Biometrika (2021), **108**, 2, *pp.* 425–442
Printed in Great Britain
doi: 10.1093/biomet/asaa066
Advance Access publication 2 February 2021

Estimating differential latent variable graphical models with applications to brain connectivity

BY S. NA

Department of Statistics, University of Chicago,
5747 South Ellis Avenue, Chicago, Illinois 60637, U.S.A.
senna@uchicago.edu

M. KOLAR

Booth School of Business, University of Chicago,
5807 South Woodlawn Avenue, Chicago, Illinois 60637, U.S.A.
mladen.kolar@chicagobooth.edu

AND O. KOYEJO

Department of Computer Science, University of Illinois at Urbana-Champaign,
201 North Goodwin Avenue, Urbana, Illinois 61801, U.S.A.
sanmi@illinois.edu

SUMMARY

Differential graphical models are designed to represent the difference between the conditional dependence structures of two groups, and thus are of particular interest for scientific investigations. Motivated by modern applications, this manuscript considers an extended setting where each group is generated by a latent variable Gaussian graphical model. Due to the existence of latent factors, the differential network is decomposed into sparse and low-rank components, both of which are symmetric indefinite matrices. We estimate these two components simultaneously using a two-stage procedure: (i) an initialization stage, which computes a simple, consistent estimator, and (ii) a convergence stage, implemented using a projected alternating gradient descent algorithm applied to a nonconvex objective, initialized using the output of the first stage. We prove that given the initialization, the estimator converges linearly with a nontrivial, minimax optimal statistical error. Experiments on synthetic and real data illustrate that the proposed nonconvex procedure outperforms existing methods.

Some key words: Alternating projected gradient descent; Differential network; Functional connectivity; Latent variable Gaussian graphical model.

1. INTRODUCTION

Gaussian graphical models (Lauritzen, 1996) are used to capture complex relationships among observed variables in a variety of fields, ranging from computational biology (Friedman, 2004) and genetics (Lauritzen & Sheehan, 2003) to neuroscience (Smith et al., 2011). Each node in a graphical model represents an observed variable and the undirected edge between two nodes is present if the nodes are conditionally dependent given all the other variables; thus, graphical models are highly interpretable and have been adopted for a wide variety of applications.

Of particular interest in this manuscript are applications to cognitive neuroscience, specifically functional connectivity; the study of functional interactions between brain regions, thought to be necessary for cognition (Bullmore & Sporns, 2009). Importantly, functional connectivity is a promising biomarker for mental disorders (Castellanos et al., 2013), where the primary object of study is the differential network, that is, the differences in connectivity between healthy individuals and patients. See Bielza & Larrañaga (2014) for a detailed review. In genetics, scientists are interested in understanding differences in gene networks between experimental conditions, that is, the case-control study, to elucidate potential mechanisms underlying genetic functions. The differential network between two groups provides important signals for detecting differences. The interested reader can find more details on estimating genetic network differences in Hudson et al. (2009), de la Fuente (2010) and Ideker & Krogan (2012).

In many applications, it is clear that relationships between the observed variables are confounded by the presence of unobserved, latent factors. For example, physiological and demographic factors may have confounding effects on graphical model estimates in neuroscience and genetics (Gaggiotti et al., 2009; Willi & Hoffmann, 2009; Durkee et al., 2012). The standard approach of estimating sparse Gaussian graphical models is of limited use here as, due to confounding, the marginal precision matrix is not sparse. Instead of sparsity of the marginal graph, latent variable Gaussian graphical models exploit the observation that the marginal graph of the observed variables can be decomposed into a superposition of a sparse matrix and a low-rank matrix (Chandrasekaran et al., 2012; Meng et al., 2014).

This manuscript addresses the estimation of differential networks with latent factors. Suppose two groups of observed variables are drawn from latent variable Gaussian graphical models, and one is interested in differences in the conditional dependence structure between the two groups, which can be reduced to estimating the difference of their respective precision matrices. For this task, we develop a novel estimation procedure that does not require separate estimation for each group, which allows for robust estimation even if each group contains hub nodes. We propose a two-stage algorithm to optimize a nonconvex objective. In the first stage, we derive a simple, consistent estimator, which then serves as initialization for the next stage. In the second stage, we employ projected alternating gradient descent with a constant step size. The iterates are proven to linearly converge to a region around the ground truth, whose radius is characterized by the statistical error. Compared with potential convex approaches, our nonconvex approach would enjoy lower computation costs and hence be more time efficient. Extensive experiments validate our conceptual and theoretical claims.

2. BACKGROUND

2.1. *Notation*

Throughout the paper we use $\mathbb{S}^{d\times d}$, $\mathbb{Q}^{d\times d}$, I_d to denote the set of $d\times d$ symmetric, orthogonal matrices and identity matrices respectively. Given an integer d, we let $[d] = \{1, 2, \ldots, d\}$ be the index set. For any two scalars a and b, we write $a \lesssim b$ if $a \leqslant cb$ for some constant c. Similarly, $a \gtrsim b$ if $b \leqslant ca$ for some constant c. We write $a \asymp b$ if $a \lesssim b$ and $b \lesssim a$. We use $a \wedge b = \min(a, b)$ and $a \vee b = \max(a, b)$. For matrices $A, B \in \mathbb{S}^{d\times d}$, we write $A \prec B$ if $B - A$ is positive definite, and $A \preceq B$ if $B - A$ is positive semidefinite. We use $\langle A, B\rangle = \mathrm{tr}(A^{\mathrm{T}}B)$. For a matrix A, $\sigma_{\min}(A)$ and $\sigma_{\max}(A)$ denote the minimum and maximum singular values, respectively. For a vector a, $\|a\|_p$ denotes its ℓ_p norm, $p \geqslant 1$, and $\|a\|_0 = |\mathrm{supp}(a)|$ denotes the number of nonzero entries of a. For a matrix A, $\|A\|_p$ denotes the matrix induced p-norm, $\|A\|_{\mathrm{F}}$ denotes the Frobenius norm, $\|A\|_*$ denotes the nuclear norm, and $\|A\|_{p,\,q} = \big\{\sum_j \big(\sum_i |A_{ij}|^p\big)^{q/p}\big\}^{1/q}$, which

is calculated by computing the ℓ_q norm of the vector with each entry corresponding to the ℓ_p norm of a column of A. For example, $\|A\|_{0,q} = \|a\|_q$, where the jth entry of a is $a_j = \|A_{\cdot,j}\|_0$, and similarly $\|A\|_{p,\infty} = \|a\|_\infty$ with $a_j = \|A_{\cdot,j}\|_p$. Given a set $\mathcal{C} \subseteq \mathbb{R}^{d\times r}$, the projection operator $\mathcal{P}_{\mathcal{C}}(\cdot)$ is defined as $\mathcal{P}_{\mathcal{C}}(U) = \arg\min_{V\in\mathcal{C}} \|V - U\|_{\mathrm{F}}$.

2.2. *Preliminaries and related work*

A Gaussian graphical model (Lauritzen, 1996) consists of a graph $G = (V, E)$, where $V = \{1, \ldots, d\}$ is the set of vertices and E is the set of edges, and a d-dimensional random vector $X = (X_1, \ldots, X_d)^{\mathrm{T}} \sim N(\mu^\star_X, \Sigma^\star_X)$ that is Markov with respect to G. The precision matrix of X, $\Omega^\star_X = (\Sigma^\star_X)^{-1}$, encodes the conditional independence relationships underlying X and the graph structure G, where

$$X_i \perp\!\!\!\perp X_j \mid \{X_k : k \in V\backslash\{i,j\}\} \iff (i,j) \notin E \iff (\Omega^\star_X)_{i,j} = 0.$$

See Drton & Maathuis (2017) for a recent overview of the literature on structure learning of Gaussian graphical models with applications.

Latent variable Gaussian graphical models extend the applicability of Gaussian graphical models by assuming the existence of latent factors, $X_{\mathrm{H}} \in \mathbb{R}^r$, that confound the observed conditional independence structure of the observed variables $X_{\mathrm{O}} \in \mathbb{R}^d$. In particular, the observed and hidden components are assumed to be jointly normally distributed as $(X_{\mathrm{O}}^{\mathrm{T}}, X_{\mathrm{H}}^{\mathrm{T}})^{\mathrm{T}} \sim N(\mu^\star, \Sigma^\star)$ with

$$\mu^\star = \begin{pmatrix} \mu^\star_{X_{\mathrm{O}}} \\ \mu^\star_{X_{\mathrm{H}}} \end{pmatrix}, \quad \Sigma^\star = \begin{pmatrix} \Sigma^\star_{\mathrm{OO}} & \Sigma^\star_{\mathrm{OH}} \\ \Sigma^\star_{\mathrm{HO}} & \Sigma^\star_{\mathrm{HH}} \end{pmatrix}, \quad \Omega^\star = (\Sigma^\star)^{-1} = \begin{pmatrix} \Omega^\star_{\mathrm{OO}} & \Omega^\star_{\mathrm{OH}} \\ \Omega^\star_{\mathrm{HO}} & \Omega^\star_{\mathrm{HH}} \end{pmatrix}. \tag{1}$$

While the joint precision matrix $\Omega^\star$ is commonly assumed sparse, the marginal precision matrix of the observed component $X_{\mathrm{O}} \sim N(\mu^\star_{X_{\mathrm{O}}}, \Sigma^\star_{\mathrm{OO}})$ is given as

$$(\Sigma^\star_{\mathrm{OO}})^{-1} = \Omega^\star_{\mathrm{OO}} - \Omega^\star_{\mathrm{OH}}(\Omega^\star_{\mathrm{HH}})^{-1}\Omega^\star_{\mathrm{HO}}, \tag{2}$$

and in general is not sparse. The marginal precision matrix of observed variables X_{O} has a sparse plus low-rank structure, since the precision matrix of the conditional distribution of X_{O} given X_{H}, $\Omega^\star_{\mathrm{OO}} = \{\Sigma^\star_{\mathrm{OO}} - \Sigma^\star_{\mathrm{OH}}(\Sigma^\star_{\mathrm{HH}})^{-1}\Sigma^\star_{\mathrm{HO}}\}^{-1}$, is sparse and positive definite, while the second term in (2) is a rank-r positive semidefinite matrix, which in general is not sparse.

We study the problem of estimating the differential network, which is characterized by the difference between two precision matrices, from two groups of samples distributed according to latent variable Gaussian graphical models. More specifically, suppose that we have independent observations of d variables from two groups of subjects: $X_i = (X_{i1}, \ldots, X_{id})^{\mathrm{T}} \sim N(\mu^\star_X, \Sigma^\star_X)$ for $i = 1, \ldots, n_X$ from one group and $Y_i = (Y_{i1}, \ldots, Y_{id})^{\mathrm{T}} \sim N(\mu^\star_Y, \Sigma^\star_Y)$ for $i = 1, \ldots, n_Y$ from the other. The differential network is defined as the difference between two precision matrices, denoted as $\Delta^\star = \Omega^\star_X - \Omega^\star_Y$, where $\Omega^\star_X = (\Sigma^\star_X)^{-1}$ and $\Omega^\star_Y = (\Sigma^\star_Y)^{-1}$. We assume that the differential network can be decomposed as

$$\Delta^\star = S^\star + R^\star, \tag{3}$$

where $S^\star$ is sparse, $R^\star$ is low rank and they are both symmetric, but indefinite, matrices. Such a structure arises under the assumption that the group-specific precision matrices have the sparse plus low-rank structure as in (2). Here, $R^\star$ corresponds to the difference of two low-rank matrices,

whose rank is upper bounded by the sum of their ranks, hence it is natural for $R^\star$ to be low rank. However, imposing the sparse plus low-rank structure on the differential networks puts fewer restrictions on the data-generating process. For example, (3) also appears if one group is from the latent model while the other is from the regular graphical model.

Estimating the differential network $\Delta^\star$ can be naïvely achieved by estimating group-specific precision matrices first and then taking their difference. A related approach is to learn the group-specific precision matrices by maximizing the penalized joint likelihood of samples from both groups with a penalty that encourages the estimated precision matrices to have the same support. Both of these approaches require imposing strong assumptions on the individual precision matrices and are not robust in practice (Shojaie, 2020). For example, when hub nodes are present in a group-specific network (Barabási & Oltvai, 2004), estimation of an individual precision matrix is challenging as the sparsity assumption is violated, while direct estimation of the differential network is possible without imposing overly restrictive assumptions. Zhao et al. (2014) directly estimated the differential network $\Delta^\star$ by minimizing $\|\Delta\|_{1,1}$ subject to the constraint $\|\hat{\Sigma}_X \Delta \hat{\Sigma}_Y - (\hat{\Sigma}_Y - \hat{\Sigma}_X)\|_{\infty,\infty} \leqslant \lambda$. Under suitable conditions, the truncated and symmetrized estimator satisfies $\|\hat{\Delta} - \Delta^\star\|_{\mathrm{F}} \lesssim \{(n_X \wedge n_Y)^{-1} \|\Delta^\star\|_{0,1} \log d\}^{1/2}$. Liu et al. (2014) and Kim et al. (2019) developed procedures for estimation and inference of differential networks when X and Y follow a general exponential family distribution. See Shojaie (2020) for a recent review. In the presence of latent factors, the differential network is not guaranteed to be sparse and, therefore, the aforementioned methods are not applicable. We develop a methodology to learn the differential network from latent variable Gaussian graphical models.

Chandrasekaran et al. (2012) estimated a precision matrix under a latent variable Gaussian graphical model by minimizing the penalized negative Gaussian loglikelihood,

$$(\hat{S}_X, \hat{R}_X) = \arg\min_{S,R} \operatorname{tr}\{(S+R)\,\hat{\Sigma}_X\} - \log\det(S+R) + \lambda_n\big(\gamma\|S\|_{1,1} + \|R\|_*\big), \quad \text{subject to } S+R \succ 0, \quad -R \succeq 0, \tag{4}$$

where $\hat{\Sigma}_X$ is a sample covariance based on n_X samples. Under suitable identifiability and regularity conditions, $\gamma^{-1}\|\hat{S}_X - S_X^\star\|_{\infty,\infty} \vee \|\hat{R}_X - R_X^\star\|_2 \lesssim (d/n_X)^{1/2}$ when $\lambda_n \asymp (d/n_X)^{1/2}$. Meng et al. (2014) developed an alternating direction method of multipliers for more efficient minimization of (4) and showed that $\|\hat{\Omega}_X - \Omega_X^\star\|_{\mathrm{F}} \lesssim (s \log d/n_X)^{1/2} + (rd/n_X)^{1/2}$ with $s = \|S_X^\star\|_{0,1}$ being the overall sparsity of $S_X^\star$. The main drawback of minimizing (4) arises from the fact that in each iteration of the algorithm, the matrix R is updated without taking its low-rank structure into account. Xu et al. (2017) explicitly represented the low-rank matrix as $R = -UU^{\mathrm{T}}$ for $U \in \mathbb{R}^{d\times r}$ and minimized the resulting nonconvex objective using alternating gradient descent. Our alternating gradient descent procedure is closely related to this work, but more challenging in several aspects. First, the loglikelihood is not readily available for differential networks. We hence rely on a quasilikelihood, which reaches its minimum at $\Delta^\star$. Second, the low-rank matrix $R^\star$ in our set-up is indefinite, so we have to estimate the positive index of inertia for $R^\star$ as well. Third, in order to establish theoretical properties of our estimator, we avoid relying on the concentration of $\|\hat{\Sigma}_X\|_1$ that requires $n_X \asymp d^2$. By a more careful analysis, we improve the sample complexity to $n_X \asymp d \log d$.

Finally, our work is related to a growing literature on robust estimation where parameter matrices have the sparse plus low-rank structure. Example applications include robust principal component analysis (Candès et al., 2011; Chandrasekaran et al., 2011), robust matrix sensing (Fazel et al., 2008) and robust multi-task learning (Chen et al., 2011). Zhang et al. (2018) proposed a unified framework to analyse convergence of alternating gradient descent when applied on

sparse plus low-rank recovery. However, our problem is more challenging and does not satisfy conditions required by their framework. In particular, we use noisy covariance matrices to recover the difference of their true inverses via a quadratic loss. The Hessian matrix in our problem is $(\hat{\Sigma}_Y \otimes \hat{\Sigma}_X + \hat{\Sigma}_X \otimes \hat{\Sigma}_Y)/2$, with $\otimes$ denoting the Kronecker product, which is different compared to examples in robust estimation where the expectation of the Hessian is the identity. As a result, Condition 4.4 in Zhang et al. (2018) fails to hold and hence we need a problem-oriented analysis. We address three main technical challenges. First, the low-rank matrix $R^\star$ is indefinite, while the existing procedures only handle positive semidefinite low-rank matrices. We develop an estimator that consistently recovers the positive index of inertia of $R^\star$. Second, the analysis of the estimator is challenging as the incoherence condition is naturally imposed on $U^\star$, but in the analysis, e.g., when bounding the error in the gradient of the loss, the low-rank component $U^\star$ is always multiplied by a sample covariance matrix. Finally, we use properties of the Wishart distribution to provide finer analysis, avoid any concentration of sample covariance matrices in the $\|\cdot\|_1$ norm and improve the sample complexity established in Xu et al. (2017).

3. Methodology

3.1. *Empirical loss*

We introduce the estimator of the differential network $\Delta^\star$ based on observations from latent variable Gaussian graphical models described in § 2.2. Since $\Delta^\star$ satisfies $\left(\Sigma_X^\star \Delta^\star \Sigma_Y^\star + \Sigma_Y^\star \Delta^\star \Sigma_X^\star\right)/2 - (\Sigma_Y^\star - \Sigma_X^\star) = 0$, one can minimize the quadratic loss $\mathcal{L}(\Delta) = \mathrm{tr}\left\{\Delta \Sigma_X^\star \Delta \Sigma_Y^\star/2 - \Delta\left(\Sigma_Y^\star - \Sigma_X^\star\right)\right\}$. This loss has been used in Xu & Gu (2016) and Yuan et al. (2017) to learn sparse differential networks. Using the decomposition in (3) and substituting the true covariance matrices with sample estimates, we arrive at the following empirical loss:

$$\mathcal{L}_n(S, R) = \mathrm{tr}\left\{(S+R)\hat{\Sigma}_X(S+R)\hat{\Sigma}_Y/2 - (S+R)(\hat{\Sigma}_Y - \hat{\Sigma}_X)\right\}, \tag{5}$$

where $S \in \mathbb{S}^{d\times d}$ denotes the sparse component, $R \in \mathbb{S}^{d\times d}$ denotes the low-rank component with rank r, and $\hat{\Sigma}_X = n_X^{-1}\sum_{i=1}^{n_X}(X_i - \hat{\mu}_X)(X_i - \hat{\mu}_X)^{\mathrm{T}}$ with $\hat{\mu}_X = n_X^{-1}\sum_{i=1}^{n_X} X_i$ and $\hat{\Sigma}_Y$ is similarly defined. The empirical loss $\mathcal{L}_n(S, R)$ in (5) is convex with respect to the pair (S, R), and strongly convex if either of the two components is fixed.

Directly minimizing $\mathcal{L}_n(S, R)$ over a suitable constraint set would be computationally challenging as in each iteration R would need to be updated in $\mathbb{R}^{d\times d}$, without utilizing its low-rank structure. To that end, we explicitly factorize R as $R = U\Lambda U^{\mathrm{T}}$, where columns of $U \in \mathbb{R}^{d\times r}$ are aligned with eigenvectors that correspond to nonzero eigenvalues, and $\Lambda \in \mathbb{R}^{r\times r}$ is the diagonal sign matrix with diagonal elements being the sign of each eigenvalue. Without loss of generality, we assume Λ has $+1$ entries on the diagonal first, followed by -1 entries. This factorization implicitly imposes the constraints that $\mathrm{rank}(R) = r$ and $R = R^{\mathrm{T}}$. Different from estimating the single latent variable Gaussian graphical model in (2), where the low-rank component is positive semidefinite and can be factorized as $R = UU^{\mathrm{T}}$, $R^\star$ in our model (3) is only symmetric as it corresponds to the difference of two low-rank positive semidefinite matrices. Thus, $R^\star = U^\star \Lambda^\star U^{\star\mathrm{T}}$ and we need to estimate $\Lambda^\star$ as well. Plugging the factorization into (5), we aim to minimize the following empirical nonconvex objective:

$$\bar{\mathcal{L}}_n(S, U, \Lambda) = \mathcal{L}_n(S, U\Lambda U^{\mathrm{T}}) = \mathrm{tr}\big\{(S + U\Lambda U^{\mathrm{T}})\hat{\Sigma}_X(S + U\Lambda U^{\mathrm{T}})\hat{\Sigma}_Y/2 - (S + U\Lambda U^{\mathrm{T}})(\hat{\Sigma}_Y - \hat{\Sigma}_X)\big\}, \tag{6}$$

over a suitable constraint set that we discuss next.

We assume that $S^\star \in \mathbb{S}^{d\times d}$ has at most s nonzero entries overall, and each column, or row, has at most a certain fraction of nonzero entries. In particular, we assume

$$S^\star \in \mathcal{S}(\alpha, s) = \left\{S \in \mathbb{S}^{d\times d} : \|S\|_{0,1} \leqslant s, \|S\|_{0,\infty} \leqslant \alpha d\right\}$$

for some integer s and fraction $\alpha \in (0, 1)$. Furthermore, to make the low-rank component separable from the sum $S^\star + R^\star$, we require $R^\star$ to be not too sparse. One way to ensure identifiability is to impose the incoherence condition (Candès & Romberg, 2007), which prevents the information in column, or row, spaces of $R^\star$ from being concentrated in a few columns. The incoherence condition guarantees that the elements of $R^\star$ are roughly of the same magnitude and are not spiky. It is commonly used in the literature on low-rank matrix recovery (Chen et al., 2014; Chen, 2015; Yi et al., 2016). Specifically, suppose $R^\star = L^\star \Xi^\star L^{\star\mathrm{T}}$ is the reduced eigenvalue decomposition, where $L^\star \in \mathbb{R}^{d\times r}$ satisfies $L^{\star\mathrm{T}}L^\star = I_r$ and $\Xi^\star = \mathrm{diag}(\lambda_1^{R^\star}, \ldots, \lambda_r^{R^\star})$. Then, we assume $L^\star$ satisfies the β-incoherence condition, that is,

$$L^\star \in \mathcal{U}(\beta) = \left\{L \in \mathbb{R}^{d\times r} \mid \|L^{\mathrm{T}}\|_{2,\infty} \leqslant (\beta r/d)^{1/2}\right\}.$$

Without loss of generality, the eigenvalues are ordered so that, for some integer $r_1 \in \{0, \ldots, r\}$, $\mathrm{sign}(\lambda_i^{R^\star}) = 1$ for $1 \leqslant i \leqslant r_1$ and $\mathrm{sign}(\lambda_i^{R^\star}) = -1$ for $r_1 + 1 \leqslant i \leqslant r$. Here, r_1, called the positive index of inertia of $R^\star$, is unique by Sylvester's law of inertia (cf. Theorem 4.5.8 in Horn & Johnson, 2013), although eigenvalue decomposition is not.

3.2. *Two-stage algorithm*

We develop a two-stage algorithm to estimate the tuple $(S^\star, U^\star, \Lambda^\star)$. We start by introducing the second stage. Given a suitably chosen initial point (S^0, U^0, Λ^0), obtained by the first stage that we introduce later, we use the projected alternating gradient descent procedure to minimize the following nonconvex optimization problem:

$$\begin{aligned} &\min_{S,\,U}\ \bar{\mathcal{L}}_n(S, U, \Lambda^0) + \frac{1}{2}\|U_1^{\mathrm{T}}U_2\|_{\mathrm{F}}^2, \\ &\text{subject to } S \in \mathcal{S}(\bar{\alpha}, \bar{s}), \quad U \in \mathcal{U}(4\beta\|U^0\|_2^2), \end{aligned} \tag{7}$$

where $U = (U_1, U_2)$ with $U_1 \in \mathbb{R}^{d\times\hat{r}_1}$, $U_2 \in \mathbb{R}^{d\times(r-\hat{r}_1)}$, $\hat{r}_1$ is the number of $+1$ entries of Λ^0, used as an estimate of r_1, and $\bar{\alpha}, \bar{s}$ are user-defined tuning parameters. The quadratic penalty in (7) biases the components U_1, U_2 of the matrix U to be orthogonal, and can also be written as $\|U^{\mathrm{T}}U - \Lambda^0 U^{\mathrm{T}} U \Lambda^0\|_{\mathrm{F}}^2/16$.

Before we detail the steps of the algorithm, we define two truncation operators that correspond to two different sparsity structures. For any integer s and $A \in \mathbb{R}^{d\times d}$, the hard-truncation operator $\mathcal{J}_s(\cdot) : \mathbb{R}^{d\times d} \mapsto \mathbb{R}^{d\times d}$ is defined as

$$[\mathcal{J}_s(A)]_{i,j} = \begin{cases} A_{i,j} & \text{if } |A_{i,j}| \text{ is one of the largest } s \text{ elements of } A, \\ 0 & \text{otherwise.} \end{cases}$$

For any $\alpha \in (0, 1)$, the dispersed-truncation operator $\mathcal{T}_\alpha(\cdot) : \mathbb{R}^{d\times d} \mapsto \mathbb{R}^{d\times d}$ is defined as

$$[\mathcal{T}_\alpha(A)]_{i,j} = \begin{cases} A_{i,j} & \text{if } |A_{i,j}| \text{ is one of the largest } \alpha d \text{ elements for both } A_{i,\cdot} \text{ and } A_{\cdot,j}, \\ 0 & \text{otherwise.} \end{cases}$$

In the above definitions, $\mathcal{J}_s(A)$ keeps the largest s entries of A, while $\mathcal{T}_\alpha(A)$ keeps the largest α fraction of entries in each row and column. Therefore, the operator $\mathcal{J}_s(\cdot)$ projects iterates to the constraint set $\|S\|_{0,1} \leqslant s$, while $\mathcal{T}_\alpha(\cdot)$ projects to the set $\|S\|_{0,\infty} \leqslant \alpha d$.

We summarize the projected alternating gradient descent procedure in Algorithm 1. Both the sparse and low-rank components are updated, with the other component being fixed, by the gradient descent step with a constant step size, followed by a projection step. Explicit formulas for $\nabla_S \bar{\mathcal{L}}_n$ and $\nabla_U \bar{\mathcal{L}}_n$ are provided in the Supplementary Material. The sign matrix Λ^0 is not updated in the algorithm. We will show later that, under suitable conditions, the first-stage estimate consistently recovers $\Lambda^\star$, that is, $\Lambda^0 = \Lambda^\star$. Computationally, the update of the low-rank matrix in each iteration requires only updating the factor U, which can be done efficiently.

Algorithm 1. Stage II: projected alternating gradient descent for solving (7).

Input: Sample covariance matrices $\hat{\Sigma}_X$, $\hat{\Sigma}_Y$; initial point tuple (S^0, U^0, Λ^0); step sizes η_1, η_2; tuning parameters $\bar{\alpha}$, $\bar{s}$, β.

For $k = 0$ to $k = K - 1$

$S^{k+1/2} = S^k - \eta_1 \nabla_S \bar{\mathcal{L}}_n(S^k, U^k, \Lambda^0)$;

$S^{k+1} = \mathcal{T}_{\bar{\alpha}}\left\{\mathcal{J}_{\bar{s}}(S^{k+1/2})\right\}$;

Let $\mathcal{C}^k = \mathcal{U}(4\beta\|U^k\|_2^2)$;

$U^{k+1/2} = U^k - \eta_2 \nabla_U \bar{\mathcal{L}}_n(S^k, U^k, \Lambda^0) - \frac{\eta_2}{2} U^k (U^{k\mathrm{T}} U^k - \Lambda^0 U^{k\mathrm{T}} U^k \Lambda^0)$;

$U^{k+1} = \mathcal{P}_{\mathcal{C}^k}(U^{k+1/2})$;

Output S^K, U^K.

The projection operator $\mathcal{P}_{\mathcal{U}(\beta)}(\cdot)$ can be computed in a closed form as

$$[\mathcal{P}_{\mathcal{U}(\beta)}(U)]_{i,\cdot} = \begin{cases} U_{i,\cdot} & \text{if } \|U_{i,\cdot}\|_2 \leqslant (\beta r/d)^{1/2}, \\ (\beta r/d)^{1/2}/\|U_{i,\cdot}\|_2 \cdot U_{i,\cdot} & \text{otherwise.} \end{cases}$$

Next, we describe how to get a good initial point, (S^0, U^0, Λ^0), needed for Algorithm 1. The requirements on the initial point are presented in Theorem 1. Our initial point is obtained from a rough estimator of $\Delta^\star$. Let $\hat{\Delta}^0 = (\tilde{\Sigma}_X)^{-1} - (\tilde{\Sigma}_Y)^{-1}$, where $\tilde{\Sigma}_X = n_X/(n_X - d - 2)\,\hat{\Sigma}_X$, similarly for $\tilde{\Sigma}_Y$, is the scaled sample covariance matrix. The scaled covariance matrix, the so-called Kaufman–Hartlap correction (Paz & Sánchez, 2015), is used for the initialization step so that $\mathbb{E}(\tilde{\Sigma}_X^{-1}) = \Omega_X^\star$. By rescaling the sample covariance, we are able to show that $\|(\tilde{\Sigma}_X)^{-1} - \Omega_X^\star\|_{\infty,\infty} \asymp (\log d/n_X)^{1/2}$ with high probability, leading to a better sample size compared to $\|(\hat{\Sigma}_X)^{-1} - \Omega_X^\star\|_{\infty,\infty} \asymp d/n_X + (\log d/n_X)^{1/2}$. We obtain S^0 by truncating $\hat{\Delta}^0$. Next, we extract r eigenvectors, corresponding to the top r eigenvalues in magnitude of the residual matrix $R^0 = \hat{\Delta}^0 - S^0$. Then U^0 and Λ^0 are further derived from the reduced matrix. See Algorithm 2 for details. Theorem 2 shows that the positive index of inertia is correctly recovered by the initial step, $\Lambda^0 = \Lambda^\star$, and (S^0, U^0) lies in a sufficiently small neighbourhood of $(S^\star, U^\star)$.

Throughout the two-stage algorithm, we only compute the (reduced) eigenvalue decomposition once in the first stage. Therefore, it is computationally efficient compared to related convex approaches, mentioned in the Supplementary Material, where in each iteration one needs to compute an eigenvalue decomposition to update R.

In our experiments, we set $\bar{\alpha} = \hat{\alpha}$ and $\bar{s} = \hat{s}$, and use cross-validation to select them together with r and β. Our theory requires more stringent conditions on $\bar{\alpha}, \bar{s}$ in Algorithm 1 than on $\hat{\alpha}, \hat{s}$ in Algorithm 2, where we only require $\hat{\alpha} \geqslant \alpha$ and $\hat{s} \geqslant s$. See Theorems 1 and 2.

Algorithm 2. Stage I: initialization.

Input: Scaled sample covariance matrices $\tilde{\Sigma}_X$, $\tilde{\Sigma}_Y$; tuning parameters $\hat{\alpha}$, $\hat{s}$, r, β.
Let $\hat{\Delta}^0 = (\tilde{\Sigma}_X)^{-1} - (\tilde{\Sigma}_Y)^{-1}$, $S^0 = \mathcal{T}_{\hat{\alpha}}\{\mathcal{J}_{\hat{s}}(\hat{\Delta}^0)\}$, $R^0 = \hat{\Delta}^0 - S^0$;
Compute $R^0 = L^0 \Xi^0 L^{0\mathrm{T}}$, the eigenvalue decomposition of R^0. Let $\Xi^0_r \in \mathbb{R}^{r\times r}$ be the diagonal matrix with largest r eigenvalues in magnitude and $L^0_r \in \mathbb{R}^{d\times r}$ be the corresponding eigenvectors;
Let $\hat{r}_1 = |\{i \in [r] : [\Xi^0_r]_{i,i} > 0\}|$, $\Lambda^0 = \mathrm{diag}(I_{\hat{r}_1}, -I_{r-\hat{r}_1})$, and P^0 be the permutation matrix such that $\mathrm{sign}(\Xi^0_r) = P^0 \Lambda^0 P^{0\mathrm{T}}$;
Let $\bar{U}^0 = L^0_r |\Xi^0_r|^{1/2} P^0$, where $|\Xi^0_r|$ is computed elementwise;
Let $U^0 = \mathcal{P}_{\mathcal{C}}\left(\bar{U}^0\right)$ with $\mathcal{C} = \mathcal{U}(4\beta \|\bar{U}^0\|_2^2)$;
Output S^0, U^0, Λ^0.

4. THEORETICAL ANALYSIS

We establish the convergence rate of iterates generated by Algorithm 1 by first assuming that the initial point (S^0, U^0, Λ^0) lies in a suitable neighbourhood around $(S^\star, U^\star, \Lambda^\star)$. Next, we prove that the output of Algorithm 2 satisfies the requirements on the initial point with high probability. The convergence rate of Algorithm 1 consists of two parts: the statistical rate and the algorithmic rate. The statistical rate appears due to the approximation of population loss by the empirical loss, and it depends on the sample size, dimension and the problem parameters, including the condition numbers of covariance matrices. The algorithmic rate characterizes the linear rate of convergence of the projected gradient descent iterates to a point that is within statistical error of the true parameters.

The convergence rate is established under the following two assumptions.

Assumption 1. (*Constraint sets.*) Let $\Delta^\star = S^\star + R^\star$ be the differential network and $R^\star = L^\star \Xi^\star L^{\star\mathrm{T}}$ be the reduced eigenvalue decomposition of the rank-r matrix $R^\star$. There exist α, β and s such that $S^\star \in \mathcal{S}(\alpha, s)$ and $L^\star \in \mathcal{U}(\beta)$.

Assumption 2. There exist $0 < \sigma^X_d \leqslant \sigma^X_1 < \infty$ and $0 < \sigma^Y_d \leqslant \sigma^Y_1 < \infty$ such that $\sigma^X_d I_d \preceq \Sigma^\star_X \preceq \sigma^X_1 I_d$ and $\sigma^Y_d I_d \preceq \Sigma^\star_Y \preceq \sigma^Y_1 I_d$.

We start by defining the distance function that will be used to measure the convergence rate of the low-rank component. From the reduced eigenvalue decomposition of $R^\star$, $R^\star = L^\star \Xi^\star L^{\star\mathrm{T}} = U^\star \Lambda^\star U^{\star\mathrm{T}}$ with $\Lambda^\star = \mathrm{sign}(\Xi^\star) = \mathrm{diag}(I_{r_1}, -I_{r-r_1})$ and $U^\star = L^\star(\Xi^\star \Lambda^\star)^{1/2}$. While $\Lambda^\star$ is uniquely characterized by the positive index of inertia r_1, $U^\star$ is not unique in the sense that it is possible to have $U^\star \Lambda^\star U^{\star\mathrm{T}} = U \Lambda^\star U^{\mathrm{T}}$, but $U \neq U^\star$. We deal with this nonuniqueness issue by using the following distance function.

DEFINITION 1 (Distance function). *Given two matrices* $U_1, U_2 \in \mathbb{R}^{d\times r}$ *and an integer* $r' \in \{0, \ldots, r\}$, *we define* $\Pi_{r'}(U_1, U_2) = \inf_{Q \in \mathcal{Q}^{r\times r}_{r'}} \|U_1 - U_2 Q\|_{\mathrm{F}}$, *where*

$$\mathcal{Q}_{r'}^{r\times r} = \left\{Q \in \mathbb{Q}^{r\times r} : Q\Lambda Q^{\mathrm{T}} = \Lambda \textit{ with } \Lambda = \mathrm{diag}(I_{r'}, -I_{r-r'})\right\}$$
$$= \left\{Q \in \mathbb{Q}^{r\times r} : Q = \mathrm{diag}(Q_1, Q_2) \textit{ with } Q_1 \in \mathbb{Q}^{r'\times r'}, Q_2 \in \mathbb{Q}^{r-r'\times r-r'}\right\}.$$

In the following, we will simply use $\Pi(\cdot,\cdot)$ to represent $\Pi_{r_1}(\cdot,\cdot)$ with r_1 being the positive inertia of $R^\star$. Based on the following lemma, we see that $\Pi(U, U^\star)$ measures $\|U\Lambda^\star U^{\mathrm{T}} - U^\star\Lambda^\star U^{\star\mathrm{T}}\|_{\mathrm{F}}$.

LEMMA 1 (Properties of $\Pi(\cdot,\cdot)$). *Suppose $U^\star \in \mathbb{R}^{d\times r}$ has orthogonal columns and $\Lambda^\star = \mathrm{diag}(I_{r_1}, I_{r-r_1})$. Let σ_1 (σ_r) be the largest (smallest) singular value of $U^\star$ and let $U \in \mathbb{R}^{d\times r}$.*

(a) *If $\Pi(U, U^\star) \leqslant \sigma_1$, then $\|U\Lambda^\star U^{\mathrm{T}} - U^\star\Lambda^\star U^{\star\mathrm{T}}\|_{\mathrm{F}} \leqslant 3\sigma_1\Pi(U, U^\star)$.*
(b) *If $\|U\Lambda^\star U^{\mathrm{T}} - U^\star\Lambda^\star U^{\star\mathrm{T}}\|_2 \leqslant \sigma_r^2/2$, then $\Pi(U, U^\star) \leqslant \{(\sqrt{2}-1)^{1/2}\sigma_r\}^{-1}\|U\Lambda^\star U^{\mathrm{T}} - U^\star\Lambda^\star U^{\star\mathrm{T}}\|_{\mathrm{F}}$.*

By Lemma 1, $U\Lambda^\star U^{\mathrm{T}} = U^\star\Lambda^\star U^{\star\mathrm{T}} \iff \Pi^2(U, U^\star) = 0$. Thus, once we can correctly recover $\Lambda^\star$, that is $\hat{\Lambda} = \Lambda^\star$, the distance function in Definition 1 is a reasonable surrogate for $\|\hat{R} - R^\star\|_{\mathrm{F}}$, since $\|\hat{R} - R^\star\|_{\mathrm{F}} = \|\hat{U}\hat{\Lambda}\hat{U}^{\mathrm{T}} - U^\star\Lambda^\star U^{\star\mathrm{T}}\|_{\mathrm{F}} = \|\hat{U}\Lambda^\star\hat{U}^{\mathrm{T}} - U^\star\Lambda^\star U^{\star\mathrm{T}}\|_{\mathrm{F}} \asymp \Pi(\hat{U}, U^\star)$.

Let $\sigma_1^{R^\star} = \sigma_{\max}(R^\star)$, $\sigma_r^{R^\star} = \sigma_{\min}(R^\star)$ and define the condition numbers $\kappa_X = \sigma_1^X/\sigma_d^X$, $\kappa_Y = \sigma_1^Y/\sigma_d^Y$ and $\kappa_{R^\star} = \sigma_1^{R^\star}/\sigma_r^{R^\star}$. We further define the following quantities that depend only on the covariance matrices:

$$T_1 = \left\{\frac{\kappa_X\kappa_Y\left(\|\Omega_Y^\star\|_1\|\Sigma_X^\star\|_1 + \|\Omega_X^\star\|_1\|\Sigma_Y^\star\|_1\right)}{\sigma_d^X\sigma_d^Y}\right\}^2, \qquad T_2 = \left(\frac{1}{\sigma_d^X} + \frac{1}{\sigma_d^X}\right)^2,$$
$$T_3 = \left(\frac{\|\Sigma_X^\star\|_1}{\sigma_1^X}\right)^2 + \left(\frac{\|\Sigma_Y^\star\|_1}{\sigma_1^Y}\right)^2, \qquad T_4 = \frac{(\sigma_d^X\sigma_d^Y)^2}{\kappa_X^4\kappa_Y^4\left\{(\sigma_1^Y\|\Sigma_X^\star\|_1)^2 + (\sigma_1^X\|\Sigma_Y^\star\|)^2\right\}},$$
$$T_5 = \left\{\|(\Omega_X^\star)^{1/2}\|_1^2 + \|(\Omega_Y^\star)^{1/2}\|_1^2\right\}^2, \qquad T_6 = \left(\frac{\kappa_X}{\sigma_d^X} + \frac{\kappa_Y}{\sigma_d^Y}\right)^2.$$

Finally, for $S \in \mathbb{S}^{d\times d}$ and $U \in \mathbb{R}^{d\times r}$, we define the total error distance to be

$$TD(S, U) = \|S - S^\star\|_{\mathrm{F}}^2/\sigma_1^{R^\star} + \Pi^2(U, U^\star).$$

The error for the sparse component is scaled by $\sigma_1^{R^\star}$ in order to have the two error terms on the same scale, based on the first part of Lemma 1. With this, we have the following result on the convergence of iterates obtained by Algorithm 1.

THEOREM 1 (Convergence of Algorithm 1). *Suppose Assumptions 1 and 2 hold. Furthermore, suppose the following conditions hold:*

(a) *sample size*

$$(n_X \wedge n_Y) \geqslant C_1\left\{\frac{d\log d}{T_3\beta} \vee \frac{(\kappa_X\kappa_Y)^4\left(T_1\cdot s\log d + T_2\cdot rd\right)}{(\sigma_r^{R^\star})^2}\right\}, \tag{8}$$

and sparsity proportion $\alpha \leqslant c_1 T_4/(\beta r \kappa_{R^\star})$;

(b) *step sizes* $\eta_1 \leqslant c_2/\left(\sigma_1^X \sigma_1^Y \kappa_X \kappa_Y\right)$, $\eta_2 = c_3 \eta_1/\sigma_1^{R^\star}$, *and tuning parameters* $2(\bar{s}/s) - 1 \geqslant \bar{\alpha}/\alpha \geqslant C_2 (\kappa_X \kappa_Y)^4$;

(c) *initialization point* $\Lambda^0 = \Lambda^\star$, $S^0 \in \mathbb{S}^{d\times d}$, $U^0 \in \mathcal{U}(9\beta\sigma_1^{R^\star})$ *with* $TD(S^0, U^0) \leqslant c_4 \sigma_r^{R^\star}/(\kappa_X \kappa_Y)^2$.

Then the iterates (S^k, U^k) *of Algorithm 1 satisfy* $S^k \in \mathbb{S}^{d\times d}$ *and*

$$TD(S^k, U^k) \leqslant \left(1 - \frac{c_5}{\kappa_X^2 \kappa_Y^2 \kappa_{R^\star}}\right)^k TD(S^0, U^0) + \frac{C_3 \kappa_X^2 \kappa_Y^2}{\sigma_r^{R^\star}} \cdot \frac{T_1 \cdot s \log d + T_2 \cdot rd}{n_X \wedge n_Y} \tag{9}$$

with probability at least $1 - C_4/d^2$ *for some fixed constants* $(C_i)_{i=1}^4$ *sufficiently large and* $(c_i)_{i=1}^5$ *sufficiently small.*

The two terms in (9) correspond to the algorithmic and the statistical rate of convergence, respectively. The statistical error is of the order $O\{(s \log d + rd)/(n_X \wedge n_Y)\}$, which matches the minimax optimal rate (Chandrasekaran et al., 2012). In particular, the term $O\{s \log d/(n_X \wedge n_Y)\}$ corresponds to the statistical error of estimating $S^\star$, while $O\{rd/(n_X \wedge n_Y)\}$ corresponds to the statistical error of estimating $R^\star$. We stress that the condition on α is common in the related literature. For example, Yi et al. (2016) require $\alpha \lesssim 1/\beta r(\kappa_{R^\star})^2$, which is stronger than our condition in terms of the power of $\kappa_{R^\star}$, and Zhang et al. (2018) require $\alpha \lesssim 1/\beta r \kappa_{R^\star}$, which is comparable with ours. Under the condition on α, we have $\bar{\alpha} < 1$. The sample complexity requirement in (8) has an extra $d \log d/\beta$ term compared to typical results in robust estimation. See Corollaries 4.11 and 4.13 in Zhang et al. (2018) for results in robust matrix sensing and robust principal component analysis. This increased sample complexity is common in estimating latent variable Gaussian graphical models. For example, Xu et al. (2017) require $n_X \gtrsim d^2$ to show convergence of $\|\hat{\Sigma}_X\|_1$. Theorem 1 improves the sample size requirement to $d \log d$. In (6), we need to control the low-rank components $\hat{\Sigma}_X U^\star$ (and $\hat{\Sigma}_Y U^\star$), and the large sample size guarantees that the incoherence condition can transfer from $U^\star$ to $\hat{\Sigma}_X U^\star$. Furthermore, the covariance matrices $\hat{\Sigma}_X$ and $\hat{\Sigma}_Y$ work as design matrices in (6), and bring additional challenges compared to robust estimation problems. The design matrix in robust principal component analysis is the identity, while in robust matrix sensing its expectation is also the identity. Thus, their loss functions all satisfy Condition 4.4 in Zhang et al. (2018), which is not the case for (6). Without Condition 4.4, their proof strategy fails to show the convergence of the alternating gradient descent. By direct analysis, we first establish what conditions we need on $\hat{\Sigma}_X U^\star$ and $\hat{\Sigma}_Y U^\star$, and then show that these conditions hold under the incoherence condition on $U^\star$. Finally, we observe that the algorithmic error decreases exponentially and, after $O[\log\{n_X \wedge n_Y/(s \log d + rd)\}]$ iterations, the statistical error is the dominant term.

Next, we show that the output (S^0, U^0, Λ^0) of Algorithm 2 satisfies the requirements on the initialization point of Algorithm 1 presented in condition (c) of Theorem 1. The requirement that $S^0 \in \mathbb{S}^{d\times d}$ is easy to achieve. The following lemma suggests that $\Lambda^0 = \Lambda^\star$ is implied by an upper bound on $\|R^0 - R^\star\|_2$, which further connects to the upper bound on $TD(S^0, U^0)$ by Lemma 1.

LEMMA 2. *For any* $R \in \mathbb{S}^{d\times d}$, *let* $R = L \Xi L^{\mathrm{T}}$ *be the eigenvalue decomposition. Let* $\Xi_r \in \mathbb{R}^{r\times r}$ *be the diagonal matrix with the* r *largest entries of* Ξ *in magnitude, and let* $\hat{r}_1$ *be the number of positive entries of* Ξ_r. *If* $\|R - R^\star\|_2 \leqslant \sigma_r^{R^\star}/3$, *then* $\hat{r}_1 = r_1$ *and* $\Lambda_r = \mathrm{diag}(I_{\hat{r}_1}, -I_{r-\hat{r}_1}) = \Lambda^\star$.

The next theorem shows the sample complexity under which the conditions on the initial point are satisfied and $\|R^0 - R^\star\|_2 \leqslant \sigma_r^{R^\star}/3$, which implies $\Lambda^0 = \Lambda^\star$, using Lemma 2.

THEOREM 2 (Initialization). *Suppose Assumptions 1 and 2 hold. If* $\hat{\alpha} \geqslant \alpha$, $\hat{s} \geqslant s$ *and the sample sizes and dimension satisfy*

$$(n_X \wedge n_Y) \geqslant \frac{C_1\left(T_5 \hat{s} \log d + T_6 d\right)}{(\sigma_r^{R^\star})^2}, \qquad d \geqslant C_2 \beta \hat{s}^{1/2} r \kappa_{R^\star},$$

then $S^0 \in \mathbb{S}^{d\times d}$, $\|R^0 - R^\star\|_2 \leqslant \sigma_r^{R^\star}/4$, $U^0 \in \mathcal{U}(9\beta\sigma_1^{R^\star})$ *and*

$$TD(S^0, U^0) \leqslant C_3 \left\{ \frac{r\left(T_5 \cdot \hat{s} \log d + T_6 \cdot d\right)}{\sigma_r^{R^\star}(n_X \wedge n_Y)} + \frac{\hat{s}\beta^2 r^3 \kappa_{R^\star} \sigma_1^{R^\star}}{d^2} \right\}$$

with probability $1 - C_4/d^2$ *for some fixed constants* $(C_1)_{i=1}^4$ *sufficiently large. Furthermore, if* $\hat{s} \asymp s$,

$$(n_X \wedge n_Y) \gtrsim \frac{r\kappa_X^2 \kappa_Y^2}{(\sigma_r^{R^\star})^2}\left(T_5 \cdot s \log d + T_6 \cdot d\right), \qquad d \gtrsim \beta s^{1/2} r^{3/2} \kappa_{R^\star} \kappa_X \kappa_Y, \tag{10}$$

then $TD(S^0, U^0) \lesssim \sigma_r^{R^\star} / (\kappa_X \kappa_Y)^2$.

From Theorem 2, the requirement for the initial point is satisfied under (10). The sample complexity required for initialization, $O\{(rs \log d + rd)/(\sigma_r^{R^\star})^2\}$, is smaller than that for convergence, $O\{d \log d/\beta + (s \log d + rd)/(\sigma_r^{R^\star})^2\}$. When $d \gtrsim \beta s^{1/2} r^{3/2} \kappa_{R^\star} \kappa_X \kappa_Y$ and $\alpha \lesssim T_4/(\beta r \kappa_{R^\star})$, combining Theorems 1 and 2 shows that the iterates generated by the two-stage algorithm converge linearly to a point with an unavoidable minimax optimal statistical error.

We briefly discuss exact recovery of the support of $S^\star$ and the rank of $R^\star$. Throughout the paper we assume that the rank r of $R^\star$ is known. This assumption is commonly used in the literature on alternating gradient descent for low-rank matrix recovery (e.g., Yi et al., 2016; Xu et al., 2017; Zhang et al., 2018). The rank r is used to truncate the eigenvalues of R^0 and to choose the number of columns of the iterates U in Algorithm 1. However, we note that the rank r can be exactly recovered under a suitable assumption on the signal strength, $\sigma_r^{R^\star}$. After dropping higher-order terms, Theorem 2 shows that $\|R^0 - R^\star\|_2 \lesssim \{d/(n_X \wedge n_Y)\}^{1/2}$. Therefore, if $\sigma_r^{R^\star} \gtrsim 2\{d/(n_X \wedge n_Y)\}^{1/2}$, one can recover r by thresholding small eigenvalues of R^0. From the proof of Lemma S5 in the Supplementary Material, $\|S^0 - S^\star\|_{\infty,\infty} \leqslant \{\log d/(n_x \wedge n_Y)\}^{1/2}$, which allows us to recover the support of $S^\star$ by thresholding elements of S^0 that are smaller in magnitude than $\{\log d/(n_x \wedge n_Y)\}^{1/2}$ if the nonzero elements of $S^\star$ are bigger than $2\{\log d/(n_x \wedge n_Y)\}^{1/2}$ in magnitude. Finally, if the support set of $S^\star$ is consistently estimated, then α and s can be estimated as well. Therefore, under suitable assumptions on the signal strength, α, s and r are all consistently estimable. Similar signal strength assumptions are also needed even for convex approaches to exactly recover the sparsity and rank (Chandrasekaran et al., 2012; Zhao et al., 2014).

5. SIMULATIONS

5.1. *Data generation and implementation details*

We compare the performance of our estimator with two procedures that directly learn the differential network under the sparsity assumption, ℓ_1-minimization (Zhao et al., 2014) and ℓ_1-penalized quadratic loss (Yuan et al., 2017), and two procedures that separately learn latent

Table 1. *Competing methods*

Abbr.	Reference	Type	Set-up
M1	Zhao et al. (2014)	Joint, convex	Differential network is sparse
M2	Yuan et al. (2017)	Joint, convex	Differential network is sparse
M3	Chandrasekaran et al. (2012)	Separate, convex	Single network is sparse + low-rank
M4	Xu et al. (2017)	Separate, nonconvex	Single network is sparse + low-rank
M*	Present paper	Joint, nonconvex	Differential network is sparse + low-rank

variable Gaussian graphical models, sparse plus low-rank penalized Gaussian likelihood (4) (Chandrasekaran et al., 2012) and constrained Gaussian likelihood (Xu et al., 2017). Table 1 summarizes the procedures. In the Supplementary Material we provide additional simulation results, including comparison with alternative convex approaches.

Data are generated from the latent variable Gaussian graphical model (1) described in § 2.2. We set $\mu^\star_{X_\mathrm{O}} = \mu^\star_{X_\mathrm{H}} = \mu^\star_{Y_\mathrm{O}} = \mu^\star_{Y_\mathrm{H}} = 0$. The blocks of $\Omega^\star$ are generated separately. For $\Omega^\star_{\mathrm{OO}} \in \mathbb{R}^{d\times d}$ we set the diagonal entries to be one and, following Xia et al. (2015), the off-diagonal entries to be generated according to one of the following four models:

Model 1. $(\Omega^{\star(1)}_{\mathrm{OO}})_{i,i+1} = (\Omega^{\star(1)}_{\mathrm{OO}})_{i+1,i} = 0.6$, $(\Omega^{\star(1)}_{\mathrm{OO}})_{i,i+2} = (\Omega^{\star(1)}_{\mathrm{OO}})_{i+2,i} = 0.3$;

Model 2. $(\Omega^{\star(2)}_{\mathrm{OO}})_{i,j} = (\Omega^{\star(2)}_{\mathrm{OO}})_{j,i} = 0.5$ for $i = 10k-9$, $10k-6 \leqslant j \leqslant 10k$, $1 \leqslant k \leqslant d/10$;

Model 3. $(\Omega^{\star(3)}_{\mathrm{OO}})_{i,j} = (\Omega^{\star(3)}_{\mathrm{OO}})_{j,i} \sim 0.8 \cdot \mathrm{Ber}(0.1)$ for $i+1 \leqslant j \leqslant i+3$;

Model 4. $(\Omega^{\star(4)}_{\mathrm{OO}})_{i,j} = (\Omega^{\star(4)}_{\mathrm{OO}})_{j,i} \sim 0.5 \cdot \mathrm{Ber}(0.5)$ for $i = 2k-1$, $2k \leqslant j \leqslant (2k+2) \wedge d$, $1 \leqslant k \leqslant d/2$.

The blocks $\Omega^\star_{\mathrm{OH}}, \Omega^\star_{\mathrm{HO}}$ are generated entrywise from the following mixture distribution:

$$(\Omega^\star_{\mathrm{HO}})_{j,i} = (\Omega^\star_{\mathrm{OH}})_{i,j} \sim 0.1 \cdot \delta_0 + 0.9 \cdot \mathrm{Un}(0.5, 1) \qquad (i = 1, \ldots, d,\ j = 1, \ldots, r),$$

and $\Omega^\star_{\mathrm{HH}} = I_r$. Combining the blocks, we get the following four models:

$$\Omega^{\star(i)} = \begin{pmatrix} \Omega^{\star(i)}_{\mathrm{OO}} & \Omega^\star_{\mathrm{OH}} \\ \Omega^\star_{\mathrm{HO}} & \Omega^\star_{\mathrm{HH}} \end{pmatrix}, \qquad i = 1, 2, 3, 4.$$

Last, we let $\Sigma^\star_i = [D^{1/2}\{\Omega^{\star(i)} + (\iota_i + 1)I_{d+r}\}D^{1/2}]^{-1}$, where $\iota_i = \left|\min\left\{\mathrm{eig}(\Omega^{\star(i)})\right\}\right|$ and $D \in \mathbb{R}^{(d+r)\times(d+r)}$ is a diagonal scaling matrix with $D_{i,i} \sim \mathrm{Un}(0.5, 2.5)$. In our models, each latent variable is connected to roughly 90% of observed covariates, and hence the effect of latent variables is spread out and the corresponding low-rank matrix is incoherent (Chandrasekaran et al., 2012). We generate X using $\Sigma^\star_1$ and denote it as the control group, while generating Y using $\Sigma^\star_i$, $i = 2, 3, 4$, and denote it as the test $i-1$ group. Under this generation process, both X_O and Y_O have precision matrices with sparse plus low-rank structure.

Throughout the simulations, we set the sample size equal for both groups, $n_X = n_Y = n$. For each combination of the tuple (n, d, r), we generate a training and a validation set with sample size n. For each method, we choose the corresponding tuning parameters that minimize the empirical loss $\mathcal{L}_n(\hat{S}, \hat{R})$ on the validation set. The alternative loss functions are discussed in the Supplementary Material. We measure the performance by $\|\hat{S} - S^\star\|_\mathrm{F}$ and $\|\hat{\Delta} - \Delta^\star\|_\mathrm{F}/\sqrt{\sigma_{\max}(R^\star)}$, where the latter is used as a surrogate for the total error distance $TD(\hat{S}, \hat{U})$. Errors are computed on test sets with the same sample size based on 40 independent runs. For our method, the step sizes are set as $\eta_1 = 0.5$, $\eta_2 = \eta_1/\sigma^2_{\max}(U^0)$, where U^0 is the output of the initialization step; the

Table 2. *Simulation results for five algorithms. The estimation errors of the differential network and its sparse component are averaged over* 40 *independent runs, with the standard error given in parentheses. The control group is generated by the covariance* $\Sigma_1^\star$*, while the test i group is generated by* $\Sigma_{i+1}^\star$ *for* $i = 1, 2, 3$

Method	Control – Test 1		Control – Test 2		Control – Test 3	
	$\|\hat{S} - S^\star\|_{\mathrm{F}}$	$\frac{\|\hat{\Delta} - \Delta^\star\|_{\mathrm{F}}}{\sigma_{\max}(R^\star)^{1/2}}$	$\|\hat{S} - S^\star\|_{\mathrm{F}}$	$\frac{\|\hat{\Delta} - \Delta^\star\|_{\mathrm{F}}}{\sigma_{\max}(R^\star)^{1/2}}$	$\|\hat{S} - S^\star\|_{\mathrm{F}}$	$\frac{\|\hat{\Delta} - \Delta^\star\|_{\mathrm{F}}}{\sigma_{\max}(R^\star)^{1/2}}$
			$n = 1000, d = 100, r = 1$			
M*	20.02 (0.56)	10.35 (0.45)	18.73 (0.71)	9.33 (0.53)	18.59 (0.59)	7.94 (0.20)
M1	26.40 (0.67)	11.18 (0.27)	27.58 (0.93)	11.26 (0.37)	30.22 (0.67)	12.07 (0.26)
M2	30.05 (0.32)	12.48 (0.14)	31.04 (0.53)	12.41 (0.21)	32.77 (0.46)	12.75 (0.18)
M3	22.49 (0.35)	9.52 (0.14)	22.54 (0.44)	9.23 (0.17)	22.91 (0.47)	9.18 (0.18)
M4	33.72 (0.61)	14.16 (0.26)	33.62 (0.63)	13.61 (0.26)	34.45 (0.58)	13.62 (0.23)
			$n = 10000, d = 100, r = 2$			
M*	12.55 (0.35)	4.87 (0.13)	11.10 (0.38)	4.52 (0.14)	10.61 (0.38)	4.37 (0.15)
M1	39.50 (0.87)	14.91 (0.33)	50.09 (0.36)	19.49 (0.14)	37.64 (0.56)	14.82 (0.22)
M2	27.86 (0.25)	10.41 (0.09)	32.99 (0.22)	12.77 (0.09)	29.58 (0.22)	11.56 (0.09)
M3	30.54 (0.17)	11.51 (0.06)	34.11 (0.20)	13.27 (0.08)	31.80 (0.14)	12.47 (0.06)
M4	18.88 (0.19)	6.58 (0.07)	17.44 (0.21)	6.44 (0.09)	14.63 (0.30)	5.60 (0.11)
			$n = 200, d = 50, r = 0$			
M*	11.40 (0.41)	11.40 (0.41)	11.73 (0.24)	11.73 (0.24)	9.86 (0.44)	9.86 (0.44)
M1	10.88 (0.33)	10.88 (0.33)	11.92 (0.27)	11.92 (0.27)	10.64 (0.25)	10.64 (0.25)
M2	11.37 (0.54)	11.37 (0.54)	*10.81* (0.40)	10.81 (0.40)	10.37 (0.38)	10.37 (0.38)
M3	11.04 (0.16)	10.85 (0.17)	11.23 (0.18)	10.86 (0.17)	10.48 (0.20)	10.37 (0.21)
M4	13.51 (0.60)	13.51 (0.60)	14.79 (0.65)	14.79 (0.65)	12.71 (0.67)	12.71 (0.67)

M*, the proposed method; M1, ℓ_1-minimization in Zhao et al. (2014); M2, ℓ_1-penalized quadratic loss in Yuan et al. (2017); M3, penalized Gaussian likelihood in Chandrasekaran et al. (2012); M4, constrained Gaussian likelihood in Xu et al. (2017). Detailed descriptions of each method are given in Table 1, and the choice of tuning parameters is discussed in § 5.1.

sparsity proportion $\bar{\alpha}$ $(= \hat{\alpha})$ is chosen from $\{0.01, 0.03, 0.05, 0.1, 0.3, 0.5, 0.8\}$ and $\bar{s}$ $(= \hat{s})$ from $\{2d, 4d, 6d, 15d, 25d, 30d\}$; the rank used in Algorithms 1 and 2 is chosen from $\{0, 1, 2, 3, 4\}$; and the incoherence parameter β is chosen from $\{1, 3\}$. For the methods of Zhao et al. (2014) and Yuan et al. (2017), we use the loss function (5) to choose among five different λ values, which denote tuning parameters in their papers and are generated automatically by their packages. For the method of Chandrasekaran et al. (2012), we use the implementation in Ma et al. (2013), where we greedily choose the tuning parameters $\alpha \in \{0.01, 0.05, 0.1\}$ and $\beta \in \{0.15, 0.25, 0.35\}$, see (2.1) in Ma et al. (2013), while other parameters including the step size, augmented Lagrange multiplier and initialization are kept as in their implementation. For the method of Xu et al. (2017), we select the rank and sparsity in the same way as for our method, while the other parameters are kept as in Xu et al. (2017).

5.2. *Results*

Simulation results are summarized in Table 2. We see that our method outperforms other methods when $r = 2$, corresponding to the case where $\mathrm{rank}(R^\star) = 4$ as $R^\star$ is the difference of two low-rank components. When $r = 1$, Chandrasekaran et al. (2012) is comparable with our method on the first two data generating models, while our method compares favourably in the third case. When $r = 0$, there are no latent variables and our method is comparable to methods of Zhao et al. (2014) and Yuan et al. (2017) that are specifically designed for sparse differential network

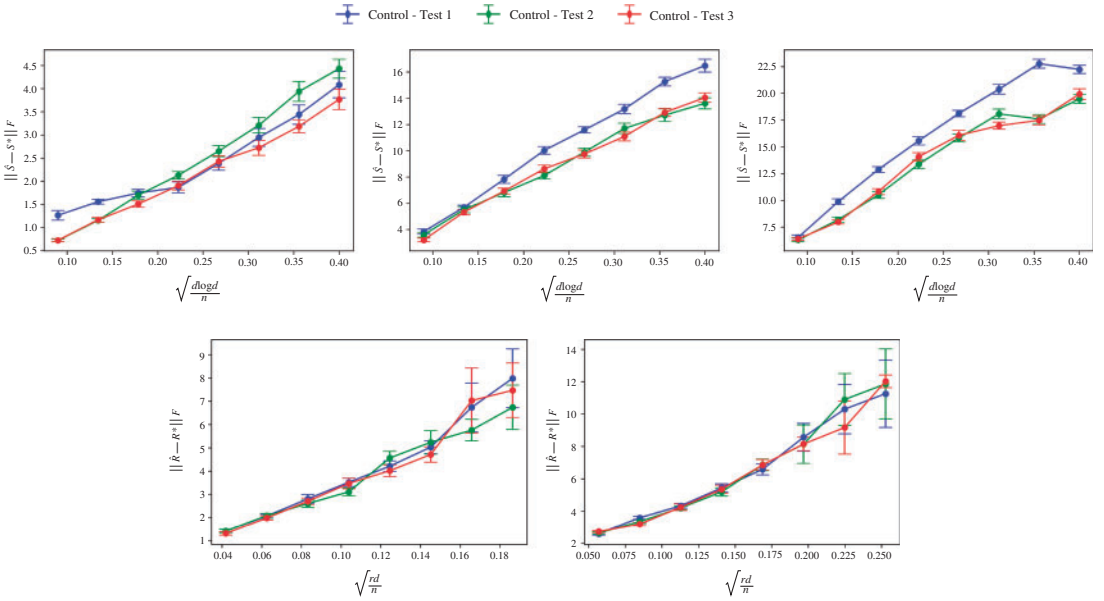

Fig. 1. Statistical rate of convergence. All trends in the figures increase linearly, which validates the results in Theorem 1. Upper: Statistical rate of convergence of estimating $S^\star$. From left to right, $(d, r) = (50, 0), (100, 1), (150, 2)$. Lower: Statistical rate of convergence of estimating $R^\star$. The left panel corresponds to $(d, r) = (100, 1)$, while the right panel corresponds to $(d, r) = (150, 2)$.

estimation without considering latent variables. In comparison, the approach of Chandrasekaran et al. (2012) misestimates the low-rank component. Overall, the proposed nonconvex method accurately estimates both the low-rank and sparse components at a low computational cost. In the Supplementary Material, we show that when the differential network has the sparse plus low-rank structure, while the group specific precision matrices do not have any structure, our method outperforms all the competitors significantly.

The upper and lower rows of Fig. 1 illustrate the statistical rate of convergence by plotting $\|\hat{S} - S^\star\|_{\mathrm{F}}$ versus $(d \log d/n)^{1/2}$ and $\|\hat{R} - R^\star\|_{\mathrm{F}}$ versus $(rd/n)^{1/2}$, respectively. We set $(d, r) = (50, 0), (100, 1), (150, 2)$ for each case and vary n only. Although the estimation errors for $S^\star$ and $R^\star$ are combined in Theorem 1, we expect a linear increasing trend in both figures since $d \log d/n \asymp rd/n$. In the Supplementary Material, we illustrate that the rank and the positive index of inertia are consistently selected by cross-validation.

6. APPLICATION TO FMRI FUNCTIONAL CONNECTIVITY

We apply our method to the task of estimating differential brain functional connectivity from functional magnetic resonance imaging, fMRI. In particular, we analyse the Center for Biomedical Research Excellence, COBRE dataset, which is publicly available in the `nilearn` Python package (Abraham et al., 2014). This dataset includes fMRI data from 146 subjects across two groups: 74 subjects are healthy controls and 72 subjects are diagnosed with schizophrenia. Each subject data includes resting-state fMRI time series with 150 samples. We remove time-points with excessive motion as recommended by standard analyses, and apply the Harvard–Oxford Atlas to automatically generate 48 regions of interests. This dataset has been carefully analysed using the NeuroImaging Analysis Kit, `https://github.com/SIMEXP/niak`.

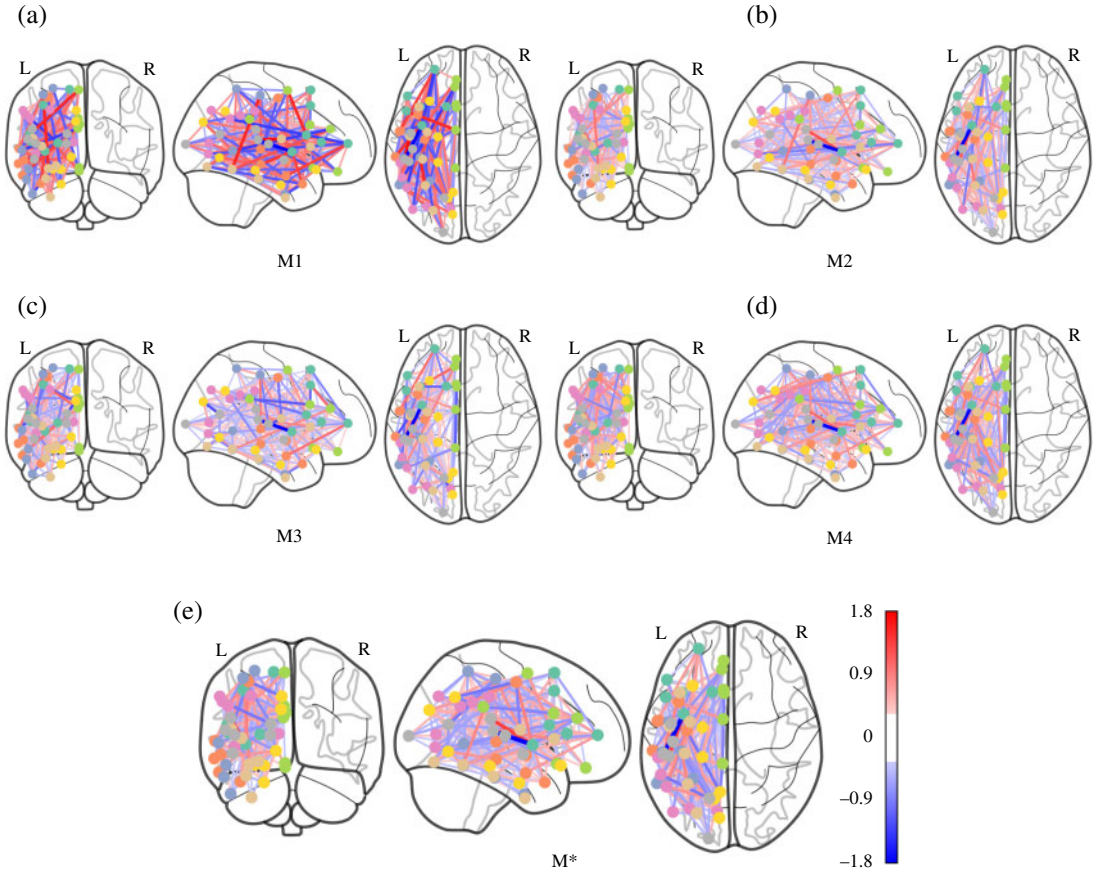

Fig. 2. Glass brains for the estimated sparse component of the differential network.

We first estimate the differential network between the schizophrenia and control groups. We collect all the fMRI series in one group across all subjects, thus assuming individuals in the same group share the same brain functional connectivity. Equivalently, we simply stack all time series from the subjects together to obtain one dataset for each group. The proposed approach and all the baseline methods remain the same as described in § 5. The sparse plus low-rank decomposition is reasonable for estimating the differential network as it considers the most pressing demographic confounders such as age and gender, and the number of confounders is assumed to be small compared with the number of nodes in a brain, which is a conventional set-up in fMRI study (Greve et al., 2013; Geng et al., 2019). The sparse component of the differential network is the parameter of scientific interest.

The estimated sparse component is reported in Fig. 2, where each region corresponds to a vertex, each edge corresponds to an entry of the precision matrix, and the colour corresponds to the magnitude of the entry. Since the Harvard–Oxford Atlas is a three-dimensional parcellation atlas with lateralized labels, we show the detected connectomes in the left hemisphere only. We see that the Zhao et al. (2014) approach fails to recover a clear pattern; Chandrasekaran et al. (2012) recovers one negative edge in the *Central Opercular Cortex* and one negative edge in the *Middle Frontal Gyrus*; our method, together with the methods in Yuan et al. (2017) and Xu et al. (2017), show that the sparse network has two obvious edges, one positive and one negative, in the *Central Opercular Cortex* area, which is also consistent with some recent analysis that also discovered

that the *Central Opercular Cortex* is one of the regions that differs the most for schizophrenia (Sheffield et al., 2015; Geng et al., 2019). Upon closer analysis, we find that the network estimated by the proposed method has smaller quadratic loss (5) on the test sets. We let $\mathcal{L}_n(\hat{\Delta})$ denote the empirical test loss, where $\hat{\Delta}$ is the estimator and the covariance matrices are calculated on the test set. Our estimator has the test loss $\mathcal{L}_n(\hat{\Delta}) = -3.64$, with $\|\nabla\mathcal{L}_n(\hat{\Delta})\|_{\infty,\infty} = 0.11$ and $\|\nabla\mathcal{L}_n(\hat{\Delta})\|_{\mathrm{F}} = 1.56$. All three quantities are smaller than the other methods, though they are not designed for minimizing (5).

To further quantitatively validate our claims, we consider an individual-level analysis. In particular, we select 10 subjects from each group and consider the 190 possible pairs among them; 100 out of 190 pairs are across-group while the remaining 90 pairs are within-group. We estimate the differential network for each pair and calculate $\|\hat{S}\|_{\mathrm{F}}$. Based on the group differences, one expects the sparse differential network for within-group pairs to have smaller norms than for across-group pairs. Applying an unpaired two-sample t test, the p-value for the proposed method is 0.09 while it is greater than 0.16 for M1 to M4. This further validates that our method also outperforms other methods at the individual level.

7. DISCUSSION

Extending our approach to identify the difference in the complete connectivity of the graph, which includes latent variables, is of additional interest. Vinyes & Obozinski (2018) studied the problem of identification and estimation of the complete connectivity of the graph in the presence of latent variables using a carefully designed convex penalty. In the limit of an infinite amount of data, under suitable assumptions, their procedure is able to identify the complete graph structure. However, the finite-sample properties of this procedure are not known. High-dimensional settings present several challenges. First, additional assumptions are needed on the low-rank component, $R^\star$, such as a sparsity assumption on the effect of latent variables on observed variables. However, such an assumption makes identification of parameters more difficult, since we need to be able to distinguish the low-rank component from the sparse component. This is an identification problem and is a challenge for both convex and nonconvex approaches. Second, when estimating a differential network, the matrix $R^\star$ is indefinite and, as a result, development of a new penalty is required. Third, the initialization step in our algorithm requires us to compute the inverse of the sample covariance matrix, which is rank deficient in a high-dimensional setting. Therefore, a suitable and computationally efficient initialization strategy needs to be developed in a high-dimensional setting. Finally, the gradient error $\|\nabla_S\bar{\mathcal{L}}_n(S^\star, U^k, \Lambda^\star) - \nabla_S\bar{\mathcal{L}}_n(S^\star, U^\star, \Lambda^\star)\|_{\mathrm{F}}$, which is a key ingredient in the proof, is well controlled only if $\hat{\Sigma}_X U^\star$, and $\hat{\Sigma}_Y U^\star$, satisfies the incoherence condition. Since the incoherence condition is imposed on $U^\star$, we need the sample size to satisfy $(n_X \wedge n_Y) \gtrsim d \log d$. One possible approach to developing a nonconvex estimation procedure for high-dimensional differential network estimation could be based on a thresholding step for the low-rank component (Yu et al., 2019).

Recent work on differential networks has focused on statistical inference, including developing statistical tests for the global null $H_0 : \Delta^\star = 0$ (Xia et al., 2015; Cai et al., 2019) and development of confidence intervals for elements of the differential network (Kim et al., 2019). The regression approach of Ren et al. (2015) can be used to construct asymptotically normal estimators of the elements of the differential network in the presence of latent variables. Such an approach would require both the individual precision matrices to be sparse and the correlation between latent and observed variables to be weak. How to develop an inference procedure that requires only weak conditions on the differential network remains an open problem.

In our simulation and real data application, we propose choosing the tuning parameters using cross-validation. Zhao et al. (2014) proposed tuning the parameters by optimizing the approximate Akaike information criterion in the context of sparse differential network estimation; however, there are no theoretical guarantees associated with the chosen parameters. Extending the ideas of Foygel & Drton (2010) in the context of sparse plus low-rank estimation, and showing that the Akaike or Bayesian information criterion can be used for consistent recovery is of both practical and theoretical interest, as it would allow for faster parameter tuning compared to cross-validation.

Acknowledgement

We are grateful to the editor, associate editor and two referees for their insightful comments, which have led to significant improvement of our paper. We thank Huili Yuan and Ruibin Xi for sharing their R code. This work was partially supported by the William S. Fishman Faculty Research Fund at the University of Chicago Booth School of Business. It was completed in part with resources supported by the University of Chicago Research Computing Center. Koyejo is also affiliated with the Beckman Institute for Advanced Science and Technology at the University of Illinois at Urbana-Champaign.

Supplementary material

Supplementary material available at *Biometrika* online includes extended proofs, additional simulations and discussions on potentially applicable convex approaches. Our code is available at `https://github.com/senna1128/Differential-Network-Estimation-via-Nonconvex-Approach`.

References

Abraham, A., Pedregosa, F., Eickenberg, M., Gervais, P., Mueller, A., Kossaifi, J., Gramfort, A., Thirion, B. & Varoquaux, G. (2014). Machine learning for neuroimaging with scikit-learn. *Front. Neuroinform.* **8**, DOI:10.3389/fninf.2014.00014.

Barabási, A.-L. & Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Rev. Genet.* **5**, 101–13.

Bielza, C. & Larrañaga, P. (2014). Bayesian networks in neuroscience: A survey. *Front. Comput. Neurosci.* **8**, 131.

Bullmore, E. & Sporns, O. (2009). Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Rev. Neurosci.* **10**, 186–98.

Cai, T. T., Li, H., Ma, J. & Xia, Y. (2019). Differential Markov random field analysis with an application to detecting differential microbial community networks. *Biometrika* **106**, 401–16.

Candès, E. J., Li, X., Ma, Y. & Wright, J. (2011). Robust principal component analysis? *J. Assoc. Comp. Mach.* **58**, 11.

Candès, E. J. & Romberg, J. (2007). Sparsity and incoherence in compressive sampling. *Inverse Problems* **23**, 969–85.

Castellanos, F. X., Di Martino, A., Craddock, R. C., Mehta, A. D. & Milham, M. P. (2013). Clinical applications of the functional connectome. *NeuroImage* **80**, 527–40.

Chandrasekaran, V., Parrilo, P. A. & Willsky, A. S. (2012). Latent variable graphical model selection via convex optimization. *Ann. Statist.* **40**, 1935–67.

Chandrasekaran, V., Sanghavi, S., Parrilo, P. A. & Willsky, A. S. (2011). Rank-sparsity incoherence for matrix decomposition. *SIAM J. Optimiz.* **21**, 572–96.

Chen, J., Zhou, J. & Ye, J. (2011). Integrating low-rank and group-sparse structures for robust multi-task learning. In *Proc. Int. Conf. Knowledge Discovery and Data Mining*, pp. 42–50.

Chen, Y. (2015). Incoherence-optimal matrix completion. *IEEE Trans. Info. Theory* **61**, 2909–23.

Chen, Y., Bhojanapalli, S., Sanghavi, S. & Ward, R. (2014). Coherent matrix completion. *Proc. Mach. Learn. Res.* **32**, 674–82.

de la Fuente, A. (2010). From 'differential expression' to 'differential networking': Identification of dysfunctional regulatory networks in diseases. *Trends Genet.* **26**, 326–33.

Drton, M. & Maathuis, M. H. (2017). Structure learning in graphical modeling. *Ann. Rev. Statist. Appl.* **4**, 365–93.

Durkee, T. , Kaess, M., Carli, V., Parzer, P., Wasserman, C., Floderus, B., Apter, A., Balazs, J., Barzilay, S., Bobes, J. et al. (2012). Prevalence of pathological internet use among adolescents in Europe: Demographic and social factors. *Addiction* **107**, 2210–22.

FAZEL, M., CANDÈS, E. J., RECHT, B. & PARRILO, P. (2008). Compressed sensing and robust recovery of low rank matrices. In *Proc. 42nd Asilomar Conf. Signals, Systems and Computers*. New York: IEEE.

FOYGEL, R. & DRTON, M. (2010). Extended Bayesian information criteria for Gaussian graphical models. In *Advances in Neural Information Processing Systems 23*. J. D. Lafferty, C. K. I.Williams, J. Shawe-Taylor, R. S. Zemel and A. Culotta, eds, pp. 604–12. Red Hook, NY: Curran Associates, Inc.

FRIEDMAN, N. (2004). Inferring cellular networks using probabilistic graphical models. *Science* **303**, 799–805.

GAGGIOTTI, O. E., BEKKEVOLD, D., JØRGENSEN, H. B., FOLL, M., CARVALHO, G. R., ANDRE, C. & RUZZANTE, D. E. (2009). Disentangling the effects of evolutionary, demographic, and environmental factors influencing genetic structure of natural populations: Atlantic herring as a case study. *Evolution* **63**, 2939–51.

GENG, S., YAN, M., KOLAR, M. & KOYEJO, S. (2019). Partially linear additive Gaussian graphical models. *Proc. Mach. Learn. Res.* **97**, 2180–90.

GREVE, D. N. , BROWN, G. G., MUELLER, B. A., GLOVER, G., LIU, T. T. (2013). A survey of the sources of noise in fMRI. *Psychometrika* **78**, 396–416.

HORN, R. A. & JOHNSON, C. R. (2013). *Matrix Analysis*. Cambridge: Cambridge University Press, 2nd ed.

HUDSON, N. J., REVERTER, A. & DALRYMPLE, B. P. (2009). A differential wiring analysis of expression data correctly identifies the gene containing the causal mutation. *PLoS Comput. Biol.* **5**, e1000382.

IDEKER, T. & KROGAN, N. J. (2012). Differential network biology. *Molec. Sys. Biol.* **8**, 565.

KIM, B., LIU, S. & KOLAR, M. (2019). Two-sample inference for high-dimensional Markov networks. *arXiv:* 1905.00466.

LAURITZEN, S. L. (1996). *Graphical Models*. New York: Oxford University Press.

LAURITZEN, S. L. & SHEEHAN, N. A. (2003). Graphical models for genetic analyses. *Statist. Sci.* **18**, 489–514.

LIU, S., QUINN, J. A., GUTMANN, M. U., SUZUKI, T. & SUGIYAMA, M. (2014). Direct learning of sparse changes in Markov networks by density ratio estimation. *Neural Comput.* **26**, 1169–97.

MA, S., XUE, L. & ZOU, H. (2013). Alternating direction methods for latent variable Gaussian graphical model selection. *Neural Comput.* **25**, 2172–98.

MENG, Z., ERIKSSON, B. & HERO III, A. O. (2014). Learning latent variable Gaussian graphical models. *Proc. Mach. Learn. Res.* **32**, 1269–77.

PAZ, D. J. & SÁNCHEZ, A. G. (2015). Improving the precision matrix for precision cosmology. *Mon. Not. R. Astron. Soc.* **454**, 4326–34.

REN, Z., SUN, T., ZHANG, C.-H. & ZHOU, H. H. (2015). Asymptotic normality and optimalities in estimation of large Gaussian graphical models. *Ann. Statist.* **43**, 991–1026.

SHEFFIELD, J. M., REPOVS, G., HARMS, M. P., CARTER, C. S., GOLD, J. M., MACDONALD III, A. W., RAGLAND, J. D., SILVERSTEIN, S. M., GODWIN, D. & BARCH, D. M. (2015). Fronto-parietal and cingulo-opercular network integrity and cognition in health and schizophrenia. *Neuropsychologia* **73**, 82–93.

SHOJAIE, A. (2020). Differential network analysis: A statistical perspective. *WIREs Comput. Statist.* **2020**, e1508.

SMITH, S. M., MILLER, K. L., SALIMI-KHORSHIDI, G., WEBSTER, M., BECKMANN, C. F., NICHOLS, T. E., RAMSEY, J. D. & WOOLRICH, M. W. (2011). Network modelling methods for fMRI. *NeuroImage* **54**, 875–91.

VINYES, M. & OBOZINSKI, G. (2018). Learning the effect of latent variables in Gaussian graphical models with unobserved variables. *arXiv:*1807.07754v2.

WILLI, Y. & HOFFMANN, A. A. (2009). Demographic factors and genetic variation influence population persistence under environmental change. *J. Evolut. Biol.* **22**, 124–33.

XIA, Y., CAI, T. & CAI, T. T. (2015). Testing differential networks with applications to the detection of gene-gene interactions. *Biometrika* **102**, 247–66.

XU, P. & GU, Q. (2016). Semiparametric differential graph models. In *Advances in Neural Information Processing Systems 29*. Red Hook, NY: Curran Associates, Inc., pp. 1064–72.

XU, P., MA, J. & GU, Q. (2017). Speeding up latent variable Gaussian graphical model estimation via nonconvex optimization. In *Advances in Neural Information Processing Systems 30*. Red Hook, NY: Curran Associates, Inc., pp. 1930–41.

YI, X., PARK, D., CHEN, Y. & CARAMANIS, C. (2016). Fast algorithms for robust PCA via gradient descent. In *Advances in Neural Information Processing Systems 29*. Red Hook, NY: Curran Associates, Inc., pp. 4152–60.

YU, M., GUPTA, V. & KOLAR, M. (2019). Recovery of simultaneous low rank and two-way sparse coefficient matrices, a nonconvex approach. *arxiv:*1802.06967v2.

YUAN, H., XI, R., CHEN, C. & DENG, M. (2017). Differential network analysis via lasso penalized D-trace loss. *Biometrika* **104**, 755–70.

ZHANG, X., WANG, L. & GU, Q. (2018). A unified framework for nonconvex low-rank plus sparse matrix recovery. *Proc. Mach. Learn. Res.* **84**, 1097–107.

ZHAO, S. D., CAI, T. T. & LI, H. (2014). Direct estimation of differential networks. *Biometrika* **101**, 253–68.

[*Received on* 12 *September* 2019. *Editorial decision on* 13 *May* 2020]