



Article

Entropy and the Kullback–Leibler Divergence for Bayesian Networks: Computational Complexity and Efficient Implementation

Marco Scutari

Istituto Dalle Molle di Studi Sull'Intelligenza Artificiale (IDSIA), 6900 Lugano, Switzerland; scutari@bnlearn.com

Abstract: Bayesian networks (BNs) are a foundational model in machine learning and causal inference. Their graphical structure can handle high-dimensional problems, divide them into a sparse collection of smaller ones, underlies Judea Pearl's causality, and determines their explainability and interpretability. Despite their popularity, there are almost no resources in the literature on how to compute Shannon's entropy and the Kullback–Leibler (KL) divergence for BNs under their most common distributional assumptions. In this paper, we provide computationally efficient algorithms for both by leveraging BNs' graphical structure, and we illustrate them with a complete set of numerical examples. In the process, we show it is possible to reduce the computational complexity of KL from cubic to quadratic for Gaussian BNs.

Keywords: Bayesian networks; Shannon entropy; Kullback–Leibler divergence



Citation: Scutari, M. Entropy and the Kullback–Leibler Divergence for Bayesian Networks: Computational Complexity and Efficient Implementation. *Algorithms* **2024**, *17*, 24. <https://doi.org/10.3390/a17010024>

Academic Editors: Kevin B Korb, Steven Mascaro, Erik P. Nyberg and Frank Werner

Received: 29 November 2023

Revised: 31 December 2023

Accepted: 4 January 2024

Published: 6 January 2024



Copyright: © 2024 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Bayesian networks [1] (BNs) have played a central role in machine learning research since the early days of the field as expert systems [2,3], graphical models [4,5], dynamic and latent variables models [6], and as the foundation of causal discovery [7] and causal inference [8]. They have also found applications as diverse as comorbidities in clinical psychology [9], the genetics of COVID-19 [10], the Sustainable Development Goals of the United Nations [11], railway disruptions [12] and industry 4.0 [13].

Machine learning, however, has evolved to include a variety of other models and reformulated them into a very general information-theoretic framework. The central quantities of this framework are Shannon's entropy and the Kullback–Leibler divergence. Learning models from data relies crucially on the former to measure the amount of information captured by the model (or its complement, the amount of information lost in the residuals) and on the latter as the loss function we want to minimise. For instance, we can construct variational inference [14], the Expectation-Maximisation algorithm [15], Expectation Propagation [16] and various dimensionality reduction approaches such as t-SNE [17] and UMAP [18] using only these two quantities. We can also reformulate classical maximum-likelihood and Bayesian approaches to the same effect, from logistic regression to kernel methods to boosting [19,20].

Therefore, the lack of literature on how to compute the entropy of a BN and the Kullback–Leibler divergence between two BNs is surprising. While both are mentioned in Koller and Friedman [5] and discussed at a theoretical level in Moral et al. [21] for discrete BNs, no resources are available on any other type of BN. Furthermore, no numerical examples of how to compute them are available even for discrete BNs. *We fill this gap in the literature by:*

- *Deriving efficient formulations of Shannon's entropy and the Kullback–Leibler divergence for Gaussian BNs and conditional linear Gaussian BNs.*
- *Exploring the computational complexity of both for all common types of BNs.*

- Providing step-by-step numeric examples for all computations and all common types of BNs.

Our aim is to make apparent how both quantities are computed in their closed-form exact expressions and what is the associated computational cost.

The common alternative is to estimate both Shannon's entropy and the Kullback–Leibler divergence empirically using Monte Carlo sampling. Admittedly, this approach is simple to implement for all types of BNs. However, it has two crucial drawbacks:

1. Using asymptotic estimates voids the theoretical properties of many machine learning algorithms: Expectation-Maximisation is not guaranteed to converge [5], for instance.
2. The number of samples required to estimate the Kullback–Leibler divergence accurately on the tails of the global distribution of both BNs is also an issue [22], especially when we need to evaluate it repeatedly as part of some machine learning algorithm. The same is true, although to a lesser extent, for Shannon's entropy as well. In general, the rate of convergence to the true posterior in Monte Carlo particle filters is proportional to the number of variables squared [23].

Therefore, efficiently computing the exact value of Shannon's entropy and the Kullback–Leibler divergence is a valuable research endeavour with a practical impact on BN use in machine learning. To help its development, we implemented the methods proposed in the paper in our *bnlearn* R package [24].

The remainder of the paper is structured as follows. In Section 2, we provide the basic definitions, properties and notation of BNs. In Section 3, we revisit the most common distributional assumptions in the BN literature: discrete BNs (Section 3.1), Gaussian BNs (Section 3.2) and conditional linear Gaussian BNs (Section 3.3). We also briefly discuss exact and approximate inferences for these types of BNs in Section 3.4 to introduce some key concepts for later use. In Section 4, we discuss how we can compute Shannon's entropy and the Kullback–Leibler divergence for each type of BN. We conclude the paper by summarising and discussing the relevance of these foundational results in Section 5. Appendix A summarises all the computational complexity results from earlier sections, and Appendix B contains additional examples we omitted from the main text for brevity.

2. Bayesian Networks

Bayesian networks (BNs) are a class of probabilistic graphical models defined over a set of random variables $\mathbf{X} = \{X_1, \dots, X_N\}$, each describing some quantity of interest, that are associated with the nodes of a directed acyclic graph (DAG) \mathcal{G} . Arcs in \mathcal{G} express direct dependence relationships between the variables in \mathbf{X} , with graphical separation in \mathcal{G} implying conditional independence in probability. As a result, \mathcal{G} induces the factorisation

$$P(\mathbf{X} \mid \mathcal{G}, \Theta) = \prod_{i=1}^N P(X_i \mid \Pi_{X_i}, \Theta_{X_i}), \quad (1)$$

in which the global distribution (of \mathbf{X} , with parameters Θ) decomposes into one local distribution for each X_i (with parameters Θ_{X_i} , $\bigcup_{X_i} \Theta_{X_i} = \Theta$) conditional on its parents Π_{X_i} .

This factorisation is as effective at reducing the computational burden of working with BNs as the DAG underlying the BN is sparse, meaning that each node X_i has a small number of parents ($|\Pi_{X_i}| < c$, usually with $c \in [2, 5]$). For instance, learning BNs from data is only feasible in practice if this holds. The task of learning a BN $\mathcal{B} = (\mathcal{G}, \Theta)$ from a data set \mathcal{D} containing n observations comprises two steps:

$$\underbrace{P(\mathcal{G}, \Theta \mid \mathcal{D})}_{\text{learning}} = \underbrace{P(\mathcal{G} \mid \mathcal{D})}_{\text{structure learning}} \cdot \underbrace{P(\Theta \mid \mathcal{G}, \mathcal{D})}_{\text{parameter learning}}.$$

If we assume that parameters in different local distributions are independent [25], we can perform parameter learning independently for each node. Each $X_i \mid \Pi_{X_i}$ will have a low-dimensional parameter space Θ_{X_i} , making parameter learning computationally

efficient. On the other hand, structure learning is well known to be both NP-hard [26] and NP-complete [27], even under unrealistically favourable conditions such as the availability of an independence and inference oracle [28]. However, if \mathcal{G} is sparse, heuristic learning algorithms have been shown to run in quadratic time [29]. Exact learning algorithms, which have optimality guarantees that heuristic algorithms lack, retain their exponential complexity but become feasible for small problems because sparsity allows for tight bounds on goodness-of-fit scores and the efficient pruning of the space of the DAGs [30–32].

3. Common Distributional Assumptions for Bayesian Networks

While there are many possible choices for the distribution of \mathbf{X} in principle, the literature has focused on three cases.

3.1. Discrete BNs

Discrete BNs [25] assume that both \mathbf{X} and the X_i are multinomial random variables (The literature sometimes denotes discrete BNs as “dBNs” or “DBNs”; we do not do that in this paper to avoid confusion with dynamic BNs, which are also commonly denoted as “dBNs”). Local distributions take the form

$$X_i \mid \Pi_{X_i} \sim \text{Mul}(\pi_{ik|j}), \quad \pi_{ik|j} = P(X_i = k \mid \Pi_{X_i} = j);$$

their parameters are the conditional probabilities of X_i given each configuration of the values of its parents, usually represented as a conditional probability table (CPT) for each X_i . The $\pi_{ik|j}$ can be estimated from data via the sufficient statistic $\{n_{ijk}, i = 1, \dots, N; j = 1, \dots, q_i; k = 1, \dots, r_i\}$, the corresponding counts tallied from $\{X_i, \Pi_{X_i}\}$ using maximum likelihood, Bayesian or shrinkage estimators as described in Koller and Friedman [5] and Hausser and Strimmer [33].

The global distribution takes the form of an N -dimensional probability table with one dimension for each variable. Assuming that each X_i takes at most l values, the table will contain $|\text{Val}(\mathbf{X})| = O(l^N)$ cells, where $\text{Val}(\cdot)$ denotes the possible (configurations of the) values of its argument. As a result, it is impractical to use for medium and large BNs. Following standard practices from categorical data analysis [34], we can produce the CPT for each X_i from the global distribution by marginalising (that is, summing over) all the variables other than $\{X_i, \Pi_{X_i}\}$ and then normalising over each configuration of Π_{X_i} . Conversely, we can compose the global distribution from the local distributions of the X_i by multiplying the appropriate set of conditional probabilities. The computational complexity of the composition is $O(Nl^N)$ because applying (1) for each of the l^N cells yields

$$P(\mathbf{X} = \mathbf{x}) = \prod_{i=1}^N P(X_i = x_i \mid \Pi_{X_i} = \mathbf{x}_{\Pi_{X_i}}),$$

which involves N multiplications. As for the decomposition, for each node, we:

1. Sum over $N - |\Pi_{X_i}| - 1$ variables to produce the joint probability table for $\{X_i, \Pi_{X_i}\}$, which contains $O(l^{|\Pi_{X_i}|+1})$ cells. The value of each cell is the sum of $O(l^{N-|\Pi_{X_i}|-1})$ probabilities.
2. Normalise the columns of the joint probability table for $\{X_i, \Pi_{X_i}\}$ over each of the $O(l^{|\Pi_{X_i}|})$ configurations of values of Π_{X_i} , which involves summing $O(l)$ probabilities and dividing them by their total.

The resulting computational complexity is

$$\underbrace{O(l^{|\Pi_{X_i}|+1} \cdot l^{N-|\Pi_{X_i}|-1})}_{\text{marginalisation}} + \underbrace{O(l \cdot l^{|\Pi_{X_i}|})}_{\text{normalisation}} = O(l^N + l^{|\Pi_{X_i}|+1}) \quad (2)$$

for each node and $O(Nl^N + l \sum_{i=1}^N l^{|\Pi_{X_i}|})$ for the whole BN.

Example 1 (Composing and decomposing a discrete BN). *For reasons of space, this example is presented as Example A1 in Appendix B.*

3.2. Gaussian BNs

Gaussian BNs [35] (GBNs) model \mathbf{X} with a multivariate normal random variable $N(\boldsymbol{\mu}_{\mathcal{B}}, \Sigma_{\mathcal{B}})$ and assume that the X_i are univariate normals linked by linear dependencies,

$$X_i \mid \Pi_{X_i} \sim N(\mu_{X_i} + \Pi_{X_i} \boldsymbol{\beta}_{X_i}, \sigma_{X_i}^2), \quad (3)$$

which can be equivalently written as linear regression models of the form

$$X_i = \mu_{X_i} + \Pi_{X_i} \boldsymbol{\beta}_{X_i} + \varepsilon_{X_i}, \quad \varepsilon_{X_i} \sim N(0, \sigma_{X_i}^2). \quad (4)$$

The parameters in (3) and (4) are the regression coefficients $\boldsymbol{\beta}_{X_i}$ associated with the parents Π_{X_i} , an intercept term μ_{X_i} and the variance $\sigma_{X_i}^2$. They are usually estimated by maximum likelihood, but Bayesian and regularised estimators are available as well [1].

The link between the parameterisation of the global distribution of a GBN and that of its local distributions is detailed in Pourahmadi [36]. We summarise it here for later use.

- *Composing the global distribution.* We can create an $N \times N$ lower triangular matrix $C_{\mathcal{B}}$ from the regression coefficients in the local distributions such that $C_{\mathcal{B}} C_{\mathcal{B}}^T$ gives $\Sigma_{\mathcal{B}}$ after rearranging rows and columns. In particular, we:
 1. Arrange the nodes of \mathcal{B} in the (partial) topological ordering induced by \mathcal{G} , denoted $X_{(i)}, i = 1, \dots, N$.
 2. The i th row of $C_{\mathcal{B}}$ (denoted $C_{\mathcal{B}}[i; \cdot]$, $i = 1, \dots, N$) is associated with $X_{(i)}$. We compute its elements from the parameters of $X_{(i)} \mid \Pi_{X_{(i)}}$ as

$$C_{\mathcal{B}}[i; i] = \sqrt{\sigma_{X_{(i)}}^2} \quad \text{and} \quad C_{\mathcal{B}}[i; \cdot] = \boldsymbol{\beta}_{X_{(i)}} C_{\mathcal{B}}[\Pi_{X_{(i)}}; \cdot],$$

where $C_{\mathcal{B}}[\Pi_{X_{(i)}}; \cdot]$ are the rows of $C_{\mathcal{B}}$ that correspond to the parents of $X_{(i)}$. The rows of $C_{\mathcal{B}}$ are filled following the topological ordering of the BN.

3. Compute $\tilde{\Sigma}_{\mathcal{B}} = C_{\mathcal{B}} C_{\mathcal{B}}^T$.
4. Rearrange the rows and columns of $\tilde{\Sigma}_{\mathcal{B}}$ to obtain $\Sigma_{\mathcal{B}}$.

Intuitively, we construct $C_{\mathcal{B}}$ by propagating the node variances along the paths in \mathcal{G} while combining them with the regression coefficients, which are functions of the correlations between adjacent nodes. As a result, $C_{\mathcal{B}} C_{\mathcal{B}}^T$ gives $\Sigma_{\mathcal{B}}$ after rearranging the rows and columns to follow the original ordering of the nodes.

The elements of the mean vector $\boldsymbol{\mu}_{\mathcal{B}}$ are similarly computed as $E(X_{(i)}) = \Pi_{X_{(i)}} \boldsymbol{\beta}_{X_{(i)}} + \mu_{X_{(i)}}$ iterating over the variables in topological order.

- *Decomposing the global distribution.* Conversely, we can derive the matrix $C_{\mathcal{B}}$ from $\Sigma_{\mathcal{B}}$ by reordering its rows and columns to follow the topological ordering of the variables in \mathcal{G} and computing its Cholesky decomposition. Then

$$R = I_N - \text{diag}(C_{\mathcal{B}}) C_{\mathcal{B}}^{-1},$$

contains the regression coefficients $\boldsymbol{\beta}_{X_{(i)}}$ in the elements corresponding to $X_{(i)}, \Pi_{X_{(i)}}$ (Here $\text{diag}(C_{\mathcal{B}})$ is a diagonal matrix with the same diagonal elements as $C_{\mathcal{B}}$ and I_N is the identity matrix.) Finally, we compute the intercepts μ_{X_i} as $\boldsymbol{\mu}_{\mathcal{B}} - R \boldsymbol{\mu}_{\mathcal{B}}$ by reversing the equations we used to construct $\boldsymbol{\mu}_{\mathcal{B}}$ above.

The computational complexity of composing the global distribution is bound by the matrix multiplication $C_{\mathcal{B}} C_{\mathcal{B}}^T$, which is $O(N^3)$; if we assume that \mathcal{G} is sparse as in Scutari et al. [29], the number of arcs is bound by some cN , computing the $\boldsymbol{\mu}_{\mathcal{B}}$ takes $O(N)$ operations. The complexity of decomposing the global distribution is also $O(N^3)$ because both inverting $C_{\mathcal{B}}$ and multiplying the result by $\text{diag}(C_{\mathcal{B}})$ are $O(N^3)$.

Example 2 (Composing and decomposing a GBN). Consider the GBN \mathcal{B} from Figure 1 top. The topological ordering of the variables defined by \mathcal{B} is $\{\{X_1, X_2\}, X_4, X_3\}$, so

$$C_{\mathcal{B}} = \begin{matrix} & X_1 & X_2 & X_4 & X_3 \\ \begin{matrix} X_1 \\ X_2 \\ X_4 \\ X_3 \end{matrix} & \begin{pmatrix} 0.894 & 0 & 0 & 0 \\ 0 & 0.774 & 0 & 0 \\ 1.341 & 2.014 & 1.049 & 0 \\ 1.610 & 2.416 & 1.258 & 0.948 \end{pmatrix} \end{matrix}$$

where the diagonal elements are

$$C_{\mathcal{B}}[X_1; X_1] = \sqrt{0.8}, \quad C_{\mathcal{B}}[X_2; X_2] = \sqrt{0.6}, \quad C_{\mathcal{B}}[X_4; X_4] = \sqrt{1.1}, \quad C_{\mathcal{B}}[X_3; X_3] = \sqrt{0.9};$$

and the elements below the diagonal are taken from the corresponding cells of

$$\begin{aligned} C_{\mathcal{B}}[X_4; \cdot] &= (1.5 \quad 2.6) \begin{pmatrix} 0.894 & 0 & 0 & 0 \\ 0 & 0.774 & 0 & 0 \end{pmatrix}, \\ C_{\mathcal{B}}[X_3; \cdot] &= (1.2) (1.341 \quad 2.014 \quad 1.049 \quad 0). \end{aligned}$$

Computing $C_{\mathcal{B}} C_{\mathcal{B}}^T$ gives

$$\tilde{\Sigma}_{\mathcal{B}} = \begin{matrix} & X_1 & X_2 & X_4 & X_3 \\ \begin{matrix} X_1 \\ X_2 \\ X_4 \\ X_3 \end{matrix} & \begin{pmatrix} 0.800 & 0 & 1.200 & 1.440 \\ 0 & 0.600 & 1.560 & 1.872 \\ 1.200 & 1.560 & 6.956 & 8.347 \\ 1.440 & 1.872 & 8.347 & 10.916 \end{pmatrix} \end{matrix}$$

and reordering the rows and columns of $\tilde{\Sigma}_{\mathcal{B}}$ gives

$$\Sigma_{\mathcal{B}} = \begin{matrix} & X_1 & X_2 & X_3 & X_4 \\ \begin{matrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{matrix} & \begin{pmatrix} 0.800 & 0 & 1.440 & 1.200 \\ 0 & 0.600 & 1.872 & 1.560 \\ 1.440 & 1.872 & 10.916 & 8.347 \\ 1.200 & 1.560 & 8.347 & 6.956 \end{pmatrix} \end{matrix}.$$

The elements of the corresponding expectation vector $\mu_{\mathcal{B}}$ are then

$$\begin{aligned} E(X_1) &= 2.400, \\ E(X_2) &= 1.800, \\ E(X_4) &= 0.2 + 1.5 E(X_1) + 2.6 E(X_2) = 8.480, \\ E(X_3) &= 2.1 + 1.2 E(X_4) = 12.276. \end{aligned}$$

Starting from $\Sigma_{\mathcal{B}}$, we can reorder its rows and columns to obtain $\tilde{\Sigma}_{\mathcal{B}}$. The Cholesky decomposition of $\tilde{\Sigma}_{\mathcal{B}}$ is $C_{\mathcal{B}}$. Then

$$\begin{aligned} \sigma_{X_1}^2 &= C_{\mathcal{B}}[X_1; X_1]^2 = 0.8, & \sigma_{X_2}^2 &= C_{\mathcal{B}}[X_2; X_2]^2 = 0.6, \\ \sigma_{X_3}^2 &= C_{\mathcal{B}}[X_3; X_3]^2 = 0.9, & \sigma_{X_4}^2 &= C_{\mathcal{B}}[X_4; X_4]^2 = 0.11. \end{aligned}$$

The coefficients β_{X_i} of the local distributions are available from

$$R = I_N - \underbrace{\begin{bmatrix} 0.894 & 0 & 0 & 0 \\ 0 & 0.774 & 0 & 0 \\ 0 & 0 & 1.049 & 0 \\ 0 & 0 & 0 & 0.948 \end{bmatrix}}_{\text{diag}(C_B)} \underbrace{\begin{bmatrix} 1.118 & 0 & 0 & 0 \\ 0 & 1.291 & 0 & 0 \\ -1.430 & -2.479 & 0.953 & 0 \\ 0 & 0 & -1.265 & 1.054 \end{bmatrix}}_{C_B^{-1}}$$

$$= \begin{matrix} & X_1 & X_2 & X_4 & X_3 \\ \begin{matrix} X_1 \\ X_2 \\ X_4 \\ X_3 \end{matrix} & \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1.500 & 2.600 & 0 & 0 \\ 0 & 0 & 1.200 & 0 \end{pmatrix} \end{matrix}$$

where we can read $R_{X_4, X_1} = 1.5 = \beta_{X_4, X_1}$, $R_{X_4, X_2} = 2.6 = \beta_{X_4, X_2}$, $R_{X_3, X_4} = 1.2 = \beta_{X_3, X_4}$.

We can read the standard errors of X_1 , X_2 , X_3 and X_4 directly from the diagonal elements of C_B , and we can compute the intercepts from $\mu_B - R\mu_B$ which amounts to

$$\begin{aligned} \mu_{X_1} &= E(X_1) = 2.400, \\ \mu_{X_2} &= E(X_2) = 1.800, \\ \mu_{X_4} &= E(X_4) - E(X_1)\beta_{X_4, X_1} - E(X_2)\beta_{X_4, X_2} = 0.200, \\ \mu_{X_3} &= E(X_3) - E(X_4)\beta_{X_3, X_4} = 2.100. \end{aligned}$$

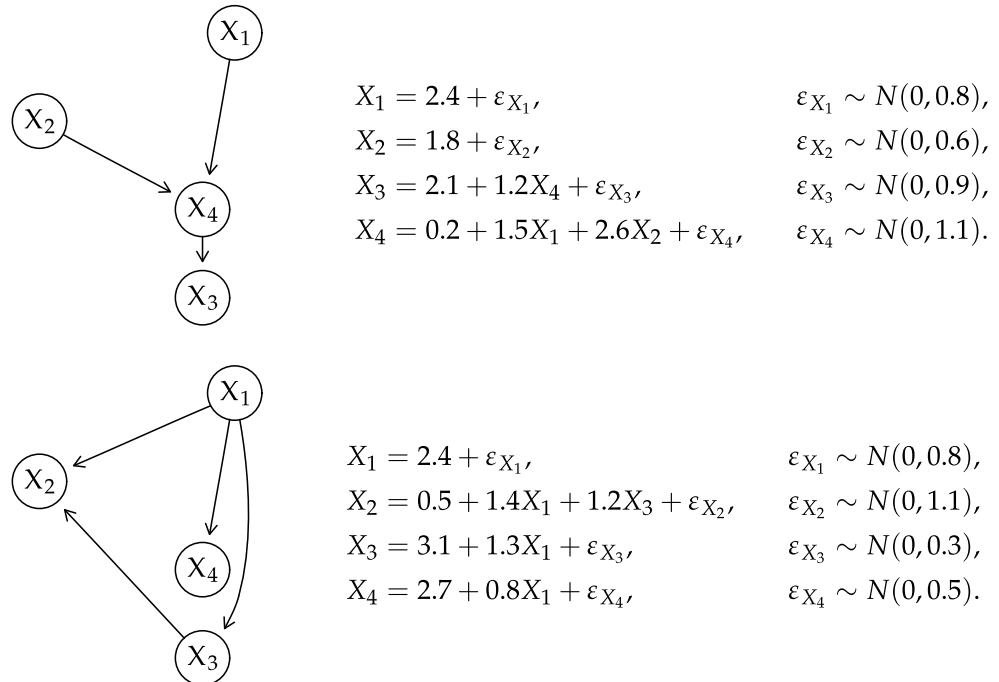


Figure 1. DAGs and local distributions for the GBNs B (top) and B' (bottom) used in Examples 2 and 6–9.

3.3. Conditional Linear Gaussian BNs

Finally, *conditional linear Gaussian BNs* [37] (CLGBNs) subsume discrete BNs and GBNs as particular cases by combining discrete and continuous random variables in a mixture model. If we denote the former with \mathbf{X}_D and the latter with \mathbf{X}_G , so that $\mathbf{X} = \mathbf{X}_D \cup \mathbf{X}_G$, then:

- Discrete $X_i \in \mathbf{X}_D$ are only allowed to have discrete parents (denoted Δ_{X_i}), and are assumed to follow a multinomial distribution parameterised with CPTs. We can estimate their parameters in the same way as those in a discrete BN.

- Continuous $X_i \in \mathbf{X}_G$ are allowed to have both discrete and continuous parents (denoted $\Gamma_{X_i}, \Delta_{X_i} \cup \Gamma_{X_i} = \Pi_{X_i}$). Their local distributions are

$$X_i \mid \Pi_{X_i} \sim N\left(\mu_{X_i, \delta_{X_i}} + \Gamma_{X_i} \beta_{X_i, \delta_{X_i}}, \sigma_{X_i, \delta_{X_i}}^2\right),$$

which is equivalent to a mixture of linear regressions against the continuous parents with one component for each configuration $\delta_{X_i} \in \text{Val}(\Delta_{X_i})$ of the discrete parents:

$$X_i = \mu_{X_i, \delta_{X_i}} + \Gamma_{X_i} \beta_{X_i, \delta_{X_i}} + \varepsilon_{X_i, \delta_{X_i}}, \quad \varepsilon_{X_i, \delta_{X_i}} \sim N\left(0, \sigma_{X_i, \delta_{X_i}}^2\right).$$

If X_i has no discrete parents, the mixture reverts to a single linear regression like that in (4). The parameters of these local distributions are usually estimated by maximum likelihood like those in a GBN; we have used hierarchical regressions with random effects in our recent work [38] for this purpose as well. Bayesian and regularised estimators are also an option [5].

If the CLGBN comprises $|\mathbf{X}_D| = M$ discrete nodes and $|\mathbf{X}_G| = N - M$ continuous nodes, these distributional assumptions imply the partial topological ordering

$$\underbrace{\{X_{(1)}, \dots, X_{(M)}\}}_{\text{discrete nodes}}, \underbrace{\{X_{(M+1)}, \dots, X_{(N)}\}}_{\text{continuous nodes}}. \quad (5)$$

The discrete nodes jointly follow a multinomial distribution, effectively forming a discrete BN. The continuous nodes jointly follow a multivariate normal distribution, parameterised as a GBN, for each configuration of the discrete nodes. Therefore, the global distribution is a Gaussian mixture in which the discrete nodes identify the components, and the continuous nodes determine their distribution. The practical link between the global and local distributions follows directly from Sections 3.1 and 3.2.

Example 3 (Composing and decomposing a CLGBN). *For reasons of space, this example is presented as Example A2 in Appendix B.*

The complexity of composing and decomposing the global distribution is then

$$\underbrace{O(Ml^M)}_{\text{convert between CPTs and component probabilities}} + \underbrace{O((N-M)^3 l^{\text{Val}(\Delta)})}_{\text{(de)compose the distinct component distributions}}$$

where $\Delta = \bigcup_{X_i \in \mathbf{X}_G} \Delta_{X_i}$ are the discrete parents of the continuous nodes.

3.4. Inference

For BNs, *inference* broadly denotes obtaining the conditional distribution of a subset of variables conditional on a second subset of variables. Following older terminology from expert systems [2], this is called formulating a *query* in which we ask the BN about the probability of an *event* of interest after observing some *evidence*. In conditional probability queries, the event of interest is the probability of one or more events in (or the whole distribution of) some variables of interest conditional on the values assumed by the evidence variables. In maximum a posteriori (“most probable explanation”) queries, we condition the values of the evidence variables to predict those of the event variables.

All inference computations on BNs are completely automated by *exact* and *approximate* algorithms, which we will briefly describe here. We refer the interested reader to the more detailed treatment in Castillo et al. [2] and Koller and Friedman [5].

Exact inference algorithms use local computations to compute the value of the query. The seminal works of Lauritzen and Spiegelhalter [39], Lauritzen and Wermuth [37] and Lauritzen and Jensen [40] describe how to transform a discrete BN or a (CL)GBN into

a *junction tree* as a preliminary step before using belief propagation. Cowell [41] uses elimination trees for the same purpose in CLGBNs. (A junction tree is an undirected tree whose nodes are the cliques in the moral graph constructed from the BN and their intersections. A clique is the maximal subset of nodes such that every two nodes in the subset are adjacent).

Namasivayam et al. [42] give the computational complexity of constructing the junction tree from a discrete BN as $O(Nw + wl^wN)$ where w is the maximum number of nodes in a clique and, as before, l is the maximum number of values that a variable can take. We take the complexity of belief propagation to be $O(Nwl^w + |\Theta|)$, as stated in Lauritzen and Spiegelhalter [39] (“The global propagation is no worse than the initialisation [of the junction tree]”). This is confirmed by Pennock [43] and Namasivayam and Prasanna [44].

As for GBNs, we can also perform exact inference through their global distribution because the latter has only $O(N^2 + N)$ parameters. The computational complexity of this approach is $O(N^3)$ because of the cost of composing the global distribution, which we derived in Section 3.2. However, all the operations involved are linear, making it possible to leverage specialised hardware such as GPUs and TPUs to the best effect. Koller and Friedman [5] (Section 14.2.1) note that “inference in linear Gaussian networks is linear in the number of cliques, and at most cubic in the size of the largest clique” when using junction trees and belief propagation. Therefore, junction trees may be significantly faster for GBNs when $w \ll N$. However, the correctness and convergence of belief propagation in GBNs require a set of sufficient conditions that have been studied comprehensively by Malioutov et al. [45]. Using the global distribution directly always produces correct results.

Approximate inference algorithms use Monte Carlo simulations to sample from the global distribution of \mathbf{X} through the local distributions and estimate the answer queries by computing the appropriate summary statistics on the particles they generate. Therefore, they mirror the Monte Carlo and Markov chain Monte Carlo approaches in the literature: rejection sampling, importance sampling, and sequential Monte Carlo among others. Two state-of-the-art examples are the *adaptive importance sampling* (AIS-BN) scheme [46] and the *evidence pre-propagation importance sampling* (EPIS-BN) [47].

4. Shannon Entropy and Kullback–Leibler Divergence

The general definition of Shannon entropy for the probability distribution P of \mathbf{X} is

$$H(P) = E_P(-\log P(\mathbf{X})) = - \int_{\text{Val}(\mathbf{X})} P(\mathbf{x}) \log P(\mathbf{x}) d\mathbf{x}. \quad (6)$$

The Kullback–Leibler divergence between two distributions P and Q for the same random variables \mathbf{X} is defined as

$$\text{KL}(P \parallel Q) = E_{P(\mathbf{X})} \left(-\log \frac{P(\mathbf{X})}{Q(\mathbf{X})} \right) = - \int_{\text{Val}(\mathbf{X})} P(\mathbf{x}) \log \frac{P(\mathbf{x})}{Q(\mathbf{x})} d\mathbf{x}. \quad (7)$$

They are linked as follows:

$$\underbrace{E_{P(\mathbf{X})} \left(-\log \frac{P(\mathbf{X})}{Q(\mathbf{X})} \right)}_{\text{KL}(P(\mathbf{X}) \parallel Q(\mathbf{X}))} = \underbrace{E_{P(\mathbf{X})}(-\log P(\mathbf{X}))}_{H(P(\mathbf{X}))} + \underbrace{E_{P(\mathbf{X})}(\log Q(\mathbf{X}))}_{H(P(\mathbf{X}), Q(\mathbf{X}))} \quad (8)$$

where $H(P(\mathbf{X}), Q(\mathbf{X}))$ is the cross-entropy between $P(\mathbf{X})$ and $Q(\mathbf{X})$. For the many properties of these quantities, we refer the reader to Cover and Thomas [48] and Csiszár and Shields [49]. Their use and interpretation are covered in depth (and breadth!) in Murphy [19,20] for general machine learning and in Koller and Friedman [5] for BNs.

For a BN \mathcal{B} encoding the probability distribution of \mathbf{X} , (6) decomposes into

$$H(\mathcal{B}) = \sum_{i=1}^N H(X_i \mid \Pi_{X_i}^{\mathcal{B}})$$

where $\Pi_{X_i}^{\mathcal{B}}$ are the parents of X_i in \mathcal{B} . While this decomposition looks similar to (1), we see that its terms are not necessarily orthogonal, unlike the local distributions.

As for (7), we cannot simply write

$$\text{KL}(\mathcal{B} \parallel \mathcal{B}') = \sum_{i=1}^N \text{KL}(X_i \mid \Pi_{X_i}^{\mathcal{B}} \parallel X_i \mid \Pi_{X_i}^{\mathcal{B}'})$$

because, in the general case, the nodes X_i have different parents in \mathcal{B} and \mathcal{B}' . This issue impacts the complexity of computing Kullback–Leibler divergences in different ways depending on the type of BN.

4.1. Discrete BNs

For discrete BNs, $H(\mathcal{B})$ does not decompose into orthogonal components. As pointed out in Koller and Friedman [5] (Section 8.4.12),

$$H(X_i \mid \Pi_{X_i}^{\mathcal{B}}) = \sum_{j=1}^{q_i} P(\Pi_{X_i}^{\mathcal{B}} = j) H(X_i \mid \Pi_{X_i}^{\mathcal{B}} = j) \quad \text{where}$$

$$H(X_i \mid \Pi_{X_i}^{\mathcal{B}} = j) = - \sum_{k=1}^{r_i} \pi_{ik|j}(\mathcal{B}) \log \pi_{ik|j}(\mathcal{B}). \quad (9)$$

If we estimated the conditional probabilities $\pi_{ik|j}(\mathcal{B})$ from data, the $P(\Pi_{X_i}^{\mathcal{B}} = j)$ are already available as the normalising constants of the individual conditional distributions $\{\pi_{ik|j}(\mathcal{B}), j = 1, \dots, q_i\}$ in the local distribution of X_i . In this case, the complexity of computing $H(X_i \mid \Pi_{X_i}^{\mathcal{B}})$ is linear in the number of parameters: $O(|\Theta|) = \sum_{i=1}^N O(|\Theta_{X_i}|)$.

In the general case, we need exact inference to compute the probabilities $P(\Pi_{X_i}^{\mathcal{B}} = j)$. Fortunately, they can be readily extracted from the junction tree derived from \mathcal{B} as follows:

1. Identify a clique containing both X_i and $\Pi_{X_i}^{\mathcal{B}}$. Such a clique is guaranteed to exist by the family preservation property [5] (Definition 10.1).
2. Compute the marginal distribution of $\Pi_{X_i}^{\mathcal{B}}$ by summing over the remaining variables in the clique.

Combining the computational complexity of constructing the junction tree from Section 3.4 and that of marginalisation, which is at most $O(l^{w-1})$ for each node as in (2), we have

$$\underbrace{O(Nw + wl^w N)}_{\text{create the junction tree}} + \underbrace{O(Nl^{w-1})}_{\text{compute the } P(\Pi_{X_i}^{\mathcal{B}} = j)} + \underbrace{O(|\Theta|)}_{\text{compute } H(\mathcal{B})} =$$

$$O(N(w(1 + l^w) + l^{w-1}) + |\Theta|),$$

which is exponential in the maximum clique size w . (The maximum clique size in a junction tree is proportional to the *treewidth* of the BN the junction tree is created from, which is also used in the literature to characterise computational complexity in BNs.) Interestingly, we do not need to perform belief propagation, so computing $H(\mathcal{B})$ is more efficient than other inference tasks.

Example 4 (Entropy of a discrete BN). *For reasons of space, this example is presented as Example A3 in Appendix B.*

The Kullback–Leibler divergence has a similar issue, as noted in Koller and Friedman [5] (Section 8.4.2). The best and most complete explanation of how to compute it for discrete

BNs is in Moral et al. [21]. After decomposing $KL(\mathcal{B} \parallel \mathcal{B}')$ following (8) to separate $H(\mathcal{B})$ and $H(\mathcal{B}, \mathcal{B}')$, Moral et al. [21] show that the latter takes the form

$$H(\mathcal{B}, \mathcal{B}') = \sum_{i=1}^N \sum_{j \in Val(\Pi_{X_i}^{\mathcal{B}'})} \left[\sum_{k=1}^{r_i} \pi_{ikj}(\mathcal{B}) \log \pi_{ikj}(\mathcal{B}') \right] \quad (10)$$

where:

- $\pi_{ikj}(\mathcal{B}) = P(X_i = k, \Pi_{X_i}(\mathcal{B}') = j)$ is the probability assigned by \mathcal{B} to $X_i = k$ given that the variables that are parents of X_i in \mathcal{B}' take value j ;
- $\pi_{ikj}(\mathcal{B}') = P(X_i = k \mid \Pi_{X_i}(\mathcal{B}') = j)$ is the (k, j) element of the CPT of X_i in \mathcal{B}' .

In order to compute the $\pi_{ikj}(\mathcal{B})$, we need to transform \mathcal{B} into its junction tree and use belief propagation to compute the joint distribution of $X_i \cup \Pi_{X_i}^{\mathcal{B}'}$. As a result, $H(\mathcal{B}, \mathcal{B}')$ does not decompose at all: each $\pi_{ikj}(\mathcal{B})$ can potentially depend on the whole BN \mathcal{B} .

Algorithmically, to compute $KL(\mathcal{B} \parallel \mathcal{B}')$ we:

1. Transform \mathcal{B} into its junction tree.
2. Compute the entropy $H(\mathcal{B})$.
3. For each node X_i :
 - (a) Identify $\Pi_{X_i}^{\mathcal{B}'}$, the parents of X_i in \mathcal{B}' .
 - (b) Obtain the distribution of the variables $\{X_i, \Pi_{X_i}^{\mathcal{B}'}\}$ from the junction tree of \mathcal{B} , consisting of the probabilities $\pi_{ikj}(\mathcal{B})$.
 - (c) Read the $\pi_{ikj}(\mathcal{B}')$ from the local distribution of X_i in \mathcal{B}' .
4. Use the $\pi_{ikj}(\mathcal{B})$ and the $\pi_{ikj}(\mathcal{B}')$ to compute (10).

The computational complexity of this procedure is as follows:

$$\underbrace{O\left(N(w(1 + l^w) + l^{w-1}) + |\Theta|\right)}_{\text{create the junction tree of } \mathcal{B} \text{ and computing } H(\mathcal{B})} + \underbrace{O(Nl^c(Nwl^w + |\Theta|))}_{\text{produce the } \pi_{ikj}(\mathcal{B})} + \underbrace{O(|\Theta|)}_{\text{compute } H(\mathcal{B}, \mathcal{B}')} = O\left(N^2wl^{w+c} + N(w + wl^w + l^{w-1}) + (Nl^c + 2)|\Theta|\right). \quad (11)$$

As noted in Moral et al. [21], computing the $\pi_{ikj}(\mathcal{B})$ requires a separate run of belief propagation for each configuration of the $\Pi_{X_i}^{\mathcal{B}'}$, for a total of $\sum_{i=1}^N l^{|\Pi_{X_i}^{\mathcal{B}'}|}$ times. If we assume that the DAG underlying \mathcal{B}' is sparse, we have that $|\Pi_{X_i}^{\mathcal{B}'}| \leq c$ and the overall complexity of this step becomes $O(Nl^c \cdot (Nwl^w + |\Theta|))$, N times that listed in Section 3.4. The caching scheme devised by Moral et al. [21] is very effective in limiting the use of belief propagation, but it does not alter its exponential complexity.

Example 5 (KL between two discrete BNs). Consider the discrete BN \mathcal{B} from Figure 2 top. Furthermore, consider the BN \mathcal{B}' from Figure 2 bottom. We constructed the global distribution of \mathcal{B} in Example A1; we can similarly compose the global distribution of \mathcal{B}' , shown below.

$X_1 = a$						$X_1 = b$					
$X_2 = c$			$X_2 = d$			$X_2 = c$			$X_2 = d$		
X_3			X_3			X_3			X_3		
X_4	e	f	X_4	e	f	X_4	e	f	X_4	e	f
g	0.013	0.033	g	0.022	0.054	g	0.016	0.144	g	0.016	0.139
h	0.072	0.062	h	0.029	0.025	h	0.013	0.040	h	0.079	0.243

Since both global distributions are limited in size, we can then compute the Kullback–Leibler divergence between \mathcal{B} and \mathcal{B}' using (7).

$$\begin{aligned} \text{KL}(\mathcal{B} \parallel \mathcal{B}') &= -0.013 \log 0.013 - 0.016 \log 0.016 - 0.022 \log 0.022 - 0.016 \log 0.016 - \\ &\quad 0.072 \log 0.072 - 0.013 \log 0.013 - 0.029 \log 0.029 - 0.079 \log 0.079 - \\ &\quad 0.033 \log 0.033 - 0.144 \log 0.144 - 0.054 \log 0.054 - 0.139 \log 0.139 - \\ &\quad - 0.062 \log 0.062 - 0.04 \log 0.04 - 0.025 \log 0.025 - 0.243 \log 0.243 = 0.687 \end{aligned}$$

In the general case, when we cannot use the global distributions, we follow the approach described in Section 4.1. Firstly, we apply (8) to write

$$\text{KL}(\mathcal{B} \parallel \mathcal{B}') = H(\mathcal{B}) - H(\mathcal{B}, \mathcal{B}');$$

we have from Example A3 that $H(\mathcal{B}) = 2.440$. As for the cross-entropy $H(\mathcal{B}, \mathcal{B}')$, we apply (10):

1. We identify the parents of each node in \mathcal{B}' :

$$\Pi_{X_1}^{\mathcal{B}'} = \{\emptyset\}, \quad \Pi_{X_2}^{\mathcal{B}'} = \{X_1, X_4\}, \quad \Pi_{X_3}^{\mathcal{B}'} = \{X_1\}, \quad \Pi_{X_4}^{\mathcal{B}'} = \{X_3\}.$$

2. We construct a junction tree from \mathcal{B} and we use it to compute the distributions $P(X_1)$, $P(X_2, X_1, X_4)$, $P(X_3, X_1)$ and $P(X_4, X_3)$.

X_1		$\{X_1, X_4\}$			
a	b	$\{a, g\}$	$\{a, h\}$	$\{b, g\}$	$\{b, h\}$
0.53	0.47	0.070	0.110	0.053	0.107
		0.089	0.261	0.076	0.235

X_3	X_1		X_4	X_3	
	e	f		e	f
	0.289	0.312		0.120	0.167
	0.241	0.158		0.481	0.231

3. We compute the cross-entropy terms for the individual variables in \mathcal{B} and \mathcal{B}' :

$$\begin{aligned} H(X_1^{\mathcal{B}}, X_1^{\mathcal{B}'}) &= 0.53 \log 0.31 + 0.47 \log 0.69 = -0.795; \\ H(X_2^{\mathcal{B}}, X_2^{\mathcal{B}'}) &= 0.070 \log 0.38 + 0.089 \log 0.62 + 0.110 \log 0.71 + 0.261 \log 0.29 + \\ &\quad 0.053 \log 0.51 + 0.076 \log 0.49 + 0.107 \log 0.14 + 0.235 \log 0.86 \\ &= -0.807; \\ H(X_3^{\mathcal{B}}, X_3^{\mathcal{B}'}) &= 0.289 \log 0.44 + 0.241 \log 0.56 + 0.312 \log 0.18 + 0.158 \log 0.82 \\ &= -0.943; \\ H(X_4^{\mathcal{B}}, X_4^{\mathcal{B}'}) &= 0.120 \log 0.26 + 0.481 \log 0.74 + 0.167 \log 0.50 + 0.231 \log 0.50 \\ &= -0.582; \end{aligned}$$

which sum up to $H(\mathcal{B}, \mathcal{B}') = \sum_{i=1}^N H(X_i^{\mathcal{B}}, X_i^{\mathcal{B}'}) = -3.127$.

4. We compute $\text{KL}(\mathcal{B} \parallel \mathcal{B}') = 2.440 - 3.127 = 0.687$, which matches the value we previously computed from the global distributions.

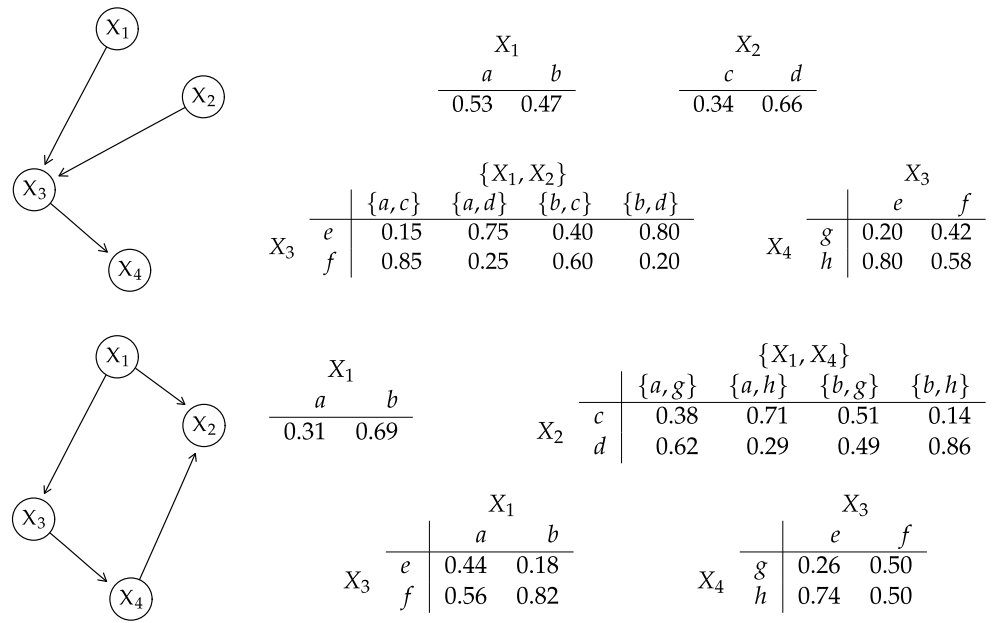


Figure 2. DAGs and local distributions for the discrete BNs \mathcal{B} (top) and \mathcal{B}' (bottom) used in Examples 1, 4 and 5.

4.2. Gaussian BNs

$H(\mathcal{B})$ decomposes along with the local distributions $X_i \mid \Pi_{X_i}$ in the case of GBNs: from (3), each $X_i \mid \Pi_{X_i}$ is a univariate normal with variance $\sigma_{X_i}^2(\mathcal{B})$ and therefore

$$H(X_i \mid \Pi_{X_i}^{\mathcal{B}}) = \frac{1}{2} \log(2\pi\sigma_{X_i}^2(\mathcal{B})) + \frac{1}{2} \quad (12)$$

which has a computational complexity of $O(1)$ for each node, $O(N)$ overall. Equivalently, we can start from the global distribution of \mathcal{B} from Section 3.2 and consider that

$$\det(\Sigma) = \det(C_{\mathcal{B}'} C_{\mathcal{B}'}^T) = \det(C_{\mathcal{B}})^2 = \left(\prod_{i=1}^N C_{\mathcal{B}}[i; i] \right)^2 = \prod_{i=1}^N \sigma_{X_i}^2(\mathcal{B}) \quad (13)$$

because $C_{\mathcal{B}}$ is lower triangular. The (multivariate normal) entropy of \mathbf{X} then becomes

$$\begin{aligned} H(\mathcal{B}) &= \frac{N}{2} + \frac{N}{2} \log 2\pi + \frac{1}{2} \log \det(\Sigma) = \frac{N}{2} + \frac{N}{2} \log 2\pi + \frac{1}{2} \sum_{i=1}^N \log \sigma_{X_i}^2(\mathcal{B}) \\ &= \sum_{i=1}^N \frac{1}{2} + \frac{1}{2} \log(2\pi\sigma_{X_i}^2(\mathcal{B})) = \sum_{i=1}^N H(X_i \mid \Pi_{X_i}^{\mathcal{B}}) \end{aligned}$$

in agreement with (12).

Example 6 (Entropy of a GBN). For reasons of space, this example is presented as Example A4 in Appendix B.

In the literature, the Kullback–Leibler divergence between two GBNs \mathcal{B} and \mathcal{B}' is usually computed using the respective global distributions $N(\mu_{\mathcal{B}}, \Sigma_{\mathcal{B}})$ and $N(\mu_{\mathcal{B}'}, \Sigma_{\mathcal{B}'})$ [50–52]. The general expression is

$$\text{KL}(\mathcal{B} \parallel \mathcal{B}') = \frac{1}{2} \left[\text{tr}(\Sigma_{\mathcal{B}'}^{-1} \Sigma_{\mathcal{B}}) + (\mu_{\mathcal{B}'} - \mu_{\mathcal{B}})^T \Sigma_{\mathcal{B}'}^{-1} (\mu_{\mathcal{B}'} - \mu_{\mathcal{B}}) - N + \log \frac{\det(\Sigma_{\mathcal{B}'})}{\det(\Sigma_{\mathcal{B}})} \right], \quad (14)$$

which has computational complexity

$$\begin{aligned}
 & \underbrace{O(2N^3 + 2N)}_{\text{compute } \mu_B, \mu_{B'}, \Sigma_B, \Sigma_{B'}} + \underbrace{O(N^3)}_{\text{invert } \Sigma_{B'}} + \underbrace{O(N^3)}_{\text{multiply } \Sigma_B^{-1} \text{ and } \Sigma_{B'}} + \underbrace{O(N)}_{\text{trace of } \Sigma_B^{-1} \Sigma_{B'}} + \\
 & \underbrace{O(N^2 + 2N)}_{\text{compute } (\mu_{B'} - \mu_B)^T \Sigma_{B'}^{-1} (\mu_{B'} - \mu_B)} + \underbrace{O(N^3)}_{\text{determinant of } \Sigma_{B'}} + \underbrace{O(N^3)}_{\text{determinant of } \Sigma_B} = \\
 & O(6N^3 + N^2 + 5N). \quad (15)
 \end{aligned}$$

The spectral decomposition $\Sigma_{B'} = U \Lambda_{B'} U^T$ gives the eigenvalues $\text{diag}(\Lambda_{B'}) = \{\lambda_1(B'), \dots, \lambda_N(B')\}$ to compute $\Sigma_{B'}^{-1}$ and $\det(\Sigma_{B'})$ efficiently as illustrated in the example below. (Further computing the spectral decomposition of Σ_B to compute $\det(\Sigma_B)$ from the eigenvalues $\{\lambda_1(B), \dots, \lambda_N(B)\}$ does not improve complexity because it just replaces a single $O(N^3)$ operation with another one.) We thus somewhat improve the overall complexity of $\text{KL}(B \parallel B')$ to $O(5N^3 + N^2 + 6N)$.

Example 7 (General-case KL between two GBNs). Consider the GBN B Figure 1 top, which we know has global distribution

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} \sim N \left(\begin{bmatrix} 2.400 \\ 1.800 \\ 12.276 \\ 8.848 \end{bmatrix}, \begin{bmatrix} 0.800 & 0 & 1.440 & 1.200 \\ 0 & 0.600 & 1.872 & 1.560 \\ 1.440 & 1.872 & 10.916 & 8.347 \\ 1.200 & 1.560 & 8.347 & 6.956 \end{bmatrix} \right)$$

from Example 2. Furthermore, consider the GBN B' from Figure 1 bottom, which has global distribution

$$\begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} \sim N \left(\begin{bmatrix} 2.400 \\ 11.324 \\ 6.220 \\ 4.620 \end{bmatrix}, \begin{bmatrix} 0.800 & 2.368 & 1.040 & 0.640 \\ 2.368 & 8.541 & 3.438 & 1.894 \\ 1.040 & 3.438 & 1.652 & 0.832 \\ 0.640 & 1.894 & 0.832 & 1.012 \end{bmatrix} \right).$$

In order to compute $\text{KL}(B \parallel B')$, we first invert $\Sigma_{B'}$ to obtain

$$\Sigma_{B'}^{-1} = \begin{bmatrix} 9.945 & -1.272 & -2.806 & -1.600 \\ -1.272 & 0.909 & -1.091 & 0 \\ -2.806 & -1.091 & 4.642 & 0 \\ -1.600 & 0 & 0 & 2.000 \end{bmatrix},$$

which we then multiply by Σ_B to compute the trace $\text{tr}(\Sigma_{B'}^{-1} \Sigma_B) = 57.087$. We also use $\Sigma_{B'}^{-1}$ to compute $(\mu_{B'} - \mu_B)^T \Sigma_{B'}^{-1} (\mu_{B'} - \mu_B) = 408.362$. Finally, $\det(\Sigma_{B'}) = 0.475$, $\det(\Sigma_B) = 0.132$ and therefore

$$\text{KL}(B \parallel B') = \frac{1}{2} \left[57.087 + 408.362 - 4 + \log \left(\frac{0.475}{0.132} \right) \right] = 230.0846. \quad (16)$$

As an alternative, we can compute the spectral decompositions $\Sigma_B = U_B \Lambda_B U_B^T$ and $\Sigma_{B'} = U_{B'} \Lambda_{B'} U_{B'}^T$ as an intermediate step. Multiplying the sets of eigenvalues

$$\Lambda_B = \text{diag}(\{18.058, 0.741, 0.379, 0.093\}) \quad \text{and} \quad \Lambda_{B'} = \text{diag}(\{11.106, 0.574, 0.236, 0.087\})$$

gives the corresponding determinants; and it allows us to easily compute

$$\Sigma_{B'}^{-1} = U_{B'} \Lambda_{B'}^{-1} U_{B'}^T, \quad \text{where} \quad \Lambda_{B'}^{-1} = \text{diag} \left(\left\{ \frac{1}{11.106}, \frac{1}{0.574}, \frac{1}{0.236}, \frac{1}{0.087} \right\} \right)$$

for use in both the quadratic form and in the trace.

However, computing $\text{KL}(\mathcal{B} \parallel \mathcal{B}')$ from the global distributions $N(\mu_{\mathcal{B}}, \Sigma_{\mathcal{B}})$ and $N(\mu_{\mathcal{B}'}, \Sigma_{\mathcal{B}'})$ disregards the fact that BNs are sparse models that can be characterised more compactly by $(\mu_{\mathcal{B}}, C_{\mathcal{B}})$ and $(\mu_{\mathcal{B}'}, C_{\mathcal{B}'})$ as shown in Section 3.2. In particular, we can revisit several operations that are in the high-order terms of (15):

- *Composing the global distribution from the local ones.* We avoid computing $\Sigma_{\mathcal{B}}$ and $\Sigma_{\mathcal{B}'}$, thus reducing this step to $O(2N)$ complexity.
- *Computing the trace $\text{tr}(\Sigma_{\mathcal{B}'}^{-1} \Sigma_{\mathcal{B}})$.* We can reduce the computation of the trace as follows.
 1. We can replace $\Sigma_{\mathcal{B}}$ and $\Sigma_{\mathcal{B}'}$ in the trace with any reordered matrix [53] (Result 8.17): we choose to use $\tilde{\Sigma}_{\mathcal{B}'}$ and $\tilde{\Sigma}_{\mathcal{B}}^*$ where $\tilde{\Sigma}_{\mathcal{B}'}$ is defined as before and $\tilde{\Sigma}_{\mathcal{B}}^*$ is $\Sigma_{\mathcal{B}}$ with the rows and columns reordered to match $\tilde{\Sigma}_{\mathcal{B}'}$. Formally, this is equivalent to $\tilde{\Sigma}_{\mathcal{B}}^* = P \Sigma_{\mathcal{B}} P^T$ where P is a permutation matrix that imposes the desired node ordering: since both the rows and the columns are permuted in the same way, the diagonal elements of $\tilde{\Sigma}_{\mathcal{B}}$ are the same as those of $\tilde{\Sigma}_{\mathcal{B}}^*$ and the trace is unaffected.
 2. We have $\tilde{\Sigma}_{\mathcal{B}'} = C_{\mathcal{B}'} C_{\mathcal{B}'}^T$.
 3. As for $\tilde{\Sigma}_{\mathcal{B}}^*$, we can write $\tilde{\Sigma}_{\mathcal{B}}^* = P \tilde{\Sigma}_{\mathcal{B}} P = (P C_{\mathcal{B}})(P C_{\mathcal{B}})^T = C_{\mathcal{B}}^* (C_{\mathcal{B}}^*)^T$ where $C_{\mathcal{B}}^* = P C_{\mathcal{B}}$ is the lower triangular matrix $C_{\mathcal{B}}$ with the rows re-ordered to match $\tilde{\Sigma}_{\mathcal{B}'}$. Note that $C_{\mathcal{B}}^*$ is not lower triangular unless \mathcal{G} and \mathcal{G}' have the same partial node ordering, which implies $P = I_N$.

Therefore

$$\text{tr}(\Sigma_{\mathcal{B}'}^{-1} \Sigma_{\mathcal{B}}) = \text{tr}\left((C_{\mathcal{B}'}^{-1} C_{\mathcal{B}}^*)^T (C_{\mathcal{B}'}^{-1} C_{\mathcal{B}}^*)\right) = \|C_{\mathcal{B}'}^{-1} C_{\mathcal{B}}^*\|_F^2 \quad (17)$$

where the last step rests on Seber [53] (Result 4.15). We can invert $C_{\mathcal{B}'}$ in $O(N^2)$ time following Stewart [54] (Algorithm 2.3). Multiplying $C_{\mathcal{B}'}^{-1}$ and $C_{\mathcal{B}}^*$ is still $O(N^3)$. The Frobenius norm $\|\cdot\|_F$ is $O(N^2)$ since it is the sum of the squared elements of $C_{\mathcal{B}'}^{-1} C_{\mathcal{B}}^*$.

- *Computing the determinants $\det(\Sigma_{\mathcal{B}'})$ and $\det(\Sigma_{\mathcal{B}})$.* From (13), each determinant can be computed in $O(N)$.
- *Computing the quadratic term $(\mu_{\mathcal{B}'} - \mu_{\mathcal{B}})^T \Sigma_{\mathcal{B}'}^{-1} (\mu_{\mathcal{B}'} - \mu_{\mathcal{B}})$.* Decomposing $\Sigma_{\mathcal{B}'}^{-1}$ leads to

$$(\mu_{\mathcal{B}'} - \mu_{\mathcal{B}})^T \Sigma_{\mathcal{B}'}^{-1} (\mu_{\mathcal{B}'} - \mu_{\mathcal{B}}) = (C_{\mathcal{B}'}^{-1} (\mu_{\mathcal{B}'}^* - \mu_{\mathcal{B}}^*))^T C_{\mathcal{B}'}^{-1} (\mu_{\mathcal{B}'}^* - \mu_{\mathcal{B}}^*), \quad (18)$$

where $\mu_{\mathcal{B}'}^*$ and $\mu_{\mathcal{B}}^*$ are the mean vectors re-ordered to match $C_{\mathcal{B}'}^{-1}$. The computational complexity is still $O(N^2 + 2N)$ because $C_{\mathcal{B}'}^{-1}$ is available from previous computations.

Combining (17), (13) and (18), the expression in (14) becomes

$$\text{KL}(\mathcal{B} \parallel \mathcal{B}') = \frac{1}{2} \left[\|C_{\mathcal{B}'}^{-1} C_{\mathcal{B}}^*\|_F^2 + (C_{\mathcal{B}'}^{-1} (\mu_{\mathcal{B}'}^* - \mu_{\mathcal{B}}^*))^T C_{\mathcal{B}'}^{-1} (\mu_{\mathcal{B}'}^* - \mu_{\mathcal{B}}^*) - N + 2 \log \frac{\prod_{i=1}^N C_{\mathcal{B}'}[i; i]}{\prod_{i=1}^N C_{\mathcal{B}}[i; i]} \right]. \quad (19)$$

The overall complexity of (19) KL is

$$\underbrace{O(2N^2 + 2N)}_{\text{compute } \mu_{\mathcal{B}}, \mu_{\mathcal{B}'}, C_{\mathcal{B}}, C_{\mathcal{B}'}} + \underbrace{O(2N^2 + N^3)}_{\text{compute } \|C_{\mathcal{B}'}^{-1} C_{\mathcal{B}}\|_F^2} + \underbrace{O(N^2 + 2N)}_{\text{compute the quadratic form}} + \underbrace{O(2N)}_{\text{compute } \det(\Sigma_{\mathcal{B}}), \det(\Sigma_{\mathcal{B}'})} = O(N^3 + 5N^2 + 6N); \quad (20)$$

while still cubic, the leading coefficient suggests that it should be about 5 times faster than the variant of (15) using the spectral decomposition.

Example 8 (Sparse KL between two GBNs). Consider again the two GBNs from Example 7. The corresponding matrices

$$C_B = \begin{matrix} & X_1 & X_2 & X_4 & X_3 \\ \begin{matrix} X_1 \\ X_2 \\ X_4 \\ X_3 \end{matrix} & \begin{pmatrix} 0.894 & 0 & 0 & 0 \\ 0 & 0.774 & 0 & 0 \\ 1.341 & 2.014 & 1.049 & 0 \\ 1.610 & 2.416 & 1.258 & 0.948 \end{pmatrix} \end{matrix}, \quad C_{B'} = \begin{matrix} & X_1 & X_3 & X_4 & X_2 \\ \begin{matrix} X_1 \\ X_3 \\ X_4 \\ X_2 \end{matrix} & \begin{pmatrix} 0.894 & 0 & 0 & 0 \\ 1.163 & 0.548 & 0 & 0 \\ 0.715 & 0 & 0.707 & 0 \\ 2.647 & 0.657 & 0 & 1.049 \end{pmatrix} \end{matrix}$$

readily give the determinants of Σ_B and $\Sigma_{B'}$ following (13):

$$\det(C_B) = (0.894 \cdot 0.774 \cdot 1.049 \cdot 0.948)^2 = 0.475,$$

$$\det(C_{B'}) = (0.894 \cdot 0.548 \cdot 0.707 \cdot 1.049)^2 = 0.132.$$

As for the Frobenius norm in (17), we first invert $C_{B'}$ to obtain

$$C_{B'}^{-1} = \begin{matrix} & X_1 & X_3 & X_4 & X_2 \\ \begin{matrix} X_1 \\ X_3 \\ X_4 \\ X_2 \end{matrix} & \begin{pmatrix} 1.118 & 0 & 0 & 0 \\ -2.373 & 1.825 & 0 & 0 \\ -1.131 & 0 & 1.414 & 0 \\ -1.334 & -1.144 & 0 & 0.953 \end{pmatrix} \end{matrix};$$

then we reorder the rows and columns of C_B to follow the same node ordering as $C_{B'}$ and compute

$$\left\| \begin{pmatrix} 1.118 & 0 & 0 & 0 \\ -2.373 & 1.825 & 0 & 0 \\ -1.131 & 0 & 1.414 & 0 \\ -1.334 & -1.144 & 0 & 0.953 \end{pmatrix} \begin{pmatrix} 0.894 & 0 & 0 & 0 \\ 1.610 & 0.948 & 1.258 & 2.416 \\ 1.341 & 0 & 1.049 & 2.014 \\ 0 & 0 & 0 & 0.774 \end{pmatrix} \right\|_F^2 = 57.087$$

which, as expected, matches the value of $\text{tr}(\Sigma_{B'}^{-1}\Sigma_B)$ we computed in Example 7. Finally, $C_{B'}^{-1}(\mu_{B'}^* - \mu_B^*)$ in (18) is

$$\begin{pmatrix} 1.118 & 0 & 0 & 0 \\ -2.373 & 1.825 & 0 & 0 \\ -1.131 & 0 & 1.414 & 0 \\ -1.334 & -1.144 & 0 & 0.953 \end{pmatrix} \left[\begin{pmatrix} 2.400 \\ 6.220 \\ 4.620 \\ 11.324 \end{pmatrix} - \begin{pmatrix} 2.400 \\ 12.1276 \\ 8.848 \\ 1.800 \end{pmatrix} \right] = \begin{pmatrix} 0 \\ -11.056 \\ -5.459 \\ 16.010 \end{pmatrix}.$$

The quadratic form is then equal to 408.362, which matches the value of $(\mu_{B'} - \mu_B)^T \Sigma_{B'}^{-1}(\mu_{B'} - \mu_B)$ in Example 7. As a result, the expression for $\text{KL}(\mathcal{B} \parallel \mathcal{B}')$ is the same as in (16).

We can further reduce the complexity (20) of (19) when an approximate value of KL is suitable for our purposes. The only term with cubic complexity is $\text{tr}(\Sigma_{B'}^{-1}\Sigma_B) = \|C_{B'}^{-1}C_B^*\|_F^2$: reducing it to quadratic complexity or lower will eliminate the leading term of (20), making it quadratic in complexity. One way to do this is to compute a lower and an upper bound for $\text{tr}(\Sigma_{B'}^{-1}\Sigma_B)$, which can serve as an interval estimate, and take their geometric mean as an approximate point estimate.

A lower bound is given by Seber [53] (Result 10.39):

$$\text{tr}(\Sigma_{B'}^{-1}\Sigma_B) \geq \log \det(\Sigma_{B'}^{-1}\Sigma_B) + N = -\log \det(\Sigma_{B'}) + \log \det(\Sigma_B) + N, \quad (21)$$

which conveniently reuses the values of $\det(\Sigma_B)$ and $\det(\Sigma_{B'})$ we have from (13). For an upper bound, Seber [53] (Result 10.59) combined with Seber [53] (Result 4.15) gives

$$\text{tr}(\Sigma_{B'}^{-1}\Sigma_B) \leq \text{tr}(\Sigma_{B'}^{-1}) \text{tr}(\Sigma_B) = \text{tr}((C_{B'}C_B^T)^{-1}) \text{tr}(C_B C_B^T) = \|C_{B'}^{-1}\|_F^2 \|C_B\|_F^2, \quad (22)$$

a function of C_B and $C_{B'}$ that can be computed in $O(2N^2)$ time. Note that, as far as the point estimate is concerned, we do not care about how wide the interval is: we only need its geometric mean to be an acceptable approximation of $\text{tr}(\Sigma_{B'}^{-1}\Sigma_B)$.

Example 9 (Approximate KL). From Example 7, we have that $\text{tr}(\Sigma_{B'}^{-1}\Sigma_B) = 57.087$, $\det(\Sigma_{B'}) = 0.475$ and $\det(\Sigma_B) = 0.132$. The lower bound in (21) is then

$$-\log \det(\Sigma_{B'}) + \log \det(\Sigma_B) + 4 = 5.281$$

and the upper bound in (22) is

$$\|C_{B'}^{-1}\|_F^2 \|C_B\|_F^2 = 17.496 \cdot 19.272 = 337.207.$$

Their geometric mean is 42.199, which can serve as an approximate value for $\text{KL}(\mathcal{B} \parallel \mathcal{B}')$.

If we are comparing two GBNs whose parameters (but not necessarily network structures) have been learned from the same data, we can sometimes approximate $\text{KL}(\mathcal{B} \parallel \mathcal{B}')$ using the local distributions $X_i \mid \Pi_{X_i}^{\mathcal{B}}$ and $X_i \mid \Pi_{X_i}^{\mathcal{B}'}$ directly. If \mathcal{B} and \mathcal{B}' have compatible partial orderings, we can define a common total node ordering for both such that

$$\begin{aligned} \text{KL}(\mathcal{B} \parallel \mathcal{B}') &= \text{KL}\left(X_{(1)} \mid \{X_{(2)}, \dots, X_{(N)}\} \cdots X_N \parallel X_{(1)} \mid \{X_{(2)}, \dots, X_{(N)}\} \cdots X_N\right) \\ &= \text{KL}\left(X_{(1)} \mid \Pi_{X_{(1)}}^{\mathcal{B}} \cdots X_{(N)} \mid \Pi_{X_{(N)}}^{\mathcal{B}} \parallel X_{(1)} \mid \Pi_{X_{(1)}}^{\mathcal{B}'} \cdots X_{(N)} \mid \Pi_{X_{(N)}}^{\mathcal{B}'}\right). \end{aligned}$$

By “compatible partial orderings”, we mean two partial orderings that can be sorted into at least one shared total node ordering that is compatible with both. The product of the local distributions in the second step is obtained from the chain decomposition in the first step by considering the nodes in the conditioning other than the parents to have associated regression coefficients equal to zero. Then, following the derivations in Cavanaugh [55] for a general linear regression model, we can write the empirical approximation

$$\text{KL}\left(X_i \mid \Pi_{X_i}^{\mathcal{B}} \parallel X_i \mid \Pi_{X_i}^{\mathcal{B}'}\right) \approx \frac{1}{2} \left(\log \frac{\hat{\sigma}_{X_i}^2(\mathcal{B}')}{\hat{\sigma}_{X_i}^2(\mathcal{B})} + \frac{\hat{\sigma}_{X_i}^2(\mathcal{B})}{\hat{\sigma}_{X_i}^2(\mathcal{B}')} - 1 \right) + \frac{1}{2n} \left(\frac{\|\hat{\mathbf{x}}_i(\mathcal{B}) - \hat{\mathbf{x}}_i(\mathcal{B}')\|_2^2}{\hat{\sigma}_{X_i}^2(\mathcal{B}')} \right) \quad (23)$$

where, following a similar notation to (4):

- $\hat{\mu}_{X_i}(\mathcal{B})$, $\hat{\beta}_{X_i}(\mathcal{B})$, $\hat{\mu}_{X_i}(\mathcal{B}')$, $\hat{\beta}_{X_i}(\mathcal{B}')$ are the estimated intercepts and regression coefficients;
- $\hat{\mathbf{x}}_i(\mathcal{B})$ and $\hat{\mathbf{x}}_i(\mathcal{B}')$ are the $n \times 1$ vectors

$$\hat{\mathbf{x}}_i(\mathcal{B}) = \hat{\mu}_{X_i}(\mathcal{B}) + \mathbf{x}[\cdot; \Pi_{X_i}(\mathcal{B})] \hat{\beta}_{X_i}(\mathcal{B}), \quad \hat{\mathbf{x}}_i(\mathcal{B}') = \hat{\mu}_{X_i}(\mathcal{B}') + \mathbf{x}[\cdot; \Pi_{X_i}(\mathcal{B}')] \hat{\beta}_{X_i}(\mathcal{B}'),$$

the fitted values computed from the data observed for X_i , $\Pi_{X_i}(\mathcal{B})$, $\Pi_{X_i}(\mathcal{B}')$;

- $\sigma_{X_i}^2(\mathcal{B})$ and $\sigma_{X_i}^2(\mathcal{B}')$ are the residual variances in \mathcal{B} and \mathcal{B}' .

We can compute the expression in (23) for each node in

$$\underbrace{O(n(|\Pi_{X_i}(\mathcal{B})| + |\Pi_{X_i}(\mathcal{B}')| + 2))}_{\text{compute } \hat{\mathbf{x}}_i(\mathcal{B}) \text{ and } \hat{\mathbf{x}}_i(\mathcal{B}')} + \underbrace{O(n)}_{\text{compute the norm } \|\hat{\mathbf{x}}_i(\mathcal{B}) - \hat{\mathbf{x}}_i(\mathcal{B}')\|_2^2} = O(n(|\Pi_{X_i}(\mathcal{B})| + |\Pi_{X_i}(\mathcal{B}')| + 5/2)),$$

which is linear in the sample size if both \mathcal{G} and \mathcal{G}' are sparse because $|\Pi_{X_i}(\mathcal{B})| \leq c$, $|\Pi_{X_i}(\mathcal{B}')| \leq c$. In this case, the overall computational complexity simplifies to $O(nN(2c + 5/2))$. Furthermore, as we pointed out in Scutari et al. [29], the fitted values $\hat{\mathbf{x}}_i(\mathcal{B})$, $\hat{\mathbf{x}}_i(\mathcal{B}')$ are computed as a by-product of parameter learning: if we consider them to be already available, the above computational complexity is reduced to just $O(n)$ for a single node and $O(nN)$ overall.

We can also replace the fitted values $\hat{\mathbf{x}}_i(\mathcal{B})$, $\hat{\mathbf{x}}_i(\mathcal{B}')$ in (23) with the corresponding residuals $\hat{\varepsilon}_i(\mathcal{B})$, $\hat{\varepsilon}_i(\mathcal{B}')$ because

$$\|\hat{\mathbf{x}}_i(\mathcal{B}) - \hat{\mathbf{x}}_i(\mathcal{B}')\|_2^2 = \|(\mathbf{x}[\cdot; X_i] - \hat{\mathbf{x}}_i(\mathcal{B})) - (\mathbf{x}[\cdot; X_i] - \hat{\mathbf{x}}_i(\mathcal{B}'))\|_2^2 = \|\hat{\varepsilon}_i(\mathcal{B}) - \hat{\varepsilon}_i(\mathcal{B}')\|_2^2$$

if the latter are available but the former are not.

Example 10 (KL between GBNs with parameters estimated from data). *For reasons of space, this example is presented as Example A5 in Appendix B.*

4.3. Conditional Gaussian BNs

The entropy $H(\mathcal{B})$ decomposes into a separate $H(X_i | \Pi_{X_i}^{\mathcal{B}})$ for each node, of the form (9) for discrete nodes and (12) for continuous nodes with no discrete parents. For continuous nodes with both discrete and continuous parents,

$$H(X_i | \Pi_{X_i}^{\mathcal{B}}) = \frac{1}{2} \sum_{\delta_{X_i} \in \text{Val}(\Delta_{X_i})} \pi_{\delta_{X_i}} \log(2\pi\sigma_{X_i, \delta_{X_i}}^2(\mathcal{B})) + \frac{1}{2}, \quad (24)$$

where $\pi_{\delta_{X_i}}$ represents the probability associated with the configuration δ_{X_i} of the discrete parents Δ_{X_i} . This last expression can be computed in $O(|\text{Val}(\Delta_{X_i})|)$ time for each node. Overall, the complexity of computing $H(\mathcal{B})$ is

$$O\left(\sum_{X_i \in \mathbf{X}_D} |\Theta_{X_i}| + \sum_{X_i \in \mathbf{X}_G} \max\{1, |\text{Val}(\Delta_{X_i})|\}\right).$$

where the max accounts for the fact that $|\text{Val}(\Delta_{X_i})| = 0$ when $\Delta_{X_i} = \emptyset$ but the computational complexity is $O(1)$ for such nodes.

Example 11 (Entropy of a CLGBN). *For reasons of space, this example is presented as Example A6 in Appendix B.*

As for $\text{KL}(\mathcal{B} \parallel \mathcal{B}')$, we could not find any literature illustrating how to compute it. The partition of the nodes in (5) implies that

$$\text{KL}(\mathcal{B} \parallel \mathcal{B}') = \underbrace{\text{KL}(\mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_D^{\mathcal{B}'})}_{\text{discrete nodes}} + \underbrace{\text{KL}(\mathbf{X}_G^{\mathcal{B}} \mid \mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_G^{\mathcal{B}'} \mid \mathbf{X}_D^{\mathcal{B}'})}_{\text{continuous nodes}}. \quad (25)$$

We can compute the first term following Section 4.1: $\mathbf{X}_D^{\mathcal{B}}$ and $\mathbf{X}_D^{\mathcal{B}'}$ form two discrete BNs whose DAGs are the spanning subgraphs of \mathcal{B} and \mathcal{B}' and whose local distributions are the corresponding ones in \mathcal{B} and \mathcal{B}' , respectively. The second term decomposes into

$$\text{KL}(\mathbf{X}_G^{\mathcal{B}} \mid \mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_G^{\mathcal{B}'} \mid \mathbf{X}_D^{\mathcal{B}'}) = \sum_{\mathbf{x}_D \in \text{Val}(\mathbf{X}_D)} P(\mathbf{X}_D^{\mathcal{B}} = \mathbf{x}_D) \text{KL}(\mathbf{X}_G^{\mathcal{B}} \mid \mathbf{X}_D^{\mathcal{B}} = \mathbf{x}_D \parallel \mathbf{X}_G^{\mathcal{B}'} \mid \mathbf{X}_D^{\mathcal{B}'} = \mathbf{x}_D) \quad (26)$$

similarly to (10) and (24). We can compute it using the multivariate normal distributions associated with the $\mathbf{X}_D^{\mathcal{B}} = \mathbf{x}_D$ and the $\mathbf{X}_D^{\mathcal{B}'} = \mathbf{x}_D$ in the global distributions of \mathcal{B} and \mathcal{B}' .

Example 12 (General-case KL between two CLGBNs). *Consider the CLGBNs \mathcal{B} from Figure 3 top, which we already used in Examples 3 and 11, and \mathcal{B}' from Figure 3 bottom. The variables $\mathbf{X}_D^{\mathcal{B}'}$ identify the following mixture components in the global distribution of \mathcal{B}' :*

$$\begin{aligned} \{a, c, e\}, \{b, c, e\}, \{a, d, e\}, \{b, d, e\} &\mapsto \{e\}, \\ \{a, c, f\}, \{b, c, f\}, \{a, d, f\}, \{b, d, f\} &\mapsto \{f\}. \end{aligned}$$

Therefore, \mathcal{B}' only encodes two different multivariate normal distributions.

Firstly, we construct two discrete BNs using the subgraphs spanning $\mathbf{X}_D^{\mathcal{B}} = \mathbf{X}_D^{\mathcal{B}'} = \{X_1, X_2, X_3\}$ in \mathcal{B} and \mathcal{B}' , which have arcs $\{X_1 \rightarrow X_2\}$ and $\{X_1 \rightarrow X_2, X_2 \rightarrow X_3\}$, respectively. The CPTs for X_1, X_2 and X_3 are the same as in \mathcal{B} and in \mathcal{B}' . We then compute $\text{KL}(\mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_D^{\mathcal{B}'}) = 0.577$ following Example 5.

Secondly, we construct the multivariate normal distributions associated with the components of \mathcal{B}' following Example 3 (in which we computed those of \mathcal{B}). For $\{e\}$, we have

$$\begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix} \sim N \left(\begin{bmatrix} 0.300 \\ 1.400 \\ 1.140 \end{bmatrix}, \begin{bmatrix} 0.160 & 0.000 & 0.032 \\ 0.000 & 1.690 & 1.183 \\ 0.032 & 1.183 & 2.274 \end{bmatrix} \right);$$

for $\{f\}$, we have

$$\begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix} \sim N \left(\begin{bmatrix} 1.000 \\ 0.500 \\ 0.650 \end{bmatrix}, \begin{bmatrix} 0.090 & 0.000 & 0.018 \\ 0.000 & 2.250 & 1.575 \\ 0.018 & 1.575 & 2.546 \end{bmatrix} \right).$$

Then,

$$\begin{aligned} & \text{KL}(\mathbf{X}_G^{\mathcal{B}} \mid \mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_G^{\mathcal{B}'} \mid \mathbf{X}_D^{\mathcal{B}'}) \\ &= \sum_{x_1 \in \{a,b\}} \sum_{x_2 \in \{c,d\}} \sum_{x_3 \in \{e,f\}} P(\mathbf{X}_D^{\mathcal{B}} = \{x_1, x_2, x_3\}) \cdot \\ & \quad \text{KL}(\mathbf{X}_G^{\mathcal{B}} \mid \mathbf{X}_D^{\mathcal{B}} = \{x_1, x_2, x_3\} \parallel \mathbf{X}_G^{\mathcal{B}'} \mid \mathbf{X}_D^{\mathcal{B}'} = \{x_1, x_2, x_3\}) \\ &= \underbrace{0.040 \times 1.721}_{\{a,c,e\}} + \underbrace{0.036 \times 1.721}_{\{b,c,e\}} + \underbrace{0.040 \times 2.504}_{\{a,d,e\}} + \underbrace{0.084 \times 2.504}_{\{b,d,e\}} + \\ & \quad \underbrace{0.16 \times 4.303}_{\{a,c,f\}} + \underbrace{0.144 \times 4.303}_{\{b,c,f\}} + \underbrace{0.16 \times 6.31}_{\{a,d,f\}} + \underbrace{0.336 \times 6.31}_{\{b,d,f\}} \\ &= 4.879 \end{aligned}$$

$$\text{and } \text{KL}(\mathcal{B} \parallel \mathcal{B}') = \text{KL}(\mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_D^{\mathcal{B}'}) + \text{KL}(\mathbf{X}_G^{\mathcal{B}} \mid \mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_G^{\mathcal{B}'} \mid \mathbf{X}_D^{\mathcal{B}'} = \mathbf{x}_D) = 0.577 + 4.879 = 5.456.$$

The computational complexity of this basic approach to computing $\text{KL}(\mathcal{B} \parallel \mathcal{B}')$ is

$$\underbrace{O(Mwl^{w+c} + M(w + wl^w + l^{w-1}) + (Ml^c + 2)|\Theta_{\mathbf{x}_D}|)}_{\text{compute } \text{KL}(\mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_D^{\mathcal{B}'})} + \underbrace{O(l^M \cdot (6(N-M)^3 + (N-M)^2 + 5(N-M)))}_{\text{compute all the } \text{KL}(\mathbf{X}_G^{\mathcal{B}} \mid \mathbf{X}_D^{\mathcal{B}} = \mathbf{x}_D \parallel \mathbf{X}_G^{\mathcal{B}'} \mid \mathbf{X}_D^{\mathcal{B}'} = \mathbf{x}_D)}, \quad (27)$$

which we obtain by adapting (11) and (15) to follow the notation $|\mathbf{X}_D| = M$ and $|\mathbf{X}_G| = N - M$ we established in Section 3.3. The first term implicitly covers the cost of computing the $P(\mathbf{X}_D^{\mathcal{B}} = \mathbf{x}_D)$, which relies on exact inference like the computation of $\text{KL}(\mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_D^{\mathcal{B}'})$. The second term is exponential in M , which would lead us to conclude that it is computationally unfeasible to compute $\text{KL}(\mathcal{B} \parallel \mathcal{B}')$ whenever we have more than a few discrete variables in \mathcal{B} and \mathcal{B}' . Certainly, this would agree with Hershey and Olsen [22], who reviewed various scalable approximations of the KL divergence between two Gaussian mixtures.

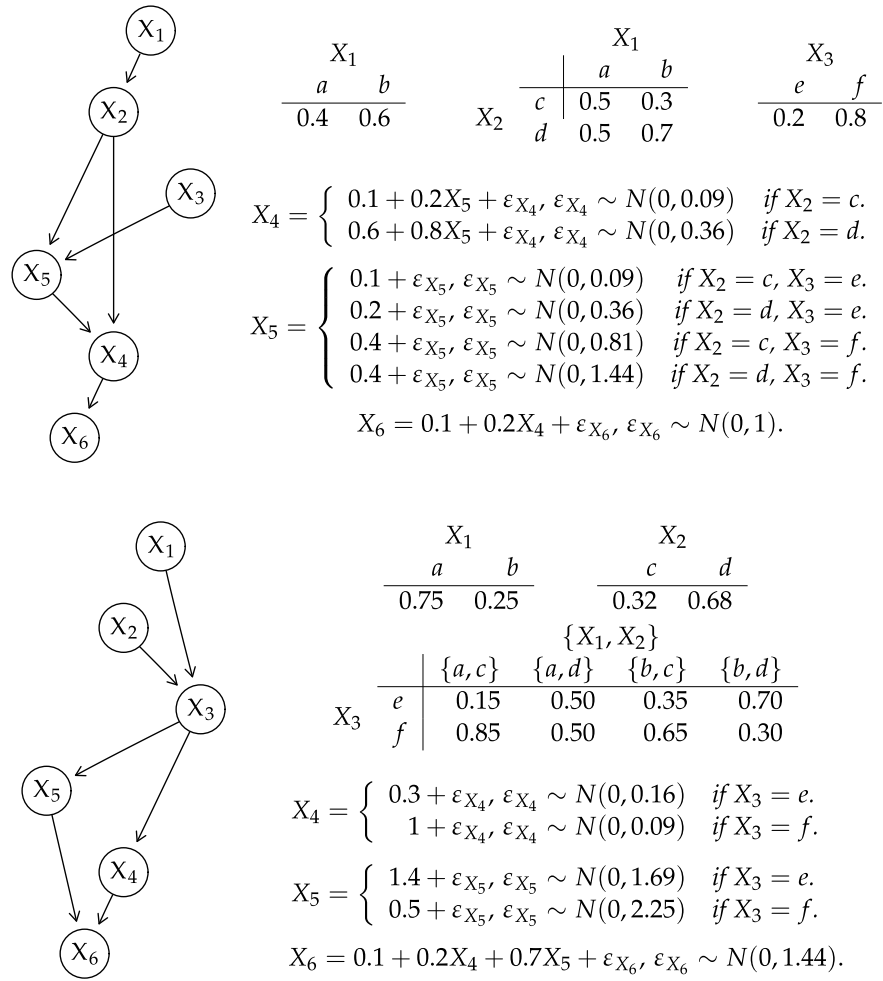


Figure 3. DAGs and local distributions for the CLGBNs \mathcal{B} (top) and \mathcal{B}' (bottom) used in Examples 3 and 11–13.

However, we would again disregard the fact that BNs are sparse models. Two properties of CLGBNs that are apparent from Examples 3 and 12 allow us to compute (26) efficiently:

- We can reduce $\mathbf{X}_G^{\mathcal{B}} | \mathbf{X}_D^{\mathcal{B}}$ to $\mathbf{X}_G^{\mathcal{B}} | \Delta^{\mathcal{B}}$ where $\Delta^{\mathcal{B}} = \bigcup_{X_i \in \mathbf{X}_G} \Delta_{X_i}^{\mathcal{B}} \subseteq \mathbf{X}_D^{\mathcal{B}}$. In other words, the continuous nodes are conditionally independent on the discrete nodes that are not their parents ($\mathbf{X}_D^{\mathcal{B}} \setminus \Delta^{\mathcal{B}}$) given their parents ($\Delta^{\mathcal{B}}$). The same is true for $\mathbf{X}_G^{\mathcal{B}'} | \mathbf{X}_D^{\mathcal{B}'}$. The number of distinct terms in the summation in (26) is then given by $|\text{Val}(\Delta^{\mathcal{B}} \cup \Delta^{\mathcal{B}'})|$ which will be smaller than $|\text{Val}(\mathbf{X}_D^{\mathcal{B}})|$ in sparse networks.
- The conditional distributions $\mathbf{X}_G^{\mathcal{B}} | \mathbf{X}_D^{\mathcal{B}} = \delta$ and $\mathbf{X}_G^{\mathcal{B}'} | \mathbf{X}_D^{\mathcal{B}'} = \delta$ are multivariate normals (not mixtures). They are also faithful to the subgraphs spanning the continuous nodes \mathbf{X}_G , and we can represent them as GBNs whose parameters can be extracted directly from \mathcal{B} and \mathcal{B}' . Therefore, we can use the results from Section 4.2 to compute their Kullback–Leibler divergences efficiently.

As a result, (26) simplifies to

$$\text{KL}(\mathbf{X}_G^{\mathcal{B}} | \mathbf{X}_D^{\mathcal{B}} \parallel \mathbf{X}_G^{\mathcal{B}'} | \mathbf{X}_D^{\mathcal{B}'}) = \sum_{\delta \in \text{Val}(\Delta^{\mathcal{B}} \cup \Delta^{\mathcal{B}'})} \text{P}(\{\Delta^{\mathcal{B}} \cup \Delta^{\mathcal{B}'}\} = \delta) \text{KL}(\mathbf{X}_G^{\mathcal{B}} | \{\Delta^{\mathcal{B}} \cup \Delta^{\mathcal{B}'}\} = \delta \parallel \mathbf{X}_G^{\mathcal{B}'} | \{\Delta^{\mathcal{B}} \cup \Delta^{\mathcal{B}'}\} = \delta).$$

where $P(\{\Delta^B \cup \Delta^{B'}\} = \delta)$ is the probability that the nodes $\Delta^B \cup \Delta^{B'}$ take value δ as computed in \mathcal{B} . In turn, (27) reduces to

$$\underbrace{O\left(Mwl^{w+c} + M(w + wl^w + l^{w-1}) + (Ml^c + 2)|\Theta_{\mathcal{X}_D}|\right)}_{\text{compute KL}(\mathbf{X}_D^B \parallel \mathbf{X}_D^{B'})} + \underbrace{O\left(l^{|\text{Val}(\Delta^B \cup \Delta^{B'})|} \cdot \left((N-M)^3 + 5(N-M)^2 + 6(N-M)\right)\right)}_{\text{compute all the KL}(\mathbf{X}_G^B \parallel \{\Delta^B \cup \Delta^{B'}\} = \delta \parallel \mathbf{X}_G^{B'} \parallel \{\Delta^B \cup \Delta^{B'}\} = \delta)}.$$

because we can replace l^M with $l^{|\text{Val}(\Delta^B \cup \Delta^{B'})|}$, which is an upper bound to the unique components in the mixture, and because we replace the complexity in (15) with that (20). We can also further reduce the second term to quadratic complexity as we discussed in Section 4.2. The remaining drivers of the computational complexity are:

- the maximum clique size w in the subgraph spanning \mathbf{X}_D^B ;
- the number of arcs from discrete nodes to continuous nodes in both \mathcal{B} and \mathcal{B}' and the overlap between Δ^B and $\Delta^{B'}$.

Example 13 (Sparse KL between two CLGBNs). Consider again the CLGBNs \mathcal{B} and \mathcal{B}' from Example 12. The node sets $\Delta^B = \{X_2, X_3\}$ and $\Delta^{B'} = \{X_3\}$ identify four KL divergences to compute: $\text{Val}(\Delta^B \cup \Delta^{B'}) = \{\{c, e\}, \{c, f\}, \{d, e\}, \{d, f\}\}$.

$$\begin{aligned} \text{KL}(\mathbf{X}_G^B \parallel \mathbf{X}_D^B \parallel \mathbf{X}_G^{B'} \parallel \mathbf{X}_D^{B'}) = & \\ & P(\{\Delta^B \cup \Delta^{B'}\} = \{c, e\}) \text{KL}(\mathbf{X}_G^B \parallel \{\Delta^B \cup \Delta^{B'}\} = \{c, e\} \parallel \mathbf{X}_G^{B'} \parallel \{\Delta^B \cup \Delta^{B'}\} = \{c, e\}) + \\ & P(\{\Delta^B \cup \Delta^{B'}\} = \{c, f\}) \text{KL}(\mathbf{X}_G^B \parallel \{\Delta^B \cup \Delta^{B'}\} = \{c, f\} \parallel \mathbf{X}_G^{B'} \parallel \{\Delta^B \cup \Delta^{B'}\} = \{c, f\}) + \\ & P(\{\Delta^B \cup \Delta^{B'}\} = \{d, e\}) \text{KL}(\mathbf{X}_G^B \parallel \{\Delta^B \cup \Delta^{B'}\} = \{d, e\} \parallel \mathbf{X}_G^{B'} \parallel \{\Delta^B \cup \Delta^{B'}\} = \{d, e\}) + \\ & P(\{\Delta^B \cup \Delta^{B'}\} = \{d, f\}) \text{KL}(\mathbf{X}_G^B \parallel \{\Delta^B \cup \Delta^{B'}\} = \{d, f\} \parallel \mathbf{X}_G^{B'} \parallel \{\Delta^B \cup \Delta^{B'}\} = \{d, f\}) \end{aligned}$$

All the BNs in the Kullback–Leibler divergences are GBNs whose structure and local distributions can be read from \mathcal{B} and \mathcal{B}' . The four GBNs associated with $\mathbf{X}_G^B \parallel \{\Delta^B \cup \Delta^{B'}\}$ have nodes $\mathbf{X}_G^B = \{X_4, X_5, X_6\}$, arcs $\{X_5 \rightarrow X_4, X_4 \rightarrow X_6\}$ and the local distributions listed in Figure 3. The corresponding GBNs associated with $\mathbf{X}_G^{B'} \parallel \{\Delta^B \cup \Delta^{B'}\}$ are, in fact, only two distinct GBNs associated with $\{e\}$ and $\{f\}$. They have arcs $\{X_4 \rightarrow X_6, X_5 \rightarrow X_6\}$ and local distributions: for $\{e\}$,

$$\begin{aligned} X_4 &= 0.3 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.16), \\ X_5 &= 1.4 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 1.69), \\ X_6 &= 0.1 + 0.2X_4 + 0.7X_5 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1.44); \end{aligned}$$

for $\{f\}$,

$$\begin{aligned} X_4 &= 1.0 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.09), \\ X_5 &= 0.5 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 2.25), \\ X_6 &= 0.1 + 0.2X_4 + 0.7X_5 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1.44). \end{aligned}$$

Plugging in the numbers,

$$\begin{aligned} \text{KL}(\mathbf{x}_G^B \mid \mathbf{x}_D^B \parallel \mathbf{x}_G^{B'} \mid \mathbf{x}_D^{B'}) &= \underbrace{0.076 \times 1.721}_{\{c,e\}} + \underbrace{0.304 \times 4.303}_{\{c,f\}} + \\ &\quad \underbrace{0.124 \times 2.504}_{\{d,e\}} + \underbrace{0.496 \times 6.310}_{\{d,f\}} = 4.879 \end{aligned}$$

which matches the value we computed in Example 12.

5. Conclusions

We started this paper by reviewing the three most common distributional assumptions for BNs: discrete BNs, Gaussian BNs (GBNs) and conditional linear Gaussian BNs (CLGBNs). Firstly, we reviewed the link between the respective global and local distributions, and we formalised the computational complexity of decomposing the former into the latter (and vice versa).

We then leveraged these results to study the complexity of computing Shannon's entropy. We can, of course, compute the entropy of a BN from its global distribution using standard results from the literature. (In the case of discrete BNs and CLGBNs, only for small networks because $|\Theta|$ grows combinatorially.) However, this is not computationally efficient because we incur the cost of composing the global distribution. While the entropy does not decompose along with the local distributions for either discrete BNs or CLGBNs, we show that it is nevertheless efficient to compute it from them.

Computing the Kullback–Leibler divergence between two BNs following the little material found in the literature is more demanding. The discrete case has been thoroughly investigated by Moral et al. [21]. However, the literature typically relies on composing the global distributions for GBNs and CLGBNs. Using the local distributions, thus leveraging the intrinsic sparsity of BNs, we showed how to compute the Kullback–Leibler divergence exactly with greater efficiency. For GBNs, we showed how to compute the Kullback–Leibler divergence approximately with quadratic complexity (instead of cubic). If the two GBNs have compatible node orderings and their parameters are estimated from the same data, we can also approximate their Kullback–Leibler divergence with complexity that scales with the number of parents of each node. All these results are summarised in Table A1 in Appendix A.

Finally, we provided step-by-step numeric examples of how to compute Shannon's entropy and the Kullback–Leibler divergence for discrete BNs, GBNs and CLGBNs. (See also Appendix B). Considering this is a highly technical topic, and no such examples are available anywhere in the literature, we feel that they are helpful in demystifying this topic and in integrating BNs into many general machine learning approaches.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Computational Complexity Results

For ease of reference, we summarise here all the computational complexity results in this paper, including the type of BN and the page where they have been derived.

Table A1. Summary of all the computational complexity results in this paper, including the type of BN and the page where they have been derived.

Composing and decomposing the global distributions		
$O(Nl^N + l \sum_{i=1}^N l^{ \Pi_{X_i} })$	discrete BNs	Section 3.1
$O(N^3)$	GBNs	Section 3.2
$O(Ml^M + (N - M)^3 l^\Delta)$	CLGBNs	Section 3.3
Computing Shannon's entropy		
$O(N(w(1 + l^w) + l^{w-1}) + \Theta)$	discrete BNs	Section 4.1
$O(N)$	GBNs	Section 4.2
$O(\sum_{X_i \in \mathbf{X}_D} \Theta_{X_i} + \sum_{X_i \in \mathbf{X}_G} \max\{1, \text{Val}(\Delta_{X_i}) \})$	CLGBNs	Section 4.3
Computing the Kullback–Leibler divergence		
$O(N^2 w l^{w+c} + N(w + w l^w + l^{w-1}) + (Nl^c + 2) \Theta)$	discrete BNs	Section 4.1
$O(6N^3 + N^2 + 5N)$	GBNs	Section 4.2
$O(Mw l^{w+c} + M(w + w l^w + l^{w-1}) + (Ml^c + 2) \Theta_{\mathbf{X}_D}) + O(l^M \cdot (6(N - M)^3 + (N - M)^2 + 5(N - M)))$	CLGBNs	Section 4.3
Sparse Kullback–Leibler divergence		
$O(N^3 + 5N^2 + 6N)$	GBNs	Section 4.2
$O(Mw l^{w+c} + M(w + w l^w + l^{w-1}) + (Ml^c + 2) \Theta_{\mathbf{X}_D}) + O(l^{ \text{Val}(\Delta^{\mathbf{B} \cup \mathbf{B}'}) \cdot ((N - M)^3 + 5(N - M)^2 + 6(N - M))})$	CLGBNs	Section 4.3
Approximate Kullback–Leibler divergence		
$O(7N^2 + 6N)$	GBNs	Section 4.2
Efficient empirical Kullback–Leibler divergence		
$O(nN(2c + 5/2))$	GBNs	Section 4.2

Appendix B. Additional Examples

Example A1 (Composing and decomposing a discrete BN). Consider the discrete BN \mathcal{B} shown in Figure 2 (top). Composing its global distribution entails computing the joint probabilities of all possible states of all variables,

$$\{a, b\} \times \{c, d\} \times \{e, f\} \times \{g, h\},$$

and arranging them in the following four-dimensional probability table in which each dimension is associated with one of the variables.

$X_1 = a$						$X_1 = b$					
$X_2 = c$			$X_2 = d$			$X_2 = c$			$X_2 = d$		
X_3			X_3			X_3			X_3		
X_4	e	f	X_4	e	f	X_4	e	f	X_4	e	f
g	0.005	0.064	g	0.052	0.037	g	0.013	0.040	g	0.050	0.026
h	0.022	0.089	h	0.210	0.051	h	0.051	0.056	h	0.199	0.036

The joint probabilities are computed by multiplying the appropriate cells of the CPTs, for instance

$$\begin{aligned}
 P(\mathbf{X} = \{a, d, f, h\}) &= \\
 P(X_1 = a) P(X_2 = d) P(X_3 = f \mid X_1 = a, X_2 = d) P(X_4 = h \mid X_3 = f) &= \\
 0.53 \cdot 0.66 \cdot 0.25 \cdot 0.58 &= 0.051.
 \end{aligned}$$

Conversely, we can decompose the global distribution into the local distributions by summing over all variables other than the nodes and their parents. For X_1 , this means

$$\begin{aligned} P(X_1 = a) &= \sum_{x_2 \in \{c,d\}} \sum_{x_3 \in \{e,f\}} \sum_{x_4 \in \{g,h\}} P(X_1 = a, X_2 = x_2, X_3 = x_3, X_4 = x_4) \\ &= 0.005 + 0.064 + 0.022 + 0.089 + 0.052 + 0.037 + 0.210 + 0.051 = 0.53, \\ P(X_1 = b) &= \sum_{x_2 \in \{c,d\}} \sum_{x_3 \in \{e,f\}} \sum_{x_4 \in \{g,h\}} P(X_1 = b, X_2 = x_2, X_3 = x_3, X_4 = x_4) \\ &= 0.013 + 0.040 + 0.051 + 0.056 + 0.050 + 0.026 + 0.199 + 0.036 = 0.47. \end{aligned}$$

Similarly, for X_2 we obtain

$$\begin{aligned} P(X_2 = c) &= \sum_{x_1 \in \{a,b\}} \sum_{x_3 \in \{e,f\}} \sum_{x_4 \in \{g,h\}} P(X_1 = x_1, X_2 = c, X_3 = x_3, X_4 = x_4) \\ &= 0.005 + 0.064 + 0.022 + 0.089 + 0.013 + 0.040 + 0.051 + 0.056 = 0.34, \\ P(X_2 = d) &= \sum_{x_1 \in \{a,b\}} \sum_{x_3 \in \{e,f\}} \sum_{x_4 \in \{g,h\}} P(X_1 = x_1, X_2 = d, X_3 = x_3, X_4 = x_4) \\ &= 0.052 + 0.037 + 0.210 + 0.051 + 0.050 + 0.026 + 0.199 + 0.036 = 0.66. \end{aligned}$$

For X_4 , we first compute the joint distribution of X_4 and X_3 by marginalising over X_1 and X_2 ,

$$\begin{aligned} &\underbrace{\begin{matrix} & e & f \\ g & \begin{pmatrix} 0.005 & 0.064 \\ 0.022 & 0.089 \end{pmatrix} \\ h & \end{matrix}}_{\{a,c\}} + \underbrace{\begin{matrix} & e & f \\ g & \begin{pmatrix} 0.052 & 0.037 \\ 0.210 & 0.051 \end{pmatrix} \\ h & \end{matrix}}_{\{a,d\}} + \underbrace{\begin{matrix} & e & f \\ g & \begin{pmatrix} 0.013 & 0.040 \\ 0.051 & 0.056 \end{pmatrix} \\ h & \end{matrix}}_{\{b,c\}} + \\ &\underbrace{\begin{matrix} & e & f \\ g & \begin{pmatrix} 0.050 & 0.026 \\ 0.199 & 0.036 \end{pmatrix} \\ h & \end{matrix}}_{\{b,d\}} = \begin{matrix} & e & f \\ g & \begin{pmatrix} 0.120 & 0.167 \\ 0.481 & 0.232 \end{pmatrix} \\ h & \end{matrix}; \end{aligned}$$

from which we obtain the CPT for $X_4 \mid X_3$ by normalising its columns.

As for X_3 , we marginalise over X_2 to obtain the joint distribution of X_3 , X_1 and X_2

$$\begin{array}{cccc} & \{a,c\} & \{a,d\} & \{b,c\} & \{b,d\} \\ e & (0.005 + 0.022 = 0.027) & (0.052 + 0.210 = 0.262) & (0.013 + 0.051 = 0.064) & (0.050 + 0.199 = 0.248) \\ f & (0.064 + 0.089 = 0.153) & (0.037 + 0.051 = 0.087) & (0.040 + 0.056 = 0.096) & (0.026 + 0.036 = 0.062) \end{array}$$

and we obtain the CPT for $X_3 \mid X_1, X_2$ by normalising its columns as we did earlier with X_4 .

Example A2 (Composing and decomposing a CLGBN). Consider the CLGBN \mathcal{B} from Figure 3 top. The $M = 3$ discrete variables at the top of the network have the joint distribution below:

$\{X_1, X_2, X_3\}$							
$\{a, c, e\}$	$\{b, c, e\}$	$\{a, d, e\}$	$\{b, d, e\}$	$\{a, c, f\}$	$\{b, c, f\}$	$\{a, d, f\}$	$\{b, d, f\}$
0.040	0.036	0.040	0.084	0.160	0.144	0.160	0.336

Its elements identify the components of the mixture that make up the global distribution of \mathcal{B} , and the associated probabilities are the probabilities of those components.

We can then identify which parts of the local distributions of the $N - M = 3$ continuous variables (X_4 , X_5 and X_6) we need to compute $P(X_4, X_5, X_6 \mid X_1, X_2, X_3)$ for each element of the mixture. The graphical structure of \mathcal{B} implies that $P(X_4, X_5, X_6 \mid X_1, X_2, X_3) = P(X_4, X_5, X_6 \mid X_2, X_3)$ because the continuous nodes are d -separated from X_1 by their parents. As

a result, the following mixture components will share identical distributions which only depend on the configurations of X_2 and X_3 :

$$\begin{aligned}\{a, c, e\}, \{b, c, e\} &\mapsto \{c, e\}, & \{a, d, e\}, \{b, d, e\} &\mapsto \{d, e\}, \\ \{a, c, f\}, \{b, c, f\} &\mapsto \{c, f\}, & \{a, d, f\}, \{b, d, f\} &\mapsto \{d, f\}.\end{aligned}$$

For the mixture components with a distribution identified by $\{c, e\}$, the relevant parts of the distributions of X_4 , X_5 and X_6 are:

$$\begin{aligned}X_4 &= 0.1 + 0.2X_5 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.09); \\ X_5 &= 0.1 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 0.09); \\ X_6 &= 0.1 + 0.2X_4 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1).\end{aligned}$$

We can treat them as the local distributions in a GBN over $\{X_4, X_5, X_6\}$ with a DAG equal to the subgraph of \mathcal{B} spanning only these nodes. If we follow the steps outlined in Section 3.2 and illustrated in Example 2, we obtain

$$\begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix} \sim N \left(\begin{bmatrix} 0.120 \\ 0.100 \\ 0.124 \end{bmatrix}, \Sigma_{\{c,e\}}(\mathcal{B}) = \begin{bmatrix} 0.094 & 0.018 & 0.019 \\ 0.018 & 0.090 & 0.004 \\ 0.019 & 0.004 & 1.004 \end{bmatrix} \right)$$

which is the multivariate normal distribution associated with the components $\{a, c, e\}$ and $\{b, c, e\}$ in the mixture. Similarly, the relevant parts of the distributions of X_4 , X_5 and X_6 for $\{d, e\}$ are

$$\begin{aligned}X_4 &= 0.6 + 0.8X_5 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.36); \\ X_5 &= 0.2 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 0.36); \\ X_6 &= 0.1 + 0.2X_4 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1);\end{aligned}$$

and jointly

$$\begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix} \sim N \left(\begin{bmatrix} 0.760 \\ 0.200 \\ 0.252 \end{bmatrix}, \Sigma_{\{d,e\}}(\mathcal{B}) = \begin{bmatrix} 0.590 & 0.288 & 0.118 \\ 0.288 & 0.360 & 0.058 \\ 0.118 & 0.058 & 1.024 \end{bmatrix} \right)$$

for the components $\{a, d, e\}$ and $\{b, d, e\}$. For the components $\{a, c, f\}$ and $\{b, c, f\}$, the local distributions identified by $\{c, f\}$ are

$$\begin{aligned}X_4 &= 0.1 + 0.2X_5 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.09); \\ X_5 &= 0.4 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 0.81); \\ X_6 &= 0.1 + 0.2X_4 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1);\end{aligned}$$

and the joint distribution of X_4 , X_5 and X_6 is

$$\begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix} \sim N \left(\begin{bmatrix} 0.180 \\ 0.400 \\ 0.136 \end{bmatrix}, \Sigma_{\{c,f\}}(\mathcal{B}) = \begin{bmatrix} 0.122 & 0.162 & 0.024 \\ 0.162 & 0.810 & 0.032 \\ 0.024 & 0.032 & 1.005 \end{bmatrix} \right).$$

Finally, the local distributions identified by $\{d, f\}$ are

$$\begin{aligned}X_4 &= 0.6 + 0.8X_5 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.36); \\ X_5 &= 0.4 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 1.44); \\ X_6 &= 0.1 + 0.2X_4 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1);\end{aligned}$$

and the joint distribution of X_4 , X_5 and X_6 for the components $\{a, d, f\}$, $\{b, d, f\}$ is

$$\begin{bmatrix} X_4 \\ X_5 \\ X_6 \end{bmatrix} \sim N \left(\begin{bmatrix} 0.920 \\ 0.400 \\ 0.284 \end{bmatrix}, \Sigma_{\{d,f\}}(\mathcal{B}) = \begin{bmatrix} 1.282 & 1.152 & 0.256 \\ 1.152 & 1.440 & 0.230 \\ 0.256 & 0.230 & 1.051 \end{bmatrix} \right).$$

We follow the same steps in reverse to decompose the global distribution into the local distributions. The joint distribution of \mathbf{X} is a mixture with multivariate normal components and the associated probabilities. The latter are a function of the discrete variables X_1 , X_2 , X_3 : rearranging them as the three-dimensional table

$X_1 = a$				$X_1 = b$			
		X_2				X_2	
		c	d			c	d
X_3	e	0.040	0.040	X_3	e	0.036	0.084
	f	0.160	0.160		f	0.144	0.336

gives us the typical representation of $P(X_1, X_2, X_3)$, which we can work with by operating over the different dimensions. We can then compute the conditional probability tables in the local distributions of X_1 and X_3 by marginalising over the remaining variables:

$$\begin{aligned} P(X_1) &= \sum_{X_2 \in \{c,d\}} \sum_{X_3 \in \{e,f\}} P(X_1, X_2, X_3) \\ &= \begin{pmatrix} \overset{a}{0.040 + 0.160 + 0.040 + 0.160} & \overset{b}{0.036 + 0.144 + 0.084 + 0.336} \end{pmatrix} \\ &= \begin{pmatrix} \overset{a}{0.4} & \overset{b}{0.6} \end{pmatrix}, \\ P(X_3) &= \sum_{X_1 \in \{a,b\}} \sum_{X_2 \in \{c,d\}} P(X_1, X_2, X_3) \\ &= \begin{pmatrix} \overset{e}{0.040 + 0.040 + 0.036 + 0.084} & \overset{f}{0.160 + 0.160 + 0.144 + 0.336} \end{pmatrix} \\ &= \begin{pmatrix} \overset{e}{0.2} & \overset{f}{0.8} \end{pmatrix}. \end{aligned}$$

As for X_2 , we marginalise over X_3 and normalise over X_1 to obtain

$$P(X_2 | X_1) = \sum_{X_3 \in \{e,f\}} \frac{P(X_1, X_2, X_3)}{P(X_1)} = \begin{matrix} \begin{matrix} a & b \end{matrix} \\ c \left(\begin{matrix} \frac{0.040 + 0.160}{0.4} & \frac{0.036 + 0.144}{0.6} \end{matrix} \right) \\ d \left(\begin{matrix} \frac{0.040 + 0.160}{0.4} & \frac{0.084 + 0.336}{0.6} \end{matrix} \right) \end{matrix} = \begin{matrix} \begin{matrix} a & b \end{matrix} \\ c \left(\begin{matrix} 0.5 & 0.3 \end{matrix} \right) \\ d \left(\begin{matrix} 0.5 & 0.7 \end{matrix} \right) \end{matrix}.$$

The multivariate normal distributions associated with the mixture components are a function of the continuous variables X_4 , X_5 , X_6 . X_4 has only one discrete parent (X_2), X_5 has two (X_2 and X_3) and X_6 has none. Therefore, we only need to examine four mixture components to obtain the parameters of the local distributions of all three variables: one for which $\{X_2 = c, X_3 = e\}$, one for which $\{X_2 = d, X_3 = e\}$, one for which $\{X_2 = c, X_3 = f\}$ and one for which $\{X_2 = d, X_3 = f\}$.

If we consider the first mixture component $\{a, c, e\}$, we can apply the steps described Section 3.2 to decompose it into the local distributions of X_4 , X_5 , X_6 and obtain

$$\begin{aligned} X_4 &= 0.1 + 0.2X_5 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.09); \\ X_5 &= 0.1 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 0.09); \\ X_6 &= 0.1 + 0.2X_4 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1). \end{aligned}$$

Similarly, the third mixture component $\{a, d, e\}$ yields

$$\begin{aligned} X_4 &= 0.6 + 0.8X_5 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.36); \\ X_5 &= 0.2 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 0.36); \\ X_6 &= 0.1 + 0.2X_4 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1). \end{aligned}$$

The fifth mixture component $\{a, c, f\}$ yields

$$\begin{aligned} X_4 &= 0.1 + 0.2X_5 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.09); \\ X_5 &= 0.4 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 0.81); \\ X_6 &= 0.1 + 0.2X_4 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1). \end{aligned}$$

The seventh mixture component $\{a, d, f\}$ yields

$$\begin{aligned} X_4 &= 0.6 + 0.8X_5 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 0.36); \\ X_5 &= 0.4 + \varepsilon_{X_5}, & \varepsilon_{X_5} &\sim N(0, 1.44); \\ X_6 &= 0.1 + 0.2X_4 + \varepsilon_{X_6}, & \varepsilon_{X_6} &\sim N(0, 1). \end{aligned}$$

Reorganising these distributions by variables we obtain the local distributions of \mathcal{B} shown in Figure 3 top.

Example A3 (Entropy of a discrete BN). Consider again the discrete BN from Example A1. In this simple example, we can use its global distribution and (6) to compute

$$\begin{aligned} H(\mathcal{B}) &= -0.005 \log 0.005 - 0.013 \log 0.013 - 0.052 \log 0.052 - 0.050 \log 0.050 - \\ &\quad - 0.064 \log 0.064 - 0.040 \log 0.040 - 0.037 \log 0.037 - 0.026 \log 0.026 - \\ &\quad - 0.022 \log 0.022 - 0.051 \log 0.051 - 0.210 \log 0.210 - 0.199 \log 0.199 - \\ &\quad - 0.089 \log 0.089 - 0.056 \log 0.056 - 0.051 \log 0.051 - 0.036 \log 0.036 = 2.440. \end{aligned}$$

In the general case, we compute $H(\mathcal{B})$ from the local distributions using (9). Since X_1 and X_2 have no parents, their entropy components simply sum over their marginal distributions:

$$\begin{aligned} H(X_1) &= -0.53 \log 0.53 - 0.47 \log 0.47 = 0.691, \\ H(X_2) &= -0.34 \log 0.34 - 0.66 \log 0.66 = 0.641. \end{aligned}$$

For X_3 ,

$$H(X_3 \mid X_1, X_2) = \sum_{x_1 \in \{a, b\}} \sum_{x_2 \in \{c, d\}} P(X_1 = x_1, X_2 = x_2) H(X_3 \mid X_1 = x_1, X_2 = x_2)$$

where

$$\begin{aligned} H(X_3 \mid X_1 = a, X_2 = c) &= -0.15 \log 0.15 - 0.85 \log 0.85 = 0.423, \\ H(X_3 \mid X_1 = a, X_2 = d) &= -0.75 \log 0.75 - 0.25 \log 0.25 = 0.562, \\ H(X_3 \mid X_1 = b, X_2 = c) &= -0.40 \log 0.40 - 0.60 \log 0.60 = 0.673, \\ H(X_3 \mid X_1 = b, X_2 = d) &= -0.80 \log 0.80 - 0.20 \log 0.20 = 0.500; \end{aligned}$$

and where (multiplying the marginal probabilities for X_1 and X_2 , which are marginally independent)

$$\begin{aligned} P(X_1 = a, X_2 = c) &= 0.180, & P(X_1 = a, X_2 = d) &= 0.350, \\ P(X_1 = b, X_2 = c) &= 0.160, & P(X_1 = b, X_2 = d) &= 0.310; \end{aligned}$$

giving

$$H(X_3 \mid X_1, X_2) = (0.180 \cdot 0.423 + 0.350 \cdot 0.562 + 0.160 \cdot 0.673 + 0.310 \cdot 0.500) = 0.536.$$

Finally, for X_4

$$H(X_4 | X_3) = \sum_{x_3 \in \{e, f\}} P(X_3 = x_3) H(X_4 | X_3 = x_3)$$

where

$$H(X_4 | X_3 = e) = -0.20 \log 0.20 - 0.80 \log 0.80 = 0.500,$$

$$H(X_4 | X_3 = f) = -0.42 \log 0.42 - 0.58 \log 0.58 = 0.680;$$

and $P(X_3 = e) = 0.601$, $P(X_3 = f) = 0.399$, giving

$$H(X_4 | X_3) = 0.601 \cdot 0.500 + 0.399 \cdot 0.680 = 0.572.$$

Combining all these figures, we obtain $H(\mathcal{B})$ as

$$H(X_1) + H(X_2) + H(X_3 | X_1, X_2) + H(X_4 | X_3) = 0.691 + 0.641 + 0.536 + 0.572 = 2.440$$

as before.

In general, we would have to compute the probabilities of the parent configurations of each node using a junction tree as follows:

1. We construct the moral graph of \mathcal{B} , which contains the same arcs (but undirected) as its DAG plus $X_1 - X_2$.
2. We identify two cliques $C_1 = \{X_1, X_2, X_3\}$ and $C_2 = \{X_3, X_4\}$ and a separator $S_{12} = \{X_3\}$.
3. We connect them to create the junction tree $C_1 - S_{12} - C_2$.
4. We initialise the cliques with the respective distributions $P(C_1) = P(X_1, X_2, X_3)$, $P(C_2) = P(X_3, X_4)$ and $P(S_{12}) = P(X_3)$.
5. We compute $P(X_1, X_2) = \sum_{x_3 \in \{e, f\}} P(C_1)$ and $P(X_3) = P(S_{12})$.

Example A4 (Entropy of a GBN). Consider the GBN \mathcal{B} from Figure 1 top, whose global distribution we derived in Example 2. If we plug its covariance matrix $\Sigma_{\mathcal{B}}$ into the entropy formula for the multivariate normal distribution we obtain

$$H(\mathcal{B}) = \frac{4}{2} + \frac{4}{2} \log 2\pi + \frac{1}{2} \log \det(\Sigma_{\mathcal{B}}) = 2 + 3.676 + 0.5 \log 0.475 = 5.304.$$

Equivalently, plugging the $\sigma_{X_i}^2(\mathcal{B})$ into (12) we have

$$H(\mathcal{B}) = \sum_{i=1}^N H(X_i | \Pi_{X_i}^{\mathcal{B}}) = \frac{1}{2} [\log(2\pi \cdot 0.8) + \log(2\pi \cdot 0.6) + \log(2\pi \cdot 0.9) + \log(2\pi \cdot 1.1)] + \frac{4}{2} = 5.304.$$

Example A5 (KL between GBNs with parameters estimated from data). Consider the DAGs for the BNs \mathcal{B} and \mathcal{B}' and the 10 observations shown in Figure A1. The partial topological ordering of the nodes in \mathcal{B} is $\{\{X_1, X_2\}, X_4, X_3\}$ and that in \mathcal{B}' is $\{X_1, X_2, \{X_3, X_4\}\}$: the total ordering that is compatible with both is $\{X_1, X_2, X_4, X_3\}$.

If we estimate the parameters of the local distributions of \mathcal{B} by maximum likelihood we obtain

$$\begin{aligned} X_1 &= 2.889 + \varepsilon_{X_1}, & \varepsilon_{X_1} &\sim N(0, 0.558), \\ X_2 &= 1.673 + \varepsilon_{X_2}, & \varepsilon_{X_2} &\sim N(0, 1.595), \\ X_3 &= 0.896 + 1.299X_4 + \varepsilon_{X_3}, & \varepsilon_{X_3} &\sim N(0, 1.142), \\ X_4 &= -2.095 + 2.222X_1 + 2.613X_2 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 1.523), \end{aligned}$$

and the associated fitted values are

$$\begin{aligned}\hat{\mathbf{x}}_1(\mathcal{B}) &= (2.889, 2.889, 2.889, 2.889, 2.889, 2.889, 2.889, 2.889, 2.889, 2.889), \\ \hat{\mathbf{x}}_2(\mathcal{B}) &= (1.673, 1.673, 1.673, 1.673, 1.673, 1.673, 1.673, 1.673, 1.673, 1.673), \\ \hat{\mathbf{x}}_3(\mathcal{B}) &= (17.293, 14.480, 8.675, 13.937, 14.846, 12.801, 13.449, 2.394, 9.670, 14.381), \\ \hat{\mathbf{x}}_4(\mathcal{B}) &= (13.307, 11.447, 5.852, 8.635, 8.475, 9.018, 10.370, 2.376, 7.014, 10.489).\end{aligned}$$

Similarly, for \mathcal{B}' we obtain

$$\begin{aligned}X_1 &= 2.889 + \varepsilon_{X_1}, & \varepsilon_{X_1} &\sim N(0, 0.558), \\ X_2 &= 3.505 - 0.634X_1 + \varepsilon_{X_2}, & \varepsilon_{X_2} &\sim N(0, 1.542), \\ X_3 &= 7.284 + 2.933X_2 + \varepsilon_{X_3}, & \varepsilon_{X_3} &\sim N(0, 6.051), \\ X_4 &= 5.151 + 2.120X_2 + \varepsilon_{X_4}, & \varepsilon_{X_4} &\sim N(0, 3.999),\end{aligned}$$

and the associated fitted values are

$$\begin{aligned}\hat{\mathbf{x}}_1(\mathcal{B}) &= (2.889, 2.889, 2.889, 2.889, 2.889, 2.889, 2.889, 2.889, 2.889, 2.889), \\ \hat{\mathbf{x}}_2(\mathcal{B}) &= (1.207, 2.304, 1.778, 1.625, 1.754, 2.044, 1.127, 2.037, 0.840, 2.019), \\ \hat{\mathbf{x}}_3(\mathcal{B}) &= (15.529, 17.760, 9.408, 11.931, 12.261, 14.009, 11.918, 6.528, 7.019, 15.564), \\ \hat{\mathbf{x}}_4(\mathcal{B}) &= (11.110, 12.722, 6.686, 8.509, 8.748, 10.011, 8.500, 4.604, 4.959, 11.135).\end{aligned}$$

Therefore,

$$\begin{aligned}\|\hat{\mathbf{x}}_1(\mathcal{B}) - \hat{\mathbf{x}}_1(\mathcal{B}')\|_2^2 &= 0, & \|\hat{\mathbf{x}}_2(\mathcal{B}) - \hat{\mathbf{x}}_2(\mathcal{B}')\|_2^2 &= 2.018, \\ \|\hat{\mathbf{x}}_3(\mathcal{B}) - \hat{\mathbf{x}}_3(\mathcal{B}')\|_2^2 &= 54.434, & \|\hat{\mathbf{x}}_4(\mathcal{B}) - \hat{\mathbf{x}}_4(\mathcal{B}')\|_2^2 &= 21.329;\end{aligned}$$

and the values of the Kullback–Leibler divergence for the individual nodes are

$$\begin{aligned}\text{KL}(X_1 | \Pi_{X_1}^{\mathcal{B}} \| X_1 | \Pi_{X_1}^{\mathcal{B}'}) &\approx \frac{1}{2} \left(\log \frac{0.558}{0.558} + \frac{0.558}{0.558} - 1 \right) + \frac{1}{20} \left(\frac{0}{0.558} \right) = 0, \\ \text{KL}(X_2 | \Pi_{X_2}^{\mathcal{B}} \| X_2 | \Pi_{X_2}^{\mathcal{B}'}) &\approx \frac{1}{2} \left(\log \frac{1.542}{1.595} + \frac{1.595}{1.542} - 1 \right) + \frac{1}{20} \left(\frac{2.018}{1.542} \right) = 0.066, \\ \text{KL}(X_3 | \Pi_{X_3}^{\mathcal{B}} \| X_3 | \Pi_{X_3}^{\mathcal{B}'}) &\approx \frac{1}{2} \left(\log \frac{6.051}{1.142} + \frac{1.142}{6.051} - 1 \right) + \frac{1}{20} \left(\frac{54.434}{6.051} \right) = 0.878, \\ \text{KL}(X_4 | \Pi_{X_4}^{\mathcal{B}} \| X_4 | \Pi_{X_4}^{\mathcal{B}'}) &\approx \frac{1}{2} \left(\log \frac{3.999}{1.523} + \frac{1.523}{3.999} - 1 \right) + \frac{1}{20} \left(\frac{21.329}{3.999} \right) = 0.440,\end{aligned}$$

which sum up to $\text{KL}(\mathcal{B} \| \mathcal{B}') \approx 1.383$. The exact value, which we can compute as shown in Section 4.2, is 1.692.

The quality of the empirical approximation improves with the number of observations. For reference, we generated the data in Figure A1 from the GBN in Example 2. With a sample of size $n = 100$ from the same network, $\text{KL}(\mathcal{B} \| \mathcal{B}') \approx 1.362$ with $\text{KL}(\mathcal{B} \| \mathcal{B}') = 1.373$; with $n = 1000$, $\text{KL}(\mathcal{B} \| \mathcal{B}') \approx 1.343$ with $\text{KL}(\mathcal{B} \| \mathcal{B}') = 1.345$.

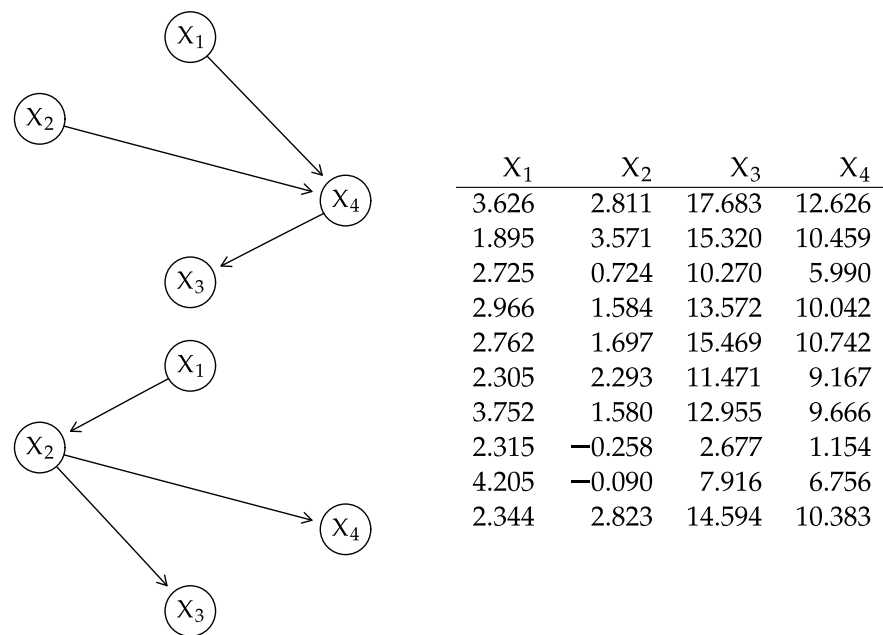


Figure A1. The DAGs for the GBNs \mathcal{B} (top left) and \mathcal{B}' (bottom left) and the data (right) used in Example A5.

Example A6 (Entropy of a CLGBN). Consider again the CLGBN \mathcal{B} from Figure 3 (top). For such a simple BN, we can use its global distribution (which we derived in Example A2) directly to compute the entropies of the multivariate normal distributions associated with the mixture components

$$H(X_4, X_5, X_6 \mid \{c, e\}) = \frac{3}{2} + \frac{3}{2} \log(2\pi) + \frac{1}{2} \log \det(\Sigma_{\{c,e\}}(\mathcal{B})) = 1.849,$$

$$H(X_4, X_5, X_6 \mid \{d, e\}) = \frac{3}{2} + \frac{3}{2} \log(2\pi) + \frac{1}{2} \log \det(\Sigma_{\{d,e\}}(\mathcal{B})) = 3.235,$$

$$H(X_4, X_5, X_6 \mid \{c, f\}) = \frac{3}{2} + \frac{3}{2} \log(2\pi) + \frac{1}{2} \log \det(\Sigma_{\{c,f\}}(\mathcal{B})) = 2.947,$$

$$H(X_4, X_5, X_6 \mid \{d, f\}) = \frac{3}{2} + \frac{3}{2} \log(2\pi) + \frac{1}{2} \log \det(\Sigma_{\{d,f\}}(\mathcal{B})) = 3.928;$$

and to combine them by weighting with the component probabilities

$$\begin{aligned} H(X_4, X_5, X_6 \mid X_1, X_2, X_3) &= \underbrace{0.040 \cdot 1.849}_{\{a,c,e\}} + \underbrace{0.036 \cdot 1.849}_{\{b,c,e\}} + \underbrace{0.040 \cdot 3.235}_{\{a,d,e\}} + \underbrace{0.084 \cdot 3.235}_{\{b,d,e\}} + \\ &\quad \underbrace{0.160 \cdot 2.947}_{\{a,c,f\}} + \underbrace{0.144 \cdot 2.947}_{\{b,c,f\}} + \underbrace{0.160 \cdot 3.928}_{\{a,d,f\}} + \underbrace{0.336 \cdot 3.928}_{\{b,d,f\}} = 3.386. \end{aligned}$$

The entropy of the discrete variables is

$$\begin{aligned} H(X_1, X_2, X_3) &= -0.040 \log 0.040 - 0.036 \log 0.036 - 0.040 \log 0.040 - 0.084 \log 0.084 - \\ &\quad 0.160 \log 0.160 - 0.144 \log 0.144 - 0.160 \log 0.160 - 0.336 \log 0.336 = 1.817 \end{aligned}$$

and then $H(\mathcal{B}) = H(X_1, X_2, X_3) + H(X_4, X_5, X_6 \mid X_1, X_2, X_3) = 5.203$.

If we use the local distributions instead, we can compute the entropy of the discrete variables using (9) from Section 4.1:

$$\begin{aligned} H(X_1) &= -0.4 \log 0.4 - 0.6 \log 0.6 = 0.673, \\ H(X_2 | X_1) &= 0.4(-0.5 \log 0.5 - 0.5 \log 0.5) + 0.6(-0.3 \log 0.3 - 0.7 \log 0.7) = 0.644, \\ H(X_3) &= -0.2 \log 0.2 - 0.8 \log 0.8 = 0.500. \end{aligned}$$

We can compute the entropy of the continuous variables with no discrete parents using (12) from Section 4.2:

$$H(X_6 | X_4) = \frac{1}{2} \log(2\pi \cdot 1) + \frac{1}{2} = 1.419.$$

Finally, we can compute the entropy of the continuous variables with discrete parents using (24) from Section 4.3:

$$\begin{aligned} H(X_4 | X_2, X_5) &= 0.38 \left(\frac{1}{2} \log(2\pi \cdot 0.09) + \frac{1}{2} \right) + 0.62 \left(\frac{1}{2} \log(2\pi \cdot 0.36) + \frac{1}{2} \right) \\ &= 0.645, \\ H(X_5 | X_2, X_3) &= 0.076 \left(\frac{1}{2} \log(2\pi \cdot 0.09) + \frac{1}{2} \right) + 0.124 \left(\frac{1}{2} \log(2\pi \cdot 0.36) + \frac{1}{2} \right) + \\ &\quad 0.304 \left(\frac{1}{2} \log(2\pi \cdot 0.81) + \frac{1}{2} \right) + 0.496 \left(\frac{1}{2} \log(2\pi \cdot 1.44) + \frac{1}{2} \right) \\ &= 1.322. \end{aligned}$$

As before, we confirm that overall

$$\begin{aligned} H(\mathcal{B}) &= H(X_1) + H(X_2 | X_1) + H(X_3) + H(X_4 | X_2, X_5) + H(X_5 | X_2, X_3) + \\ &\quad H(X_6 | X_4) = 0.673 + 0.644 + 0.500 + 0.645 + 1.322 + 1.419 = 5.203. \end{aligned}$$

References

- Scutari, M.; Denis, J.B. *Bayesian Networks with Examples in R*, 2nd ed.; Chapman & Hall: Boca Raton, FL, USA, 2021.
- Castillo, E.; Gutiérrez, J.M.; Hadi, A.S. *Expert Systems and Probabilistic Network Models*; Springer: Berlin/Heidelberg, Germany, 1997.
- Cowell, R.G.; Dawid, A.P.; Lauritzen, S.L.; Spiegelhalter, D.J. *Probabilistic Networks and Expert Systems*; Springer: Berlin/Heidelberg, Germany, 1999.
- Pearl, J. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*; Morgan Kaufmann: Burlington, MA, USA, 1988.
- Koller, D.; Friedman, N. *Probabilistic Graphical Models: Principles and Techniques*; MIT Press: Cambridge, MA, USA, 2009.
- Murphy, K.P. *Dynamic Bayesian Networks: Representation, Inference and Learning*. Ph.D. Thesis, Computer Science Division, UC Berkeley, Berkeley, CA, USA, 2002.
- Spirtes, P.; Glymour, C.; Scheines, R. *Causation, Prediction, and Search*; MIT Press: Cambridge, MA, USA, 2000.
- Pearl, J. *Causality: Models, Reasoning and Inference*, 2nd ed.; Cambridge University Press: Cambridge, UK, 2009.
- Borsboom, D.; Deserno, M.K.; Rhemtulla, M.; Epskamp, S.; Fried, E.I.; McNally, R.J.; Robinaugh, D.J.; Perugini, M.; Dalege, J.; Costantini, G.; et al. Network Analysis of Multivariate Data in Psychological Science. *Nat. Rev. Methods Prim.* **2021**, *1*, 58.
- Carapito, R.; Li, R.; Helms, J.; Carapito, C.; Gujja, S.; Rolli, V.; Guimaraes, R.; Malagon-Lopez, J.; Spinnhirny, P.; Lederle, A.; et al. Identification of Driver Genes for Critical Forms of COVID-19 in a Deeply Phenotyped Young Patient Cohort. *Sci. Transl. Med.* **2021**, *14*, 1–20.
- Requejo-Castro, D.; Giné-Garriga, R.; Pérez-Foguet, A. Data-driven Bayesian Network Modelling to Explore the Relationships Between SDG 6 and the 2030 Agenda. *Sci. Total Environ.* **2020**, *710*, 136014. [PubMed]
- Zilko, A.A.; Kurowicka, D.; Goverde, R.M.P. Modeling Railway Disruption Lengths with Copula Bayesian Networks. *Transp. Res. Part C Emerg. Technol.* **2016**, *68*, 350–368.
- Gao, R.X.; Wang, L.; Helu, M.; Teti, R. Big Data Analytics for Smart Factories of the Future. *CIRP Ann.* **2020**, *69*, 668–692.
- Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational Inference: A Review for Statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877.
- Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood From Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. (Ser. B)* **1977**, *39*, 1–22. [CrossRef]
- Minka, T.P. Expectation Propagation for Approximate Bayesian Inference. In Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence (UAI), Seattle, WA, USA, 2–5 August 2001; pp. 362–369.

17. van der Maaten, L.; Hinton, G. Visualizing Data Using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–3605.
18. Becht, E.; McInnes, L.; Healy, J.; Dutertre, C.A.; Kwok, I.W.H.; Ng, L.G.; Ginhoux, F.; Newell, E.W. Dimensionality Reduction for Visualizing Single-Cell Data Using UMAP. *Nat. Biotechnol.* **2019**, *37*, 38–44. [\[CrossRef\]](#)
19. Murphy, K.P. *Probabilistic Machine Learning: An Introduction*; MIT Press: Cambridge, MA, USA, 2022.
20. Murphy, K.P. *Probabilistic Machine Learning: Advanced Topics*; MIT Press: Cambridge, MA, USA, 2023.
21. Moral, S.; Cano, A.; Gómez-Olmedo, M. Computation of Kullback–Leibler Divergence in Bayesian Networks. *Entropy* **2021**, *23*, 1122. [\[CrossRef\]](#)
22. Hershey, J.R.; Olsen, P.A. Approximating the Kullback Leibler Divergence Between Gaussian Mixture Models. In Proceedings of the 32nd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Honolulu, HI, USA, 15–20 April 2007; Volume IV, pp. 317–320.
23. Beskos, A.; Crisan, D.; Jasra, A. On the Stability of Sequential Monte Carlo Methods in High Dimensions. *Ann. Appl. Probab.* **2014**, *24*, 1396–1445. [\[CrossRef\]](#)
24. Scutari, M. Learning Bayesian Networks with the bnlearn R Package. *J. Stat. Softw.* **2010**, *35*, 1–22. [\[CrossRef\]](#)
25. Heckerman, D.; Geiger, D.; Chickering, D.M. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Mach. Learn.* **1995**, *20*, 197–243. [\[CrossRef\]](#)
26. Chickering, D.M.; Heckerman, D. *Learning Bayesian Networks is NP-Hard*; Technical Report MSR-TR-94-17; Microsoft Corporation: Redmond, WA, USA, 1994.
27. Chickering, D.M. Learning Bayesian Networks is NP-Complete. In *Learning from Data: Artificial Intelligence and Statistics V*; Fisher, D.; Lenz, H., Eds.; Springer: Berlin/Heidelberg, Germany, 1996; pp. 121–130.
28. Chickering, D.M.; Heckerman, D.; Meek, C. Large-sample Learning of Bayesian Networks is NP-hard. *J. Mach. Learn. Res.* **2004**, *5*, 1287–1330.
29. Scutari, M.; Vitolo, C.; Tucker, A. Learning Bayesian Networks from Big Data with Greedy Search: Computational Complexity and Efficient Implementation. *Stat. Comput.* **2019**, *25*, 1095–1108. [\[CrossRef\]](#)
30. Cussens, J. Bayesian Network Learning with Cutting Planes. In Proceedings of the 27th Conference on Uncertainty in Artificial Intelligence (UAI), Barcelona, Spain, 14–17 July 2011; pp. 153–160.
31. Suzuki, J. An Efficient Bayesian Network Structure Learning Strategy. *New Gener. Comput.* **2017**, *35*, 105–124. [\[CrossRef\]](#)
32. Scanagatta, M.; de Campos, C.P.; Corani, G.; Zaffalon, M. Learning Bayesian Networks with Thousands of Variables. *Adv. Neural Inf. Process. Syst. (Nips)* **2015**, *28*, 1864–1872.
33. Hausser, J.; Strimmer, K. Entropy Inference and the James-Stein Estimator, with Application to Nonlinear Gene Association Networks. *J. Mach. Learn. Res.* **2009**, *10*, 1469–1484.
34. Agresti, A. *Categorical Data Analysis*, 3rd ed.; Wiley: Hoboken, NJ, USA, 2012.
35. Geiger, D.; Heckerman, D. Learning Gaussian Networks. In Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence (UAI), Seattle, WA, USA, 29–31 July 1994; pp. 235–243.
36. Pourahmadi, M. Covariance Estimation: The GLM and Regularization Perspectives. *Stat. Sci.* **2011**, *26*, 369–387.
37. Lauritzen, S.L.; Wermuth, N. Graphical Models for Associations between Variables, Some of which are Qualitative and Some Quantitative. *Ann. Stat.* **1989**, *17*, 31–57. [\[CrossRef\]](#)
38. Scutari, M.; Marquis, C.; Azzimonti, L. Using Mixed-Effect Models to Learn Bayesian Networks from Related Data Sets. In Proceedings of the International Conference on Probabilistic Graphical Models, Almería, Spain, 5–7 October 2022; Volume 186, pp. 73–84.
39. Lauritzen, S.L.; Spiegelhalter, D.J. Local Computation with Probabilities on Graphical Structures and their Application to Expert Systems (with discussion). *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **1988**, *50*, 157–224.
40. Lauritzen, S.L.; Jensen, F. Stable Local Computation with Conditional Gaussian Distributions. *Stat. Comput.* **2001**, *11*, 191–203. [\[CrossRef\]](#)
41. Cowell, R.G. Local Propagation in Conditional Gaussian Bayesian Networks. *J. Mach. Learn. Res.* **2005**, *6*, 1517–1550.
42. Namasivayam, V.K.; Pathak, A.; Prasanna, V.K. Scalable Parallel Implementation of Bayesian Network to Junction Tree Conversion for Exact Inference. In Proceedings of the 18th International Symposium on Computer Architecture and High Performance Computing, Ouro Preto, Brazil, 17–20 October 2006; pp. 167–176.
43. Pennock, D.M. Logarithmic Time Parallel Bayesian Inference. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI), Pittsburgh, PA, USA, 31 July–4 August 2023; pp. 431–438.
44. Namasivayam, V.K.; Prasanna, V.K. Scalable Parallel Implementation of Exact Inference in Bayesian Networks. In Proceedings of the 12th International Conference on Parallel and Distributed Systems (ICPADS), Minneapolis, MN, USA, 12–15 July 2006; pp. 1–8.
45. Malioutov, D.M.; Johnson, J.K.; Willsky, A.S. Walk-Sums and Belief Propagation in Gaussian Graphical Models. *J. Mach. Learn. Res.* **2006**, *7*, 2031–2064.
46. Cheng, J.; Druzdzel, M.J. AIS-BN: An Adaptive Importance Sampling Algorithm for Evidential Reasoning in Large Bayesian Networks. *J. Artif. Intell. Res.* **2000**, *13*, 155–188. [\[CrossRef\]](#)
47. Yuan, C.; Druzdzel, M.J. An Importance Sampling Algorithm Based on Evidence Pre-Propagation. In Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI), Acapulco, Mexico, 7–10 August 2003; pp. 624–631.
48. Cover, T.M.; Thomas, J.A. *Elements of Information Theory*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2006.

49. Csiszár, I.; Shields, P. *Information Theory and Statistics: A Tutorial*; Now Publishers Inc.: Delft, The Netherlands, 2004.
50. Gómez-Villegas, M.A.; Main, P.; Susi, R. Sensitivity of Gaussian Bayesian Networks to Inaccuracies in Their Parameters. In Proceedings of the 4th European Workshop on Probabilistic Graphical Models (PGM), Cuenca, Spain, 17–19 September 2008; pp. 265–272.
51. Gómez-Villegas, M.A.; Main, P.; Susi, R. The Effect of Block Parameter Perturbations in Gaussian Bayesian Networks: Sensitivity and Robustness. *Inf. Sci.* **2013**, *222*, 439–458. [[CrossRef](#)]
52. Gorgen, C.; Leonelli, M. Model-Preserving Sensitivity Analysis for Families of Gaussian Distributions. *J. Mach. Learn. Res.* **2020**, *21*, 1–32.
53. Seber, G.A.F. *A Matrix Handbook for Statisticians*; Wiley: Hoboken, NJ, USA, 2008.
54. Stewart, G.W. *Matrix Algorithms, Volume I: Basic Decompositions*; SIAM: Philadelphia, PA, USA, 1998.
55. Cavanaugh, J.E. Criteria for Linear Model Selection Based on Kullback's Symmetric Divergence. *Aust. N. Z. J. Stat.* **2004**, *46*, 197–323. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.