Two-way Sparse Network Inference for Count Data

Sijia Li
University of Leeds
sijia.li.ldn@gmail.com

Martín López-García
University of Leeds
M.LopezGarcia@leeds.ac.uk

Neil D. Lawrence
University of Cambridge
ndl21@cam.ac.uk

Luisa Cutillo*
University of Leeds
L.Cutillo@leeds.ac.uk
*Corresponding author

Abstract

Classically, statistical datasets have a larger number of data points than features ($n > p$). The standard model of classical statistics caters for the case where data points are considered conditionally independent given the parameters. However, for $n \approx p$ or $p > n$ such models are poorly determined. Kalaitzis et al. (2013) introduced the Bigraphical Lasso, an estimator for sparse precision matrices based on the Cartesian product of graphs. Unfortunately, the original Bigraphical Lasso algorithm is not applicable in case of large p and n due to memory requirements. We exploit eigenvalue decomposition of the Cartesian product graph to present a more efficient version of the algorithm which reduces memory requirements from $O(n^2p^2)$ to $O(n^2 + p^2)$. Many datasets in different application fields, such as biology, medicine and social science, come with count data, for which Gaussian based models are not applicable. Our multiway network inference approach can be used for discrete data.

Our methodology accounts for the dependencies across both instances and features, reduces the computational complexity for high dimensional data and enables to deal with both discrete and continuous data. Numerical studies on both synthetic and real datasets are presented to showcase the performance of our method.

Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS) 2022, Valencia, Spain. PMLR: Volume 151.

1 INTRODUCTION

In this research, we develop a tensor-decomposition two-way network inference approach for count data. Firstly, we present a Scalable Bigraphical Lasso algorithm, reducing both the space complexity and the computational complexity of the inference. Secondly, we extend the Bigraphical model to count data by means of a semiparametric approach. Our proposed methodology not only accounts for the dependencies across both instances and features, but also reduces the computational complexity for high dimensional data.

The main motivation of this research is that real world problems often come with correlations between several dimensions. Recently, Gaussian graphical models have been developed with tensor decomposition for multiway network inference. For example, Tsiligkaridis and Hero (2013) and Zhou (2014) studied a matrix normal distribution where the precision matrix corresponds to the Kronecker product between the row-specific and the column-specific precision matrices. Kalaitzis et al. (2013) introduced Bigraphical Lasso, and Greenewald et al. (2019) introduced TeraLasso, both studying a multivariate normal distribution where the precision matrix corresponds to a Kronecker sum instead.

Many datasets in different application fields come with count data, for which Gaussian based models are not applicable. Some methods use other distributions to infer network from the data. Jia et al. (2017) infers the gene regulation networks with a Poisson-Gamma based Bayesian Hierarchical Model, borrowing information across cells. McDavid et al. (2019) infers the gene regulation networks with a multivariate Hurdle model (zero-inflated mixed Gaussian). Several approaches have extended the use of Gaussian models to an appropriate continuous transformation of count data. Liu et al. (2009) and Liu et al. (2012) proposed a semiparametric approach, and Roy and Dunson (2020) proposed a nonparametric approach, while Chiquet et al. (2019) considered Bayesian Hierarchical Models. However, all these methods only produce a one-way network infer-

ence. Bartlett et al. (2021) proposed a Bayesian model with a prior having decoupled two-way sparsity to infer a dynamic network structure through time, however, the method still depends on a pre-inferred or known ordering of time. Our method extends the Gaussian Copula transformation to enable a two-way network inference, where the structure in both dimensions is to be inferred simultaneously.

This paper is structured as follows: In Section 2 we present a detailed review on relevant literature; In Section 3 we present our Scalable Bigraphical Lasso algorithm for Gaussian data; In Section 4 we propose a semiparametric extension to the Bigraphical model for count data; In Section 5 we showcase the performance of our method on both synthetic and real datasets.

2 BACKGROUND

2.1 From the matrix normal model to the Kronecker sum structure

For a Gaussian density, a sparse precision matrix defines a weighted undirected graph in Gaussian Markov random field relationship (Lauritzen, 1996), encoding conditional independence between variables in the Gaussian model. Therefore we can induce the network structure from the support of the precision matrix.

A matrix normal model with the Kronecker sum structure was proposed in Kalaitzis et al. (2013). If a $p \times n$ random matrix $\mathbf{Y}$ follows a matrix normal distribution,

$$\mathbf{Y} \sim \mathbf{MN}_{p\times n}\left(\mathbf{M}; \mathbf{\Psi}_{n\times n}^{-1}, \mathbf{\Theta}_{p\times p}^{-1}\right),$$

with $\mathbf{M}$ a $p \times n$ matrix, and with precision matrix $\mathbf{\Psi}_{n\times n}$ indicating the dependency structure in rows, and precision matrix $\mathbf{\Theta}_{p\times p}$ indicating the dependency structure in columns, the model can be reparametrized such that the vectorised random matrix follows a np-dimensional multivariate normal distribution (denoted as $\mathbf{mN}$):

$$\mathrm{vec}\left(\mathbf{Y}\right) \sim \mathbf{mN}_{np}\left(\mathbf{0}_{np}, \left(\mathbf{\Psi}_{n\times n} \otimes \mathbf{\Theta}_{p\times p}\right)^{-1}\right),$$

where $\otimes$ denotes the Kronecker product (KP), $\mathbf{\Psi}_{n\times n} \otimes \mathbf{\Theta}_{p\times p}$ is the overall precision matrix, and $\mathbf{0}_{np}$ is a column vector of zeros of length np. Kalaitzis et al. (2013) proposed to use the Kronecker sum (KS) $\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p} = \mathbf{\Psi}_{n\times n} \otimes I_p + I_n \otimes \mathbf{\Theta}_{p\times p}$ to structure the overall precision matrix. In a KS-structured matrix normal distribution, for a $p \times n$ random matrix $\mathbf{Y}$, we write

$$\mathrm{vec}\left(\mathbf{Y}\right) \sim \mathbf{mN}_{np}\left(\mathbf{0}_{np}, \left(\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}\right)^{-1}\right).$$

The KS-structure has several advantages. Firstly, in algebraic graph theory, the Kronecker sum corresponds to the Cartesian product of graphs (Sabidussi, 1959). A KS-structured model therefore provides intuitive and interpretable results. Secondly, for high-dimensional data, the KS-structure enhances the sparsity of the network, reducing the computation complexity and the memory requirement.

2.2 Rank-based estimation in a Gaussian graphical model

To model count data or other non-Gaussian data in a Gaussian graphical model, the Gaussian copula can be applied to transfer these data into a latent Gaussian variable. Liu et al. (2012) proposed a semiparametric Gaussian copula for one-way network inference. For a $p \times n$ matrix $\mathbf{Y}$, Liu et al. (2012) considered it as n samples of a $p-$dimensional vector $(Y_{1j}, \ldots, Y_{pj})$. Liu et al. (2012) assumed that there exist functions $f = \{f_i\}_{i=1}^{p}$ such that for $j = 1, \ldots, n$:

$$\left(f_1\left(Y_{1j}\right), \ldots, f_p\left(Y_{pj}\right)\right) \sim \mathbf{mN}_p\left(\mathbf{0}_p, \mathbf{\Theta}_{p\times p}^{-1}\right),$$

where $\mathbf{\Theta}_{p\times p}$ is an unknown precision matrix. In this case $Y_j = (Y_{1j}, \ldots, Y_{pj})$ is said to follow a nonparanormal multivariate normal distribution, $Y_j \sim \mathbf{NPN}\left(\mathbf{0}_p, \mathbf{\Theta}_{p\times p}^{-1}, f\right)$. Then they inferred the precision matrix $\mathbf{\Theta}_{p\times p}$ with the following objective function from *graphical lasso* (Friedman et al., 2008):

$$\min_{\mathbf{\Theta}_{p\times p}} \left\{ \mathrm{tr}\left(\mathbf{\Theta}_{p\times p}\mathbf{S}\right) - \log\left|\mathbf{\Theta}_{p\times p}\right| + \beta \sum_{i_1, i_2} \mathbf{\Theta}_{i_1 i_2} \right\},$$

where $\mathbf{S}$ is the empirical covariance matrix of $\left(f_1\left(Y_{1j}\right), \ldots, f_p\left(Y_{pj}\right)\right)$, $j = 1, \ldots, n$ in *graphical lasso*, and β is the regularization parameter controlling sparsity. Liu et al. (2012) used the estimated correlation matrix $\hat{\mathbf{S}}$ instead of $\mathbf{S}$, estimated using Kendall's tau or Spearman's rho. In particular, one defines $\Delta_i(j, j') = Y_{ij} - Y_{ij'}$, so that

(Kendall's tau)

$$\hat{\tau}_{i_1 i_2} = \frac{2}{n\left(n-1\right)} \sum_{j<j'} \mathrm{sign}\left(\Delta_{i_1}(j, j')\Delta_{i_2}(j, j')\right),$$

(Spearman's rho)

$$\hat{\rho}_{i_1 i_2} = \frac{\sum_{j=1}^{n}\left(r_{i_1 j}^{(c)} - \bar{r}_j^{(c)}\right)\left(r_{i_2 j}^{(c)} - \bar{r}_j^{(c)}\right)}{\sqrt{\sum_{j=1}^{n}\left(r_{i_1 j}^{(c)} - \bar{r}_j^{(c)}\right)^2\left(r_{i_2 j}^{(c)} - \bar{r}_j^{(c)}\right)^2}},$$

where $r_{ij}^{(c)}$ is the rank of Y_{ij} among $Y_{1j}, \ldots, Y_{pj}$ and $\bar{r}_j^{(c)} = \frac{1}{p}\sum_{i=1}^{p} r_{ij}^{(c)} = \frac{1+p}{2}$. Correspondingly,

$$\hat{\mathbf{S}}_{i_1 i_2} = \begin{cases} \sin\left(\frac{\pi}{2}\hat{\tau}_{i_1 i_2}^{(c)}\right), & i_1 \neq i_2, \\ 1, & i_1 = i_2. \end{cases}$$

$$\hat{\mathbf{S}}_{i_1 i_2} = \begin{cases} 2\sin\left(\frac{\pi}{6}\hat{\rho}^{(c)}_{i_1 i_2}\right), & i_1 \neq i_2, \\ 1, & i_1 = i_2. \end{cases}$$

Ning and Liu (2013) extended the matrix-normal distribution with Kronecker product structure to non-Gaussian data with a similar semiparametric approach applied on both the row vectors and the column vectors of $\mathbf{Y}$.

2.3 Background on Bigraphical lasso

Bigraphical Lasso is introduced by Kalaitzis et al. (2013). Let $\mathbf{Y} \in \mathbb{R}^{n\times p}$ be a random matrix. If its rows are generated as i.i.d. samples from $N(0, \mathbf{\Sigma}_{p\times p})$, then the sampling distribution of the sufficient statistic $\mathbf{Y}^\top\mathbf{Y}$ is Wishart $(n, \mathbf{\Sigma}_{p\times p})$. At the same time, if the columns are generated as *i.i.d.* samples from $\mathcal{N}(0, \mathbf{\Gamma}_{p\times p})$, then the sampling distribution is Wishart $(n, \mathbf{\Gamma}_{p\times p})$. Combining these sufficient statistics in a model for the entire matrix $\mathbf{Y}$ as

$$p(\mathbf{Y}) \propto \exp\{-\mathrm{tr}\left(\mathbf{\Psi}_{n\times n}\mathbf{Y}\mathbf{Y}^\top\right) - \mathrm{tr}\left(\mathbf{\Theta}_{p\times p}\mathbf{Y}^\top\mathbf{Y}\right)\}$$

is equivalent to a joint factorised Gaussian distribution for the entries of $\mathbf{Y}$, with a precision matrix given by the KS:

$$\mathbf{\Omega} = \mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p} = \mathbf{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \mathbf{\Theta}_{p\times p}. \quad (1)$$

Through this representation we obtain a parameter vector of size $O\left(n^2 + p^2\right)$ instead of the usual $O\left(n^2p^2\right)$.

Given data in the form of some design matrix $\mathbf{Y}$, the Bigraphical Lasso model proposed in Kalaitzis et al. (2013) estimates the sparse KS-structured inverse covariance of a matrix normal by minimising the ℓ_1-penalized negative likelihood function of $(\mathbf{\Psi}_{n\times n}$, $\mathbf{\Theta}_{p\times p})$:

$$\min_{\mathbf{\Theta}_{p\times p}, \mathbf{\Psi}_{n\times n}} \Big\{ n\mathrm{tr}\left(\mathbf{\Theta}_{p\times p}\mathbf{S}\right) + p\mathrm{tr}\left(\mathbf{\Psi}_{n\times n}\mathbf{T}\right) - \log\left|\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}\right| + \beta_1||\mathbf{\Psi}_{n\times n}||_1 + \beta_2||\mathbf{\Theta}_{p\times p}||_1 \Big\}, \quad (2)$$

where $\mathbf{S} \triangleq \frac{1}{n}\mathbf{Y}^\top\mathbf{Y}$ and $\mathbf{T} \triangleq \frac{1}{p}\mathbf{Y}\mathbf{Y}^\top$ are empirical covariances across the samples and features respectively. A solution simultaneously estimates two graphs — one over the columns of $\mathbf{Y}$, corresponding to the sparsity pattern of $\mathbf{\Theta}_{p\times p}$, and another over the rows of $\mathbf{Y}$, corresponding to the sparsity pattern of $\mathbf{\Psi}_{n\times n}$.

The original paper of Kalaitzis et al. (2013) proposes a *flip-flop* approach first optimizing over $\mathbf{\Psi}_{n\times n}$, while holding $\mathbf{\Theta}_{p\times p}$ fixed, and then optimizing over $\mathbf{\Theta}_{p\times p}$ while holding $\mathbf{\Psi}_{n\times n}$ fixed. They show that in case of no regularization, the first step of the optimization problem is reduced to

$$\min_{\mathbf{\Psi}_{n\times n}} \Big\{ p\mathrm{tr}\left(\mathbf{\Psi}_{n\times n}\mathbf{T}\right) - \ln|\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}| \Big\}.$$

Obtaining the stationary point:

$$\mathbf{T} - \tfrac{1}{2p}\mathbf{T}\circ\mathbf{I} = \tfrac{1}{p}\mathrm{tr}_p\left(\mathbf{W}\right) - \tfrac{1}{2p}\mathrm{tr}_p\left(\mathbf{W}\right)\circ\mathbf{I}\,, \quad (3)$$

where $\circ$ is the Hadamard product and we define $\mathbf{W} \triangleq \left(\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}\right)^{-1}$. The block-wise trace $\mathrm{tr}_p\left(\cdot\right)$ is an operator that to each $np \times np$ matrix $\mathbf{M}$ written in terms of n^2 many $p \times p$ blocks

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{11} & \dots & \mathbf{M}_{1n} \\ \vdots & \ddots & \vdots \\ \mathbf{M}_{n1} & \dots & \mathbf{M}_{nn} \end{bmatrix},$$

associates the matrix of traces of each $p \times p$ block:

$$\mathrm{tr}_p\left(\mathbf{M}\right) = \begin{bmatrix} \mathrm{tr}\left(\mathbf{M}_{11}\right) & \dots & \mathrm{tr}\left(\mathbf{M}_{1n}\right) \\ \vdots & \ddots & \vdots \\ \mathrm{tr}\left(\mathbf{M}_{n1}\right) & \dots & \mathrm{tr}\left(\mathbf{M}_{nn}\right) \end{bmatrix},$$

as defined in Kalaitzis et al. (2013). While their approach dramatically reduces the computational complexity of the problem, its memory requirements (i.e. space complexity) are prohibitive for problems involving large n or p.

Our contribution in Section 3 is to give a more efficient solution in terms of computational and space complexity.

3 SCALABLE BIGRAPHICAL LASSO ALGORITHM

Consider the eigen-decomposition of the two precision matrices $\mathbf{\Psi}_{n\times n} = \mathbf{U}\Lambda_1\mathbf{U}^\top$ and $\mathbf{\Theta}_{p\times p} = \mathbf{V}\Lambda_2\mathbf{V}^\top$, where $\Lambda_1 \in \mathbb{R}^{n\times n}$ and $\Lambda_2 \in \mathbb{R}^{p\times p}$ are eigenvalues diagonal matrices and $\mathbf{U} = (u_{ij}) \in \mathbb{R}^n$ and $\mathbf{V} = (v_{ij}) \in \mathbb{R}^p$ are orthogonal eigenvectors matrices associated respectively to $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$. It follows that Equation (1) can be rewritten as

$$\mathbf{\Omega} = \left(\mathbf{U}\otimes\mathbf{V}\right)\left[\Lambda_1\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\Lambda_2\right]\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right). \quad (4)$$

Inversion of a symmetric matrix for which an eigenvalue decomposition is provided is achieved through inversion of the eigenvalues,

$$\mathbf{W} = \mathbf{\Omega}^{-1} = \left(\mathbf{U}\otimes\mathbf{V}\right)\left[\Lambda_1\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\Lambda_2\right]^{-1}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right).$$

Taking

$$\left(\mathbf{I}_n\otimes\mathbf{V}^\top\right)\left(\mathbf{I}_n\otimes\mathbf{I}_p\right) = \mathbf{I}_n\otimes\mathbf{V}^\top,$$

then

$$\mathbf{W}\boldsymbol{\Omega} = \mathbf{I}_n \otimes \mathbf{I}_p \tag{5}$$

can be premultiplied by $\mathbf{I}_n \otimes \mathbf{V}^\top$ to provide

$$\left(\mathbf{I}_n \otimes \mathbf{V}^\top\right)\mathbf{W}\boldsymbol{\Omega} = \left(\mathbf{U} \otimes \mathbf{I}_p\right)\mathbf{D}\left(\mathbf{U}^\top \otimes \mathbf{V}^\top\right)\boldsymbol{\Omega}, \tag{6}$$

where $\mathbf{D} = [\Lambda_1 \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \Lambda_2]^{-1}$ is a diagonal matrix. The detailed proof of Eq. (4) and Eq. (6) can be found in the Supplementary Material A.1. Multiply both sides of Equation (6) by $\mathbf{I}_n \otimes \mathbf{V}$, we have

$$\mathbf{I}_n \otimes \mathbf{I}_p = \left(\mathbf{U} \otimes \mathbf{I}_p\right)\mathbf{D}\left(\mathbf{U}^\top \otimes \mathbf{I}_p\right)\left(\boldsymbol{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \Lambda_2\right), \tag{7}$$

Detailed proof of Eq. (7) can be found in the Supplementary Material A.2. Eq (7) can be rewritten in a similar form as Equation (5)

$$\hat{\mathbf{W}}\hat{\boldsymbol{\Omega}} = \mathbf{I}_n \otimes \mathbf{I}_p,$$

where

$$\hat{\mathbf{W}} = \left[\mathbf{U} \otimes \mathbf{I}_p\right]\mathbf{D}\left[\mathbf{U}^\top \otimes \mathbf{I}_p\right]$$

and

$$\hat{\boldsymbol{\Omega}} = \boldsymbol{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \Lambda_2.$$

We partition $\hat{\mathbf{W}}$ and $\hat{\boldsymbol{\Omega}}$ into blocks

$$\hat{\mathbf{W}} = \begin{bmatrix} \hat{\mathbf{W}}_{11} & \hat{\mathbf{W}}_{1\backslash 1} \\ \hat{\mathbf{W}}_{\backslash 11} & \hat{\mathbf{W}}_{\backslash 1\backslash 1} \end{bmatrix},$$

$$\hat{\boldsymbol{\Omega}} = \begin{bmatrix} \hat{\boldsymbol{\Omega}}_{11} & \hat{\boldsymbol{\Omega}}_{1\backslash 1} \\ \hat{\boldsymbol{\Omega}}_{\backslash 11} & \hat{\boldsymbol{\Omega}}_{\backslash 1\backslash 1} \end{bmatrix},$$

where $\hat{\mathbf{W}}_{11}$ and $\hat{\boldsymbol{\Omega}}_{11}$ are $p \times p$ matrices and $\hat{\mathbf{W}}_{\backslash 11}$ and $\hat{\boldsymbol{\Omega}}_{\backslash 11}$ are $p\,(n-1) \times p$ matrices. Then from the bottom-left block of

$$\hat{\mathbf{W}}\hat{\boldsymbol{\Omega}} = \hat{\mathbf{W}}\left(\boldsymbol{\Psi}_{n\times n} \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \Lambda_2\right) = \mathbf{I}_n \otimes \mathbf{I}_p \tag{8}$$

we get

$$\hat{\mathbf{W}}_{\backslash 11}\left(\psi_{11}\mathbf{I}_p + \Lambda_2\right) + \hat{\mathbf{W}}_{\backslash 1\backslash 1}\left(\boldsymbol{\psi}_{\backslash 11} \otimes \mathbf{I}_p\right) = \mathbf{0}_{n-1} \otimes \mathbf{I}_p,$$

where we use notation $\boldsymbol{\Psi}_{n\times n} = \left(\psi_{ij}\right)_{i,j=1,\ldots,n}$ and $\boldsymbol{\psi}_{\backslash 11}$ represents the corresponding sub-block. Post multiplying both sides of the last equation by $\left(\psi_{11}\mathbf{I}_p + \Lambda_2\right)^{-1}$ we have

$$\hat{\mathbf{W}}_{\backslash 11} + \hat{\mathbf{W}}_{\backslash 1\backslash 1}\begin{bmatrix} \left(\psi_{11}\mathbf{I}_p + \Lambda_2\right)^{-1}\psi_{21} \\ \vdots \\ \left(\psi_{11}\mathbf{I}_p + \Lambda_2\right)^{-1}\psi_{n1} \end{bmatrix} = \mathbf{0}_{n-1} \otimes \mathbf{I}_p. \tag{9}$$

Detailed proof of Eq. (9) can be found in the Supplementary Material A.3.

Decomposing $\hat{\mathbf{W}}_{\backslash 1\backslash 1}$ in $(n-1)$ adjacent blocks $\hat{\mathbf{W}}_{\backslash 1k} \in \mathbb{R}^{(n-1)p\times p}$, $\forall k \in \{2,\ldots,n\}$, then Equation (9) can be rewritten as

$$\hat{\mathbf{W}}_{\backslash 11} + \hat{\mathbf{W}}_{\backslash 12}\left(\psi_{11}\mathbf{I}_p + \Lambda_2\right)^{-1}\psi_{21} + \ldots \\ \cdots + \hat{\mathbf{W}}_{\backslash 1n}\left(\psi_{11}\mathbf{I}_p + \Lambda_2\right)^{-1}\psi_{n1} = \mathbf{0}_{n-1} \otimes \mathbf{I}_p.$$

Proposition 3.1 Following the assumptions and calculations above we have

$$\mathrm{tr}_p\left(\mathbf{W}\right) = \mathrm{tr}_p\left(\hat{\mathbf{W}}\right).$$

The proof of Proposition 3.1 is in the Supplementary Material. Proposition 3.1 enables us to make use of the stationary point given in Equation (3). As described in Kalaitzis et al. (2013), we can partition the empirical covariance $\mathbf{T}$ as

$$\mathbf{T} = \begin{bmatrix} \mathbf{t}_{11} & \mathbf{t}_{1\backslash 1} \\ \mathbf{t}_{\backslash 11} & \mathbf{T}_{\backslash 1\backslash 1} \end{bmatrix},$$

where $\mathbf{t}_{\backslash 11} \in \mathbb{R}^{n-1}$ and $\mathbf{T}_{\backslash 1\backslash 1} \in \mathbb{R}^{(n-1)\times(n-1)}$. In particular, from the lower left block of (3) we get

$$\mathbf{t}_{\backslash 11} = \frac{1}{p}\mathrm{tr}_p\left(\mathbf{W}_{\backslash 11}\right).$$

Taking the block-wise trace $\mathrm{tr}_p\left(\cdot\right)$ of both sides of (9), gives

$$p\mathbf{t}_{\backslash 11} + \mathbf{A}_{\backslash 1\backslash 1}\boldsymbol{\psi}_{\backslash 11} = \mathbf{0}_{n-1}, \tag{10}$$

where $\mathbf{A}^\top_{\backslash 1\backslash 1} \in \mathbb{R}^{(n-1)\times(n-1)}$ is:

$$\mathbf{A}^\top_{\backslash 1\backslash 1} \triangleq \begin{bmatrix} \mathrm{tr}_p\left\{\hat{\mathbf{W}}_{\backslash 12}\left(\psi_{11}\mathbf{I}_p + \boldsymbol{\Lambda}_2\right)^{-1}\right\}^\top \\ \vdots \\ \mathrm{tr}_p\left\{\hat{\mathbf{W}}_{\backslash 1n}\left(\psi_{11}\mathbf{I}_p + \boldsymbol{\Lambda}_2\right)^{-1}\right\}^\top \end{bmatrix}. \tag{11}$$

The problem posed in Equation (10) is addressed via a lasso regression. In Proposition 3.2 we use some of the previous decomposition in order to reduce the computational complexity of the problem.

Proposition 3.2 Following the assumptions and calculations above we have

$$\mathrm{tr}_p\left\{\hat{\mathbf{W}}_{\backslash 1k}\left(\psi_{11}\mathbf{I}_p + \boldsymbol{\Lambda}_2\right)^{-1}\right\} \\ = \sum_{j=1}^{p}\frac{1}{\psi_{11} + \lambda_{2j}}\begin{bmatrix} \sum_{i=1}^{n}\frac{u_{2i}u_{ki}}{\lambda_{1i}+\lambda_{21}} \\ \vdots \\ \sum_{i=1}^{n}\frac{u_{ni}u_{ki}}{\lambda_{1i}+\lambda_{2p}} \end{bmatrix},$$

where $\lambda_{11}\ldots\lambda_{1n}$ and $\lambda_{21}\ldots\lambda_{2p}$ are the diagonal values of $\Lambda_1 \in \mathbb{R}^{n\times n}$ and $\Lambda_2 \in \mathbb{R}^{p\times p}$, respectively. The proof of Proposition 3.2 is in the Supplementary Material.

We note that by imposing an ℓ_1 penalty on $\boldsymbol{\Psi}_{\backslash 11}$, the problem posed in (10) reduces to a lasso regression involving now only the matrix $\mathbf{U}$, the diagonal of $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$, and ψ_{11}. This decomposition frees the prohibitive amount of memory needed to store the matrix $\hat{\mathbf{W}}$, which is of size n^2p^2.

The lasso regression will provide an estimation on the first column of $\boldsymbol{\Psi}_{n\times n}$. For the update of all the other

columns $\boldsymbol{\Psi}_{\setminus ii}$ we need to reiterate the same approach. Indeed we partition $\boldsymbol{\Psi}_{n\times n}$ into $\psi_{ii}, \boldsymbol{\psi}_{\setminus ii}$ and $\boldsymbol{\Psi}_{\setminus i\setminus i}$ for $i = 1, \ldots, n$. We then find a sparse solution of $p\mathbf{t}_{\setminus ii} + \mathbf{A}_{\setminus i\setminus i}\boldsymbol{\psi}_{\setminus ii} = \mathbf{0}_{n-1}$ with *lasso* regression. Given the new value $\boldsymbol{\psi}_{\setminus ii}$ we then compute the eigenvalues matrix $\boldsymbol{\Lambda}_1$ and eigenvectors matrix $\mathbf{U}$ of $\boldsymbol{\Psi}_{n\times n}$. This will provide the updated values to be used in Proposition 3.2. Hence, after n steps, the columns of $\boldsymbol{\Psi}_{n\times n}$ are estimated. Similarly the estimation of $\boldsymbol{\Theta}_{p\times p}$, for fixed $\boldsymbol{\Psi}_{n\times n}$, becomes directly analogous to the above simply by *transposing* the design matrix (samples become features and vice-versa) and is obtained in p steps. In our experiments the precision matrices $\boldsymbol{\Psi}_{n\times n}$ and $\boldsymbol{\Theta}_{p\times p}$ are initialised as identity matrices. The empirical mean matrix is removed from each dataset.

Algorithm 1 scBiGLasso

Input: Maximum iteration number N, tolerance ε, m many observations of $p \times n$ matrices $\mathbf{Y}^{(k)}$, $k = 1, \ldots, m$. β_1, β_2 and initial estimates of $\boldsymbol{\Psi}_{n\times n}$ and $\boldsymbol{\Theta}_{p\times p}$, $\boldsymbol{\Psi}^{(0)}_{n\times n}$ and $\boldsymbol{\Theta}^{(0)}_{p\times p}$.

For each $\mathbf{Y}^{(k)}$, $\mathbf{T^{(k)}} \leftarrow p^{-1}\mathbf{Y^{(k)}}\mathbf{Y^{(k)}}^\top$.

$\hat{\mathbf{T}} \leftarrow \frac{1}{m}\sum_{k=1}^{m}\hat{\mathbf{T}}^{(k)}$

repeat

 # Estimate $\boldsymbol{\Psi}_{n\times n}$:

for iteration $\tau = 1, \ldots, N$ **do**

for $i = 1, \ldots, n$ **do**

Partition $\boldsymbol{\Psi}^{(\tau-1)}_{n\times n}$ into $\psi^{(\tau-1)}_{ii}, \boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}$ and $\boldsymbol{\Psi}^{(\tau-1)}_{\setminus i\setminus i}$.

Calculate $\mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}$ as in Equation (11) with $\psi^{(\tau-1)}_{ii}$.

With *Lasso* regression, find a sparse solution of $p\mathbf{t}_{i\setminus i} + \mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}\boldsymbol{\psi}^{(\tau)}_{i\setminus i} = \mathbf{0}_{n-1}$.

Update the eigen-decomposition of the precision matrix $\boldsymbol{\Psi}^{(\tau)}_{n\times n} = \mathbf{U}\boldsymbol{\Lambda}_1\mathbf{U}^\top$

 # Estimate $\boldsymbol{\Theta}_{p\times p}$:

Proceed as if estimating $\boldsymbol{\Psi}_{n\times n}$ with input $\mathbf{Y}^\top, \beta_1, \beta_2$.

$$\Delta\boldsymbol{\Psi}^{(\tau)} = \|\boldsymbol{\Psi}^{(\tau)}_{n\times n} - \boldsymbol{\Psi}^{(\tau-1)}_{n\times n}\|^2_F$$

$$\Delta\boldsymbol{\Theta}^{(\tau)} = \|\boldsymbol{\Theta}^{(\tau)}_{p\times p} - \boldsymbol{\Theta}^{(\tau-1)}_{p\times p}\|^2_F$$

until Maximum iteration number reached, or $\max_{\tau^*=\tau-2,\tau-1,\tau}\left\{(\Delta\boldsymbol{\Psi}^{(\tau^*)} + \Delta\boldsymbol{\Theta}^{(\tau^*)})\right\} < \varepsilon$, for $\tau \geq 3$.

The approach is summarised in Algorithm 1 for Gaussian data. We point out that the convergence of Algorithm 1 could also be directly verified on the value of the objective function (2) at each step, but, due to the computation of $|\boldsymbol{\Psi}_{n\times n} \oplus \boldsymbol{\Theta}_{p\times p}|$, when $p, n >> 100$ this becomes unfeasible. Indeed, the space complexity can be reduced from $O\left(n^2p^2\right)$ to $O\left(n^2 + p^2\right)$ by means of Proposition 3.3.

Proposition 3.3 Following the assumptions and calculations above we have

$$|\boldsymbol{\Psi}_{n\times n} \oplus \boldsymbol{\Theta}_{p\times p}| = \prod_{i=1}^{n}\prod_{j=1}^{p}\left(\lambda_{1i} + \lambda_{2j}\right).$$

The proof of Proposition 3.3 is in the Supplementary Material. It follows that:

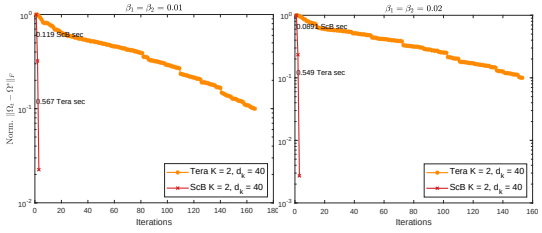
$$\log|\boldsymbol{\Psi}_{n\times n} \oplus \boldsymbol{\Theta}_{p\times p}| = \sum_{i=1}^{n}\sum_{j=1}^{p}\log|\lambda_{1i} + \lambda_{2j}| = K.$$

Hence we can write the objective function as

$$\min_{\boldsymbol{\Theta}_{p\times p}, \boldsymbol{\Psi}_{n\times n}}\Big\{n\mathrm{tr}\left(\boldsymbol{\Theta}_{p\times p}\mathbf{S}\right) + p\mathrm{tr}\left(\boldsymbol{\Psi}_{n\times n}\mathbf{T}\right) - K + \beta_1||\boldsymbol{\Psi}_{n\times n}||_1 + \beta_2||\boldsymbol{\Theta}_{p\times p}||_1\Big\}.$$

Note that this scalable version of the Bigraphical Lasso enables higher dimensional problems. This is mainly due to the fact that in our implementation there is no need to directly evaluate the matrix $\mathbf{W}$. Instead we just need the eigen-decomposition of the two precision matrices $\boldsymbol{\Psi}_{n\times n}$ and $\boldsymbol{\Theta}_{p\times p}$. In the original paper Kalaitzis et al. (2013) at each step i the blocks of $\mathbf{W}$ are explicitly updated and of course were involved in the next step of the estimation. In particular $\mathbf{W}_{\setminus ii}$ is computed via backward-substitution in Equation (9) and W_{11} via backward-substitution in Equation (8).

In summary, as we are not interested in the estimation of the overall $\hat{\mathbf{W}}$ nor $\boldsymbol{\Omega}$, we will never explicitly update them, but we will rather focus on the estimation of $\boldsymbol{\Psi}_{n\times n}$ and $\boldsymbol{\Theta}_{p\times p}$. This leads to a space complexity reduction from $O(n^2p^2)$ to $O(n^2 + p^2)$ by means of Proposition 3.2 and Proposition 3.3.



(a) (b)

Figure 1: ScB and Tera convergence rates and times with regularisation parameters $\beta_1 = \beta_2 \in \{0.01, 0.02\}$.

Our model provides a Scalable Bigraphical lasso algorithm (ScB) and as such benefits of the same statistical convergence properties. A subgaussian concentration

inequality (Greenewald et al., 2019, Lemma 19, Supplementary Material) gives rates of statistical convergence (Greenewald et al., 2019, Theorems 1-3) of the TeraLasso estimator as well as the Bigraphical Lasso estimator, when the sample size is low. In Figure 1 we show the numerical convergence rates and times of ScB with respect to the Frobenius norm for the precision matrix, compared to the Teralasso approach with K=2.

4 NONPARANORMAL BIGRAPHICAL MODEL

The method in Section 3 only deals with Gaussian data, while in real world many data come in the form of count data. In this section, we introduce a Gaussian copula based method to adapt Algorithm 1 for count data. We start with the definition of the matrix nonparanormal distribution with a Kronecker sum structure.

Definition 4.1 Consider a $p \times n$ non-Gaussian data matrix $\mathbf{Y}$. $\mathbf{Y}$ follows a matrix nonparanormal distribution with a Kronecker sum structure $\mathbf{MNPN}_{KS}\left(\mathbf{M}; \mathbf{\Psi}_{n\times n}^{-1}, \mathbf{\Theta}_{p\times p}^{-1}; f\right)$, with mean matrix $\mathbf{M}$, and where $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$ are the row-specific and the column-specific precision matrices, if and only if there exists a set of monotonic transformations $f = \{f_{ij}\}_{i=1,\dots,p}^{j=1,\dots,n}$ such that

$$\text{vec}[f(\mathbf{Y})] \sim \mathbf{mN}\left(\text{vec}(\mathbf{M}), (\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p})^{-1}\right).$$

In this paper, we only consider the model after centering, i.e $\text{vec}(\mathbf{M}) = \mathbf{0}_{np}$. The choices $f_{ij}(Y_{ij}) = Y_{ij}$ and $f_{ij}(Y_{ij}) = \log Y_{ij}$ give us multivariate Normal distribution and multivariate log-Normal distribution respectively. Since we only require f to be monotone, this model provides us with a wider family of distributions to work on, thus extends the Bigraphical model to non-Gaussian data. We note that the model in Definition 4.1 can be viewed as a latent model, with latent variable $\mathbf{Z} = f(\mathbf{Y})$ and $\text{vec}(\mathbf{Z}) \sim \mathbf{mN}\left(\mathbf{0}_{np}, (\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p})^{-1}\right)$.

Following the arguments in Kalaitzis et al. (2013) and Greenewald et al. (2019), the supports of $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$ encode the dependence structure of the row variables and the column variables, respectively. Further discussion and mathematical details of the decomposition of the latent model are in the Supplementary Material A.7.

In the next section, we introduce a method to infer the nonparanormal distribution without explicitly defining f.

4.1 Estimation of the precision matrices

We now consider estimation of the precision matrices $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$. Like the lasso methods applied in one-way network inference and in Gaussian Bigraphical models, we enforce sparsity on $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$ by regularization on the negative log-likelihood, which gives us the objective function:

$$\min_{\mathbf{\Psi}_{n\times n}, \mathbf{\Theta}_{p\times p}} \Big\{ p\text{tr}\left(\mathbf{\Psi}_{n\times n}\mathbf{T}\right) + n\text{tr}\left(\mathbf{\Theta}_{p\times p}\mathbf{S}\right) - K + \beta_1\|\mathbf{\Psi}_{n\times n}\|_1 + \beta_2\|\mathbf{\Theta}_{p\times p}\|_1 \Big\},$$

where $\mathbf{T} = \frac{1}{p}\left(\mathbf{Z}\mathbf{Z}^\top\right)$ is the empirical covariance matrix along the rows, and $\mathbf{S} = \frac{1}{n}\left(\mathbf{Z}^\top\mathbf{Z}\right)$ is the empirical covariance matrix along the columns. The only problem that remains now is to estimate the empirical covariance matrices $\mathbf{T}$ and $\mathbf{S}$. When estimating one-way network, Liu et al. (2012) proposed the nonparanormal skeptic, exploiting Kendall's tau or Spearman's rho, without explicitly calculating the marginal transforming function f. Similarly, we define Kendall's tau and Spearman's rho along rows and columns. More specifically, let $r_{ij}^{(c)}$ be the rank of Y_{ij} among $Y_{1j}, \dots, Y_{pj}$ and $\bar{r}_j^{(c)} = \frac{1}{p}\sum_{i=1}^p r_{ij} = \frac{p+1}{2}$. Define $\Delta_i(j, j') = Y_{ij} - Y_{ij'}$. We consider the following statistics:

(Column-wise Kendall's tau)

$$\hat{\tau}_{i_1 i_2}^{(c)} = \frac{2}{n(n-1)} \sum_{j<j'} \text{sign}\left(\Delta_{i_1}(j, j')\Delta_{i_2}(j, j')\right),$$

(Column-wise Spearman's rho)

$$\hat{\rho}_{i_1 i_2}^{(c)} = \frac{\sum_{j=1}^n \left(r_{i_1 j}^{(c)} - \bar{r}_j^{(c)}\right)\left(r_{i_2 j}^{(c)} - \bar{r}_j^{(c)}\right)}{\sqrt{\sum_{j=1}^n \left(r_{i_1 j}^{(c)} - \bar{r}_j^{(c)}\right)^2 \left(r_{i_2 j}^{(c)} - \bar{r}_j^{(c)}\right)^2}}.$$

Similarly, let $r_{ij}^{(r)}$ be the rank of Y_{ij} among $Y_{i1}, \dots, Y_{in}$ and $\bar{r}_i^{(r)} = \frac{1}{n}\sum_{j=1}^n r_{ij} = \frac{n+1}{2}$. Define $\Delta_j(i, i') = Y_{ij} - Y_{i'j}$. We consider the following statistics:

(Row-wise Kendall's tau)

$$\hat{\tau}_{j_1 j_2}^{(r)} = \frac{2}{p(p-1)} \sum_{i<i'} \text{sign}\left(\Delta_{j_1}(i, i')\Delta_{j_2}(i, i')\right),$$

(Row-wise Spearman's rho)

$$\hat{\rho}_{j_1 j_2}^{(r)} = \frac{\sum_{i=1}^p \left(r_{i j_1}^{(r)} - \bar{r}_i^{(r)}\right)\left(r_{i j_2}^{(r)} - \bar{r}_i^{(r)}\right)}{\sqrt{\sum_{i=1}^p \left(r_{i j_1}^{(r)} - \bar{r}_i^{(r)}\right)^2 \left(r_{i j_2}^{(r)} - \bar{r}_i^{(r)}\right)^2}}.$$

And the following estimated covariance matrices using Kendall's tau and Spearman's rho:

$$\hat{\mathbf{T}}_{j_1 j_2} = \begin{cases} \sin\left(\frac{\pi}{2}\hat{\tau}_{j_1 j_2}^{(r)}\right), & j_1 \neq j_2, \\ 1, & j_1 = j_2. \end{cases} \tag{12}$$

$$\hat{\mathbf{T}}_{j_1 j_2} = \begin{cases} 2\sin\left(\frac{\pi}{6}\hat{\rho}^{(r)}_{j_1 j_2}\right), & j_1 \neq j_2, \\ 1, & j_1 = j_2. \end{cases} \quad (13)$$

$$\hat{\mathbf{S}}_{i_1 i_2} = \begin{cases} \sin\left(\frac{\pi}{2}\hat{\tau}^{(c)}_{i_1 i_2}\right), & i_1 \neq i_2, \\ 1, & i_1 = i_2. \end{cases}$$

$$\hat{\mathbf{S}}_{i_1 i_2} = \begin{cases} 2\sin\left(\frac{\pi}{6}\hat{\rho}^{(c)}_{i_1 i_2}\right), & i_1 \neq i_2, \\ 1, & i_1 = i_2. \end{cases}$$

In Algorithm 2 we summarise the Nonparanormal Scalable Bigraphical Lasso approach for count data.

Algorithm 2 Nonparanormal scBiGLasso

Input: Maximum iteration number N, tolerance ε, m many observations of $p \times n$ count matrices $\mathbf{Y}^{(k)}$, $k = 1, \ldots, m$. β_1, β_2 and initial estimates of $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$, $\mathbf{\Psi}^{(0)}_{n\times n}$ and $\mathbf{\Theta}^{(0)}_{p\times p}$.
For each $\mathbf{Y}^{(k)}$, calculate $\hat{\mathbf{T}}^{(k)}$ according to Equation (12) or (13).
$\hat{\mathbf{T}} \leftarrow \frac{1}{m}\sum_{k=1}^{m} \hat{\mathbf{T}}^{(k)}$
repeat
  # *Estimate* $\mathbf{\Psi}_{n\times n}$:
  for iteration $\tau = 1, \ldots, N$ **do**
    for $i = 1, \ldots, n$ **do**
      Partition $\mathbf{\Psi}^{(\tau-1)}_{n\times n}$ into $\psi^{(\tau-1)}_{ii}, \boldsymbol{\psi}^{(\tau-1)}_{i\setminus i}$ and $\mathbf{\Psi}^{(\tau-1)}_{\setminus i\setminus i}$.
      Calculate $\mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}$ as in Equation (11) with $\psi^{(\tau-1)}_{ii}$.
      With *Lasso* regression, find a sparse solution of $p\mathbf{t}_{i\setminus i} + \mathbf{A}^{(\tau-1)}_{\setminus i\setminus i}\boldsymbol{\psi}^{(\tau)}_{i\setminus i} = \mathbf{0}_{n-1}$.
      Update the eigen-decomposition of the precision matrix $\mathbf{\Psi}^{(\tau)}_{n\times n} = \mathbf{U}\mathbf{\Lambda}_1\mathbf{U}^\top$
  # *Estimate* $\mathbf{\Theta}_{p\times p}$:
  Proceed as if estimating $\mathbf{\Psi}_{n\times n}$ with input $\mathbf{Y}^\top, \beta_1, \beta_2$.

$$\Delta\mathbf{\Psi}^{(\tau)} = \|\mathbf{\Psi}^{(\tau)}_{n\times n} - \mathbf{\Psi}^{(\tau-1)}_{n\times n}\|^2_F$$

$$\Delta\mathbf{\Theta}^{(\tau)} = \|\mathbf{\Theta}^{(\tau)}_{p\times p} - \mathbf{\Theta}^{(\tau-1)}_{p\times p}\|^2_F$$

until Maximum iteration number reached, or $\max_{\tau^*=\tau-2,\tau-1,\tau}\left\{(\Delta\mathbf{\Psi}^{(\tau^*)} + \Delta\mathbf{\Theta}^{(\tau^*)})\right\} < \varepsilon$, for $\tau \geq 3$.

5 NUMERICAL RESULTS

In this Section, we implement our Scalable Bigraphical Lasso algorithm where covariance matrices are estimated using Kendall's tau. After precision matrices $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$ are inferred, they are transformed into binary matrices to reveal the network structures, where any non-zero value in the precision matrices become 1 and any zero value stays zero. We illustrate an application of our overall approach on both synthetic and real datasets as described in the following subsections. Code to reproduce our results are available on GitHub.

5.1 Synthetic Gaussian Data

To demonstrate the efficiency of our Scalable Bigraphical Lasso algorithm (Algorithm 1), we generate sparse positive definite matrices $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$. Then simulate m many $p \times n$ Gaussian data $Y^{(k)}_G$, $k = 1, \ldots, m$ from mN $\left(\mathbf{0}, (\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p})^{-1}\right)$. We plug $Y^{(k)}_G$, $k = 1, \ldots, m$ into our implemented Algorithm 1, Bigraphical Lasso from Kalaitzis et al. (2013) and TeraLasso from Greenewald et al. (2019). Figure 1 shows a comparison between the convergence times of Algorithm 1 and Bigraphical Lasso for increasing problem dimensions $n = p$. We can observe that, as expected, Algorithm 1 converges in significantly faster times, allowing one to tackle higher dimensional problems in practice. Table 1 shows the network recovery when $n = p = 100$. We can see that our method provides high Accuracy while improving greatly on speed; see Section 5.2 for the definition of Accuracy.

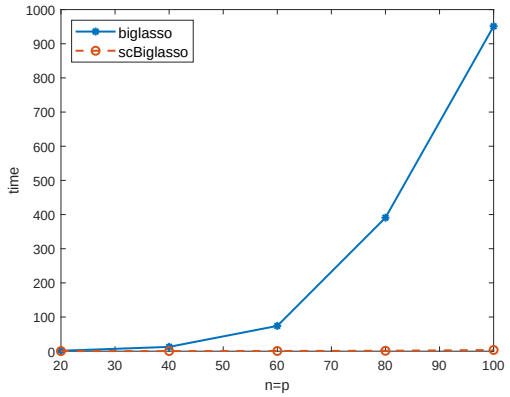


Figure 2: Computational convergence time (*seconds*) comparison between Bigraphical Lasso (Kalaitzis et al. (2013)) and Algorithm 1, for increasing values of the dataset dimensions $n = p$.

Table 1: Comparison between computational convergence times, Accuracy of $\mathbf{\Psi}$ and of $\mathbf{\Theta}$ for Bigraphical Lasso (Kalaitzis et al. (2013)), TeraLasso (Greenewald et al. (2019)) and Algorithm 1, for a synthetic Gaussian dataset with dimensions $n = p = 100$.

Method	Accuracy_Ψ	Accuracy_Θ	Time(s)
Biglasso	0.9032	0.9028	951.15
ScBiglasso	0.9032	0.9028	3.50
TeraLasso	0.5416	0.4323	0.3696

5.2 Synthetic count data

We generate and process Gaussian Copula-based count data through the following steps:

1. Generate sparse positive definite matrix $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$. Calculate the Kronecker sum of $\mathbf{\Psi}_{n\times n}$ and $\mathbf{\Theta}_{p\times p}$.Generate m multivariate-normal vectors of length $p \times n$ from $\text{mN}\left(\mathbf{0}, \mathbf{\Omega}^{-1}\right)$, where $\mathbf{\Omega} = \mathbf{\Psi}_{n\times n} \otimes I_p + I_n \otimes \mathbf{\Theta}_{p\times p}$.

2. Centre each of the m multivariate-normal vectors around their mean, and reshape the vectors into $p \times n$ matrices $X^{(1)}, \ldots, X^{(m)}$.

3. For each $X^{(k)}$, $k = 1, \ldots, m$, calculate the matrix $P^{(k)}$ such that $P_{ij}^{(k)} = \Phi\left(X_{ij}^{(k)}\right)$, where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution.

4. For each $k = 1, \ldots, m$, produce the negative binomial variable $Y_{ij}^{(k)} = QNB\left(P_{ij}^{(k)}, r, p\right)$, where $QNB(\cdot, r, p)$ is the quantile function of Negative-Binomial (r, p), with r the number of success to be observed and p the success rate.

Below we describe some of the criteria we use to assess the recovery of the synthetic network. Denote TP as the number of *True Positives* in the network recovery, TN as the number of *True Negatives* in the network recovery, FP as the number of *False Positives* in the network recovery, and FN the number of *False Negatives* in the network recovery, then we can define

$$Precision = \frac{TP}{TP+FP}, \quad Recall = \frac{TP}{TP+FN},$$
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN},$$
$$TPR = \frac{TP}{TP+FN}, \quad FPR = \frac{FP}{TN+FP}.$$

Figure 3 shows some results from synthetic data. Figure 3 (a) is the Precision-Recall of the recovery of $\mathbf{\Psi}_{n\times n}$ with changing β_1 (different points on the graph) and β_2 (different colours on the graph). Two arbitrary values of β_2 have been chosen to illustrate how the results do not depend on β_2. This is expected as β_1 is the regularization parameter for $\mathbf{\Psi}_{n\times n}$, while β_2 corresponds to $\mathbf{\Theta}_{p\times p}$. A similar result is shown in Figure 3 (b), where the Precision-Recall of the recovery of $\mathbf{\Theta}_{p\times p}$ heavily depends on the choice of β_2, regardless of the β_1 value. Figure 3 (c)(d) show that high values of TPR and Accuracy, with low values of FPR, can be achieved for appropriate choices of β_1 and β_2 in the range $[0.005, 0.016]$.

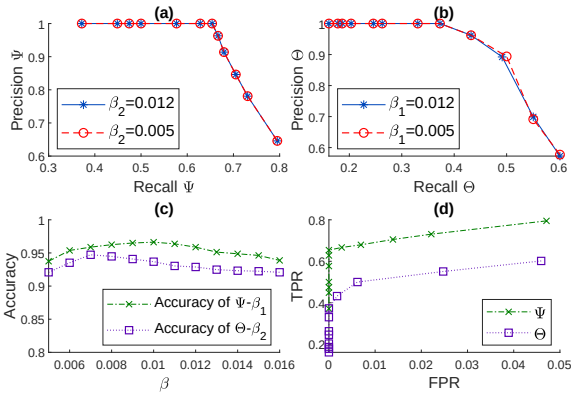


Figure 3: Synthetic network recovery results. **(a)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}_{n\times n}$; **(b)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Theta}_{p\times p}$; **(c)** Accuracy vs corresponding regularization parameter β_1 (β_2) of the network recovery relating to the support of $\mathbf{\Psi}_{n\times n}$ ($\mathbf{\Theta}_{p\times p}$) and **(d)** TPR-FPR of the network recovery relating to the support of $\mathbf{\Psi}_{n\times n}$ ($\mathbf{\Theta}_{p\times p}$), where the corresponding regularization parameter β_1 (β_2) $\in \{0.005 : 0.001 : 0.0016\}$.

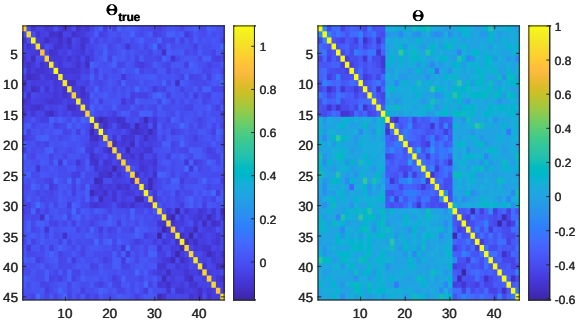


Figure 4: Synthetic network recovery. We generated synthetic data as described in Section 5.2 using a block-diagonal precision matrix for $\mathbf{\Theta}_0$ plus Gaussian noise (Left plot). On the right we plot the estimated $\mathbf{\Theta}$ via our method. In this example, we used $\beta_2 = 0.0002$.

Figure 4 shows network recovery for another synthetic count dataset, where the original precision matrix $\mathbf{\Theta}_0$ was generated with block diagonals and Gaussian noise throughout the matrix. We observe that our method leads to good recovery of the corresponding blocks. Further discussion on the choice of optimal regularization parameters $\beta = (\beta_1, \beta_2)$ is in the Supplementary Material.

5.3 mESC scRNA-seq data

We use a single cell gene expression dataset from mouse embryonic stem cells (mESC) available in Buettner et al. (2015). The data consist of measurements of gene counts in 182 single cells at different stages of the cell cycle. We will refer to the three phases as G1, S and G2M. About 700 genes are annotated as cell cycle related. Of these, we considered 167 genes more active

during mitosis, the cell division phase and last part of the cell cycle (G2M). In our dataset there are 65 cells in the G2M phase.

In Figures 5 and 6, we show how our model allows the identification of the sub-population of cells that correspond to the G2M stage. In Figure 5 we show the estimated precision matrices for the cells (left) and the genes (right). We use a binary transformation where only the negative values are considered an edge in the network. In Figure 6 we plot the corresponding networks, over imposing the clusters found with the label propagation approach developed by Raghavan et al. (2007). We note that $\sim 92\%$ of the G2M cells are clustered in two densely connected modules ($\mathbf{\Psi}$ network plot in Figure 6), while no connection is measured between cells in different phases of the cell cycle. As expected, on the other hand, the mitosis genes are all densely connected in a single cluster ($\mathbf{\Theta}$ network plot in Figure 6).

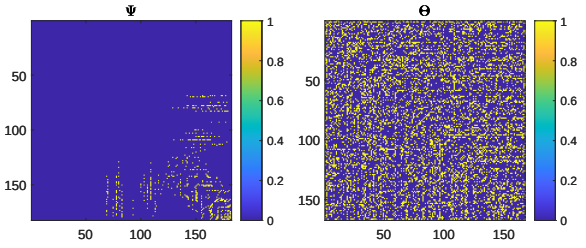


Figure 5: Networks recovered by our proposed Scalable Bigraphical Lasso algorithm combined with the non-paranormal transformation as described in Section 4.2, $(\beta_1, \beta_2) = (0.014, 0.001)$.

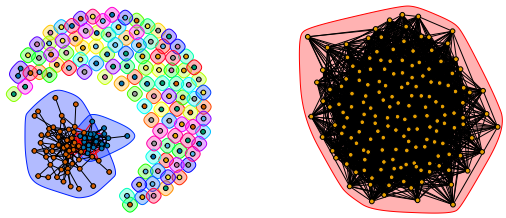


Figure 6: $\mathbf{\Psi}$ (left) and $\mathbf{\Theta}$ (right) induced networks.

6 CONCLUSIONS

In this work, we present a Scalable Bigraphical Lasso algorithm. In particular, we exploit eigenvalue decomposition of the Cartesian product graph to present a more efficient version of the algorithm presented in Kalaitzis et al. (2013). Our approach reduces memory requirements from $O(n^2p^2)$ to $O(n^2 + p^2)$, and reduces the computational time by up to a factor of 200 in our experiments (case $p = n = 100$ in Figure 2 and Table 1). Note that comparisons for $n = p > 100$ were restricted because of the memory limitation in Kalaitzis et al. (2013). Additionally, we propose a Gaussian-copula based model and a semiparametric approach that enables the application of the proposed Bigraphical model to non-Gaussian data. This is particularly relevant for count data applications, such as single cell data. Future work will include optimisation of the choice of the regularization parameters, and potential extension to k-way network inference for non-Gaussian data, with $k > 2$.

Data availability

The code and data is available at `https://github.com/luisacutillo78/Scalable_Bigraphical_Lasso.git`.

Acknowledgements

Sijia Li was supported by an EPSRC Doctoral Training Partnership (reference EP/R513258/1) through the University of Leeds. The authors would like to thank Michael Croucher for the support in optimizing the MATLAB code. Sijia Li would like to thank Nicole Mücke for the mentorship on the writing. Luisa Cutillo and Neil Lawrence would like to acknowledge the Marie Curie fellowship CONTESSA (ID: 660388), during which the main ideas of this research were conceived. The authors would like to thank the reviewers and editor for their constructive criticism of the manuscript.

References

Alfredo Kalaitzis, John Lafferty, Neil D Lawrence, and Shuheng Zhou. The bigraphical lasso. In *International Conference on Machine Learning*, pages 1229–1237. PMLR, 2013.

Theodoros Tsiligkaridis and Alfred O Hero. Covariance estimation in high dimensions via kronecker product expansions. *IEEE Transactions on Signal Processing*, 61(21):5347–5360, 2013.

Shuheng Zhou. Gemini: Graph estimation with matrix variate normal instances. *The Annals of Statistics*, 42(2):532–562, 2014.

Kristjan Greenewald, Shuheng Zhou, and Alfred Hero III. Tensor graphical lasso (teralasso). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 81(5):901–931, 2019.

Bochao Jia, Suwa Xu, Guanghua Xiao, Vishal Lamba, and Faming Liang. Learning gene regulatory networks from next generation sequencing data. *Biometrics*, 73(4):1221–1230, 2017.

Andrew McDavid, Raphael Gottardo, Noah Simon, and Mathias Drton. Graphical models for zero-inflated single cell gene expression. *The annals of applied statistics*, 13(2):848, 2019.

Han Liu, John Lafferty, and Larry Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *Journal of Machine Learning Research*, 10(10), 2009.

Han Liu, Fang Han, Ming Yuan, John Lafferty, and Larry Wasserman. High-dimensional semiparametric gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326, 2012.

Arkaprava Roy and David B Dunson. Nonparametric graphical model for counts. *Journal of Machine Learning Research*, 21(229):1–21, 2020.

Julien Chiquet, Stephane Robin, and Mahendra Mariadassou. Variational inference for sparse network reconstruction from count data. In *International Conference on Machine Learning*, pages 1162–1171. PMLR, 2019.

Thomas E Bartlett, Ioannis Kosmidis, and Ricardo Silva. Two-way sparsity for time-varying networks with applications in genomics. *The Annals of Applied Statistics*, 15(2):856–879, 2021.

Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon Press, 1996.

Gert Sabidussi. Graph multiplication. *Mathematische Zeitschrift*, 72(1):446–457, 1959.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

Yang Ning and Han Liu. High-dimensional semiparametric bigraphical models. *Biometrika*, 100(3):655–670, 2013.

Florian Buettner, Kedar N Natarajan, F Paolo Casale, Valentina Proserpio, Antonio Scialdone, Fabian J Theis, Sarah A Teichmann, John C Marioni, and Oliver Stegle. Computational analysis of cell-to-cell heterogeneity in single-cell rna-sequencing data reveals hidden subpopulations of cells. *Nature biotechnology*, 33(2):155–160, 2015.

Usha Nandini Raghavan, Réka Albert, and Soundar Kumara. Near linear time algorithm to detect community structures in large-scale networks. *Physical review E*, 76(3):036106, 2007.

John Lafferty, Han Liu, and Larry Wasserman. Sparse nonparametric graphical models. *Statistical Science*, 27(4):519–537, 2012.

Ulrich Knauer and Kolja Knauer. *Algebraic graph theory*. de Gruyter, 2019.

D. M. Cvetković, Michael Doob, and Horst Sachs. *Spectra of Graphs: Theory and Applications, 3rd rev. enl. ed.* New York: Wiley, 1998.

Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

Supplementary Material: Two-way Sparse Network Inference for Count Data

A Mathematical analysis

In this Section, we provide detailed proofs for some of the properties and results in the main paper.

A.1 Proof of Equations (4) and (6)

Equation (4) in the main paper follows from the following:

$$\boldsymbol{\Omega} \;=\; \boldsymbol{\Psi}_{n\times n} \oplus \boldsymbol{\Theta}_{p\times p} \;=\; \mathbf{U}\Lambda_1\mathbf{U}^\top \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \mathbf{V}\Lambda_2\mathbf{V}^\top \;=\; (\mathbf{U}\otimes\mathbf{V})\left[\Lambda_1 \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \Lambda_2\right]\left(\mathbf{U}^\top \otimes \mathbf{V}^\top\right).$$

Equation (6) within the main paper follows from Equation (4). In particular, we have

$$\begin{aligned}
(\mathbf{U}\otimes\mathbf{I}_p)\,\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right)\boldsymbol{\Omega} &= (\mathbf{U}\otimes\mathbf{I}_p)\,\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right)\left(\boldsymbol{\Psi}_{n\times n}\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{V}\Lambda_2\mathbf{V}^\top\right)\\
&= (\mathbf{U}\otimes\mathbf{I}_p)\,\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{I}_p\right)\left(\mathbf{I}_n\otimes\mathbf{V}^\top\right)\left(\boldsymbol{\Psi}_{n\times n}\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\mathbf{V}\Lambda_2\mathbf{V}^\top\right)\\
&= (\mathbf{U}\otimes\mathbf{I}_p)\,\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{I}_p\right)\left(\boldsymbol{\Psi}_{n\times n}\otimes\mathbf{V}^\top + \mathbf{I}_n\otimes\Lambda_2\mathbf{V}^\top\right)\\
&= \mathbf{I}_n\otimes\mathbf{V}^\top.
\end{aligned}$$

A.2 Proof of Equation (7)

Note that $\mathbf{W}\boldsymbol{\Omega} = \mathbf{I}_{np}$, therefore we can write Equation (6) as:

$$\left(\mathbf{I}_n\otimes\mathbf{V}^\top\right)\mathbf{W}\boldsymbol{\Omega} = (\mathbf{U}\otimes\mathbf{I}_p)\,\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right)\boldsymbol{\Omega}.$$

Multiply both sides of the equation above by $\mathbf{I}_n\otimes\mathbf{V}$:

$$\left(\mathbf{I}_n\otimes\mathbf{V}^\top\right)\mathbf{W}\boldsymbol{\Omega}\left(\mathbf{I}_n\otimes\mathbf{V}\right) = (\mathbf{U}\otimes\mathbf{I}_p)\,\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right)\boldsymbol{\Omega}\left(\mathbf{I}_n\otimes\mathbf{V}\right).$$

From the right-hand side, we get $(\mathbf{U}\otimes\mathbf{I}_p)\,\mathbf{D}\left(\mathbf{U}^\top\otimes\mathbf{I}_p\right)\left(\boldsymbol{\Psi}_{n\times n}\otimes\mathbf{I}_p + \mathbf{I}_n\otimes\Lambda_2\right)$. On the left-hand side, remember that $\mathbf{W}\boldsymbol{\Omega} = \mathbf{I}_{np}$, so

$$\left(\mathbf{I}_n\otimes\mathbf{V}^\top\right)\mathbf{W}\boldsymbol{\Omega}\left(\mathbf{I}_n\otimes\mathbf{V}\right) = \mathbf{I}_n\otimes\mathbf{I}_p.$$

Indeed we get Equation (7) in the main paper.

A.3 Proof of Equation (9)

In order to prove Equation (9), we first note that, from the bottom-left block of

$$\hat{\mathbf{W}}\hat{\boldsymbol{\Omega}} = \begin{bmatrix}\hat{\mathbf{W}}_{11} & \hat{\mathbf{W}}_{1\backslash 1}\\ \hat{\mathbf{W}}_{\backslash 11} & \hat{\mathbf{W}}_{\backslash 1\backslash 1}\end{bmatrix}\begin{bmatrix}\psi_{11}\mathbf{I}_p + \Lambda_2 & \dots & \psi_{1n}\mathbf{I}_p\\ \vdots & \ddots & \vdots\\ \psi_{n1}\mathbf{I}_p & \dots & \psi_{nn}\mathbf{I}_p + \Lambda_2\end{bmatrix} = \mathbf{I}_n\otimes\mathbf{I}_p$$

we get

$$\hat{\mathbf{W}}_{\backslash 11}\hat{\boldsymbol{\Omega}}_{11} + \hat{\mathbf{W}}_{\backslash 1\backslash 1}\hat{\boldsymbol{\Omega}}_{\backslash 11} = \hat{\mathbf{W}}_{\backslash 11}\left(\psi_{11}\mathbf{I}_p + \Lambda_2\right) + \hat{\mathbf{W}}_{\backslash 1\backslash 1}\left(\boldsymbol{\psi}_{\backslash 11}\otimes\mathbf{I}_p\right) = \mathbf{0}_{n-1}\otimes\mathbf{I}_p.$$

Thus, multiplying both sides of the last equation by $\left(\psi_{11}\mathbf{I}_p + \Lambda_2\right)^{-1}$, one has

$$\hat{\mathbf{W}}_{\backslash 11} + \hat{\mathbf{W}}_{\backslash 1\backslash 1}\begin{bmatrix}\left(\psi_{11}\mathbf{I}_p + \Lambda_2\right)^{-1}\psi_{21}\\ \vdots\\ \left(\psi_{11}\mathbf{I}_p + \Lambda_2\right)^{-1}\psi_{n1}\end{bmatrix} = \mathbf{0}_{n-1}\otimes\mathbf{I}_p.$$

A.4 Proof of Proposition 3.1

Proposition 3.1 follows from the fact that

$$\left[\mathbf{I}_n \otimes \mathbf{V}^\top\right] \mathbf{W} \left[\mathbf{I}_n \otimes \mathbf{V}\right] = \left[\mathbf{U} \otimes \mathbf{I}_p\right] \mathbf{D} \left[\mathbf{U}^\top \otimes \mathbf{I}_p\right] = \hat{\mathbf{W}}.$$

Then, the $p \times p$ blocks of $\mathbf{W}$ and $\hat{\mathbf{W}}$ hold a similarity relation:

$$\hat{\mathbf{W}}_{ij} = \mathbf{V}^\top \mathbf{W}_{ij} \mathbf{V}$$

and hence $\mathrm{tr}_p\left(\mathbf{W}\right) = \mathrm{tr}_p\left(\hat{\mathbf{W}}\right)$.

A.5 Proof of Proposition 3.2

To prove Proposition 3.2, we note that

$$\hat{\mathbf{W}}_{\setminus 1 \setminus 1} = \left[\mathbf{U}_{\setminus 1} \otimes \mathbf{I}_p\right] \mathbf{D} \left[\mathbf{U}_{\setminus 1}^\top \otimes \mathbf{I}_p\right] = \begin{bmatrix} u_{21}\mathbf{I}_p & \dots & u_{2n}\mathbf{I}_p \\ \vdots & \ddots & \vdots \\ u_{n1}\mathbf{I}_p & \dots & u_{nn}\mathbf{I}_p \end{bmatrix} \mathbf{D} \begin{bmatrix} u_{21}\mathbf{I}_p & \dots & u_{n1}\mathbf{I}_p \\ \vdots & \ddots & \vdots \\ u_{2n}\mathbf{I}_p & \dots & u_{nn}\mathbf{I}_p \end{bmatrix},$$

where $\mathbf{U}_{\setminus 1} \in \mathbb{R}^{(n-1)\times n}$ is the matrix formed by the last $n-1$ rows of $\mathbf{U}$. Then, we can decompose $\hat{\mathbf{W}}_{\setminus 1 \setminus 1}$ in $(n-1) \times (n-1)$ blocks $[\hat{\mathbf{W}}_{\setminus 1 \setminus 1}]_{\ell,k} \in \mathbb{R}^{p \times p}$, with

$$[\hat{\mathbf{W}}_{\setminus 1 \setminus 1}]_{\ell,k} = \begin{bmatrix} \sum_{i=1}^n \frac{u_{(\ell)i} u_{ki}}{\lambda_{1i} + \lambda_{21}} & \dots & 0 \\ 0 & \dots & \sum_{i=1}^n \frac{u_{(\ell)i} u_{ki}}{\lambda_{1i} + \lambda_{2p}} \end{bmatrix}, \quad \ell, k \in \{2, \dots, n\}.$$

Note that if we partition the $\hat{\mathbf{W}}$ into four blocks starting from any other $\hat{\mathbf{W}}_{hh}$ with $h \in \{1, \dots, n\}$ the above sums would be over $i \in \{1, \dots, h-1, h+1, \dots, n\}$. This formulation allows us to write each trace term of Equation (10) in the main paper as

$$\mathrm{tr}_p\left\{\hat{\mathbf{W}}_{\setminus 1 k} \left(\psi_{11}\mathbf{I}_p + \mathbf{\Lambda}_2\right)^{-1}\right\} = \begin{bmatrix} \mathrm{tr}\left\{\hat{\mathbf{W}}_{\setminus 1 \setminus 1}\right\}_{1,k} \left(\psi_{11}\mathbf{I}_p + \mathbf{\Lambda}_2\right)^{-1} \\ \vdots \\ \mathrm{tr}\left\{\hat{\mathbf{W}}_{\setminus 1 \setminus 1}\right\}_{(n-1),k} \left(\psi_{11}\mathbf{I}_p + \mathbf{\Lambda}_2\right)^{-1} \end{bmatrix}, \quad k \in \{1, \dots, n-1\},$$

More explicitly,

$$\begin{aligned} \mathrm{tr}_p\left\{\hat{\mathbf{W}}_{\setminus 1 k} \left(\psi_{11}\mathbf{I}_p + \mathbf{\Lambda}_2\right)^{-1}\right\} &= \begin{bmatrix} \sum_{j=1}^p \sum_{i=1}^n \frac{1}{\psi_{11} + \lambda_{2j}} \frac{u_{2i} u_{ki}}{\lambda_{1i} + \lambda_{21}} \\ \vdots \\ \sum_{j=1}^p \sum_{i=1}^n \frac{1}{\psi_{11} + \lambda_{2j}} \frac{u_{ni} u_{ki}}{\lambda_{1i} + \lambda_{2p}} \end{bmatrix} = \begin{bmatrix} \sum_{j=1}^p \frac{1}{\psi_{11} + \lambda_{2j}} \sum_{i=1}^n \frac{u_{2i} u_{ki}}{\lambda_{1i} + \lambda_{21}} \\ \vdots \\ \sum_{j=1}^p \frac{1}{\psi_{11} + \lambda_{2j}} \sum_{i=1}^n \frac{u_{ni} u_{ki}}{\lambda_{1i} + \lambda_{2p}} \end{bmatrix} \\ &= \sum_{j=1}^p \frac{1}{\psi_{11} + \lambda_{2j}} \begin{bmatrix} \sum_{i=1}^n \frac{u_{2i} u_{ki}}{\lambda_{1i} + \lambda_{21}} \\ \vdots \\ \sum_{i=1}^n \frac{u_{ni} u_{ki}}{\lambda_{1i} + \lambda_{2p}} \end{bmatrix}. \end{aligned}$$

A.6 Proof of Proposition 3.3

Proposition 3.3 follows from the fact that

$$\mathbf{W} = \mathbf{\Omega}^{-1} = \left(\mathbf{U} \otimes \mathbf{V}\right) \left[\mathbf{\Lambda}_1 \otimes \mathbf{I}_p + \mathbf{I}_n \otimes \mathbf{\Lambda}_2\right]^{-1} \left(\mathbf{U}^\top \otimes \mathbf{V}^\top\right),$$

and

$$\mathbf{D} = \begin{bmatrix} \frac{1}{\lambda_{11}+\lambda_{21}} & \dots & 0 & \dots & 0 & \dots & 0 \\ \vdots & \ddots & \vdots & \dots & \vdots & \dots & \vdots \\ 0 & \dots & \frac{1}{\lambda_{11}+\lambda_{2p}} & \dots & 0 & \dots & 0 \\ \vdots & \dots & \vdots & \ddots & \vdots & \dots & \vdots \\ 0 & \dots & 0 & \dots & \frac{1}{\lambda_{1n}+\lambda_{21}} & \dots & 0 \\ \vdots & \dots & \vdots & \dots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \dots & 0 & \dots & \frac{1}{\lambda_{1n}+\lambda_{2p}} \end{bmatrix},$$

where $\lambda_{11} \dots \lambda_{1n}$ are the diagonal values of $\mathbf{\Lambda}_1 \in \mathbb{R}^{n\times n}$ and $\lambda_{21} \dots \lambda_{2p}$ are the diagonal values of $\mathbf{\Lambda}_2 \in \mathbb{R}^{p\times p}$. Then, we can write

$$\begin{aligned} |\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}| &= |\left(\mathbf{U}\otimes\mathbf{V}\right)\mathbf{D}^{-1}\left(\mathbf{U}^\top\otimes\mathbf{V}^\top\right)| = |\mathbf{U}\otimes\mathbf{V}|^2|\mathbf{D}^{-1}| = |\mathbf{U}|^{2p}|\mathbf{V}|^{2n}\prod_{i=1}^{n}\prod_{j=1}^{p}\left(\lambda_{1i}+\lambda_{2j}\right) \\ &= \prod_{i=1}^{n}\prod_{j=1}^{p}\left(\lambda_{1i}+\lambda_{2j}\right). \end{aligned}$$

A.7 Some mathematical details for Section 4.1

Consider the $p \times n$ random matrix $\mathbf{Y} = (Y_{ij})$, $i = 1, \dots, p,\ j = 1, \dots, n$. Consider for each row vectors of $\mathbf{Y}$, $Y_i = (Y_{i1}, \dots, Y_{in})^\top,\ i = 1, \dots, p$, the marginal distributions $F_1^{(r)}, \dots, F_j^{(r)}, \dots, F_n^{(r)}$, where the superscript (r) denotes marginal distributions in row vector. Then by Sklar's theorem, for a $n-$dimensional distribution function $\Phi_{\left(\mathbf{0_n},\mathbf{\Psi^{-1}_{n\times n}}\right)}$, there exists copula $C^{(r)}$ such that

$$\Phi_{\left\{\mathbf{0_n},\mathbf{\Psi^{-1}_{n\times n}}\right\}}\left(\Phi^{-1}\left(F_1^{(r)}\left(Y_{i1}\right)\right),\dots,\Phi^{-1}\left(F_n^{(r)}\left(Y_{in}\right)\right)\right) = C^{(r)}\left(F_1^{(r)}\left(Y_{i1}\right),\dots,F_n^{(r)}\left(Y_{in}\right)\right).$$

That is, there exist functions $f^{(r)} = \left\{f_j^{(r)}\right\}_{j=1}^{n}$ such that for each row vectors of $\mathbf{Y}$, $Y_i = (Y_{i1}, \dots, Y_{in})^\top,\ i = 1, \dots, p,\ Z_i^{(r)} \equiv f^{(r)}\left(Y_i\right) \sim \mathrm{mN}\left(\mathbf{0}_n, \mathbf{\Psi}_{n\times n}\right)$, where $f^{(r)}\left(Y_i\right) = \left(f_1^{(r)}\left(Y_{i1}\right),\dots,f_n^{(r)}\left(Y_{in}\right)\right)$. Then we say $Y_i = (Y_{i1}, \dots, Y_{in})^\top$ has a nonparanormal distribution and write

$$Y_i \sim \mathrm{NPN}\left(\mathbf{0}_n, \mathbf{\Psi}^{-1}_{n\times n}, f^{(r)}\right).$$

According to Lemma 3.1 in Lafferty et al. (2012), the dependence between $Y_{i1},\ \dots,\ Y_{in}, i = 1, \dots, p$, can be illustrated by a Gauss-Markov Graph $G_r = \{V_r, E_r\}$ corresponding to precision matrix $\mathbf{\Psi}_{n\times n}$. This is equivalent to have latent variable $\mathbf{Z^{(r)}} = \mathbf{f^{(c)}}(\mathbf{Y_i}) \sim \mathrm{mN}\left(\mathbf{0_n}, \mathbf{\Psi^{-1}_{n\times n}}\right), \mathbf{i = 1}, \dots, \mathbf{p}$.

Similarly, for each column vector of $\mathbf{Y}$, $Y_j = (Y_{1j}, \dots, Y_{pj})^\top,\ j = 1, \dots, n$, we consider marginal distributions $F_1^{(c)}, \dots, F_i^{(c)}, \dots, F_n^{(c)}$, where the superscript (c) denotes marginal distributions in column vector. Then by Sklar's theorem, for a $p-$dimensional distribution function $\Phi_{\left(\mathbf{0_p},\mathbf{\Psi^{-1}_{p\times p}}\right)}$, there exists copula $C^{(c)}$ such that

$$\Phi_{\left(\mathbf{0_p},\mathbf{\Theta^{-1}_{p\times p}}\right)}\left(\Phi^{-1}\left(F_1^{(c)}\left(Y_{1j}\right)\right),\dots,\Phi^{-1}\left(F_p^{(c)}\left(Y_{pj}\right)\right)\right) = C^{(c)}\left(F_1^{(c)}\left(Y_{1j}\right),\dots,F_n^{(c)}\left(Y_{pj}\right)\right).$$

That is, there exist functions $f^{(c)} = \left\{f_i^{(c)}\right\}_{i=1}^{p}$ such that for each column vector of $\mathbf{Y}$, $Y_j = (Y_{1j}, \dots, Y_{pj})^\top,\ j = 1, \dots, n,\ Z_j^{(c)} \equiv f^{(c)}\left(Y_j\right) \sim \mathrm{mN}\left(\mathbf{0}_p, \mathbf{\Theta}^{-1}_{p\times p}\right)$, where $f^{(c)}\left(Y_j\right) = \left(f_1^{(c)}\left(Y_{1j}\right),\dots,f_p^{(c)}\left(Y_{pj}\right)\right)$. Then we say $Y_j = (Y_{1j}, \dots, Y_{pj})^\top$ has a nonparanormal distribution and write

$$Y_j \sim \mathrm{NPN}\left(\mathbf{0}_p, \mathbf{\Theta}^{-1}_{p\times p}, f^{(c)}\right).$$

The dependence between $Y_{1j}, \ldots, Y_{pj}$ can be illustrated by a Gauss-Markov Graph $G_c = \{V_c, E_c\}$ corresponding to precision matrix $\mathbf{\Theta}_{p\times p}$. This is equivalent to have latent variable $\mathbf{Z^{(c)}} = \mathbf{f^{(c)}}(\mathbf{Y_j}) \sim \text{mN}\left(\mathbf{0_p}, \mathbf{\Theta^{-1}_{p\times p}}\right), \mathbf{j} = \mathbf{1}, \ldots, \mathbf{n}$.

Consider the Cartesian product between G_c and G_r:

$$G_c \square G_r = \left(V_r \times V_c, \{(v_1, v_2), (v_1, v_2^{'}) | v_1 \in G_c, (v_2, v_2^{'}) \in E_r\} \bigcup \{(v_1, v_2), (v_1^{'}, v_2) | v_2 \in G_r, (v_1, v_1^{'}) \in E_c\}\right).$$

According to Theorem 4.3.5 in Knauer and Knauer (2019) (where Cartesian product we defined here was called Box product), the mapping $V_1 \times V_2 \longrightarrow G_c \square G_r$ is bimorphism.

From the perspective of Gauss-Markov graph, we propose to view that after the Cartesian product of G_c and G_r, the latent variables was mapped to a new set of latent variable $\mathbf{Z}$ for the total mateix $\mathbf{Y}$, $Z^{(c)} \times Z^{(r)} \longrightarrow$ Z. As in Gauss-Markov graph, the support of precision matrix defines the adjacency matrix of the corresponding graph, and by Cvetković et al. (1998), Cartesian product of graphs (Reffered to as "sum" in Cvetković et al. (1998)) corresponds to the Kronecker sum of Adjacency matrices. Therefore, the Cartesian product of Gauss-Markov graphs corresponds to the Kronecker sum of precision matrices.

Assume the overall graph illustrating relationships inside $\mathbf{Y}$ is the Cartesian product of the graph G_r and G_c, denoted as $G_r \square G_c$. Then the overall graph $G_r \square G_c$ is a Gauss-Markov Graph corresponding to precision matrix $\mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}$. Then we can assume for each Y_{ij}, there exists functions $f = \{f_{ij}\}_{\{i,j\}}$ and the latent variable $Z_{ij} = f_{ij}(Y_{ij})$ such that $Z^{(c)} \times Z^{(r)} \longrightarrow \mathbf{Z}$, and

$$\text{vec}(\mathbf{Z}) \equiv f(\text{vec}(\mathbf{Y})) \sim \text{mN}\left(\mathbf{0}_{np}, \mathbf{\Omega}^{-1}\right),$$

where $\mathbf{\Omega} = \mathbf{\Psi}_{n\times n} \oplus \mathbf{\Theta}_{p\times p}$ is the corresponding precision matrix.

B The effect of regularization parameters

Our algorithms depend on the regularization parameters β_1 and β_2. Figure 8 below illustrates the effect of these parameters on the performance of our algorithms. We generated two random sparse positive-definite matrices with a sparsity of 0.1 and non-zero entries normally distributed with mean 1 and variance 2. These were used as precision matrices $\mathbf{\Psi}_0$ and $\mathbf{\Theta}_0$ to create the Kronecker product matrix Ω_0 as plotted in Figure 7. This synthetic dataset corresponds to the experiment plotted in Figure 3 of our paper.

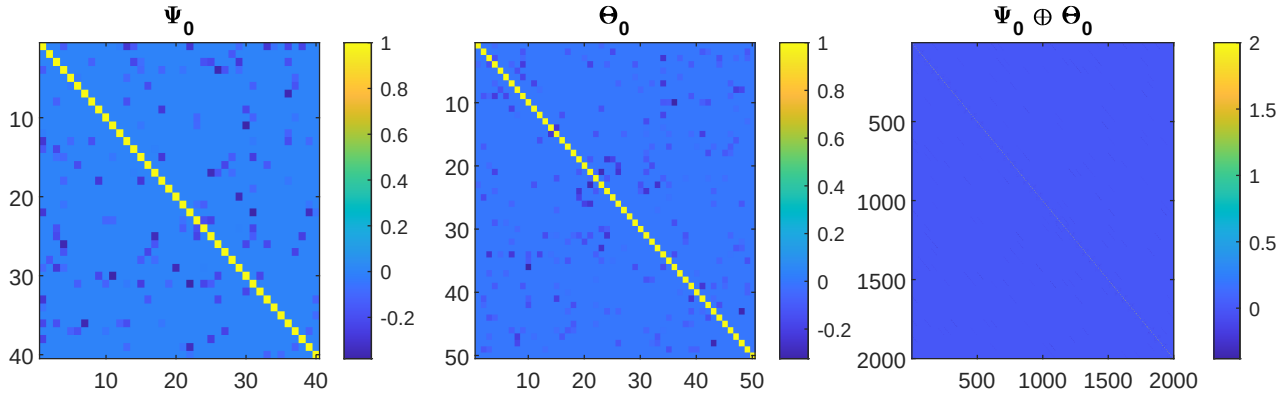


Figure 7: Precision matrix $\mathbf{\Psi}_0$ (left), $\mathbf{\Theta}_0$ (centre) and corresponding Kronecker product matrix Ω_0 (right) for our exemplar synthetic dataset.

In Figure 8(a)-(b) we show the Precision-Recall for the estimated precision matrices. In particular, subfigure (a) refers to the estimate of $\mathbf{\Psi}_{n\times n}$ when varying β_1, while subfigure (b) refers to the estimate of $\mathbf{\Theta}_{p\times p}$ when varying β_2. These curves suggest that optimal choices of β_1 lie within the interval $[0.007, 0.01]$ and similarly β_2 should lie within the interval $[0.006, 0.008]$. When choosing values within these intervals, one tries to strike a balance between Precision and Recall. In order to explore further the impact of the regularization parameters, we also

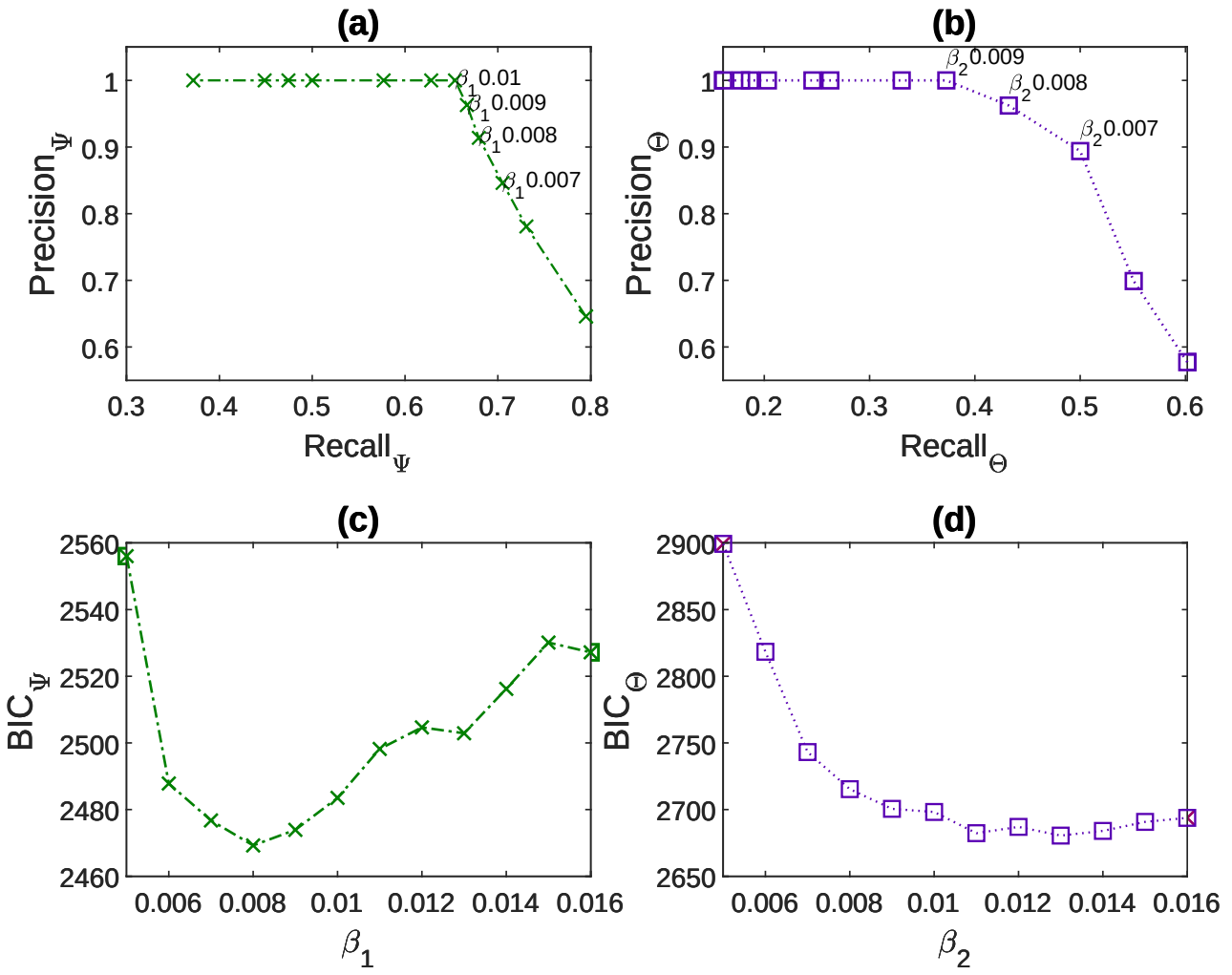


Figure 8: Synthetic network recovery results. Bayesian Information Criterion and regularization parameters. **(a)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Psi}_{n\times n}$; **(b)** Precision-Recall of the network recovery relating to the support of $\mathbf{\Theta}_{p\times p}$;
Bayesian Information Criterion and regularization parameters. **(c)** β_1-BIC_Ψ; **(d)** β_2-BIC_Θ;

computed the Bayesian Information Criteria (BIC) described in Schwarz (1978). In subfigures (c) and (d) we plot the BIC curves corresponding to the estimated precision matrices when varying β_1 and β_2 respectively. BIC is an heuristic criteria that helps selecting from several models. Ones with lower BIC values are generally preferred, however, a lower BIC does not necessarily indicate one model is better than another and further investigation is usually needed. The BIC curve depicted in subfigure (c) confirms the suggestion on the optimal choices for the regularization parameters obtained with the Precision-Recall plot, but the BIC curve in subfigure (d) suggest a different range for optimal regularization parameter in $[0.01, 0.016]$. Therefore, when dealing with problems without known truth, although BIC can be used to help identify the interval of potential optimal regularization parameters, it is not necessarily accurate and should be used with caution. Alternative methods to find the optimal regularization parameter should be explored in the future.