Using Mixed-Effects Models to Learn Bayesian Networks from Related Data Sets

Marco Scutari SCUTARI@BNLEARN.COM
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Lugano, Switzerland

Christopher Marquis CHRISTOPHER.MARQUIS@EPFL.CH
Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland

Laura Azzimonti LAURA.AZZIMONTI@IDSIA.CH
Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Lugano, Switzerland

Abstract

We commonly assume that data are a homogeneous set of observations when learning the structure of Bayesian networks. However, they often comprise different data sets that are related but not homogeneous because they have been collected in different ways or from different populations.

In our previous work (Azzimonti et al., 2021), we proposed a closed-form Bayesian Hierarchical Dirichlet score for discrete data that pools information across related data sets to learn a single encompassing network structure, while taking into account the differences in their probabilistic structures. In this paper, we provide an analogous solution for learning a Bayesian network from continuous data using mixed-effects models to pool information across the related data sets. We study its structural, parametric, predictive and classification accuracy and we show that it outperforms both conditional Gaussian Bayesian networks (that do not perform any pooling) and classical Gaussian Bayesian networks (that disregard the heterogeneous nature of the data). The improvement is marked for low sample sizes and for unbalanced data sets.

Keywords: Gaussian Bayesian networks; conditional Gaussian Bayesian networks; random effects models; structure learning.

1. Introduction

Bayesian networks (BNs; Koller and Friedman, 2009) are a graphical model defined over a set of random variables $\mathbf{X} = \{X_1, \ldots, X_N\}$, 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

$$\mathrm{P}(\mathbf{X} \,|\, \mathcal{G}, \Theta) = \prod\nolimits_{i=1}^{N} \mathrm{P}(X_i \,|\, \Pi_{X_i}, \Theta_{X_i}), \tag{1}$$

in which the joint probability distribution of $\mathbf{X}$ (with parameters Θ) decomposes in one local distribution for each X_i (with parameters Θ_{X_i}, $\bigcup_{i=1}^{N} \Theta_{X_i} = \Theta$) conditional on its parents Π_{X_i}. In this paper we will assume that $\mathbf{X}$ is a multivariate normal random variable and that the X_i are univariate normals linked by linear dependencies, thus focusing on the class of BNs known as *Gaussian BNs* (GBNs; Geiger and Heckerman, 1994). The parameters of their local distributions can be equivalently written as the partial correlations between X_i and each parent given the others or as the coefficients $\boldsymbol{\beta}_i$ of the linear regression model,

$$X_i = \mu_i + \mathbf{\Pi}_{X_i}\boldsymbol{\beta}_i + \varepsilon_i, \qquad \varepsilon_i \sim N(\mathbf{0}, \sigma_i^2 \mathbf{I}_n), \tag{2}$$

where $\mathbf{\Pi}_{X_i}$ is the design matrix associated to the parents of X_i, n is the sample size and $\mathbf{I}_n$ is an $n \times n$ identity matrix, so that $\Theta_{X_i} = \{\mu_i, \boldsymbol{\beta}_i, \sigma_i^2\}$. GBNs are a particular case of conditional Gaussian BNs (CGBNs; Lauritzen and Wermuth, 1989), which assume that $\mathbf{X}$ is a mixture of multivariate normals and thus can model discrete as well as continuous variables. Discrete X_i are only allowed to have discrete parents and are assumed to follow a multinomial distribution with parameters Θ_{X_i} organised in conditional probability tables. Continuous X_i are allowed to have both discrete and continuous parents (denoted Δ_{X_i} and Γ_{X_i} respectively, with $\Delta_{X_i} \cup \Gamma_{X_i} = \Pi_{X_i}$), and their local distributions are mixtures of linear regressions with one component for each configuration of Δ_{X_i}:

$$X_{ij} = \mu_{ij} + \mathbf{\Gamma}_{X_i}\boldsymbol{\beta}_{ij} + \varepsilon_{ij}, \qquad \varepsilon_{ij} \sim N(\mathbf{0}, \sigma_{ij}^2 \mathbf{I}_{n_j}),\ j = 1, \ldots, |\Delta_{X_i}|,\ \sum\nolimits_j n_j = n. \tag{3}$$

X_{ij} refers to X_i for the jth configuration of Δ_{X_i}, $\sigma_{ij}^2 \mathbf{I}_{n_j}$ is the associated $n_j \times n_j$ diagonal covariance matrix and $\mathbf{\Gamma}_{X_i}$ is the design matrix for the continuous parents Γ_{X_i}. Hence $\Theta_{X_i} = \bigcup_{j=1}^{|\Delta_{X_i}|} \{\mu_{ij}, \boldsymbol{\beta}_{ij}, \sigma_{ij}^2\}$. If X_i has no discrete parents, (3) simplifies to (2). In fact, CGBNs include GBNs and discrete BNs (Heckerman et al., 1995) as particular cases if $\mathbf{X}$ only contains continuous or discrete variables respectively.

Learning a BN from a data set $\mathcal{D}$ involves two steps: *structure learning* and *parameter learning*. Structure learning consists in finding the DAG $\mathcal{G}$ that encodes the dependence structure of the data and is usually performed by maximising $\mathrm{P}(\mathcal{G} \mid \mathcal{D})$ or some alternative goodness-of-fit measure (score-based learning). As an alternative, $\mathcal{G}$ can also be estimated by using conditional independence tests to identify which conditional independence relationships are supported by the data (constraint-based learning). In the case of GBNs, structure learning can be implemented either by finding the DAG that maximises the Bayesian Information Criterion (BIC; Schwarz, 1978)

$$\mathrm{BIC}(\mathcal{G}, \Theta \mid \mathcal{D}) = \sum\nolimits_{i=1}^{N} \log \mathrm{P}(X_i \mid \Pi_{X_i}, \Theta_{X_i}) - \frac{\log(n)}{2} |\Theta_{X_i}| \tag{4}$$

or $\mathrm{P}(\mathcal{G} \mid \mathcal{D})$, which is called the BGe score, or by testing partial correlations with Fisher's Z test or the exact Student t test. Both approaches, as well as the algorithms used to learn $\mathcal{G}$ from $\mathcal{D}$ are reviewed in Scutari et al. (2019). Parameter learning then consists in estimating the parameters Θ given the $\mathcal{G}$ obtained from structure learning, typically by means of maximum likelihood or Bayesian methods. The parameters of a GBN are usually estimated using the maximum likelihood estimates of regression coefficients from classical statistics.

All tests and scores in common use in the literature assume that observations are independent and identically distributed, which implies that $\mathcal{D}$ is homogeneous. This assumption, however, is easily violated when we pool together several small data sets, collected under similar but not identical conditions, in order to achieve a sample size large enough for relevant relationships to be detectable. Typical examples are data collected within multi-centre clinical trials (Spiegelhalter et al., 2004), in which differences in both protocols and patient populations across institutions should be properly modelled, and within ecology and environmental studies, in which different patterns of measurement errors and limitations in different environments should be taken into account (Qian et al., 2010). The task of efficiently modelling such related data sets is usually tackled by hierarchical models (Gelman et al., 2014), which pool the information common to the different subsets of the data while encoding the information that is specific to each subset. *Linear mixed-effects models* (Pinheiro and Bates, 2000) are classical statistical methods used to model correlated data and data with a multilevel or hierarchical structure, such as related data sets. In this paper, we will integrate them with GBNs to produce a new class of BNs that have a parametrisation similar to that of CGBNs but make better use of the information in the data. Under the assumption that each related data set has a GBN as a generative model, and that all GBNs have the same underlying network structure but different parameter values, we will show that it is possible to learn such BNs accurately from data by searching for high-scoring DAGs with hill-climbing and BIC. Our aim is to complement our previous work (Azzimonti et al., 2021) on structure learning from related data sets in discrete BNs.

This paper is structured as follow. Section 2 describes linear mixed-effects models, GBNs and CGBNs; our approach to integrating linear mixed-effects models with GBNs; and how they can be learned from related data sets. Section 3 showcases that our approach produces BNs that outperform GBNs, which disregard the heterogeneous nature of the data, and CGBNs, which do not perform any pooling, especially at small sample sizes and when the data are unbalanced. These results are summarised and discussed in Section 4.

2. Linear Mixed-Effects Models and Bayesian Networks

Assume that we have an observed variable F denoting which data set each observation belongs to and that all related datasets comprise the same continuous variables $\mathbf{X}$. Furthermore, the data must be complete and we assume that there are no latent variables.

On the one hand, we could model such data with a GBN defined over $\mathbf{X}$, disregarding their heterogeneous nature, and the local distributions of the nodes would take the form shown in (2). In this setting, we estimate the parameters of the GBNs for the related data sets with identical estimates with a *complete pooling* of their information. This is undesirable because F will act as a latent variable, possibly introducing spurious arcs in the BN and biasing the estimates of the $\boldsymbol{\beta}_i$. Moreover, the σ_i^2 will be artificially inflated because the corresponding ε_i are likely to be heteroscedastic, possibly decreasing the statistical significance of the $\boldsymbol{\beta}_i$ and introducing false negative arcs in structure learning.

On the other hand, we could model such data with a CGBN defined over $\mathbf{X} \cup F$, in which F is the only discrete node and a parent of all the other nodes. The local distributions of the nodes would take the form shown in (3) and would be correctly specified. However, each regression in the mixture would be estimated using only the data of the corresponding data set and there would be *no pooling* of information across the related data set. This may make both structure and parameter learning unreliable for sparse data and for related data sets with unbalanced sample sizes.

Partial pooling is a compromise between these two extremes that can outperform both. We implement it by defining the BN over $\mathbf{X} \cup F$, making F a parent of all the X_i and modelling local distributions by means of *linear mixed-effects* (LME; Pinheiro and Bates, 2000) models. LME models are hierarchical models that extend the classic linear regression model by adding a second set of coefficients, called "random effects" and usually denoted $\mathbf{b}_i$, which are jointly distributed as a multivariate normal. The other coefficients are called "fixed effects".

Extending the notation of (2), we can write the local distribution of X_i as

$$X_i = \mu_i + \boldsymbol{\Pi}_{X_i}\boldsymbol{\beta}_i + \mathbf{Z}_i\mathbf{b}_i + \varepsilon_i, \qquad \mathbf{b}_i \sim N(\mathbf{0}, \boldsymbol{\Sigma}_i),\ \varepsilon_i \sim N(\mathbf{0}, \sigma_i^2\mathbf{I}_n), \tag{5}$$

where $\boldsymbol{\Pi}_{X_i}$ is the design matrix associated to the nodes in $\mathbf{X}$ that are parents of X_i, $\boldsymbol{\beta}_i$ is the vector of fixed effects, $\mathbf{Z}_i$ is the design matrix of the random effects and $\mathbf{b}_i$ is the vector of random effects. For simplicity, we assume in the following that the sets of random effects associated with the $|F|$ related data sets are independent of each other and that we have a random effect for each

Π_{X_i} and for the intercept. As a result, $\mathbf{Z}_i$ is a block matrix of size size $n \times (|\Pi_{X_i}| + 1)|F|$, whose blocks are associated to different data sets (see Bates et al. (2015) for more details), and $\mathbf{b}_i$ is the $(|\Pi_{X_i}| + 1)|F| \times 1$ vector composed by random effects for different data sets.

The local distribution in (5) can also be written in the same form as the local distribution of continuous variable in a CGBN (3):

$$X_{ij} = (\mu_{ij} + b_{ij0}) + \mathbf{\Pi}_{X_i}(\boldsymbol{\beta}_i + \mathbf{b}_{ij}) + \varepsilon_{ij}, \quad \begin{pmatrix} b_{ij0} \\ \mathbf{b}_{ij} \end{pmatrix} \sim N(\mathbf{0}, \tilde{\mathbf{\Sigma}}_i), \; \varepsilon_{ij} \sim N(\mathbf{0}, \sigma_i^2 \mathbf{I}_{n_j}), \tag{6}$$

where $j = 1, \ldots, |F|$ are the related data sets as encoded in the node F which is the only discrete parent Δ_{X_i}; $\mathbf{\Pi}_{X_i}$ from (5) coincides with $\mathbf{\Gamma}_{X_i}$ from (3); b_{ij0} is the intercept term in the random effects and the $\mathbf{b}_{ij}$ are the coefficients of the random effects for the jth related data set; $\tilde{\mathbf{\Sigma}}_i$ is the $n_j \times n_j$ block of $\mathbf{\Sigma}_i$ associated to the jth related data set. The residual variances are here assumed to be homoscedastic and to be independent of the data set. This assumption can be easily relaxed to the heteroscedastic case where $\varepsilon_{ij} \sim N(\mathbf{0}, \sigma_{ij}^2 \mathbf{I}_{n_j})$.

From (6), we can see that $\mathbf{\Pi}_{X_i}\mathbf{b}_{ij}$ encodes the deviations of the regression coefficients for the related data sets from the shared $\mathbf{\Pi}_{X_i}\boldsymbol{\beta}_i$ values. Indeed, the coefficients $\mathbf{b}_i$ associated with the random effects have mean zero, and they naturally represent the deviations of the effects of the parents in individual data sets from their average effects across data sets, which is represented by the fixed effects $\boldsymbol{\beta}_i$. The common distribution of the random effects $\mathbf{b}_i$ in (5) produces both a pooling effect and a shrinkage of the parameters associated to individual data sets towards their common average. The magnitude of this effect is implicitly determined by the sample size of each related data set and by the effect size of each parent, in contrast with the distributional assumptions in Azzimonti et al. (2021) which contain an explicit shrinkage parameter. Moreover, the inclusion of random effects in (6) induces a decomposition of the variance of X_{ij} in two independent terms associated with $\mathbf{\Pi}_{X_i}\mathbf{b}_{ij}$ and ε_{ij}, allowing for different variances in different data sets even though the residuals ε_{ij} are homoscedastic.

We will perform parameter learning using standard statistical results for LMEs, as described in Demidenko (2009), while for structure learning we will use BIC to assess the goodness of fit and implement score-based learning. We will not consider constraint-based learning, which can be implemented using statistical tests for nested models to simultaneously test the significance of the fixed and random effects associated with a parent, following Pinheiro and Bates (2000) and Demidenko (2009). The expression for the BIC score arising from (5) has the same general form as (4), but the characterisation of number of parameters $|\Theta_{X_i}|$, where $\Theta_{X_i} = \{\boldsymbol{\beta}_i, \mathbf{b}_i, \mathbf{\Sigma}_i, \sigma_i^2\}$, is more complex. In a GBN, each local distribution has $|\Pi_{X_i}| + 2$ parameters; in a CGBN, $(|\Pi_{X_i}| + 2)|\Delta_{X_i}|$. In the proposed model we cannot simply add the number of random effects to the count: we must also estimate their covariance structure $\mathbf{\Sigma}_i$. Overall, the number of free parameters is

$$|\Theta_{X_i}| = \underbrace{|\Pi_{X_i}| + 1}_{\text{fixed effects}} + \underbrace{|\Pi_{X_i}| + 1 + \binom{|\Pi_{X_i}| + 1}{2}}_{\text{random effects}} + \underbrace{1}_{\text{residuals}} = \frac{|\Pi_{X_i}|^2 + 5|\Pi_{X_i}| + 6}{2}, \tag{7}$$

since $\mathbf{\Sigma}_i$ contains $|\Pi_{X_i}| + 1$ variances and $\binom{|\Pi_{X_i}|+1}{2}$ covariances, if there are $|\Pi_{X_i}| + 1$ random effects.

Although it is beyond the scope of this paper, note that we can leverage the similarities between (5) and (3) to improve the learning of general CGBNs by using generalised LMEs (Demidenko, 2009) and thus inducing information pooling and shrinking in their local distributions.

3. Simulation Study

We showcase the properties of this combination of LMEs and BNs with a simulation study. The experimental design is as follows:

1. For each of $N = 10, 20, 50$, for each of $\overline{|\Pi_{X_i}|} = 1, 2, 4$, and for each $|F| = 2, 5, 10, 20, 50$ we generate DAGs ensuring that all nodes are connected, and that there is an arc pointing from F to each node. Here $\overline{|\Pi_{X_i}|}$ denotes the average number of parents of each node X_i and represents the density of the DAG. In each DAG, Arcs are included independently of each other with probability $p = \overline{|\Pi_{X_i}|} \cdot 2/N$ to obtain $\overline{|\Pi_{X_i}|}$ arcs on average, and DAGs that are not connected are discarded.

2. For each DAG, we assign F a uniform distribution to simulate a balanced sample. For each of the continuous variables X_i and each related data set $j = 1, \ldots, |F|$, we sample the regression coefficients of (3) as

$$\boldsymbol{\beta}_{ij} \sim N(\boldsymbol{\beta}_i + \mathbf{b}_{ij}, \sigma^2_{\boldsymbol{\beta}_{ij}} \mathbf{I}_{|\Pi_{X_i}|+1}), \qquad \boldsymbol{\beta}_i = \mathbf{2},\ \mathbf{b}_{ij} \sim N(\mathbf{0}, \mathbf{I}_{|\Pi_{X_i}|+1}), \sigma^2_{\boldsymbol{\beta}_{ij}} \sim \chi^2_1$$

   and we set the standard error of the residuals σ^2_{ij} so that the Π_{X_i} explain 85% of the variance of X_i for each of the $|F|$ related data sets, thus ensuring local distributions are not singular.

We generate 5 BNs (denoted $\mathcal{B}_{\mathrm{TRUE}}$) for each configuration of N, $\overline{|\Pi_{X_i}|}$, $|F|$ and for each BN we generate different balanced and unbalanced data sets with different number of observations.

From each data set, we learn three different types of BNs using the implementation of hill-climbing and BIC in the **bnlearn** R package (Scutari, 2010):

1. *Complete pooling* ($\mathcal{B}_{\mathrm{GBN}}$): we learn a GBN from $\mathbf{X}$, completely disregarding F.

2. *No pooling* ($\mathcal{B}_{\mathrm{CGBN}}$): we learn a CGBN from $\{\mathbf{X}, F\}$ following (3).

3. *Partial pooling* ($\mathcal{B}_{\mathrm{LME}}$): we learn a CGBN from $\{\mathbf{X}, F\}$ following (5).

We assess the structural accuracy of the learned DAGs with the Structural Hamming distance (SHD; Tsamardinos et al., 2006) between the network structure of $\mathcal{B}_{\mathrm{TRUE}}$ and each of the structures of the $\mathcal{B}_{\mathrm{GBN}}$, $\mathcal{B}_{\mathrm{CGBN}}$, $\mathcal{B}_{\mathrm{LME}}$ learned from the data generated from $\mathcal{B}_{\mathrm{TRUE}}$. Similarly, we asses the parametric accuracy of the BNs by computing the Kullback-Leibler distance (KL; Kullback, 1959) between $\mathcal{B}_{\mathrm{TRUE}}$ (with the true parameter values) and $\mathcal{B}_{\mathrm{GBN}}$, $\mathcal{B}_{\mathrm{CGBN}}$, $\mathcal{B}_{\mathrm{LME}}$ (where parameters are estimated from data). For brevity, we will only discuss how these metrics compare for $\mathcal{B}_{\mathrm{CGBN}}$ and $\mathcal{B}_{\mathrm{LME}}$: the $\mathcal{B}_{\mathrm{LME}}$ have lower SHD values than the $\mathcal{B}_{\mathrm{GBN}}$ for 94% of the data sets in our simulation and lower KL for 96%. Even in the most adverse combination of experimental settings ($N = 10$, $\overline{|\Pi_{X_i}|} = 4$), the $\mathcal{B}_{\mathrm{LME}}$ have lower SHD for 65% of the data sets and lower KL for 93%. With such a stark difference in performance, it is clear that $\mathcal{B}_{\mathrm{LME}}$ should be always be preferred to $\mathcal{B}_{\mathrm{GBN}}$.

3.1 Balanced Data Sets

Firstly, we generate 5 data sets with $n_j = 10, 20, 50, 100$ observations in each related data set from each generated BN. Figure 1 shows the difference between the $\mathcal{B}_{\mathrm{LME}}$ and the $\mathcal{B}_{\mathrm{CGBN}}$ in terms of SHD and KL. The $\mathcal{B}_{\mathrm{LME}}$ increasingly outperform the $\mathcal{B}_{\mathrm{CGBN}}$ in terms of KL as the number of nodes increases (top left), as the density of the networks increases (top right) and as the number of

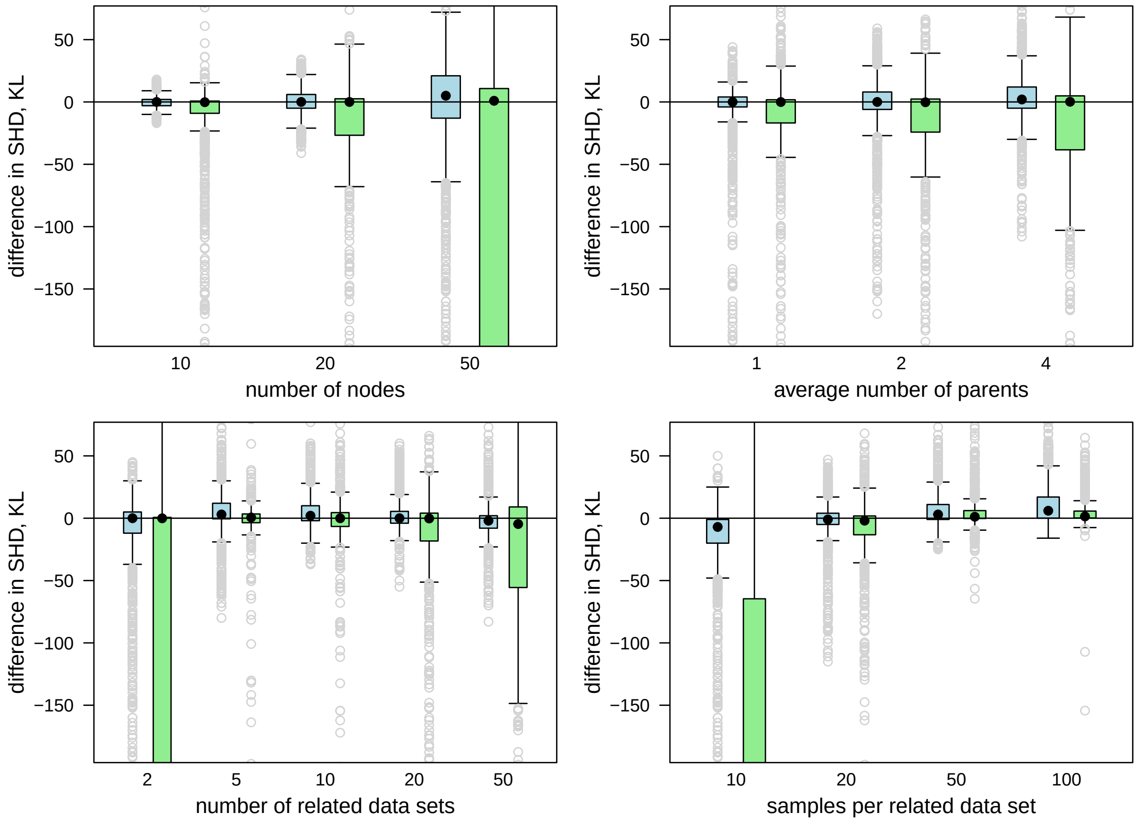

Figure 1: $\text{SHD}(\mathcal{B}_{\text{LME}}) - \text{SHD}(\mathcal{B}_{\text{CGBN}})$ (blue) and $\text{KL}(\mathcal{B}_{\text{TRUE}}, \mathcal{B}_{\text{LME}}) - \text{KL}(\mathcal{B}_{\text{TRUE}}, \mathcal{B}_{\text{CGBN}})$ (green) for all values of N (top left), $\overline{|\Pi_{X_i}|}$ (top right), $|F|$ (bottom left) and n_j (bottom right). Negative values favour $\mathcal{B}_{\text{LME}}$. The range of the KL differences is such that it is impossible to show the whole extent of some boxplots while keeping the others visible.

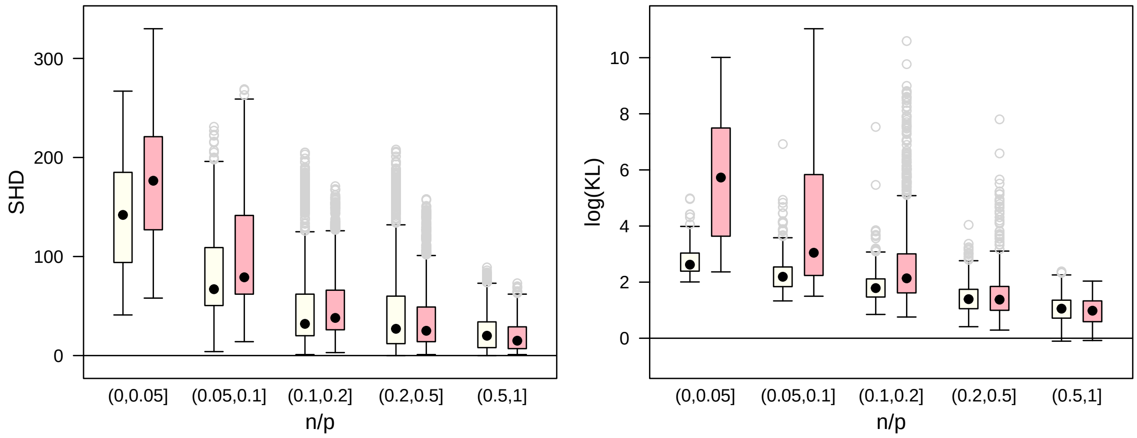

Figure 2: SHD (left) and KL (right) for the $\mathcal{B}_{\text{LME}}$ (ivory) and the $\mathcal{B}_{\text{CGBN}}$ (pink).

related data sets grows from 5 to 50 (bottom left). Unexpectedly, the $\mathcal{B}_{\mathrm{LME}}$ markedly outperform the $\mathcal{B}_{\mathrm{CGBN}}$ when $|F| = 2$ even though mixed-effects models are known to provide better performance for large $|F|$ (Pinheiro and Bates, 2000). The two approaches appear to be tied in terms of SHD, and to have similar rates of false negative arcs (that are in $\mathcal{B}_{\mathrm{TRUE}}$ but not in $\mathcal{B}_{\mathrm{CGBN}}$ or $\mathcal{B}_{\mathrm{LME}}$) and of false positive arcs (that are in $\mathcal{B}_{\mathrm{CGBN}}$ or $\mathcal{B}_{\mathrm{LME}}$ but not in $\mathcal{B}_{\mathrm{TRUE}}$). However, we can see in the bottom-right panel that for $n_j = 10$ the $\mathcal{B}_{\mathrm{LME}}$ outperform the $\mathcal{B}_{\mathrm{CGBN}}$ also in terms of SHD. As n_j increases, both the $\mathcal{B}_{\mathrm{LME}}$ and the $\mathcal{B}_{\mathrm{CGBN}}$ gradually converge to their large sample behaviour and learn the generating BN well, with only minor differences in both SHD and KL.

This leads us to investigate the relative sample efficiency of the $\mathcal{B}_{\mathrm{LME}}$ and the $\mathcal{B}_{\mathrm{CGBN}}$ in Figure 2. Plotting both SHD (left panel) and KL (right panel) against the number of samples per parameter in $\mathcal{B}_{\mathrm{TRUE}}$ (denoted n/p with $n = |F|n_j$ and $p = |\Theta_{\mathcal{B}_{\mathrm{TRUE}}}|$), we can see that the $\mathcal{B}_{\mathrm{LME}}$ have lower SHD and lower KL than the $\mathcal{B}_{\mathrm{CGBN}}$ for ratios up to 0.2. In particular, the difference in KL can easily be an order of magnitude in that range. We attribute this dominance to the pooling effect, which can result in better network scoring and better parameter estimation at lower sample sizes when sharing information has the most impact. When the the number of samples per parameter increases above 0.5 the two approaches are roughly equivalent, because each related data set contains enough information to score networks and estimate parameters appropriately and pooling has therefore a smaller effect. Notably the $\mathcal{B}_{\mathrm{CGBN}}$ do not outperform the $\mathcal{B}_{\mathrm{LME}}$ even in this case.

3.2 Predictive and Classification Accuracy

We also evaluate the $\mathcal{B}_{\mathrm{LME}}$ and the $\mathcal{B}_{\mathrm{CGBN}}$ on their:

- *Predictive accuracy:* predicting each X_i from other variables in $\mathbf{X}$, with and without knowledge of F. Measured with the average over the X_i of the relative errors between observed and predicted values (here denoted x_{ik} and $\widehat{x}_{ik}$ for node X_i and observation k) in absolute value:

$$\mathrm{RMAD} = \frac{1}{N}\sum_{i=1}^{N}\left[\frac{1}{n}\sum_{k=1}^{n}\left|\frac{x_{ik} - \widehat{x}_{ik}}{x_{ik}}\right|\right]$$

- *Classification accuracy:* predicting F from $\mathbf{X}$. Measured with the F_1 score (the harmonic average between precision and recall). Following Kuhn and Johnson (2013), when $|F|$ is greater than 2 we compute the F_1 score as the average of the one-vs-rest F_1 scores for each related data set.

To evaluate both measures of accuracy, we generate an additional sample of 1000 observations from $\mathcal{B}_{\mathrm{TRUE}}$ for each BN in our experimental design. Predicted values are computed as the posterior expectations for the relevant variables using likelihood-weighting approximate inference (Koller and Friedman, 2009). We show both performance measures in Figure 3. From the top-left plot, we can see that the $\mathcal{B}_{\mathrm{LME}}$ always have larger F_1 values than the $\mathcal{B}_{\mathrm{CGBN}}$, and that they have only negligible departures from $\mathrm{F}_1 = 1$ up to $|F| = 20$. We can also see that the $\mathcal{B}_{\mathrm{LME}}$ dominate the $\mathcal{B}_{\mathrm{CGBN}}$ for all n/p, and that the $\mathcal{B}_{\mathrm{CGBN}}$ only approach the $\mathcal{B}_{\mathrm{LME}}$ in performance as n approaches p top right panel). Similarly, the difference in predictive accuracy between the $\mathcal{B}_{\mathrm{LME}}$ and the $\mathcal{B}_{\mathrm{CGBN}}$ is always in favour of the former until $n/p < 0.2$ for both known and unknown F (bottom right panel), and it is not markedly different from zero for all the considered $|F|$ (bottom left panel).

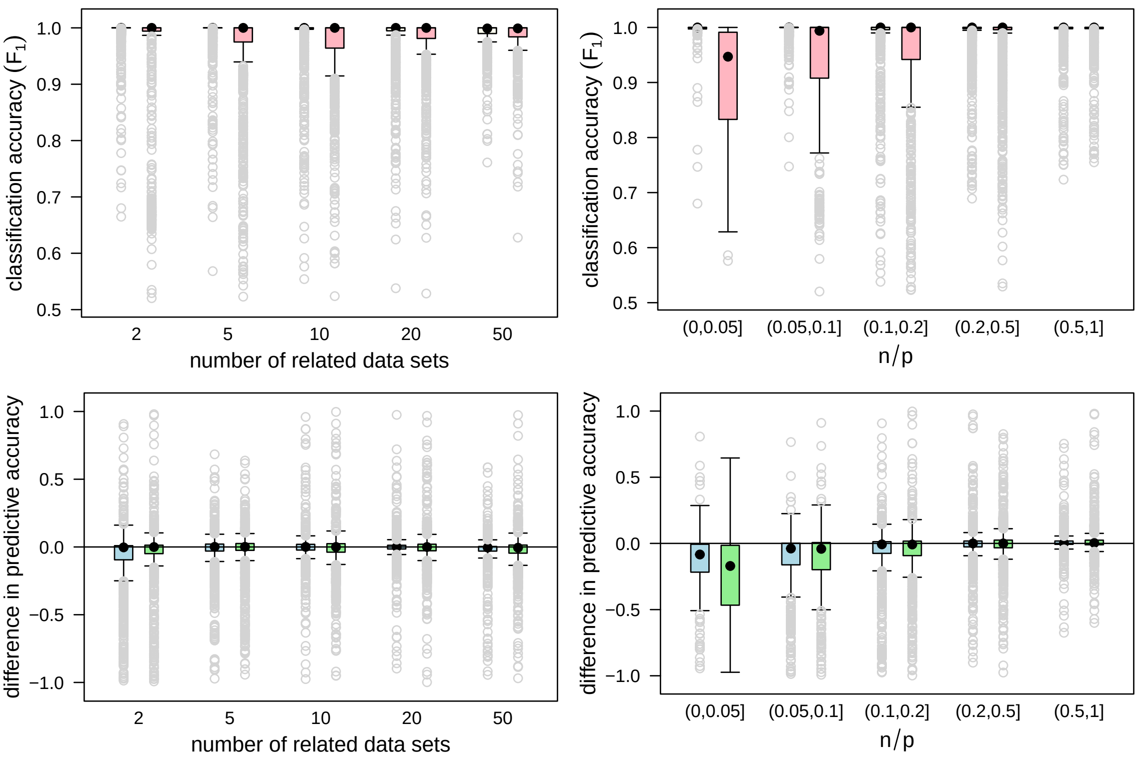

Figure 3: Classification accuracy against $|F|$ (top left) and n/p (top right) for the $\mathcal{B}_{\mathrm{LME}}$ (ivory) and the $\mathcal{B}_{\mathrm{CGBN}}$ (pink) for balanced data sets. Difference in predictive accuracy between the $\mathcal{B}_{\mathrm{LME}}$ and the $\mathcal{B}_{\mathrm{CGBN}}$ against $|F|$ (bottom left) and n/p (bottom right) knowing F (blue) and not knowing F (green); negative values favour $\mathcal{B}_{\mathrm{LME}}$.

3.3 Unbalanced Data Sets

Another scenario in which pooling can make a difference is when the sample is unbalanced, that is, when some of the related data sets contain markedly fewer observations that others. To evaluate the effect of the lack of balance, we modify our simulation experiments as follows:

1. we consider only those $\mathcal{B}_{\mathrm{TRUE}}$ with $|F| = 5, 10, 20$;
2. for each of them, we set the overall sample size to $n = |F|n_j$, with $n_j = 10, 20, 50, 100$;
3. and we assign $0.3n$ observations to each of two related data sets while distributing the remaining $0.4n$ evenly among the others.

The resulting $\mathcal{B}_{\mathrm{LME}}$ and $\mathcal{B}_{\mathrm{CGBN}}$ behave largely as with balanced data sets, but with $\mathcal{B}_{\mathrm{LME}}$ increasingly outperforming $\mathcal{B}_{\mathrm{CGBN}}$ as $|F|$ increases (and, as a result, the data sets become more and more unbalanced). For instance, the proportion of the simulations in which $\mathcal{B}_{\mathrm{LME}}$ outperforms $\mathcal{B}_{\mathrm{CGBN}}$ in terms of KL changes from 52% (balanced data sets) to 60% (unbalanced data sets) when

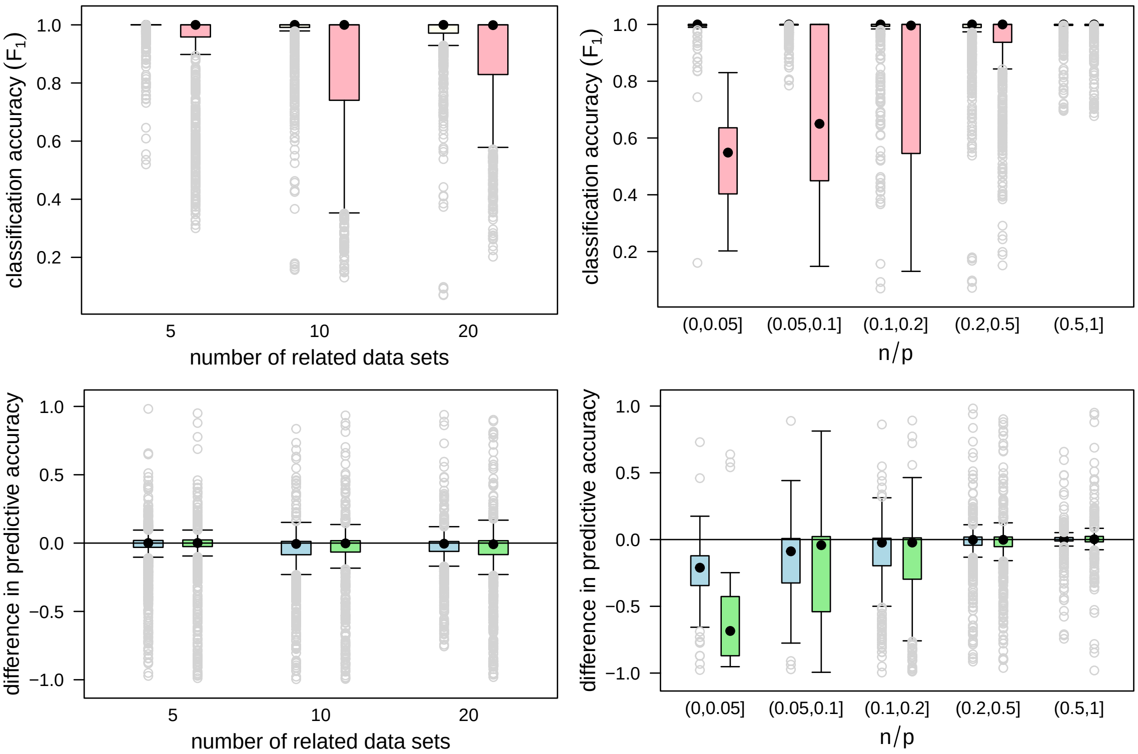

Figure 4: Classification accuracy against $|F|$ (top left) and n/p (top right) for the $\mathcal{B}_{\text{LME}}$ (ivory) and the $\mathcal{B}_{\text{CGBN}}$ (pink) for unbalanced data sets. Difference in predictive accuracy between the $\mathcal{B}_{\text{LME}}$ and the $\mathcal{B}_{\text{CGBN}}$ against $|F|$ (bottom left) and n/p (bottom right) knowing F (blue) and not knowing F (green); negative values favour $\mathcal{B}_{\text{LME}}$.

$|F| = 20$. Similarly, $\mathcal{B}_{\text{LME}}$ increasingly outperforms $\mathcal{B}_{\text{CGBN}}$ as n/p decreases, with $\mathcal{B}_{\text{LME}}$ outperforming $\mathcal{B}_{\text{CGBN}}$ in 78% of the simulations (unbalanced data set) instead of 75% (balanced data sets) in which $n/p < 0.2$. Plotting the KL and SHD of $\mathcal{B}_{\text{LME}}$ and $\mathcal{B}_{\text{CGBN}}$ side by side produces boxplots that are very similar to those in Figure 2 both marginally and conditionally on the individual values of $|F|$ (figures not shown for brevity). The same is true for predictive and classification accuracies, which we show in Figure 4. The difference in classification accuracy is increasingly in favour of $\mathcal{B}_{\text{LME}}$ as $|F|$ increases (top left) and as n/p decreases (top right). On the other hand, predictive accuracy is similar for all $|F|$ (bottom left), but is increasingly in favour of $\mathcal{B}_{\text{LME}}$ as n/p decreases for both known and unknown F (bottom right). The improvement in both the predictive and classification accuracy of $\mathcal{B}_{\text{LME}}$ is more marked in the unbalanced case compared to the balanced one.

3.4 Homogeneous Data Sets

Finally, we compare $\mathcal{B}_{\text{LME}}$, $\mathcal{B}_{\text{GBN}}$ and $\mathcal{B}_{\text{CGBN}}$ when the data are homogeneous, that is, when the data do not comprise multiple related data sets with different distributions. For this purpose, we

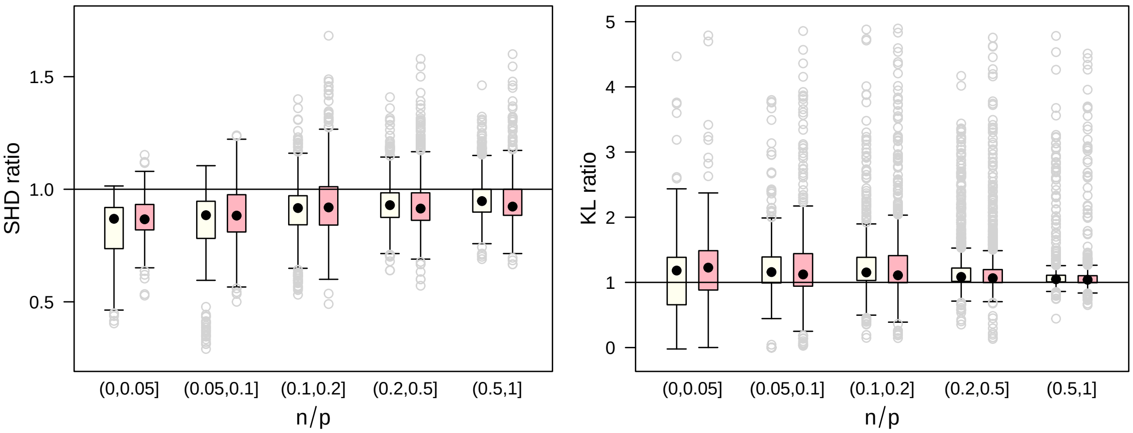

Figure 5: Left panel: $\mathrm{SHD}(\mathcal{B}_{\mathrm{LME}})/\,\mathrm{SHD}(\mathcal{B}_{\mathrm{GBN}})$ (ivory) and $\mathrm{SHD}(\mathcal{B}_{\mathrm{CGBN}})/\,\mathrm{SHD}(\mathcal{B}_{\mathrm{GBN}})$ (pink). Right panel: $\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{LME}})/\,\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{GBN}})$ (ivory) and $\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{CGBN}})/\,\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{GBN}})$ (pink). Both are computed from homogeneous data and shown as a function of n/p. Values larger than 1 favour $\mathcal{B}_{\mathrm{GBN}}$.

take the $\mathcal{B}_{\mathrm{TRUE}}$ we generated earlier and we modify them so that

$$\boldsymbol{\beta}_{ij} = \boldsymbol{\beta}_{i1} \qquad \text{and} \qquad \sigma^2_{ij} = \sigma^2_{i1} \qquad \text{for all } i = 1, \ldots, N \text{ and } j = 1, \ldots, |F|.$$

We then proceed to resample 5 balanced data sets with $n_j = 10, 20, 50, 100$ as in the first simulation setup (balanced data sets). Note that $\mathcal{B}_{\mathrm{TRUE}}$ still contains a node F identifying $|F| = 2, 5, 10, 20, 50$ related data sets, so the generated data will contain a set of labels that are effectively non-informative of the distribution of $\mathbf{X}$. In this scenario, $\mathcal{B}_{\mathrm{GBN}}$ is the correctly specified model while both $\mathcal{B}_{\mathrm{LME}}$ and $\mathcal{B}_{\mathrm{CGBN}}$ are over-parametrised. For this reason we will compare $\mathcal{B}_{\mathrm{LME}}$ and $\mathcal{B}_{\mathrm{CGBN}}$ relative to $\mathcal{B}_{\mathrm{GBN}}$. Relevant results from this modified simulation are shown in Figure 5.

In terms of SHD, $\mathcal{B}_{\mathrm{LME}}$ is roughly equivalent to $\mathcal{B}_{\mathrm{CGBN}}$ (better 35% of the time, equal 16%, worse 47%): computing $\mathrm{SHD}(\mathcal{B}_{\mathrm{LME}})/\,\mathrm{SHD}(\mathcal{B}_{\mathrm{CGBN}})$ we find that the median ratio is equal to 1 and the interquartile range is $[0.95, 1.05]$, that is, 50% of the simulations fall in this interval (figures are not shown for brevity). Furthermore, $\mathcal{B}_{\mathrm{LME}}$ is better than $\mathcal{B}_{\mathrm{GBN}}$ 75% of the time but the improvement is modest: the ratio $\mathrm{SHD}(\mathcal{B}_{\mathrm{LME}})/\,\mathrm{SHD}(\mathcal{B}_{\mathrm{GBN}})$ has median 0.93 and interquartile range $[0.87, 1]$. We attribute this effect to the implicit regularisation provided by LMEs over classical regressions model as a result of shrinking the coefficients $\mathbf{b}_{ij}$.

In terms of KL, we obtain as expected that $\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{GBN}})$ is smaller than $\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{LME}})$ for 78% of the simulations and smaller than $\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{CGBN}})$ for 75% of the simulations. The difference in performance, however, is small: $\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{LME}})/\,\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{GBN}})$ has median 1.07 and interquartile range $[1, 1.23]$, and $\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{CGBN}})/\,\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{GBN}})$ has median 1.07 and interquartile range $[1, 1.35]$. This suggests that even though $\mathcal{B}_{\mathrm{CGBN}}$ is more over-parametrised than $\mathcal{B}_{\mathrm{LME}}$, the impact of over-parametrisation is modest. And in fact, we find that $\mathcal{B}_{\mathrm{LME}}$ and $\mathcal{B}_{\mathrm{CGBN}}$ are very close in performance: the ratio $\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{LME}})/\,\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{CGBN}})$ has median 1.001 and interquartile range $[0.923, 1.027]$ (figures not shown for brevity).

If we restrict ourselves to simulations for which $n/p < 0.1$, the difference between $\mathcal{B}_{\mathrm{LME}}$ and $\mathcal{B}_{\mathrm{CGBN}}$ becomes more marked in terms of KL ($\mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{LME}}) / \mathrm{KL}(\mathcal{B}_{\mathrm{TRUE}}, \mathcal{B}_{\mathrm{CGBN}})$ decreases in median from 1.001 to 0.731 and the interquartile range shifts to $[0.003, 1.047]$), while it remains comparable over the complete set of simulations in terms of SHD ($\mathrm{SHD}(\mathcal{B}_{\mathrm{LME}}) / \mathrm{SHD}(\mathcal{B}_{\mathrm{CGBN}})$ has median 1.01 and interquartile range $[0.88, 1.07]$). Hence we conclude that the $\mathcal{B}_{\mathrm{LME}}$ fit homogeneous data better than the $\mathcal{B}_{\mathrm{CGBN}}$ for small sample sizes. The difference between the two becomes smaller as the sample size grows and the pooling effect is reduced, as expected from the theoretical foundations of LMEs in Section 2.

As for predictive accuracy, $\mathcal{B}_{\mathrm{LME}}$ has lower error rates than $\mathcal{B}_{\mathrm{CGBN}}$ for 60% of the simulations but the difference between the two is small both for the whole set of simulations and when we restrict ourselves to simulation for which $n/p < 0.1$ (figures not shown for brevity). We attribute this effect to the implicit regularisation introduced by LMEs. The ratio $\mathrm{RMAD}(\mathcal{B}_{\mathrm{LME}}) / \mathrm{RMAD}(\mathcal{B}_{\mathrm{CGBN}})$ has median 1.03 in the former case and 1.1 in the latter, although the interquartile range increases in size from $[0.942, 1.306]$ to $[0.818, 2.190]$ suggesting more variability for small sample sizes. The comparison of $\mathcal{B}_{\mathrm{GBN}}$ with respect to $\mathcal{B}_{\mathrm{LME}}$ and $\mathcal{B}_{\mathrm{CGBN}}$ shows that $\mathcal{B}_{\mathrm{GBN}}$ performs at least as well as $\mathcal{B}_{\mathrm{LME}}$ and $\mathcal{B}_{\mathrm{CGBN}}$ in almost all simulations (figures not shown for brevity). Moreover, all the results in terms of predictive accuracy are equivalent when F is known and unknown, since $\mathbf{X}$ is now independent from F by construction. For the same reason, classification accuracy is not relevant in this simulation scenario.

4. Conclusions

In this paper we revisited the problem of learning the structure of a BN from related data sets. In our previous work (Azzimonti et al., 2021) we focused on discrete BNs. Here we consider GBNs instead: we propose to use LMEs to produce a CGBN in which local distributions are replaced with LMEs to separate the effects that are specific to individual data sets (the random effects) from those that are common to all of them (the fixed effects).

LMEs are well-established models in the literature, and their favourable properties are advantageous in BN learning. In particular, the automatic pooling of information between the related data sets results in BNs with better structural and parametric accuracy for small sample sizes and unbalanced data sets (that is, when some of the related data sets are markedly smaller than others). Our experimental evaluation suggests that our approach always outperforms GBNs because they disregard the heterogeneity in the data (complete pooling). Our approach is also at least as good as using a standard CGBN (no pooling) in terms of structural accuracy, and outperforms it in terms of parametric accuracy in most simulations. Even when the data are not heterogeneous, our approach has comparable performance despite being over-parametrised compared to a correctly specified (GBN) model.

Furthermore, our approach can be extended to model related data sets containing both discrete and continuous variables by using generalised LMEs. The performance observed in this paper is encouraging in this respect.

References

L. Azzimonti, G. Corani, and M. Scutari. A Bayesian Hierarchical Score for Structure Learning from Related Data Sets. *International Journal of Approximate Reasoning*, 142:248–265, 2021.

D. M. Bates, M. Mächler, B. Bolker, and S. Walker. Fitting Linear Mixed-Effects Models Using lme4. *Journal of Statistical Software*, 67(1):1–48, 2015.

E. Demidenko. *Mixed Models: Theory and Applications with R*. Wiley, 2nd edition, 2009.

D. Geiger and D. Heckerman. Learning Gaussian Networks. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 235–243, 1994.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*. CRC press, 3rd edition, 2014.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian Networks: The Combination of Knowledge and Statistical Data. *Machine Learning*, 20(3):197–243, 1995.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

M. Kuhn and K. Johnson. *Applied Predictive Modeling*. Springer, 2013.

S. Kullback. *Information Theory and Statistics*. Wiley, 1959.

S. L. Lauritzen and N. Wermuth. Graphical Models for Associations Between Variables, Some of which are Qualitative and Some Quantitative. *The Annals of Statistics*, 17(1):31–57, 1989.

J. C. Pinheiro and D. M. Bates. *Mixed-effects models in S and S-PLUS*. Springer, 2000.

S. S. Qian, T. F. Cuffney, I. Alameddine, G. McMahon, and K. H. Reckhow. On the Application of Multilevel Modeling in Environmental and Ecological studies. *Ecology*, 91(2):355–361, 2010.

G. Schwarz. Estimating the Dimension of a Model. *The Annals of Statistics*, 6(2):461–464, 1978.

M. Scutari. Learning Bayesian Networks with the bnlearn R Package. *Journal of Statistical Software*, 35(3):1–22, 2010.

M. Scutari, C. E. Graafland, and J. M. Gutiérrez. Who Learns Better Bayesian Network Structures: Accuracy and Speed of Structure Learning Algorithms. *International Journal of Approximate Reasoning*, 115:235–253, 2019.

D. J. Spiegelhalter, K. R. Abrams, and J. P. Myles. *Bayesian Approaches to Clinical Trials and Health-Care Evaluation*. Wiley, 2004.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The Max-Min Hill-Climbing Bayesian Network Structure Learning Algorithm. *Machine Learning*, 65(1):31–78, 2006.