Sparse-Group Lasso for Graph Learning From Multi-Attribute Data

Jitendra K. Tugnait, *Life Fellow, IEEE*

***Abstract*—We consider the problem of inferring the conditional independence graph (CIG) of high-dimensional Gaussian vectors from multi-attribute data. Most existing methods for graph estimation are based on single-attribute models where one associates a scalar random variable with each node. In multi-attribute graphical models, each node represents a random vector. In this paper, we present a sparse-group lasso based penalized log-likelihood approach for graph learning from multi-attribute data. Existing works on multi-attribute graphical modeling have considered only group lasso penalty. The main objective of this paper is to explore the use of sparse-group lasso for multi-attribute graph estimation. An alternating direction method of multipliers (ADMM) algorithm is presented to optimize the objective function to estimate the inverse covariance matrix. Sufficient conditions for consistency and sparsistency of the estimator are provided. Numerical results based on synthetic as well as real data are presented.**

***Index Terms*—Graph learning, inverse covariance estimation, undirected graph, sparse-group lasso, multi-attribute data.**

I. Introduction

GRAPHICAL models provide a powerful tool for analyzing multivariate data [1], [2]. A central concept is that of conditional independence. In graphical models, graphs display the conditional independence structure of the variables, and learning the graph structure is equivalent to learning a factorization of the joint probability distribution of these random variables. In an undirected graphical model, the conditional dependency structure among p random variables $x_1, x_1, \cdots, x_p$, ($\boldsymbol{x} = [x_1\ x_2\ \cdots\ x_p]^\top$), is represented using an undirected graph $\mathcal{G} = (V, \mathcal{E})$, where $V = \{1, 2, \cdots, p\} = [1, p]$ is the set of p nodes corresponding to the p random variables x_is, and $\mathcal{E} \subseteq V \times V$ is the set of undirected edges describing conditional dependencies among the components of $\boldsymbol{x}$. The graph $\mathcal{G}$ then is a conditional independence graph (CIG) where there is no edge between nodes i and j (i.e., $\{i, j\} \notin \mathcal{E}$) iff x_i and x_j are conditionally independent given the remaining $p-2$ variable [2, p. 60]. Graphical models based on gene expression data have been used for data visualization and biological hypothesis generation (i.e., exploratory data analysis) in [3]. Some other applications include classification and exploratory data analysis in intensive care monitoring [4], financial time series [5], [6], social networks [7], gene regulatory networks [8], [9], and analysis of fMRI (functional magnetic resonance imaging) data [10].

Gaussian graphical models (GGMs) are CIGs where $\boldsymbol{x}$ is multivariate Gaussian. Suppose $\boldsymbol{x}$ has positive-definite covariance matrix $\boldsymbol{\Sigma}$ with inverse covariance matrix $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$. Then Ω_{ij}, the (i, j)-th element of $\boldsymbol{\Omega}$, is zero iff x_i and x_j are conditionally independent. Given n samples of $\boldsymbol{x}$, in high-dimensional (data-starved) settings, one estimates $\boldsymbol{\Omega}$ under some sparsity constraints; see [3], [11], [12], [13], [14]. In these graphs each node represents a scalar random variable; we will call such a graph $\mathcal{G}$ a *single-attribute graphical model* for $\boldsymbol{x}$. In many applications, there may be more than one random variable associated with a node. This class of graphical models has been called *multi-attribute graphical models* in [8], [9], [15] where a focus is on application to biological regulatory networks. In [16], [17] image graphs for grayscale texture images are inferred for modeling dependence of a pixel on neighboring pixels; here one has one variable per pixel node. These approaches do not apply to color images where one has three variables (RGB — red, green, blue — color components) per pixel node. Image graphs for color images is an example of multi-attribute graphical models.

In this paper we consider p random vectors $\boldsymbol{z}_i \in \mathbb{R}^m$, $i = 1, 2, \cdots, p$, $m \geq 2$. We associate $\boldsymbol{z}_i$ with the ith node of an undirected graph $\mathcal{G} = (V, \mathcal{E})$ where $V = [1, p]$, and $\mathcal{E} \subseteq V \times V$ is the set of edges that describe the conditional dependencies among vectors $\{\boldsymbol{z}_i,\ i \in V\}$. As in the scalar case ($m = 1$), there is no edge between node i and node j in $\mathcal{G}$ iff random vectors $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are conditionally independent given the remaining $p-2$ vectors.

We now consider some specific cases where use of multi-attribute models is relevant. Gene regulatory networks have been considered in [8], [9], [15], [18], [19]. Antibodies are proteins, and proteins are encoded by genes. Protein profiles and gene profiles have been used in [8], [9], [15], [18], [19] for graph modeling to investigate links between various proteins/genes based on their two profiles using data from the US national cancer institute NCI-60 database for 60 human tumor cell lines. In [15] a network with 91 nodes (p = 91 genes), each with two (m=2) attributes comprised of protein and gene profiles, is considered based on n =60 samples (cell lines). Since these molecular profiles are on the same set of biological samples, it is argued in [15] (also by others) that the multi-attribute graphical model "proposes a consensus version of the interactions at hand in the

Manuscript received September 2, 2020; revised December 27, 2020 and January 19, 2021; accepted February 3, 2021. Date of publication February 8, 2021; date of current version March 30, 2021. The associate editor coordinating the review of this paper and approving it for publication was Dr. Justin Dauwels. This work was supported by National Science Foundation under Grants CCF-1617610 and ECCS-2040536.

The author is with the Department of Electrical & Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: tugnajk@eng.auburn.edu).

Digital Object Identifier 10.1109/TSP.2021.3057699

cell, and one which is hopefully more robust to noise," compared with single-attribute models based on protein profile alone or gene profile alone. Similar conclusions are reached in [8], [9], [18], [19]. In [16], [17] image graphs for grayscale images are inferred for modeling dependence of a pixel on neighboring pixels; here one has one variable per pixel node. These approaches do not apply to color images where one has three variables (RGB color components) per pixel node. Image graphs for color images is an example of multi-attribute graphical models with a pixel represented by a graph node and three attributes (RGB components) per pixel. Graph-based transform coding and its potential application to signal/image compression is discussed in [20] which is a potential application of image graphs; see also [21], [22]. These contributions are restricted to single attribute models. Connections among different industries in the US are explored in [18] to see if the GDP (gross domestic product) of one industry has some effect on that of other industries. Regional GDP data available from U.S. Department of Commerce website for 8 regions (New England, Mideast, Great Lakes, etc.) and 20 industries (utilities, construction, manufacturing, etc.) are used resulting in a multi-attribute graph with $p = 20$ nodes (industries) and $m = 8$ attributes (regions). Exploration in [18] takes regions into consideration, since significant differences in relations may exist because of regional characteristics, which are not possible to capture using only national data. In another application in [23], daily time series data from Hong Kong to analyze air pollution of Hong Kong via single attribute graphical models is considered, following the earlier work of [24] based on dependent time series. The time series data of the daily average for four pollutants over three monitoring stations are used in [23] to construct a time series graph of $p = 12$ nodes with $m = 1$ attribute. If the objective is to study conditional dependencies between various pollutants, then a more appropriate model would be a multi-attribute model with $p = 4$ pollutant nodes, each with $m = 3$ attributes, reflecting measurements at the three monitoring stations.

A. Related Work

For high-dimensional linear regression problems with grouped covariates, it has been shown in [25], [26] that imposing an additional within group level sparsity constraint can lead to improved classification performance. This is the sparse-group lasso approach. These papers are concerned with algorithm development and do not offer any theoretical analysis, and also do not consider graphical models, single or multi-attribute. Multi-attribute graphical model learning has been considered in [8], [9], [15] and [18]. In [8], [9] a group lasso based penalized log-likelihood approach is investigated. A primal-dual optimization algorithm is given and a theoretical analysis of true graph recovery with high probability is provided following the single-attribute results of [27]. In [15], [18] group lasso based penalized pseudo-likelihood approaches are considered for multi-attribute graph estimation. While [15] offers no theoretical analysis, in [18] sufficient conditions for convergence in the Frobenius norm of the inverse covariance estimator to the true value are presented following the single-attribute results of [28]. Sparse-group lasso penalty has also been used in [3] for joint graphical lasso in the context of multiple classes. As noted in [8], [9], the approach of [3] can be used for multi-attribute graphical model learning. An alternating direction method of multipliers (ADMM) algorithm is presented in [3], but there is no theoretical analysis regarding graph estimator.

B. Our Contributions

In this paper, we present a sparse-group lasso based penalized log-likelihood approach for graph learning from multi-attribute data, whereas [8], [9], [15], [18] consider only group lasso which is a special case of sparse-group lasso. Our penalty is similar, but not identical, to the group graphical lasso penalty in [3]. In [3], correlations between data from different classes are ignored, whereas correlations between data from different attributes are central to our approach. In group lasso, sparsity penalty is imposed on all entries of $\mathbf{\Omega}$ associated with a pair of nodes, as a group. In sparse-group lasso, an additional sparsity penalty is imposed on each off-diagonal Ω_{ij}. An alternating direction method of multipliers (ADMM) algorithm is presented to optimize the objective function to estimate the inverse covariance matrix. We provide sufficient conditions for convergence in the Frobenius norm of the estimator to the true value, a rate of convergence, and also consider sparsistency; [8], [9] provide conditions only for consistent graph edge recovery, but not for consistent inverse covariance matrix estimation, and [15] offers no theoretical analysis. Related works of [25], [26] dealing with sparse-group lasso do not offer any theoretical analysis (such as our Theorems 1 and 2), and do not consider graphical models.

Our theoretical results follow the single-attribute method of [29] for consistency and the method of [30] for sparsistency, resulting in much simpler, and checkable, sufficient conditions, whereas [8], [9] require an "irrepresentable condition" ([9, condition (12)]) which is hard to verify. The sufficient conditions for convergence given in [18] also require an "irrepresentable (incoherence) condition" ([18, condition (C2)], [28, condition (C2)]) which is hard to verify. We require no such conditions.

We test the proposed approach on synthetic as well as real data. While the ground truth is unknown in the real data applications, requiring domain expert knowledge to interpret the results, the synthetic data examples clearly demonstrate the advantages of using sparse-group lasso instead of just group-lasso or just lasso.

C. Outline and Notation

The rest of the paper is organized as follows. The system model is presented in Sec. II where we describe the multi-attribute graphical model with m random variables per node, and also an associated larger single-attribute graph. A sparse-group lasso based penalized log-likelihood approach for graph learning from multi-attribute data is proposed in Sec. III. An ADMM algorithm is presented in Sec. III-A to optimize the objective function to estimate the inverse covariance matrix and the edges in the graph. In Sec. IV we analyze consistency (Theorem 1) and sparsistency (Theorem 2) of the proposed approach. Numerical results based on synthetic as well as real data are presented in

Sec. V to illustrate the proposed approach. Proofs of Theorems 1 and 2 are given in Appendices A and B, respectively.

Given $\boldsymbol{A} \in \mathbb{R}^{p\times p}$, we use $\phi_{\min}(\boldsymbol{A}), \phi_{\max}(\boldsymbol{A}), |\mathbf{A}|, \text{tr}(\boldsymbol{A})$ and $\text{etr}(\boldsymbol{A})$ to denote the minimum eigenvalue, maximum eigenvalue, determinant, trace, and exponential of trace of $\boldsymbol{A}$, respectively. For a matrix $\boldsymbol{B} \in \mathbb{R}^{p\times q}$, we define the operator norm, the Frobenius norm and the vectorized ℓ_1 norm, respectively, as $\|\boldsymbol{B}\| = \sqrt{\phi_{\max}(\boldsymbol{B}^\top \boldsymbol{B})}$, $\|\boldsymbol{B}\|_F = \sqrt{\text{tr}(\boldsymbol{B}^\top \boldsymbol{B})}$ and $\|\boldsymbol{B}\|_1 = \sum_{i,j} |B_{ij}|$ where B_{ij} is the (i,j)-th element of $\boldsymbol{B}$. We also denote B_{ij} by $[\boldsymbol{B}]_{ij}$. Given $\boldsymbol{A} \in \mathbb{R}^{p\times p}$, $\boldsymbol{A}^+ = \text{diag}(\boldsymbol{A})$ is a diagonal matrix with the same diagonal as $\boldsymbol{A}$, and $\boldsymbol{A}^- = \boldsymbol{A} - \boldsymbol{A}^+$ is $\boldsymbol{A}$ with all its diagonal elements set to zero. Symbol $\otimes$ denotes the matrix Kronecker product and $\mathbf{1}_A$ is the indicator function, equaling one if A is true, zero otherwise. For $\boldsymbol{y}_n, \boldsymbol{x}_n \in \mathbb{R}^p$, $\boldsymbol{y}_n \asymp \boldsymbol{x}_n$ means that $\boldsymbol{y}_n = \mathcal{O}(\boldsymbol{x}_n)$ and $\boldsymbol{x}_n = \mathcal{O}(\boldsymbol{y}_n)$, where the latter means there exists $0 < M < \infty$ such that $\|\boldsymbol{x}_n\| \le M\|\boldsymbol{y}_n\|\ \forall n \ge 1$. The notation $\boldsymbol{y}_n = \mathcal{O}_P(\boldsymbol{x}_n)$ for random vectors $\boldsymbol{y}_n, \boldsymbol{x}_n \in \mathbb{R}^p$ means that for any $\varepsilon > 0$, there exists $0 < M < \infty$ such that $P(\|\boldsymbol{y}_n\| \le M\|\boldsymbol{x}_n\|) \ge 1 - \varepsilon\ \forall n \ge 1$.

II. System Model

Consider p jointly Gaussian random vectors $\boldsymbol{z}_i \in \mathbb{R}^m$, $i = 1, 2, \cdots, p$. We associate $\boldsymbol{z}_i$ with the ith node of an undirected graph $\mathcal{G} = (V, \mathcal{E})$ where $V = [1, p]$ is the set of p nodes, and $\mathcal{E} \subseteq V \times V$ is the set of edges that describe the conditional dependencies among vectors $\{\boldsymbol{z}_i,\ i \in V\}$. As in the scalar case ($m = 1$), there is no edge between node i and node j in $\mathcal{G}$ (i.e., $\{i, j\} \notin \mathcal{E}$) iff random vectors $\boldsymbol{z}_i$ and $\boldsymbol{z}_j$ are conditionally independent given all the remaining random vectors $\boldsymbol{z}_\ell$ corresponding to the remaining $p - 2$ nodes in V, i.e., for $\ell \in V\backslash\{j, k\}$ [8], [9]. This is the multi-attribute Gaussian graphical model of interest in this paper. The term multi-attribute Gaussian graphical model has been used in [15] for such models. Define the mp-vector

$$\boldsymbol{x} = [\boldsymbol{z}_1^\top\ \boldsymbol{z}_2^\top\ \cdots\ \boldsymbol{z}_p^\top]^\top \in \mathbb{R}^{mp}. \tag{1}$$

Suppose we have n i.i.d. observations $\boldsymbol{x}(t), t = 0, 1, \cdots, n-1$, of zero-mean $\boldsymbol{x}$. Our objective is to estimate the inverse covariance matrix $(\mathbb{E}\{\mathbf{x}\mathbf{x}^\top\})^{-1}$ and to determine if edge $\{i, j\} \in \mathcal{E}$, given data $\{\boldsymbol{x}(t)\}_{t=0}^{n-1}$.

Let us associate $\boldsymbol{x}$ with an "enlarged" graph $\bar{\mathcal{G}} = (\bar{V}, \bar{\mathcal{E}})$, where $\bar{V} = [1, mp]$ and $\bar{\mathcal{E}} \subseteq \bar{V} \times \bar{V}$. Now $[\boldsymbol{z}_j]_\ell$, the ℓth component of $\boldsymbol{z}_j$ associated with node j of $\mathcal{G} = (V, \mathcal{E})$, is the random variable $x_q = [\boldsymbol{x}]_q$, where $q = (j-1)m + \ell$, $j = 1, 2, \cdots, p$ and $\ell = 1, 2, \cdots, m$. The random variable x_q is associated with node q of $\bar{\mathcal{G}} = (\bar{V}, \bar{\mathcal{E}})$. Corresponding to the edge $\{j, k\} \in \mathcal{E}$ in the multi-attribute $\mathcal{G} = (V, \mathcal{E})$, there are m^2 edges $\{q, r\} \in \bar{\mathcal{E}}$ specified by $q = (j-1)m + s$ and $r = (k-1)m + t$, where $s = 1, 2, \cdots, m$ and $t = 1, 2, \cdots, m$. The graph $\bar{\mathcal{G}} = (\bar{V}, \bar{\mathcal{E}})$ is a single-attribute graph. In order for $\bar{\mathcal{G}}$ to reflect the conditional independencies encoded in $\mathcal{G}$, we must have the equivalence

$$\{j, k\} \notin \mathcal{E} \Leftrightarrow\ \bar{\mathcal{E}}^{(jk)} \cap \bar{\mathcal{E}} = \emptyset, \tag{2}$$

where

$$\bar{\mathcal{E}}^{(jk)} = \big\{\{q, r\}\,\big|\, q = (j-1)m + s,\ r = (k-1)m + t,\ s, t = 1, 2, \cdots, m\big\}. \tag{3}$$

Let $\boldsymbol{R}_{xx} = \mathbb{E}\{\boldsymbol{x}\boldsymbol{x}^\top\} \succ \mathbf{0}$ and $\boldsymbol{\Omega} = \boldsymbol{R}_{xx}^{-1}$. Define the (j,k)th $m \times m$ subblock $\boldsymbol{\Omega}^{(jk)}$ of $\boldsymbol{\Omega}$ as

$$[\boldsymbol{\Omega}^{(jk)}]_{st} = [\boldsymbol{\Omega}]_{(j-1)m+s,(k-1)m+t},\ s, t = 1, 2, \cdots, m. \tag{4}$$

It is established in [9, Sec. 2.1] that

$$\boldsymbol{\Omega}^{(jk)} = \mathbf{0} \Leftrightarrow\ \boldsymbol{z}_j \text{ and } \boldsymbol{z}_k \text{ are conditionally independent} \Leftrightarrow\ \{j, k\} \notin \mathcal{E}. \tag{5}$$

Since $\boldsymbol{\Omega}^{(jk)} = \mathbf{0}$ is equivalent to $[\boldsymbol{\Omega}]_{qr} = 0$ for every $\{q, r\} \in \bar{\mathcal{E}}^{(jk)}$, and since, by [1, Proposition 5.2], $[\boldsymbol{\Omega}]_{qr} = 0$ iff x_q and x_r are conditionally independent, hence, iff $\{q, r\} \notin \bar{\mathcal{E}}$, it follows that equivalence (2) holds true.

III. Sparse-Group Graphical Lasso

Given n samples $\{\boldsymbol{x}(t)\}_{t=0}^{n-1}$ of zero-mean $\boldsymbol{x}$, define the sample covariance $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{t=0}^{n-1} \boldsymbol{x}(t)\boldsymbol{x}^\top(t)$. Let $\boldsymbol{X} = [\boldsymbol{x}(0)\, \boldsymbol{x}(1)\, \cdots\, \boldsymbol{x}(n-1)\,]^\top \in \mathbb{R}^{n\times(mp)}$. We have the log-likelihood (up to some constants)

$$\ln f_{\boldsymbol{X}}(\boldsymbol{X}) = \ln(|\boldsymbol{\Omega}|) - \text{tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Omega}). \tag{6}$$

To estimate sparse $\boldsymbol{\Omega}$, consider minimization of a penalized version of the negative log-likelihood

$$L(\boldsymbol{X}; \boldsymbol{\Omega}) = -\ln f_{\boldsymbol{X}}(\boldsymbol{X}) + P(\boldsymbol{\Omega}) \tag{7}$$

using a sparse-group lasso penalty [25], where

$$P(\boldsymbol{\Omega}) = \alpha\lambda\,\|\boldsymbol{\Omega}^-\|_1 + (1-\alpha)\lambda \sum_{j\ne k}^{p} \|\boldsymbol{\Omega}^{(jk)}\|_F, \tag{8}$$

$\lambda > 0$ is a penalty (tuning) parameter used to control sparsity, and $0 \le \alpha \le 1$ yields a convex combination of lasso and group lasso penalties ($\alpha = 0$ gives the group-lasso fit while $\alpha = 1$ yields the lasso fit). In (8), an ℓ_1 penalty term is applied to each off-diagonal element of $\boldsymbol{\Omega}$ via $\alpha\lambda\,\|\boldsymbol{\Omega}^-\|_1$ (lasso), and to the off-block-diagonal group of m^2 terms in (4)–(5) via $(1-\alpha)\lambda\sum_{j\ne k}^{p} \|\boldsymbol{\Omega}^{(jk)}\|_F$ (group lasso). The function $L(\boldsymbol{X}; \boldsymbol{\Omega})$ is strictly convex in $\boldsymbol{\Omega} \succ \mathbf{0}$.

A. Optimization

We will use the ADMM approach [31] with variable splitting. The method is similar, but not identical, to a method in [3]. Using variable splitting, consider

$$\min_{\boldsymbol{\Omega}\succ\mathbf{0}, \boldsymbol{W}} \Big\{\text{tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Omega}) - \ln(|\boldsymbol{\Omega}|) + P(\boldsymbol{W})\Big\} \text{ subject to } \boldsymbol{\Omega} = \boldsymbol{W}. \tag{9}$$

The scaled augmented Lagrangian for this problem is [31]

$$L_\rho = \text{tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Omega}) - \ln(|\boldsymbol{\Omega}|) + P(\boldsymbol{W}) + \frac{\rho}{2}\|\boldsymbol{\Omega} - \boldsymbol{W} + \boldsymbol{U}\|_F^2 \tag{10}$$

where $\boldsymbol{U}$ is the dual variable, and $\rho > 0$ is the penalty parameter. Given the results $\boldsymbol{\Omega}^{(i)}, \boldsymbol{W}^{(i)}, \boldsymbol{U}^{(i)}$ of the ith iteration, in the $(i+1)$st iteration, an ADMM algorithm executes the following three updates:

a) $\boldsymbol{\Omega}^{(i+1)} \leftarrow \arg\min_{\boldsymbol{\Omega}} L_a(\boldsymbol{\Omega})$, $L_a(\boldsymbol{\Omega}) := \text{tr}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Omega}) - \ln(|\boldsymbol{\Omega}|) + \frac{\rho}{2}\|\boldsymbol{\Omega} - \boldsymbol{W}^{(i)} + \boldsymbol{U}^{(i)}\|_F^2$

b) $\boldsymbol{W}^{(i+1)} \leftarrow \arg\min_{\boldsymbol{W}} L_b(\boldsymbol{W})$, $L_b(\boldsymbol{W}) := \alpha\lambda\,\|\boldsymbol{W}^-\|_1 + (1-\alpha)\lambda\sum_{i\ne j}^{p} \|\boldsymbol{W}^{(ij)}\|_F + \frac{\rho}{2}\|\boldsymbol{\Omega}^{(i+1)} - \boldsymbol{W} + \boldsymbol{U}^{(i)}\|_F^2$

c) $\boldsymbol{U}^{(i+1)} \leftarrow \boldsymbol{U}^{(i)} + (\boldsymbol{\Omega}^{(i+1)} - \boldsymbol{W}^{(i+1)})$

A necessary and sufficient condition for a global optimum in update (a) is that the gradient of $L_a(\boldsymbol{\Omega})$ w.r.t. $\boldsymbol{\Omega}$, given by (11), vanishes, with $\boldsymbol{\Omega} = \boldsymbol{\Omega}^\top \succ \mathbf{0}$:

$$\mathbf{0} = \frac{\partial L_a(\boldsymbol{\Omega})}{\partial \boldsymbol{\Omega}} = \hat{\boldsymbol{\Sigma}} - \boldsymbol{\Omega}^{-1} + \rho(\boldsymbol{\Omega} - \boldsymbol{W}^{(i)} + \boldsymbol{U}^{(i)}) . \quad (11)$$

The solution to (11) follows from [31, Sec. 6.5]. Rewrite (11) as

$$\hat{\boldsymbol{\Sigma}} - \rho\left(\boldsymbol{W}^{(i)} - \boldsymbol{U}^{(i)}\right) = \boldsymbol{\Omega}^{-1} - \rho\boldsymbol{\Omega} . \quad (12)$$

Let $\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^\top$ denote the eigen-decomposition of the symmetric matrix $\hat{\boldsymbol{\Sigma}} - \rho(\boldsymbol{W}^{(i)} - \boldsymbol{U}^{(i)})$ where $\boldsymbol{D}$ is diagonal with real values on the diagonal, and $\boldsymbol{V}\boldsymbol{V}^\top = \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I}$. Then $\boldsymbol{\Omega}^{(i+1)} = \boldsymbol{V}\tilde{\boldsymbol{D}}\boldsymbol{V}^\top$ where $\tilde{\boldsymbol{D}}$ is the diagonal matrix with ℓth diagonal element

$$\tilde{\boldsymbol{D}}_{\ell\ell} = \frac{-\boldsymbol{D}_{\ell\ell} + \sqrt{\boldsymbol{D}_{\ell\ell}^2 + 4\rho}}{2\rho} .$$

Since eigenvalues $\tilde{\boldsymbol{D}}_{\ell\ell} > 0$, $\boldsymbol{\Omega}^{(i+1)} \succ \mathbf{0}$, and as shown in [31, Sec. 6.5] for a similar problem, it satisfies (11).

Now we turn to update (b). Notice that $L_b(\boldsymbol{W})$ is completely separable w.r.t. each $m \times m$ subblock $\boldsymbol{W}^{(jk)}$ (defined as in (4)). Therefore, we solve $(\boldsymbol{W}^{(jk)})^{(i+1)} \leftarrow \arg\min_{\boldsymbol{W}^{(jk)}} J_{bjk}(\boldsymbol{W}^{(jk)})$, for subblock indexed by (j,k), where

$$J_{bjk}(\boldsymbol{W}^{(jk)}) := \alpha\lambda\left(\mathbf{1}_{j=k}\|(\boldsymbol{W}^{(jk)})^-\|_1 + \mathbf{1}_{j\neq k}\|\boldsymbol{W}^{(jk)}\|_1\right)$$

$$+\mathbf{1}_{j\neq k}(1-\alpha)\lambda\,\|\boldsymbol{W}^{(jk)}\|_F + \frac{\rho}{2}\|\left(\boldsymbol{\Omega}^{(i+1)} - \boldsymbol{W} + \boldsymbol{U}^{(i)}\right)^{(jk)}\|_F^2$$

Note that sparse lasso penalty applies only to off-diagonal elements of $\boldsymbol{W}$, hence to off-diagonal elements of $\boldsymbol{W}^{(jj)}$ and all elements of $\boldsymbol{W}^{(jk)}$, $j \neq k$. The group-lasso penalty applies only to off-diagonal subblocks $\boldsymbol{W}^{(jk)}$, $j \neq k$. Therefore, for $j = k = 1, 2, \cdots, p$ (that is, diagonal subblocks), we have

$$[(\boldsymbol{W}^{(jj)})^{(i+1)}]_{st} = \begin{cases} [(\boldsymbol{\Omega}^{(jj)})^{(i+1)}]_{ss} & \text{if } s = t \\ S([(\boldsymbol{\Omega}^{(jj)})^{(i+1)}]_{st}, \frac{\alpha\lambda}{\rho}) & \text{if } s \neq t \end{cases}$$

where

$$S(a,\beta) := (1 - \beta/|a|)_+ a, \quad (a)_+ := \max(0, a),$$

denotes scalar soft thresholding. For $j \neq k$, following [25], [26], the solution to update (b) is given by

$$[(\boldsymbol{W}^{(jk)})^{(i+1)}]_{st} = \begin{cases} [(\boldsymbol{\Omega}^{(jk)})^{(i+1)}]_{ss}, & \text{if } s = t \\ S([\boldsymbol{A}]_{st}, \frac{\alpha\lambda}{\rho})\left(1 - \frac{(1-\alpha)\lambda}{\rho\|\boldsymbol{S}(\boldsymbol{A}, \frac{\alpha\lambda}{\rho})\|_F}\right)_+, & s \neq t \end{cases}$$

where $\boldsymbol{A} = (\boldsymbol{\Omega}^{(jk)})^{(i+1)} - (\boldsymbol{U}^{(jk)})^{(i)}$ and $\boldsymbol{S}(\boldsymbol{A}, \alpha)$ denotes elementwise matrix soft thresholding, specified by $[\boldsymbol{S}(\boldsymbol{A}, \alpha)]_{st} := S([\boldsymbol{A}]_{st}, \alpha)$. Finally, update (c) is $\boldsymbol{U}^{(i+1)} = \boldsymbol{U}^{(i)} + (\boldsymbol{\Omega}^{(i+1)} - \boldsymbol{W}^{(i+1)})$ [31].

A pseudocode for the ADMM algorithm used in this paper is given in Algorithm 1 where we use the stopping (convergence) criterion following [31, Sec. 3.3.1] and varying penalty parameter ρ following [31, Sec. 3.4.1]. The stopping criterion is based on primal and dual residuals being small where, in our case, at $(i+1)$st iteration, the primal residual is given by $\boldsymbol{\Omega}^{(i+1)} - \boldsymbol{W}^{(i+1)}$ and the dual residual by $\rho^{(i)}(\boldsymbol{W}^{(i+1)} - \boldsymbol{W}^{(i)})$. Convergence criterion is met when the norms of these residuals are below primary and dual tolerances τ_{pri} and τ_{dual}, respectively; see line 8 of Algorithm 1. In turn, τ_{pri} and τ_{dual} are chosen using an absolute and relative criterion as in line 8 of Algorithm 1 where τ_{abs} and τ_{rel} are user chosen absolute and relative tolerances, respectively. As stated in [31, Sec. 3.4.1], one may use "possibly different penalty parameters $\rho^{(i)}$ for each iteration, with the goal

Algorithm 1: ADMM Algorithm for Sparse-Group Graphical Lasso.

Input: Number of samples n, number of nodes p, number of attributes m, data $\{\boldsymbol{x}(t)\}_{t=0}^{n-1}$, $\boldsymbol{x} \in \mathbb{R}^{mp}$, regularization and penalty parameters λ, α and ρ_0, tolerances τ_{abs} and τ_{rel}, variable penalty factor μ, maximum number of iterations i_{max}

Output: estimated inverse covariance $\hat{\boldsymbol{\Omega}}$ and edge-set $\hat{\mathcal{E}}$

1: Calculate sample covariance $\hat{\boldsymbol{\Sigma}} = \frac{1}{n}\sum_{t=0}^{n-1} \boldsymbol{x}(t)\boldsymbol{x}^\top(t)$ (after centering $\boldsymbol{x}(t)$).

2: Initialize: $\boldsymbol{U}^{(0)} = \boldsymbol{W}^{(0)} = \mathbf{0}$, $\boldsymbol{\Omega}^{(0)} = (\mathrm{diag}(\hat{\boldsymbol{\Sigma}}))^{-1}$, where $\boldsymbol{U}, \boldsymbol{W} \in \mathbb{R}^{(mp)\times(mp)}$, $\rho^{(0)} = \rho_0$

3: converged = **false**, $i = 0$

4: **while** converged = **false** **and** $i \leq i_{max}$, **do**

5: Eigen-decompose $\hat{\boldsymbol{\Sigma}} - \rho^{(i)}(\boldsymbol{W}^{(i)} - \boldsymbol{U}^{(i)})$ as $\hat{\boldsymbol{\Sigma}} - \rho^{(i)}(\boldsymbol{W}^{(i)} - \boldsymbol{U}^{(i)}) = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^\top$ with diagonal matrix $\boldsymbol{D}$ consisting of eigenvalues. Define diagonal matrix $\tilde{\boldsymbol{D}}$ with ℓth diagonal element $\tilde{\boldsymbol{D}}_{\ell\ell} = (-\boldsymbol{D}_{\ell\ell} + \sqrt{\boldsymbol{D}_{\ell\ell}^2 + 4\rho^{(i)}})/(2\rho^{(i)})$. Set $\boldsymbol{\Omega}^{(i+1)} = \boldsymbol{V}\tilde{\boldsymbol{D}}\boldsymbol{V}^\top$.

6: Define soft thresholding scalar operator $S(a,\beta) := (1 - \beta/|a|)_+ a$ where $(a)_+ := \max(0, a)$. The diagonal $m \times m$ subblocks of $\boldsymbol{W}$ are updated as

$$[(\boldsymbol{W}^{(jj)})^{(i+1)}]_{st} = \begin{cases} [(\boldsymbol{\Omega}^{(jj)})^{(i+1)}]_{ss} & \text{if } s = t \\ S([(\boldsymbol{\Omega}^{(jj)})^{(i+1)}]_{st}, \frac{\alpha\lambda}{\rho^{(i)}}) & \text{if } s \neq t \end{cases}$$

$j = 1, 2, \cdots, p,\ s, t = 1, 2, \cdots, m$. The off-diagonal $m \times m$ subblocks of $\boldsymbol{W}$ are updated as (denote $\boldsymbol{A} = (\boldsymbol{\Omega}^{(jk)})^{(i+1)} - (\boldsymbol{U}^{(jk)})^{(i)}$)

$$[(\boldsymbol{W}^{(jk)})^{(i+1)}]_{st} = \begin{cases} [(\boldsymbol{\Omega}^{(jk)})^{(i+1)}]_{ss} & \text{if } s = t \\ S([\boldsymbol{A}]_{st}, \frac{\alpha\lambda}{\rho^{(i)}})\left(1 - \frac{(1-\alpha)\lambda}{\rho\|\boldsymbol{S}(\boldsymbol{A}, \frac{\alpha\lambda}{\rho^{(i)}})\|_F}\right)_+ & \text{if } s \neq t \end{cases}$$

where $\boldsymbol{S}(\boldsymbol{A}, \alpha)$ denotes elementwise matrix soft thresholding, specified by $[\boldsymbol{S}(\boldsymbol{A}, \alpha)]_{st} := S([\boldsymbol{A}]_{st}, \alpha)$, and $j \neq k = 1, 2, \cdots, p,\ s, t = 1, 2, \cdots, m$.

7: Dual update $\boldsymbol{U}^{(i+1)} = \boldsymbol{U}^{(i)} + (\boldsymbol{\Omega}^{(i+1)} - \boldsymbol{W}^{(i+1)})$.

8: Check convergence. Set tolerances

$$\tau_{pri} = mp\,\tau_{abs} + \tau_{rel}\ \max(\|\boldsymbol{\Omega}^{(i+1)}\|_F, \|\boldsymbol{W}^{(i+1)}\|_F)$$

$$\tau_{dual} = mp\,\tau_{abs} + \tau_{rel}\,\|\boldsymbol{U}^{(i+1)}\|_F/\rho^{(i)} .$$

Define $d_p = \|\boldsymbol{\Omega}^{(i+1)} - \boldsymbol{W}^{(i+1)}\|_F$ and $d_d = \rho^{(i)}\|\boldsymbol{W}^{(i+1)} - \boldsymbol{W}^{(i)}\|_F$. If $(d_p \leq \tau_{pri})$ **and** $(d_d \leq \tau_{dual})$, set converged = **true**.

9: Update penalty parameter ρ :

$$\rho^{(i+1)} = \begin{cases} 2\rho^{(i)} & \text{if } d_p > \mu d_d \\ \rho^{(i)}/2 & \text{if } d_d > \mu d_p \\ \rho^{(i)} & \text{otherwise} . \end{cases}$$

We also need to set $\boldsymbol{U}^{(i+1)} = \boldsymbol{U}^{(i+1)}/2$ for $d_p > \mu d_d$ and $\boldsymbol{U}^{(i+1)} = 2\boldsymbol{U}^{(i+1)}$ for $d_d > \mu d_p$.

10: $i \leftarrow i + 1$

11: **end while**

12: For $j \neq k$, if $\|\boldsymbol{W}^{(jk)}\|_F > 0$, assign edge $\{j,k\} \in \hat{\mathcal{E}}$, else $\{j,k\} \notin \hat{\mathcal{E}}$. Inverse covariance estimate $\hat{\boldsymbol{\Omega}} = \boldsymbol{W}$.

of improving the convergence in practice, as well as making performance less dependent on the initial choice of the penalty parameter." Line 9 of Algorithm 1 follows typical choices given in [31, Sec. 3.4.1]. For all numerical results presented later, we used $\rho_0 = 2$, $\mu = 10$, and $\tau_{abs} = \tau_{rel} = 10^{-4}$.

The objective function $L(\boldsymbol{X}; \boldsymbol{\Omega})$, given by (7), is strictly convex, and its domain is the set of strictly positive definite matrices (because of $-\ln(|\boldsymbol{\Omega}|)$ and using the log-determinant barrier function [27]). It is also closed, proper and lower semi-continuous. Hence, for any fixed $\rho > 0$, the ADMM algorithm is guaranteed to converge [31, Sec. 3.2], in the sense that we have primal residual convergence to 0, dual residual convergence to 0, and objective function convergence to the optimal value. For varying ρ, the convergence of ADMM has not been proven [31, Sec. 3.4.1].

B. Parameter Tuning, Model Selection and Debiasing

Now we briefly discuss some practical aspects of applying the ADMM algorithm, such as how to select the tuning parameters λ and α. This, in turn, dictates how many edges are connected in the graph. It is also well-known that lasso and related approaches yield biased estimates of inverse covariance [32], [33]. To debias (approximately), for real data, we will mimic the adaptive lasso approach of [32] to propose an adaptive sparse-group lasso approach.

1) Parameter Tuning and Model Selection: In practice, one would select λ and α via cross-validation or an information criterion. Let $\hat{\boldsymbol{\Sigma}}$ and $\hat{\bar{\mathcal{E}}}$ denote the estimated inverse covariance matrix and estimated enlarged edge-set (defined in Sec. II), respectively, and let $|\hat{\bar{\mathcal{E}}}|$ denote the cardinality (# of nonzero elements) of $\hat{\bar{\mathcal{E}}}$. Noting that $\hat{\boldsymbol{\Sigma}}$ is symmetric with nonzero diagonal elements, the number of free nonzero elements of $\hat{\boldsymbol{\Sigma}}$ equal $\frac{1}{2}|\hat{\bar{\mathcal{E}}}|+pm$. For synthetic and real data results presented later, we used the Bayesian information criterion (BIC)

$$\text{BIC}(\lambda, \alpha) = \text{tr}(\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Omega}}) - \ln(|\hat{\boldsymbol{\Omega}}|) + \frac{\ln(n)}{n}\left(\frac{1}{2}|\hat{\bar{\mathcal{E}}}| + pm\right)$$

based on optimized $-\ln f_{\boldsymbol{X}}(\boldsymbol{X}) \propto \frac{n}{2}(\text{tr}(\hat{\boldsymbol{\Sigma}}\hat{\boldsymbol{\Omega}}) - \ln|\hat{\boldsymbol{\Omega}}|)$. The pair (λ, α) is selected to minimize BIC. Instead of searching over a two-dimensional space (λ, α), we first search over a grid of λ values with fixed $\alpha = 0.1\,(= \alpha_0)$, (somewhat arbitrary, $\alpha = 0$ will make it group lasso). Then with λ fixed at the selected value, we search over a grid of α values over [0,1].

We search over λ values in the range $[\lambda_\ell, \lambda_u]$ selected via the following heuristic. For $\alpha = \alpha_0$, we first find the smallest λ, labeled λ_{sm}, for which we get a no-edge model (i.e., $|\hat{\mathcal{E}}| = 0$). To this end, searching over a grid of λ values, we find the largest λ for which the corresponding $|\hat{\mathcal{E}}| > \tau_{th} \approx 0$, i.e, for which we get close to a no-edge model; we take this value of λ as λ_{sm}. We picked $\tau_{th} = 0.002\,p(p-1)/2$ (0.2% of possible $p(p-1)/2$ edges are connected). Then we set $\lambda_u = \lambda_{sm}/2$ and $\lambda_\ell = \lambda_u/10$. The given choice of λ_u precludes "extremely" sparse models while that of λ_ℓ precludes "very" dense models (e.g., more than 50% connected edges). We search over a grid of λ values in the range $[\lambda_\ell, \lambda_u]$ to minimize BIC.

2) Debiasing: Adaptive Sparse-Group Lasso: Lasso and related approaches yield biased estimates [32], [33]. To approximately debias, we will mimic the adaptive lasso approach of [32] to propose an adaptive sparse-group lasso approach where we replace $P(\boldsymbol{\Omega})$ in (8) with $\hat{P}(\boldsymbol{\Omega})$ given below. Given estimates $\hat{\Omega}_{ij}$ obtained from the proposed non-adaptive sparse-group lasso approach, we define

$$\hat{P}(\boldsymbol{\Omega}) = \alpha\lambda \sum_{i\neq j=1}^{mp} \frac{|\Omega_{ij}|}{|\hat{\Omega}_{ij}|} + (1-\alpha)\lambda \sum_{k\neq \ell=1}^{p} \frac{\|\boldsymbol{\Omega}^{(k\ell)}\|_F}{\|\hat{\boldsymbol{\Omega}}^{(k\ell)}\|_F}. \quad (13)$$

Thus, we replace $\alpha\lambda$ with $\alpha\lambda/|\hat{\Omega}_{ij}|$ and $(1-\alpha)\lambda$ with $(1-\alpha)\lambda/\|\hat{\boldsymbol{\Omega}}^{(k\ell)}\|_F$. Now run the ADMM algorithm using adaptive weights (with "obvious" modifications) using the previously selected values of λ and α. Higher $|\hat{\Omega}_{ij}|$ values decrease the penalty, while lower values increase the penalty. We will illustrate this approach later when exploring both synthetic and real data networks.

Selection of λ and α is done via BIC, as in Sec. III-B1 for sparse-group lasso, with the following exception. We select λ_{sm} as before, using sparse-group lasso. Then for adaptive sparse-group lasso (comprised of two steps of sparse-group lasso followed by adaptive sparse-group lasso), we set $\lambda_u = \lambda_{sm}/6$ and $\lambda_\ell = \lambda_u/10$. Notice that we now have the upper limit λ_u one-third of the upper limit for sparse-group lasso. This is based on empirical evidence, and the following observation. Elements of $\boldsymbol{\Omega}$ estimated as zero in the first step of sparse-group lasso will stay zero in the next adaptive step. Too high a value of λ in sparse-group lasso stage results in higher number of zero elements in $\hat{\boldsymbol{\Omega}}$, which in adaptive version will remain as zeros. Reduced λ allows these elements to stay non-zero, thereby allowing adaptive sparse-group lasso to "properly" process such elements. Computation of BIC is done after the second, adaptive step.

IV. Theoretical Analysis

In this section we analyze consistency (Theorem 1) and sparsistency (Theorem 2) of the proposed approach. For consistency we follow the method of [29] which deals with single attribute models and lasso penalty in a high-dimensional setting where we allow p to be a function of sample size n, denoted as p_n. High-dimensional setting allows consideration of the case where number of unknowns $mp(mp+1)/2$ in $\boldsymbol{\Omega}$ is much greater than (or comparable to) the sample size n [34], and as n increases, p_n may increase too maintaining more unknowns than sample size. We also allow λ to be a function of sample size n, denoted as λ_n.

Assume

A1) Define the true edge set $\mathcal{E}_0 = \{\{i,j\} : \boldsymbol{\Omega}_0^{(ij)} \neq \boldsymbol{0},\ i \neq j\}$ where $\boldsymbol{\Omega}_0$ denotes the true inverse covariance of $\boldsymbol{x}$. Assume that $\text{card}(\mathcal{E}_0) = |(\mathcal{E}_0)| \leq s_{n0}$.

A2) The minimum and maximum eigenvalues of $\boldsymbol{\Sigma}_0 = \boldsymbol{\Omega}_0^{-1} \succ \boldsymbol{0}$ satisfy

$$0 < \beta_{\min} \leq \phi_{\min}(\boldsymbol{\Sigma_0}) \leq \phi_{\max}(\boldsymbol{\Sigma_0}) \leq \beta_{\max} < \infty.$$

Here $\beta_{\min}$ and $\beta_{\max}$ are not functions of n.

Let $\hat{\boldsymbol{\Omega}}_\lambda = \arg\min_{\boldsymbol{\Omega}\succ\mathbf{0}} L(\boldsymbol{X};\boldsymbol{\Omega})$. Theorem 1, proved in Appendix A, establishes consistency of $\hat{\boldsymbol{\Omega}}_\lambda$.

Theorem 1. (Consistency): For $\tau > 2$, let

$$C_0 = 40 \max_k(\Sigma_{0kk})\sqrt{2\left(\tau + \ln(4)/\ln(mp_n)\right)}. \quad (14)$$

Given real numbers $\delta_1 \in (0,1)$, $\delta_2 > 0$ and $C_1 > 0$, let $C_2 = \sqrt{m} + 1 + C_1$, and

$$M = (1+\delta_1)^2(2C_2+\delta_2)C_0/\beta_{\min}^2, \quad (15)$$

$$r_n = \sqrt{\frac{(mp_n + m^2 s_{n0})\ln(mp_n)}{n}} = o(1), \quad (16)$$

$$N_1 = 2\left(\ln(4) + \tau\,\ln(mp_n)\right), \quad (17)$$

$$N_2 = \arg\min\left\{n\,:\, r_n \le \frac{\delta_1\beta_{\min}}{(1+\delta_1)^2(2C_2+\delta_2)C_0}\right\}. \quad (18)$$

Suppose the regularization parameter λ_n and $\alpha \in [0,1]$ satisfy

$$\frac{C_1C_0}{1+\alpha(m-1)}\sqrt{\left(1+\frac{p_n}{ms_{n0}}\right)\frac{\ln(mp_n)}{n}} \ge \frac{\lambda_n}{m} \ge C_0\sqrt{\frac{\ln(mp_n)}{n}}. \quad (19)$$

Then if the sample size $n > \max\{N_1, N_2\}$ and assumptions (A1)-(A2) hold true, $\hat{\boldsymbol{\Omega}}_\lambda$ satisfies

$$\|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\|_F \le Mr_n \quad (20)$$

with probability greater than $1 - 1/(mp_n)^{\tau-2}$. In terms of rate of convergence, $\|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\|_F = \mathcal{O}_P(r_n)$ •

Comments on Theorem 1.

- In Theorem 1, the number of attributes m are fixed (not a function of sample size n) whereas number of nodes p and hence, number of connected edges s_0 (s_{n0}) of $\mathcal{E}_0$ are allowed to be a function of n. For Theorem 1 to hold, $\lim_{n\to\infty} r_n = 0$. Clearly the results hold if p is fixed, independent of n.
- The bounds on λ_n in (19) could restrict maximum value of α for larger m values, if C_1 is chosen to be too small. Upperbound of (19) always works for $\alpha = 0$ so long as $C_1 \ge 1$. If we pick $C_1 \ge m$, then the lower bound in (19) is less than the upper bound for every $\alpha \in [0,1]$.

Sparsistency refers to the property that all parameters that are zero are actually estimated as zero with probability tending to one, as $n \to \infty$ [30]. Theorem 2, stated below and proved in Appendix B, deals with sparsistency of $\hat{\boldsymbol{\Omega}}_\lambda$. Its proof follows that of [30, Theorem 2] pertaining to lasso and a larger class of penalty functions (including some non-convex functions).

Theorem 2. (Sparsistency): Suppose all assumptions and conditions of Theorem 1 hold true so that (20) holds. In addition, suppose that there exists a sequence $\eta_n \to 0$ such that $\|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\| = \mathcal{O}_P(\eta_n)$ and $\sqrt{\ln(mp_n)/n} + \eta_n = \mathcal{O}(\lambda_n)$. Then with probability tending to one, $\hat{\Omega}_{\lambda ik} = 0$ for all $\{i,k\} \in \bar{\mathcal{E}}_0^c$ and $\hat{\boldsymbol{\Omega}}_\lambda^{(j\ell)} = 0$ for all $\{j,\ell\} \in \mathcal{E}_0^c$. •

Remark 1: For both consistency and sparsistency to be satisfied, the chosen regularization parameters λ_n's need to be compatible. Theorem 1 imposes upper and lower bounds on the rate of λ_n and Theorem 2 specifies a lower bound. Therefore, for both consistency and sparsistency to be satisfied, we must have

$$\sqrt{\ln(mp_n)/n} + \eta_n \asymp \lambda_n \asymp m\sqrt{\left(1+\frac{p_n}{ms_{n0}}\right)\frac{\ln(mp_n)}{n}}. \quad (21)$$

Its consequences depend upon η_n required to attain $\|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\| = \mathcal{O}_P(\eta_n)$. As in [30], we consider two cases, using the inequalities $\|\boldsymbol{A}\|_F/\sqrt{mp_n} \le \|\boldsymbol{A}\| \le \|\boldsymbol{A}\|_F$ for $\boldsymbol{A} \in \mathbb{R}^{(mp_n)\times(mp_n)}$.

a) Since $\|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\| \le \|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\|_F$, in the worst case where the two have the same order, $\|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\| = \mathcal{O}_P(\sqrt{\frac{(mp_n+m^2s_{n0})\ln(mp_n)}{n}})$ so that $\eta_n = \sqrt{\frac{(mp_n+m^2s_{n0})\ln(mp_n)}{n}}$. Then for (21) to hold true, we should have $1 + m\sqrt{(p_n/m)+s_{n0}} \asymp m\sqrt{1+(p_n/(ms_{n0}))}$, which holds only if $s_{n0} = \mathcal{O}(1)$.

b) Since $\|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\|_F/\sqrt{mp_n} \le \|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\|$, in the optimistic case where the two have the same order, $\|\hat{\boldsymbol{\Omega}}_\lambda - \boldsymbol{\Omega}_0\| = \mathcal{O}_P(\sqrt{(1+\frac{ms_{n0}}{p_n})\ln\frac{mp_n}{n}})$ so that $\eta_n = \sqrt{(1+\frac{ms_{n0}}{p_n})\ln\frac{mp_n}{n}}$. Then for (21) to hold true, we should have $1 + \sqrt{1+\frac{ms_{n0}}{p_n}} \asymp m\sqrt{1+\frac{p_n}{ms_{n0}}}$, which holds only if $s_{n0} = \mathcal{O}(p_n)$.

V. Numerical Examples

We now present numerical results for both synthetic and real data to illustrate the proposed approach. In synthetic data examples the ground truth is known and this allows for assessment of the efficacy of various approaches. In real data examples where the ground truth is unknown, our goal is visualization and exploration of the dependency structures underlying the data, similar to [3], [9], [16], [17].

Various aspects of single-attribute versus multi-attribute graphical modeling have been well-covered in [8], [9], [15], [18]. So we will focus more on sparse-group lasso versus group lasso comparisons for multi-attribute graph estimation. For our numerical examples, in Algorithm 1, we used $\rho_0 = 2$, $\mu = 10$, $\tau_{abs} = \tau_{rel} = 10^{-4}$, with maximum number of iterations i_{max} set at 300 and 2000 for synthetic and real data, respectively.

A. Synthetic Data: Chain Graph

We consider a chain graph (an example in [9]) where p nodes are connected in succession. In the upper triangular $\bar{\boldsymbol{\Omega}}$, using the notation of (4), we set $[\bar{\boldsymbol{\Omega}}^{(jk)}]_{st} = 0.5^{|s-t|}$ for $j = k = 1,\cdots,p$, $s,t = 1,\cdots,m$. For $j \ne k$, if the two nodes are not connected, we have $\bar{\boldsymbol{\Omega}}^{(jk)} = \mathbf{0}$, and if nodes j and k are connected in the chain graph, then $[\bar{\boldsymbol{\Omega}}^{(jk)}]_{st}$ is uniformly distributed over $[-0.4,-0.1]\cup[0.1,0.4]$ if $s \ne t$, and $[\bar{\boldsymbol{\Omega}}^{(jk)}]_{st} = 0$ if $s = t$. Now add $\gamma\boldsymbol{I}$ to $\bar{\boldsymbol{\Omega}}$ with γ picked to make minimum eigenvalue of $\bar{\boldsymbol{\Omega}} + \gamma\boldsymbol{I}$ equal to 0.5. This is similar to simulation example 3 in [9, Sec. 5.1]. With $\boldsymbol{\Phi}\boldsymbol{\Phi}^\top = (\bar{\boldsymbol{\Omega}} + \gamma\boldsymbol{I})^{-1}$, we generate $\boldsymbol{x} = \boldsymbol{\Phi}\boldsymbol{w}$ with $\boldsymbol{w} \in \mathbb{R}^{mp}$ as Gaussian $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0},\boldsymbol{I})$. We generate n i.i.d. observations from $\boldsymbol{x}$, with $m = 3$, $p \in \{100, 400\}$, $n \in \{100, 200, 400, 800\}$. The true value of edgeset cardinality $|\mathcal{E}| = 2(p-1)$.

Simulation results based on 100 runs are shown in Figs. 1–5 for the following approaches: Lasso (proposed ADMM with

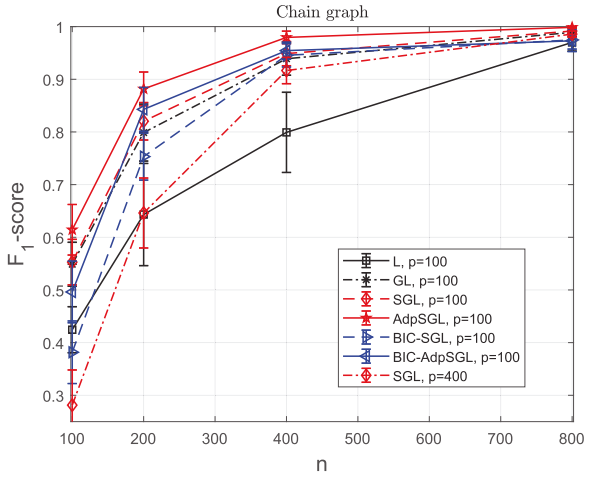


Fig. 1. F_1-score (mean ± 1 std) vs sample size n, for chain graph with $m = 3$.

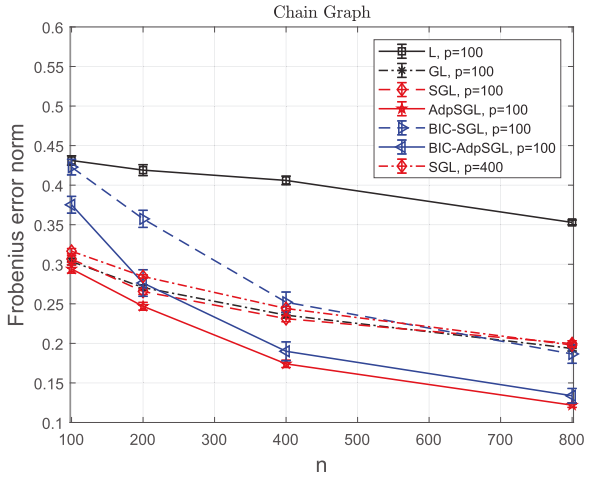


Fig. 2. Error norm $\|\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0\|_F / \|\mathbf{\Omega}_0\|_F$ (mean ± 1 std) vs sample size n, for chain graph with $m = 3$.

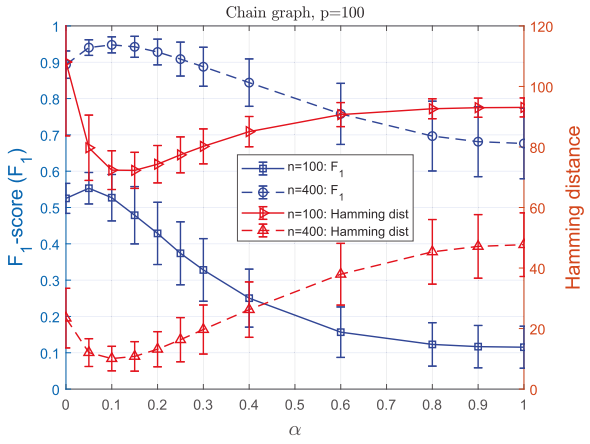


Fig. 3. F_1-score and Hamming distance (mean ± 1 std) vs α, for chain graph with $m = 3$, $p = 100$, λ_n as for Fig. 1. ($\{i, j\}$ and $\{j, i\}$ are counted as one edge in computing the Hamming distance.)

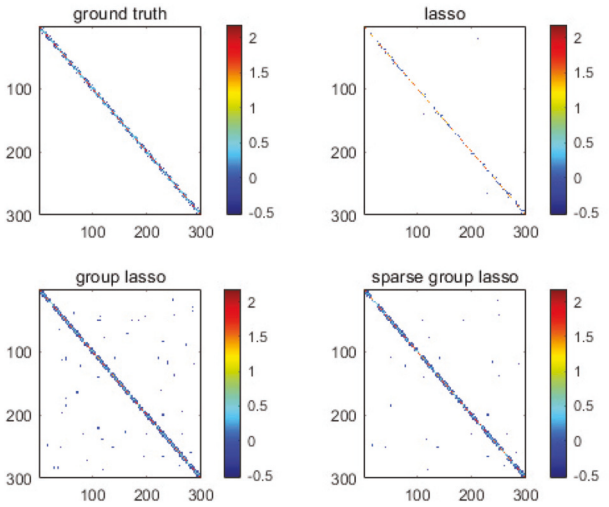


Fig. 4. True and estimated precision matrices, chain graph, $p = 100$, $m = 3$, $n = 200$. Top left is the ground truth, top right the lasso estimate, bottom left is group lasso estimate and bottom right is the sparse-group lasso estimate. All entries that are exactly zero, are color coded as white.

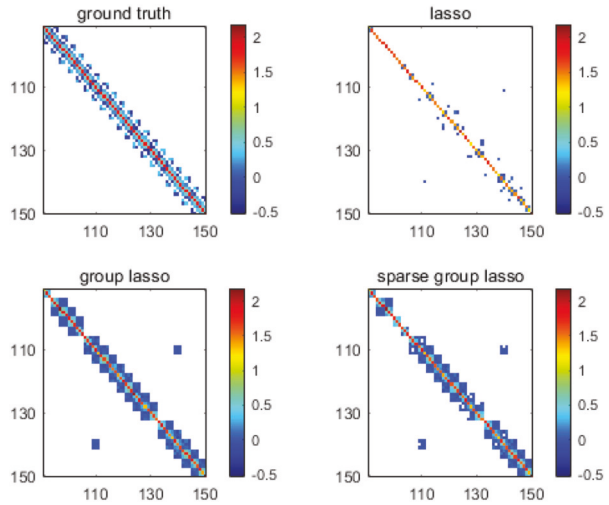


Fig. 5. Subsets of true and estimated precision matrices of Fig. 4, for ease of viewing.

$\alpha = 1$, labeled "L"), Group Lasso (proposed ADMM with $\alpha = 0$, labeled "GL"), Sparse-Group Lasso (proposed ADMM with α as a parameter, labeled "SGL"), adaptive sparse-group lasso (as detailed in Sec. III-B2, labeled "AdpSGL"), and approaches labeled "BIC-SGL" and "BIC-AdpSGL" which are approaches SGL and AdpSGL for which the tuning parameters (λ, α) were selected in each run via BIC as discussed in Sec. III-B. For $p = 100$ and each value of sample size n, for approaches L, GL, SGL and AdpSGL, we selected the tuning parameters (λ, α) by searching over a two-dimensional grid to maximize the F_1-score (averaged over 100 runs), where F_1-score is defined as

TABLE I
TIMINGS FOR ADMM ALGORITHM 1 FOR L (LABELED L(ADMM)), GL AND SGL, AND FOR QUIC ALGORITHM [35] (LABELED L(QUIC)) IMPLEMENTED IN MATLAB, BASED ON 100 RUNS, CHAIN GRAPH

	time (s) for $p = 100$, $m = 3$			
n	L (ADMM)	L (QUIC)	GL	SGL
100	0.735 ± 0.043	0.316 ±0.050	1.16 ± 0.206	1.42 ±0.433
200	0.784 ± 0.045	0.340 ±0.040	1.44 ± 0.505	1.20 ±0.218
400	0.925 ± 0.061	0.378 ±0.029	1.37 ± 0.098	1.31 ±0.083
800	0.974 ± 0.027	0.541 ±0.019	1.38 ± 0.033	1.32 ±0.029
	time (s) for $p = 400$, $m = 3$			
n	L (ADMM)	L (QUIC)	GL	SGL
100	16.28 ± 1.37	57.70 ±2.60	18.09 ± 0.75	19.12 ±0.71
200	14.72 ± 1.29	57.65 ±1.78	18.05 ± 1.75	17.61 ±1.39
400	13.82 ± 1.29	61.69 ±8.03	17.76 ± 2.20	17.40 ±1.94
800	15.31 ± 1.79	85.32 ±9.64	18.69 ± 2.60	18.72 ±2.46

$$F_1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}},$$

$$\text{precision} = \frac{|\hat{\mathcal{E}} \cap \mathcal{E}_0|}{|\hat{\mathcal{E}}|}, \quad \text{recall} = \frac{|\hat{\mathcal{E}} \cap \mathcal{E}_0|}{|\mathcal{E}_0|}$$

and $\mathcal{E}_0$ and $\hat{\mathcal{E}}$ denote the true and estimated edge sets, respectively. In Fig. 1 we show the F_1-score and in Fig. 2 we show the normalized Frobenius error $\|\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0\|_F / \|\mathbf{\Omega}_0\|_F$. We see that GL, SGL and AdpSGL approaches are significantly superior to L in terms of both F_1-score and Frobenius error, particularly at smaller sample sizes. While AdpSGL is also significantly better than GL, SGL is only slightly better than GL. For instance, at $n = 200$ ($p = 100$), the F_1-score is 0.6632, 0.7988, 0.8203 and 0.8817 for L, GL, SGL, and AdpSGL, respectively. For the practical case where the tuning parameters (λ, α) have to be selected based on data, the results for BIC-SGL and BIC-AdpSGL show a loss in F_1-score which significantly narrows with increasing n. For instance, for $n = 100, 200, 400$, and $p = 100$, the F_1-scores are $0.4962,\ 0.8429,\ 0.9544$ for BIC-AdpSGL compared to $0.6143,\ 0.8817,\ 0.9795$ for AdpSGL.

For $p = 400$, we show results only for SGL. We first selected (λ, α) by searching over a two-dimensional grid to maximize the F_1-score, for $p = 100$ and $n = 200$. Based on 100 runs, the selected values were $\lambda = 0.255$ and $\alpha = 0.05$. Then we scale λ for other values of p and n based on (19): $\lambda_n = C\sqrt{\ln(mp)/n}$ with $C = 0.255\sqrt{200/\ln(300)}$. Figs. 1 and 2 show the F_1-score and Frobenius error , respectively, under the label "SGL, p=400."

Table I shows some statistics regarding average timings (mean ± 1 std) per run for the approaches L, GL and SGL using the (λ, α) values optimized for each n and $p = 100$ for the F_1-score (as discussed for Fig. 1) and λ scaled for $p = 400$ using (19), where the ADMM algorithm was implemented in MATLAB R2020b, and run on a Window Home 10 operating system with processor Intel(R) Core(TM) i5-6400T CPU @2.20 GHz with 12 GB RAM. We also show timings for the fast Hessian-based quadratic approximation approach of [35] for lasso ($\alpha = 1$), called QUIC. For $p = 100$ and $n = 100,\ 200,\ 400,\ 800$, the F_1-scores were $0.425,\ 0.652,\ 0.799,\ 0.970$ for L(ADMM) and $0.424,\ 0.652,\ 0.799,\ 0.970$ for L(QUIC), respectively, and for $p = 400$ and $n = 100,\ 200,\ 400,\ 800$, the F_1-scores were $0.263,\ 0.417,\ 0.576,\ 0.914$ for L(ADMM) and $0.263,\ 0.417,\ 0.577,\ 0.914$ for L(QUIC), respectively. As seen in Fig. 1, as sample size becomes large (e.g. $n = 800$), the F_1-score of approach L becomes comparable to GL and SGL, therefore, one may wish to use just lasso for multi-attribute models using fast lasso solvers such as QUIC. We implemented QUIC in MATLAB. The main computational requirement in ADMM is eigen-decomposition in line 5 of the ADMM algorithm whose complexity is $\mathcal{O}((mp)^3)$. Therefore, comparing the results for $p = 100$ and $p = 400$, one would expect to see an increase in timing of the order of approximately 64, but the times displayed in Table I scale by less than 64. While the fast algorithm QUIC is indeed faster than ADMM Algorithm 1 for $p = 100$, it is not so for $p = 400$. We do note that the timing comparisons between QUIC and ADMM given in [35] (for a different example) are based on QUIC implemented in C++ with a MATLAB interface but ADMM implemented in MATLAB and without using variable penalty ρ.

TABLE II
CHAIN GRAPH, $p = 100$: F_1-SCORE AND TIMINGS BASED ON 100 RUNS FOR THE APPROACHES OF KOLAR [9] AND DANAHER *ET AL.* [3], IMPLEMENTED IN MATLAB

	Kolar [9]		Danaher et al. [3]	
n	F_1-score	time (s)	F_1-score	time (s)
100	0.466 ± 0.079	205.0 ±18.7	0.079 ± 0.036	0.75 ±0.05
200	0.802 ± 0.056	46.0 ±18.7	0.179 ± 0.052	0.74 ±0.07
400	0.941 ± 0.033	23.5 ±6.3	0.353 ±0.065	0.72 ±0.02
800	0.989 ± 0.008	15.6 ±3.9	0.532 ±0.048	0.70 ±0.09

Note that QUIC speeds up the algorithm by processing only a subset of variables ("free variables" [35]) for Newton direction computation at any given iteration, with the remaining variables ("fixed variables") in the precision matrix left unprocessed. The number of free variables depends upon λ, with higher λ leading to fewer free variables (sparser estimate), and vice versa. Numerical results in [35, Table 2] show that if one selects optimum λ (the one that results "in the discovery of the correct number of non-zeros" in the precision matrix [35, p. 2936]), then QUIC converges the fastest. If the chosen λ results in denser estimated precision matrix, one has higher number of free variables, and consequently larger time to convergence. Since, using (19), we scaled λ values for $p = 400$ from λ values chosen to optimize performance for $p = 100$, they are not necessarily optimal for $p = 400$ (because (19) and Theorem 1 hold for large n). For optimal λ's one expects the timings to scale as $\mathcal{O}((mp)^3)$, in particular, by a factor of 64 from $p = 100$ to $p = 400$. But for non-optimal λ values this factor can be much larger, as seen in [35, Table 2] for a different problem. Note that ADMM does not make use of this partition of variables into free and fixed sets, therefore, its timings are not as dependent upon choice of λ values. (Another fast gradient-based lasso solver is given in [36] which we have not tested.)

Table II shows the results (F_1-score and timings) of using the approaches of [9] and [3] on chain graph with $p = 100$, $m = 3$. For Kolar [9], we implemented the primal-dual algorithm given therein with error tolerance of 10^{-4} and selected the tuning parameter λ by searching over a grid of values to maximize F_1-score. The approach of [3] is closer to our proposed ADMM

solution, and was implemented similar to our Algorithm 1, and we search for (λ, α) over a two-dimensional grid to maximize the F_1-score. Since for connected nodes j and k in the chain graph, we pick $[\mathbf{\Omega}^{(jk)}]_{st} = 0$ if $s = t$, (an example in [9]), and since for a given edge $\{j, k\}$, [3] exploits $[\mathbf{\Omega}^{(jk)}]_{st}$ only for $s = t$, and not $s \neq t$ (unlike ours and Kolar's approach), one would not expect [3] to do well for this example (as was also noted in [9]). The F_1-scores of Kolar are quite close to the SGL results (for $n = 100, 200, 400, 800$, the F_1-scores are $0.466, 0.802, 0.941, 0.989$ for Kolar compared to $0.549, 0.780, 0.939, 0.989$ for GL) but timings are significantly higher (for $n = 100, 200, 400, 800$, the timings are $205.0, 46.0, 23.5, 15.6$s for Kolar compared to $1.163, 1.442, 1.372, 1.375$s for GL). The approach of [3] yields quite poor F_1 scores, as seen in Table II.

Fig. 3 shows F_1-score vs α for $p = 100$ and $n = 100$ or 400, with λ values equal to the optimized values for SGL, as used for the results of Fig. 1. For example, we have $(\lambda, \alpha) = (0.3253, 0.05)$ for $n = 100$ and $(\lambda, \alpha) = (0.18, 0.1)$ for $n = 400$. Parameter $\alpha = 0$ leads to group-lasso while $\alpha = 1$ is purely lasso. For $n = 100$, the peak empirical F_1-score is 0.551 at $\alpha = 0.05$ (compared to 0.525 at $\alpha = 0$ (group lasso) and 0.115 at $\alpha = 1$ (lasso)), and the minimum Hamming distance between $\mathcal{E}_0$ and $\hat{\mathcal{E}}$ is 72.3 at $\alpha = 0.15$ (compared to 108.2 at $\alpha = 0$ and 93.1 at $\alpha = 1$). For $n = 400$, the peak empirical F_1-score is 0.948 at $\alpha = 0.10$ (compared to 0.893 at $\alpha = 0$ and 0.676 at $\alpha = 1$), and the minimum Hamming distance is 10.08 at $\alpha = 0.10$ (compared to 23.5 at $\alpha = 0$ and 47.7 at $\alpha = 1$). Fig. 3 highlights possible advantages of using sparse-group lasso instead of just group-lasso or just lasso: performance could be significantly improved by allowing $\alpha \neq 0$ or 1; however, the gains may not be as significant if one optimizes w.r.t. λ for each α separately. While group-lasso enforces multi-attribute graph modeling explicitly, there are some gains to be had by also incorporating sparsity within the groups if such is the case; see also [25], [26].

For the chain graph, $p = 100$, $m = 3$, and $n = 200$, we show the true (ground truth) and estimated $(mp) \times (mp)$ inverse covariance matrices for a single run in Figs. 4 and 5 (the latter is a subset of the former, scaled for ease of viewing), using approaches lasso (L, $\alpha = 1$), group lasso (GL, $\alpha = 0$) and sparse group lasso (SGL, $\alpha = 0.05$), implemented with the corresponding optimized tuning parameters as for Fig. 1. In these figures all matrix entries that are exactly zero, are color coded as white, other matrix elements follow the displayed colorbar coding. Lasso does not impose group penalty, hence, for $j = k$, estimate of the $m \times m$ subblock $[\mathbf{\Omega}^{(jk)}]_{st}$, $s, t = 1, 2, \cdots, m$, yields nonzero diagonals ($s = t$) and several zero off-diagonals, unlike GL and SGL. On the other hand, GL imposes only group penalty which can lead to errors in "full" $m \times m$ subblocks (notice the "thick" nonzero subblocks off the diagonal in Fig. 5) whereas errors in SGL result in "partial" subblocks since individual elements are also penalized. The F_1-scores for the given run were 0.690, 0.809 and 0.833 for L, GL and SGL, respectively.

B. Synthetic Data: Erdös-Rènyi Graph

Now we consider an Erdös-Rènyi graph where p nodes are connected to each other with probability $p_{er} = 0.05$. In the

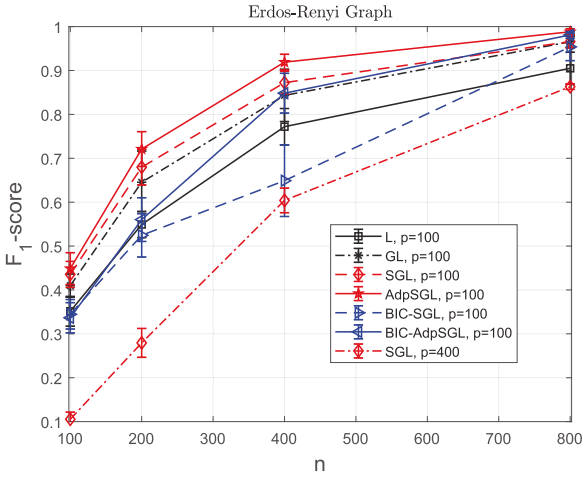


Fig. 6. F_1-score (mean ± 1 std) vs sample size n, for Erdös-Rènyi graph with node connection probability of 0.05, $m = 3$.

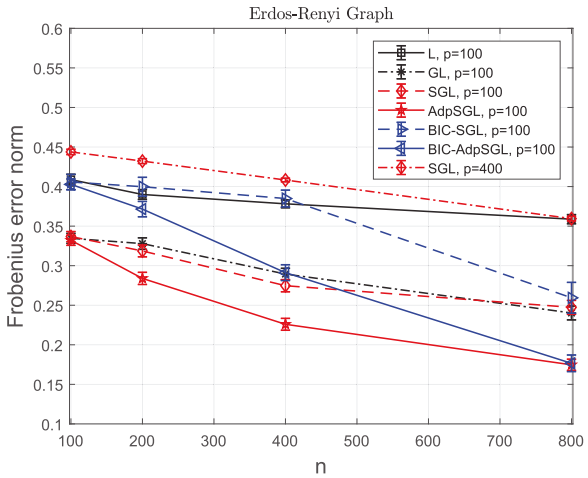


Fig. 7. Error norm $\|\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0\|_F / \|\mathbf{\Omega}_0\|_F$ (mean ± 1 std) vs sample size n, for Erdös-Rènyi graph with node connection probability of 0.05, $m = 3$.

upper triangular $\bar{\mathbf{\Omega}}$, using the notation of (4), we set $[\bar{\mathbf{\Omega}}^{(jk)}]_{st} = 0.5^{|s-t|}$ for $j = k = 1, \cdots, p$, $s, t = 1, \cdots, m$. For $j \neq k$, if the two nodes are not connected, we have $\bar{\mathbf{\Omega}}^{(jk)} = \mathbf{0}$, and if nodes j and k are connected in the chain graph, then $[\bar{\mathbf{\Omega}}^{(jk)}]_{st}$ is uniformly distributed over $[-0.4, -0.1] \cup [0.1, 0.4]$ if $s \neq t$, and $[\bar{\mathbf{\Omega}}^{(jk)}]_{st} = 0$ if $s = t$. Now add $\gamma \boldsymbol{I}$ to $\mathbf{\Omega}$ with γ picked to make minimum eigenvalue of $\mathbf{\Omega} + \gamma \boldsymbol{I}$ equal to 0.5. With $\mathbf{\Phi}\mathbf{\Phi}^\top = (\mathbf{\Omega} + \gamma \boldsymbol{I})^{-1}$, we generate $\boldsymbol{x} = \mathbf{\Phi}\boldsymbol{w}$ with $\boldsymbol{w} \in \mathbb{R}^{mp}$ as Gaussian $\boldsymbol{w} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. We generate n i.i.d. observations from $\boldsymbol{x}$, with $m = 3$, $p \in \{100, 400\}$, $n \in \{100, 200, 400, 800\}$. We then have $\frac{1}{2}\mathbb{E}\{|\mathcal{E}|\} = 247.5$ and 3990 for $p = 100$ and 400, respectively.

Simulation results based on 100 runs are shown in Figs. 6, 7 and 8 for $p = 100$ and 400 nodes. Figs. 6, 7 and 8 are counterparts of Figs. 1, 2 and 3 pertaining to the chain graph. For $p = 400$, we show results only for SGL in Figs. 6 and 7, similar to Figs. 1 and 2, where we first selected the tuning parameters (λ, α) by searching over a two-dimensional grid to maximize the F_1-score

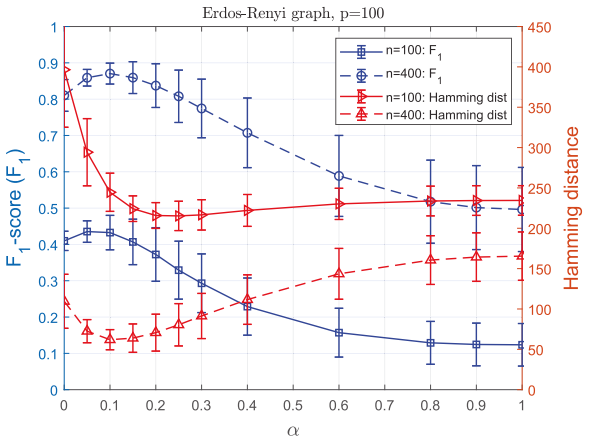


Fig. 8. F_1-score and Hamming dist. (mean ± 1 std) vs α for Erdös-Rènyi graph with node connection probability of 0.05, $m = 3$, $p = 100$, λ_n as for Fig. 6.

for $(p, n) = (400, 400)$, resulting in $(\lambda, \alpha) = (0.08, 0.05)$. Then we scale λ for other values of n based on (19): $\lambda_n = C_{400}/\sqrt{n}$ with $C_{400} = 0.08\sqrt{400}$ for $p = 400$. For $p = 100$, for each n, the tuning parameters were selected by exhaustive search or via BIC, exactly as for Figs. 1 and 2. We see that the performance order among these approaches is as follows: AdpSGL is better than SGL which is better than GL, and all the three approaches are superior to L in terms of both the F_1-score and Frobenius error. For the practical case where the tuning parameters (λ, α) have to be selected based on data, the results for BIC-SGL and BIC-AdpSGL show a loss in F_1-score which significantly narrows with increasing n. For instance, for $n = 100, 200, 400, 800$, and $p = 100$, the F_1-scores are $0.3366,\ 0.5603,\ 0.8483,\ 0.9810$ for BIC-AdpSGL compared to $0.4487,\ 0.7210,\ 0.9190,\ 0.9882$ for AdpSGL.

Fig. 8 shows F_1-score vs α for $p = 100$ and $n = 100$ or 400, with λ_n's as for Fig. 6, and Fig. 8 is the counterpart of Fig. 3 pertaining to the chain graph. The discussion pertaining to Fig. 3 applies here too.

C. Real Data Example: Gross Domestic Product (GDP) Network

As in [18], we consider regional GDP data obtained from the U.S. Department of Commerce (DoC) website.[1] We picked data for the following 20 different industries with DoC labels: (1) utilities (uti), (2) construction (cons) , (3) manufacturing (manu), (4) Durable goods manufacturing (durable), (5) nondurable goods manufacturing (nondu), (6) wholesale trade (wholesale), (7) retail trade (retail), (8) transportation and warehousing (trans), (9) information (info), (10) finance and insurance (finance), (11) real estate and rental and leasing (real), (12) professional, scientific and technical services (prof), (13) management of companies and enterprises (manage), (14) administrative and waste management services (admin), (15) educational services (edu), (16) health care and social assistance (health), (17) arts, entertainment and recreation (arts), (18) accommodation and food services (food), (19) other services except government (other), and (20) government (gov). The quarterly data are available from the first quarter of 2005 to the first quarter of 2020, but we used the data from first quarter of 2005 to the first quarter of 2016 since some of the later data are undisclosed (confidential). We used data pertaining to 8 regions in the US: Far West (FWST: Alaska, California, Hawaii, Nevada, Oregon and Washington), Great Lakes (GLAK: Illinois, Indiana, Michigan, Ohio and Wisconsin), Mideast (MEST: Delaware, D.C., Maryland, New Jersey, New York and Pennsylvania), New England (NENG: Connecticut, Maine, Massachusetts, New Hampshire, Rhode Island and Vermont), Plains (PLNS: Iowa, Kansas, Minnesota, Missouri, Nebraska, North Dakota and South Dakota), Rocky Mountain (RKMN: Colorado, Idaho, Montana, Utah and Wyoming), Southeast (SEST: Alabama, Arkansas, Florida, Georgia, Kentucky, Louisiana, Mississippi, North Carolina, South Carolina, Tennessee, Virginia and West Virginia), and Southwest (SWST: Arizona, New Mexico, Oklahoma and Texas).

We model this GDP network as a multi-attribute graphical model with $p = 20$ nodes (industries) and $m = 8$ attributes (regions). The objective is visualization and exploration of the dependency structures (conditional dependencies) among these 20 industries. One could merge the region-wise data into nationwide data and then estimate the single-attribute graph for these industries; this would ignore regional differences and more granular information available in the data. Or, one could separately fit graphs for each of the eight regions, and then combine them somehow for a final graph for the entire nation; this ignores any group structure that may be present in the data, and moreover, it is not clear how to combine the various fitted graphs. Multi-attribute graphical modeling preserves/exploits the group structure while maintaining regional differences.

Thus we have a 160-dimensional $(m = 8, p = 20, mp = 160)$ single-attribute time series with n=48 samples. We pre-process the data by first detrending it (i.e., remove the best straight-line fit linear trend from each component series using the MATLAB function detrend). Then we normalized the entire dataset to have a mean-square value of one. To estimate the multi-attribute graph, we use the BIC-based method outlined in Sec. III-B to select the tuning parameters λ and α. The selected values turn out be $(\lambda, \alpha) = (1, 0.5)$ for sparse-group lasso and $(\lambda, \alpha) = (0.167, 0.1)$ for adaptive sparse-group lasso. The estimated graph using the sparse-group lasso ADMM algorithm has 46 distinct edges (out of possible 190) and the graph is shown in Fig. 9. For Fig. 9, we take $\|\hat{\boldsymbol{\Omega}}^{(jk)}\|_F$ as the edge weight for edge $\{j, k\}$, normalize maximum value to one, and show the resulting graph with colored edge weights and link thickness also reflecting edge weight. With maximum edge weight $\|\hat{\boldsymbol{\Omega}}^{(jk)}\|_F$ normalized to one, we quantize the interval [0,1] to 4 link thicknesses. The estimated graph using adaptive sparse-group lasso has 54 edges and the graph is shown in Fig. 10. We repeated this procedure (BIC-based tuning parameter selection, for both lasso and adaptive lasso graphs) for single-attribute graphical modeling for each of the eight attributes and

[1] https://www.bea.gov/index.html

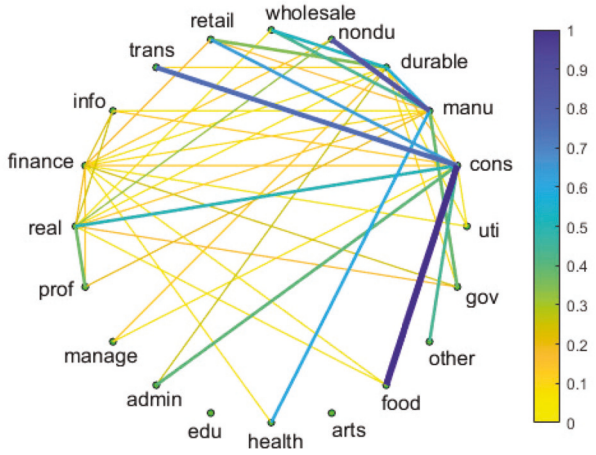


Fig. 9. GDP graph: sparse-group lasso, $\frac{1}{2}|\hat{\mathcal{E}}| = 46$.

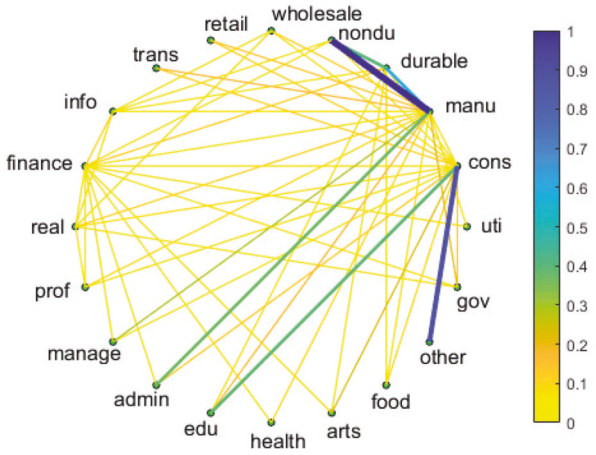


Fig. 10. GDP graph: adaptive sparse-group lasso, $\frac{1}{2}|\hat{\mathcal{E}}| = 54$.

nationwide data (pool all regional data). The selected λ values range from 0.04 to 0.075 for lasso, and from 0.013 to 0.018 for adaptive lasso. The resulting adaptive lasso graphs are shown in Fig. 11.

Some edge statistics are tabulated in Table III. Comparing multi-attribute graph of Fig. 10 with the single attribute graphs in Fig. 11, as well as examining the statistics in Table III, we see that multi-attribute graph captures edges not found in single attribute graphs while rejecting some edges in single attribute graphs that do not find support across the various single attribute graphs. It is also interesting to note that there are no edges common to all eight single attribute regional graphs.

The average node degree in Fig. 10 is 5.4. In Fig. 10, there are two groups of most connected industries: the first group is comprised of construction, manufacturing, and finance and insurance, with 16, 17 and 13 edges, respectively, and the second group is comprised of durable goods manufacturing, non-durable goods manufacturing, information, and real estate and rental and leasing, with 8, 5, 6 and 7 edges, respectively. The hubs comprised of construction, manufacturing, and finance and insurance are not surprising as they are typical drivers

TABLE III
NUMBER OF ESTIMATED EDGES IN REGIONAL/NATIONWIDE SINGLE ATTRIBUTE (SA) GDP GRAPHS AND MULTI-ATTRIBUTE (MA) GDP GRAPH. POSSIBLE EDGES 190 ($\{i,j\}$ AND $\{j,i\}$ ARE COUNTED AS ONE EDGE), SAMPLE SIZE n=48, p=20, $m = 8$. SEE THE TEXT FOR REGION LABELS

data	approach: non-adaptive	approach: adaptive	common to adaptive MA and adaptive SA
MA	46	54	
SA: FWST	32	37	13
SA: GLAK	53	45	11
SA: MEST	41	45	16
SA: NENG	51	47	17
SA: PLNS	30	44	14
SA: RKMT	51	46	16
SA: SEST	46	42	9
SA: SWST	38	43	16
SA: nationwide	37	29	10

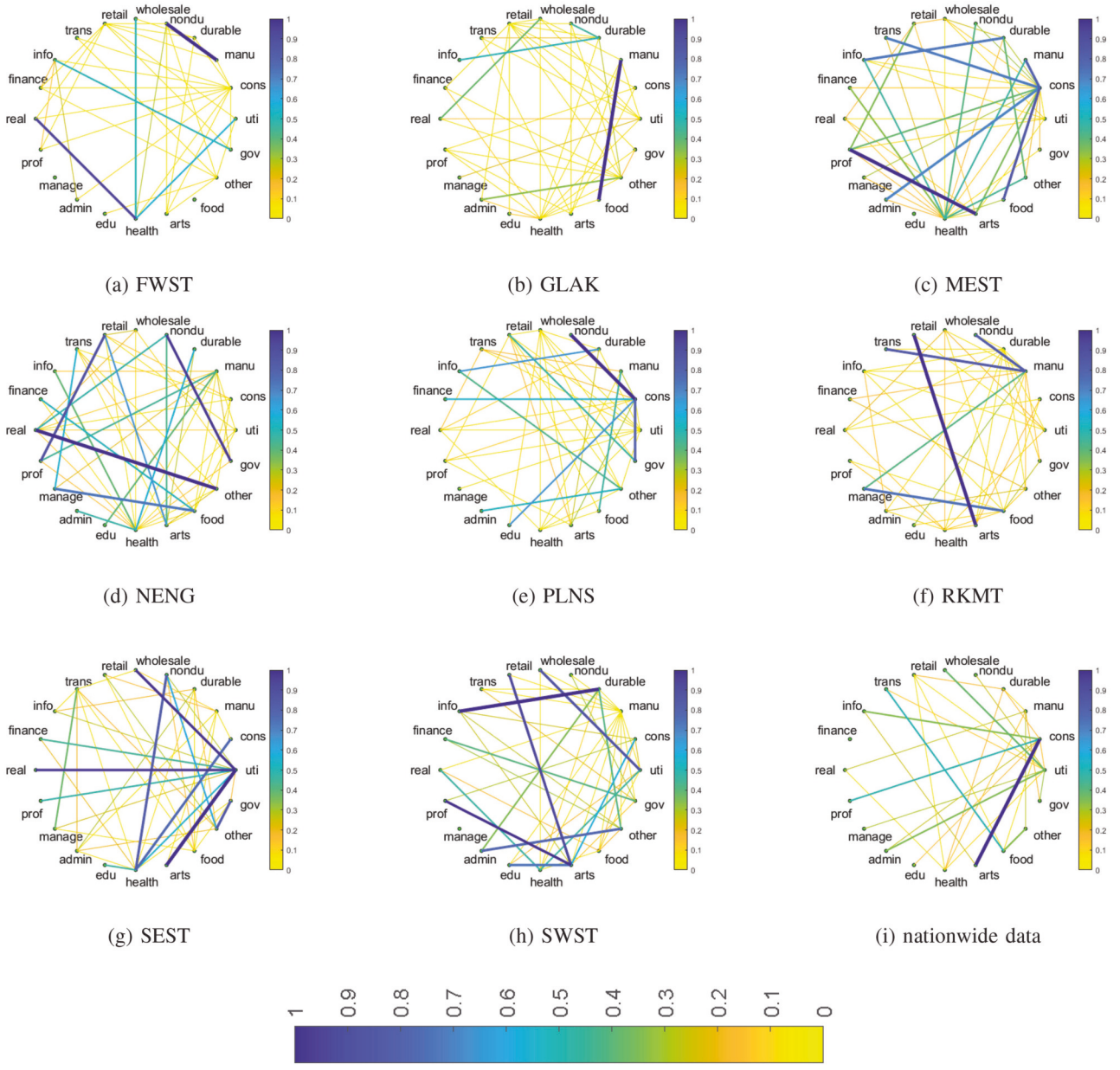
of GDP. Educational services is connected to durable goods manufacturing, construction and manufacturing in Fig. 10 but is not connected to any node in Fig. 9. Noting that educational services sector includes food and accommodation services to the students, its conditional dependence on durable goods manufacturing, construction and manufacturing in Fig. 10 seems to be plausible given construction and maintenance of student housing and food services, in addition to that of classrooms. In view of the synthetic data results, adaptive sparse-group lasso graph of Fig. 10 would appear to be more accurate than the sparse-group lasso graph of Fig. 9 where educational services node sits in isolation. An in-depth analysis would require domain expertise.

D. Real Data Example: Graphs of Color Texture Images

Following [16], [17] where grayscale texture images from a University of Southern California (USC-SIPI) database are considered, we now consider color textures from the Amsterdam Library of Textures (ALOT)[2] [37]. We use two versions of the image 108 (fake fur), images c111.png and c111r60.png (160×160 patches shown in Fig. 12 where c111 is labeled as image 1 and c111r60 is labeled as image 2), photographed from different angles. These two images are 3072×1536 RGB color pixels (1024×1536 for each of RGB). For image 1 we extracted rows 1 through 160, and columns 186 through 345, and for image 2 we extracted rows 700 through 859, and columns 1 through 160, to create the 160×160 patches used for inferring image graphs.

The 160×160 patches were partitioned into non-overlapping 8×8 blocks, vectorized into 64-pixel columns, 3 colors associated with each pixel. Thus, we have $m = 3, p = 64$ and $n = 400$. The data were centered and mean-square value normalized to one before processing. To select λ and α, we use the BIC-based method outlined in Sec. III-B. For texture shown in Fig. 12(a),

[2]http://aloi.science.uva.nl/public_alot



(a) FWST (b) GLAK (c) MEST

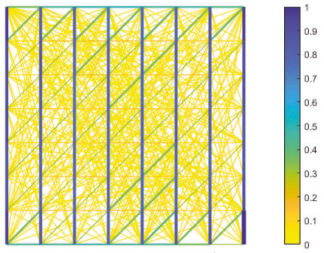
(d) NENG (e) PLNS (f) RKMT

(g) SEST (h) SWST (i) nationwide data

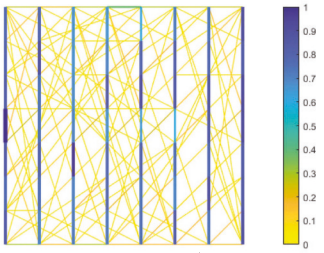
Fig. 11. GDP graphs for 8 regions and nationwide data: single attribute adaptive lasso.

the selected values turn out be $(\lambda, \alpha) = (0.0625, 0.25)$ for sparse-group lasso and $(\lambda, \alpha) = (0.0417, 0.1)$ for adaptive sparse-group lasso, and for texture shown in Fig. 12(d), the selected values are $(\lambda, \alpha) = (0.0775, 0.25)$ for sparse-group lasso and $(\lambda, \alpha) = (0.0517, 0.1)$ for adaptive sparse-group lasso. Using these values for graph estimation, we obtain $\frac{1}{2}|\hat{\mathcal{E}}| = 554$ and $\frac{1}{2}|\hat{\mathcal{E}}| = 208$ with sparse-group lasso and adaptive sparse-group lasso, respectively, for image 1 with estimated graphs shown in Figs. 12(b) and 12(c), respectively. For image 2, we obtain $\frac{1}{2}|\hat{\mathcal{E}}| = 467$ and $\frac{1}{2}|\hat{\mathcal{E}}| = 204$ with sparse-group lasso and adaptive sparse-group lasso, respectively, with estimated graphs shown in Figs. 12(e) and 12(f), respectively. We take $\|\hat{\mathbf{\Omega}}^{(jk)}\|_F$ as the edge weight for edge $\{j, k\}$, normalize maximum value to one, and show the resulting graphs (arranged as 8×8 nodes) with colored edge weights and link thickness also reflecting edge weight. (With maximum edge weight $\|\hat{\mathbf{\Omega}}^{(jk)}\|_F$ normalized to one, we quantize the interval [0,1] to 4 link thicknesses.) Compare Figs. 12(a), 12(b) and 12(c), and Figs. 12(d), 12(e) and 12(f), respectively, to note that the strong link weights follow the texture orientation: primarily vertical and some slanting left in Figs. 12(a), 12(b) and 12(c), and primarily slanting right and some horizontal in Figs. 12(d), 12(e) and 12(f). Weaker edge weights connect pixels in "other" directions. These observations provide "visual" support for fitted graphs, confirming our intuition; we do not know the ground truth. Observe also that the results obtained via adaptive sparse-group lasso are much sparser than those via sparse-group lasso. This is unlike the GDP graphs in Sec. V-C where the sample size is quite small ($n = 48$

(a) image 1

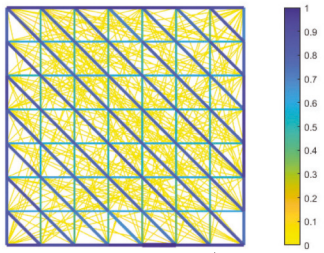


(b) Graph of (a), $|\hat{\mathcal{E}}| = 554$, sparse-group lasso

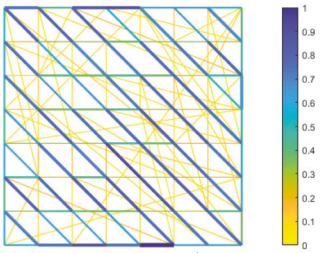


(c) Graph of (a), $|\hat{\mathcal{E}}| = 208$, adaptive sparse-group lasso

(d) image 2



(e) Graph of (d), $|\hat{\mathcal{E}}| = 467$, sparse-group lasso



(f) Graph of (d), $|\hat{\mathcal{E}}| = 204$, adaptive sparse-group lasso

Fig. 12. Color texture graph example (textures are from http://aloi.science.uva.nl/public_alot).

for $(p, m) = (20, 8)$ for GDP graphs compared to $n = 400$ for $(p, m) = (64, 3)$ for image graphs).

VI. Conclusion

We proposed a sparse-group lasso based penalized log-likelihood approach for graph learning from multi-attribute data. Prior work of [8], [9], [15], [18] considers only group lasso which is a special case of sparse-group lasso. An ADMM algorithm was presented to optimize the objective function to estimate the inverse covariance matrix and the edges in the graph. We provided sufficient conditions for convergence in the Frobenius norm of the estimator to the true value, a rate of convergence, and also considered sparsistency.

We tested the proposed approach on synthetic as well as real data. While the ground truth is unknown in the real data applications, requiring domain expert knowledge to interpret the results (estimated graphs), the synthetic data examples clearly demonstrate the advantages of using sparse-group lasso instead of just group-lasso or just lasso.

It is of interest to perform theoretical analysis of the adaptive sparse-group lasso approach outlined in Sec. III-B2.

Appendix A
Proof of Theorem 1

Here we prove Theorem 1. First we need Lemmas 1 and 2. Lemma 1 below is specialization of [27, Lemma 1] to Gaussian random vectors. It follows from [27, Lemma 1] after setting the sub-Gaussian parameter σ in [27, Lemma 1] to 1.

Lemma 1: Consider a zero-mean Gaussian random vector $\boldsymbol{z} \in \mathbb{R}^p$ with covariance $\boldsymbol{R} \succ \mathbf{0}$. Given n i.i.d. samples $\boldsymbol{z}(t)$, $t = 1, 2, \cdots, n$, of $\mathbf{z}$, let $\hat{\boldsymbol{R}} = (1/n)\sum_{t=1}^{n} \boldsymbol{z}\boldsymbol{z}^\top$ denote the sample covariance matrix. Then $\hat{\boldsymbol{R}}$ satisfies the tail bound

$$P\left(\left|[\hat{\boldsymbol{R}} - \boldsymbol{R}]_{ij}\right| > \delta\right) \leq 4\,\exp\left(-\frac{n\delta^2}{3200\max_i(R_{ii}^2)}\right) \quad (22)$$

for all $\delta \in (0, 40\max_i(R_{ii}))$ •

Now we state Lemma 2 and provide a proof.

Lemma 2: Under Assumption (A2), the sample covariance $\hat{\boldsymbol{\Sigma}}$ satisfies the tail bound

$$P\left(\max_{k,l}\left|[\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0]_{kl}\right| > C_0\sqrt{\frac{\ln(mp_n)}{n}}\right) \leq \frac{1}{(mp_n)^{\tau-2}} \quad (23)$$

for $\tau > 2$, if the sample size $n > N_1$, where C_0 is defined in (14) and N_1 is defined in (17). •

Proof: Applying Lemma 1 to our problem, we have

$$P\left(\left|[\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0]_{kl}\right| > \delta\right) \leq 4\,\exp\left(-\frac{n\delta^2}{3200(\Sigma_{0kk}^2)_{max}}\right) \quad (24)$$

for all $\delta \in (0, 40\max_k(\Sigma_{0kk}))$ where $(\Sigma_{0kk}^2)_{max} = \max_{1\leq k\leq mp_n}(\Sigma_{0kk}^2)$. Applying the union bound over all $(mp_n)^2$ entries of $\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0$, we have

$$P\left(\max_{k,l}\left|[\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0]_{kl}\right| > \delta\right) \leq P_{tb}$$

$$= 4(mp_n)^2\,\exp\left(-\frac{n\delta^2}{3200\max_k(\Sigma_{0kk}^2)}\right), \quad (25)$$

for all $\delta \in (0, 40\max_k(\Sigma_{0kk}))$. Let $c_* := 1/(40\max_k(\Sigma_{0kk}))$. Suppose δ is such that

$$c_*\delta = \sqrt{\frac{N_1}{n}} = \sqrt{\frac{2\,\ln(4(mp_n)^\tau)}{n}}$$

where we have used the expression for N_1 from (17). Then $c_*\delta < 1$ for $n > N_1$, and therefore, the bound (25) holds true since $\delta \in (0, 40\max_k(\Sigma_{0kk})) = (0, c_*^{-1})$. Using the definitions

of c_* and N_1, C_0 specified in (14) can be expressed as

$$C_0 = c_*^{-1}\sqrt{\frac{N_1}{\ln(mp_n)}}\,.$$

Suppose we choose δ as

$$\delta = C_0\sqrt{\frac{\ln(mp_n)}{n}} \;\Rightarrow\; \delta = c_*^{-1}\sqrt{\frac{N_1}{n}}\,, \tag{26}$$

then $c_*\delta < 1$ for $n > N_1$, therefore, the bound (25) holds true. Now it remains to show that P_{tb} equals $1/(mp_n)^{\tau-2}$ for δ chosen as in (26). We have

$$\begin{aligned} P_{tb} &= 4(mp_n)^2\,\exp\left(-\frac{n\delta^2}{3200\max_k(\Sigma^2_{0kk})}\right)\\ &= 4(mp_n)^2\,\exp\left(-\frac{nC_0^2\ln(mp_n)}{2nc_*^{-2}}\right)\\ &= 4(mp_n)^2\,\exp\left(-N_1/2\right) = \frac{4(mp_n)^2}{\exp\left(\ln(4(mp_n)^\tau)\right)}\\ &= \frac{1}{(mp_n)^{\tau-2}}\,. \end{aligned} \tag{27}$$

This proves the desired result ■

Now we are ready to prove Theorem 1.

Proof of Theorem 1: Let $\boldsymbol{\Omega} = \boldsymbol{\Omega}_0 + \boldsymbol{\Delta}$ with both $\boldsymbol{\Omega},\ \boldsymbol{\Omega}_0 \succ \mathbf{0}$, and

$$Q(\boldsymbol{\Omega}) := L(\boldsymbol{X};\boldsymbol{\Omega}) - L(\boldsymbol{X};\boldsymbol{\Omega}_0)\,. \tag{28}$$

The estimate $\hat{\boldsymbol{\Omega}}_\lambda$, denoted by $\hat{\boldsymbol{\Omega}}$ hereafter suppressing dependence upon λ, minimizes $Q(\boldsymbol{\Omega})$, or equivalently, $\hat{\boldsymbol{\Delta}} = \hat{\boldsymbol{\Omega}} - \boldsymbol{\Omega}_0$ minimizes $G(\boldsymbol{\Delta}) := Q(\boldsymbol{\Omega}_0 + \boldsymbol{\Delta})$. We will follow, for the most part, the method of proof of [29, Theorem 1] pertaining to lasso penalty. Consider the set

$$\Theta_n(M) := \left\{\boldsymbol{\Delta} \,:\, \boldsymbol{\Delta} = \boldsymbol{\Delta}^\top,\ \|\boldsymbol{\Delta}\|_F = Mr_n\right\} \tag{29}$$

where M and r_n are as in (15) and (16), respectively. Since $G(\hat{\boldsymbol{\Delta}}) \le G(\mathbf{0}) = 0$, if we can show that $\inf_{\boldsymbol{\Delta}}\{G(\boldsymbol{\Delta}) : \boldsymbol{\Delta} \in \Theta_n(M)\} > 0$, then the minimizer $\hat{\boldsymbol{\Delta}}$ must be inside $\Theta_n(M)$, and hence $\|\hat{\boldsymbol{\Delta}}\|_F \le Mr_n$. It is shown in [29, (9)] that

$$\ln(|\boldsymbol{\Omega_0} + \boldsymbol{\Delta}|) - \ln(|\boldsymbol{\Omega}_0|) = \mathrm{tr}(\boldsymbol{\Sigma_0}\boldsymbol{\Delta}) - A_1 \tag{30}$$

where, with $\boldsymbol{H}(\boldsymbol{\Omega}_0, \boldsymbol{\Delta}, v) = (\boldsymbol{\Omega_0} + v\boldsymbol{\Delta})^{-1} \otimes (\boldsymbol{\Omega_0} + v\boldsymbol{\Delta})^{-1}$ and v denoting a scalar,

$$A_1 = \mathrm{vec}(\boldsymbol{\Delta})^\top \left(\int_0^1 (1-v)\boldsymbol{H}(\boldsymbol{\Omega}_0, \boldsymbol{\Delta}, v)\,dv\right)\mathrm{vec}(\boldsymbol{\Delta})\,. \tag{31}$$

Noting that $\boldsymbol{\Omega}^{-1} = \boldsymbol{\Sigma}$ and setting $\bar{\lambda}_1 = \alpha\lambda_n$ and $\bar{\lambda}_2 = (1-\alpha)\lambda_n$, we can rewrite $G(\boldsymbol{\Delta})$ as

$$G(\boldsymbol{\Delta}) = A_1 + A_2 + A_3 + A_4\,, \tag{32}$$

where

$$A_2 = \mathrm{tr}\left((\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0)\boldsymbol{\Delta}\right)\,, \tag{33}$$

$$A_3 = \bar{\lambda}_1\left(\|\boldsymbol{\Omega}_0^- + \boldsymbol{\Delta}^-\|_1 - \|\boldsymbol{\Omega}_0^-\|_1\right)\,, \tag{34}$$

$$A_4 = \bar{\lambda}_2 \sum_{i,j=1;i\neq j}^{p_n}\left(\|\boldsymbol{\Omega}_0^{(ij)} + \boldsymbol{\Delta}^{(ij)}\|_F - \|\boldsymbol{\Omega}_0^{(ij)}\|_F\right)\,. \tag{35}$$

Following [29, p. 502], we have

$$A_1 \ge \frac{\|\boldsymbol{\Delta}\|_F^2}{2(\|\boldsymbol{\Omega}_0\| + \|\boldsymbol{\Delta}\|)^2} \ge \frac{\|\boldsymbol{\Delta}\|_F^2}{2\left(\beta_{\min}^{-1} + Mr_n\right)^2} \tag{36}$$

where we have used the fact that $\|\boldsymbol{\Omega}_0\| = \|\boldsymbol{\Sigma}_0^{-1}\| = \phi_{\max}(\boldsymbol{\Sigma}_0^{-1}) = (\phi_{\min}(\boldsymbol{\Sigma}_0))^{-1} \le \beta_{\min}^{-1}$ and $\|\boldsymbol{\Delta}\| \le \|\boldsymbol{\Delta}\|_F = Mr_n = \mathcal{O}(r_n)$. We now consider A_2 in (33). We have

$$A_2 = \underbrace{\sum_{i,j=1;i\neq j}^{mp_n}[\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0]_{ij}\Delta_{ji}}_{L_1} + \underbrace{\sum_{i=1}^{mp_n}[\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0]_{ii}\Delta_{ii}}_{L_2} \tag{37}$$

To bound L_1, using Lemma 2, with probability $> 1 - 1/(mp_n)^{\tau-2}$,

$$|L_1| \le \|\boldsymbol{\Delta}^-\|_1 \max_{i,j}\left|[\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0]_{ij}\right| \le \|\boldsymbol{\Delta}^-\|_1\, C_0\sqrt{\ln(mp_n)/n}\,. \tag{38}$$

Similarly, by Cauchy-Schwartz inequality, Lemma 2 and (16),

$$\begin{aligned} |L_2| &\le \|\boldsymbol{\Delta}^+\|_1\, C_0\sqrt{\frac{\ln(mp_n)}{n}} \le C_0\sqrt{\frac{mp_n\ln(mp_n)}{n}}\,\|\boldsymbol{\Delta}^+\|_F\\ &\le \|\boldsymbol{\Delta}^+\|_F C_0 r_n\,. \end{aligned} \tag{39}$$

Therefore, with probability $> 1 - 1/(mp_n)^{\tau-2}$,

$$|A_2| \le \|\boldsymbol{\Delta}^-\|_1\, C_0\sqrt{\frac{\ln(mp_n)}{n}} + \|\boldsymbol{\Delta}^+\|_F C_0 r_n\,. \tag{40}$$

We now derive a different bound on A_2. Define $\tilde{\boldsymbol{\Delta}} \in \mathbb{R}^{p_n\times p_n}$ with (i,j)-th element $\tilde{\Delta}_{ij} = \|\boldsymbol{\Delta}^{(ij)}\|_F$, where $\boldsymbol{\Delta}^{(ij)}$ is defined from $\boldsymbol{\Delta}$ similar to (4). By Cauchy-Schwartz inequality,

$$\begin{aligned} \|\boldsymbol{\Delta}^-\|_1 &= \sum_{i,j=1;i\neq j}^{mp_n}|\Delta_{ij}| \le m\|\tilde{\boldsymbol{\Delta}}^-\|_1\\ &\quad + \underbrace{\left(\sum_{k=1}^{p_n}\|\boldsymbol{\Delta}^{(kk)}\|_1 - \|\boldsymbol{\Delta}^+\|_1\right)}_{=:B}\,. \end{aligned} \tag{41}$$

Then using $\sum_k\|\boldsymbol{\Delta}^{(kk)}\|_1 \le m\sum_k\tilde{\Delta}_{kk} \le m\sqrt{p_n}\,\|\tilde{\boldsymbol{\Delta}}^+\|_F$, we have

$$\begin{aligned} |L_2| + C_0\sqrt{\ln(mp_n)/n}\,B &\le C_0\sqrt{\ln(mp_n)/n}\,\left(\sum_{k=1}^{p_n}\|\boldsymbol{\Delta}^{(kk)}\|_1\right)\\ &\le \|\tilde{\boldsymbol{\Delta}}^+\|_F\sqrt{m}\,C_0 r_n \end{aligned}$$

Therefore, an alternative bound is

$$|A_2| \le m\|\tilde{\boldsymbol{\Delta}}^-\|_1\, C_0\sqrt{\ln(mp_n)/n} + \sqrt{m}\,\|\tilde{\boldsymbol{\Delta}}^+\|_F C_0 r_n\,. \tag{42}$$

We now bound A_3 in (34). Considering the true enlarged edge-set $\bar{\mathcal{E}}_0$ corresponding to $\mathcal{E}_0$ (see Sec. II for $\bar{\mathcal{E}}$), let $\bar{\mathcal{E}}_0^c$ denote its complement. For an index set $\boldsymbol{B}$ and a matrix $\boldsymbol{C} \in \mathbb{R}^{p_n\times p_n}$, we write $\boldsymbol{C}_{\boldsymbol{B}}$ to denote a matrix in $\mathbb{R}^{p_n\times p_n}$ such that $[\boldsymbol{C}_{\boldsymbol{B}}]_{ij} = C_{ij}$ if $(i,j) \in \boldsymbol{B}$, and $[\boldsymbol{C}_{\boldsymbol{B}}]_{ij} = 0$ if $(i,j) \notin \boldsymbol{B}$. Then, by definition, $\boldsymbol{\Delta}^- = \boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0} + \boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0^c}$, and $\|\boldsymbol{\Delta}^-\|_1 = \|\boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1 + \|\boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0^c}\|_1$. We have

$$\begin{aligned} A_3 &= \bar{\lambda}_1(\|\boldsymbol{\Omega}_0^- + \boldsymbol{\Delta}^-\|_1 - \|\boldsymbol{\Omega}_0^-\|_1)\\ &= \bar{\lambda}_1(\|\boldsymbol{\Omega}_0^- + \boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1 + \|\boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0^c}\|_1 - \|\boldsymbol{\Omega}_0^-\|_1)\\ &\ge \bar{\lambda}_1(\|\boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0^c}\|_1 - \|\boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1) \end{aligned} \tag{43}$$

where we have used the triangle inequality $\|\boldsymbol{\Omega}_0^- + \boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1 \ge \|\boldsymbol{\Omega}_0^-\|_1 - \|\boldsymbol{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1$. Next we bound A_4 in (35). Considering the true edge-set $\mathcal{E}_0$ for the multi-attribute graph, let $\mathcal{E}_0^c$ denote

its complement. If the edge $\{i,j\} \in \mathcal{E}_0^c$, then $\mathbf{\Omega}_0^{(ij)} = \mathbf{0}$, therefore, $\|\mathbf{\Omega}_0^{(ij)} + \mathbf{\Delta}^{(ij)}\|_F - \|\mathbf{\Omega}_0^{(ij)}\|_F = \|\mathbf{\Delta}^{(ij)}\|_F$. For $\{i,j\} \in \mathcal{E}_0$, by the triangle inequality, $\|\mathbf{\Omega}_0^{(ij)} + \mathbf{\Delta}^{(ij)}\|_F - \|\mathbf{\Omega}_0^{(ij)}\|_F \geq -\|\mathbf{\Delta}^{(ij)}\|_F$. Thus

$$A_4 \geq \bar{\lambda}_2(\|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0^c}\|_1 - \|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0}\|_1)\,. \tag{44}$$

Split A_2 as $A_2 = \alpha A_2 + (1-\alpha)A_2$, apply bound (40) to αA_2 and (42) to $(1-\alpha)A_2$, use

$$\|\mathbf{\Delta}^-\|_1 = \|\mathbf{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1 + \|\mathbf{\Delta}^-_{\bar{\mathcal{E}}_0^c}\|_1\,, \quad \|\tilde{\mathbf{\Delta}}^-\|_1 = \|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0}\|_1 + \|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0^c}\|_1\,,$$

and $C_0' = C_0\sqrt{\ln(mp_n)/n}$, to yield

$$\begin{aligned}
A_2 + A_3 + A_4 \geq\ & -|A_2| + \bar{\lambda}_1(\|\mathbf{\Delta}^-_{\bar{\mathcal{E}}_0^c}\|_1 - \|\mathbf{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1) \\
& + \bar{\lambda}_2(\|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0^c}\|_1 - \|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0}\|_1) \\
\geq\ & -(\alpha\|\mathbf{\Delta}^+\|_F + (1-\alpha)\sqrt{m}\,\|\tilde{\mathbf{\Delta}}^+\|_F)C_0 r_n \\
& + \|\mathbf{\Delta}^-_{\bar{\mathcal{E}}_0^c}\|_1(\bar{\lambda}_1 - \alpha C_0') \\
& + \|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0^c}\|_1(\bar{\lambda}_2 - (1-\alpha)mC_0') - \|\mathbf{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1(\bar{\lambda}_1 + \alpha C_0') \\
& - \|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0}\|_1(\bar{\lambda}_2 + (1-\alpha)mC_0') \\
\geq\ & -(\alpha + (1-\alpha)\sqrt{m})\,\|\mathbf{\Delta}\|_F C_0 r_n - \|\mathbf{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1(\bar{\lambda}_1 + \alpha C_0') \\
& - \|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0}\|_1(\bar{\lambda}_2 + (1-\alpha)mC_0')
\end{aligned} \tag{45}$$

where, for the last inequality above, we used the fact that for λ_n as in (19), $\bar{\lambda}_1 - \alpha C_0' \geq 0$ and $\bar{\lambda}_2 - (1-\alpha)mC_0' \geq 0$, and $\|\mathbf{\Delta}^+\|_F \leq \|\mathbf{\Delta}\|_F$, $\|\tilde{\mathbf{\Delta}}^+\|_F \leq \|\mathbf{\Delta}\|_F$. By Cauchy-Schwartz inequality,

$$\|\mathbf{\Delta}^-_{\bar{\mathcal{E}}_0}\|_1 \leq \sqrt{m^2 s_{n0}}\,\|\mathbf{\Delta}^-_{\bar{\mathcal{E}}_0}\|_F \leq m\sqrt{s_{n0}}\,\|\mathbf{\Delta}\|_F\,, \tag{46}$$

$$\|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0}\|_1 \leq \sqrt{s_{n0}}\,\|\tilde{\mathbf{\Delta}}^-_{\mathcal{E}_0}\|_F \leq \sqrt{s_{n0}}\,\|\tilde{\mathbf{\Delta}}\|_F = \sqrt{s_{n0}}\,\|\mathbf{\Delta}\|_F\,. \tag{47}$$

Using (45)–(47) and $\alpha_m := (\alpha + (1-\alpha)\sqrt{m})$, we have

$$\begin{aligned}
A_2 + A_3 + A_4 \geq & -\Big[C_0'\left(1 + \alpha_m\sqrt{1 + p_n/(ms_{n0})}\right) + \bar{\lambda}_1 \\
& + (\bar{\lambda}_2/m)\Big]\, m\sqrt{s_{n0}}\,\|\mathbf{\Delta}\|_F \\
\geq & -\left[(\sqrt{m}+1)C_0 r_n + (\bar{\lambda}_1 + \frac{\bar{\lambda}_2}{m})m\sqrt{s_{n0}}\right]\,\|\mathbf{\Delta}\|_F \\
\geq & -C_2 C_0 r_n\,\|\mathbf{\Delta}\|_F
\end{aligned} \tag{48}$$

where in the last inequality above, we used the fact that for λ_n as in (19), $m\sqrt{s_{n0}}(\bar{\lambda}_1 + (\bar{\lambda}_2/m)) \leq C_1 C_0 r_n$, and $\alpha_m \leq \sqrt{m}$. Using (32), the bound (36) on A_1 and (48) on $A_2 + A_3 + A_4$, and $\|\mathbf{\Delta}\|_F = Mr_n$, we have with probability $> 1 - 1/(mp_n)^{\tau-2}$,

$$G(\mathbf{\Delta}) \geq\ \|\mathbf{\Delta}\|_F^2\left[\frac{1}{2(\beta_{\min}^{-1} + Mr_n)^2} - C_2\frac{C_0}{M}\right]\,. \tag{49}$$

For $n \geq N_2$, if we pick M as specified in (15), we obtain $Mr_n \leq Mr_{N_2} \leq \delta_1/\beta_{\min}$. Then

$$\frac{1}{2(\beta_{\min}^{-1} + Mr_n)^2} \geq \frac{\beta_{\min}^2}{2(1+\delta_1)^2} = \frac{(2C_2 + \delta_2)C_0}{2M} > C_2\frac{C_0}{M}\,,$$

implying $G(\mathbf{\Delta}) > 0$. This proves the desired result. ■

Appendix B
Proof of Theorem 2

Consider the (i,k)th element $\hat{\Omega}_{\lambda ik}$ of the sparse-group lasso estimate $\hat{\mathbf{\Omega}}_\lambda$. Since $\hat{\mathbf{\Omega}}_\lambda$ minimizes the penalized negative log-likelihood $L(\boldsymbol{X};\mathbf{\Omega})$ given by (7), we must have

$$\begin{aligned}
0 = \frac{\partial L(\boldsymbol{X};\mathbf{\Omega})}{\partial\hat{\Omega}_{ik}} &= \hat{\Sigma}_{ki} - [\hat{\mathbf{\Omega}}_\lambda^{-1}]_{ki} + \alpha\lambda_n\frac{\hat{\Omega}_{\lambda ik}}{|\hat{\Omega}_{\lambda ik}|} \\
&\quad + (1-\alpha)\lambda_n\frac{\hat{\Omega}_{\lambda ik}}{\|\hat{\mathbf{\Omega}}_\lambda^{(j\ell)}\|_F} \\
&= \hat{\Sigma}_{ik} - \check{\Sigma}_{\lambda ik} + \alpha\lambda_n\frac{\hat{\Omega}_{\lambda ik}}{|\hat{\Omega}_{\lambda ik}|} + (1-\alpha)\lambda_n\frac{\hat{\Omega}_{\lambda ik}}{\|\hat{\mathbf{\Omega}}_\lambda^{(j\ell)}\|_F} =: A
\end{aligned} \tag{50}$$

where

$$\check{\mathbf{\Sigma}}_\lambda := \hat{\mathbf{\Omega}}_\lambda^{-1},$$

Ω_{ik} is an element in $m \times m$ $\mathbf{\Omega}^{(j\ell)}$, we use the notation

$$\frac{\partial L(\boldsymbol{X};\mathbf{\Omega})}{\partial\hat{\Omega}_{ik}} = \frac{\partial L(\boldsymbol{X};\mathbf{\Omega})}{\partial\Omega_{ik}}\Big|_{\mathbf{\Omega}=\hat{\mathbf{\Omega}}_\lambda}$$

and assume that $\hat{\Omega}_{\lambda ik} \neq 0$.

To prove the desired result, the term $\alpha\lambda_n(\hat{\Omega}_{\lambda ik}/|\hat{\Omega}_{\lambda ik}|) + (1-\alpha)\lambda_n(\hat{\Omega}_{\lambda ik}/\|\hat{\mathbf{\Omega}}_\lambda{}^{(j\ell)}\|_F)$ on the right-side of (50) must dominate the term $\hat{\Sigma}_{ik} - \check{\Sigma}_{\lambda ik}$ whenever true value $\Omega_{0ij} = 0$. Then the sign of $\frac{\partial L(\boldsymbol{X};\mathbf{\Omega})}{\partial\hat{\Omega}_{ik}}$ in (50) is the same as $\text{sign}(\hat{\Omega}_{\lambda ik})$ with probability tending to one, which yields the desired result, as is shown in what follows. At the optimal solution, by the KKT conditions, one must have A in (50) equal to zero. Suppose that for $\{i,k\} \in \bar{\mathcal{E}}_0^c$, one has $\hat{\Omega}_{ik} \neq 0$ when $A = 0$. Suppose that $\hat{\Omega}_{\lambda ik} < 0$, implying that for some $\delta > 0$, $\hat{\Omega}_{\lambda ik} + \delta < 0$, since, by Theorem 1, $\hat{\Omega}_{\lambda ik}$ converges to $\Omega_{0ik} = 0$ for $\{i,k\} \in \bar{\mathcal{E}}_0^c$. Since $\hat{\Omega}_{\lambda ik}$ minimizes $L(\boldsymbol{X};\mathbf{\Omega})$, and $\frac{\partial L(\boldsymbol{X};\mathbf{\Omega})}{\partial\hat{\Omega}_{ik}} = 0$, we must have $I_1 := \frac{\partial L(\boldsymbol{X};\mathbf{\Omega})}{\partial(\hat{\Omega}_{ik}+\delta)} > 0$ for $\delta > 0$. If λ_n dominates $\hat{\Sigma}_{ik} - \check{\Sigma}_{\lambda ik}$ in (50), $I_1 > 0$ implies that $\hat{\Omega}_{\lambda ik} + \delta > 0$, contradicting the assumption that $\hat{\Omega}_{\lambda ik} + \delta < 0$. Therefore, $\hat{\Omega}_{\lambda ik} \not< 0$. We argue similarly that $\hat{\Omega}_{\lambda ik} \not> 0$. Therefore, $\hat{\Omega}_{\lambda ik} = 0$ for $\{i,k\} \in \bar{\mathcal{E}}_0^c$, with probability tending to one.

It remains to investigate the conditions under which λ_n dominates $\hat{\Sigma}_{ik} - \check{\Sigma}_{\lambda ik}$. Rewrite

$$\hat{\Sigma}_{ik} - \check{\Sigma}_{\lambda ik} = \underbrace{\hat{\Sigma}_{ik} - \Sigma_{0ik}}_{=:I_3} + \underbrace{\Sigma_{0ik} - \check{\Sigma}_{\lambda ik}}_{=:I_4}\,. \tag{51}$$

By Lemma 2, $\max_{i,k}|I_3| = \mathcal{O}_P(\sqrt{\frac{\ln(mp_n)}{n}})$. By [30, Lemma 1],

$$\begin{aligned}
|I_4| &\leq \|\mathbf{\Sigma}_0 - \check{\mathbf{\Sigma}}_\lambda\| = \|\check{\mathbf{\Sigma}}_\lambda(\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0)\mathbf{\Sigma}_0\| \\
&\leq \|\check{\mathbf{\Sigma}}_\lambda\| \cdot \|(\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0)\| \cdot \|\mathbf{\Sigma}_0\|\,.
\end{aligned} \tag{52}$$

By Assumption (A2), $\|\mathbf{\Sigma}_0\| = \mathcal{O}(1)$. Furthermore,

$$\begin{aligned}
\|\check{\mathbf{\Sigma}}_\lambda\| = \|\hat{\mathbf{\Omega}}_\lambda^{-1}\| &= \phi_{\min}^{-1}(\hat{\mathbf{\Omega}}_\lambda) \\
&\leq \Big(\phi_{\min}(\mathbf{\Omega}_0) + \phi_{\min}(\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0)\Big)^{-1} \\
&= (\mathcal{O}_P(1) + \mathcal{O}_P(\eta_n))^{-1} = \mathcal{O}_P(1)\,,
\end{aligned} \tag{53}$$

where we have used the fact that since $\|\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0\| = \mathcal{O}_P(\eta_n)$, $\phi_{\min}(\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0) \leq \|\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0\| = \mathcal{O}_P(\eta_n)$, and by Weyl's inequality, $\phi_{\min}(\boldsymbol{A}+\boldsymbol{B}) \geq \phi_{\min}(\boldsymbol{A}) + \phi_{\min}(\boldsymbol{B})$. Hence,

$$\max_{i,k} |I_4| = \mathcal{O}_P\left(\|\hat{\mathbf{\Omega}}_\lambda - \mathbf{\Omega}_0\|\right) = \mathcal{O}_P\left(\eta_n\right). \tag{54}$$

It then follows that

$$|\hat{\Sigma}_{ik} - \check{\Sigma}_{\lambda ik}| \leq |I_3| + |I_4| = \mathcal{O}_P\left(\sqrt{\frac{\ln(mp_n)}{n}} + \eta_n\right). \tag{55}$$

Suppose $\mathcal{O}(\lambda_n) = \sqrt{\ln(mp_n)/n} + \eta_n$. Then $\alpha\lambda_n(\hat{\Omega}_{\lambda ik}/|\hat{\Omega}_{\lambda ik}|) + (1-\alpha)\lambda_n(\hat{\Omega}_{\lambda ik}/\|\hat{\mathbf{\Omega}}_\lambda^{(j\ell)}\|_F)$ dominates $|\hat{\Sigma}_{ik} - \check{\Sigma}_{\lambda ik}|$ with probability tending to one. This completes the proof. ■

REFERENCES

[1] S. L. Lauritzen, *Graphical Models*. Oxford, UK: Oxford Univ. Press, 1996.

[2] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. New York, NY, USA: Wiley, 1990.

[3] P. Danaher, P. Wang, and D. M. Witten, "The joint graphical lasso for inverse covariance estimation across multiple classes," *J. Royal Stat. Soc., Series B (Methodological)*, vol. 76, pp. 373–397, 2014.

[4] U. Gather, M. Imhoff, and R. Fried, "Graphical models for multivariate time series from intensive care monitoring," *Statist. Med.*, vol. 21, no. 18, 2002.

[5] K. Khare, S.-Y. Oh, and B. Rajaratnam, "A convex pseudolikelihood framework for high dimensional partial correlation estimation with convergence guarantees," *J. Royal Stat. Soc., Series B (Methodological)*, vol. 77, no. 4, pp. 803–825, 2015.

[6] A. Ali, K. Khare, S.-Y. Oh, and B. Rajaratnam, "Generalized pseudolikelihood methods for inverse covariance estimation," in *Proc. Artif. Intell. Statist.*, Ft. Lauderdale, Fl, USA, Apr. 2017, pp. 280–288.

[7] A. Goldenberg, A. X. Zheng, S. E. Fienberg, and E. M. Airoldi, "A survey of statistical network models," *Found. Trends Mach. Learn.*, vol. 2, no. 2, pp. 129–233, 2009.

[8] M. Kolar, H. Liu, and E. P. Xing, "Markov network estimation from multi-attribute data," in *Proc. 30th Int. Conf. Mach. Learn.*, Atlanta, GA, 2013, pp. 73–81.

[9] M. Kolar, H. Liu, and E.P. Xing, "Graph estimation from multi-attribute data," *J. Mach. Learn. Res.*, vol. 15, no. 51, pp. 1713–1750, 2014.

[10] S. Ryali, T. Chen, K. Supekar, and V. Menon, "Estimation of functional connectivity in fMRI data using stability selection-based sparse partial correlation with elastic net penalty," *NeuroImage*, vol. 59, no. 4, pp. 3852–3861, 2012.

[11] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the Lasso," *Ann. Statist.*, vol. 34, no. 3, pp. 1436–1462, 2006.

[12] K. Mohan, P. London, M. Fazel, D. Witten, and S. I. Lee, "Node-based learning of multiple Gaussian graphical models," *J. Mach. Learn. Res.*, vol. 15, 2014.

[13] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical lasso," *Biostatistics*, vol. 9, no. 3, pp. 432–441, Jul. 2008.

[14] O. Banerjee, L.E. Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *J. Mach. Learn. Res.*, vol. 9, no. 15, pp. 485–516, 2008.

[15] J. Chiquet, G. Rigaill, and M. Sundquist, "A multiattribute Gaussian graphical model for inferring multiscale regulatory networks: An application in breast cancer," in *Gene Regulatory Networks. Methods in Molecular Biology*, G. Sanguinetti and V. Huynh-Thu, Eds., vol. 1883. Humana Press, New York, NY, 2019, pp. 143–160.

[16] E. Pavez and A. Ortega, "Generalized Laplacian precision matrix estimation for graph signal processing," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Shanghai, China, Mar. 2016, pp. 6350–6354.

[17] E. Pavez, H. E. Egilmez, and A. Ortega, "Learning graphs with monotone topology properties and multiple connected components," *IEEE Trans. Signal Process.*, vol. 66, no. 9, pp. 2399–2413, May 2018.

[18] X. Du and S. Ghosal, "Multivariate Gaussian network structure learning," *J. Stat. Planning Inference*, vol. 199, pp. 327–342, Mar. 2019.

[19] N. Katenka and E. D. Kolaczyk, "Multi-attribute networks and the impact of partial information on inference and characterization," *Ann. Appl. Statist.*, vol. 6, no. 3, pp. 1068–1094, 2012.

[20] G. Fracastoro, D. Thanou, and P. Frossard, "Graph transform optimization with application to image compression," *IEEE Trans. Image Process.*, vol. 29, pp. 419-432, 2020, doi: 10.1109/TIP.2019.2932853.

[21] E. Pavez, H. E. Egilmez, Y. Wang, and A. Ortega, "GTT: Graph template transforms with applications to image coding," in *Proc. Picture Coding Symp.*, Cairns, Australia, 2015, pp. 199–203.

[22] X. Dong, D. Thanou, M. Rabbat, and P. Frossard, "Learning graphs from data," *IEEE Signal Process. Mag.*, vol. 36, no. 3, pp. 44–63, May 2019.

[23] F. Hu, Z. Lu, H. Wong, and T.P. Yuen, "Analysis of air quality time series of Hong Kong with graphical modeling," *Environmetrics*, vol. 27, no. 3, pp. 169–181, May 2016.

[24] R. Dahlhaus, "Graphical interaction models for multivariate time series," *Metrika*, vol. 51, pp. 157–172, 2000.

[25] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group lasso," *J. Comput. Graph. Stat.*, vol. 22, pp. 231–245, 2013.

[26] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," 2010, *arXiv:1001.0736.*

[27] P. Ravikumar, M.J. Wainwright, G. Raskutti, and B. Yu, "High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence," *Electron. J. Statist.*, vol. 5, pp. 935–980, 2011.

[28] J. Peng, P. Wang, N. Zhou, and J. Zhu, "Partial correlation estimation by joint sparse regression models," *J. Amer. Stat. Assoc.*, vol. 104, no. 486, pp. 735–746, 2009.

[29] A. J. Rothman, P.J. Bickel, E. Levina, and J. Zhu, "Sparse permutation invariant covariance estimation," *Electron. J. Statist.*, vol. 2, pp. 494–515, 2008.

[30] C. Lam and J. Fan, "Sparsistency and rates of convergence in large covariance matrix estimation," *Ann. Statist.*, vol. 37, no. 6B, pp. 4254-4278, 2009.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.

[32] H. Zou, "The adaptive lasso and it oracle properties," *J. Amer. Stat. Assoc.*, vol. 101, no. 476, pp. 1418–1429, 2006.

[33] J. Fan, Y. Feng, and Y. Wu, "Network exploration via the adaptive lasso and SCAD penalties," *Ann. Appl. Statist.*, vol. 3, no. 2, pp. 521–541, 2009.

[34] J. Janková and S. van de Geer, "Inference in high-dimensional graphical models," in *Handbook of Graphical Models*, M. H. Maathuis, M. Drton, S. Lauritzen, and W. Wainwright, Eds. Boca Raton, FL, USA: CRC Press, 2018, pp. 325–348.

[35] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar, "QUIC: quadratic approximation for sparse inverse covariance estimation," *J. Mach. Learn. Res.*, vol. 15, no. 83, pp. 2911–2947, 2014.

[36] B. Rolfs, B. Rajaratnam, D. Guillot, I. Wong, and A. Maleki, "Iterative thresholding algorithm for sparse inverse covariance estimation," in *Proc. Adv. Neural Inf. Process. Syst.*, Red Hook, New York, NY, USA, vol. 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 1574–1582.

[37] G. Burghouts and J.-M. Geusebroek, "Material-specific adaptation of color invariant features," *Pattern Recognit. Lett.*, vol. 30, no. 3, pp. 306–313, 2009.

Jitendra K. Tugnait (Life Fellow, IEEE) received the B.Sc. (Hons.) degree in electronics and electrical communication engineering from Punjab Engineering College, Chandigarh, India in 1971, and the M.S. and E.E. degrees in electrical engineering from Syracuse University, Syracuse, NY, USA, in 1973 and 1974, respectively, and the Ph.D. degree in electrical engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 1978.

From 1978 to 1982, he was an Assistant Professor of electrical and computer engineering with the University of Iowa, Iowa City, IA, USA. From June 1982 to September 1989, he was with Long Range Research Division, Exxon Production Research Company, Houston, TX, USA. In September 1989, he joined as a Professor with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA , where he is currently the James B. Davis Professor. His current research interests include statistical signal processing, wireless physical and secure communications, and multiple target tracking.

Dr. Tugnait was an Associate Editor for the IEEE TRANSACTIONS ON AUTOMATIC CONTROL, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE SIGNAL PROCESSING LETTERS, and the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the Senior Area Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, and the Senior Editor of the IEEE WIRELESS COMMUNICATIONS LETTERS.