

Learning Bayesian Networks from Ordinal Data*

Xiang Ge Luo

D-BSSE, ETH Zurich, Mattenstrasse 26, 4058 Basel, Switzerland

XIANGGE.LUO@BSSE.ETHZ.CH

Giusi Moffa

Department of Mathematics and Computer Science, University of Basel, Basel, Switzerland

Division of Psychiatry, University College London, London, UK

Jack Kuipers

D-BSSE, ETH Zurich, Mattenstrasse 26, 4058 Basel, Switzerland

JACK.KUIPERS@BSSE.ETHZ.CH

Editor: Gal Elidan

Abstract

Bayesian networks are a powerful framework for studying the dependency structure of variables in a complex system. The problem of learning Bayesian networks is tightly associated with the given data type. Ordinal data, such as stages of cancer, rating scale survey questions, and letter grades for exams, are ubiquitous in applied research. However, existing solutions are mainly for continuous and nominal data. In this work, we propose an iterative score-and-search method - called the Ordinal Structural EM (OSEM) algorithm - for learning Bayesian networks from ordinal data. Unlike traditional approaches designed for nominal data, we explicitly respect the ordering amongst the categories. More precisely, we assume that the ordinal variables originate from marginally discretizing a set of Gaussian variables, whose structural dependence in the latent space follows a directed acyclic graph. Then, we adopt the Structural EM algorithm and derive closed-form scoring functions for efficient graph searching. Through simulation studies, we illustrate the superior performance of the OSEM algorithm compared to the alternatives and analyze various factors that may influence the learning accuracy. Finally, we demonstrate the practicality of our method with a real-world application on psychological survey data from 408 patients with co-morbid symptoms of obsessive-compulsive disorder and depression.

Keywords: Bayesian Networks, Ordinal Data, Structural EM Algorithm, Structure Learning.

1. Introduction

Many problems in applied statistics involve characterizing the relationships amongst a set of random variables in a complex system, aiming to describe high dimensional joint probability distributions, which may be of use in prediction and causal inference. Probabilistic graphical models, which incorporate graphical structures into probabilistic reasoning, are popular and powerful frameworks for analyzing these complex systems. The idea is to factorize the joint probability distribution p for the variables $\mathbf{X} = (X_1, \dots, X_n)^\top$ with respect to a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} is the set of nodes representing the variables, and \mathcal{E} the edges encoding a set of independence relationships (Lauritzen, 1996; Koller and Friedman, 2009).

*. **R** code is available at <https://github.com/xgluo/OSEM>

In this work, we focus on Bayesian networks - a special family of probabilistic graphical models where the underlying structure \mathcal{G} is a directed acyclic graph (DAG) - also named DAG models in the literature (Geiger and Heckerman, 2002; Pearl, 2014). The joint probability distribution p can be fully specified by a set of parameters θ and factorizes according to \mathcal{G} as

$$p(\mathbf{x} \mid \theta, \mathcal{G}) = p(x_1, \dots, x_n \mid \theta, \mathcal{G}) = \prod_{i=1}^n p(x_i \mid \mathbf{x}_{pa(i)}, \theta_i, \mathcal{G}), \quad (1)$$

where \mathbf{x} is a realization for \mathbf{X} , $\theta = \cup_{i=1}^n \theta_i$, and the subsets $\{\theta_i\}_{i=1}^n$ are assumed to be disjoint. We denote the parents of node i by $pa(i)$ such that there is an directed edge from j to i if $j \in pa(i)$. Hence, we can also read the factorization in (1) by saying that a variable X_i is conditionally independent from its non-descendants given its parents $\mathbf{X}_{pa(i)}$ in \mathcal{G} . This is known as the Markov property (Lauritzen, 1996). We denote a Bayesian network by a set $\mathcal{B} = (\mathcal{G}, \theta)$. Given a data sample \mathcal{D} , learning a Bayesian network, therefore, means estimating both the network structure \mathcal{G} and the parameters θ .

1.1 Structure Learning for Bayesian Networks

Structure learning for Bayesian networks is an NP-hard problem (Chickering et al., 2004), mainly because the number of possible DAGs grows super-exponentially with the number of nodes n . Existing approaches to tackle this problem fall roughly into three categories:

- Constraint-based methods perform conditional independence tests for each pair of nodes given a subset of adjacent nodes in order of increasing complexity. Examples are the PC (Spirtes et al., 2000), FCI (Spirtes, 2001), and RFCI algorithms (Colombo et al., 2012). These methods are fast and computationally feasible when the underlying graph is sparse (Kalisch and Bühlmann, 2007; Uhler et al., 2013).
- Score-and-search methods rely on a scoring function and an algorithm that searches through the DAG space. One example is the greedy equivalence search (GES), which starts with an empty graph and goes through two phases of adding and removing edges based on the scores (Chickering, 2002). Recently, score-and-search algorithms based on integer linear programming have also been introduced (Cussens et al., 2017). More general approaches to score-and-search include sampling based methods, which aims to construct the full posterior distribution of DAGs given the data (Madigan et al., 1995; Giudici and Castelo, 2003; Friedman and Koller, 2003; Grzegorzczuk and Husmeier, 2008; Kuipers and Moffa, 2017), as well as strategies relying on dynamic programming (Koivisto and Sood, 2004). These methods can be slower than the constraint-based ones, but in general, provide better performance when computationally feasible (Heckerman et al., 1999).
- Lastly, the hybrid approaches bring together the above two solutions by restricting the initial search space using a constraint-based method in order to improve the speed and accuracy even further (Tsamardinos et al., 2006; Nandy et al., 2018; Kuipers et al., 2018).

For a broader overview on Bayesian networks we refer to the comprehensive review by Daly et al. (2011) as well as the textbook by Koller and Friedman (2009). A recent comparison of structure learning approaches can be found in Constantinou et al. (2021) for example.

1.2 Structure Learning with Ordinal Data

Alongside the search scheme, it is also important to properly account for the type of random variables studied and define a suitable scoring function. The focus of this work is on categorical variables, which can be either nominal or ordinal, depending on whether the ordering of the levels is relevant. Examples of nominal variables include sex (male, female, non-binary), genotype (AA, Aa, aa), and fasting before a blood test (yes, no). They are invariant to any random permutation of the categories. The listing order of the categories is, by contrast, an inherent property for ordinal variables. Examples are stages of disease (I, II, III), survey questions with Likert scales (strongly disagree, disagree, undecided, agree, strongly agree), as well as discretized continuous data, such as the body mass index (underweight, normal weight, overweight, obese), age groups (children, youth, adults, seniors) etc. (McDonald, 2009; Agresti, 2010). For discretized data, it is often difficult or impossible to access the underlying continuous source for practical or confidential reasons. The interactions amongst the latent variables can only manifest through their ordinal counterparts. From a technical perspective it is thus natural to think of ordinal variables as generated from a set of continuous latent variables through discretization.

Despite the prevalence of ordinal variables in applied research, not much attention has been paid to the problem of learning Bayesian networks from ordinal data. One solution is to apply constraint-based methods with a suitable conditional independence test, including but not limited to the Jonckheere-Terpstra test in the OPC algorithm of Musella (2013), copula-based tests (Cui et al., 2016), and likelihood-ratio tests (Tsagris et al., 2018). While a possible extension along this line is to explore more appropriate tests to improve the search, constraint-based methods, in general, tend to favor sparser graphs and can be dependent on the testing order (Uhler et al., 2013; Colombo and Maathuis, 2014). In the score-and-search regime, no scoring functions exist for ordinal variables, so metrics ignoring the ordering and based on the multinomial distribution are the typical alternatives. An obvious drawback is the loss of information associated with ignoring the ordering among the categories, resulting in inaccurate statistical analyses. Another potential issue is the tendency to overparameterization, especially when the number of levels is greater than 3. In this case, neither statistical nor computational efficiency can be achieved (Talvitie et al., 2019). However, for very large number of levels a continuous approximation may be adequate.

In this work, we develop an iterative score-and-search scheme specifically for ordinal data. Before describing the algorithm in detail, we briefly review the challenges. On the one hand, we need a scoring function that can preserve the ordinality amongst the categories. On the other hand, that function should also satisfy three important properties: decomposability, score equivalence, and consistency (Koller and Friedman, 2009). Decomposability of a score is the key to fast searching. Modifying a local structure does not require recomputing the score of the entire graph, which tremendously reduces the computational overhead. Score equivalence is less crucial but is necessary for search schemes relying on Markov equivalent classes, such as GES or hybrid methods that initialize with a PC output.

Here, two DAGs belong to the same Markov equivalent class if and only if they share the same skeleton and v -structures (Verma et al., 1991). A Markov equivalent class can be uniquely represented by a completed partially directed acyclic graph (CPDAG). Finally, a consistent score can identify the true DAG if sufficient data is provided. Examples of scores that satisfy all three properties are the BDe score (Heckerman and Geiger, 1995) for (nominal) categorical data and the BGe score (Geiger and Heckerman, 2002; Kuipers et al., 2014) for continuous data. It is, however, nontrivial to develop such a score for ordinal data.

To capture the ordinality, we use a widely accepted parameterization based on the multivariate probit models (Ashford and Sowden, 1970; Bock and Gibbons, 1996; Chib and Greenberg, 1998; Daganzo, 2014), with no additional covariates with respect to the variables whose joint distribution we wish to model and hence focusing on their dependency structure. As illustrated in Figure 1(a), each ordinal variable is assumed to be obtained by marginally discretizing a latent Gaussian variable. In this case, the order of the categories is encoded by the continuous nature of the Gaussian distribution.

More specifically we can have at least two possible formulations, both with their pros and cons. The first one (Figure 1(b)) assumes that the latent variables \mathbf{Y} jointly follow a multivariate Gaussian distribution, which factorizes according to a DAG. This approach resembles the Gaussian DAG model (Heckerman and Geiger, 1995; Geiger and Heckerman, 2002) in the latent space, with additional connections to the ordinal variables \mathbf{X} . Unlike in the continuous case, marginalizing out the latent variables will result in a non-decomposable observed likelihood. The conditional independence relationships will also disappear. This is because DAG models are not closed under marginalization and conditioning (Richardson et al., 2002; Silva and Ghahramani, 2009; Colombo et al., 2012). In Figure 1(b), the Gaussian variables Y_2 and Y_3 are conditionally independent given Y_1 , whereas the ordinal variables X_2 and X_3 are not independent given X_1 due to the presence of a common ancestor Y_1 in the latent space. As a result, traditional closed-form scoring functions such as the BGe metric (Heckerman and Geiger, 1995) cannot be applied, which is generally the case in the presence of hidden variables.

Another setup (Figure 1(c)) uses a non-linear structural equation model with the probit function as the link. In this case it is assumed that a DAG structure describes the relationship between the observed variables, whereas the latent Gaussian variables only facilitate the flow of information and can be marginalized out. It is equivalent to the probit directed mixed graph model of Silva and Ghahramani (2009). In this model, the observed likelihood is decomposable. We can recover the graph $X_2 \leftarrow X_1 \rightarrow X_3$ after marginalization. However, the resulting score is not score-equivalent because the joint probability distribution fails to meet the complete model equivalence assumption as described by Geiger and Heckerman (2002).

Our solution builds on the first formulation, which we call the *latent Gaussian DAG model*. The key is to observe that the complete-data likelihood is both decomposable and score-equivalent. Further adding a BIC-type penalty will still preserve these key properties of the score, as well as consistency (Schwarz et al., 1978; Rissanen, 1987; Barron et al., 1998). The Structural EM algorithm of Friedman (1997) addresses the problem of structure learning in the presence of incomplete data by embedding the search for the optimal structure into an Expectation-Maximization (EM) algorithm. We apply the Structural EM

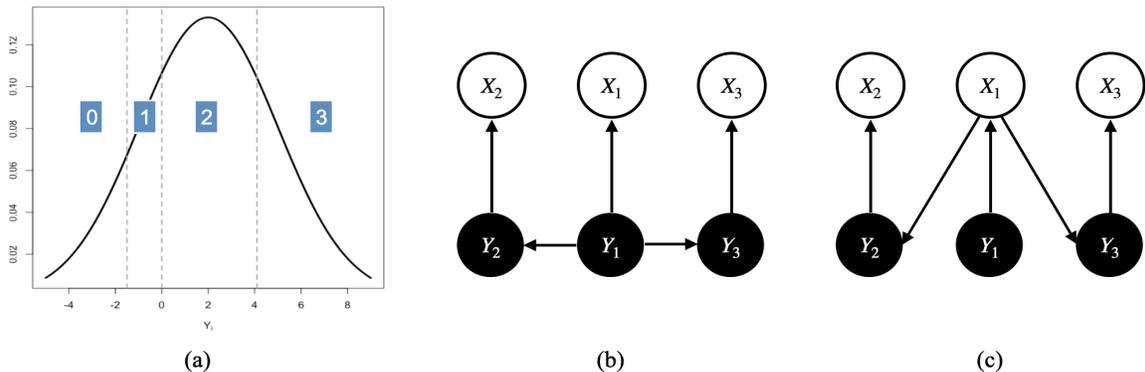


Figure 1: (a) We assume that an ordinal variable is obtained by marginally discretizing a latent Gaussian variable. (b) and (c) are examples of three-node graphs for two different probit models discussed in Section 1.2. X_1 , X_2 and X_3 are ordinal variables, each obtained by discretizing a latent variable Y_i with associated Gaussian parameters θ_i , $i = 1, 2, 3$. Latent nodes are shaded for clarity.

to deal with the latent Gaussian variable construction of ordinal data in a procedure which we call the *Ordinal Structural EM (OSEM)* algorithm. Note that Webb and Forster (2008) also used the latent Gaussian DAG model and aimed to determine the network structure from the ordinal data using a reversible jump MCMC method. Our approach differs in that it explicitly exploits score decomposability, allowing for search schemes that are computationally more efficient, such as order and partition MCMC (Friedman and Koller, 2003; Kuipers and Moffa, 2017), or hybrid methods.

The rest of this article is structured as follows. In Section 2, we formally define the latent Gaussian DAG model and introduce the corresponding identifiability constraints. In Section 3, we integrate the model into the Structural EM framework. In Section 4, we use synthetic and real data to illustrate the superior performance of our proposed method compared to the alternatives in terms of structure recovery and prediction. In Section 5, we apply our method to a psychological survey data set. Finally, in Section 6, we discuss the implications of our results as well as possible directions for future work.

2. The Latent Gaussian DAG Model

Let $\mathbf{X} = (X_1, \dots, X_n)^\top$ be a collection of n ordinal variables, where X_i takes values in the set $\{\tau(i, 1), \tau(i, 2), \dots, \tau(i, L_i)\}$ with $\tau(i, 1) < \tau(i, 2) < \dots < \tau(i, L_i)$, $i = 1, \dots, n$. We assume that the number of levels $L_i \geq 2$, so each variable should at least be binary. It is common to set $\tau(i, j) = j - 1$ for all $1 \leq j \leq L_i$, *i.e.* $\tau(i, 1) = 0$, $\tau(i, 2) = 1$, and so on. Further, we assume that each X_i is obtained by discretizing an underlying Gaussian variable Y_i using the thresholds $-\infty =: \alpha(i, 0) < \alpha(i, 1) < \dots < \alpha(i, L_i - 1) < \alpha(i, L_i) := \infty$. Let $\boldsymbol{\alpha}_i = (\alpha(i, 0), \dots, \alpha(i, L_i))^\top$ and $\boldsymbol{\alpha} = \{\boldsymbol{\alpha}_i\}_{i=1}^n$. The hidden variables $\mathbf{Y} = (Y_1, \dots, Y_n)^\top$

jointly follow a multivariate Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, which factorizes according to some DAG \mathcal{G} . Formally, the latent Gaussian DAG model is given by

$$\begin{aligned}
 Y_i \mid \mathbf{y}_{pa(i)}, \vartheta_i, \mathcal{G} &\sim \mathcal{N}(\mu_i + \sum_{j \in pa(i)} b_{ji}(y_j - \mu_j), v_i) \\
 P(X_i = \tau(i, l) \mid Y_i = y_i, \boldsymbol{\alpha}_i) &= \mathbf{1}(y_i \in [\alpha(i, l - 1), \alpha(i, l))), \quad l = 1, \dots, L_i \\
 p(\mathbf{x}, \mathbf{y} \mid \theta, \mathcal{G}) &= \prod_{i=1}^n \phi(y_i \mid \mathbf{y}_{pa(i)}, \vartheta_i, \mathcal{G}) p(x_i \mid y_i, \boldsymbol{\alpha}_i)
 \end{aligned} \tag{2}$$

where $\theta = \cup_{i=1}^n \theta_i$ with $\theta_i = (\vartheta_i, \boldsymbol{\alpha}_i)$, $\vartheta_i = (\mu_i, \mathbf{b}_i, v_i)$ and $\mathbf{b}_i = (b_{ji})_{j \in pa(i)}$ for all $i = 1, \dots, n$. Since the discretization is marginal and deterministic, the joint probability distribution $p(\mathbf{x}, \mathbf{y} \mid \theta, \mathcal{G})$ remains decomposable and score-equivalent as in the Gaussian DAG model (Geiger and Heckerman, 2002). Alternatively we can parametrize the complete joint distribution of the hidden and observed variables using a mean vector $\boldsymbol{\mu} = (\mu_1, \dots, \mu_n)^\top \in \mathbb{R}^n$ and a symmetric positive definite covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$,

$$p(\mathbf{x}, \mathbf{y} \mid \theta, \mathcal{G}) = \phi(\mathbf{y} \mid \boldsymbol{\mu}, \Sigma, \mathcal{G}) p(\mathbf{x} \mid \mathbf{y}, \boldsymbol{\alpha}). \tag{3}$$

It follows from Silva and Ghahramani (2009) that we can write the transformation between $\{\mathbf{b}_i, v_i\}_{i=1}^n$ and Σ as

$$\Sigma = (\mathbf{I} - \mathbf{B})^{-1} \mathbf{V} (\mathbf{I} - \mathbf{B})^{-\top}, \tag{4}$$

where $(\mathbf{B})_{ij} = b_{ji}$ and \mathbf{V} is an n -by- n diagonal matrix with $\mathbf{V}_{ii} = v_i$.

2.1 Identifiability

Different underlying Gaussian variables \mathbf{Y} may generate the same contingency table for \mathbf{X} , by simply shifting and scaling the thresholds using the corresponding means and variances. For example, we can obtain the same ordinal variable X by discretizing either $Y_1 \sim \mathcal{N}(0, 1)$ at $\{-1, 1\}$ or $Y_2 \sim \mathcal{N}(-1, 100)$ at $\{-11, 9\}$. In other words, there is not a one-to-one mapping between the cell probabilities in the table and the parameters $\theta = (\boldsymbol{\alpha}, \boldsymbol{\mu}, \Sigma)$. Thus, we need to impose some constraints to ensure model identifiability.

Typically, each dimension of \mathbf{X} requires two constraints. Webb and Forster (2008) choose to fix the lowest and the highest thresholds for deriving the ordinal variables. For binary variables, they restrict the only threshold at zero and set $v_i = 1$. Instead, we find it computationally more convenient to standardize each latent dimension. More precisely, we let $\mu_i = 0$ for all $i = 1, \dots, n$ and constrain the covariance matrix to be in correlation form using a diagonal matrix $D := \text{diag}(d_1, \dots, d_n)$ with $d_i = \sqrt{\Sigma_{ii}}$, *i.e.* we replace Σ by its transformation $D^{-1} \Sigma D^{-1}$. Due to its symmetry the correlation matrix is identified by $\frac{n(n-1)}{2}$ off-diagonal parameters $\Sigma_{ij} = \Sigma_{ji} = \rho_{ij}$ for all $i \neq j$. Imposing these constraints ensures that the thresholds and the correlation matrix will be identifiable, which is sufficient for deducing the hidden DAG \mathcal{G} . The mean vector and the variances remain unidentifiable.

3. The Ordinal Structural EM Algorithm

Given a sample $\mathcal{D}_{\mathbf{X}} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$ of size N for \mathbf{X} , our goal is to learn both the parameters and the structure of the Bayesian network $\mathcal{B} = (\mathcal{G}, \theta)$ which best explains the observed

data. Making inference about Bayesian networks with unknown structure is an intrinsically challenging problem, and in the case of ordinal variables an additional difficulty originates from the presence of latent variables. To estimate the parameters θ for a given DAG \mathcal{G} , we can use the Expectation-Maximization (EM) algorithm (Dempster et al., 1977), a common approach to handle maximum likelihood estimation in the presence of missing data. Conversely, with fully observed data, we may use one of the methods discussed in Section 1.1 for structure learning. In the presence of hidden variables, the above strategies fail, because the marginal density for \mathbf{X} no longer decomposes over each node and its parents. In the sequel, we consider combining the latent Gaussian DAG model with the Structural EM algorithm of Friedman (1997), where the structure learning step is wrapped inside the EM procedure. The resulting framework is the Ordinal Structural EM (OSEM) algorithm.

3.1 The EM and the Structural EM Algorithms

In the presence of latent variables, computing the maximum likelihood estimates for θ by maximizing the observed data log-likelihood

$$\ell(\theta; \mathcal{D}_{\mathbf{X}}) = \log p(\mathcal{D}_{\mathbf{X}} | \theta) = \log \int p(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathbf{Y}} | \theta) d\mathbf{y}^1 \cdots d\mathbf{y}^N \quad (5)$$

can be intractable. The EM algorithm addresses this problem by alternating between two steps:

- **E-step** (Expectation): given the observed data and the current estimate of the parameters $\theta^{(t)}$, $t \in \{0, 1, 2, \dots\}$, compute the expected value of the complete-data (including the hidden variables) likelihood with respect to the distribution of the hidden variables conditional on the observed variables,

$$\begin{aligned} Q(\theta, \theta^{(t)}) &= \mathbb{E}_{\mathbf{Y} | \mathbf{x}, \theta^{(t)}} [\log p(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathbf{Y}} | \theta)] \\ &= \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \theta^{(t)}} [\log p(\mathbf{X}^j, \mathbf{Y}^j | \theta) | \mathbf{X}^j = \mathbf{x}^j]; \end{aligned} \quad (6)$$

- **M-step** (Maximization): update the parameters by maximising the Q function with respect to θ as

$$\theta^{(t+1)} = \arg \max_{\theta \in \Theta} Q(\theta, \theta^{(t)}). \quad (7)$$

The EM procedure guarantees convergence of the observed data likelihood $\ell(\theta^{(t)}; \mathcal{D}_{\mathbf{X}})$ to a local maximum as $t \rightarrow \infty$ (Dempster et al., 1977).

The Structural EM algorithm is an extension of the original EM for learning Bayesian networks. The idea is to recursively choose a structure and a set of parameters that improve the following *expected scoring function*:

$$\tilde{Q}(\mathcal{G}, \theta; \mathcal{G}^{(t)}, \theta^{(t)}) = \underbrace{\mathbb{E}_{\mathbf{Y} | \mathbf{x}, \mathcal{G}^{(t)}, \theta^{(t)}} [\log p(\mathcal{D}_{\mathbf{X}}, \mathcal{D}_{\mathbf{Y}} | \mathcal{G}, \theta)]}_{Q(\mathcal{G}, \theta; \mathcal{G}^{(t)}, \theta^{(t)})} - \text{Penalty}(\mathcal{G}, \theta). \quad (8)$$

Here, we choose a BIC penalty term in order to ensure that the score Eq. (8) remains decomposable, score-equivalent, and consistent (Theorem 18.2 of Koller and Friedman (2009)). In the following sections, we elaborate on the steps in the OSEM algorithm.

3.2 Initialization

The original algorithm of Friedman (1997) used a random initialization for both the structure \mathcal{G} and the parameters θ . However, we can achieve faster convergence if we can cheaply obtain rough estimates of the parameters which maximise the penalised likelihood function, and start the EM procedure from such estimates.

Given the identifiability constraints in Section 2.1, we only need to estimate the thresholds α once at the very beginning. More precisely, given a sample $\mathcal{D}_{\mathbf{X}}$, we can estimate the thresholds $\alpha(i, l)$ for all $i = 1, \dots, n$ and $l = 1, \dots, L_i - 1$ as

$$\hat{\alpha}(i, l) = \Phi^{-1}\left(\frac{1}{N} \sum_{j=1}^N \mathbb{1}(x_i^j \leq \tau(i, l))\right), \quad (9)$$

which are the empirical quantiles of the standard normal distribution.

Next, we assume that the initial DAG $\mathcal{G}^{(0)}$ is a full DAG. Conditioned on the thresholds, we want to find the correlation matrix by maximizing the observed data log-likelihood,

$$\Sigma^{(0)} \mid \hat{\alpha} = \arg \max_{\Sigma} \log p(\mathcal{D}_{\mathbf{X}} \mid \mathbf{0}, \Sigma, \mathcal{G}^{(0)}), \quad (10)$$

which can be computationally too expensive. Alternatively, we can use a pairwise likelihood approach (Kuk and Nott, 2000; Renard et al., 2004; Varin and Czado, 2010), where we estimate each off-diagonal entry ρ_{ij} in the correlation matrix as

$$\rho_{ij}^{(0)} \mid \hat{\alpha} = \arg \max_{\rho_{ij}} \log p\left(\mathcal{D}_{X_i \cup X_j} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho_{ij} \\ \rho_{ij} & 1 \end{bmatrix}, \mathcal{G}^{(0)}\right). \quad (11)$$

Let $\theta^{(0)} = \{\Sigma^{(0)}, \hat{\alpha}\}$ with $(\Sigma^{(0)})_{ij} = \rho_{ij}^{(0)}$ for all $i \neq j$ and $(\Sigma^{(0)})_{ii} = 1$ for all $i = 1, \dots, n$. This approach is fast and easy to implement at the expense of accuracy. If the resulting correlation matrix is not positive-definite, one remedy is to smooth the matrix by coercing the non-positive eigenvalues into slightly positive numbers. As this is only an initialization, the difference caused by smoothing should not severely impact the following stages.

3.3 Structure Update

The algorithm iterates through two steps: the structure update and the parameter update. Given the current estimate $(\mathcal{G}^{(t)}, \theta^{(t)})$, the structure update searches for the DAG that maximizes the expected scoring function by computing the expected statistics instead of the actual ones in the complete-data setting.

In our case, computing the expected values in Eq. (6) is expensive, so we use the Monte Carlo EM algorithm of Wei and Tanner (1990) as an approximation. For each observation \mathbf{x}^j , we draw a sample of size K from the truncated multivariate normal distribution

$$\mathbf{Y}^j \mid \mathbf{x}^j, \Sigma^{(t)}, \hat{\alpha}, \mathcal{G}^{(t)} \sim \text{TMN}(\mathbf{0}, \Sigma^{(t)} \mid \mathbf{x}^j, \hat{\alpha}), \quad (12)$$

where the sampling region of the Gaussian data is restricted to the domain specified by \mathbf{x}^j and $\hat{\boldsymbol{\alpha}}$. Then, we can approximate the expected covariance matrix as

$$\hat{\Sigma} = \frac{1}{N} \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} [(\mathbf{Y}^j)(\mathbf{Y}^j)^\top] \approx \frac{1}{N} \frac{1}{K} \sum_{j=1}^N \sum_{k=1}^K (\mathbf{y}^{j(k)})(\mathbf{y}^{j(k)})^\top. \quad (13)$$

From Eq. (2) and Eq. (3), we decompose the Q function with respect to each node and its parents,

$$Q(\mathcal{G}, \theta; \mathcal{G}^{(t)}, \theta^{(t)}) = \sum_{i=1}^n \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log \frac{\phi(\mathbf{Y}_{\{i\} \cup pa(i)}^j | (\mathbf{0}, \Sigma)_{\{i\} \cup pa(i)}, \mathcal{G})}{\phi(\mathbf{Y}_{pa(i)}^j | (\mathbf{0}, \Sigma)_{pa(i)}, \mathcal{G})} \right], \quad (14)$$

where $(\mathbf{0}, \Sigma)_O$ are the parameters indexed by the set O for $O \subseteq \{1, \dots, n\}$. Let $\hat{\theta} = \{\hat{\Sigma}, \hat{\boldsymbol{\alpha}}\}$. By substituting Eq. (13) into Eq. (14) and using the standard normal density function, we rewrite Eq. (8) as

$$\begin{aligned} \tilde{Q}(\mathcal{G}, \hat{\theta}; \mathcal{G}^{(t)}, \theta^{(t)}) &= Q(\mathcal{G}, \hat{\theta}; \mathcal{G}^{(t)}, \theta^{(t)}) - \frac{\log N}{2} \dim(\mathcal{G}) \\ &= \sum_{i=1}^n \left(-\frac{N}{2} \left[\log \left(\hat{\Sigma}_{i,i} - \hat{\Sigma}_{i,pa(i)} \hat{\Sigma}_{pa(i),pa(i)}^{-1} \hat{\Sigma}_{pa(i),i} \right) \right] - \frac{\log N}{2} \dim(Y_i, \mathbf{Y}_{pa(i)}) \right), \end{aligned} \quad (15)$$

which is an expected version of the BIC score as in the complete-data case. Therefore, to update the structure we need to solve the following maximization problem,

$$\mathcal{G}^{(t+1)} = \arg \max_{\mathcal{G}: \text{DAG over } (\mathbf{X}, \mathbf{Y})} \tilde{Q}(\mathcal{G}, \hat{\theta}; \mathcal{G}^{(t)}, \theta^{(t)}), \quad (16)$$

which we can address by using any of the existing score-based or hybrid schemes, such as the GES and MCMC samplers.

3.4 Parameter Update

Conditioned on the structure estimate $\mathcal{G}^{(t+1)}$, the goal in the parameter update is to find $\theta^{(t+1)}$ such that

$$\theta^{(t+1)} = \arg \max_{\theta \in \Theta} Q(\mathcal{G}^{(t+1)}, \theta; \mathcal{G}^{(t)}, \theta^{(t)}) - \frac{\log N}{2} \dim(Y_i, \mathbf{Y}_{pa(i)}). \quad (17)$$

The Q function can be expressed in terms of the conditional parameters from Eq. (2):

$$\begin{aligned} & \sum_{i=1}^n \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log \phi(Y_i^j | \mathbf{Y}_{pa(i)}^j, \theta_i, \mathcal{G}^{(t+1)}) \right] \\ &= \sum_{i=1}^n -\frac{N}{2} \left(\log(2\pi v_i) + \frac{1}{v_i} \frac{1}{N} \sum_{j=1}^N \underbrace{\mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[(Y_i^j - \mathbf{b}_i^\top \mathbf{Y}_{pa(i)}^j)^2 \right]}_{\approx \frac{1}{K} \sum_{k=1}^K \left[(y_i^{j(k)} - \mathbf{b}_i^\top \mathbf{y}_{pa(i)}^{j(k)})^2 \right]} \right). \end{aligned} \quad (18)$$

The truncated multivariate normal samples in the previous structure update can be recycled here, because the parameters $\theta^{(t)}$ have not yet been updated. To maximize $Q(\mathcal{G}^{(t+1)}, \theta; \mathcal{G}^{(t)}, \theta^{(t)})$ we can then effectively use the multi-step conditional maximization (Meng and Rubin, 1993) following the topological order in $\mathcal{G}^{(t+1)}$. Suppose we have estimated $\theta_{pa(i)}^{(t+1)}$ and consider now updating θ_i . Let $\mathbf{Y}_{pa(i)}^{(k)} = (\mathbf{y}_{pa(i)}^{1(k)}, \dots, \mathbf{y}_{pa(i)}^{N(k)})^\top \in \mathbb{R}^{N \times |pa(i)|}$ and $\mathbf{y}_i^{(k)} = (y_i^{1(k)}, \dots, y_i^{N(k)})^\top \in \mathbb{R}^N$ represent the sampled design matrix and the sampled response vector respectively. We first differentiate $Q(\mathcal{G}^{(t+1)}, \theta; \mathcal{G}^{(t)}, \theta^{(t)})$ with respect to \mathbf{b}_i and obtain

$$\mathbf{b}_i^{(t+1)} = \left(\sum_{k=1}^K \left(\mathbf{Y}_{pa(i)}^{(k)} \right)^\top \mathbf{Y}_{pa(i)}^{(k)} \right)^{-1} \left(\sum_{k=1}^K \left(\mathbf{Y}_{pa(i)}^{(k)} \right)^\top \mathbf{y}_i^{(k)} \right). \quad (19)$$

Conditioned on $\mathbf{b}_i^{(t+1)}$, the update for v_i is then

$$v_i^{(t+1)} = \frac{1}{N} \frac{1}{K} \sum_{j=1}^N \sum_{k=1}^K \left(y_i^{j(k)} - (\mathbf{b}_i^{(t+1)})^\top \mathbf{y}_{pa(i)}^{j(k)} \right)^2. \quad (20)$$

By introducing the BIC penalty, this procedure is equivalent to performing nodewise regression with best subset selection (James et al., 2013) in the latent space. Only minor modifications in the active components of $\mathbf{b}_i^{(t+1)}$ are required.

For the purpose of identifiability and the next structure update, the final step is to convert $\theta^{(t+1)} = (\mathbf{b}_i^{(t+1)}, v_i^{(t+1)})_{i=1}^n$ to its corresponding correlation form with Eq. (4) and rescaling.

3.5 Summary

The latent Gaussian DAG model assumes that the ordinal variables originate from element-wise discretization of a set of Gaussian variables, which follow a DAG structure. Given observations from the ordinal data only, we call Ordinal Structural EM (OSEM) algorithm the process of estimating the hidden DAG structure, and we summarize the procedure in Algorithm 1, along with more technical details (Appendix A). Theorem 3.1 of Friedman (1997) states that the penalized observed data log-likelihood can only improve at each iteration, and therefore, guarantees the convergence of the algorithm to a local optimum.

4. Experimental Results

In this section, we first use simulation studies to assess the performance of our proposed method in recovering the hidden network structure from observed ordinal data. Then, we evaluate its predictive performance in terms of average log loss using real data sets.

4.1 Structure Recovery

We randomly generate DAGs using the `randDAG` function from the `pcalg` package (Kalisch et al., 2012), followed by generating the corresponding Gaussian data according to the topological order in the DAGs. Then, we discretize the Gaussian data to obtain an ordinal data set, from which we try to learn the original structure describing the relationship between

the latent Gaussian variables. We compare our Ordinal Structural EM algorithm (OSEM) to five other existing approaches:

1. PC algorithm with the nominal G^2 test (NPC);
2. PC algorithm of Musella (2013) with the ordinal Jonckheere–Terpstra test (OPC);
3. PC algorithm with the Gaussian test using Fisher’s z–transformation (GPC);
4. hybrid method with the BDe score and NPC as initial search space (BDe);
5. hybrid method with the BGe score and GPC as initial search space (BGe).

Note that none of these methods assume a latent Gaussian DAG model. Namely, they all build a network model directly on the observed ordinal variables. OSEM on the other hand estimates the hidden structure on the latent Gaussian variables. Therefore a direct comparison of the performance with respect to the network structure needs to be taken with caution, especially in light of the fact that conditional independence relationships do not automatically extend from the latent to the observed space (see also Section 1.2). To the best of our knowledge, however, no generative model for ordinal data currently exists that provides both decomposability and score equivalence in the observed space. In the absence of alternative methods which are capable of dealing with the latent variable construction of ordinal data, it may still be insightful to compare the network accuracy of OSEM to that of the most popular methods currently in use to learn network models for various data types. Finally, to compare OSEM to other algorithms in a manner that is agnostic to the data generating scheme, we will evaluate predictive performance on real data sets in Section 4.2. Detailed implementations for all methods are summarised in Appendix B.7.

4.1.1 PERFORMANCE METRICS

We assess the performance of the model by comparing the estimated structure with the true structure. Notice that we can only identify the true DAG up to its Markov equivalence class, so it is more appropriate to compare the corresponding CPDAGs. However, the metrics based on CPDAG differences can still be too harsh, because many directed edges are induced by a few number of v-structures. Therefore, we choose to compare the patterns of the DAGs in the sense of Meek (1995), where the v-structures are the only directed edges. In other words, we measure the structural hamming distance (SHD) between two DAGs using the skeleton differences plus the v-structure differences.

The estimated DAG or CPDAG is first converted into a pattern. If an edge in the estimated pattern matches exactly the same direction as the corresponding edge in the true pattern, we count it as 1 true positive (TP). If the direction is wrong but exactly one of this edge and the corresponding edge is undirected, then we count it as 0.5 TP. We subtract the number of true positives from the total number of estimated edges to get the number of false positives (FP). Furthermore, we compute the true positive rate (TPR) and the false positive rate (FPR_p) as

$$\text{TPR} = \frac{\text{TP}}{\text{P}} \quad \text{and} \quad \text{FPR}_p = \frac{\text{FP}}{\text{P}}, \quad (21)$$

where P is the total number of edges in the true pattern. Scaling the false positives by P instead of the number of real negatives N can lead to a more comparable visualization, because if the sparsity of the network stays the same, N grows quadratically as n increases. We illustrate the comparisons mainly through Receiver Operating Characteristics (ROC) curves. In particular, we plot the TPR against the FPR_p with the following penalization parameters:

- significance level for the conditional independence tests:
 $\alpha \in \{0.001, 0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2, 0.25, 0.3\}$.
- equivalent sample size for the BDe score:
 $\chi \in \{0.0001, 0.001, 0.01, 0.1, 1, 10, 20, 40, 60, 80\}$;
- coefficient to be multiplied by the BIC penalty:
 $\lambda \in \{1, 1.5, 2, 2.5, 3, 4, 6, 10, 20, 30\}$;
- coefficient to be multiplied by the precision matrix in the BGe score:
 $a_m \in \{0.0001, 0.001, 0.01, 0.025, 0.05, 0.1, 0.25, 0.5, 1, 1.5\}$, and the degree of freedom a_w in the Wishart distribution is set to be $n + a_m + 1$.

Unlike comparing skeletons, the ROC curves created this way can be non-monotonic. Some edges in the patterns may jump between being directed and undirected when the penalty is too small or too large, so the number of TPs may not be monotonically increasing. Furthermore, one can read the SHDs scaled by P directly from the plots, which are the Manhattan distances from the curves to the point $(0, 1)$.

4.1.2 SIMULATION RESULTS

We generate random DAGs with 4 expected neighbours per node. The edge weights are uniformly sampled from the intervals $(-1, -0.4) \cup (0.4, 1)$. We consider 3 network sizes $n \in \{12, 20, 30\}$ and 3 sample sizes $N \in \{300, 500, 800\}$. For each DAG, we generate the associated Gaussian data and perform element-wise discretization randomly using the symmetric Dirichlet distribution $\text{Dir}(L_i, \nu)$, where L_i is the number of ordinal levels in the i^{th} dimension and ν is a concentration parameter. More specifically, we first generate the cell probabilities in the ordinal contingency table of X_i from $\text{Dir}(L_i, \nu)$. According to the cell probabilities, we can compute reversely the thresholds for cutting the Gaussian variable Y_i using the normal quantile function (Figure S2). Here, we draw L_i randomly from the interval $[2, 4]$ to mimic the typical number of ordinal levels and choose ν to be 2 to avoid highly skewed contingency tables.

For each configuration, we repeat 100 times the process of randomly generating the data and estimating the structure using the six approaches, followed by plotting the ROC curves accordingly. Each point in Figure 2 represents a tuple (TPR, FPR_p). The lines are created by interpolating the average TPR against the average FPR_p at each penalization value. Unlike FPR_n, FPR_p can be greater than 1, but only points within $[0, 1]$ are shown.

In general, the three PC-based methods alone are too conservative to recover the patterns of the true underlying structures, regardless of how we increase the significance level.

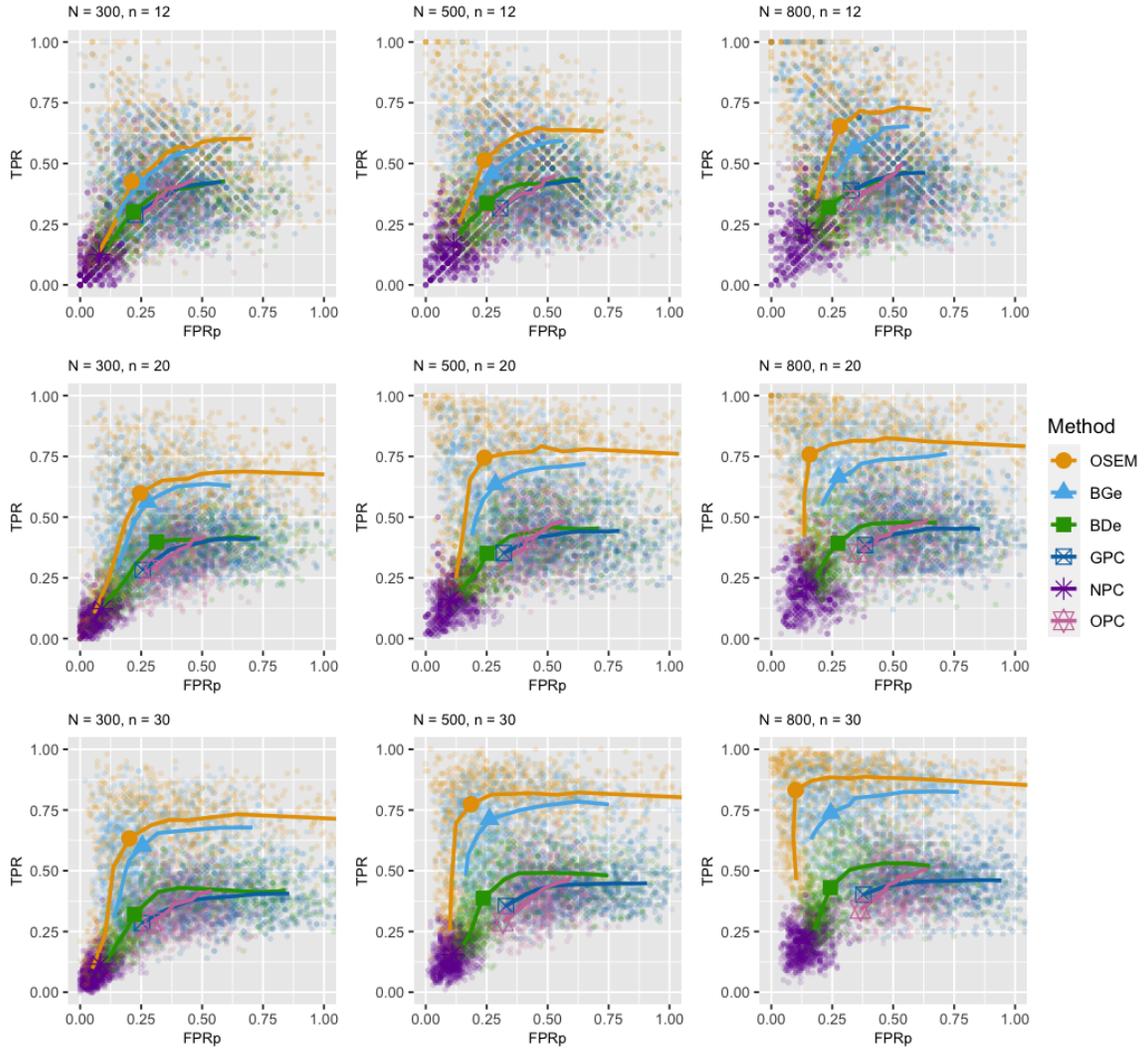


Figure 2: The comparison of the performance in recovering the true pattern between the OSEM algorithm and five other existing approaches: BGe, BDe, GPC, NPC, and OPC. The ROC curves are created by plotting the TPR against the FPR_p for sample sizes $N \in \{300, 500, 800\}$, network sizes $n \in \{12, 20, 30\}$, 3 expected number of levels, and a Dirichlet concentration parameter of 2. Both the x-axis and the y-axis are limited to $[0, 1]$. For each method, the point on the line corresponds to the lowest SHD (the highest sum of TPR and $(1 - \text{FPR}_p)$).

Amongst them, the GPC algorithm performs the best, followed by the OPC and NPC algorithms.

While the performance of the hybrid method with the BDe score is sometimes much better than the PC algorithms, the upper bound of the BDe band barely touches the

lower bound of the BGe band. This observation is unanticipated in two ways. On one hand, ignoring the ordering amongst the categories is not as harmless as one may expect. If our modeling assumptions are correct, the common practice of using the multinomial distribution cannot be taken for granted, as the estimated DAG can be far off. On the other hand, the BGe score is more powerful than we may have anticipated. Even though treating the ordinal data as continuous is inaccurate, this is still capable of detecting many true conditional independence relationships in the latent space. Thus, leaving aside our OSEM algorithm, one should rather use the BGe score instead of the popular BDe score for Bayesian network learning from ordinal data.

In all cases, our OSEM algorithm demonstrates a strong improvement. When the sample size is large enough, the contingency tables are more accurate, so recovering the original covariance structure through the EM iterations becomes more likely. With fixed sample size, increasing the network size does not make the performance of our model deteriorate, as long as the sparsity of the network stays the same. Additional simulation results can be found in Appendix B, including 1) ROC curves for skeletons, 2) the effect of thresholds, 3) the effect of network sparsity, 4) runtimes, and 5) comparison against the most recent structure learning methods for mixed data.

4.2 Predictive Performance

For score-based approaches, we also evaluate the predictive performance using five real ordinal data sets from McNally et al. (2017) and the UCI machine learning repository (Dua and Graff, 2017) (Table 1).

| data set | Sample size | Number of variables | Average levels |
|------------------------------|-------------|---------------------|----------------|
| OCD and Depression | 408 | 24 | 4.67 |
| Congressional Voting Records | 435 | 17 | 2 |
| Contraceptive Method Choice | 1473 | 9 | 3.3 |
| Primary Tumor | 339 | 17 | 2.18 |
| SPECT Heart | 267 | 23 | 2 |

Table 1: Description of the five real data sets in Section 4.2. Except the OCD and Depression data set, all other data sets are obtained from the UCI machine learning repository.

In addition to OSEM, BGe, and BDe, we have included the PCART algorithm of Talvitie et al. (2019) for comparison, which is a score-based method for mixed continuous, nominal and ordinal data. For each data set and method, we train the network with 80% of the data points, and conditioned on the structure, we compute the log loss on the remaining 20% test cases. To ensure that the method with the continuous BGe score has a comparable log loss, we substitute its maximum a posteriori covariance matrix based on the estimated DAG into the OSEM log loss function (Appendix B.6). Taking the BDe as the baseline, which is a standard choice for categorical data, we plot in Figure 3 the relative log loss per instance corresponding to the optimal tuning parameters over 100 random splits. We

can clearly see that OSEM consistently outperforms the baseline approach and is highly competitive across the board, while BGe and PCART are much less robust. This further suggests that modelling the ordinal data as being generated from a hidden continuous space can be more effective in practice, making the OSEM algorithm a generally better choice for both structure learning and prediction. Details including the computation of the log loss, additional figures and more extensive comparisons on sub-sampled data sets can be found in Appendix B.6.

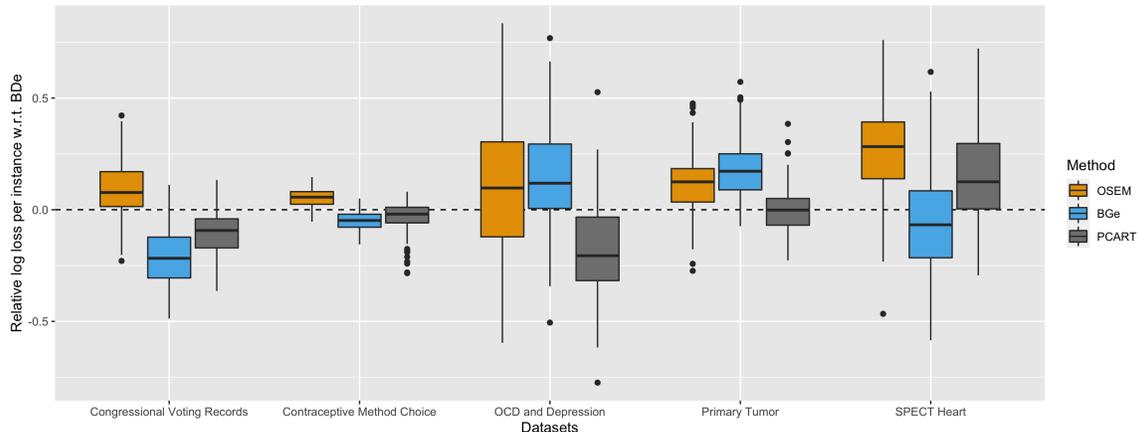


Figure 3: The relative log loss per instance with respect to the BDe baseline (dashed horizontal line at 0) for OSEM, BGe, and PCART over the five data sets and 100 random splits.

5. Application to Psychological Data

In this section, we provide an example applying our method to psychological survey data, where the potential of Bayesian network models to describe complex relationships has recently gained momentum (McNally et al., 2017; Moffa et al., 2017; Kuipers et al., 2019; Bird et al., 2019). We use a data set of size 408 from a study of the functional relationships between the symptoms of obsessive-compulsive disorder (OCD) and depression (McNally et al., 2017). It consists of 10 five-level ordinal variables representing the OCD symptoms and 14 four-level ordinal variables representing the depression symptoms, measured with the Yale-Brown Obsessive-Compulsive Scale (Y-BOCS-SR) (Steketee et al., 1996) and the Quick Inventory of Depressive Symptomatology (QIDS-SR) (Rush et al., 2003) respectively. Details are summarized in Supplementary Table S2. Note that “decreased vs. increased appetite” and “weight loss vs. gain” in the original questionnaire QIDS-SR de facto address the same questions from opposite directions. To avoid including two nodes in the network representing almost identical information, we replace each pair of variables with a one seven-level variable—*appetite* and *weight* respectively. Contradictory answers are merged into the closest levels. For example, if “decreased appetite” receives a score of 2 and “increased appetite” receives a score of 1, we assigns -1 to the variable *appetite*.

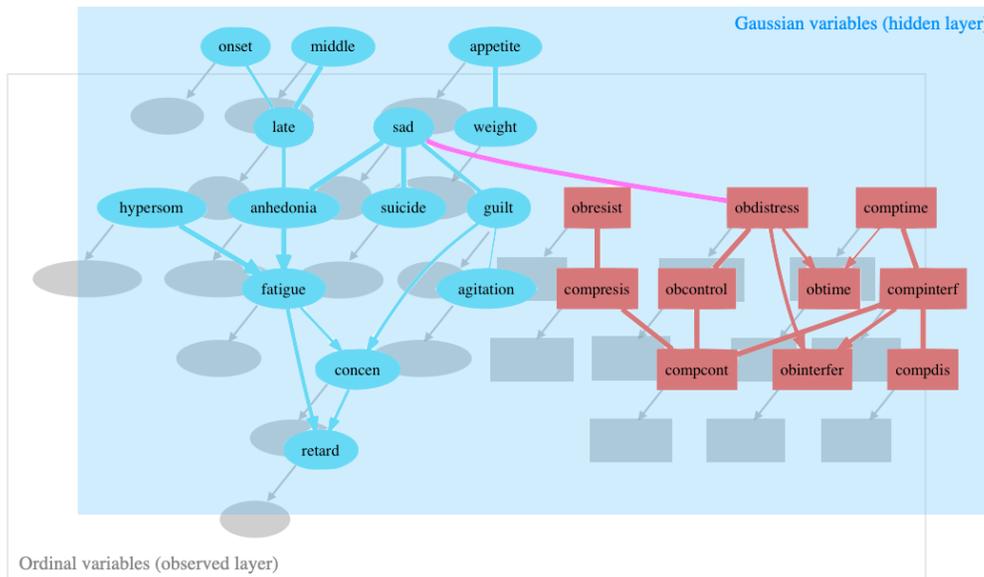


Figure 4: CPDAG of the hidden network structure estimated via the OSEM algorithm for the obsessive-compulsive disorder (OCD) symptoms (rectangles) and depression symptoms (ellipses) from a data set provided by McNally et al. (2017). The labelled nodes are the latent Gaussian variables, and the shaded nodes are the corresponding observed ordinal variables. The Monte Carlo sample size K is 5, and the penalty coefficient λ is 6. The thickness of the undirected edges reflects the percentage of time they occur in the skeletons of 500 Bootstrapped CPDAGs. If an edge between two labelled nodes is directed, then its thickness represents its directional strength. For clarity, we highlight the bridge between the two clusters of nodes ($sad - obdistress$).

In addition to having a suitable number of observations $N = 408$, variables $n = 24$, and an average number of levels greater than 4, we also check that none of the marginal contingency tables have weights concentrating on one end. Thus, our OSEM algorithm is particularly well suited in this setting. We choose the Monte Carlo sample size K to be 5 and the penalty coefficient λ to be 6, which corresponds to the highest sum of average TPR and $(1 - FPR_p)$ on the ROC curves for $N = 500$, $n = 20$ or 30, $\mathbb{E}[L_i] = 4$ or 5, and $\nu = 1$ from our simulations studies. The resulting CPDAG is shown in Figure 4. We depict the nodes related to OCD symptoms with rectangles and the nodes related to depression symptoms with ellipses. The thickness of the undirected edges reflects the percentage of time they occur in the skeletons of 500 bootstrapped CPDAGs. If an edge is directed, then its thickness represents its directional strength.

We reproduce the Bayesian network of McNally et al. (2017) by running the hill-climbing algorithm with the Gaussian BIC score and retaining the edges that appear in at least 85% of the 500 Bootstrapped networks. The corresponding CPDAG is depicted in Supplementary

Figure S11. In both graphs, the OCD and depression symptoms form separate clusters, bridged by the symptoms of sadness and distress caused by obsessions. With the exception of a few edge discrepancies (*e.g. late – anhedonia, guilt → suicide*), the two graphs appear to be mostly similar, which is consistent with our simulation results. When the average number of ordinal levels is relatively large, violating the hidden Gaussian assumption does not substantially reduce the performance in recovering the true edges. Nevertheless, the structure estimated by the OSEM algorithm should still be more reliable.

Moreover, the CPDAG estimated using the hybrid method with the BDe score (Supplementary Figure S12) is also very similar to the OSEM output, with some edges missing and all edges undirected. The drastic difference in the estimated structures between the BDe and OSEM observed in the simulation studies does not recur here. A potential reason could be that the average number of parents in the true network is fewer than one. In this case, one can check via simulations that the BDe score can sometimes provide satisfactory results without the problem of overparameterization. However, we still observe a general improvement in predictive power for OSEM over the BDe approach (Figure 3). Also, the OSEM algorithm can handle networks that are less sparse and hence should be more robust and preferable in practical applications. Supplementary Figure S13 is the CPDAG estimated using the hybrid method with the BGe score. Again, since it treats the data as continuous and uses Gaussian distribution, it looks similar to the first two networks but slightly denser. Filtering out edges that are less certain may improve the learned structure, but the outcome is unlikely to outperform the one from the OSEM algorithm.

Finally, we use heatmaps (Supplementary Figure S14) to visualize and compare the adjacency matrices of the four CPDAGs. The shade in the grid represents the percentage of time a directed edge appears in the 500 Bootstrapped CPDAGs, where an undirected edge counts half for each direction. The darker the shade, the more frequent the corresponding directed edge occurs. There are several aspects in common: first, the symptoms of the two questionnaires form relatively separate clusters in all four networks; second, the symptoms *appetite* and *weight* interact with each other but are isolated from the rest of the symptoms; third, the connection between *sad* and *obdistress* is present in all four structures. Nevertheless, there are also small differences between the network structures which may be important in practice. For example, the link *late – anhedonia* creates a connection between a small subset of symptoms and the largest cluster, a connection which we do not observe in the network estimated by McNally et al. (2017). Since our simulations suggest that the OSEM achieves better performances in structure learning from ordinal data, discovering the connections more accurately may be highly relevant in the application domain.

6. Discussion

In this work, we addressed the problem of learning Bayesian networks from ordinal data by combining the multinomial probit model (Daganzo, 2014) with the Structural EM algorithm of Friedman (1997). The resulting framework is the Ordinal Structural EM (OSEM) algorithm. By assuming that each ordinal variable is obtained by marginally discretizing an underlying Gaussian variable, we can capture the ordinality amongst the categories. By contrast, the commonly used multinomial distribution loses information related to the order of categories due to its invariance with respect to any random permutation of the levels.

Furthermore, we assumed that the hidden variables jointly follow a DAG structure. In other words, we have a Gaussian DAG model in the latent space, allowing us to exploit many of its well-established properties, such as decomposability and score-equivalence.

The location and scale of the original Gaussian distribution are no longer identifiable after discretization. Hence, we chose to standardize each latent dimension to ensure identifiability. This is computationally efficient since we only need to estimate the thresholds once in the process.

Instead of a random initialization, we started with a full DAG and used a pairwise likelihood approach for estimating the initial correlation matrix. This method reduces the number of EM iterations required for convergence. We also derived closed-form formulas for both the structure and parameter updates. In the parameter update, we can perform regression with subset selection following the topological order of a DAG, which induces little cost. Since our expected scoring function is decomposable, score-equivalent, and consistent, we can perform the structure update efficiently by using existing search schemes, such as the GES, MCMC samplers, and hybrid approaches. Unlike the reversible jump MCMC method of Webb and Forster (2008), this additional flexibility in choosing search methods allows us to further reduce the computational burden.

In the simulation studies, we compared the OSEM algorithm to other existing approaches, including three PC-based methods, two hybrid methods with the BDe and the BGe scores respectively, and several methods for mixed data. Under all configurations we tested, our method significantly improved the accuracy in recovering the hidden DAG structure. Using real ordinal data sets, we also showed the generally superior predictive performance of the OSEM algorithm over other score-based approaches, highlighting the usefulness of our modelling assumptions in real applications. In addition, we demonstrated the practicality of our method by applying it to psychological survey data.

To our surprise, the performance of the BDe score in structure recovery was much poorer than our method. Even though it is widely recognized that ignoring the ordering can lead to loss of information, we are, to the best of our knowledge, the first to quantify and visualize its impact on structural inference. The resulting deviance from the true structure was much larger than expected. Therefore, we should in general avoid using nominal methods to tackle problems involving ordinal data.

The BGe score, on the other hand, performed better than we had thought. The correlation matrix estimated directly from the ordinal data appears to be sufficient for the BGe score to identify many of the true edges, especially when the number of ordinal levels is large, so that the continuous approximation improves. However, this method can hardly go beyond the OSEM algorithm, since it still disregards the discreteness of the data.

The most dominating factor for successful structure learning is the quality of the contingency tables, which is determined by the sample size, the number of levels in each ordinal variable, as well as the position of the thresholds. First, the sample size is tightly associated with the accuracy of each cell probability, which decides to what extent the original correlation matrix can be recovered. Second, the number and position of the thresholds control the resolution of the contingency tables. In order to obtain a more reliable result, it is recommended to have more than two levels for each dimension, and the weight should not concentrate on one end of the tables. Otherwise, one may reconsider the effectiveness

of the data collected, such as whether a survey question is well-designed or whether the target population selected is appropriate.

The PC algorithms have relatively limited accuracy in estimating the latent structures. It could be interesting to see if a more suitable test for ordinal data can be developed, which is in itself a challenging task. A recent method proposed by Liu et al. (2020) may be a direction to investigate further. Furthermore, a more accurate output by the PC algorithm can also refine the initial search space for a hybrid search scheme, such as the method of Kuipers et al. (2018), which should in principle bring further improvements to our OSEM algorithm.

Because of the EM iterations, the runtime of our method is unavoidably longer than a pure score-based or hybrid approach, especially when the network size gets larger. The main bottleneck lies in the sampling from the truncated multivariate normal distribution, which is required in every iteration of the process. A possible direction for improvement is a Sequential Monte Carlo sampler (Moffa and Kuipers, 2014), where the samples can be updated from one iteration to the next instead of resampling from scratch.

Another possible direction is to create a Bayesian version of the model. One can replace the BIC penalty with the well-known Wishart prior (Geiger and Heckerman, 2002) and adapt the Bayesian Structural EM algorithm of Friedman (1998). Moreover, Bayesian model averaging (Madigan and Raftery, 1994; Friedman and Koller, 2003) may help with the problem of reaching a local optimum. Specifically, at each iteration, one can sample a collection of structures instead of checking only the highest-scoring graph. A sequential Monte Carlo method can also be applied to the structure level to overcome the computational bottleneck.

Under our current model specification, it is natural to relate to the problem of Bayesian network learning with mixed data. In particular, one may obtain a data set with both continuous and ordinal variables by first generating a Gaussian data set according to a DAG structure and then discretizing some of the variables while keeping others continuous. A similar learning framework may be applicable in this setting. The situation, however, can become much more sophisticated when one wants to include nominal categorical variables. It would therefore be interesting to extend OSEM to mixed continuous and categorical data and compare with other existing approaches (Cui et al., 2016; Tsagris et al., 2018; Talvitie et al., 2019).

Acknowledgments

The authors would like to thank Niko Beerenwinkel for helpful comments and discussions and for hosting Xiang Ge Luo during her Master’s Thesis. We also wish to thank the two anonymous reviewers and the editor for their careful reviews and constructive feedback.

Appendix A. The OSEM Algorithm

Algorithm 1: Ordinal Structural EM (OSEM)

initialization:

Let $\mathcal{G}^{(0)}$ be a full graph;

Estimate thresholds $\hat{\alpha}$: $\hat{\alpha}(i, l) = \Phi^{-1}\left(\frac{1}{N} \sum_{j=1}^N \mathbb{1}(x_i^j \leq \tau(i, l))\right)$, $l = 1, \dots, L_i - 1$;

Estimate correlation matrix $\Sigma^{(0)}$ such that for all $i, j \in \{1, \dots, n\}$, $i \neq j$,

$$\rho_{ij}^{(0)} \mid \hat{\alpha} = \operatorname{argmax}_{\rho_{ij} \in (-1, 1)} \log p\left(\mathcal{D}_{X_i \cup X_j} \mid \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho_{ij} \\ \rho_{ij} & 1 \end{bmatrix}, \mathcal{G}^{(0)}\right);$$

loop: $t = 0, 1, 2, \dots$

Sample K hidden data points for each observation \mathbf{x}^j from the distribution

$$\mathbf{Y}^j \mid \mathbf{x}^j, \hat{\alpha}, \Sigma^{(t)}, \mathcal{G}^{(t)} \sim \operatorname{TMN}(\mathbf{0}, \Sigma^{(t)} \mid \mathbf{x}^j, \hat{\alpha});$$

Structure update:

Compute expected covariance matrix

$$\hat{\Sigma} = \frac{1}{N} \frac{1}{K} \sum_{j=1}^N \sum_{k=1}^K (\mathbf{y}^{j(k)}) (\mathbf{y}^{j(k)})^\top;$$

Update the DAG structure

$$\mathcal{G}^{(t+1)} = \operatorname{argmax}_{\mathcal{G}: \text{DAG over } (\mathbf{X}, \mathbf{Y})} \sum_{i=1}^n \left(-\frac{N}{2} \left[\log \left(\hat{\Sigma}_{i,i} - \hat{\Sigma}_{i,pa(i)} \hat{\Sigma}_{pa(i),pa(i)}^{-1} \hat{\Sigma}_{pa(i),i} \right) \right] - \frac{\log N}{2} \dim(\mathbf{Y}_i, \mathbf{Y}_{pa(i)}) \right);$$

Parameter update:

Following the topological order in $\mathcal{G}^{(t+1)}$, update the parameters using linear regression with best subset selection and the BIC penalty:

$$\mathbf{b}_i^{(t+1)} = \left(\sum_{k=1}^K \left(\mathbf{Y}_{pa(i)}^{(k)} \right)^\top \mathbf{Y}_{pa(i)}^{(k)} \right)^{-1} \left(\sum_{k=1}^K \left(\mathbf{Y}_{pa(i)}^{(k)} \right)^\top \mathbf{y}_i^{(k)} \right);$$

$$v_i^{(t+1)} = \frac{1}{N} \frac{1}{K} \sum_{j=1}^N \sum_{k=1}^K \left(y_i^{j(k)} - (\mathbf{b}_i^{(t+1)})^\top \mathbf{y}_{pa(i)}^{j(k)} \right)^2;$$

Transform $\{\mathbf{b}_i^{(t+1)}, v_i^{(t+1)}\}_{i=1}^n$ into the correlation matrix $\Sigma^{(t+1)}$;

until convergence;

A.1 Threshold Estimation

We only need to estimate the thresholds once at the beginning of the EM iterations, because the discretization is performed marginally, and the marginal mean and variance are always fixed. Consider the i^{th} dimension. By the law of large numbers and the law of iterated expectation, for all $l = 1, \dots, L_i$,

$$\begin{aligned} \frac{1}{N} \sum_{j=1}^N \mathbf{1}(X_i^j = \tau(i, l)) &\rightarrow \mathbb{E}[\mathbf{1}(X_i = \tau(i, l))] \\ &= \mathbb{E}[\mathbb{E}[\mathbf{1}(X_i = \tau(i, l)) \mid Y_i]] \\ &= \mathbb{E}[P(X_i = \tau(i, l) \mid Y_i, \boldsymbol{\alpha}_i)] \\ &= \mathbb{E}[\mathbf{1}(Y_i \in [\alpha(i, l-1), \alpha(i, l)])] \\ &= P(Y_i \in [\alpha(i, l-1), \alpha(i, l)]), \end{aligned}$$

as $N \rightarrow \infty$. It follows that

$$\frac{1}{N} \sum_{j=1}^N \mathbf{1}(X_i^j \leq \tau(i, l)) \rightarrow P(Y_i < \alpha(i, l)) = \Phi(\alpha(i, l)), \quad N \rightarrow \infty.$$

Thus, given a sample $\mathcal{D}_{\mathbf{X}} = \{\mathbf{x}^1, \dots, \mathbf{x}^N\}$, we can estimate each threshold $\alpha(i, l)$ as

$$\hat{\alpha}(i, l) = \Phi^{-1}\left(\frac{1}{N} \sum_{j=1}^N \mathbf{1}(x_i^j \leq \tau(i, l))\right), \quad l = 1, \dots, L_i - 1.$$

They are simply the empirical quantiles of the standard normal distribution.

A.2 Score Used in the Structure Update

Let $P(i) := pa(i)$ and $O(i) := pa(i) \cup \{i\}$ for all $i \in \{1, \dots, n\}$. From Eq. (2), the joint probability of \mathbf{x} and \mathbf{y} can be written as

$$\begin{aligned} p(\mathbf{x}, \mathbf{y} \mid \theta, \mathcal{G}) &= \prod_{i=1}^n \phi(y_i \mid \mathbf{y}_{P(i)}, \vartheta_i, \mathcal{G}) p(x_i \mid y_i, \boldsymbol{\alpha}_i) \\ &= \prod_{i=1}^n \frac{\phi(\mathbf{y}_{O(i)} \mid (\boldsymbol{\mu}, \boldsymbol{\Sigma})_{O(i)})}{\phi(\mathbf{y}_{P(i)} \mid (\boldsymbol{\mu}, \boldsymbol{\Sigma})_{P(i)})} p(x_i \mid y_i, \boldsymbol{\alpha}_i), \end{aligned}$$

where $(\boldsymbol{\mu}, \boldsymbol{\Sigma})_{O(i)}$ is the set of parameters indexed by $O(i)$. The Q function becomes

$$\begin{aligned} Q(\mathcal{G}, \theta; \mathcal{G}^{(t)}, \theta^{(t)}) &= \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j \mid \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log p(\mathbf{X}^j, \mathbf{Y}^j \mid \mathcal{G}, \theta) \mid \mathbf{X}^j = \mathbf{x}^j \right] \\ &= \sum_{j=1}^N \sum_{i=1}^n \mathbb{E}_{\mathbf{Y}^j \mid \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log \frac{\phi(\mathbf{Y}_{O(i)}^j \mid (\boldsymbol{\mu}, \boldsymbol{\Sigma})_{O(i)})}{\phi(\mathbf{Y}_{P(i)}^j \mid (\boldsymbol{\mu}, \boldsymbol{\Sigma})_{P(i)})} p(X_i^j \mid Y_i^j, \boldsymbol{\alpha}_i) \mid X_i^j = x_i^j, \mathcal{G} \right] \\ &= \sum_{i=1}^n \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j \mid \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log \phi(\mathbf{Y}_{O(i)}^j \mid (\boldsymbol{\mu}, \boldsymbol{\Sigma})_{O(i)}) p(X_i^j \mid Y_i^j, \boldsymbol{\alpha}_i) \mid X_i^j = x_i^j, \mathcal{G} \right] \\ &\quad - \mathbb{E}_{\mathbf{Y}^j \mid \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log \phi(\mathbf{Y}_{P(i)}^j \mid (\boldsymbol{\mu}, \boldsymbol{\Sigma})_{P(i)}) \mid \mathcal{G} \right]. \end{aligned}$$

Using the multivariate normal density ϕ and the truncated multivariate normal density φ , we can expand the summands,

$$\begin{aligned}
 & \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log \phi(\mathbf{Y}_{O(i)}^j | (\boldsymbol{\mu}, \Sigma)_{O(i)}) p(X_i^j | Y_i^j, \boldsymbol{\alpha}_i) | X_i^j = x_i^j, \mathcal{G} \right] \\
 &= \int \left(\log \phi(\mathbf{y}_{O(i)}^j | (\boldsymbol{\mu}, \Sigma)_{O(i)}) \right) \varphi(\mathbf{y}^j | \theta^{(t)}, \mathcal{G}^{(t)}, \boldsymbol{\alpha}) d\mathbf{y}^j \\
 &= -\frac{1}{2} \int \left[(|P(i)| + 1) \log(2\pi) + \log(|\Sigma_{O(i), O(i)}|) \right. \\
 &\quad \left. + \text{Tr} \left((\mathbf{y}_{O(i)}^j - \boldsymbol{\mu}_{O(i)}) (\mathbf{y}_{O(i)}^j - \boldsymbol{\mu}_{O(i)})^\top \Sigma_{O(i), O(i)}^{-1} \right) \right] \varphi(\mathbf{y}^j | \theta^{(t)}, \mathcal{G}^{(t)}, \boldsymbol{\alpha}) d\mathbf{y}^j \\
 &= -\frac{1}{2} \left[(|P(i)| + 1) \log(2\pi) + \log(|\Sigma_{O(i), O(i)}|) \right. \\
 &\quad \left. + \text{Tr} \left(\mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} [(\mathbf{Y}_{O(i)}^j - \boldsymbol{\mu}_{O(i)}) (\mathbf{Y}_{O(i)}^j - \boldsymbol{\mu}_{O(i)})^\top] \Sigma_{O(i), O(i)}^{-1} \right) \right].
 \end{aligned}$$

Analogously,

$$\begin{aligned}
 & \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log \phi(\mathbf{Y}_{P(i)}^j | (\boldsymbol{\mu}, \Sigma)_{P(i)}) | \mathcal{G} \right] = -\frac{1}{2} \left[|P(i)| \log(2\pi) + \log(|\Sigma_{P(i), P(i)}|) \right. \\
 &\quad \left. + \text{Tr} \left(\mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} [(\mathbf{Y}_{P(i)}^j - \boldsymbol{\mu}_{P(i)}) (\mathbf{Y}_{P(i)}^j - \boldsymbol{\mu}_{P(i)})^\top] \Sigma_{P(i), P(i)}^{-1} \right) \right].
 \end{aligned}$$

By computing the expected statistics

$$\hat{\boldsymbol{\mu}} = \frac{1}{N} \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} [\mathbf{Y}^j]$$

and

$$\hat{\Sigma} = \frac{1}{N} \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} [(\mathbf{Y}^j - \hat{\boldsymbol{\mu}}) (\mathbf{Y}^j - \hat{\boldsymbol{\mu}})^\top],$$

we get

$$Q(\mathcal{G}, \hat{\boldsymbol{\theta}}; \mathcal{G}^{(t)}, \theta^{(t)}) = \sum_{i=1}^n -\frac{N}{2} \left[\log(2\pi) + \log \left(\frac{|\hat{\Sigma}_{O(i), O(i)}|}{|\hat{\Sigma}_{P(i), P(i)}|} \right) + 1 \right].$$

Notice that $|\hat{\Sigma}_{O(i), O(i)}|$ can be partitioned as

$$\hat{\Sigma}_{O(i), O(i)} = \begin{bmatrix} \hat{\Sigma}_{i, i} & \hat{\Sigma}_{i, P(i)} \\ \hat{\Sigma}_{P(i), i} & \hat{\Sigma}_{P(i), P(i)} \end{bmatrix}$$

with

$$|\hat{\Sigma}_{O(i), O(i)}| = |\hat{\Sigma}_{P(i), P(i)}| (\hat{\Sigma}_{i, i} - \hat{\Sigma}_{i, P(i)} \hat{\Sigma}_{P(i), P(i)}^{-1} \hat{\Sigma}_{P(i), i}).$$

Therefore,

$$Q(\mathcal{G}, \hat{\boldsymbol{\theta}}; \mathcal{G}^{(t)}, \theta^{(t)}) = \sum_{i=1}^n -\frac{N}{2} \left[\log \left(\hat{\Sigma}_{i, i} - \hat{\Sigma}_{i, P(i)} \hat{\Sigma}_{P(i), P(i)}^{-1} \hat{\Sigma}_{P(i), i} \right) + \log(2\pi) + 1 \right].$$

The formulas used in Algorithm 1 are after the adjustment for the identifiability constraints.

A.3 MLE in the Parameter Update

By substituting Eq. (2) into $Q(\mathcal{G}, \theta; \mathcal{G}^{(t)}, \theta^{(t)})$, we get

$$\begin{aligned}
 Q(\mathcal{G}, \theta; \mathcal{G}^{(t)}, \theta^{(t)}) &= \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log p(\mathbf{X}^j, \mathbf{Y}^j | \mathcal{G}, \theta) | \mathbf{X}^j = \mathbf{x}^j \right] \\
 &= \sum_{j=1}^N \sum_{i=1}^n \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log \phi(Y_i^j | \mathbf{Y}_{pa(i)}^j, \theta_i, \mathcal{G}) p(X_i^j | Y_i^j, \boldsymbol{\alpha}_i) | X_i^j = x_i^j \right] \\
 &= \sum_{i=1}^n \sum_{j=1}^N \int \left(\log \phi(y_i^j | \mathbf{y}_{pa(i)}^j, \theta_i, \mathcal{G}) \right) \varphi(\mathbf{y}^j | \theta^{(t)}, \mathcal{G}^{(t)}, \boldsymbol{\alpha}) d\mathbf{y}^j \\
 &= \sum_{i=1}^n \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[\log \phi(Y_i^j | \mathbf{Y}_{pa(i)}^j, \theta_i, \mathcal{G}) \right].
 \end{aligned}$$

Conditioned on $\theta_{pa(i)}^{(t+1)}$, let $\boldsymbol{\beta}_i = (\mu_i, \mathbf{b}_i^\top)^\top$ and $\tilde{\mathbf{Y}}_{pa(i)}^j = (1, (\mathbf{Y}_{pa(i)}^j - \boldsymbol{\mu}_{pa(i)}^{(t+1)})^\top)^\top$. Invoking the formula for normal distributions,

$$\begin{aligned}
 Q(\mathcal{G}^{(t+1)}, \theta; \mathcal{G}^{(t)}, \theta^{(t)}) &= \sum_{i=1}^n \sum_{j=1}^N \int \left[-\frac{1}{2} \log(2\pi v_i) - \frac{1}{2v_i} \left(y_i^j - \mu_i - \mathbf{b}_i^\top (\mathbf{y}_{pa(i)}^j - \boldsymbol{\mu}_{pa(i)}) \right)^2 \right] \\
 &\quad \times \varphi(\mathbf{y}^j | \theta^{(t)}, \mathcal{G}^{(t)}, \boldsymbol{\alpha}) d\mathbf{y}^j \\
 &= \sum_{i=1}^n -\frac{N}{2} \left(\log(2\pi v_i) + \frac{1}{v_i} \frac{1}{N} \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[(Y_i^j - \mu_i - \mathbf{b}_i^\top (\mathbf{Y}_{pa(i)}^j - \boldsymbol{\mu}_{pa(i)})) \right]^2 \right) \\
 &= \sum_{i=1}^n -\frac{N}{2} \left(\log(2\pi v_i) + \frac{1}{v_i} \frac{1}{N} \sum_{j=1}^N \mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[(Y_i^j - \boldsymbol{\beta}_i^\top \tilde{\mathbf{Y}}_{pa(i)}^j)^2 \right] \right).
 \end{aligned}$$

The expectations can be approximated with Monte Carlo samples of size K ,

$$\mathbb{E}_{\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}} \left[(Y_i^j - \boldsymbol{\beta}_i^\top \tilde{\mathbf{Y}}_{pa(i)}^j)^2 \right] \approx \frac{1}{K} \sum_{k=1}^K \left[(y_i^{j(k)} - \boldsymbol{\beta}_i^\top \tilde{\mathbf{y}}_{pa(i)}^{j(k)})^2 \right],$$

where $\mathbf{y}^{j(k)}$ is drawn from the conditional distributions $\mathbf{Y}^j | \mathbf{x}^j, \mathcal{G}^{(t)}, \theta^{(t)}$, $j = 1, \dots, N$. Next, we differentiate $Q(\mathcal{G}^{(t+1)}, \theta; \mathcal{G}^{(t)}, \theta^{(t)})$ with respect to $\boldsymbol{\beta}_i$ and set to zero,

$$\begin{aligned}
 \frac{\partial}{\partial \boldsymbol{\beta}_i} Q(\mathcal{G}^{(t+1)}, \theta; \mathcal{G}^{(t)}, \theta^{(t)}) &= \frac{\partial}{\partial \boldsymbol{\beta}_i} \frac{1}{v_i} \frac{1}{N} \frac{1}{K} \sum_{j=1}^N \sum_{k=1}^K \left[(y_i^{j(k)} - \boldsymbol{\beta}_i^\top \tilde{\mathbf{y}}_{pa(i)}^{j(k)})^2 \right] \\
 &= -2 \frac{1}{v_i} \frac{1}{N} \frac{1}{K} \sum_{j=1}^N \sum_{k=1}^K \left[(y_i^{j(k)} - \boldsymbol{\beta}_i^\top \tilde{\mathbf{y}}_{pa(i)}^{j(k)}) \tilde{\mathbf{y}}_{pa(i)}^{j(k)} \right] \\
 &\triangleq 0.
 \end{aligned}$$

Thus,

$$\beta_i^{(t+1)} = \left(\sum_{k=1}^K \left(\tilde{\mathbf{Y}}_{pa(i)}^{(k)} \right)^\top \tilde{\mathbf{Y}}_{pa(i)}^{(k)} \right)^{-1} \left(\sum_{k=1}^K \left(\tilde{\mathbf{Y}}_{pa(i)}^{(k)} \right)^\top \mathbf{y}_i^{(k)} \right)$$

with $\tilde{\mathbf{Y}}_{pa(i)}^{(k)} = (\tilde{\mathbf{y}}_{pa(i)}^{1(k)}, \dots, \tilde{\mathbf{y}}_{pa(i)}^{N(k)})^\top \in \mathbb{R}^{N \times (|pa(i)|+1)}$ and $\mathbf{y}_i^{(k)} = (y_i^{1(k)}, \dots, y_i^{N(k)})^\top \in \mathbb{R}^N$. Conditioned on $\beta_i^{(t+1)}$, we then differentiate $Q(\mathcal{G}^{(t+1)}, \theta; \mathcal{G}^{(t)}, \theta^{(t)})$ with respect to v_i ,

$$\frac{\partial}{\partial v_i} Q(\mathcal{G}^{(t+1)}, \theta; \mathcal{G}^{(t)}, \theta^{(t)}) = \frac{1}{v_i} - \frac{1}{v_i^2} \frac{1}{N} \frac{1}{K} \sum_{j=1}^N \sum_{k=1}^K \left[(y_i^{j(k)} - \beta_i^\top \tilde{\mathbf{y}}_{pa(i)}^{j(k)})^2 \right] \triangleq 0.$$

Therefore, the update for v_i is

$$v_i^{(t+1)} = \frac{1}{N} \frac{1}{K} \sum_{j=1}^N \sum_{k=1}^K \left[(y_i^{j(k)} - (\beta_i^{(t+1)})^\top \tilde{\mathbf{y}}_{pa(i)}^{j(k)})^2 \right].$$

Again, the formulas used in Algorithm 1 are adjusted for the identifiability constraints.

Appendix B. Additional Experimental Results

B.1 ROC Curves for Skeletons

If we plot the ROC curves for skeletons, ignoring the v-structure differences, we can see in Figure S1 that all methods show substantial improvement, and the OSEM algorithm still performs the best. In particular, the method with the BDe score, the GPC and the OPC algorithms have more significant increase in TPR from the one in Figure 2, which implies that these methods tend to inaccurately estimate the v-structures.

B.2 Effect of the Thresholds

Here we focus on the effects of thresholds on the model performance by changing the expected number of levels and the symmetric Dirichlet concentration parameter. We pick networks with 20 nodes, 4 expected neighbours, and the associated data sets of size 500.

In symmetric Dirichlet distributions $\text{Dir}(L_i, \nu)$, the parameter ν controls the density concentration in the $(L_i - 1)$ -dimensional probability simplex, which is illustrated in Figure S2. By simulating from $\text{Dir}(L_i, \nu)$ with L_i fixed and $\nu < 1$, it is more likely to obtain a contingency table where a few cells have much higher frequencies than the others. When most mass concentrates on one end of the table, recovering the original DAGs becomes more difficult, as most correlation information is lost. If $\nu = 1$, it means that we sample uniformly from the simplex. Lastly, if $\nu > 1$, the cell probabilities will be more evenly distributed.

We consider four expected values $\{2, 3, 4, 5\}$ for L_i (write $\mathbb{E}[L_i]$, $i = 1, \dots, n$) and three values $\{0.8, 1, 2\}$ for ν . Specifically, we assume that each L_i follows independently a discrete uniform distribution on the interval $[2, 2\mathbb{E}[L_i] - 2]$. When $\mathbb{E}[L_i] = 2$, all variables are binary.

The ROC curves in Figure S3 again show that our proposed method is superior to all alternatives in every case. With ν fixed, increasing $\mathbb{E}[L_i]$ significantly improves the performance of both the OSEM algorithm and the method with the BGe score. This is

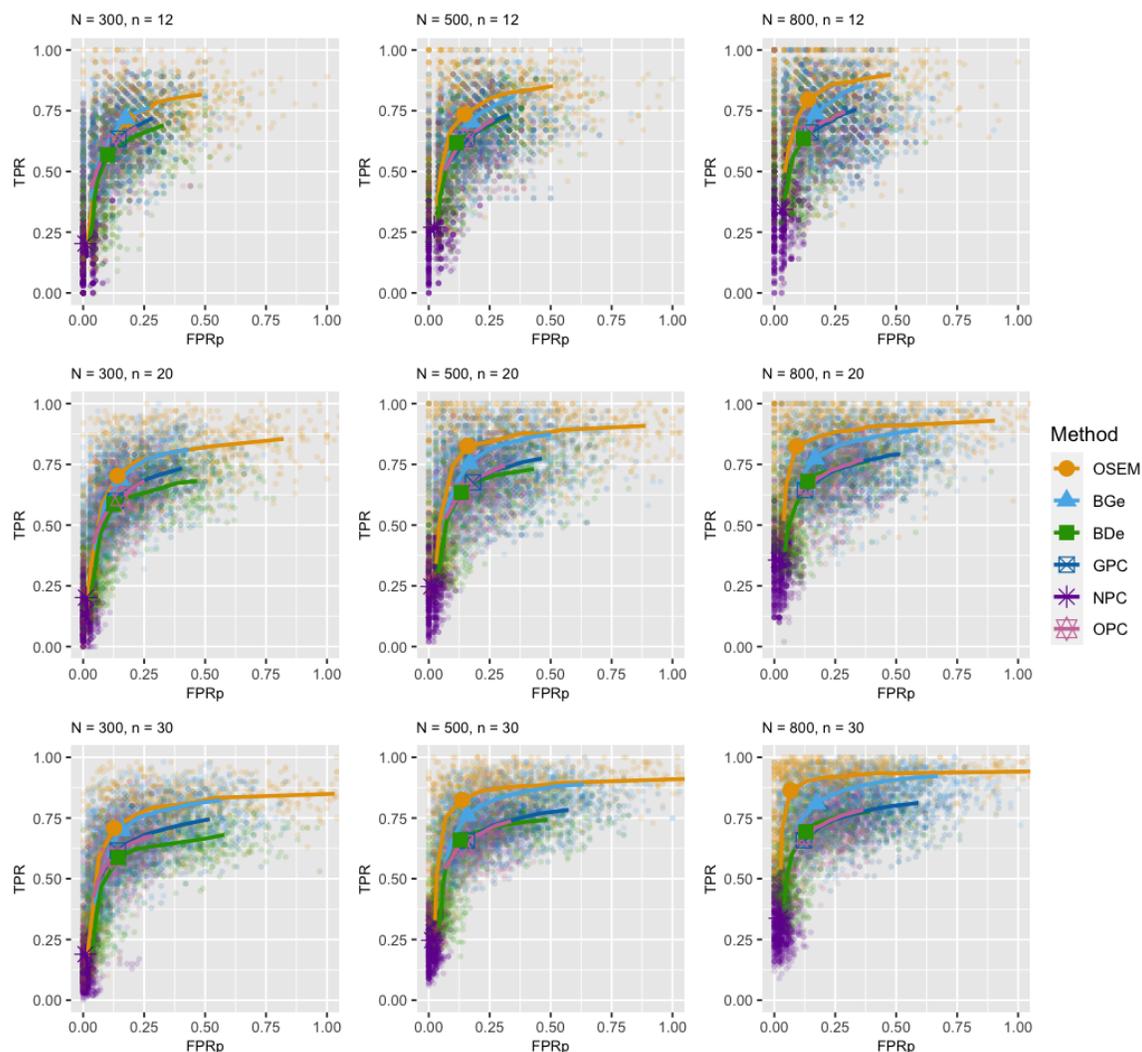


Figure S1: The same plot as Figure 2 but the TPR and FPR_p are computed using the skeletons instead of the patterns.

because the resolution of the contingency tables becomes higher. When the discretization is thin enough, the ordinal data should mimic the continuous data on a different scale, and we expect these two curves to move towards the upper left corner of the plot. For similar reasons, the GPC algorithm also slightly improves. Moreover, the rise of the curves gradually becomes slower, which suggests that having more levels is preferable, but too many levels are not necessary for successful structure learning. On the contrary, the performance of the method with the BDe score seems to worsen as $\mathbb{E}[L_i]$ increases, possibly due to the problem of overparameterization. This can be another reason why we should not apply the nominal method on ordinal data.

Conversely, with $\mathbb{E}[L_i]$ fixed, all hybrid methods show various degrees of improvement as ν increases. The OSEM and the BGe enhance the most, because the contingency tables to be learned from are more well-shaped. The results are least reliable when all variables are binary ($\mathbb{E}[L_i] = 2$) and the cell probabilities are skewed ($\nu = 0.8$). If this is the case, it is however unfortunate that none of the alternative methods will work either. Therefore, the number of thresholds and their positions do play an important role in the recovery of DAG structures, and we should not underestimate the importance of the quality of the data.

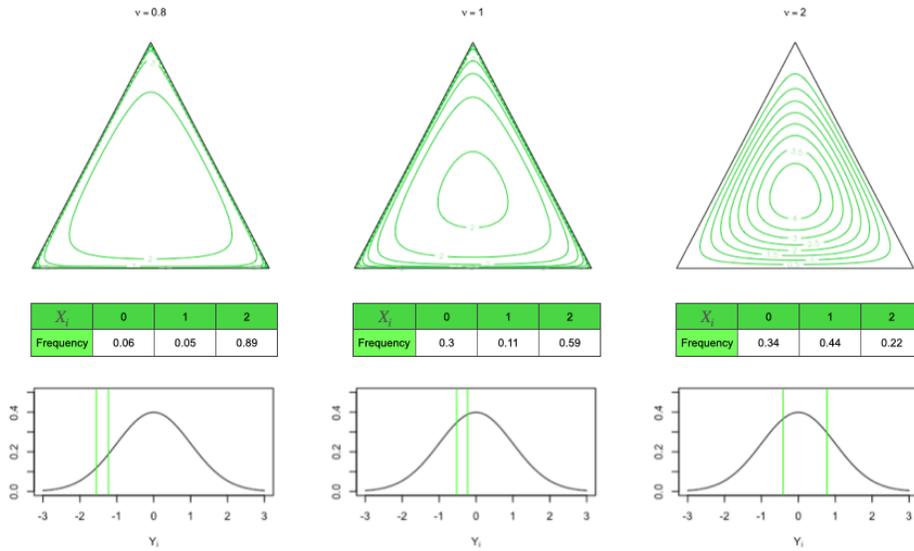


Figure S2: Contour plots of 3-dimensional symmetric Dirichlet distributions $\text{Dir}(3, \nu)$ with ν being 0.8 (left), $1 + 1 \times 10^{-9}$ (middle), and 2 (right) respectively, created using the **Compositional** package (Tsagris and Athineou, 2019). Below each contour plot, an instance of the generated contingency table for the ordinal variable X_i and the corresponding discretization of the Gaussian variable Y_i are shown.

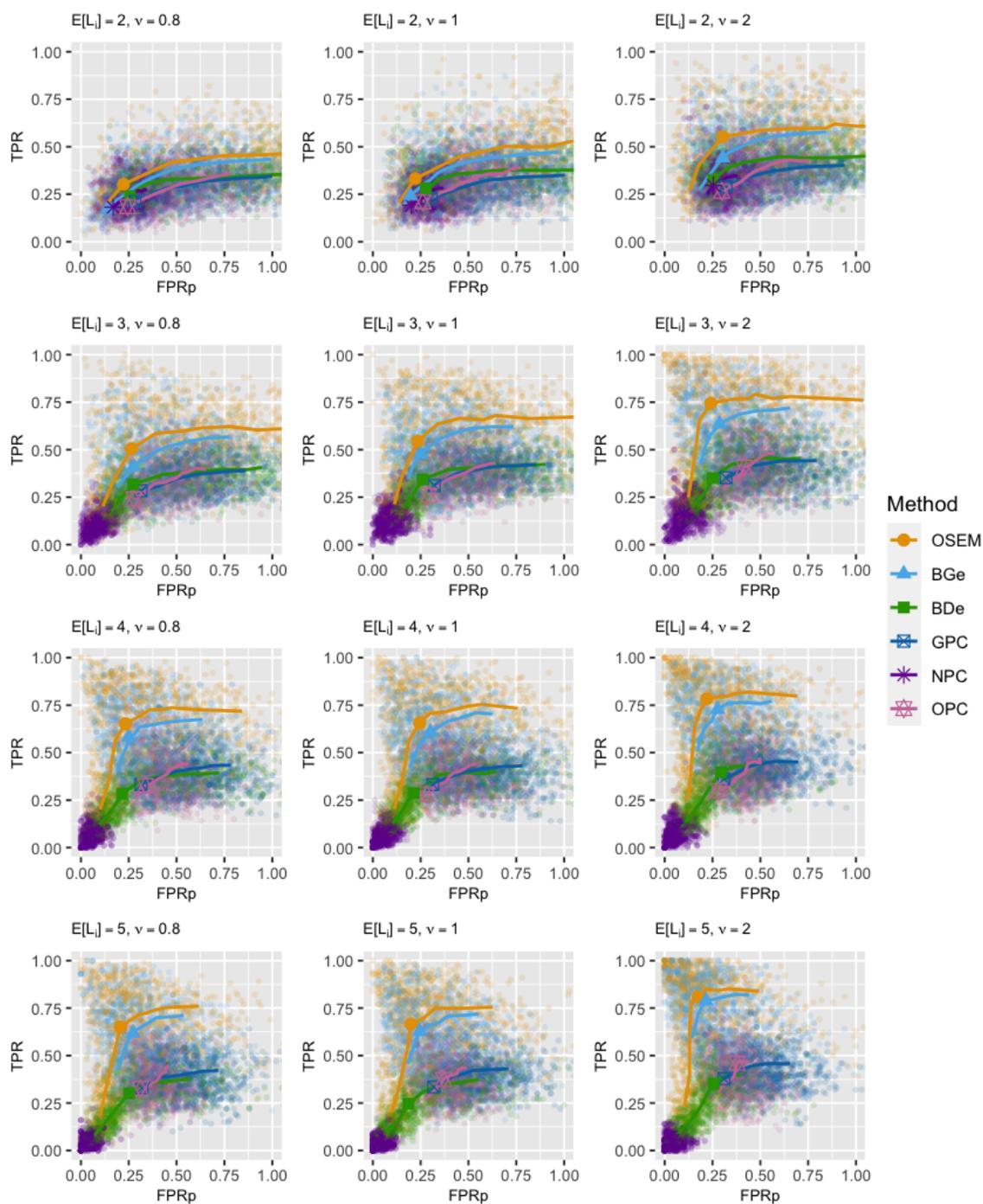


Figure S3: The same plot as Figure 2 but with sample size $N = 500$, network size $n = 20$, expected number of ordinal levels $\mathbb{E}[L_i] \in \{2, 3, 4, 5\}$, and Dirichlet concentration parameter $\nu \in \{0.8, 1, 2\}$.

B.3 Effect of Network Sparsity

With network size $n = 20$, sample size $N = 500$, expected number of ordinal levels $\mathbb{E}[L_i] = 3$ and Dirichlet concentration parameter $\nu = 2$, we also look at the effect of network sparsity on the model performance by varying the expected number of neighbours d per node. In Figure S4, it is clear that the performances of all methods deteriorate as the networks become denser. When $d = 2$, the generated networks are relatively sparse, even the GPC, the OPC and the method with the BDe score can achieve an average TPR of almost 75%, and the gap between the OSEM and the method with the BGe score is small. When $d = 5$, however, the average TPRs of all methods drop below 75%. In all cases, the OSEM algorithm outperforms the alternatives. In general, structure learning for very dense Bayesian networks is a difficult problem.

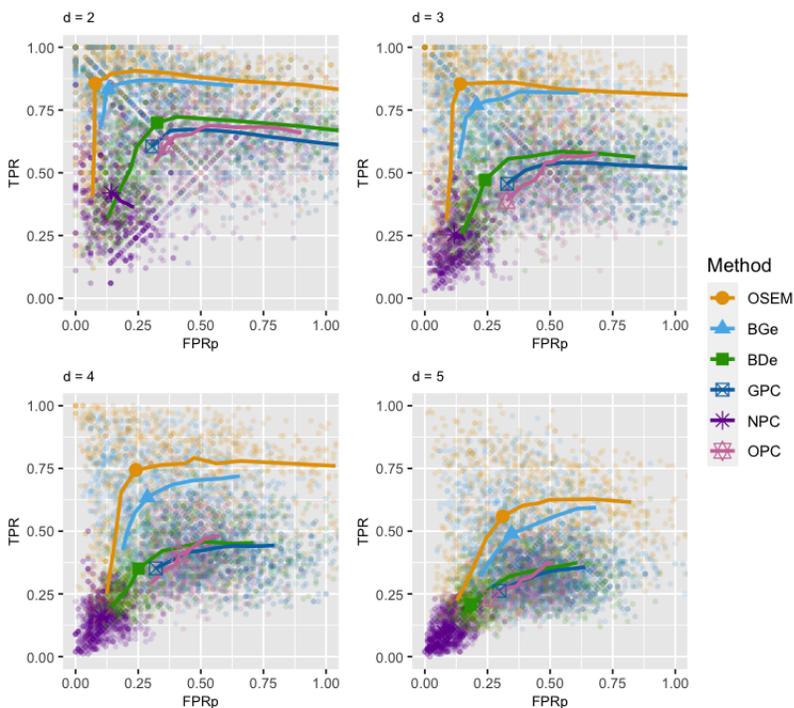


Figure S4: The same plot as Figure 2 but with sample size $N = 500$, network size $n = 20$, expected number of ordinal levels $\mathbb{E}[L_i] = 3$, Dirichlet concentration parameter $\nu = 2$, and expected number of neighbours per node $d \in \{2, 3, 4, 5\}$.

B.4 Runtimes

Due to its iterative nature, the OSEM algorithm is expected to be slower than other methods. To examine its computational cost, we perform a runtime comparison for DAGs with 12, 20, 30 nodes and 500 data points. The expected number of levels is again 3. The runtimes are associated with the cutoff values at the highest sum of average TPR and $(1 - \text{FPRp})$.

Notice that the runtimes are highly dependent on the algorithmic implementation and the software packages chosen (see Appendix B.7). For hybrid approaches, the speed and accuracy of the search schemes dominate the overall computational complexity. The runtimes for constraint-based methods rely on how the tests are coded. Therefore, the comparison in Figure S5 is intended only as a general reference limited to the functions we chose.

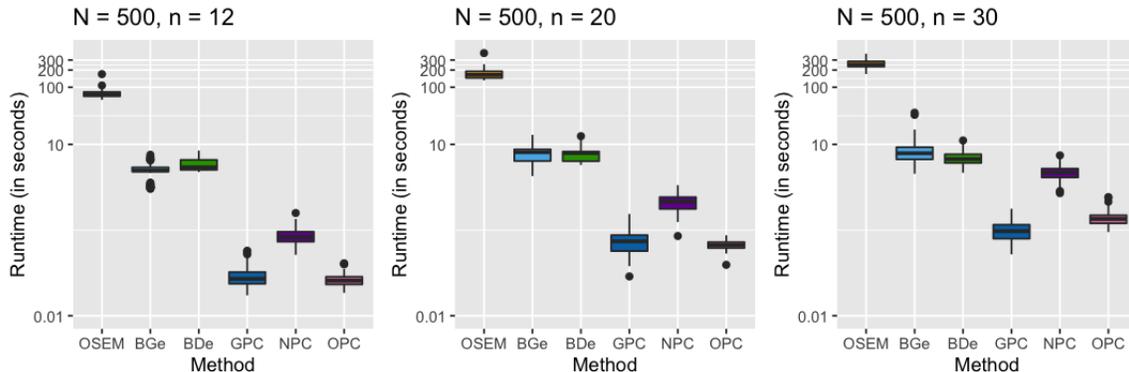


Figure S5: Runtime comparison in recovering the true patterns for random DAGs with 12, 20, 30 nodes and 500 data points. The runtimes are displayed in seconds, and the y-axis is transformed with \log_{10} .

For completeness we present runtimes for the NPC, OPC and GPC algorithms. They are less interesting due to their unsatisfying performance in the recovery of DAGs. On average, both hybrid methods with the BDe and BGe scores take less than 10 seconds to finish for all three network sizes. Our OSEM algorithm, however, takes around 1.5, 2.5 and 3.5 minutes respectively in order to converge, which is sometimes more than 20 times slower than a pure hybrid approach without EM iterations. Such a loss in computational efficiency is mainly due to sampling from the n -dimensional truncated multivariate normal distributions, especially when n is large. Consequently, the current version of our method is only feasible for small-to-medium networks. For larger networks, one may wish to conduct a pre-selection of the variables before using such models for inference.

B.5 Comparison with Existing Mixed Methods

We additionally benchmark the OSEM algorithm against the most recent structure learning methods for mixed (continuous, nominal and ordinal) variables. If we consider only the ordinal data, our OSEM algorithm clearly outperforms the following approaches in simulations (Figure S6):

- The copula PC algorithm of Cui et al. (2016) (CPC); To be consistent with the paper mentioned, we also include the rank PC algorithm of Harris and Drton (2013) (RPC);
- PC algorithm with a symmetric test based on two asymmetric likelihood-ratio tests (Tsagris et al., 2018) (MMPC);
- Two score-based approaches based on classification and regression trees (Talvitie et al., 2019): PCART and OPCART. OPCART is an extension of PCART to accommodate

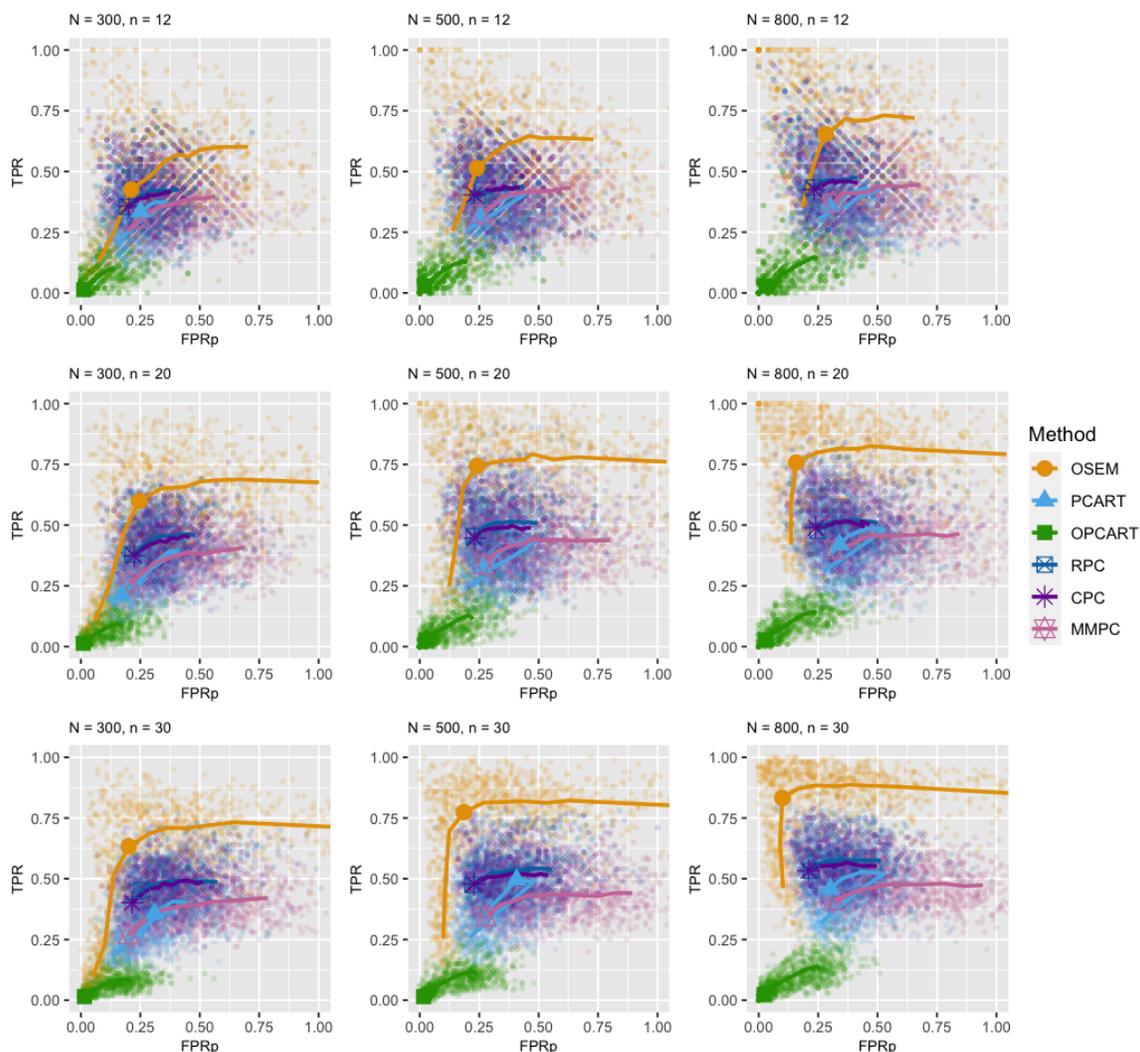


Figure S6: The comparison of the performance in recovering the true pattern between our OSEM algorithm and five other approaches: PCART, OPCART, CPC, RPC, and MMPC. The ROC curves are created as in Figure 2.

ordinal response variables in decision trees. Note that OPCART performs much worse than PCART. With the segmentation prior parameter κ set to $\frac{1}{4}$, as suggested by the authors, the estimated OPCART graph is always very sparse no matter how one changes the other hyperparameters. If one increases κ , then OPCART has similar performance as PCART but requires much more computational resources due to the segmentation step, which takes around an hour to learn one network with 20 nodes. In the section below, we also show that OPCART has worse predictive performance than PCART.

All implementations are summarised in Appendix B.7.

B.6 More about Predictive Performance

In Table 1 one can find all five data sets used to assess the predictive performance of the OSEM algorithm. In the pre-processing step, we have categorized two numerical variables in the Contraceptive Method Choice data set: *wife's age* (0: [0, 20); 1: [20, 30); 2: [30, 40); 3: [40, ∞)) and *number of children ever born* (0: [0, 1]; 1: [2, 3]; 2: [4, 5], 3: [6, ∞)). We have also removed the *class* variable with 22 levels from the Primary Tumor data set.

In cross validation, we first randomly split each data set into 80% training data and 20% test data. Using the training data, we find the optimal model with respect to each of the methods being compared (BDe, BGe, OSEM, PCART/OPCART). Conditioned on the estimated model $\hat{\mathcal{M}}$, we evaluate the discrete (negative) log loss (log-likelihood) function on the test data,

$$\log P(\mathbf{X}_{\text{test}} | \hat{\mathcal{M}}) = \sum_{i=1}^{N_{\text{test}}} \sum_{l=1}^L \mathbf{1}_{\mathbf{x}_{il}} \log \hat{p}_l, \quad (22)$$

where N_{test} is the number of unseen test cases, L is the total number of combinations of all ordinal levels, $\mathbf{1}_{\mathbf{x}_{il}}$ indicates whether the i^{th} data point matches the l^{th} combination, and \hat{p}_l is the probability of observing the l^{th} combination given the model. In particular, we select \hat{p}_l from the probability tables computed for each method being compared:

- BDe: from the BDe posterior predictive distribution (Heckerman and Geiger, 1995);
- PCART/OPCART: from the BDe posterior predictive distribution;
- OSEM: by integrating the multivariate normal distribution with zero mean vector and the optimal correlation matrix from the OSEM output over regions defined by the estimated thresholds $\hat{\alpha}$ from Eq. (9);
- BGe: by integrating the multivariate normal distribution with zero mean vector and the maximum a posteriori correlation matrix based on the estimated DAG (Viinikka et al., 2020) over regions defined by the estimated thresholds $\hat{\alpha}$ from OSEM.

For each method, we repeat the above process 100 times over a range of tuning parameters and obtain Figure S7, where the optimal values are selected and summarised in Figure 3. We also provide an example in Figure S8 to show that OPCART has worse predictive performance than PCART and hence it is omitted from the main text. In addition, we evaluate the log loss with decreasing number of data points by subsampling the data sets (Figure S9). The relative log loss of the BDe approach slightly worsens in comparison to the other three methods, possibly due to overparameterization. As sample size decreases, the uncertainty consistently increases, and the advantage of OSEM over BGe appears less pronounced.

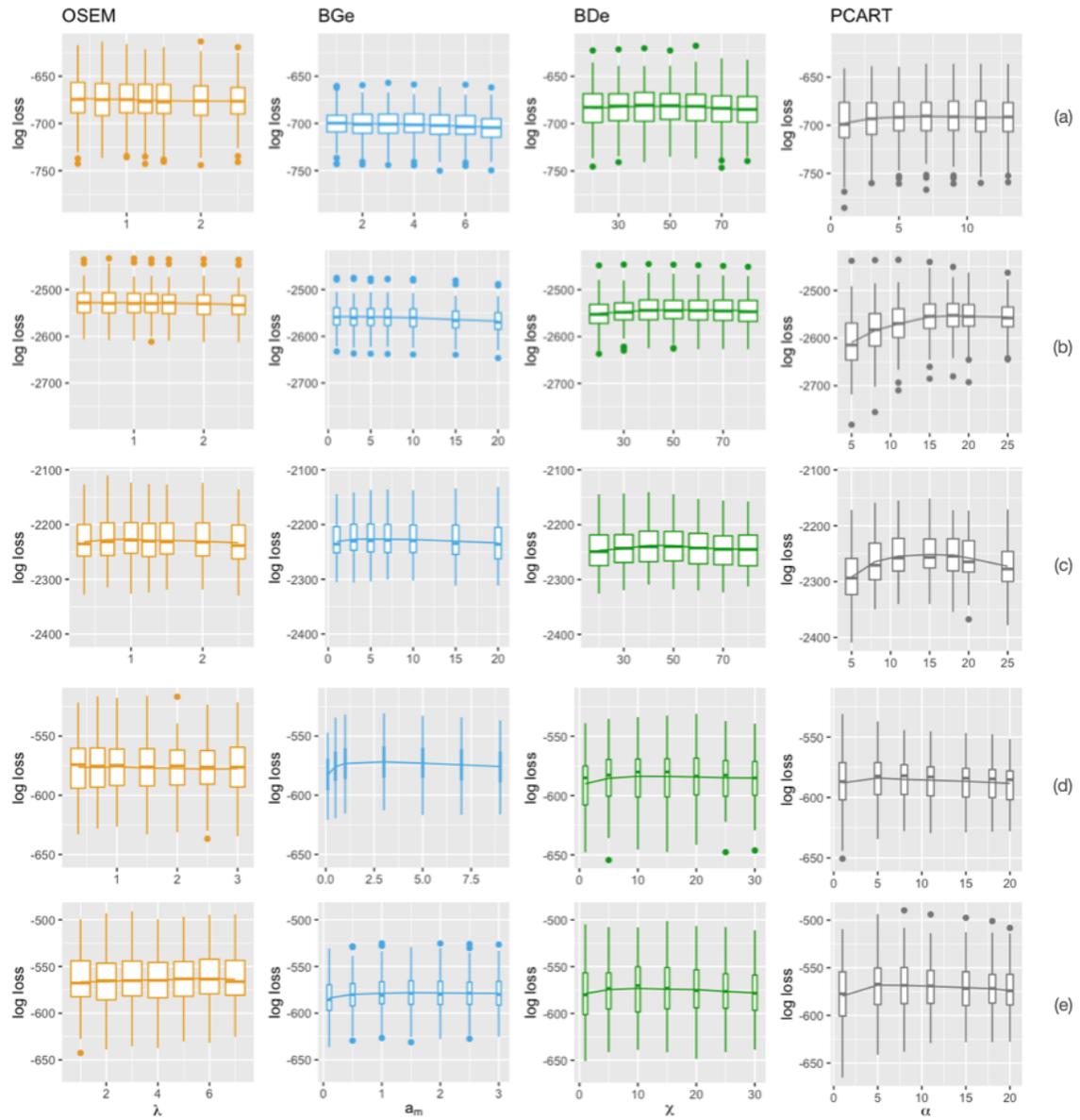


Figure S7: The comparison of log loss for four methods (OSEM, BGe, BDe, PCART) and five data sets: (a) Congressional Voting Records, (b) Contraceptive Method Choice, (c) OCD and Depression, (d) Primary Tumor, and (e) SPECT Heart.

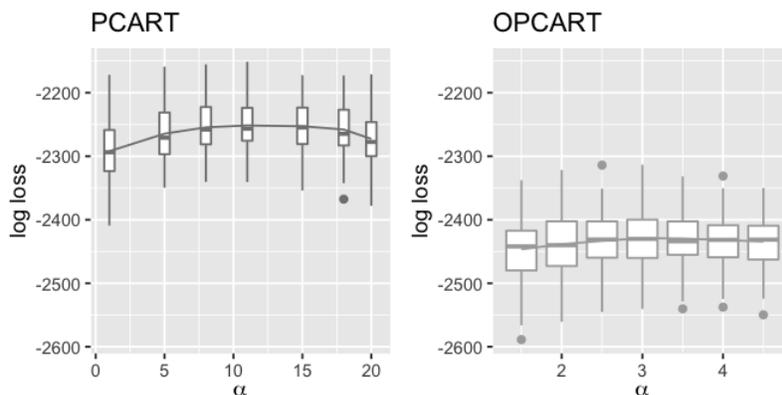


Figure S8: The comparison of log loss between PCART and OPCART for the OCD and Depression data set.

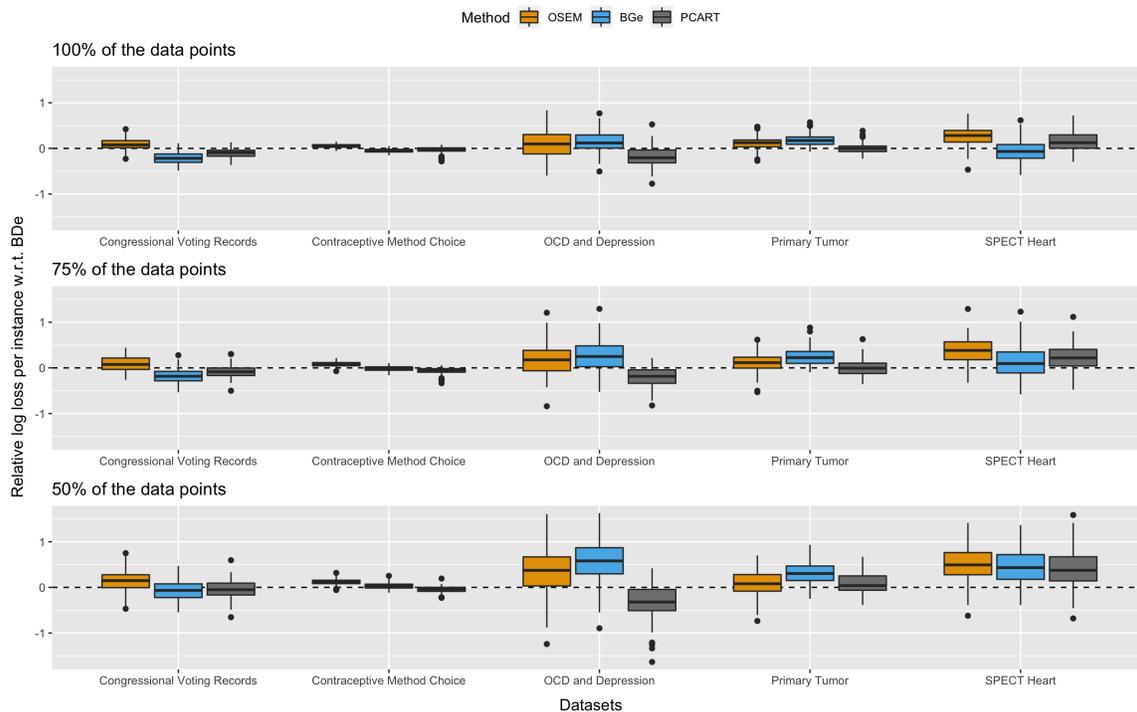


Figure S9: The comparison of log loss with decreasing sample sizes. In each simulation run, we first randomly subsample a proportion of the data points (75% or 50%) from the data sets, followed by structure learning and log loss computation.

As a sanity check, we evaluate the predictive performance using synthetic data sets where we know the ground truth networks. Considering $N = 500$, $n = 20$ and three expected number of levels $\{2, 3, 4\}$, we plot in Figure S10 the log loss against the structural hamming

distance scaled by P as defined in Section 4.1.1. As we expect, the OSEM algorithm dominates the other three methods in terms of both log loss and SHD. As the average number of levels increases, the performance of BGe becomes closer to OSEM, which is consistent with the observations in Appendix B.2. Interestingly, the optimum log loss for OSEM and BGe (solid points) do not match the minimum SHD (the top left corner of the curves), and the log loss is very flat across a wide range of tuning parameters. This may suggest that overfitting in Gaussian-based BNs can be beneficial in terms of predictive power, and the strength of regularization needs to be stronger for optimal structure learning than for prediction.

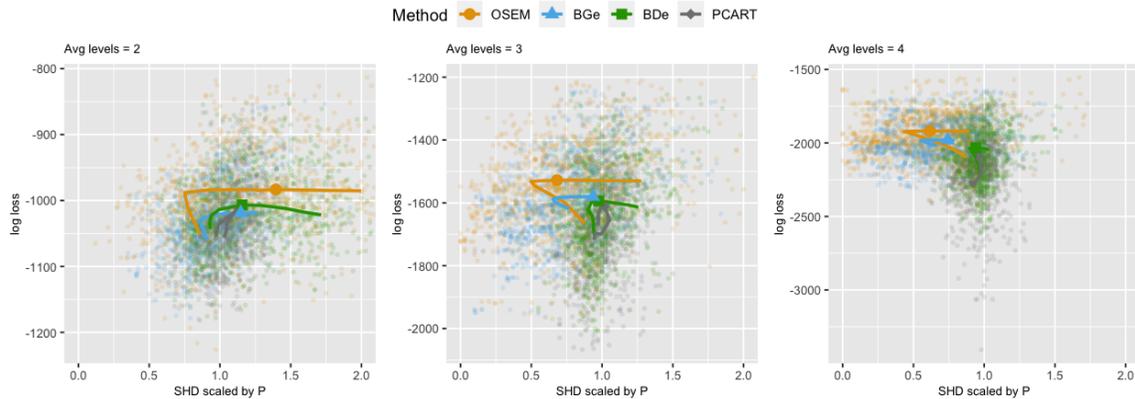


Figure S10: The comparison of log loss against SHD scaled by P for four methods (OSEM, BGe, BDe, PCART) using synthetic data. The lines are created by interpolating the average log loss against the average SHD scaled by P at each penalization value. The solid points correspond to the optimal/maximum log loss for each method.

B.7 R Implementations

| Method | R package | R functions | Reference |
|--------|-----------------------------|--|--|
| NPC | pcalg | pc, disCItest | Neapolitan et al. (2004); Kalisch et al. (2012) |
| OPC | bnlearn | pc.stable(..., test = "jt") | Scutari (2009); Musella (2013) |
| GPC | pcalg | pc, gaussCItest | Kalisch et al. (2012) |
| BDe | BiDAG | scoreparameters("bdecat",...), iterativeMCMC | Heckerman and Geiger (1995); Suter et al. (2021) |
| BGe | BiDAG | scoreparameters("bge",...), iterativeMCMC | Heckerman and Geiger (1995); Suter et al. (2021) |
| OSEM | BiDAG | iterativeMCMC, OSEM ¹ | Suter et al. (2021) |
| RPC | pcalg | pc, gaussCItest, cor(..., method = "kendall") | Kalisch et al. (2012); Harris and Drton (2013) |
| CPC | pcalg | pc, gaussCItest, inferCopulaModel ² | Kalisch et al. (2012); Cui et al. (2016) |
| MMPC | MXM | pc.skel(..., method = "comb.mm"), pc.or | Tsagris et al. (2018) |
| PCART | rpcart ³ , BiDAG | opt.pcart.cat, iterativeMCMC | Talvitie et al. (2019); Suter et al. (2021) |
| OPCART | rpcart ⁴ , BiDAG | opt.pcart, iterativeMCMC | Talvitie et al. (2019); Suter et al. (2021) |

Table S1: R implementations for simulation studies. The first six methods are discussed in Section 4, and the last five methods can be found in Appendix B.5.

1. <https://github.com/xgluo/OSEM>

2. <https://github.com/cuiruifei/CausalMissingValues>

3. <https://github.com/ttalvitie/pcart>

4. <https://github.com/xgluo/pcart>. Note that the original `rpcart` package does not have an R implementation for OPCART.

Appendix C. Additional Table and Figures

| Symptoms and the associated abbreviations | | | |
|---|-------------------|-------------------------------|------------------|
| Y-BOCS-SR | | QIDS-SR | |
| Time consumed by obsessions | <i>obtime</i> | Sleep-onset insomnia | <i>onset</i> |
| Interference due to obsessions | <i>obinterfer</i> | Middle insomnia | <i>middle</i> |
| Distress caused by obsessions | <i>obdistress</i> | Early morning awakening | <i>late</i> |
| Difficulty resisting obsessions | <i>obresist</i> | Hypersomnia | <i>hypersom</i> |
| Difficulty controlling obsessions | <i>obcontrol</i> | Sadness | <i>sad</i> |
| Time consumed by compulsions | <i>comptime</i> | Decreased/Increased appetite | <i>appetite</i> |
| Interference due to compulsions | <i>compinterf</i> | Weight loss/gain | <i>weight</i> |
| Distress caused by compulsions | <i>compdis</i> | Concentration impairment | <i>concen</i> |
| Difficulty resisting compulsions | <i>compresist</i> | Guilt and self-blame | <i>guilt</i> |
| Difficulty controlling compulsions | <i>compcont</i> | Suicidal thoughts or attempts | <i>suicide</i> |
| | | Anhedonia | <i>anhedonia</i> |
| | | Fatigability | <i>fatigue</i> |
| | | Psychomotor slowing | <i>retard</i> |
| | | Psychomotor agitation | <i>agitation</i> |

Table S2: Symptoms and the associated abbreviations on the self report versions of the Yale-Brown Obsessive-Compulsive Scale (Y-BOCS-SR) (Steketee et al., 1996) and the Quick Inventory of Depressive Symptomatology (QIDS-SR) (Rush et al., 2003). We combine the symptoms “Increased appetite” and “Decreased appetite” into one variable *appetite* with 7 levels. Analogously, the symptoms “weight gain” and “weight loss” are combined into the variable *weight*.

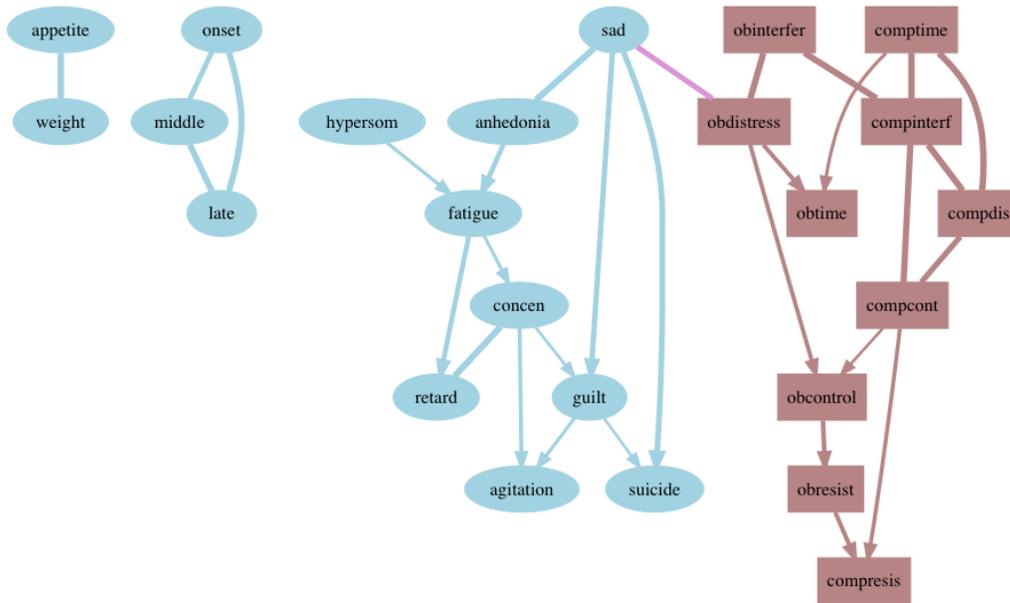


Figure S11: CPDAG of the reproduced Bayesian network using the method described in (McNally et al., 2017). As in Figure 4, rectangles represent nodes related to OCD symptoms, ellipses represent nodes related to depression symptoms, and edges between nodes with different colors are highlighted with pink.

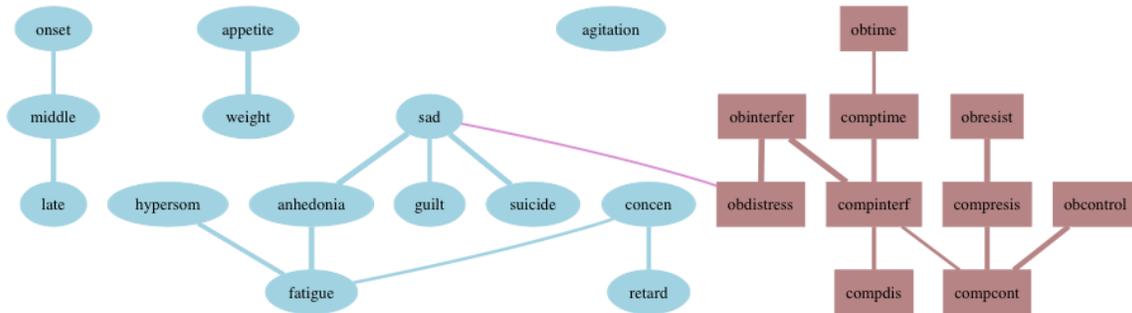


Figure S12: CPDAG estimated via the hybrid method, as described in Section 4, with the BDe score and the nominal PC output as the initial search space. The penalty for the BDe score χ is chosen to be 1.5. As in Figure 4, rectangles represent nodes related to OCD symptoms, ellipses represent nodes related to depression symptoms, and edges between nodes with different colors are highlighted with pink.

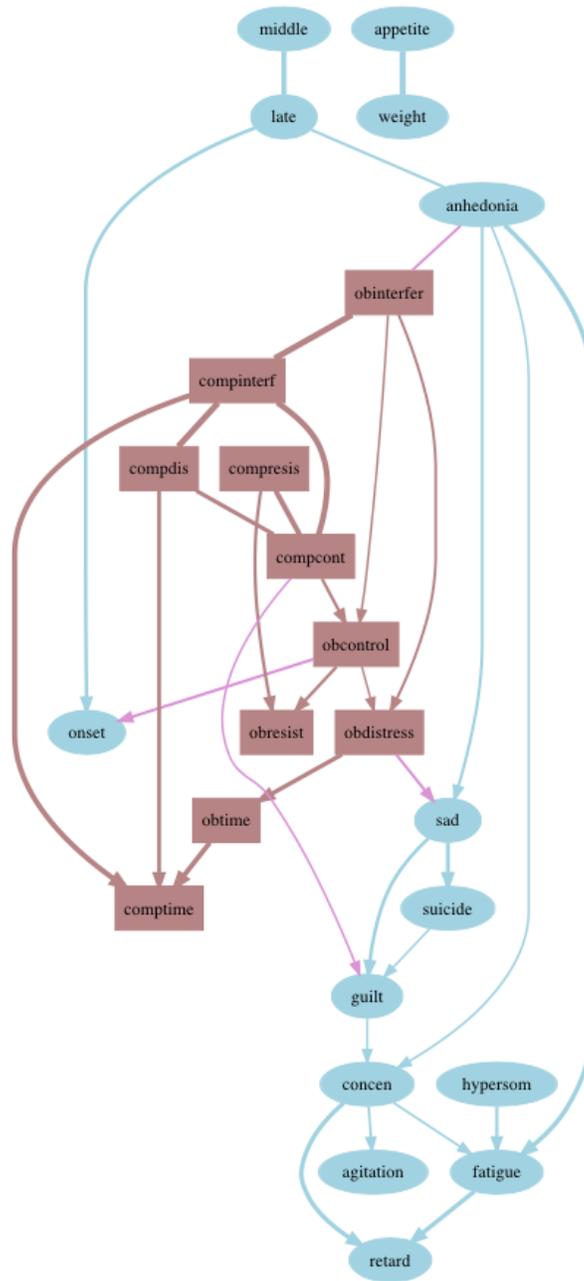


Figure S13: CPDAG estimated via the hybrid method, as described in Section 4, with the BGe score and the Gaussian PC output as the initial search space. The penalty for the BGe score a_m is chosen to be 0.05. As in Figure 4, rectangles represent nodes related to OCD symptoms, ellipses represent nodes related to depression symptoms, and edges between nodes with different colors are highlighted with pink.

LEARNING BAYESIAN NETWORKS FROM ORDINAL DATA

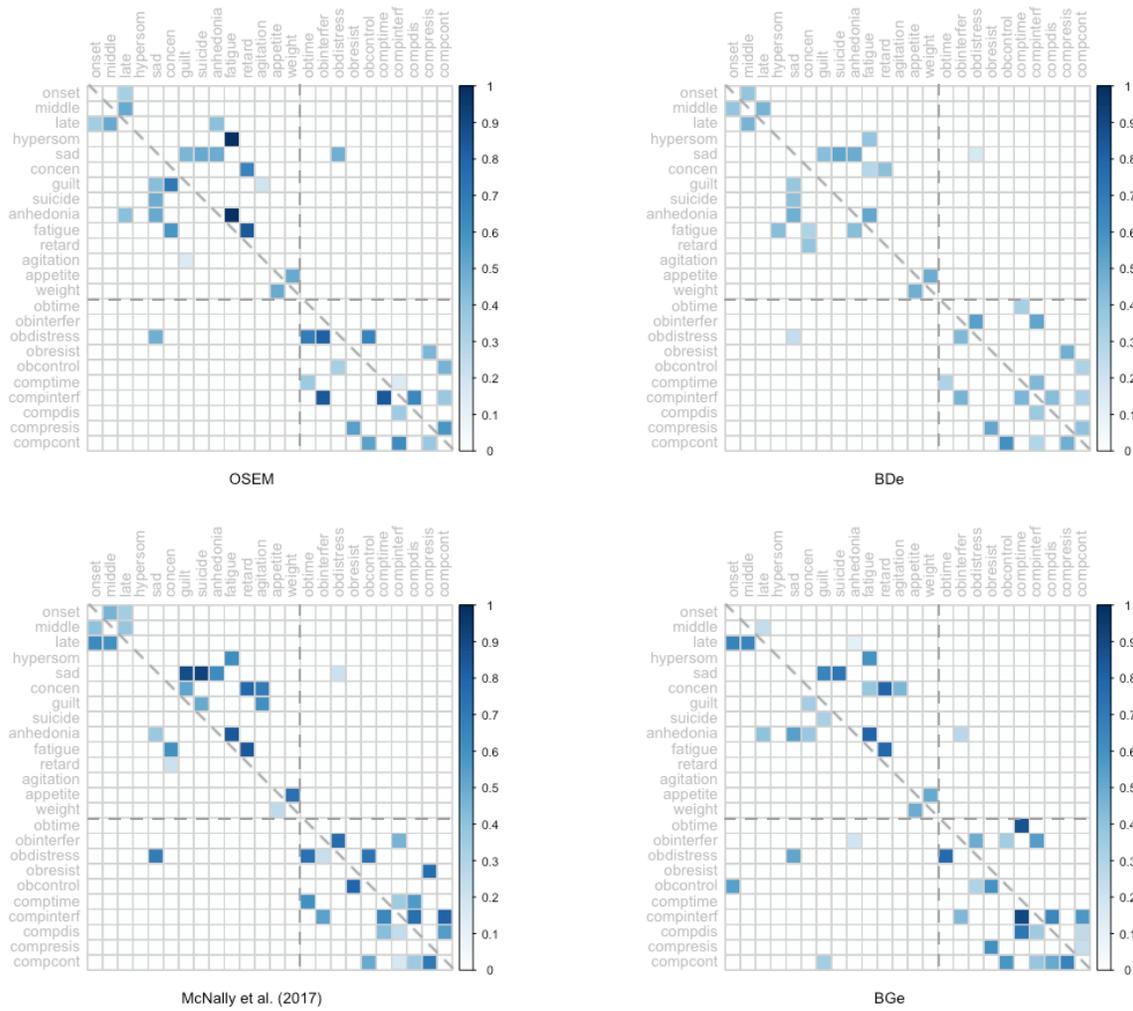


Figure S14: Heatmaps for the CPDAG adjacency matrices estimated using the four methods mentioned in Section 5. The shade in the grid represents the percentage of time a directed edge appears in the 500 Bootstrapped CPDAGs, where an undirected edge counts half for each direction. The darker the shade, the more frequent the corresponding directed edge occurs.

References

- A. Agresti. *Analysis of Ordinal Categorical Data*. John Wiley & Sons, 2010.
- J. Ashford and R. Sowden. Multi-variate probit analysis. *Biometrics*, pages 535–546, 1970.
- A. Barron, J. Rissanen, and B. Yu. The minimum description length principle in coding and modeling. *IEEE Transactions on Information Theory*, pages 2743–2760, 1998.
- J. C. Bird, R. Evans, F. Waite, B. S. Loe, and D. Freeman. Adolescent paranoia: prevalence, structure, and causal mechanisms. *Schizophrenia Bulletin*, pages 1134–1142, 2019.
- R. D. Bock and R. D. Gibbons. High-dimensional multivariate probit analysis. *Biometrics*, pages 1183–1194, 1996.
- S. Chib and E. Greenberg. Analysis of multivariate probit models. *Biometrika*, pages 347–361, 1998.
- D. M. Chickering. Optimal structure identification with greedy search. *Journal of Machine Learning Research*, pages 507–554, 2002.
- D. M. Chickering, D. Heckerman, and C. Meek. Large-sample learning of Bayesian networks is NP-hard. *Journal of Machine Learning Research*, pages 1287–1330, 2004.
- D. Colombo and M. H. Maathuis. Order-independent constraint-based causal structure learning. *The Journal of Machine Learning Research*, pages 3741–3782, 2014.
- D. Colombo, M. H. Maathuis, M. Kalisch, and T. S. Richardson. Learning high-dimensional directed acyclic graphs with latent and selection variables. *The Annals of Statistics*, pages 294–321, 2012.
- A. C. Constantinou, Y. Liu, K. Chobtham, Z. Guo, and N. K. Kitson. Large-scale empirical validation of bayesian network structure learning algorithms with noisy data. *International Journal of Approximate Reasoning*, pages 151–188, 2021.
- R. Cui, P. Groot, and T. Heskes. Copula PC algorithm for causal discovery from mixed data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 377–392. Springer, 2016.
- J. Cussens, M. Järvisalo, J. H. Korhonen, and M. Bartlett. Bayesian network structure learning with integer programming: Polytopes, facets and complexity. *Journal of Artificial Intelligence Research*, pages 185–229, 2017.
- C. Daganzo. *Multinomial Probit: the Theory and Its Application to Demand Forecasting*. Elsevier, 2014.
- R. Daly, Q. Shen, and S. Aitken. Learning Bayesian networks: approaches and issues. *The Knowledge Engineering Review*, pages 99–157, 2011.
- A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, pages 1–22, 1977.

- D. Dua and C. Graff. UCI machine learning repository, 2017. URL <http://archive.ics.uci.edu/ml>.
- N. Friedman. Learning belief networks in the presence of missing values and hidden variables. In *ICML*, pages 125–133, 1997.
- N. Friedman. The Bayesian Structural EM Algorithm. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 129–138, 1998.
- N. Friedman and D. Koller. Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks. *Machine Learning*, pages 95–125, 2003.
- D. Geiger and D. Heckerman. Parameter priors for directed acyclic graphical models and the characterization of several probability distributions. *The Annals of Statistics*, pages 1412–1440, 2002.
- P. Giudici and R. Castelo. Improving Markov chain Monte Carlo model search for data mining. *Machine Learning*, pages 127–158, 2003.
- M. Grzegorzcyk and D. Husmeier. Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move. *Machine Learning*, page 265, 2008.
- N. Harris and M. Drton. Pc algorithm for nonparanormal graphical models. *Journal of Machine Learning Research*, 2013.
- D. Heckerman and D. Geiger. Learning Bayesian networks: a unification for discrete and Gaussian domains. *Eleventh Conference on Uncertainty in Artificial Intelligence*, pages 274–285, 1995.
- D. Heckerman, C. Meek, and G. Cooper. A Bayesian approach to causal discovery. *Computation, Causation, and Discovery*, pages 141–166, 1999.
- G. James, D. Witten, T. Hastie, and R. Tibshirani. *An Introduction to Statistical Learning*. Springer, 2013.
- M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the PC-algorithm. *Journal of Machine Learning Research*, pages 613–636, 2007.
- M. Kalisch, M. Mächler, D. Colombo, M. H. Maathuis, and P. Bühlmann. Causal inference using graphical models with the R package pcalg. *Journal of Statistical Software*, pages 1–26, 2012.
- M. Koivisto and K. Sood. Exact Bayesian structure discovery in Bayesian networks. *Journal of Machine Learning Research*, pages 549–573, 2004.
- D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.
- J. Kuipers and G. Moffa. Partition MCMC for inference on acyclic digraphs. *Journal of the American Statistical Association*, pages 282–299, 2017.

- J. Kuipers, G. Moffa, D. Heckerman, et al. Addendum on the scoring of Gaussian directed acyclic graphical models. *The Annals of Statistics*, pages 1689–1691, 2014.
- J. Kuipers, P. Suter, and G. Moffa. Efficient sampling and structure learning of Bayesian networks. *arXiv preprint arXiv:1803.07859*, 2018.
- J. Kuipers, G. Moffa, E. Kuipers, D. Freeman, and P. Bebbington. Links between psychotic and neurotic symptoms in the general population: an analysis of longitudinal british national survey data using directed acyclic graphs. *Psychological Medicine*, pages 388–395, 2019.
- A. Y. Kuk and D. J. Nott. A pairwise likelihood approach to analyzing correlated binary data. *Statistics & Probability Letters*, pages 329–335, 2000.
- S. L. Lauritzen. *Graphical Models*. Clarendon Press, 1996.
- D. Liu, S. Li, Y. Yu, and I. Moustaki. Assessing partial association between ordinal variables: quantification, visualization, and hypothesis testing. *Journal of the American Statistical Association*, pages 1–35, 2020.
- D. Madigan and A. E. Raftery. Model selection and accounting for model uncertainty in graphical models using Occam’s window. *Journal of the American Statistical Association*, pages 1535–1546, 1994.
- D. Madigan, J. York, and D. Allard. Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, pages 215–232, 1995.
- J. H. McDonald. *Handbook of Biological Statistics*. Sparky House Publishing, 2009.
- R. McNally, P. Mair, B. Mugno, and B. Riemann. Co-morbid obsessive-compulsive disorder and depression: a Bayesian network approach. *Psychological Medicine*, page 1204, 2017.
- C. Meek. Causal inference and causal explanation with background knowledge. In *UAI 1995*, pages 403–410, 1995.
- X.-L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, pages 267–278, 1993.
- G. Moffa and J. Kuipers. Sequential Monte Carlo EM for multivariate probit models. *Computational Statistics & Data Analysis*, pages 252–272, 2014.
- G. Moffa, G. Catone, J. Kuipers, E. Kuipers, D. Freeman, S. Marwaha, B. R. Lennox, M. R. Broome, and P. Bebbington. Using directed acyclic graphs in epidemiological research in psychosis: an analysis of the role of bullying in psychosis. *Schizophrenia Bulletin*, pages 1273–1279, 2017.
- F. Musella. A PC algorithm variation for ordinal variables. *Computational Statistics*, pages 2749–2759, 2013.
- P. Nandy, A. Hauser, M. H. Maathuis, et al. High-dimensional consistency in score-based and hybrid structure learning. *The Annals of Statistics*, pages 3151–3183, 2018.

- R. E. Neapolitan et al. *Learning Bayesian Networks*. Pearson Prentice Hall Upper Saddle River, NJ, 2004.
- J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Elsevier, 2014.
- D. Renard, G. Molenberghs, and H. Geys. A pairwise likelihood approach to estimation in multilevel probit models. *Computational Statistics & Data Analysis*, pages 649–667, 2004.
- T. Richardson, P. Spirtes, et al. Ancestral graph Markov models. *The Annals of Statistics*, pages 962–1030, 2002.
- J. Rissanen. Stochastic complexity. *Journal of the Royal Statistical Society: Series B (Methodological)*, pages 223–239, 1987.
- A. J. Rush, M. H. Trivedi, H. M. Ibrahim, T. J. Carmody, B. Arnow, D. N. Klein, J. C. Markowitz, P. T. Ninan, S. Kornstein, R. Manber, et al. The 16-Item Quick Inventory of Depressive Symptomatology (QIDS), clinician rating (QIDS-C), and self-report (QIDS-SR): a psychometric evaluation in patients with chronic major depression. *Biological Psychiatry*, pages 573–583, 2003.
- G. Schwarz et al. Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464, 1978.
- M. Scutari. Learning Bayesian networks with the bnlearn R package. *arXiv preprint arXiv:0908.3817*, 2009.
- R. Silva and Z. Ghahramani. The hidden life of latent variables: Bayesian learning with mixed graph models. *Journal of Machine Learning Research*, pages 1187–1238, 2009.
- P. Spirtes. An anytime algorithm for causal inference. In *AISTATS*, 2001.
- P. Spirtes, C. N. Glymour, R. Scheines, and D. Heckerman. *Causation, prediction, and search*. MIT press, 2000.
- G. Steketee, R. Frost, and K. Bogart. The yale-brown obsessive compulsive scale: Interview versus self-report. *Behaviour Research and Therapy*, pages 675–684, 1996.
- P. Suter, J. Kuipers, G. Moffa, and N. Beerenwinkel. Bayesian structure learning and sampling of bayesian networks with the r package bidag. *arXiv preprint arXiv:2105.00488*, 2021.
- T. Talvitie, R. Eggeling, and M. Koivisto. Learning Bayesian networks with local structure, mixed variables, and exact algorithms. *International Journal of Approximate Reasoning*, pages 69–95, 2019.
- M. Tsagris and G. Athineou. *Compositional: Compositional Data Analysis*, 2019. URL <https://CRAN.R-project.org/package=Compositional>. R package version 3.5.

- M. Tsagris, G. Borboudakis, V. Lagani, and I. Tsamardinos. Constraint-based causal discovery with mixed data. *International Journal of Data Science and Analytics*, pages 19–30, 2018.
- I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, pages 31–78, 2006.
- C. Uhler, G. Raskutti, P. Bühlmann, and B. Yu. Geometry of the faithfulness assumption in causal inference. *The Annals of Statistics*, pages 436–463, 2013.
- C. Varin and C. Czado. A mixed autoregressive probit model for ordinal longitudinal data. *Biostatistics*, pages 127–138, 2010.
- T. Verma, J. Pearl, et al. Equivalence and synthesis of causal models. Technical report, Computer Science Department, UCLA, Los Angeles, CA, 1991.
- J. Viinikka, A. Hyttinen, J. Pensar, and M. Koivisto. Towards Scalable Bayesian Learning of Causal DAGs. *arXiv preprint arXiv:2010.00684*, 2020.
- E. L. Webb and J. J. Forster. Bayesian model determination for multivariate ordinal and binary data. *Computational Statistics & Data Analysis*, pages 2632–2649, 2008.
- G. C. Wei and M. A. Tanner. A Monte Carlo implementation of the EM algorithm and the poor man’s data augmentation algorithms. *Journal of the American Statistical Association*, pages 699–704, 1990.