, it has not yet been noted in the literature that the cIG model makes another critical assumption: all edges pointing from discrete to continuous variables must be compelled in their direction. That is, it is essential that these edges do not contradict their natural direction. To illustrate this shortcoming of the cIG model, we use a toy example. We generate data for three continuous variables as follows:

⁴ The single edge operations include: (i) adding or deleting a directed or an undirected edge, (ii) orienting an undirected edge, and (iii) reversing a directed edge or converting it into an undirected edge.

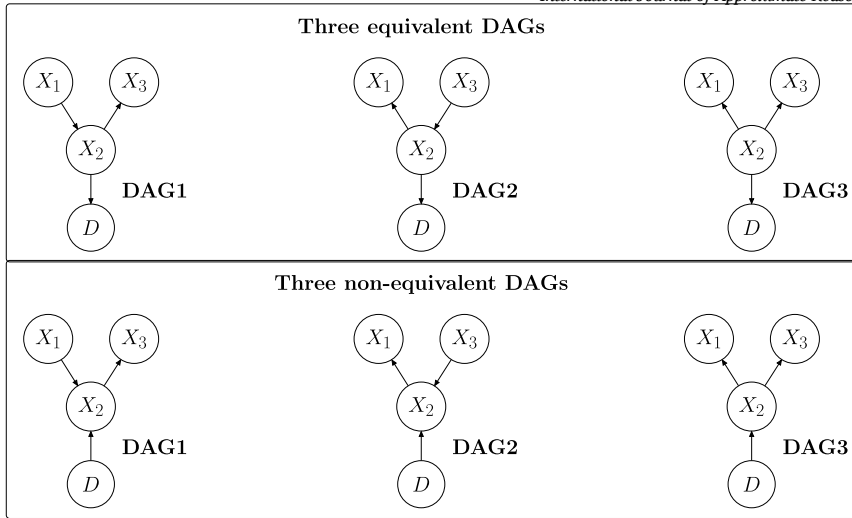


Fig. 2. Graphical Illustration 1. Top panel: With $X_2 \rightarrow D$ the three DAGs are equivalent Bottom panel: The clG model enforces the edge between D and X_2 to point in the opposite direction. This breaks the equivalence and may lead to incorrect conclusions.

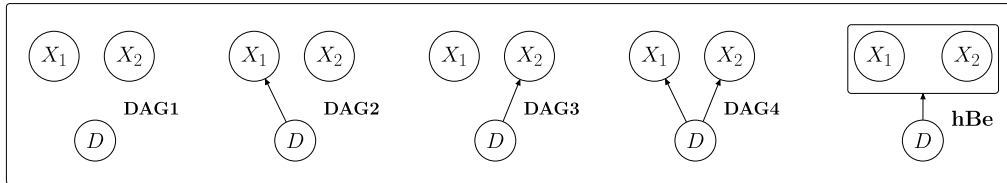


Fig. 3. Graphical Illustration 2. In DAG1, DAG2, and DAG3, X_1 and X_2 are marginally independent. In DAG4, X_1 and X_2 are independent only when conditioned on D . In all four cases, the hBe model infers the same DAG without an edge between X_1 and X_2 , making it appear that X_1 and X_2 are marginally independent. However, because the hBe score accounts for all potential effects of the discrete variables, it remains unclear whether X_1 and/or X_2 depend on D , meaning the four DAGs cannot be distinguished.

$$X_2 = \epsilon_2 \text{ and } X_i = X_2 + 0.1 \cdot \epsilon_i \text{ (} i = 1, 3 \text{) where } \epsilon_1, \epsilon_2, \epsilon_3 \sim \mathcal{N}(0, 1).$$

In addition, we define a discrete variable D , whose values depend on X_2 :

$$D = 0 \text{ for } X_2 < 0, \text{ and } D = 1 \text{ otherwise.}$$

These equations correspond to DAG3 in the top panel of Fig. 2. Since the DAGs in the upper panel are equivalent, the relations among the variables can also be expressed using equations corresponding to DAG1 or DAG2. These alternative formulations are provided in Section S6 of the supplementary material. Hence, the three DAGs in the top panel of Fig. 2 are indistinguishable, making it impossible to determine the directions of the edges between X_1 and X_2 , and between X_2 and X_3 . However, as shown in Section S6 of the supplementary material, the clG model assigns higher scores to DAG3 than to DAG1 and DAG2. This suggests that the data favor the edges $X_2 \rightarrow X_1$ and $X_2 \rightarrow X_3$. However, this result arises solely because the clG model forces the edge from D to point toward X_2 , contradicting the true direction and breaking the equivalence class.

The new mBGe model preserves the equivalence class, because, it is essentially the simplified BGe score applied to mean-adjusted variables. After the removal of D , there is no difference between the two panels of Fig. 3. For more details we refer to Section S6 of the supplementary material.

4.2. Shortcoming of the hBe model

In the hBe model, all discrete variables influence all continuous variables. As a result, any (conditional) independencies implied by the DAG hold only when conditioning on all discrete variables. Fig. 3 presents an example with four DAGs representing different scenarios. The hBe model cannot distinguish between them; it only detects that X_1 and X_2 are conditionally independent given D . However, it remains unclear whether X_1 or X_2 depend on D , and consequently whether they are marginally independent. In Section S7 of the supplementary material, we analyze synthetic data to illustrate this shortcoming.

Table 3

Average rSHD Scores for Synthetically Generated Network Data. N denotes the number of observations, while $E|\mathcal{G}|$ and $E|\tilde{\Delta}|$ refer to the expected number of edges in the DAG \mathcal{G} and in the discrete parent sets $\tilde{\Delta}$, respectively. All rSHD scores are averaged over 20 independent data instantiations and reflect the accuracy of learning the DAG among the continuous variables. The lowest average rSHD scores —along with those not significantly different from it according to a paired t -test at the $\alpha = 5\%$ level — are shown in bold. Data were generated as described in Appendix A, giving the mBGe model a natural advantage.

N	$E \mathcal{G} $ $E \tilde{\Delta} $	10 0	10 2.5	10 5	10 10	20 0	20 2.5	20 5	20 10
50	BGe	0.40	0.50	0.62	0.77	0.63	0.64	0.89	0.85
	hBe	1.74	1.42	1.35	1.52	1.21	1.21	1.12	1.09
	clG	0.83	0.92	0.91	1.20	0.84	0.93	1.09	1.07
	mBGe	0.41	0.40	0.38	0.43	0.62	0.62	0.69	0.63
100	BGe	0.15	0.30	0.53	0.71	0.49	0.57	0.59	0.86
	hBe	1.43	1.08	1.07	1.21	0.88	0.94	0.84	0.93
	clG	0.52	0.50	0.66	0.85	0.54	0.82	0.80	1.03
	mBGe	0.16	0.12	0.28	0.15	0.53	0.42	0.37	0.55
200	BGe	0.16	0.23	0.56	0.83	0.29	0.38	0.66	0.81
	hBe	0.33	0.25	0.31	0.34	0.56	0.56	0.71	0.67
	clG	0.24	0.44	0.76	0.60	0.35	0.59	0.88	0.97
	mBGe	0.16	0.11	0.14	0.19	0.31	0.21	0.42	0.43
400	BGe	0.05	0.25	0.47	1.00	0.20	0.42	0.62	0.95
	hBe	0.19	0.18	0.25	0.17	0.59	0.66	0.79	0.71
	clG	0.14	0.41	0.48	0.69	0.24	0.59	0.89	1.02
	mBGe	0.05	0.05	0.08	0.11	0.20	0.23	0.33	0.42
800	BGe	0.06	0.31	0.56	1.26	0.05	0.41	0.85	1.07
	hBe	0.31	0.35	0.10	0.23	0.82	0.74	0.65	0.83
	clG	0.15	0.17	0.43	0.65	0.19	0.62	0.84	1.13
	mBGe	0.07	0.13	0.13	0.21	0.10	0.22	0.31	0.40

5. Empirical results

We present the key empirical results in this section. Further results can be found in Sections S8 to S10 of the supplementary material.

5.1. Network structure learning

We compare the four hybrid Bayesian network models in terms of their network reconstruction accuracy. For real-world data, where the true network (CPDAG) is unknown, the accuracy of the learned networks cannot be directly assessed. For synthetic data, the true network is known; however, the model that best matches the data-generating mechanism has a natural advantage. Therefore, the goal of this study is not to compare overall model performance across settings. Instead, we generate synthetic data aligned with the mBGe model to demonstrate its ability to uncover relationships missed by the competing models.

A detailed description of how we generate datasets with $n = 10$ continuous and $m = 5$ discrete variables can be found in Appendix A. We vary the number of observations N as well as the expected numbers of edges $E|\mathcal{G}|$ and $E|\tilde{\Delta}|$ of the DAG \mathcal{G} and the discrete parent sets $\tilde{\Delta}$.

Our focus is on learning the CPDAG among the continuous variables, and we quantify the network reconstruction accuracy using relative structural Hamming distance (rSHD) scores; see Section 3 for more details. In Section S8 of the supplementary material, we evaluate model performance using a complementary criterion by computing the Kullback–Leibler (KL) divergences [46] between the true Gaussian mixture distribution (used to generate the data) and the inferred Gaussian mixture distributions.

For each combination of N , $E|\mathcal{G}|$, and $E|\tilde{\Delta}|$, we generate 20 independent data instantiations. The average rSHD scores are reported in Table 3. Given that the data generation mechanism (see Appendix A) is aligned with the mBGe model, it is not surprising that mBGe generally performs best. As expected, rSHD scores decrease with increasing sample size N and increase with higher network complexity, that is, with larger values of $E|\mathcal{G}|$ and $E|\tilde{\Delta}|$. In the absence of discrete parents ($E|\tilde{\Delta}| = 0$), the performance of the conventional BGe model [12] does not significantly differ from that of the new mBGe model. The same holds for the clG model when the sample size is sufficiently large ($N \geq 200$). This is an important finding, as it shows that the mBGe model, as well as the clG model for larger sample sizes, is not negatively affected by the presence of irrelevant discrete variables. In such cases, the models effectively behave like Gaussian Bayesian networks (cf. BGe). We observe similar trends when evaluating the performances in terms of Kullback–Leibler divergences, as reported in Section S8 of the supplementary material.

Table 4

Benchmark Real-World Data. In each dataset, every continuous variable was z-score standardized to have a mean of 0 and a variance of 1. For additional details, see the main text and Section S9 of the supplementary material.

Dataset	Abbrev.	n	m	N_{all}	Sect.	Fig.
1. Indian Liver Disease	ILD	8	2	579	5.2	5
2. Framingham Heart Study	FHS	6	4	2788	5.2	6
3. Pima Indians Diabetes	PID	5	4	392	S8	3
4. Liver Disorder Data	LDD	5	1	345	S8	3
5. Boston Housing Data	BHD	13	1	506	S8	4
6. Australian Credit Approval	ACA	6	6	650	S8	4

5.2. Predictive probabilities

Since the true networks are unknown for real-world data, we use predictive probabilities to evaluate model performance. We compute these predictive probabilities for the six datasets listed in Table 4. To avoid scale and offset effects, we z-score standardize each continuous variable within each dataset to have a mean of 0 and a variance of 1. Each dataset comprises N_{all} observations, which we split into a training set D and a validation set D^* . For the training set, we randomly sample $N \in \{50, 100, 200, 400\}$ observations and use the remaining $N^* = N_{all} - N$ observations for validation. To ensure comparability across validation sets of different sizes N^* , we compute the geometric mean predictive probabilities, as described in Section 3.

In this section of the main paper, we focus on the Indian Liver Disease (ILD) and Framingham Heart Study (FHS) datasets. Predictive probabilities for the remaining datasets (PID, LDD, BHD, and ACA) are provided in Section S9 of the supplementary material.

Geometric Mean Predictive Probabilities for ILD Data:

The Indian Liver Disease (ILD) data from Ramana et al. [47] consists of $N = 579$ complete observations (patients) and the following $n = 8$ continuous variables: total proteins, albumin, ratio albumin:globulin, alkaline phosphatase (PPT), total bilirubin, direct bilirubin, aspartate aminotransferase (AST), and alanine aminotransferase (ALT). We also include the discrete variable ‘Sex’ (male vs. female), and construct a categorical ‘Age’ variable via quantile discretization into three intervals: [04, 38], [38, 51], or [51, 90] years.

The geometric mean predictive probabilities for $N \in \{100, 200, 400\}$ are shown in Fig. 4. Most notably, the cIG model produces significantly lower predictive probabilities than the other three models, and no model outperforms the new mBGe model. As expected, predictive probabilities increase with sample size N , while relative performance differences between models tend to diminish. For the two smaller sample sizes, the mBGe model yields significantly higher predictive probabilities than the hBe model. Interestingly, across all three sample sizes, the BGe and mBGe models perform very similarly.

Geometric Mean Predictive Probabilities for FHS Data:

The Framingham Heart Study (FHS) data were collected from residents of the town of Framingham, Massachusetts (USA). In this study, we use the pre-processed version available on Kaggle.⁵ After excluding participants with prevalent stroke, hypertension, diabetes, or those taking blood pressure medication, the sample size is reduced to $N = 2788$. The dataset includes the following $n = 6$ continuous variables: Cholesterol level, body mass index (BMI), diastolic blood pressure (DIA PRESS), systolic blood pressure (SYS PRESS), heart rate, and glucose level. There are three binary discrete variables: ‘Sex’ (male vs. female), ‘Smoking’ (yes vs. no), and ‘Education’ (up to High School vs. College degree and higher). In addition, we include ‘Age’ as a categorical variable discretized into four intervals: [32, 40), [40, 50), [50, 59), or [60, 69] years.

Fig. 5 presents the geometric mean predictive probabilities for the sample sizes $N \in \{100, 200, 400\}$. As with the ILD data, predictive probabilities increase with N , while model differences shrink. In contrast to the ILD data, where the cIG model performed relatively poorly, here the hBe model yields substantially lower predictive probabilities than the other models. Across all three sample sizes, the new mBGe model achieves significantly higher predictive probabilities. For the smallest sample size, $N = 100$, the cIG model performs worse than the BGe model, whereas at $N = 400$, the cIG model outperforms the BGe model.

5.3. Learning networks from real-world data

In this subsection, we apply the proposed mBGe model to learn hybrid Bayesian networks from the Indian Liver Disease (ILD) and Framingham Heart Study (FHS) datasets. In both cases, we apply the mBGe model to the complete dataset. The resulting networks are shown in Figs. 6 and 7. In both figures, continuous variables are represented as gray nodes, while discrete variables are listed in a gray rectangle. Edges represent dependencies among continuous variables; labels on these nodes indicate which discrete variables influence their means. Only dependencies with a marginal posterior probability exceeding the threshold $\psi = 0.5$ are displayed.

Fig. 6 presents the network inferred from the Indian Liver Disease (ILD) data. The continuous variables correspond to key biochemical liver markers. The discrete variables Age and Sex influence the means of five and three of the continuous variables, respectively. The graph among the $n = 8$ mean-adjusted continuous variables contains nine edges, three of which are directed. These directed edges point towards the *Albumin* node, suggesting that its levels are influenced by *total proteins*, the albumin-to-globulin ratio, and *direct bilirubin*.

⁵ <https://www.kaggle.com/code/captainozlem/framingham-chd-preprocessing-data>.

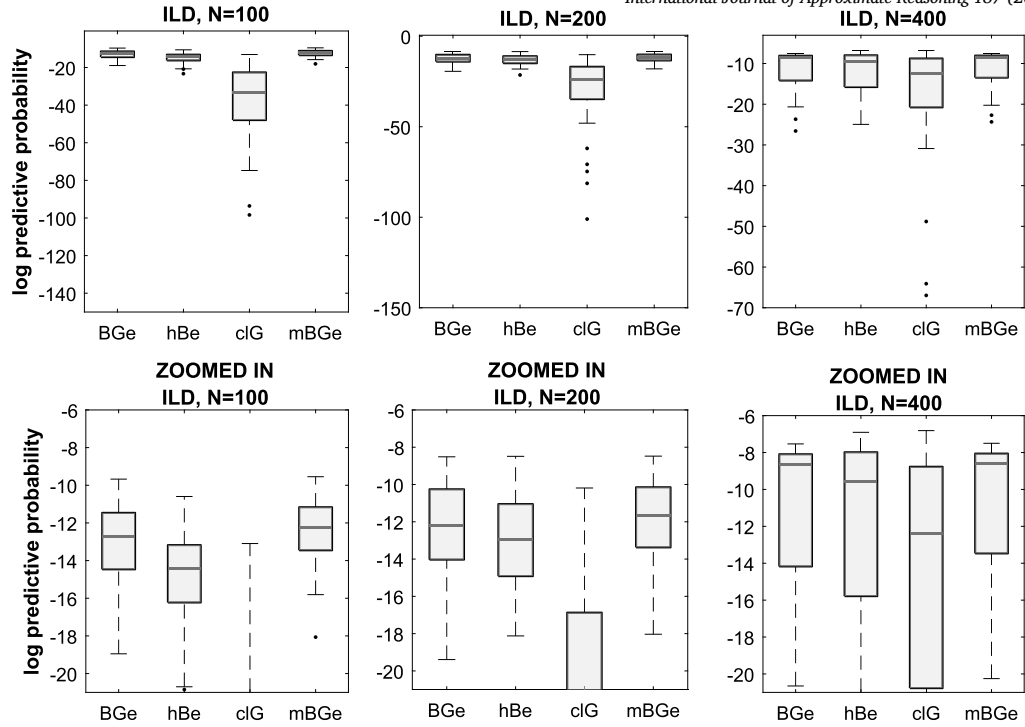


Fig. 4. Geometric Mean Predictive Probabilities (PPs) for Indian Liver Disease (ILD) Data. Each boxplot shows the distribution of log PPs computed from 100 randomly sampled data subsets, with PPs rescaled to correspond to a single observation. The columns represent three sample sizes N . Because the clG model yields much lower PPs, the bottom row provides a zoomed-in view for better comparison.

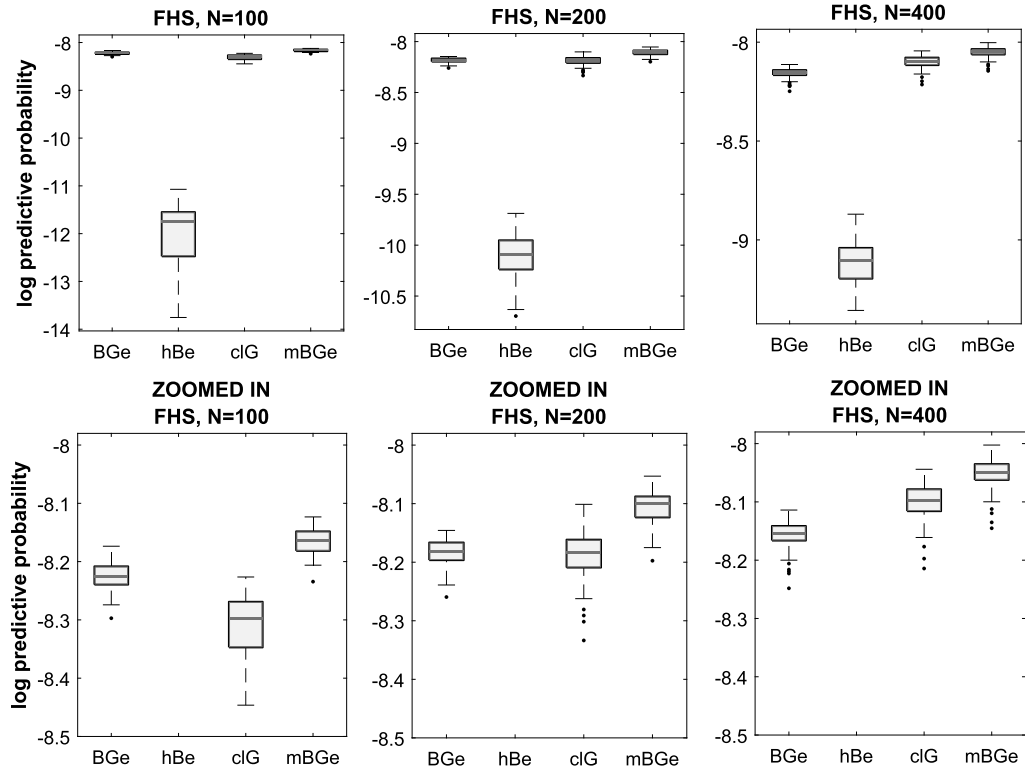


Fig. 5. Geometric Mean Predictive Probabilities (PPs) for the Framingham Heart Study (FHS) Data. Each boxplot shows the distribution of log PPs computed from 100 randomly sampled data subsets, with PPs rescaled to correspond to a single observation. The columns represent three sample sizes N . Since the hBe model yields substantially lower PPs, the bottom row provides a zoomed-in view for better comparison.

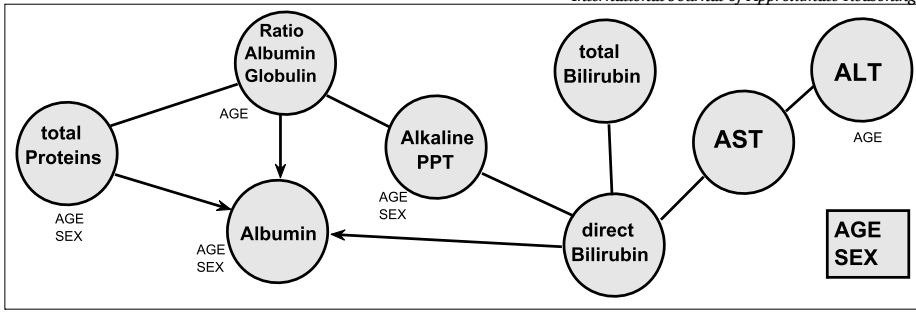


Fig. 6. Indian Liver Disease (ILD) Network. The network includes $n = 8$ continuous variables (nodes) and two discrete variables. The graph contains nine edges, three of which are directed. Node labels indicate which discrete variables influence their means.

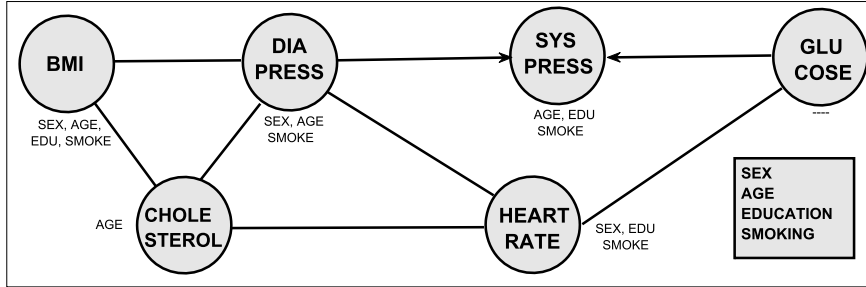


Fig. 7. Framingham Heart Study (FHS) Network. The network includes $n = 6$ continuous variables (nodes) and $m = 4$ discrete variables. The graph contains eight edges, two of which are directed. Labels show which discrete variables influence their means.

Fig. 7 presents the network inferred from the Framingham Heart Study (FHS) data. The graph among the $n = 6$ mean-adjusted continuous variables contains eight edges, two of which are directed. These directed edges suggest that systolic blood pressure (*SYS PRESS*) depends on both diastolic blood pressure (*DIA PRESS*) and *glucose* level. The mean of body mass index (*BMI*) is adjusted for the effects of all four discrete variables, whereas the mean *glucose* level appears to be independent of any discrete variable. The remaining four continuous variables are each influenced by one to three discrete variables.

In Section S10 of the supplementary material, we compare the results of the mBGe model with those of the other models. In summary, we observe that the inferred network structures differ notably across models, suggesting that they capture different relationships among the variables. Only the BGe and mBGe models yield similar DAGs among the continuous variables. This is consistent with our expectations, as the mBGe model can be viewed as a refined version of the BGe model that adjusts the means of the continuous variables for the effects of the discrete variables. Our results in Sections 5.1 and 5.2 indicate that the mBGe model may be particularly advantageous for small sample sizes. This finding is in line with our expectations, since the mBGe model requires fewer parameters to learn, which can therefore be more reliably inferred from limited training data.

6. Conclusions and discussion

We have developed a novel Bayesian model for learning Bayesian network structures from hybrid data. Existing approaches typically rely on mixture models, where discrete variable values define the mixture components. In the hBe model, each mixture component corresponds to a separate Gaussian Bayesian network over all continuous variables. In the clG model, the relationships between continuous variables and their continuous parents are modeled using mixture regression models. Our proposed model avoids mixture modeling and instead employs additive linear modeling. Multivariate linear regression is used to adjust the means of the continuous variables for the potential effects of discrete variables, while simultaneously learning a Gaussian Bayesian network (DAG) among them.

The new model builds on the well-known BGe score for Gaussian Bayesian networks and extends it by incorporating the effects of discrete variables on the mean vectors. This leads to an extended BGe score with observation-specific mean vectors, which we refer to as the mean-adjusted BGe (mBGe) model. We also propose an MCMC algorithm for sampling both network structures and their associated parameters.

In practical applications, the choice of the most appropriate model depends on the true underlying dependency mechanisms, which are typically unknown. Therefore, we do not claim that the new model is universally superior, but rather present it as a strong alternative to existing approaches. We see three potential advantages of the mBGe model: First, we believe that additive modeling is more natural than mixture modeling. Second, the mBGe model is sparser, meaning it involves fewer parameters than the existing models (see Table 2). Third, it avoids conceptual issues. Unlike the clG model, it does not yield erroneous results when edges should point from continuous to discrete variables (see Section 4.1). Unlike the hBe model, the mBGe model learns which discrete variables influence which continuous variables, allowing for marginal independencies among the continuous variables (see Section 4.2).

A disadvantage of the mBGe model is its increased computational cost. In particular, compared to the cIG model, the mBGe model does not scale as well. Model inference relies on MCMC simulations, which require convergence to the posterior distribution. The number of MCMC iterations needed to reach convergence increases with the number of continuous variables n and the number of discrete variables m . Additionally, the computational cost of each individual MCMC iteration depends on the number of observations N and the number of regression coefficients κ , which increases with the number of discrete-to-continuous edges. A table presenting the measured computational costs is provided in Section S11 of the supplementary material.

The cIG model enforces edges between discrete and continuous variables to point towards the continuous variables. If the true edges need the opposite direction, the cIG model disturbs existing equivalence classes, potentially leading to erroneous results. In our future work, we may explore possibilities of developing a score-equivalent frequentist counterpart of the Bayesian mBGe model.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

I thank the reviewers, and in particular the handling Area Editor, for their constructive feedback, and Dr. Marco Scutari for his prompt and helpful support with *bnlearn*. This paper completes the *Being Bayesian Trilogy*, which started in 2023 [36] and continued in 2024 [17]. I dedicate the trilogy to Prof. em. Dr. Wolfgang Urfer (TU Dortmund University, Germany), who set the topic of my PhD project (2003–2006) on Bayesian networks, and to Prof. Dr. Dirk Husmeier (University of Glasgow, UK), who introduced me to the field.

Appendix A. Generating synthetic datasets

We generate synthetic data with n continuous and m discrete variables, imposing relationships in line with the mBGe model. To begin, we determine a random topological ordering of the continuous variables; without loss of generality, we label them X_1, \dots, X_n . There are $n(n-1)/2$ possible edges among the continuous variables and $n \cdot m$ possible edges from discrete to continuous variables. Letting each edge $X_i \rightarrow X_j$ with $i < j$ be present with probability p_1 yields a DAG \mathcal{G} with an expected number of edges given by $E|\mathcal{G}| = \frac{n(n-1)}{2} \cdot p_1$. Correspondingly, letting each edge $Z_q \rightarrow X_i$ be present with probability p_2 yields a system of discrete parent sets $\tilde{\Delta}$ with an expected number of edges given by $E|\tilde{\Delta}| = (n \cdot m)p_2$. We generate hybrid Bayesian network data in two steps.

1. First, we generate mean-adjusted values of continuous variables. For the mean-adjusted values $y_{i,v} := x_{i,v} - \mu_{i,v}$, we have:

$$y_{i,v} = \sum_{j: (X_j \in \Pi_i)} b_{i,j} y_{j,v} + \epsilon_{i,v},$$

where $\epsilon_{i,v} \sim \mathcal{N}(0, \sigma_i^2)$. This model includes $|\Pi_i| + 1$ parameters: the regression coefficients $b_{i,j}$ and the noise standard deviation σ_i . We sample their values from uniform distributions on $[1, 2]$ and assign a random sign to each $b_{i,j}$. We then rescale the parameters to have a Euclidean norm of 1, such that: $\sigma_i^2 + \sum_{j: (X_j \in \Pi_i)} b_{i,j}^2 = 1$. For each observation v , the values $y_{i,v}$ need to be generated in topological order.

2. Second, we generate binary discrete values by sampling them from Bernoulli distributions, $z_{q,v} \sim \text{BER}(0.5)$. For the means, we set:

$$\mu_{i,v} = \beta_{i,0} + \sum_{q: (Z_q \in \Delta_i)} \theta_{i,q}^{z_{q,v}}.$$

$\theta_{i,q}$ is added to the intercept $\beta_{i,0}$ if $z_{q,v} = 1$. There are $|\Delta_i| + 1$ parameters, $\theta_{i,\cdot}$ and $\beta_{i,0}$. We sample each parameter from a uniform distribution on $[1, 2]$ and assign a random sign to each. These parameters are also re-scaled to have a Euclidean norm of 1, $\beta_{i,0}^2 + \sum_{q: (Z_q \in \Delta_i)} \theta_{i,q}^2 = 1$, in order to ensure comparable effect sizes for continuous and discrete variables.

Each observation v ($v = 1, \dots, N$) then consists of two vectors:

$\mathbf{x}_v = (x_{1,v}, \dots, x_{n,v})^\top$ and $\mathbf{z}_v = (z_{1,v}, \dots, z_{m,v})^\top$, where each continuous value is given by $x_{i,v} = y_{i,v} + \mu_{i,v}$.

Appendix B. Supplementary material

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.ijar.2025.109549>.

Data availability

Data will be made available on request.

References

- [1] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, San Francisco, CA, USA, 1988.
- [2] R. Neapolitan, Probabilistic Reasoning in Expert Systems: Theory and Algorithms, CreateSpace, Scotts Valley CA, USA, 1989.
- [3] D. Koller, N. Friedman, Probabilistic Graphical Models: Principles and Techniques, Adaptive Computation and Machine Learning Series, MIT Press, Cambridge, MA, USA, ISBN 9780262013192, 2009.
- [4] M. Scutari, J.-B. Denis, Bayesian Networks: With Examples in R, Chapman & Hall, Boca Raton, FL, 2014.
- [5] D.M. Chickering, Learning Bayesian networks is NP-complete, in: D. Fisher, H.J. Lenz (Eds.), Learning from Data: Artificial Intelligence and Statistics, vol. 5, Springer, New York, 1996, pp. 121–130.
- [6] J. Kuipers, G. Moffa, Partition MCMC for inference on acyclic digraphs, J. Am. Stat. Assoc. 112 (2017) 282–299.
- [7] M. Scutari, C.E. Graafland, J.M. Gutiérrez, Who learns better Bayesian network structures: constraint-based, score-based or hybrid algorithms?, in: International Conference on Probabilistic Graphical Models, PMLR, 2018, pp. 416–427.
- [8] M. Scutari, Bayesian network models for incomplete and dynamic data, Stat. Neerl. 74 (2020) 397–419.
- [9] N.K. Kitson, A.C. Constantinou, Z. Guo, Y. Liu, K. Chobtham, A survey of Bayesian network structure learning, Artif. Intell. Rev. 56 (2023) 8721–8814.
- [10] D. Heckerman, D. Geiger, Learning Bayesian networks: a unification for discrete and Gaussian domains, in: Proceedings of the 11th Annual Conference on Uncertainty in Artificial Intelligence (UAI-95), Morgan Kaufmann, San Francisco, CA, 1995, pp. 274–282.
- [11] D. Geiger, D. Heckerman, Learning Gaussian networks, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann, San Francisco, CA, 1994, pp. 235–243.
- [12] D. Geiger, D. Heckerman, Parameter priors for directed acyclic graphical models and the characterization of several probability distributions, Ann. Stat. 30 (2002) 1412–1440.
- [13] J. Kuipers, G. Moffa, D. Heckerman, Addendum on the scoring of Gaussian directed acyclic graphical models, Ann. Stat. 42 (2014) 1689–1691.
- [14] G.F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Mach. Learn. 9 (1992) 309–347.
- [15] D. Heckerman, D. Geiger, D.M. Chickering, Learning Bayesian networks: the combination of knowledge and statistical data, Mach. Learn. 20 (1995) 245–274.
- [16] X. Luo, G. Moffa, J. Kuipers, Learning Bayesian networks from ordinal data, J. Mach. Learn. Res. 22 (2021) 1–44.
- [17] M. Grzegorzcyk, Being Bayesian about learning Bayesian networks from ordinal data, Int. J. Approx. Reason. 170 (2024) 109205.
- [18] F. Castelletti, Learning Bayesian networks: a Copula approach for mixed-type data, Psychometrika 89 (2024) 658–686.
- [19] D. Chickering, D. Geiger, D. Heckerman, Learning Bayesian networks: search methods and experimental results, Mach. Learn. 20 (1995) 112–128.
- [20] S. Lauritzen, N. Wermuth, Graphical models for associations between variables, some of which are qualitative and some quantitative, Ann. Stat. 17 (1989) 31–57.
- [21] S. Lauritzen, Propagation of probabilities, means and variances in mixed graphical association models, J. Am. Stat. Assoc. 87 (1992) 1098–1108.
- [22] S. Lauritzen, F. Jensen, Stable local computation with conditional Gaussian distributions, Stat. Comput. 11 (2001) 191–203.
- [23] R. Cowell, Local propagation in conditional Gaussian Bayesian networks, J. Mach. Learn. Res. 6 (2005) 1517–1550.
- [24] D. Koller, U. Lerner, D. Anguelov, A general algorithm for approximate inference and its application to hybrid Bayes nets, in: K. Laskey, H. Prade (Eds.), Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, Morgan & Kauffman, 1999, pp. 324–333.
- [25] S. Moral, R. Rumi, A. Salmerón, Mixtures of truncated exponentials in hybrid Bayesian networks, in: S. Benferhat, P. Besnard (Eds.), Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Springer, Berlin Heidelberg, Berlin, Heidelberg, 2001, pp. 156–167.
- [26] V. Romero, R. Rumi, A. Salmerón, Learning hybrid Bayesian networks using mixtures of truncated exponentials, Int. J. Approx. Reason. 42 (2006) 54–68.
- [27] H. Langseth, T. Nielsen, R. Rumi, A. Salmerón, Inference in hybrid Bayesian networks, Reliab. Eng. Syst. Saf. 94 (2009) 1499–1509.
- [28] A. Salmerón, R. Rumi, H. Langseth, T. Nielsen, A. Madsen, A review of inference algorithms for hybrid Bayesian networks, J. Artif. Intell. Res. 62 (2018) 1499–1509.
- [29] N. Friedman, M. Goldszmidt, Discretizing continuous attributes while learning Bayesian networks, in: Proc. 13'th International Conference on Machine Learning (ICML), 1996, pp. 157–165.
- [30] S. Monti, G. Cooper, A multivariate discretization method for learning Bayesian networks from mixed data, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI1998), 2013, pp. 404–413.
- [31] M. Scutari, Learning Bayesian networks with the bnlearn R package, J. Stat. Softw. 35 (2010) 1–22.
- [32] M. Scutari, Bayesian network constraint-based structure learning algorithms: parallel and optimized implementations in the bnlearn R package, J. Stat. Softw. 77 (2017) 1–20.
- [33] T. Bodewes, M. Scutari, Learning Bayesian networks from incomplete data with the node-average likelihood, Int. J. Approx. Reason. 138 (2021) 145–160.
- [34] F. Rijmen, Bayesian networks with a logistic regression model for the conditional probabilities, Int. J. Approx. Reason. 48 (2008) 659–666, In Memory of Philippe Smets (1938–2005).
- [35] D.M. Chickering, Learning equivalence classes of Bayesian-network structures, J. Mach. Learn. Res. 2 (2002) 445–498.
- [36] M. Grzegorzcyk, Being Bayesian about learning Gaussian Bayesian networks from incomplete data, Int. J. Approx. Reason. 160 (2023) 108954.
- [37] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, Singapore, 2006.
- [38] D. Madigan, J. York, Bayesian graphical models for discrete data, Int. Stat. Rev. 63 (1995) 215–232.
- [39] P. Giudici, R. Castelo, Improving Markov chain Monte Carlo model search for data mining, Mach. Learn. 50 (2003) 127–158.
- [40] P. Hoff, A First Course in Bayesian Statistical Methods, Springer, New York, 2009.
- [41] N. Friedman, D. Koller, Being Bayesian about network structure: a Bayesian approach to structure discovery in Bayesian networks, Mach. Learn. 50 (2003) 95–126.
- [42] M. Grzegorzcyk, D. Husmeier, Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move, Mach. Learn. 71 (2008) 265–305.
- [43] A. Gelman, Prior distributions for variance parameters in hierarchical models (comment on article by Browne and Draper), Bayesian Anal. 1 (2006) 515–534.
- [44] I. Tsamardinos, L. Brown, C. Aliferis, The max-min hill-climbing Bayesian network structure learning algorithm, Mach. Learn. 65 (2006) 31–78.
- [45] J. Bjørnstad, Predictive likelihood: a review, Stat. Sci. 5 (1990) 242–254.
- [46] S. Kullback, R. Leibler, On information and sufficiency, Ann. Math. Stat. 22 (1951) 79–86.
- [47] B. Ramana, M. Surendra, P. Babu, N. Bala Venkateswarlu, A critical comparative study of liver patients from USA and India: an exploratory analysis, Int. J. Comput. Sci. 9 (2012) 506–516.