Accepted Manuscript

A general algorithm for covariance modeling of discrete data

Gordana C. Popovic, Francis K.C. Hui, David I. Warton

PII: S0047-259X(17)30752-2
DOI: https://doi.org/10.1016/j.jmva.2017.12.002
Reference: YJMVA 4311

To appear in: *Journal of Multivariate Analysis*

Received date : 1 April 2015

A general algorithm for covariance modeling of discrete data

Gordana C. Popovic[a,1,*], Francis K.C. Hui[b], David I. Warton[a,c,2]

[a] School of Mathematics and Statistics, The University of New South Wales, NSW 2052, Australia
[b] Mathematical Sciences Institute, The Australian National University, Acton, ACT 2601, Australia
[c] Evolution and Ecology Research Centre, The University of New South Wales, NSW 2052, Australia

Abstract

We propose an algorithm that generalizes to discrete data any given covariance modeling algorithm originally intended for Gaussian responses, via a Gaussian copula approach. Covariance modeling is a powerful tool for extracting meaning from multivariate data, and fast algorithms for Gaussian data, such as factor analysis and Gaussian graphical models, are widely available. Our algorithm makes these tools generally available to analysts of discrete data and can combine any likelihood-based covariance modeling method for Gaussian data with any set of discrete marginal distributions. Previously, tools for discrete data were generally specific to one family of distributions or covariance modeling paradigm, or otherwise did not exist. Our algorithm is more flexible than alternate methods, takes advantage of existing fast algorithms for Gaussian data, and simulations suggest that it outperforms competing graphical modeling and factor analysis procedures for count and binomial data. We additionally show that in a Gaussian copula graphical model with discrete margins, conditional independence relationships in the latent Gaussian variables are inherited by the discrete observations. Our method is illustrated with a graphical model and factor analysis on an overdispersed ecological count dataset of species abundances.

Keywords: Factor analysis, Gaussian copula, graphical model, overdispersed count data, species interaction.

1. Introduction

Models for covariance give us valuable information about the structure of multivariate data when there are a large number of response variables, and the literature on such tools for Gaussian data is quite advanced. Gaussian graphical models [2, 13, 28, 36, 41] for example, describe conditional independence relationships between variables, which can be used to distinguish between direct and indirect relationships among variables. Factor analysis models can identify latent factors which drive the covariance between variables [9]. In addition, covariance modeling of Gaussian data is a fast moving field, with interesting algorithms continually being developed, including sparse factor analysis [4] and latent variable graphical model [29]. These and other covariance modeling methods were developed in the context of Gaussian data, and equivalent algorithms for discrete data are often limited or do not exist. In this article, we aim to develop a flexible method to apply covariance models to discrete data, with particular focus on overdispersed counts, our motivating example.

Covariance modeling of discrete data has been advanced separately for each covariance modeling paradigm, and these advances generally allow only a narrow class of discrete distributions. In the context of factor analysis, for binomial and multinomial data, item response theory allows limited latent variable modeling [15]. Counts and categorical outcomes can be modeled using, for example, generalized latent variable models [18, 37], a flexible covariance modeling method. These models combine generalized linear mixed models and structural equation models into a unifying

[*] Corresponding author
Email addresses: g.popovic@unsw.edu.au (Gordana C. Popovic), fhui28@gmail.com (Francis K.C. Hui), david.warton@unsw.edu.au (David I. Warton)

[1] GCP is supported by an Australian postgraduate award from the University of New South Wales
[2] DIW is supported by Australian Research Council Discovery Projects and Future Fellow funding schemes (project number DP130102131 and FT120100501)

framework. However these cannot be used to carry out other forms of covariance modeling. More recently graphical models have also been extended to discrete data, but the solutions currently available are piecemeal. For example it is possible to build graphical models for discrete data by extending Gaussian graphical models to other members of the exponential family [2, 21]. However for many distributions, including the commonly used Poisson, these extensions place restrictions on the direction of conditional relationships between variables. To overcome this limitation, node wise graphical models have been proposed [1]; however, these are local in nature and do not estimate a global model of dependence, making them inefficient. Other models for discrete data do not allow modeling of count data [34], while others still do not allow for covariates to be included, as marginal distributions are estimated nonparametrically [10, 24]. Many of these solutions also do not take advantage of the fast graphical modeling algorithms now available for Gaussian data.

In this article, we propose a general algorithm for covariance modeling of discrete data which allows for graphical modeling, factor analysis, as well as other covariance models within a Gaussian copula framework. Our algorithm is very flexible in that it allows any set of marginal distributions to be combined with any covariance modeling algorithm that was originally designed for Gaussian data. Our method also does not restrict the direction of conditional dependence parameters between variables, while still estimating a global model. Finally, the proposed approach allows us to plug in covariance modeling algorithms designed for Gaussian data to model covariance in discrete data, thereby taking advantage of fast algorithms.

Our model and estimation method are most closely related to the Bayesian model described in [6, 14] (see Appendix A.2), and can be understood as a generalization of their method to handle a broader range of covariance modeling frameworks.

We will begin in Section 2 by describing our model and estimation procedure, and presenting a general algorithm for combining any set of marginal distributions with any covariance modeling algorithm. We will then investigate statistical and computational properties of our algorithm. In Section 3 we describe how our method can be used with both maximum likelihood and penalized likelihood covariance models, using graphical models and factor analytic models as examples, and compare our model with alternate algorithms for graphical and factor analytic models for discrete data. We finish by analyzing an example dataset in Section 4.

2. Model formation and estimation

2.1. Notation

Throughout this article, we denote the standard $\mathcal{N}(0, 1)$ univariate Gaussian density by ϕ, the corresponding distribution function by Φ and the multivariate zero mean Gaussian density with covariance matrix S by $\phi_d(\cdot; S)$.

Let y and z denote the observed data and latent variables respectively, both of which are of dimension $N \times d$ where N is the sample size and d is the dimension of the response. For instance, in our applied example in Section 4, N denote the number of sites visited and d the number of species recorded. We use y_i and z_i to denote the d-dimensional observed data and latent variable, for $i \in \{1, \ldots, N\}$, and y_{ij} with $j \in \{1, \ldots, d\}$ to refer to the scaler observation i for dimension j.

2.2. Model formulation

We model response y_i as a Gaussian copula coupled with discrete marginal distributions F_{ij}, characterized by marginal parameters (β_j, ψ_j), and correlation matrix R_θ, parameterized by a set of variables θ. The distribution of y_i is then given [31] by the d-dimensional rectangle integral

$$L_i(y_i|\beta, \psi, \theta) = \int_{B_i} \phi_d(z_i; R_\theta) dz_i \tag{1}$$

where $B_i = \bigcap_j [\Phi^{-1}\{F_{ij}(y_{ij}^-|\beta_j, \psi_j)\}, \Phi^{-1}\{F_{ij}(y_{ij}|\beta_j, \psi_j)\}]$ and $F_{ij}(y_{ij}^-) = \lim_{x \to y_{ij}^-} F_{ij}(x)$ is the left limit of F at y.

The above model can be viewed as a latent variable model. To see this, write the joint distribution of y and z (suppressing the i subscript) as

$$f(y, z) = f(y|z) f(z) = \prod_{j=1}^{d} \mathbf{1}_{[\Phi^{-1}\{F_j(y_j^-)\} \leq z_j < \Phi^{-1}\{F_j(y_j)\}]} \phi_d(\mathbf{z}_i; R_\theta).$$

and we obtain the density of y by integrating over the latent variable z, thereby arriving at Eq. (1).

2.3. Estimation

We implement a type of Monte Carlo expectation maximization (MCEM) algorithm to estimate this integral [40]. We chose an algorithm which is easy to implement, and allows the flexibility we desire. We will start by defining the MCEM algorithm, Gaussian score equation, and Dunn–Smyth residuals [8].

Definition 1 (Monte Carlo Expectation Maximization)**.** The expectation maximization (EM) algorithm [7] is a method to maximize the likelihood function in the presence of missing data z. This is done iteratively. In the E-Step one calculates the Q function, viz.

$$Q(\theta, \hat{\theta}^{(m)}) = \int_{z_i} f(z|y; \hat{\theta}^{(m)}) \ln f(z; \theta) dz,$$

which is the expectation of the log likelihood with respect to the conditional predictive distribution $f(z|y; R_{\hat{\theta}^{(m)}})$, under the current value of the model parameters $\hat{\theta}^{(m)}$ at the mth iteration. The Q function is then maximized in the M-Step to find the new value of the model parameters, viz.

$$\hat{\theta}^{(m+1)} = \arg\max_{\theta} Q(\theta, \hat{\theta}^{(m)}).$$

These steps are repeated iteratively until convergence. When the Q function is not available in closed form, a Monte Carlo estimate of the required expectation can be used instead. This is the Monte Carlo Expectation Maximization (MCEM) algorithm [40]. The Q function is replaced by

$$\tilde{Q}(\theta, \hat{\theta}^{(m)}) = \frac{1}{K} \sum_{k=1}^{K} \ln f(z_k; R_{\hat{\theta}^{(m)}}),$$

in the E-Step, where $z_1, \ldots, z_K$ are drawn from $f(z|y; \hat{\theta}^{(m)})$.

Definition 2 (Gaussian score equation for covariance parameters)**.** The solution to the Gaussian score equations gives the maximum likelihood estimate for covariance parameters θ for Gaussian data. The score equation for a zero mean multivariate Gaussian random variable is given by

$$\frac{\partial}{\partial\theta} \ell(z; R_\theta) = \sum_{i=1}^{N} \frac{\partial}{\partial\theta} \ln \phi_d(z_i; \Sigma_\theta) = 0, \tag{2}$$

where $z_1, \ldots, z_N$ are mutually independent Gaussian random variables and Σ_θ is the covariance matrix, parameterized by θ.

Definition 3 (Dunn–Smyth residuals)**.** Dunn–Smyth residuals are a useful diagnostic tool for generalized linear modeling [8]. They are used here as a device for numerical approximation of the integrand in Eq. (1). Let u_{ij} be independent draws from a standard uniform random variable $\mathcal{U}(0, 1)$. We first define $\upsilon_{ij} = F_{ij}(y_{ij}^-) + u_{ij} f_{ij}(y_{ij})$, which are uniformly distributed on the (0, 1) interval, if y_{ij} has distribution function F_{ij} [12, 30].

A Dunn–Smyth residual is then defined by $\zeta_{ij} = \Phi^{-1}(\upsilon_{ij})$. The distribution of these residuals, conditional on the data and marginal distributions, is a truncated multivariate normal with identity covariance matrix. We can write the distribution of the vector of Dunn–Smyth residuals as

$$g(\zeta_i) = \frac{\prod_{j=1}^{d} \phi(\zeta_{ij})}{\prod_{j=1}^{d} f_{ij}(y_{ij})} \mathbf{1}_{\zeta_i \in B_i}.$$

This distribution has positive probability only in the region of integration of the likelihood defined in Eq. (1), making it a candidate for importance sampling to estimate this integral. Importance sampling schemes using these and similar constructs appear by other names in Heinen and Rengifo [16], Nikoloulopoulos [32] and others, where the resulting approximations are maximized numerically. Instead of implementing a numerical optimization scheme, we note in the following result that the score function, when approximated by importance sampling with Dunn–Smyth residuals, can be rewritten as a weighted sum of Gaussian score equations. This key finding allows us to maximize the likelihood using the same algorithms developed for covariance modeling of Gaussian data.

Lemma 1. *The likelihood of the discrete Gaussian copula can be approximated by importance sampling with K sets of Dunn–Smyth residuals*

$$L(y|\beta,\psi,\theta) = \prod_{i=1}^{N} \int_{B_i} \phi_d(z_i; R_\theta) dz_i \approx \prod_{i=1}^{N} \prod_{j=1}^{d} f_{ij}(y_{ij}) \prod_{i=1}^{N} \sum_{k=1}^{K} c(\zeta_i^k; R_\theta), \tag{3}$$

where $c(\zeta_i; R_\theta) = \phi_d(\zeta_i; R_\theta) / \prod_{j=1}^{d} \phi\{\Phi^{-1}(\zeta_{ij})\}$ *and* f_{ij} *is the marginal density of variable* j *and observation* i*, and* ζ_i *are Dunn–Smyth residuals distributed according to* g*.*

The proof of Lemma 1 is given in Appendix A and follows from importance sampling arguments. We now present the main result of the article, which demonstrates the link between the Gaussian score equation and the Gaussian copula score.

Theorem 1. *An estimate of the derivative (with respect to covariance parameters) of the likelihood of the Gaussian copula with discrete margins can be written as a weighted sum of derivatives of the multivariate Gaussian distribution. So*

$$\frac{\partial \ell(y; R_\theta)}{\partial \theta} = \sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik}(R_\theta) \frac{\partial}{\partial \theta} \ln \phi_d(\zeta_i^k; R_\theta), \tag{4}$$

where $\ell(y; R_\theta) = \ln\{L(y; R_\theta)\}$*,* $w_{ik}(R_\theta) \equiv c(\zeta_i^k; R_\theta) / \sum_{m=1}^{K} c(\zeta_i^m; R_\theta)$

Proof. Differentiating the log likelihood approximation from Lemma 1, we have

$$\begin{aligned}
\frac{\partial}{\partial \theta} \ell(y; R_\theta) &= \sum_{i=1}^{N} \frac{1}{\sum_{m=1}^{K} c(\zeta_i^m; R_\theta)} \sum_{k=1}^{K} \frac{\partial}{\partial \theta} c(\zeta_i^k; R_\theta) \\
&= \sum_{i=1}^{N} \frac{1}{\sum_{m=1}^{K} c(\zeta_i^m; R_\theta)} \sum_{k=1}^{K} \frac{\partial c(\zeta_i^k; R_\theta)}{\partial \ln c(\zeta_i^k; R_\theta)} \frac{\partial}{\partial \theta} \ln c(\zeta_i^k; R_\theta) \\
&= \sum_{i=1}^{N} \frac{1}{\sum_{m=1}^{K} c(\zeta_i^m; R_\theta)} \sum_{k=1}^{K} c(\zeta_i^k; R_\theta) \frac{\partial}{\partial \theta} \ln c(\zeta_i^k; R_\theta)
\end{aligned}$$

Bringing the first fraction inside the sum over K, we obtain a weighted sum of derivatives of the multivariate Gaussian distribution

$$\frac{\partial}{\partial \theta} \ell(y; R_\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} \frac{c(\zeta_i^k; R_\theta)}{\sum_{m=1}^{K} c(\zeta_i^m; R_\theta)} \frac{\partial}{\partial \theta} \ln c(\zeta_i^k; R_\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik}(R_\theta) \frac{\partial}{\partial \theta} \ln \phi_d(\zeta_i^k; R_\theta)$$

which completes the argument. □

Covariance modeling algorithms, like those which estimate a factor analytic model or structured covariance matrices, maximize the Gaussian likelihood by design, or equivalently solve the Gaussian score equations, Eq. (2). By writing the copula score equation as a weighted sum of the Gaussian scores, we are able to utilize these algorithms with a weighted set of the Dunn–Smyth residuals. As the weights w_{ik} are a function of the parameters to be estimated, these must be iteratively updated, and so we propose the following algorithm.

2.3.1. Algorithm

To carry out covariance modeling on discrete data with a Gaussian copula, we iteratively implement the covariance modeling algorithm designed for Gaussian data on a weighted set of Dunn–Smyth residuals.

Algorithm 1 Covariance modeling for discrete data

```
For data $y$ and covariates $X$

  1. Estimate $F_{ij}(\cdot; X_i)$ using a univariate modeling algorithm (e.g., glm).
  2. For each $k \in \{1, \dots, K\}$, generate Dunn–Smyth residuals $\zeta_{ijk} = \Phi^{-1}\{\hat{F}_{ij}(y_{ijk} - 1) + u_{ijk}\hat{f}_{ij}(y_{ij})\}$.
  3. Initialize $w_{ik}^{(0)} \propto 1$ and write $\{\zeta, w^{(m)}\}$ for the set of Dunn–Smyth residuals and weights.
  4. For $m = 1, 2, \dots$, until convergence
       a Apply the covariance modeling algorithm to weighted residuals $(\zeta, w^{(m-1)})$ to obtain $\hat{\theta}^{(m)}$.
       b Recalculate weights $w_{ik}^{(m)} \propto c(\zeta_{ik}; R_{\hat{\theta}^{(m)}})$ from Theorem 1.
```

Note: As most covariance modeling algorithms use the sample covariance matrix as a sufficient statistic, we can in practice use the weighted correlation matrix of Dunn–Smyth residuals

$$R_w^{(m)} = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} w_{ik}(R_{\hat{\theta}^{(m-1)}}) \zeta_i^k (\zeta_i^k)^\top$$

as a sufficient statistic in Step 4a.

This algorithm has two estimation steps. First, marginal parameters (β, ψ) are estimated assuming independence, as with independence estimating equations [23]. Second, these estimates $(\hat{\beta}, \hat{\psi})$ are plugged into the likelihood $L(y|\beta, \psi, \theta)$, defined in Eq. (3). The resulting plug-in likelihood $L(y|\hat{\beta}, \hat{\psi}, \theta)$ is maximized for covariance parameters θ using an iterative procedure, which can be understood as a MCEM algorithm [27] where the sample for the E-step is achieved by reweighting the residuals, and the M-Step is the covariance modeling algorithm; see Appendix A.2 for proof. Such algorithms are sometimes referred to as *inference function of margins* [33] and have good asymptotic properties, including asymptotic efficiency relative to maximum likelihood [19].

Our algorithm produces consistent estimates of all model parameters (proof in Appendix A.3). It extends the flexibility of Gaussian copulas to implement any covariance modeling framework designed for Gaussian data to discrete data.

2.3.2. Comparison to other efficient methods

The main purpose of our proposed algorithm is to provide flexibility — the capacity to take data from any marginal distribution and fit any covariance modeling algorithm originally designed for Gaussian data. This flexibility comes at some cost in computational efficiency, with inefficiencies introduced at two places. First, we use a Monte Carlo approach to estimation, hence our computations scale linearly with NK rather than with N alone. Second, by using importance sampling rather than sampling directly from the posterior, a larger number of samples (K) is required for estimates with comparable accuracy. Several improvements could be made to Algorithm 1 (see, e.g., [26, 32]) to improve computational efficiency, but this would come at the cost of reducing the generality of the algorithm. However it is worth noting that our use of a two-step process, estimating marginal parameters once prior to covariance modeling, offers a significant computational saving as compared to joint optimization, as seen later in our simulations (Figure 4).

2.3.3. Bias, variance and Mean Squared Error (MSE)

For Algorithm 1, variance and bias may arise as a result of two mechanisms. First, Monte Carlo error is introduced by the importance sampling. Second, we estimate marginal model parameters assuming independence, followed by correlation parameters conditional on these.

To investigate this issue we simulate data from a bivariate Gaussian copula model with marginal Poisson distributions, no intercept and one balanced binary predictor with coefficients equal to 1 for both margins, correlation $\rho \in \{0, 0.2, 0.4, 0.8\}$, and sample sizes $N \in \{10, 100, 1000\}$. In other words, the true parameter vector is given by $\theta_0 = (1, 1, \rho)$. Furthermore, these marginal parameters lead to low means (1 and 2.72 in the two groups), and hence very low counts. We simulated 200 datasets for each of the above combinations, and computed the empirical bias, variance and MSE of the estimated parameters, averaged over the 200 datasets.

For each simulated dataset we estimate the likelihood with $K \in \{1, 10, 100, 1000\}$ blocks of uniform random variables. In this simple setting, given the low dimensionality of the problem, we can use the `optim` function in R to find maximum approximate likelihood solutions $\hat{\theta}_K$ similar to the estimation method in Heinen and Rengifo [16], Nikoloulopoulos [32] and others. We also implement Algorithm 1 to find $\tilde{\theta}_K$. We carry out 200 simulation of each with the above combinations.

Algorithm 1 is generally more biased than maximum approximate likelihood for the correlation parameter (Figure 1 top). However, MSE for Algorithm 1 is smaller when sample size (N) is small, and for moderate ρ (Figure 1 bottom). This may be due to smaller variance of estimates based on Algorithm 1, as ρ is not jointly maximized for marginal and covariance parameters. There are no clear patterns in relative bias for the marginal coefficient β for the two methods. Particularly for large sample sizes, the two methods do about equally well (Figure 2 top). MSE is generally lower for Algorithm 1 for small sample sizes, and higher for moderate sample sizes, while in large sample sizes the algorithms performed about equally well in terms of MSE.

To explore Monte Carlo error, we ran each of the above simulations 10 times, with different sets of K random uniform values but keeping y constant. Monte Carlo error is estimated as the average variance within each simulation of y. We plot Monte Carlo error relative to the sampling error of the maximum likelihood solution $\hat{\theta}$. We estimate this by noting $\hat{\theta} = \lim_{K\to\infty} \hat{\theta}_K$ and letting $K = 10{,}000$. In our simulation settings, Monte Carlo variance was always less that sampling error, with the ratio decreasing rapidly as K increases (Figure 3). With $K = 100$, the Monte Carlo error was less than 1% of sampling error for small and moderate correlations.

Finally, the computational time for both algorithms not surprisingly increased with both N and K, with `optim` being much slower that Algorithm 1 (Figure 4). The ratio of computational time also decreased with N, particularly when K is small.

2.3.4. Guidance for number of Monte Carlo samples

As both the bias and Monte Carlo error of Algorithm 1 reduce with K, while computational time increases, it is important that we chose an appropriate value for K in practical applications. One approach for choosing the number of Monte Carlo samples is to start with a relatively small value of K, and increase it with each iteration until a stopping criterion is reached [3, 22]. The stopping rule is derived from a normal approximation of the current estimate $\gamma^{(m+1)}$ based on the previous estimate $\gamma^{(m)}$. That is, if $\gamma^{(m+1)}$ is inside a $100 \times (1-\alpha)\%$ confidence ellipsoid for $\gamma^{(m)}$, then K is increased according to $K \leftarrow K + K/q$, where q is a positive constant; see [3] for details. We repeat this process until the convergence criterion is reached in three successive occasions.

3. Application to covariance modeling methods

Algorithm 1 can be implemented with covariance models estimated by penalized likelihood as well as maximum likelihood. We will demonstrate this with two examples, graphical modeling for penalized likelihood and factor analysis for maximum likelihood.

3.1. Application to graphical models

Modern implementations of graphical modeling for Gaussian data optimize a penalized likelihood with a lasso penalty [2]. Though this is not a maximum likelihood algorithm, as required by Theorem 1, we will show that Algorithm 1 can nevertheless be used to carry out graphical modeling of discrete data.

We begin by applying the relevant likelihood penalty to the approximate log likelihood in Theorem 1. Let $\Theta = R^{-1}$ be the precision matrix. The penalized log likelihood estimate then can be written as

$$\ell^{\lambda}(\mathbf{y}; \Theta) = \left[\sum_{i=1}^{N}\sum_{j=1}^{d} \ln\{f_{ij}(y_{ij})\}\right] + \sum_{i=1}^{N} \ln\left\{\sum_{k=1}^{K} c(\zeta_i^k; \Theta)\right\} - \lambda\, \|\Theta\|_1,$$

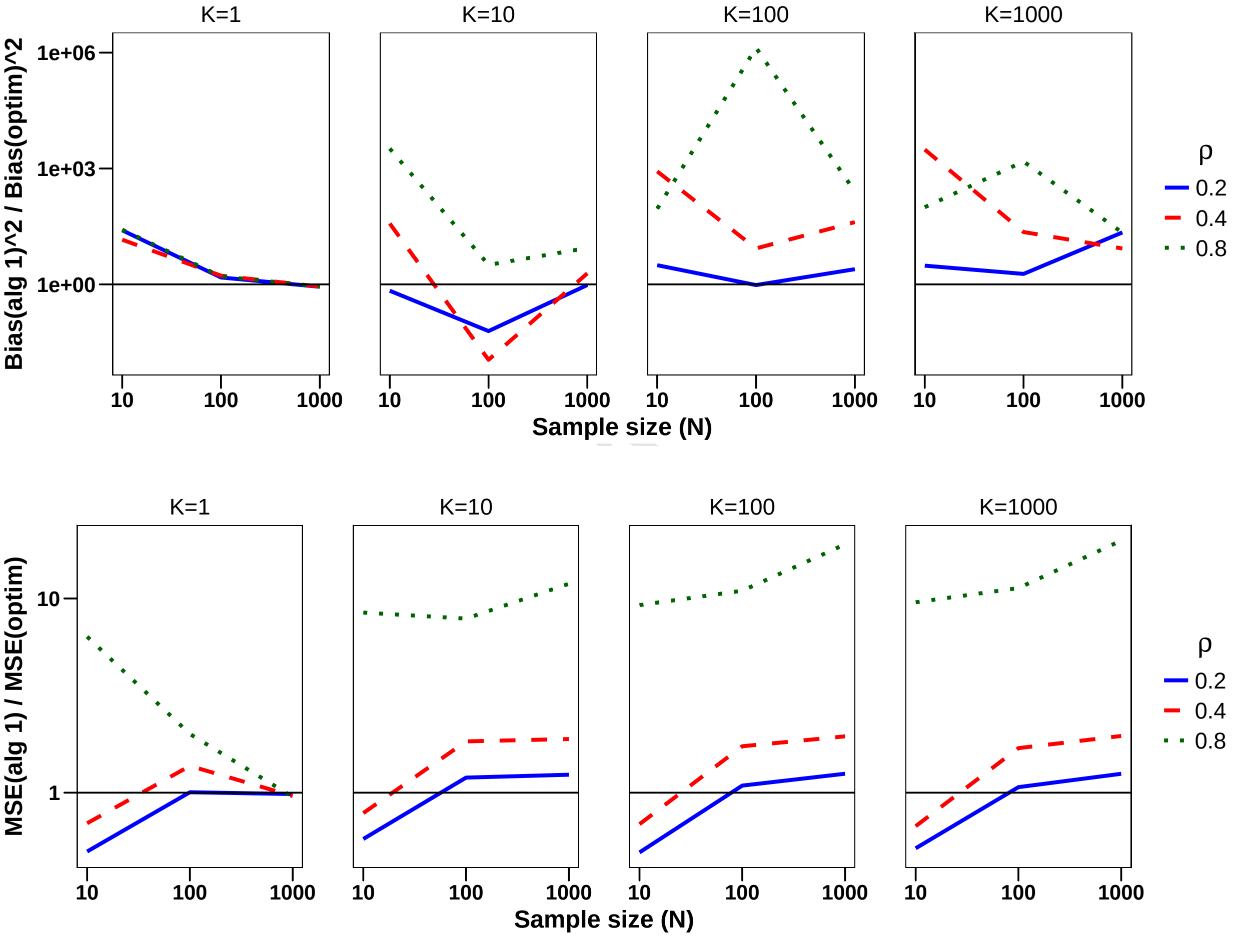


Figure 1: Bias ratio (top) and MSE ratio (bottom) for correlation parameter ρ: Algorithm 1 is generally more biased than `optim`, except for small K. However, for small sample sizes and small ρ, Algorithm 1 has smaller MSE relative to `optim`, while for larger sample sizes and large ρ, `optim` performs better.

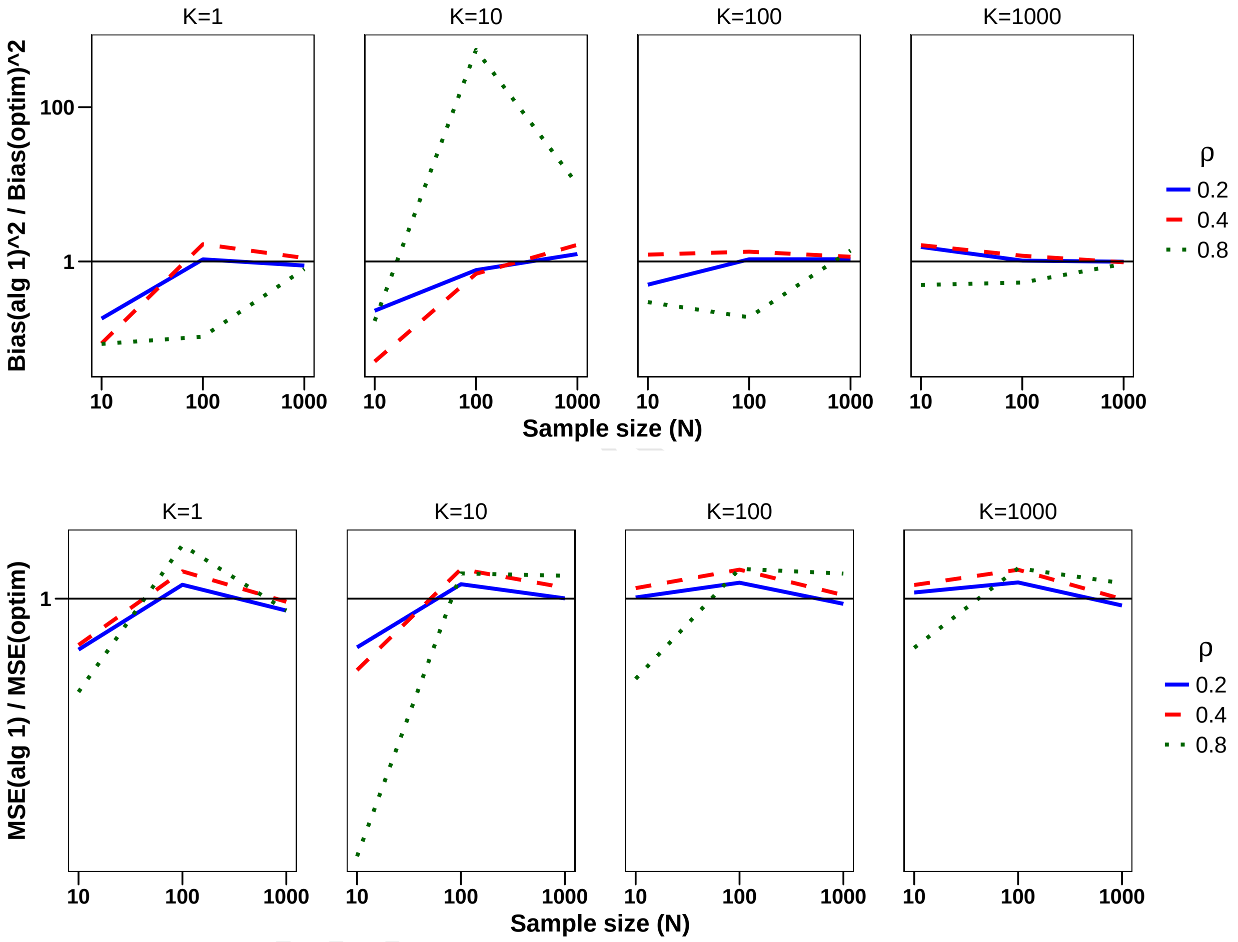


Figure 2: Bias ratio (top) and MSE ratio (bottom) marginal coefficient β: There is no clear pattern in bias, with each algorithm being less biased in some circumstances. For large ρ, and small sample sizes, Algorithm 1 has smaller MSE, while `optim` has smaller MSE for moderate sample sizes. Both algorithms perform similarly in terms of MSE for large sample sizes.

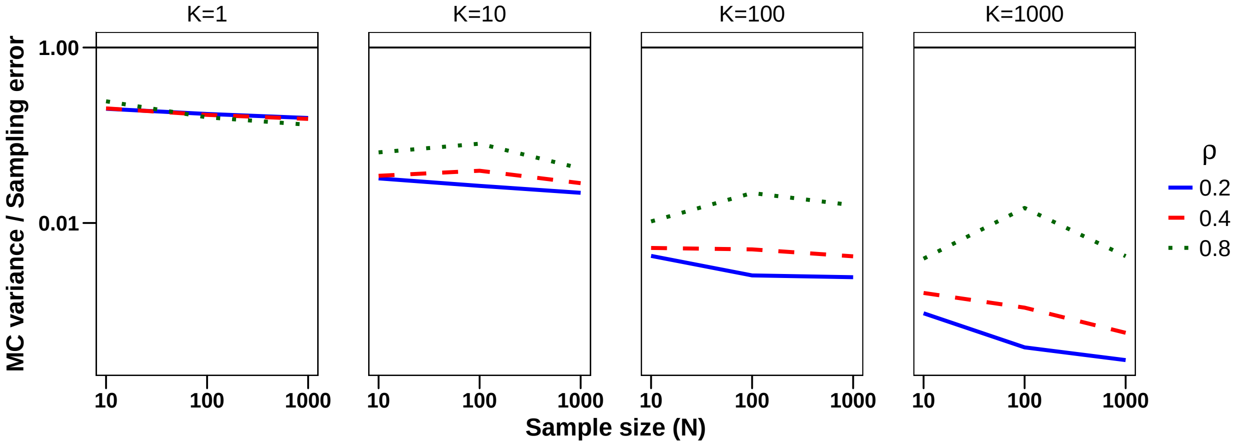


Figure 3: Ratio of Monte Carlo variance to sampling error for the covariance parameter ρ for Algorithm 1: Monte Carlo variance is less than sampling error for all K, N and ρ. Monte Carlo variance reduces relative to sampling error as the number of Monte Carlo samples (K) increases and for smaller ρ.

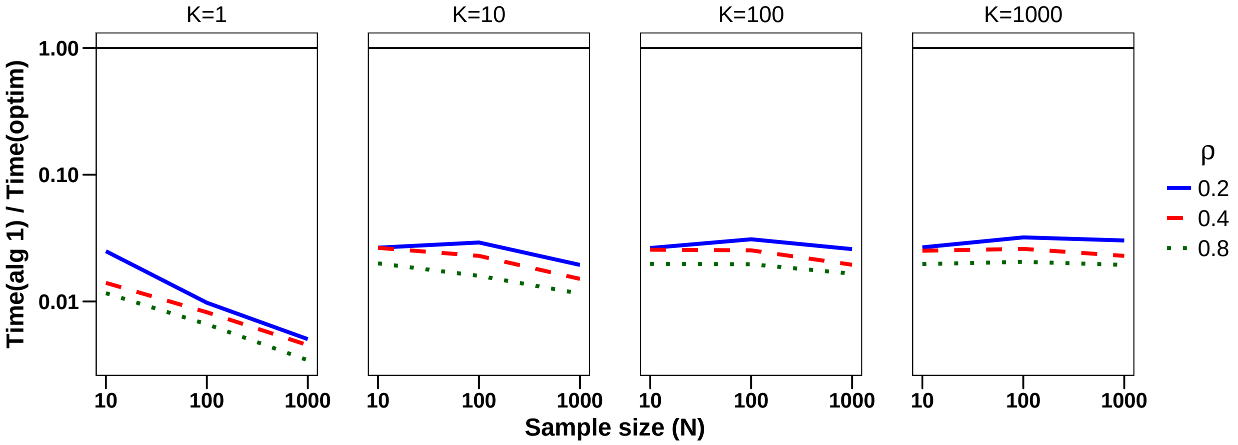


Figure 4: Ratio of time taken for Algorithm 1 and `optim`: Algorithm 1 is quicker than `optim` for all K, N and ρ, though the ratio of times is largely stable with changing K, N, and ρ.

from Lemma 1. To find the maximizer of this function, we write

$$\begin{aligned}
0 &= \frac{\partial \ell^{\lambda}(\mathbf{y};\Theta)}{\partial \Theta} = \sum_{i=1}^{N}\sum_{k=1}^{K} w_{ik}(\Theta)\left\{\frac{\partial}{\partial \Theta}\ \ln \phi_d(\zeta_i^k;\Theta)\right\} - \lambda \Gamma,\\
0 &= \sum_{i=1}^{N}\sum_{k=1}^{K} w_{ik}(\Theta)\{\Theta^{-1} - \zeta_i^k \zeta_i^{k\top}\} - \lambda\Gamma,\\
0 &= \Theta^{-1} - \left\{\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} w_{ik}(\hat{\Theta})\zeta_i^k\zeta_i^{k\top}\right\} - \lambda\Gamma, \qquad (5)
\end{aligned}$$

where $\Gamma_{q,r} = \operatorname{sign}(\Theta_{q,r})$ if $\Theta_{q,r} \neq 0$ and $\Gamma_{q,r} \in [-1,1]$ if $\Theta_{q,r} = 0$. Now, note that the subgradient equation solved by Gaussian graphical modeling algorithms like the graphical lasso [13] is $0 = \Theta^{-1} - S - \lambda\Gamma$, where S is the sample covariance matrix. Therefore, we can see that Eq. (5) is analogous to this, with the sample covariance matrix replaced by a weighted covariance matrix with weights w_{ik}. We can therefore solve Eq. (5) iteratively using the graphical lasso algorithm together with Algorithm 1.

3.2. Application to factor analysis

As a second example, consider a factor analytic model for discrete data. Factor analysis can be estimated by maximum likelihood, either by numerically solving the score equations or with the EM algorithm. The numerical algorithms are implemented to solve the following score equations [9]:

$$0 = \operatorname{diag}\left\{\Sigma^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} y_i {y_i}^\top\right)\Sigma^{-1} - \operatorname{diag}(\Sigma^{-1})\right\}, \quad 0 = \Sigma^{-1}\left(\frac{1}{N}\sum_{i=1}^{N} y_i {y_i}^\top\right)\Sigma^{-1}\Lambda - \Sigma^{-1}\Lambda.$$

Now looking at the likelihood estimate we have

$$\ell(y;\Lambda,\Psi) = \left[\sum_{i=1}^{N}\sum_{j=1}^{d} \ln\{f_{ij}(y_{ij})\}\right] + \sum_{i=1}^{N} \ln\left\{\sum_{k=1}^{K} c(\zeta_i^k;R)\right\},$$

where $R = \Lambda\Lambda^\top + \Psi$. We differentiate with respect Ψ to obtain

$$\begin{aligned}
0 &= \frac{\partial \ell(y;R)}{\partial \Psi} = \sum_{i=1}^{N}\sum_{k=1}^{K} w_{ik}(R)\left\{\frac{\partial}{\partial \Psi}\ \ln\phi_d(\zeta_i^k;R)\right\},\\
0 &= \sum_{i=1}^{N}\sum_{k=1}^{K} w_{ik}(R)\left[\operatorname{diag}\{R^{-1}\zeta_i^k\zeta_i^{k\top}R^{-1} - \operatorname{diag}(R^{-1})\}\right],\\
0 &= \operatorname{diag}\left[R^{-1}\left\{\frac{1}{N}\sum_{i}\sum_{k=1}^{K} w_{ik}(R)\zeta_i^k\zeta_i^{k\top}\right\}R^{-1} - \operatorname{diag}(R^{-1})\right].
\end{aligned}$$

Similarly the derivatives with respect to Λ give us score equations

$$0 = R^{-1}\left\{\frac{1}{N}\sum_{i=1}^{N}\sum_{k=1}^{K} w_{ik}(R)\zeta_i^k\zeta_i^{k\top}\right\}R^{-1}\Lambda - R^{-1}\Lambda.$$

Comparing these to the score equations for a factor analytic model for Gaussian data, it is clear that we can replace the data covariance matrix with a weighted covariance matrix of Dunn–Smyth residuals with weights w_{ik}, and use standard factor analysis algorithms, iteratively updating the weights, to model the correlation matrix R.

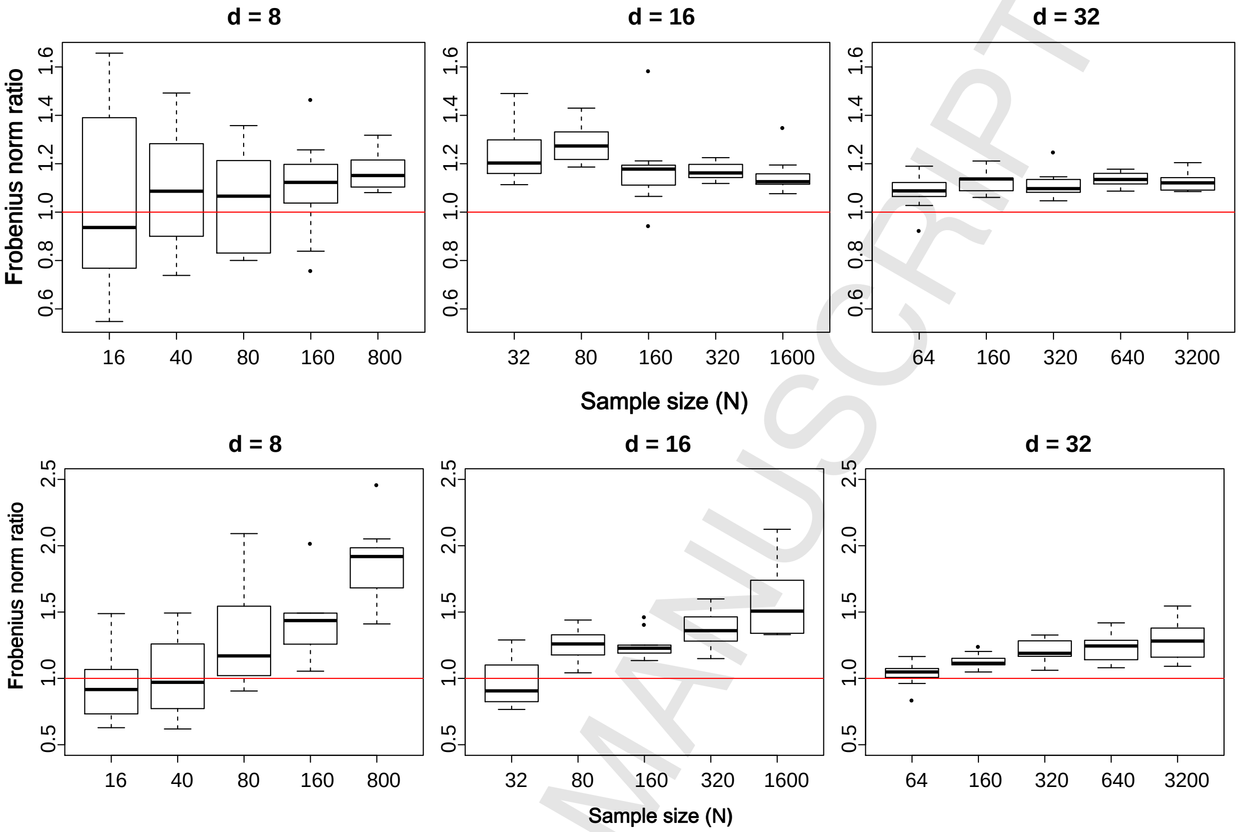


Figure 5: Comparison of Algorithm 1 to factor analysis of Pearson residuals (top) and the `lava.tobit` package (bottom). Values above the red line indicate the copula model is performing better. Algorithm 1 with $K = 50$ generally outperforms these alternatives, especially as dimension increases.

3.3. Simulation results

3.3.1. Factor analysis: Binary data

We compare Algorithm 1 to two alternative strategies for factor analysis of discrete data. We generate a one factor binomial model for binary data with probit link using the `lava.tobit` [17] package. This package is able to simulate and estimate a probit regression with latent factors using composite likelihood. In this special case, the Gaussian copula is equivalent to a hierarchical model [32], and thus can be fitted using software for hierarchical latent variable modeling, like `lava.tobit`. Additionally we will compare to a naive procedure in which we carry out a factor analysis on Pearson residuals or one set of Dunn–Smyth residuals from a binomial generalized linear model. Both these sets of residuals should be approximately normally distributed marginally, and so a factor analysis algorithm can be applied directly to these residuals for an approximate solution. Simulations used $K = 50$ sets of Dunn–Smyth residuals.

We measure the performance of factor analysis models by the Frobenius norm, as in [20], of the difference of estimated and true covariance matrices. Figure 5 (top) shows that Algorithm 1 generally outperforms the naive application of factor analysis algorithms to Pearson residuals. One set of Dunn–Smyth residuals performs similarly to Pearson residuals, and we do not include these results. Figure 5 (bottom) shows that with as few as 50 sets of Dunn–Smyth residuals Algorithm 1 generally outperforms the `lava.tobit` package in terms of accuracy.

3.3.2. Graphical model: Count data

For graphical modeling we simulate data from a a Gaussian copula model with Poisson marginal distributions, and a chosen graphical structure. We then measure how well our model, and others, are able to discover the graphical

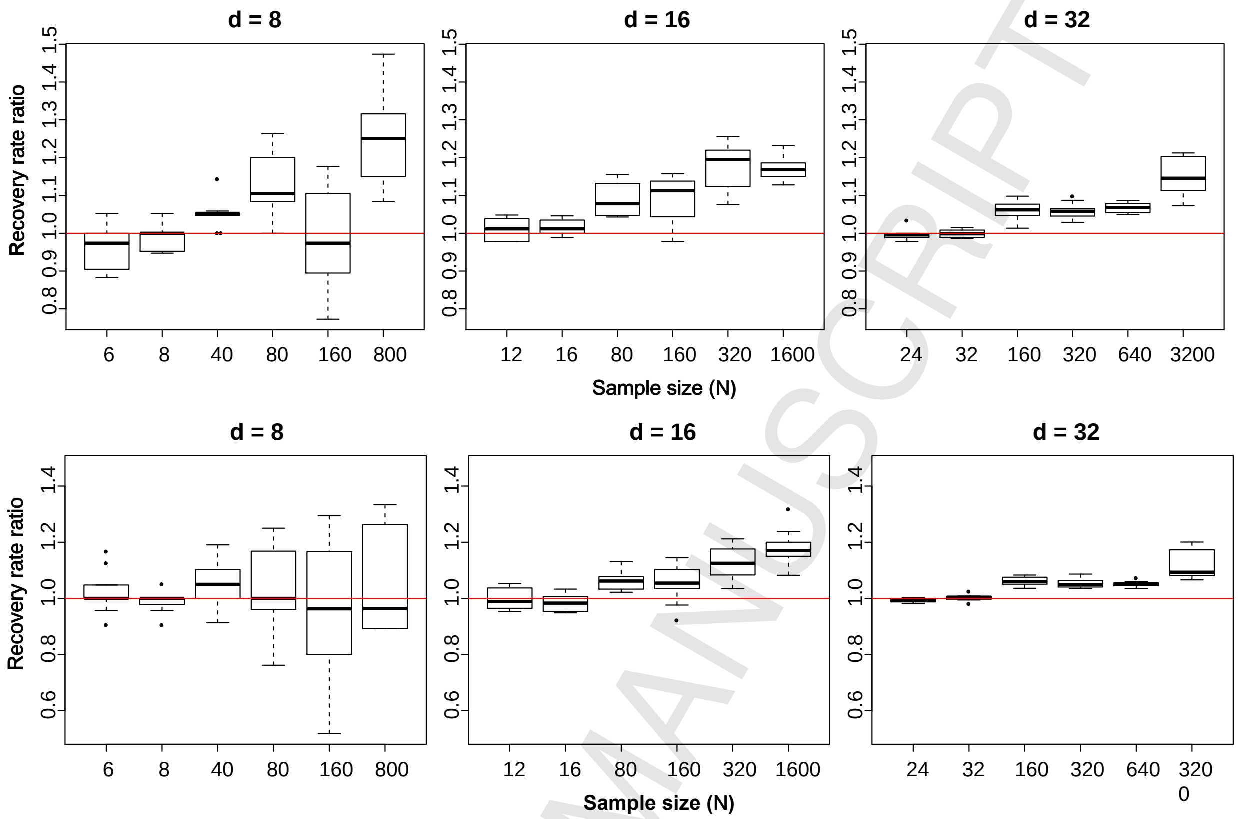


Figure 6: Comparison of Algorithm 1 to graphical modeling of Pearson residuals (top) and the local Poisson model [bottom; 1]. Values above the red line indicate the copula model has a higher recovery rate, (i.e., proportion of correctly identified conditional dependence relationships) and is therefore performing better. Algorithm 1 generally outperforms these alternatives, especially as dimension increases.

structure. We generated and estimated graphical structures and data using the huge package [42] in R, which simulates and estimates Gaussian graphical models. Graphical modeling works best for sparse matrices, so the graphs we generate have a 70% probability for conditional independence for any pair of variables. For model selection we use the StARS criterion [25] which chooses the model with the most stable graphical structure across sub samples. We then compare Algorithm 1 to the local Poisson model [1] as well as a naive application of a graphical modeling algorithm to Pearson and one set of Dunn–Smyth residuals.

We measure the performance of the graphical modeling algorithms as the proportion of correctly identified conditional dependence relationships. Figure 6 (top) shows that Algorithm 1 generally outperforms the naive application of graphical modeling algorithms to Pearson residuals, one set of Dunn–Smyth residuals performs similarly to Pearson residuals and is not shown. Algorithm 1 also generally outperforms the local Poisson model (Figure 6 (bottom)), particularly as dimension (d) increases.

Poisson distributed counts were simulated for easy comparison to the local Poisson model, but note our model can easily be extended to modeling overdispersed counts by using a negative binomial regression in the marginal model, as below.

4. Practical application

4.1. *Count data: Spiders*

We demonstrate our method on counts of the number of hunting spiders caught in traps for 12 species taken from 28 sites modeled as a function of environmental variables [38]. For these data we fit marginal negative binomial generalized linear models with all the available environmental covariates using the mvabund package [39], which also contains these data. We then use Algorithm 1 to estimate the graph of conditional independences using the graphical lasso implemented in the glasso package [13]. We also carry out a factor analysis using the factanal function in base R [35].

In Figure 7 we present output from a factor analysis and graphical model before controlling for covariates (left) and after (right). The first row are factor scores in a two factor model, the second row are loadings for each species, and the third row are the graphs obtained from a graphical model.

In our ecological example, we are interested in studying the covariance relationships before and after accounting for correlation due to the environmental covariates. With variables representing different species, we can interpret the graphical model as a model of species interactions, and attempt to identify which species interact directly with one another, and which are correlated due to their interaction with common species. The factor analysis of these data highlights latent factors which drive correlations among species, and which may be unmodeled environmental variables.

We have coded the plots of scores (Figure 7a-b) according to the presence of bare sand (filled for present and unfilled for absent), and the presence of fallen leaves (blue triangle for present and red square for absent). We observe clustering in Figure 7a according to both variables. Sites with bare sand and no fallen leaves (filled squares) load negatively on both factors (bottom left of Figure 7a), while sites with fallen leaves and no bare sand (unfilled triangles) load positively on factor 2 and negatively on factor 1 (top left of Figure 7a). No patterns are visible after controlling for these covariates (Figure 7b).

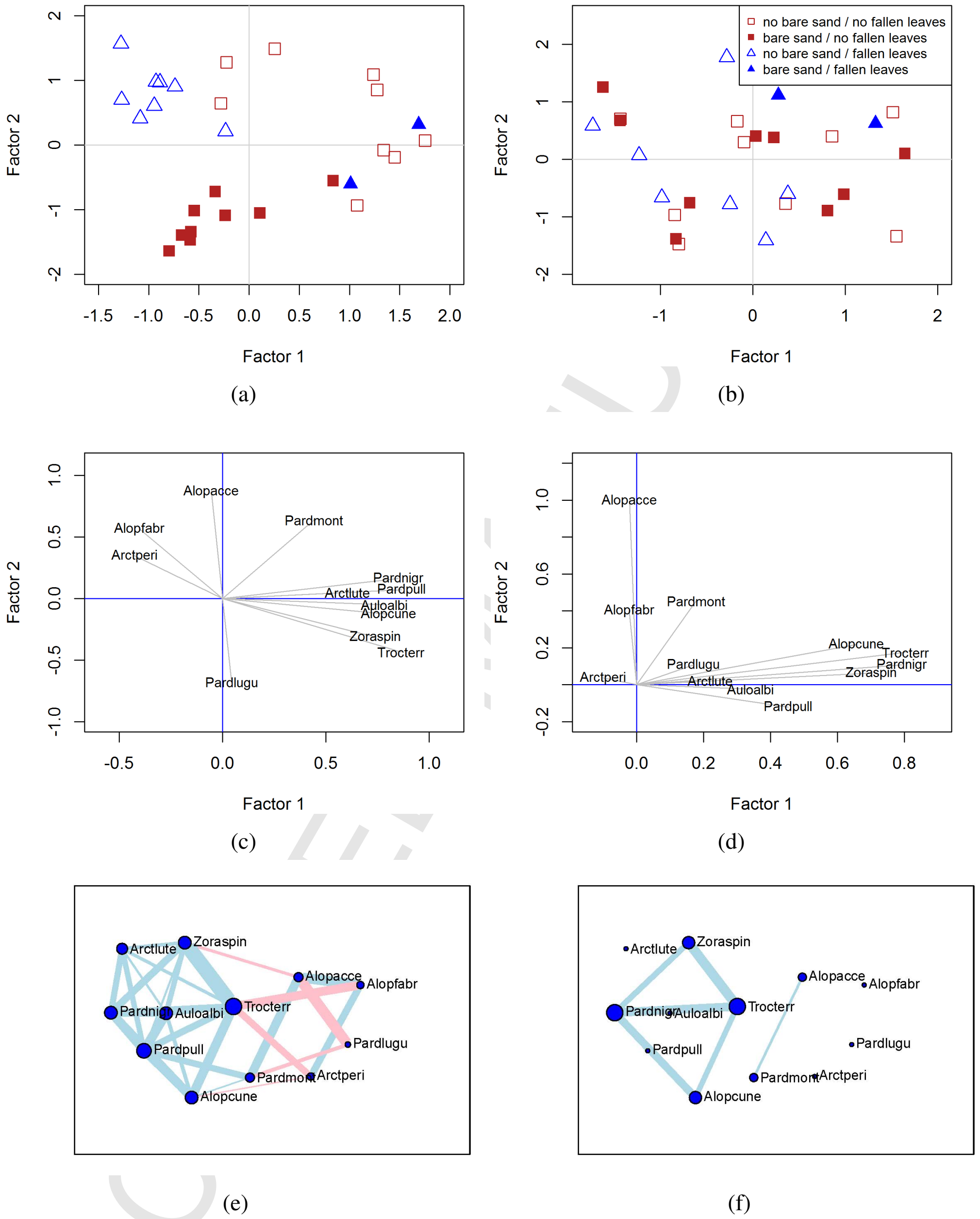
Additionally there are patterns among species in Figures 7c-f. For example, species *Pardlugu (Pardosa lugubris)* has negative interactions with both *Alopacce (Alopecosa accentuata)* and *Pardmont (Pardosa monticola)*, who interact positively with one another, before controlling for covariates (Figure 7e). However these negative interactions are absent after controlling for covariates (Figure 7f), suggesting that this negative correlation can be explained by contrasting habitat preferences. The factor loadings (Figure 7c) suggest the main difference between these species was on Factor 2, and sites seem to differ along this axis primarily in the amount of bare sand (7a), suggesting that the negative correlation can be largely explained by differences in preferences for bare sand. Specifically, *Pardlugu* seems to prefer sites with bare sand, whereas *Alopacce* and *Pardmont* do not.

5. Discussion

We have developed a general algorithm for covariance modeling of discrete data. It can combine any likelihood based covariance modeling procedure designed for Gaussian data with any set of marginal distributions, and is simple and flexible to implement. The algorithm we present does not place restrictions on the sign of covariance parameters, nor is it restricted to one or a small class of covariance models. It is fully flexible in terms of both the marginal distributions and covariance parameters, and only assumes the covariance structure of the latent variable is that of a multivariate Gaussian, and marginal distributions are correctly specified.

Simulation results show our method is not only more general than alternative proposals but also seems to have advantages in performance relative to some. For graphical modeling of counts, our model outperforms the local Poisson model [1], and has the further advantage that it can additionally accommodate covariates and overdispersion. For factor analysis of binary data, our method also outperformed the lava.tobit package on R, although at the cost of increased computation time. An alternative approach we also considered was to perform covariance modeling on a single set of residuals from univariate models, but this seemed to lose considerable statistical efficiency.

We demonstrate our method with two well known covariance modeling frameworks, but it is simple to substitute other (possibly penalized) likelihood-based covariance modeling algorithms for Gaussian data. Also, there is also no reason that all the marginal distributions need be from the same family, nor do they need to all be discrete. In principle, all combinations of covariance modeling algorithms and marginal distributions are possible, and this is a key strength of our proposed method.



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7: Results of covariance models before (left) and after (right) controlling for covariates. Specifically, we present factor scores (a and b), and factor loadings (c and d) from a factor analytic model, and the resulting graphs (e and f) based on a graphical model. We observe clustering of sites in terms of these covariates in Figure (a), while these patterns are absent in Figure (b), after controlling for covariates. For the graphical model, before controlling for covariates in Figure (e) we observe that *Pardlugu* has negative interactions with both *Alopacce* and *Pardmont*, who interact positively.However these interactions are absent after controlling for covariates in Figure (f).

Appendix A. Proofs

Appendix A.1. Proof of Lemma 1

The distribution of the randomized Dunn–Smyth residuals given the data and marginal distributions is

$$g(\zeta_i) = \frac{\prod_{j=1}^{d} \phi(\zeta_{ij})}{\prod_{j=1}^{d} f_{ij}(y_{ij})} \mathbf{1}_{\zeta_i \in B_i}. \tag{A.1}$$

We can approximate the likelihood by importance sampling with K sets of Dunn–Smyth residuals

$$L_i(y_i|\beta, \psi, \theta) = \int_{B_i} \phi_d(z_i; R_\theta) dz_i = \int_{B_i} \phi_d(z_i; R_\theta) \frac{\prod_{j=1}^{d} f_{ij}(y_{ij})}{\prod_{j=1}^{d} \phi(z_{ij})} g(z_i) dz = \prod_{j=1}^{d} f_{ij}(y_{ij}) \int_{B_i} \frac{\phi_d(z_i; R_\theta)}{\prod_{j=1}^{d} \phi(z_{ij})} g(z_i) dz$$

which can be approximated using with K samples from g, viz.

$$L_i(y_i|\beta, \psi, \theta) \approx \prod_{j=1}^{d} f_{ij}(y_{ij}) \sum_{k=1}^{K} \frac{\phi_d(\zeta_i; R_\theta)}{\prod_{j=1}^{d} \phi(\zeta_{ij})} = \prod_{j=1}^{d} f_{ij}(y_{ij}) \sum_{k=1}^{K} c(\zeta^k; \Sigma)$$

where $\zeta_i \sim g(\zeta_i)$. □

Appendix A.2. Proof of equivalence to EM algorithm

Following Dauwels et al. [6], but in a frequentist framework, we have an EM algorithm with

$$\hat{\Sigma}^{m+1} = \arg\max_{\Sigma} Q(\Sigma, \hat{\Sigma}^m),$$

where

$$Q(\Sigma, \hat{\Sigma}^{(m)}) = \sum_{i=1}^{N} \int_{z_i} f(z_i|y_i; \hat{\Sigma}^{(m)}) \ln f(z_i; \Sigma) dz_i,$$

where z_i are the latent Gaussian vectors and y_i are the discrete data, both of dimension d. Now $f(z_i; \Sigma) = \phi_d(z_i; \Sigma)$ and

$$f(y_i|z_i; \Sigma) = f(y = y'|z = z'; \Sigma) = \begin{cases} 1 & \text{if } z_i \in A_i = \cap_j [\Phi^{-1}\{F(y_{ij}^-)\}, \Phi^{-1}\{F(y_{ij})\}], \\ 0 & \text{otherwise.} \end{cases}$$

And so $f(z_i|y_i; \Sigma) \propto f(y_i|z_i; \Sigma) f(z_i; \Sigma) = \mathbf{1}_{z_i \in A} \phi_d(z_i; \Sigma)$. This is the truncated multivariate normal distribution with covariance matrix Σ.

To carry out an MCEM algorithm we need to sample from $f(\zeta_i|y_i; \hat{\Sigma}^m)$ at the mth iteration. We do this by first sampling Dunn–Smyth residuals, whose distribution is a truncated multivariate normal with identity covariance matrix (see equation A.1), and then weight observations accordingly. So the weighted sample $(\zeta_i^k, w_{ik}(\Sigma^{(m)}))$ is distributed according to $f(\zeta_i|y_i; \Sigma^{(m)})$ where $\zeta_i^k \sim g(\zeta)$ are randomized Dunn–Smyth residuals, and

$$w'_{ik}(\Sigma^{(m)}) = \frac{f(\zeta_i^k|y_i; \Sigma^{(m)})}{g(\zeta_i^k)} \propto \frac{\phi_d(\zeta_i^k; \Sigma)}{\prod_j \phi(\zeta_{ij}^k)} = c(\zeta_i^k; \Sigma), \quad w_{ik}(\Sigma^{(m)}) = \frac{w'_{ik}(\Sigma^{(m)})}{\sum_k w'_{ik}(\Sigma^{(m)})} = \frac{c(\zeta_i^k; \Sigma_\theta)}{\sum_k c(\zeta_i^k; \Sigma_\theta)}$$

as in Eq. (4). So

$$Q(\Sigma, \hat{\Sigma}^{(m)}) = \sum_{i=1}^{N} \int_{z_i} f(z_i|y_i; \hat{\Sigma}^{(m)}) \ln f(z_i; \Sigma) dz_i \approx \sum_i \sum_k w_{ik}(\Sigma^{(m)}) \ln \phi_d(\zeta_i^k; \Sigma)$$

And hence for covariance parameters θ the derivative needed for maximization is given by

$$\frac{\partial}{\partial\theta} Q(\Sigma_\theta, \hat{\Sigma}_\theta^{(m)}) \approx \sum_i \sum_k w_{ik}(\Sigma_\theta^{(m)}) \frac{\partial}{\partial\theta} \ln \phi_d(\zeta_i^k; \Sigma_\theta)$$

This is the same form as Eq. (4). □

Appendix A.3. Proof of consistency

We aim to prove the consistency of estimates obtained by estimating a Gaussian copula model with discrete marginal distributions using Algorithm 1. We follow the standard proof of consistency for maximum likelihood found in Ferguson [11], for instance. The standard proof proceeds by defining $\tau(\theta)$, which is maximized at the maximum likelihood estimate (MLE) $\hat{\theta}$,

$$\tau(\theta) = \ln \frac{\ell_n(\theta)}{\ell_n(\theta_0)} = \frac{1}{n} \sum_{i=1}^{n} \ln \frac{f(y_i;\theta)}{f(y_i;\theta_0)},$$

where y_i is a d-vector of data corresponding to observation $i \in \{1, \dots, N\}$. This quantity then converges to its expectation under θ_0 by the Strong Law of Large Numbers, viz.

$$\frac{1}{n} \sum_{i=1}^{n} \ln \frac{f(y_i;\theta)}{f(y_i;\theta_0)} \xrightarrow{P} \mathrm{E}_{\theta_0} \left\{ \ln \frac{f(y;\theta)}{f(y,\theta_0)} \right\}.$$

This expectation is equal to the negative of the Kullback–Leibler divergence,

$$\mathrm{E}_{\theta_0} \left\{ \ln \frac{f(y;\theta)}{f(y,\theta_0)} \right\} = -K(\theta_0, \theta) < 0$$

unless $f(y,\theta) = f(y;\theta_0)$. Therefore the MLE maximizes $\tau(\theta)$ (assuming identifiability), which converges to a function which is maximized by θ_0, from which $\hat{\theta} \xrightarrow{P} \theta_0$ follows. A difficulty in our case is that we are not using MLEs for estimation — Algorithm 1 is a two-step estimation procedure, where we estimate β from a marginal likelihood and then maximize the conditional likelihood given these parameter estimates. We wish to show that treating β as nuisance parameters, we can get consistent estimates of parameters of R in the covariance model.

Conditions: We assume the following mild regularity conditions, where Conditions 1–6 equivalent to those found in Chapter 10 of Casella and Berger [5], for example.

1. The observations $y_i \sim f(y, \beta, R)$ for $i \in \{1, \dots, N\}$ are independent.
2. β is identifiable, i.e., if $\beta \neq \beta'$ then $f(y, \beta, R) \neq f(y, \beta', R)$.
3. The densities $f(y, \beta, R)$ have common support, and f is differentiable in β.
4. The parameter space Ω contains an open set ω of which the true parameter β_0 is an interior point.
5. For every y in $\mathcal{Y}$, the density $f(y, \beta, R)$ is continuous and at least three times differentiable in β, and $\int f(y, \beta, R) dy$ can be differentiated three times under the integral sign.
6. There exists an open subset of $\omega \in \Omega$ containing β_0 and an integrable function $M_r(y)$, such that for every $\beta \in \omega$ and $y \in \mathcal{Y}$, $|\partial^3 \ln f(y, \beta, R)/\partial^3 \beta_r| \leq M_r(y)$ for $r \in \{1, \dots, \dim(\beta)\}$, where $E_{\beta_0}\{M_r(y)\} < \infty$
7. For $r \in \{1, \dots, \dim(\beta)\}$ there are bounded functions $V_r(y)$ such that in the neighborhood of β_0 for any fixed R, $\{\partial \ln f(y_i, \beta, R)/\partial \beta_r\}^2 \leq V_r(y)$ with $\mathrm{E}_{\theta_0}\{V_r(y)\} < \infty$.

We proceed by defining the likelihood for $\theta = (\beta, R)$ as

$$\ell_n(\theta) = \ln \ell_n(\beta, R) = \frac{1}{n} \sum_{i=1}^{n} \ln f(t_i; \beta, R),$$

where β is the $d \times K$ matrix, with $\beta_{j,k}$ being the coefficient for the kth covariate regressed on the jth variable. Let $\theta_0 = (\beta_0, R_0)$ be the true parameters, and $\hat{\beta}$ be the matrix of coefficients where the jth row is found by maximizing the jth marginal likelihood, as in step 1 of Algorithm 1,

$$\hat{\beta}_j = \mathrm{argmax}_{\beta_j} \sum_{i=1}^{n} \ln L_j(y_j, \beta_j). \tag{A.2}$$

We now state a result, without proof, concerning the consistency of the marginal parameters.

Lemma 2. *Eq.* (A.2) *is equivalent to using independence estimating equations in the GEE framework, which under Conditions 1–6, are consistent* [23]*, so* $\hat{\beta} \xrightarrow{P} \beta_0$.

Analogously to the proof of standard maximum likelihood estimation, the value $\hat{R}$ found by Algorithm 1 maximizes $\tau'(R)$, where

$$\tau'(R) = \ln \frac{\ell_n(\hat{\beta}, R)}{\ell_n(\hat{\beta}, R_0)} = \frac{1}{n} \sum_{i=1}^{n} \ln \frac{f(y_i; \hat{\beta}, R)}{f(y_i; \hat{\beta}, R_0)}.$$

However, we cannot use the Law of Large Numbers directly to show this converges to its expectation under θ_0 as each summand of $\tau'(R)$ is a function of all the data, through $\hat{\beta}$. Instead, we develop the following result.

Lemma 3. $\ell_n(\hat{\beta}, R)/n \xrightarrow{P} \mathrm{E}_{\theta_0} \{\ln f(y, \beta_0, R)\}$ *as* $n \to \infty$.

Proof. Under Conditions 1–7 one has that, for any fixed R, the Taylor expansion of the standardized likelihood around β_0 is

$$\frac{1}{n}\, \ell_n(\hat{\beta}, R) = \frac{1}{n}\, \ell_n(\beta_0, R) + \frac{1}{n}\, (\hat{\beta} - \beta_0)^\top \frac{\partial}{\partial \beta}\, \ell_n(\beta, R)\Big|_{\tilde{\beta}}, \tag{A.3}$$

where $\tilde{\beta}$ is between $\hat{\beta}$ and β_0. By the Cauchy–Schwarz inequality, the last term is

$$\left\| \frac{1}{n}\, (\hat{\beta} - \beta_0)^\top \frac{\partial}{\partial \beta}\, \ell_n(\beta, R)\Big|_{\tilde{\beta}} \right\| \leq \frac{1}{n}\, \|\hat{\beta} - \beta_0\| \times \left\| \frac{\partial}{\partial \beta}\, \ell_n(\beta, R)\Big|_{\tilde{\beta}} \right\|.$$

By Lemma 2, we know $\|\hat{\beta} - \beta_0\| = o_p(1)$. We then look at the square of the last term, viz.

$$\left\| \left\{ \frac{\partial}{\partial \beta}\, \ell_n(\beta, R)\Big|_{\tilde{\beta}} \right\} \right\|^2 = \sum_{r=1}^{\dim(\beta)} \left\{ \sum_{i=1}^{n} \frac{\partial}{\partial \beta_r} \ln f(y_i, \beta, R)\Big|_{\tilde{\beta}} \right\}^2 = O_P(n^2),$$

which follows from the regularity conditions. Hence

$$\left\| \frac{\partial}{\partial \beta}\, \ell(\beta, R)\Big|_{\tilde{\beta}} \right\| = O_P(n),$$

So the remainder term in Eq. (A.3) is given by

$$\left\| \frac{1}{n} (\hat{\beta} - \beta_0)^\top \frac{\partial}{\partial \beta}\, \ell(\beta, R)\Big|_{\tilde{\beta}} \right\| \leq \frac{1}{n}\, \|\hat{\beta} - \beta_0\| \times \left\| \frac{\partial}{\partial \beta}\, \ell_n(\beta, R)\Big|_{\tilde{\beta}} \right\| = \frac{1}{n}\, o_P(1) O_P(n) = o_P(1).$$

This in turn implies

$$\frac{1}{n}\, \ell_n(\hat{\beta}, R) = \frac{1}{n}\, \ell_n(\beta_0, R) + \frac{1}{n}\, (\hat{\beta} - \beta_0)^\top \frac{\partial}{\partial \beta}\, \ell_n(\beta, R)\Big|_{\tilde{\beta}} = \frac{1}{n}\, \ell_n(\beta_0, R) + o_P(1).$$

Hence for any R, $\ell_n(\hat{\beta}, R)/n \xrightarrow{P} \mathrm{E}_{\theta_0} \{\ln f(y, \beta_0, R)\}$. □

Now we can return to the standard proof. We have

$$\tau'(R) = \ln \frac{\ell_n(\hat{\beta}, R)}{\ell_n(\hat{\beta}, R_0)} = \frac{1}{n} \sum_{i=1}^{n} \ln \frac{f(y_i; \hat{\beta}, R)}{f(y_i; \hat{\beta}, R_0)} \xrightarrow{P} \mathrm{E}_{\theta_0} \left\{ \ln \frac{f(y; \beta_0, R)}{f(y, \beta_0, R_0)} \right\} = -K(\theta_0, \theta) < 0$$

unless $f(y, \theta) = f(y; \theta_0)$, and so $\hat{\theta} \xrightarrow{P} \theta_0$ and hence $\hat{R} \xrightarrow{P} R_0$. □

Acknowledgments. The authors thank the Editor and referees for helpful comments.

References

[1] G. Allen, Z. Liu, A local Poisson graphical model for inferring networks from sequencing data, NanoBioscience, IEEE Trans. 12 (2013) 189–198.

[2] O. Banerjee, L.E. Ghaoui, A. d'Aspremont, G. Natsoulis, Convex optimization techniques for fitting sparse Gaussian graphical models, In: Proceedings of the 23rd International Conference on Machine Learning, pp. 89–96, 2006.

[3] J.G. Booth, J.P. Hobert, Maximizing generalized linear mixed model likelihoods with an automated Monte Carlo EM algorithm, J. Roy. Statist. Soc. Ser. B 61 (1999) 265–285.

[4] C.M. Carvalho, J. Chang, J.E. Lucas, J.R. Nevins, Q. Wang, M. West, High-dimensional sparse factor modeling: Applications in gene expression genomics, J. Amer. Statist. Assoc. 103 (2008) 1438–1456.

[5] G. Casella, R.L. Berger, Statistical Inference. Vol. 2, Duxbury, Pacific Grove, CA, 2002.

[6] J. Dauwels, H. Yu, S. Xu, X. Wang, Copula Gaussian graphical model for discrete data, In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6283–6287, 2013.

[7] A.P. Dempster, N.M. Laird, D.B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, J. Roy. Statist. Soc. Ser. B 39 (1977) 1–38.

[8] P.K. Dunn, G.K. Smyth, Randomized quantile residuals, J. Comput. Graphical Statist. 5 (1996) 236–244.

[9] B.S. Everitt, An Introduction to Latent Variable Models, Springer, New York, 1984.

[10] J. Fan, H. Liu, Y. Ning, H. Zou, High dimensional semiparametric latent graphical model for mixed data, J. Roy. Statist. Soc. Ser. B 79 (2017) 405–421.

[11] T.S. Ferguson, A course in Large Sample Theory, Chapman & Hall, London, 1996.

[12] T.S. Ferguson, Mathematical Statistics: A Decision Theoretic Approach, Academic Press, New York, 1967.

[13] J. Friedman, T. Hastie, R.J. Tibshirani, Sparse inverse covariance estimation with the graphical lasso, Biostatistics 9 (2008) 432–441.

[14] J. Guo, E. Levina, G. Michailidis, J. Zhu, Graphical models for ordinal data, J. Computat. Graphical Statist. 24 (2015) 183–204.

[15] R.K. Hambleton, Fundamentals of Item Response Theory, Sage Publications, New York, 1991.

[16] A. Heinen, E. Rengifo, Multivariate autoregressive modeling of time series count data using copulas, J. Empirical Finance 14 (2007) 564–583.

[17] K.K. Holst, `lava.tobit`: LVM with censored and binary outcomes, R Package Version 0.4-7, 2012.

[18] K.K. Holst, E. Budtz-Jørgensen, Linear latent variable models: The `lava`-package, Comput. Statist. 28 (2013) 1385–1452.

[19] H. Joe, Asymptotic efficiency of the two-stage estimation method for copula-based models, J. Multivariate Anal. 94 (2005) 401–419.

[20] O. Ledoit, M. Wolf, Honey, I shrunk the sample covariance matrix, UPF Economics and Business Working Paper (691), 2003.

[21] J.D. Lee, T.J. Hastie, Learning the structure of mixed graphical models, J. Comput. Graphical Statist. 24 (2015) 230–253.

[22] R.A. Levine, G. Casella, Implementations of the Monte Carlo EM algorithm, J. Comput. Graphical Statist. 10 (2001) 422–439.

[23] K.-Y. Liang, S.L. Zeger, Longitudinal data analysis using generalized linear models, Biometrika 73 (1986)13–22.

[24] H. Liu, J. Lafferty, L.A. Wasserman, The nonparanormal: Semiparametric estimation of high dimensional undirected graphs, J. Machine Learning Res. 10 (2009) 2295–2328.

[25] H. Liu, K. Roeder, L.A. Wasserman, Stability approach to regularization selection (StARS) for high dimensional graphical models, In: J.D. Lafferty, C.K.I. Williams, J. Shawe-Taylor, R.S. Zemel, A. Culotta (Eds.), Advances in Neural Information Processing Systems 23, Curran Associates, Inc., pp. 1432–1440, 2010.

[26] G. Masarotto, C. Varin, Gaussian copula marginal regression, Electr. J. Statist. 6 (2012) 1517–1549.

[27] G.J. McLachlan, T. Krishnan, The EM algorithm and extensions, Wiley, New York, 1997.

[28] N. Meinshausen, P. Bühlmann, High-dimensional graphs and variable selection with the lasso, Ann. Statist. 34 (2006) 1436–1462.

[29] Z. Meng, B. Eriksson, A. Hero, Learning latent variable Gaussian graphical models, In: T. Jebara, E.P. Xing (Eds.), Proceedings of the 31st International Conference on Machine Learning (ICML-14), JMLR Workshop and Conference Proceedings, pp. 1269–1277, 2014.

[30] J. Nešlehová, On rank correlation measures for non-continuous random variables, J. Multivariate Anal. 98 (2007) 544–567.

[31] A.K. Nikoloulopoulos, Copula-based models for multivariate discrete response data, In: P. Jaworski, F. Durante, W.K. Härdle, Copulæ in Mathematical and Quantitative Finance: Proceedings of the Workshop Held in Cracow, 10–11 July 2012. Springer, Berlin Heidelberg, pp. 231–249, 2013.

[32] A.K. Nikoloulopoulos, On the estimation of normal copula discrete regression models using the continuous extension and simulated likelihood, J. Statist. Plann. Inference 143 (2013) 1923–1937.

[33] A.K. Nikoloulopoulos, D. Karlis, Modeling multivariate count data using copulas, Comm. Statist. Simul. Comput. 39 (2009) 172–187.

[34] P. Ravikumar, W.J. Wainwright, L.D. Lafferty, High-dimensional Ising model selection using L_1 regularized logistic regression, Ann. Statist. 38 (2010) 1287–1319.

[35] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2014.

[36] A.J. Rothman, P.J. Bickel, E. Levina, J. Zhu, Sparse permutation invariant covariance estimation, Electr. J. Statist. 2 (2008) 494–515.

[37] A. Skrondal, S. Rabe-Hesketh, Generalized Latent Variable Modeling: Multilevel, Longitudinal, and Structural Equation Models, CRC Press, London, 2004.

[38] P. van Der Aart, N. Smeenk-Enserink, Correlations between distributions of hunting spiders (*Lycosidae, Ctenidæ*) and environmental characteristics in a dune area, Nether. J. Zool. 25 (1974) 1–45.

[39] Y. Wang, U. Naumann, S. Wright, D. Warton, `mvabund`: Statistical methods for analysing multivariate abundance data, R Package Version 3.8.0, 2012.

[40] G.C. Wei, M.A. Tanner, A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithms, J. Amer. Statist. Assoc. 85 (1990) 699–704.

[41] M. Yuan, Y. Lin, Model selection and estimation in the Gaussian graphical model, Biometrika 94 (2007) 19–35.

[42] T. Zhao, H. Liu, K. Roeder, J. Lafferty, L.A. Wasserman, The `huge` package for high-dimensional undirected graph estimation in R, J. Machine Learning Res. 13 (2012) 1059–1062.