Journal of Machine Learning Research 15 (2014) 3297-3331 Submitted 9/13; Revised 5/14; Published 10/14

Learning Graphical Models With Hubs

Kean Ming Tan KEANMING@UW.EDU
Department of Biostatistics
University of Washington
Seattle WA, 98195

Palma London PALONDON@UW.EDU
Karthik Mohan KARNA@UW.EDU
Department of Electrical Engineering
University of Washington
Seattle WA, 98195

Su-In Lee SUINLEE@CS.WASHINGTON.EDU
Department of Computer Science and Engineering, Genome Sciences
University of Washington
Seattle WA, 98195

Maryam Fazel MFAZEL@UW.EDU
Department of Electrical Engineering
University of Washington
Seattle WA, 98195

Daniela Witten DWITTEN@UW.EDU
Department of Biostatistics
University of Washington
Seattle, WA 98195

Editor: Xiaotong Shen

Abstract

We consider the problem of learning a high-dimensional graphical model in which there are a few *hub* nodes that are *densely-connected* to many other nodes. Many authors have studied the use of an ℓ_1 penalty in order to learn a sparse graph in the high-dimensional setting. However, the ℓ_1 penalty implicitly assumes that each edge is equally likely and independent of all other edges. We propose a general framework to accommodate more realistic networks with hub nodes, using a convex formulation that involves a row-column overlap norm penalty. We apply this general framework to three widely-used probabilistic graphical models: the Gaussian graphical model, the covariance graph model, and the binary Ising model. An alternating direction method of multipliers algorithm is used to solve the corresponding convex optimization problems. On synthetic data, we demonstrate that our proposed framework outperforms competitors that do not explicitly model hub nodes. We illustrate our proposal on a webpage data set and a gene expression data set.

Keywords: Gaussian graphical model, covariance graph, binary network, *lasso*, hub, alternating direction method of multipliers

1. Introduction

Graphical models are used to model a wide variety of systems, such as gene regulatory networks and social interaction networks. A graph consists of a set of p nodes, each representing a variable, and a set of edges between pairs of nodes. The presence of an edge between two nodes indicates a relationship between the two variables. In this manuscript, we consider two types of graphs: conditional independence graphs and marginal independence graphs. In a conditional independence graph, an edge connects a pair of variables if and only if they are conditionally dependent—dependent conditional upon the other variables. In a marginal independence graph, two nodes are joined by an edge if and only if they are marginally dependent—dependent without conditioning on the other variables.

In recent years, many authors have studied the problem of learning a graphical model in the high-dimensional setting, in which the number of variables p is larger than the number of observations n. Let $\mathbf{X}$ be a $n \times p$ matrix, with rows $\mathbf{x}_1, \ldots, \mathbf{x}_n$. Throughout the rest of the text, we will focus on three specific types of graphical models:

1. A *Gaussian graphical model*, where $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. In this setting, $(\mathbf{\Sigma}^{-1})_{jj'} = 0$ for some $j \neq j'$ if and only if the jth and j'th variables are conditionally independent (Mardia et al., 1979); therefore, the sparsity pattern of $\mathbf{\Sigma}^{-1}$ determines the conditional independence graph.

2. A *Gaussian covariance graph model*, where $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. Then $\Sigma_{jj'} = 0$ for some $j \neq j'$ if and only if the jth and j'th variables are marginally independent. Therefore, the sparsity pattern of $\mathbf{\Sigma}$ determines the marginal independence graph.

3. A *binary Ising graphical model*, where $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are i.i.d. with density function

$$p(\mathbf{x}, \mathbf{\Theta}) = \frac{1}{Z(\mathbf{\Theta})} \exp\left[\sum_{j=1}^{p} \theta_{jj} x_j + \sum_{1 \leq j < j' \leq p} \theta_{jj'} x_j x_{j'}\right],$$

where $\mathbf{\Theta}$ is a $p \times p$ symmetric matrix, and $Z(\mathbf{\Theta})$ is the partition function, which ensures that the density sums to one. Here, $\mathbf{x}$ is a binary vector, and $\theta_{jj'} = 0$ if and only if the jth and j'th variables are conditionally independent. The sparsity pattern of $\mathbf{\Theta}$ determines the conditional independence graph.

To construct an interpretable graph when $p > n$, many authors have proposed applying an ℓ_1 penalty to the parameter encoding each edge, in order to encourage sparsity. For instance, such an approach is taken by Yuan and Lin (2007a), Friedman et al. (2007), Rothman et al. (2008), and Yuan (2008) in the Gaussian graphical model; El Karoui (2008), Bickel and Levina (2008), Rothman et al. (2009), Bien and Tibshirani (2011), Cai and Liu (2011), and Xue et al. (2012) in the covariance graph model; and Lee et al. (2007), Höfling and Tibshirani (2009), and Ravikumar et al. (2010) in the binary model.

However, applying an ℓ_1 penalty to each edge can be interpreted as placing an independent double-exponential prior on each edge. Consequently, such an approach implicitly assumes that each edge is equally likely and independent of all other edges; this corresponds to an Erdős-Rényi graph in which most nodes have approximately the same number of edges (Erdős and Rényi, 1959). This is unrealistic in many real-world networks, in which we believe that certain nodes (which, unfortunately, are not known *a priori*) have a lot more edges than other nodes. An example is the network of webpages in the World Wide Web, where a relatively small number of webpages are connected to many other webpages (Barabási and Albert, 1999). A number of authors have shown that real-world networks are *scale-free*, in the sense that the number of edges for each node follows a power-law distribution; examples include gene-regulatory networks, social networks, and networks of collaborations among scientists (among others, Barabási and Albert, 1999; Barabási, 2009; Liljeros et al., 2001; Jeong et al., 2001; Newman, 2000; Li et al., 2005). More recently, Hao et al. (2012) have shown that certain genes, referred to as *super hubs*, regulate hundreds of downstream genes in a gene regulatory network, resulting in far denser connections than are typically seen in a scale-free network.

In this paper, we refer to very densely-connected nodes, such as the "super hubs" considered in Hao et al. (2012), as *hubs.* When we refer to hubs, we have in mind nodes that are connected to a very substantial number of other nodes in the network—and in particular, we are referring to nodes that are much more densely-connected than even the most highly-connected node in a scale-free network. An example of a network containing hub nodes is shown in Figure 1.

Here we propose a convex penalty function for estimating graphs containing hubs. Our formulation simultaneously identifies the hubs and estimates the entire graph. The penalty function yields a convex optimization problem when combined with a convex loss function. We consider the application of this hub penalty function in modeling Gaussian graphical models, covariance graph models, and binary Ising models. Our formulation does not require that we know *a priori* which nodes in the network are hubs.

In related work, several authors have proposed methods to estimate a scale-free Gaussian graphical model (Liu and Ihler, 2011; Defazio and Caetano, 2012). However, those methods do not model hub nodes—the most highly-connected nodes that arise in a scale-free network are far less connected than the hubs that we consider in our formulation. Under a different framework, some authors proposed a screening-based procedure to identify hub nodes in the context of Gaussian graphical models (Hero and Rajaratnam, 2012; Firouzi and Hero, 2013). Our proposal outperforms such approaches when hub nodes are present (see discussion in Section 3.5.4).

In Figure 1, the performance of our proposed approach is shown in a toy example in the context of a Gaussian graphical model. We see that when the true network contains hub nodes (Figure 1(a)), our proposed approach (Figure 1(b)) is much better able to recover the network than is the graphical lasso (Figure 1(c)), a well-studied approach that applies an ℓ_1 penalty to each edge in the graph (Friedman et al., 2007).

We present the hub penalty function in Section 2. We then apply it to the Gaussian graphical model, the covariance graph model, and the binary Ising model in Sections 3, 4, and 5, respectively. In Section 6, we apply our approach to a webpage data set and a gene expression data set. We close with a discussion in Section 7.

2. The General Formulation

In this section, we present a general framework to accommodate network with hub nodes.

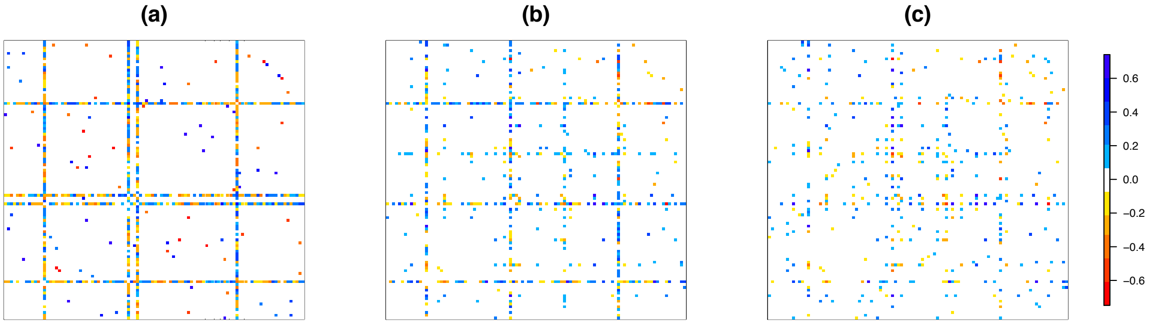

Figure 1: (a): Heatmap of the inverse covariance matrix in a toy example of a Gaussian graphical model with four hub nodes. White elements are zero and colored elements are non-zero in the inverse covariance matrix. Thus, colored elements correspond to edges in the graph. (b): Estimate from the *hub graphical lasso*, proposed in this paper. (c): Graphical lasso estimate.

2.1 The Hub Penalty Function

Let $\mathbf{X}$ be a $n \times p$ data matrix, $\boldsymbol{\Theta}$ a $p \times p$ symmetric matrix containing the parameters of interest, and $\ell(\mathbf{X}, \boldsymbol{\Theta})$ a loss function (assumed to be convex in $\boldsymbol{\Theta}$). In order to obtain a sparse and interpretable graph estimate, many authors have considered the problem

$$\underset{\boldsymbol{\Theta} \in \mathcal{S}}{\text{minimize}} \qquad \left\{ \ell(\mathbf{X}, \boldsymbol{\Theta}) + \lambda \|\boldsymbol{\Theta} - \text{diag}(\boldsymbol{\Theta})\|_1 \right\}, \tag{1}$$

where λ is a non-negative tuning parameter, $\mathcal{S}$ is some set depending on the loss function, and $\|\cdot\|_1$ is the sum of the absolute values of the matrix elements. For instance, in the case of a Gaussian graphical model, we could take $\ell(\mathbf{X}, \boldsymbol{\Theta}) = -\log\det \boldsymbol{\Theta} + \text{trace}(\mathbf{S}\boldsymbol{\Theta})$, the negative log-likelihood of the data, where $\mathbf{S}$ is the empirical covariance matrix and $\mathcal{S}$ is the set of $p \times p$ positive definite matrices. The solution to (1) can then be interpreted as an estimate of the inverse covariance matrix. The ℓ_1 penalty in (1) encourages zeros in the solution. But it typically does not yield an estimate that contains hubs.

In order to explicitly model hub nodes in a graph, we wish to replace the ℓ_1 penalty in (1) with a convex penalty that encourages a solution that can be decomposed as $\mathbf{Z}+\mathbf{V}+\mathbf{V}^T$, where $\mathbf{Z}$ is a sparse symmetric matrix, and $\mathbf{V}$ is a matrix whose columns are either entirely zero or almost entirely non-zero (see Figure 2). The sparse elements of $\mathbf{Z}$ represent edges between non-hub nodes, and the non-zero columns of $\mathbf{V}$ correspond to hub nodes. We achieve this goal via the *hub penalty function*, which takes the form

$$\text{P}(\boldsymbol{\Theta}) = \min_{\mathbf{V},\mathbf{Z}:\ \boldsymbol{\Theta}=\mathbf{V}+\mathbf{V}^T+\mathbf{Z}} \left\{ \lambda_1 \|\mathbf{Z} - \text{diag}(\mathbf{Z})\|_1 + \lambda_2 \|\mathbf{V} - \text{diag}(\mathbf{V})\|_1 + \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V} - \text{diag}(\mathbf{V}))_j\|_q \right\}. \tag{2}$$

Here λ_1, λ_2, and λ_3 are nonnegative tuning parameters. Sparsity in $\mathbf{Z}$ is encouraged via the ℓ_1 penalty on its off-diagonal elements, and is controlled by the value of λ_1. The ℓ_1 and ℓ_1/ℓ_q norms on the columns of $\mathbf{V}$ induce group sparsity when $q = 2$ (Yuan and Lin, 2007b; Simon et al., 2013); λ_3 controls the selection of hub nodes, and λ_2 controls the sparsity of

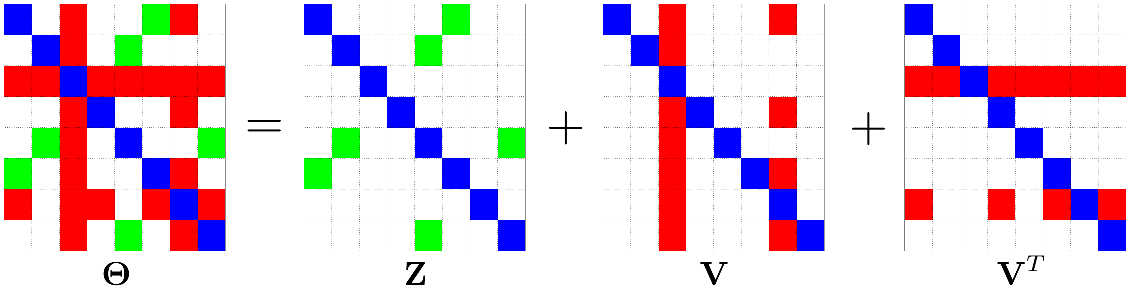

Figure 2: Decomposition of a symmetric matrix $\mathbf{\Theta}$ into $\mathbf{Z}+\mathbf{V}+\mathbf{V}^T$, where $\mathbf{Z}$ is sparse, and most columns of $\mathbf{V}$ are entirely zero. Blue, white, green, and red elements are diagonal, zero, non-zero in $\mathbf{Z}$, and non-zero due to two hubs in $\mathbf{V}$, respectively.

each hub node's connections to other nodes. The convex penalty (2) can be combined with $\ell(\mathbf{X},\mathbf{\Theta})$ to yield the convex optimization problem

$$\underset{\mathbf{\Theta}\in\mathcal{S},\mathbf{V},\mathbf{Z}}{\text{minimize}} \quad \left\{\ell(\mathbf{X},\mathbf{\Theta}) + \lambda_1\|\mathbf{Z}-\text{diag}(\mathbf{Z})\|_1 + \lambda_2\|\mathbf{V}-\text{diag}(\mathbf{V})\|_1 \right.$$
$$\left. +\lambda_3\sum_{j=1}^{p}\|(\mathbf{V}-\text{diag}(\mathbf{V}))_j\|_q \right\} \text{ subject to } \mathbf{\Theta}=\mathbf{V}+\mathbf{V}^T+\mathbf{Z}, \tag{3}$$

where the set $\mathcal{S}$ depends on the loss function $\ell(\mathbf{X},\mathbf{\Theta})$.

Note that when $\lambda_2\to\infty$ or $\lambda_3\to\infty$, then (3) reduces to (1). In this paper, we take $q=2$, which leads to estimation of a network containing dense hub nodes. Other values of q such as $q=\infty$ are also possible (see, e.g., Mohan et al., 2014). We note that the hub penalty function is closely related to recent work on overlapping group lasso penalties in the context of learning multiple sparse precision matrices (Mohan et al., 2014).

2.2 Algorithm

In order to solve (3) with $q=2$, we use an *alternating direction method of multipliers* (ADMM) algorithm (see, e.g., Eckstein and Bertsekas, 1992; Boyd et al., 2010; Eckstein, 2012). ADMM is an attractive algorithm for this problem, as it allows us to decouple some of the terms in (3) that are difficult to optimize jointly. In order to develop an ADMM algorithm for (3) with guaranteed convergence, we reformulate it as a consensus problem, as in Ma et al. (2013). The convergence of the algorithm to the optimal solution follows from classical results (see, e.g., the review papers Boyd et al., 2010; Eckstein, 2012).

In greater detail, we let $\mathbf{B}=(\mathbf{\Theta},\mathbf{V},\mathbf{Z})$, $\tilde{\mathbf{B}}=(\tilde{\mathbf{\Theta}},\tilde{\mathbf{V}},\tilde{\mathbf{Z}})$,

$$f(\mathbf{B})=\ell(\mathbf{X},\mathbf{\Theta})+\lambda_1\|\mathbf{Z}-\text{diag}(\mathbf{Z})\|_1+\lambda_2\|\mathbf{V}-\text{diag}(\mathbf{V})\|_1+\lambda_3\sum_{j=1}^{p}\|(\mathbf{V}-\text{diag}(\mathbf{V}))\|_2,$$

Algorithm 1 ADMM Algorithm for Solving (3).

```
1. Initialize the parameters:
    (a) primal variables $\mathbf{\Theta}, \mathbf{V}, \mathbf{Z}, \tilde{\mathbf{\Theta}}, \tilde{\mathbf{V}}$, and $\tilde{\mathbf{Z}}$ to the $p \times p$ identity matrix.
    (b) dual variables $\mathbf{W}_1, \mathbf{W}_2$, and $\mathbf{W}_3$ to the $p \times p$ zero matrix.
    (c) constants $\rho > 0$ and $\tau > 0$.

2. Iterate until the stopping criterion $\frac{\|\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1}\|_F^2}{\|\mathbf{\Theta}_{t-1}\|_F^2} \leq \tau$ is met, where $\mathbf{\Theta}_t$ is the value of $\mathbf{\Theta}$ obtained at the $t$th iteration:
    (a) Update $\mathbf{\Theta}, \mathbf{V}, \mathbf{Z}$:
        i. $\mathbf{\Theta} = \underset{\mathbf{\Theta} \in \mathcal{S}}{\arg\min} \left\{ \ell(\mathbf{X}, \mathbf{\Theta}) + \frac{\rho}{2} \|\mathbf{\Theta} - \tilde{\mathbf{\Theta}} + \mathbf{W}_1\|_F^2 \right\}$.
        ii. $\mathbf{Z} = S(\tilde{\mathbf{Z}} - \mathbf{W}_3, \frac{\lambda_1}{\rho})$, $\mathrm{diag}(\mathbf{Z}) = \mathrm{diag}(\tilde{\mathbf{Z}} - \mathbf{W}_3)$. Here $S$ denotes the soft-thresholding operator, applied element-wise to a matrix: $S(A_{ij}, b) = \mathrm{sign}(A_{ij}) \max(|A_{ij}| - b, 0)$.
        iii. $\mathbf{C} = \tilde{\mathbf{V}} - \mathbf{W}_2 - \mathrm{diag}(\tilde{\mathbf{V}} - \mathbf{W}_2)$.
        iv. $\mathbf{V}_j = \max\left(1 - \frac{\lambda_3}{\rho \|S(\mathbf{C}_j, \lambda_2/\rho)\|_2}, 0\right) \cdot S(\mathbf{C}_j, \lambda_2/\rho)$ for $j = 1, \ldots, p$.
        v. $\mathrm{diag}(\mathbf{V}) = \mathrm{diag}(\tilde{\mathbf{V}} - \mathbf{W}_2)$.
    (b) Update $\tilde{\mathbf{\Theta}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Z}}$:
        i. $\mathbf{\Gamma} = \frac{\rho}{6} \left[ (\mathbf{\Theta} + \mathbf{W}_1) - (\mathbf{V} + \mathbf{W}_2) - (\mathbf{V} + \mathbf{W}_2)^T - (\mathbf{Z} + \mathbf{W}_3) \right]$.
        ii. $\tilde{\mathbf{\Theta}} = \mathbf{\Theta} + \mathbf{W}_1 - \frac{1}{\rho}\mathbf{\Gamma}$;   iii. $\tilde{\mathbf{V}} = \frac{1}{\rho}(\mathbf{\Gamma} + \mathbf{\Gamma}^T) + \mathbf{V} + \mathbf{W}_2$;   iv. $\tilde{\mathbf{Z}} = \frac{1}{\rho}\mathbf{\Gamma} + \mathbf{Z} + \mathbf{W}_3$.
    (c) Update $\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3$:
        i. $\mathbf{W}_1 = \mathbf{W}_1 + \mathbf{\Theta} - \tilde{\mathbf{\Theta}}$;   ii. $\mathbf{W}_2 = \mathbf{W}_2 + \mathbf{V} - \tilde{\mathbf{V}}$;   iii. $\mathbf{W}_3 = \mathbf{W}_3 + \mathbf{Z} - \tilde{\mathbf{Z}}$.
```

and

$$g(\tilde{\mathbf{B}}) = \begin{cases} 0 & \text{if } \tilde{\mathbf{\Theta}} = \tilde{\mathbf{V}} + \tilde{\mathbf{V}}^T + \tilde{\mathbf{Z}} \\ \infty & \text{otherwise.} \end{cases}$$

Then, we can rewrite (3) as

$$\underset{\mathbf{B},\tilde{\mathbf{B}}}{\text{minimize}} \ \left\{ f(\mathbf{B}) + g(\tilde{\mathbf{B}}) \right\} \qquad \text{subject to } \mathbf{B} = \tilde{\mathbf{B}}. \tag{4}$$

The scaled augmented Lagrangian for (4) takes the form

$$\begin{aligned} L(\mathbf{B}, \tilde{\mathbf{B}}, \mathbf{W}) = {} & \ell(\mathbf{X}, \mathbf{\Theta}) + \lambda_1 \|\mathbf{Z} - \text{diag}(\mathbf{Z})\|_1 + \lambda_2 \|\mathbf{V} - \text{diag}(\mathbf{V})\|_1 \\ & + \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V} - \text{diag}(\mathbf{V}))_j\|_2 + g(\tilde{\mathbf{B}}) + \frac{\rho}{2} \|\mathbf{B} - \tilde{\mathbf{B}} + \mathbf{W}\|_F^2, \end{aligned}$$

where $\mathbf{B}$ and $\tilde{\mathbf{B}}$ are the primal variables, and $\mathbf{W} = (\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3)$ is the dual variable. Note that the scaled augmented Lagrangian can be derived from the usual Lagrangian by adding a quadratic term and completing the square (Boyd et al., 2010).

A general algorithm for solving (3) is provided in Algorithm 1. The derivation is in Appendix A. Note that only the update for $\mathbf{\Theta}$ (Step 2(a)i) depends on the form of the convex loss function $\ell(\mathbf{X}, \mathbf{\Theta})$. In the following sections, we consider special cases of (3) that lead to estimation of Gaussian graphical models, covariance graph models, and binary networks with hub nodes.

3. The Hub Graphical Lasso

Assume that $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. The well-known *graphical lasso* problem (see, e.g., Friedman et al., 2007) takes the form of (1) with $\ell(\mathbf{X}, \mathbf{\Theta}) = -\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S}\mathbf{\Theta})$, and $\mathbf{S}$ the empirical covariance matrix of $\mathbf{X}$:

$$\underset{\mathbf{\Theta} \in \mathcal{S}}{\text{minimize}} \quad \left\{ -\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S}\mathbf{\Theta}) + \lambda \sum_{j \neq j'} |\Theta_{jj'}| \right\}, \tag{5}$$

where $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} \succ 0 \text{ and } \mathbf{\Theta} = \mathbf{\Theta}^T\}$. The solution to this optimization problem serves as an estimate for $\mathbf{\Sigma}^{-1}$. We now use the hub penalty function to extend the graphical lasso in order to accommodate hub nodes.

3.1 Formulation and Algorithm

We propose the *hub graphical lasso* (HGL) optimization problem, which takes the form

$$\underset{\mathbf{\Theta} \in \mathcal{S}}{\text{minimize}} \qquad \{-\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S}\mathbf{\Theta}) + \text{P}(\mathbf{\Theta})\}. \tag{6}$$

Again, $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} \succ 0 \text{ and } \mathbf{\Theta} = \mathbf{\Theta}^T\}$. It encourages a solution that contains hub nodes, as well as edges that connect non-hubs (Figure 1). Problem (6) can be solved using Algorithm 1. The update for $\mathbf{\Theta}$ in Algorithm 1 (Step 2(a)i) can be derived by minimizing

$$-\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S}\mathbf{\Theta}) + \frac{\rho}{2}\|\mathbf{\Theta} - \tilde{\mathbf{\Theta}} + \mathbf{W}_1\|_F^2 \tag{7}$$

with respect to $\mathbf{\Theta}$ (note that the constraint $\mathbf{\Theta} \in \mathcal{S}$ in (6) is treated as an implicit constraint, due to the domain of definition of the log det function). This can be shown to have the solution

$$\mathbf{\Theta} = \frac{1}{2}\mathbf{U}\left(\mathbf{D} + \sqrt{\mathbf{D}^2 + \frac{4}{\rho}\mathbf{I}}\right)\mathbf{U}^T,$$

where $\mathbf{UDU}^T$ denotes the eigen-decomposition of $\tilde{\mathbf{\Theta}} - \mathbf{W}_1 - \frac{1}{\rho}\mathbf{S}$.

The complexity of the ADMM algorithm for HGL is $O(p^3)$ per iteration; this is the complexity of the eigen-decomposition for updating $\mathbf{\Theta}$. We now briefly compare the computational time for the ADMM algorithm for solving (6) to that of an interior point method (using the solver `Sedumi` called from `cvx`). On a 1.86 GHz Intel Core 2 Duo machine, the interior point method takes ~ 3 minutes, while ADMM takes only 1 second, on a data set with $p = 30$. We present a more extensive run time study for the ADMM algorithm for HGL in Appendix E.

3.2 Conditions for HGL Solution to be Block Diagonal

In order to reduce computations for solving the HGL problem, we now present a necessary condition and a sufficient condition for the HGL solution to be block diagonal, subject to some permutation of the rows and columns. The conditions depend only on the tuning parameters λ_1, λ_2, and λ_3. These conditions build upon similar results in the context of Gaussian graphical models from the recent literature (see, e.g., Witten et al., 2011; Mazumder and Hastie, 2012; Yang et al., 2012b; Danaher et al., 2014; Mohan et al., 2014). Let $C_1, C_2, \ldots, C_K$ denote a partition of the p features.

Theorem 1 *A sufficient condition for the HGL solution to be block diagonal with blocks given by $C_1, C_2, \ldots, C_K$ is that* $\min\left\{\lambda_1, \frac{\lambda_2}{2}\right\} > |S_{jj'}|$ *for all $j \in C_k, j' \in C_{k'}, k \neq k'$.*

Theorem 2 *A necessary condition for the HGL solution to be block diagonal with blocks given by $C_1, C_2, \ldots, C_K$ is that* $\min\left\{\lambda_1, \frac{\lambda_2+\lambda_3}{2}\right\} > |S_{jj'}|$ *for all $j \in C_k, j' \in C_{k'}, k \neq k'$.*

Theorem 1 implies that one can screen the empirical covariance matrix $\mathbf{S}$ to check if the HGL solution is block diagonal (using standard algorithms for identifying the connected components of an undirected graph; see, e.g., Tarjan, 1972). Suppose that the HGL solution is block diagonal with K blocks, containing $p_1, \ldots, p_K$ features, and $\sum_{k=1}^K p_k = p$. Then, one can simply solve the HGL problem on the features within each block separately. Recall that the bottleneck of the HGL algorithm is the eigen-decomposition for updating $\mathbf{\Theta}$. The block diagonal condition leads to massive computational speed-ups for implementing the HGL algorithm: instead of computing an eigen-decomposition for a $p \times p$ matrix in each iteration of the HGL algorithm, we compute the eigen-decomposition of K matrices of dimensions $p_1 \times p_1, \ldots, p_K \times p_K$. The computational complexity per-iteration is reduced from $O(p^3)$ to $\sum_{k=1}^K O(p_k^3)$.

We illustrate the reduction in computational time due to these results in an example with $p = 500$. Without exploiting Theorem 1, the ADMM algorithm for HGL (with a particular value of λ) takes 159 seconds; in contrast, it takes only 22 seconds when Theorem 1 is applied. The estimated precision matrix has 107 connected components, the largest of which contains 212 nodes.

3.3 Some Properties of HGL

We now present several properties of the HGL optimization problem (6), which can be used to provide guidance on the suitable range for the tuning parameters λ_1, λ_2, and λ_3. In what follows, $\mathbf{Z}^*$ and $\mathbf{V}^*$ denote the optimal solutions for $\mathbf{Z}$ and $\mathbf{V}$ in (6). Let $\frac{1}{s} + \frac{1}{q} = 1$ (recall that q appears in Equation 2).

Lemma 3 *A sufficient condition for* $\mathbf{Z}^*$ *to be a diagonal matrix is that* $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$.

Lemma 4 *A sufficient condition for* $\mathbf{V}^*$ *to be a diagonal matrix is that* $\lambda_1 < \frac{\lambda_2}{2} + \frac{\lambda_3}{2(p-1)^{1/s}}$.

Corollary 5 *A necessary condition for both* $\mathbf{V}^*$ *and* $\mathbf{Z}^*$ *to be non-diagonal matrices is that* $\frac{\lambda_2}{2} + \frac{\lambda_3}{2(p-1)^{1/s}} \leq \lambda_1 \leq \frac{\lambda_2+\lambda_3}{2}$.

Furthermore, (6) reduces to the graphical lasso problem (5) under a simple condition.

Lemma 6 *If* $q = 1$*, then (6) reduces to (5) with tuning parameter* $\min\left\{\lambda_1, \frac{\lambda_2+\lambda_3}{2}\right\}$.

Note also that when $\lambda_2 \to \infty$ or $\lambda_3 \to \infty$, (6) reduces to (5) with tuning parameter λ_1. However, throughout the rest of this paper, we assume that $q = 2$, and λ_2 and λ_3 are finite.

The solution $\hat{\mathbf{\Theta}}$ of (6) is unique, since (6) is a strictly convex problem. We now consider the question of whether the decomposition $\hat{\mathbf{\Theta}} = \hat{\mathbf{V}} + \hat{\mathbf{V}}^T + \hat{\mathbf{Z}}$ is unique. We see that the decomposition is unique in a certain regime of the tuning parameters. For instance, according to Lemma 3, when $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$, $\hat{\mathbf{Z}}$ is a diagonal matrix and hence $\hat{\mathbf{V}}$ is unique. Similarly, according to Lemma 4, when $\lambda_1 < \frac{\lambda_2}{2} + \frac{\lambda_3}{2(p-1)^{1/s}}$, $\hat{\mathbf{V}}$ is a diagonal matrix and hence $\hat{\mathbf{Z}}$ is unique. Studying more general conditions on $\mathbf{S}$ and on λ_1, λ_2, and λ_3 such that the decomposition is guaranteed to be unique is a challenging problem and is outside of the scope of this paper.

3.4 Tuning Parameter Selection

In this section, we propose a *Bayesian information criterion* (BIC)-type quantity for tuning parameter selection in (6). Recall from Section 2 that the hub penalty function (2) decomposes the parameter of interest into the sum of three matrices, $\mathbf{\Theta} = \mathbf{Z} + \mathbf{V} + \mathbf{V}^T$, and places an ℓ_1 penalty on $\mathbf{Z}$, and an ℓ_1/ℓ_2 penalty on $\mathbf{V}$.

For the graphical lasso problem in (5), many authors have proposed to select the tuning parameter λ such that $\hat{\mathbf{\Theta}}$ minimizes the following quantity:

$$-n \cdot \log\det(\hat{\mathbf{\Theta}}) + n \cdot \mathrm{trace}(\mathbf{S}\hat{\mathbf{\Theta}}) + \log(n) \cdot |\hat{\mathbf{\Theta}}|,$$

where $|\hat{\mathbf{\Theta}}|$ is the cardinality of $\hat{\mathbf{\Theta}}$, that is, the number of unique non-zeros in $\hat{\mathbf{\Theta}}$ (see, e.g., Yuan and Lin, 2007a).[1]

[1] The term $\log(n) \cdot |\hat{\mathbf{\Theta}}|$ is motivated by the fact that the degrees of freedom for an estimate involving the ℓ_1 penalty can be approximated by the cardinality of the estimated parameter (Zou et al., 2007).

Using a similar idea, we propose the following BIC-type quantity for selecting the set of tuning parameters $(\lambda_1, \lambda_2, \lambda_3)$ for (6):

$$\text{BIC}(\hat{\mathbf{\Theta}}, \hat{\mathbf{V}}, \hat{\mathbf{Z}}) = -n \cdot \log\det(\hat{\mathbf{\Theta}}) + n \cdot \text{trace}(\mathbf{S}\hat{\mathbf{\Theta}}) + \log(n) \cdot |\hat{\mathbf{Z}}| + \log(n) \cdot \left(\nu + c \cdot [|\hat{\mathbf{V}}| - \nu]\right),$$

where ν is the number of estimated hub nodes, that is, $\nu = \sum_{j=1}^{p} 1_{\{\|\hat{\mathbf{V}}_j\|_0 > 0\}}$, c is a constant between zero and one, and $|\hat{\mathbf{Z}}|$ and $|\hat{\mathbf{V}}|$ are the cardinalities (the number of unique non-zeros) of $\hat{\mathbf{Z}}$ and $\hat{\mathbf{V}}$, respectively.[2] We select the set of tuning parameters $(\lambda_1, \lambda_2, \lambda_3)$ for which the quantity $\text{BIC}(\hat{\mathbf{\Theta}}, \hat{\mathbf{V}}, \hat{\mathbf{Z}})$ is minimized. Note that when the constant c is small, $\text{BIC}(\hat{\mathbf{\Theta}}, \hat{\mathbf{V}}, \hat{\mathbf{Z}})$ will favor more hub nodes in $\hat{\mathbf{V}}$. In this manuscript, we take $c = 0.2$.

3.5 Simulation Study

In this section, we compare HGL to two sets of proposals: proposals that learn an Erdős-Rényi Gaussian graphical model, and proposals that learn a Gaussian graphical model in which some nodes are highly-connected.

3.5.1 Notation and Measures of Performance

We start by defining some notation. Let $\hat{\mathbf{\Theta}}$ be the estimate of $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ from a given proposal, and let $\hat{\mathbf{\Theta}}_j$ be its jth column. Let $\mathcal{H}$ denote the set of indices of the hub nodes in $\mathbf{\Theta}$ (that is, this is the set of true hub nodes in the graph), and let $|\mathcal{H}|$ denote the cardinality of the set. In addition, let $\hat{\mathcal{H}}_r$ be the set of *estimated hub nodes*: the set of nodes in $\hat{\mathbf{\Theta}}$ that are among the $|\mathcal{H}|$ most highly-connected nodes, and that have at least r edges. The values chosen for $|\mathcal{H}|$ and r depend on the simulation set-up, and will be specified in each simulation study.

We now define several measures of performance that will be used to evaluate the various methods.

- Number of correctly estimated edges: $\sum_{j<j'} \left(1_{\{|\hat{\Theta}_{jj'}| > 10^{-5} \text{ and } |\Theta_{jj'}| \neq 0\}}\right)$.

- Proportion of correctly estimated hub edges:

$$\frac{\sum_{j \in \mathcal{H}, j' \neq j} \left(1_{\{|\hat{\Theta}_{jj'}| > 10^{-5} \text{ and } |\Theta_{jj'}| \neq 0\}}\right)}{\sum_{j \in \mathcal{H}, j' \neq j} \left(1_{\{|\Theta_{jj'}| \neq 0\}}\right)}.$$

- Proportion of correctly estimated hub nodes:$\frac{|\hat{\mathcal{H}}_r \cap \mathcal{H}|}{|\mathcal{H}|}$.

- Sum of squared errors: $\sum_{j<j'} \left(\hat{\Theta}_{jj'} - \Theta_{jj'}\right)^2$.

[2] The term $\log(n) \cdot |\hat{\mathbf{Z}}|$ is motivated by the degrees of freedom from the ℓ_1 penalty, and the term $\log(n) \cdot \left(\nu + c \cdot [|\hat{\mathbf{V}}| - \nu]\right)$ is motivated by an approximation of the degrees of freedom of the ℓ_2 penalty proposed in Yuan and Lin (2007b).

3.5.2 Data Generation

We consider three set-ups for generating a $p \times p$ adjacency matrix $\mathbf{A}$.

I - Network with hub nodes: for all $i < j$, we set $A_{ij} = 1$ with probability 0.02, and zero otherwise. We then set A_{ji} equal to A_{ij}. Next, we randomly select $|\mathcal{H}|$ hub nodes and set the elements of the corresponding rows and columns of $\mathbf{A}$ to equal one with probability 0.7 and zero otherwise.

II - Network with two connected components and hub nodes: the adjacency matrix is generated as $\mathbf{A} = \begin{pmatrix} \mathbf{A}_1 & 0 \\ 0 & \mathbf{A}_2 \end{pmatrix}$, with $\mathbf{A}_1$ and $\mathbf{A}_2$ as in Set-up I, each with $|\mathcal{H}|/2$ hub nodes.

III - Scale-free network:[3] the probability that a given node has k edges is proportional to $k^{-\alpha}$. Barabási and Albert (1999) observed that many real-world networks have $\alpha \in [2.1, 4]$; we took $\alpha = 2.5$. Note that there is no natural notion of hub nodes in a scale-free network. While some nodes in a scale-free network have more edges than one would expect in an Erdős-Rényi graph, there is no clear distinction between "hub" and "non-hub" nodes, unlike in Set-ups I and II. In our simulation settings, we consider any node that is connected to more than 5% of all other nodes to be a hub node.[4]

We then use the adjacency matrix $\mathbf{A}$ to create a matrix $\mathbf{E}$, as

$$E_{ij} \overset{\text{i.i.d.}}{\sim} \begin{cases} 0 & \text{if } A_{ij} = 0 \\ \text{Unif}([-0.75, -0.25] \cup [0.25, 0.75]) & \text{otherwise,} \end{cases}$$

and set $\bar{\mathbf{E}} = \frac{1}{2}(\mathbf{E} + \mathbf{E}^T)$. Given the matrix $\bar{\mathbf{E}}$, we set $\mathbf{\Sigma}^{-1}$ equal to $\bar{\mathbf{E}} + (0.1 - \Lambda_{\min}(\bar{\mathbf{E}}))\mathbf{I}$, where $\Lambda_{\min}(\bar{\mathbf{E}})$ is the smallest eigenvalue of $\bar{\mathbf{E}}$. We generate the data matrix $\mathbf{X}$ according to $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. Then, variables are standardized to have standard deviation one.

3.5.3 Comparison to Graphical Lasso and Neighbourhood Selection

In this subsection, we compare the performance of HGL to two proposals that learn a sparse Gaussian graphical model.

- The graphical lasso (5), implemented using the `R` package `glasso`.
- The neighborhood selection approach of Meinshausen and Bühlmann (2006), implemented using the `R` package `glasso`. This approach involves performing p ℓ_1-penalized regression problems, each of which involves regressing one feature onto the others.

[3] Recall that our proposal is not intended for estimating a scale-free network.

[4] The cutoff threshold of 5% is chosen in order to capture the most highly-connected nodes in the scale-free network. In our simulation study, around three nodes are connected to at least $0.05 \times p$ other nodes in the network. The precise choice of cut-off threshold has little effect on the results obtained in the figures that follow.

We consider the three simulation set-ups described in the previous section with $n = 1000$, $p = 1500$, and $|\mathcal{H}| = 30$ hub nodes in Set-ups I and II. Figure 3 displays the results, averaged over 100 simulated data sets. Note that the sum of squared errors is not computed for Meinshausen and Bühlmann (2006), since it does not directly yield an estimate of $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$.

HGL has three tuning parameters. To obtain the curves shown in Figure 3, we fixed $\lambda_1 = 0.4$, considered three values of λ_3 (each shown in a different color in Figure 3), and used a fine grid of values of λ_2. The solid black circle in Figure 3 corresponds to the set of tuning parameters $(\lambda_1, \lambda_2, \lambda_3)$ for which the BIC as defined in Section 3.4 is minimized. The graphical lasso and Meinshausen and Bühlmann (2006) each involves one tuning parameter; we applied them using a fine grid of the tuning parameter to obtain the curves shown in Figure 3.

Results for Set-up I are displayed in Figures 3-I(a) through 3-I(d), where we calculate the proportion of correctly estimated hub nodes as defined in Section 3.5.1 with $r = 300$. Since this simulation set-up exactly matches the assumptions of HGL, it is not surprising that HGL outperforms the other methods. In particular, HGL is able to identify most of the hub nodes when the number of estimated edges is approximately equal to the true number of edges. We see similar results for Set-up II in Figures 3-II(a) through 3-II(d), where the proportion of correctly estimated hub nodes is as defined in Section 3.5.1 with $r = 150$.

In Set-up III, recall that we define a node that is connected to at least 5% of all nodes to be a hub. The proportion of correctly estimated hub nodes is then as defined in Section 3.5.1 with $r = 0.05 \times p$. The results are presented in Figures 3-III(a) through 3-III(d). In this set-up, only approximately three of the nodes (on average) have more than 50 edges, and the hub nodes are not as highly-connected as in Set-up I or Set-up II. Nonetheless, HGL outperforms the graphical lasso and Meinshausen and Bühlmann (2006).

Finally, we see from Figure 3 that the set of tuning parameters $(\lambda_1, \lambda_2, \lambda_3)$ selected using BIC performs reasonably well. In particular, the graphical lasso solution always has BIC larger than HGL, and hence, is not selected.

3.5.4 Comparison to Additional Proposals

In this subsection, we compare the performance of HGL to three additional proposals:

- The partial correlation screening procedure of Hero and Rajaratnam (2012). The elements of the partial correlation matrix (computed using a pseudo-inverse when $p > n$) are thresholded based on their absolute value, and a hub node is declared if the number of nonzero elements in the corresponding column of the thresholded partial correlation matrix is sufficiently large. Note that the purpose of Hero and Rajaratnam (2012) is to screen for hub nodes, rather than to estimate the individual edges in the network.

- The scale-free network estimation procedure of Liu and Ihler (2011). This is the solution to the non-convex optimization problem

$$\underset{\mathbf{\Theta} \in \mathcal{S}}{\text{minimize}} \quad \left\{ -\log\det\mathbf{\Theta} + \text{trace}(\mathbf{S\Theta}) + \alpha \sum_{j=1}^{p} \log(\|\theta_{\setminus j}\|_1 + \epsilon_j) + \sum_{j=1}^{p} \beta_j |\theta_{jj}| \right\}, \tag{8}$$

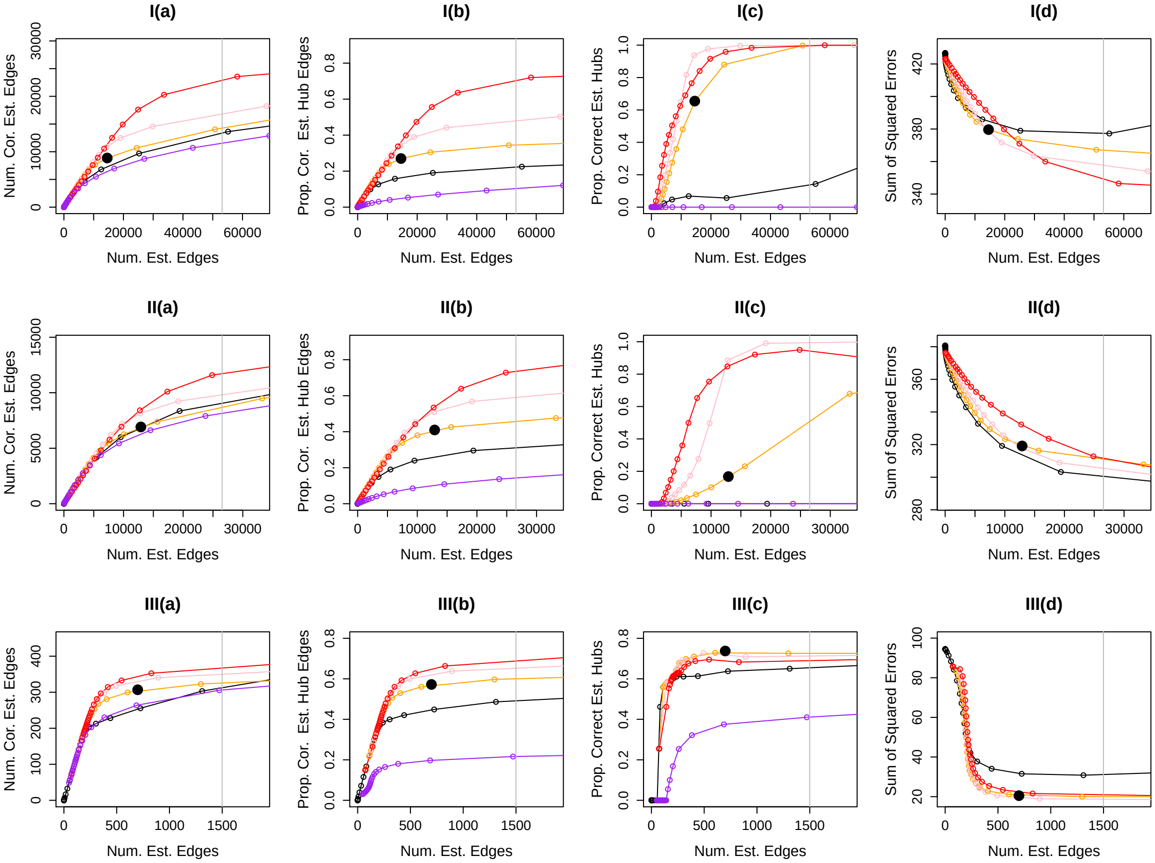

Figure 3: Simulation for Gaussian graphical model. Row I: Results for Set-up I. Row II: Results for Set-up II. Row III: Results for Set-up III. The results are for $n = 1000$ and $p = 1500$. In each panel, the x-axis displays the number of estimated edges, and the vertical gray line is the number of edges in the true network. The y-axes are as follows: Column (a): Number of correctly estimated edges; Column (b): Proportion of correctly estimated hub edges; Column (c): Proportion of correctly estimated hub nodes; Column (d): Sum of squared errors. The black solid circles are the results for HGL based on tuning parameters selected using the BIC-type criterion defined in Section 3.4. Colored lines correspond to the graphical lasso (Friedman et al., 2007) (—); HGL with $\lambda_3 = 0.5$ (—), $\lambda_3 = 1$ (—), and $\lambda_3 = 2$ (—); neighborhood selection (Meinshausen and Bühlmann, 2006) (—).

where $\theta_{\setminus j} = \{\theta_{jj'} | j' \neq j\}$, and ϵ_j, β_j, and α are tuning parameters. Here, $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} \succ 0 \text{ and } \mathbf{\Theta} = \mathbf{\Theta}^T\}$.

- Sparse partial correlation estimation procedure of Peng et al. (2009), implemented using the `R` package `space`. This is an extension of the neighborhood selection approach of Meinshausen and Bühlmann (2006) that combines p ℓ_1-penalized regression problems in order to obtain a symmetric estimator. The authors claimed that the proposal performs well in estimating a scale-free network.

We generated data under Set-ups I and III (described in Section 3.5.2) with $n = 250$ and $p = 500$,[5] and with $|\mathcal{H}| = 10$ for Set-up I. The results, averaged over 100 data sets, are displayed in Figures 4 and 5.

To obtain Figures 4 and 5, we applied Liu and Ihler (2011) using a fine grid of α values, and using the choices for β_j and ϵ_j specified by the authors: $\beta_j = 2\alpha/\epsilon_j$, where ϵ_j is a small constant specified in Liu and Ihler (2011). There are two tuning parameters in Hero and Rajaratnam (2012): (1) ρ, the value used to threshold the partial correlation matrix, and (2) d, the number of non-zero elements required for a column of the thresholded matrix to be declared a hub node. We used $d = \{10, 20\}$ in Figures 4 and 5, and used a fine grid of values for ρ. Note that the value of d has no effect on the results for Figures 4(a)-(b) and Figures 5(a)-(b), and that larger values of d tend to yield worse results in Figures 4(c) and 5(c). For Peng et al. (2009), we used a fine grid of tuning parameter values to obtain the curves shown in Figures 4 and 5. The sum of squared errors was not reported for Peng et al. (2009) and Hero and Rajaratnam (2012) since they do not directly yield an estimate of the precision matrix. As a baseline reference, the graphical lasso is included in the comparison.

We see from Figure 4 that HGL outperforms the competitors when the underlying network contains hub nodes. It is not surprising that Liu and Ihler (2011) yields better results than the graphical lasso, since the former approach is implemented via an iterative procedure: in each iteration, the graphical lasso is performed with an updated tuning parameter based on the estimate obtained in the previous iteration. Hero and Rajaratnam (2012) has the worst results in Figures 4(a)-(b); this is not surprising, since the purpose of Hero and Rajaratnam (2012) is to screen for hub nodes, rather than to estimate the individual edges in the network.

From Figure 5, we see that the performance of HGL is comparable to that of Liu and Ihler (2011) and Peng et al. (2009) under the assumption of a scale-free network; note that this is the precise setting for which Liu and Ihler (2011)'s proposal is intended, and Peng et al. (2009) reported that their proposal performs well in this setting. In contrast, HGL is not intended for the scale-free network setting (as mentioned in the Introduction, it is intended for a setting with hub nodes). Again, Liu and Ihler (2011) and Peng et al. (2009) outperform the graphical lasso, and Hero and Rajaratnam (2012) has the worst results in Figures 5(a)-(b). Finally, we see from Figures 4 and 5 that the BIC-type criterion for HGL proposed in Section 3.4 yields good results.

[5] In this subsection, a small value of p was used due to the computations required to run the R package space, as well as computational demands of the Liu and Ihler (2011) algorithm.

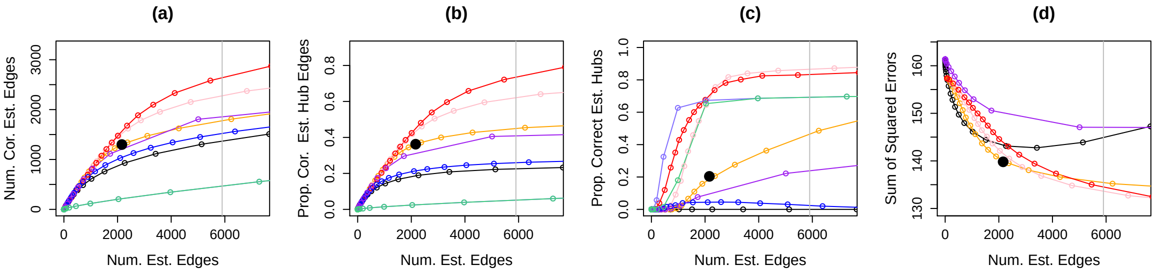

Figure 4: Simulation for the Gaussian graphical model. Set-up I was applied with $n = 250$ and $p = 500$. Details of the axis labels and the solid black circles are as in Figure 3. The colored lines correspond to the graphical lasso (Friedman et al., 2007) (——); HGL with $\lambda_3 = 1$ (——), $\lambda_3 = 2$ (——), and $\lambda_3 = 3$ (——); the hub screening procedure (Hero and Rajaratnam, 2012) with $d = 10$ (——) and $d = 20$ (——); the scale-free network approach (Liu and Ihler, 2011) (——); sparse partial correlation estimation (Peng et al., 2009) (——).

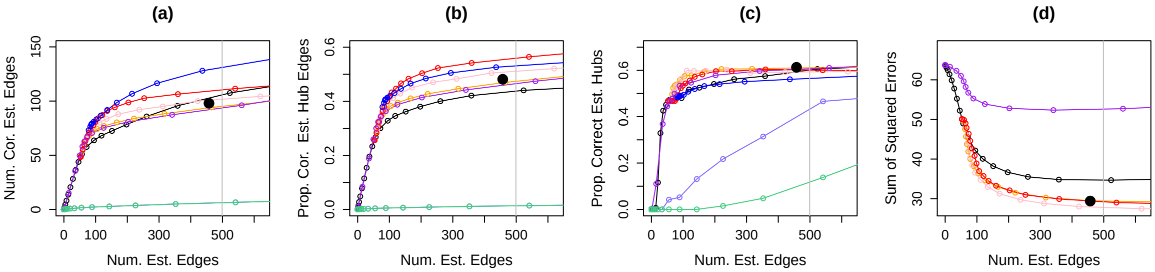

Figure 5: Simulation for the Gaussian graphical model. Set-up III was applied with $n = 250$ and $p = 500$. Details of the axis labels and the solid black circles are as in Figure 3. The colored lines correspond to the graphical lasso (Friedman et al., 2007) (——); HGL with $\lambda_3 = 1$ (——), $\lambda_3 = 2$ (——), and $\lambda_3 = 3$ (——); the hub screening procedure (Hero and Rajaratnam, 2012) with $d = 10$ (——) and $d = 20$ (——); the scale-free network approach (Liu and Ihler, 2011) (——); sparse partial correlation estimation (Peng et al., 2009) (——).

4. The Hub Covariance Graph

In this section, we consider estimation of a covariance matrix under the assumption that $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$; this is of interest because the sparsity pattern of $\mathbf{\Sigma}$ specifies the structure of the marginal independence graph (see, e.g., Drton and Richardson, 2003; Chaudhuri et al., 2007; Drton and Richardson, 2008). We extend the covariance estimator of Xue et al. (2012) to accommodate hub nodes.

4.1 Formulation and Algorithm

Xue et al. (2012) proposed to estimate $\mathbf{\Sigma}$ using

$$\hat{\mathbf{\Sigma}} = \underset{\mathbf{\Sigma} \in \mathcal{S}}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{\Sigma} - \mathbf{S}\|_F^2 + \lambda \|\mathbf{\Sigma}\|_1 \right\}, \tag{9}$$

where $\mathbf{S}$ is the empirical covariance matrix, $\mathcal{S} = \{\mathbf{\Sigma} : \mathbf{\Sigma} \succeq \epsilon \mathbf{I} \text{ and } \mathbf{\Sigma} = \mathbf{\Sigma}^T\}$, and ϵ is a small positive constant; we take $\epsilon = 10^{-4}$. We extend (9) to accommodate hubs by imposing the hub penalty function (2) on $\mathbf{\Sigma}$. This results in the *hub covariance graph* (HCG) optimization problem,

$$\underset{\mathbf{\Sigma} \in \mathcal{S}}{\text{minimize}} \qquad \left\{ \frac{1}{2} \|\mathbf{\Sigma} - \mathbf{S}\|_F^2 + \mathrm{P}(\mathbf{\Sigma}) \right\},$$

which can be solved via Algorithm 1. To update $\mathbf{\Theta} = \mathbf{\Sigma}$ in Step 2(a)i, we note that

$$\underset{\mathbf{\Sigma} \in \mathcal{S}}{\arg\min} \left\{ \frac{1}{2} \|\mathbf{\Sigma} - \mathbf{S}\|_F^2 + \frac{\rho}{2} \|\mathbf{\Sigma} - \tilde{\mathbf{\Sigma}} + \mathbf{W}_1\|_F^2 \right\} = \frac{1}{1+\rho} (\mathbf{S} + \rho \tilde{\mathbf{\Sigma}} - \rho \mathbf{W}_1)^+,$$

where $(\mathbf{A})^+$ is the projection of a matrix $\mathbf{A}$ onto the convex cone $\{\mathbf{\Sigma} \succeq \epsilon \mathbf{I}\}$. That is, if $\sum_{j=1}^p d_j \mathbf{u}_j \mathbf{u}_j^T$ denotes the eigen-decomposition of the matrix $\mathbf{A}$, then $(\mathbf{A})^+$ is defined as $\sum_{j=1}^p \max(d_j, \epsilon) \mathbf{u}_j \mathbf{u}_j^T$. The complexity of the ADMM algorithm is $O(p^3)$ per iteration, due to the complexity of the eigen-decomposition for updating $\mathbf{\Sigma}$.

4.2 Simulation Study

We compare HCG to two competitors for obtaining a sparse estimate of $\mathbf{\Sigma}$:

1. The non-convex ℓ_1-penalized log-likelihood approach of Bien and Tibshirani (2011), using the `R` package `spcov`. This approach solves

$$\underset{\mathbf{\Sigma} \succ 0}{\text{minimize}} \left\{ \log \det \mathbf{\Sigma} + \text{trace}(\mathbf{\Sigma}^{-1} \mathbf{S}) + \lambda \|\mathbf{\Sigma}\|_1 \right\}.$$

2. The convex ℓ_1-penalized approach of Xue et al. (2012), given in (9).

We first generated an adjacency matrix $\mathbf{A}$ as in Set-up I in Section 3.5.2, modified to have $|\mathcal{H}| = 20$ hub nodes. Then $\bar{\mathbf{E}}$ was generated as described in Section 3.5.2, and we set $\mathbf{\Sigma}$ equal to $\bar{\mathbf{E}} + (0.1 - \Lambda_{\min}(\bar{\mathbf{E}}))\mathbf{I}$. Next, we generated $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$. Finally, we standardized the variables to have standard deviation one. In this simulation study, we set $n = 500$ and $p = 1000$.

Figure 6 displays the results, averaged over 100 simulated data sets. We calculated the proportion of correctly estimated hub nodes as defined in Section 3.3.1 with $r = 200$. We used a fine grid of tuning parameters for Xue et al. (2012) in order to obtain the curves shown in each panel of Figure 6. HCG involves three tuning parameters, λ_1, λ_2, and λ_3. We fixed $\lambda_1 = 0.2$, considered three values of λ_3 (each shown in a different color), and varied λ_2 in order to obtain the curves shown in Figure 6.

Figure 6 does not display the results for the proposal of Bien and Tibshirani (2011), due to computational constraints in the `spcov` R package. Instead, we compared our proposal to that of Bien and Tibshirani (2011) using $n = 100$ and $p = 200$; those results are presented in Figure 10 in Appendix D.

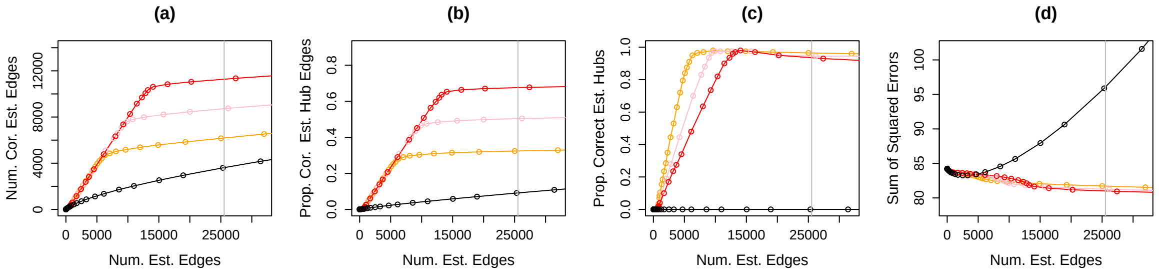

Figure 6: Covariance graph simulation with $n = 500$ and $p = 1000$. Details of the axis labels are as in Figure 3. The colored lines correspond to the proposal of Xue et al. (2012) (——); HCG with $\lambda_3 = 1$ (——), $\lambda_3 = 1.5$ (——), and $\lambda_3 = 2$ (——).

We see that HCG outperforms the proposals of Xue et al. (2012) (Figures 6 and 10) and Bien and Tibshirani (2011) (Figure 10). These results are not surprising, since those other methods do not explicitly model the hub nodes.

5. The Hub Binary Network

In this section, we focus on estimating a binary Ising Markov random field, which we refer to as a binary network. We refer the reader to Ravikumar et al. (2010) for an in-depth discussion of this type of graphical model and its applications.

In this set-up, each entry of the $n \times p$ data matrix $\mathbf{X}$ takes on a value of zero or one. We assume that the observations $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are i.i.d. with density

$$p(\mathbf{x}, \boldsymbol{\Theta}) = \frac{1}{Z(\boldsymbol{\Theta})} \exp\left[\sum_{j=1}^{p} \theta_{jj} x_j + \sum_{1 \le j < j' \le p} \theta_{jj'} x_j x_{j'}\right], \tag{10}$$

where $Z(\boldsymbol{\Theta})$ is the partition function, which ensures that the density sums to one. Here $\boldsymbol{\Theta}$ is a $p \times p$ symmetric matrix that specifies the network structure: $\theta_{jj'} = 0$ implies that the jth and j'th variables are conditionally independent.

In order to obtain a sparse graph, Lee et al. (2007) considered maximizing an ℓ_1-penalized log-likelihood under this model. Due to the difficulty in computing the log-partition function, several authors have considered alternative approaches. For instance, Ravikumar et al. (2010) proposed a neighborhood selection approach. The proposal of Ravikumar et al. (2010) involves solving p logistic regression separately, and hence, the estimated parameter matrix is not symmetric. In contrast, several authors considered maximizing an ℓ_1-penalized pseudo-likelihood with a symmetric constraint on $\mathbf{\Theta}$ (see, e.g., Höfling and Tibshirani, 2009; Guo et al., 2010, 2011).

5.1 Formulation and Algorithm

Under the model (10), the log-pseudo-likelihood for n observations takes the form

$$\sum_{j=1}^{p}\sum_{j'=1}^{p}\theta_{jj'}(\mathbf{X}^T\mathbf{X})_{jj'} - \sum_{i=1}^{n}\sum_{j=1}^{p}\log\left(1+\exp\left[\theta_{jj}+\sum_{j'\neq j}\theta_{jj'}x_{ij'}\right]\right), \tag{11}$$

where $\mathbf{x}_i$ is the ith row of the $n \times p$ matrix $\mathbf{X}$. The proposal of Höfling and Tibshirani (2009) involves maximizing (11) subject to an ℓ_1 penalty on $\mathbf{\Theta}$. We propose to instead impose the hub penalty function (2) on $\mathbf{\Theta}$ in (11) in order to estimate a sparse binary network with hub nodes. This leads to the optimization problem

$$\underset{\mathbf{\Theta}\in\mathcal{S}}{\text{minimize}} \quad \left\{-\sum_{j=1}^{p}\sum_{j'=1}^{p}\theta_{jj'}(\mathbf{X}^T\mathbf{X})_{jj'} + \sum_{i=1}^{n}\sum_{j=1}^{p}\log\left(1+\exp\left[\theta_{jj}+\sum_{j'\neq j}\theta_{jj'}x_{ij'}\right]\right) + \mathrm{P}(\mathbf{\Theta})\right\}, \tag{12}$$

where $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} = \mathbf{\Theta}^T\}$. We refer to the solution to (12) as the *hub binary network* (HBN). The ADMM algorithm for solving (12) is given in Algorithm 1. We solve the update for $\mathbf{\Theta}$ in Step 2(a)i using the Barzilai-Borwein method (Barzilai and Borwein, 1988). The details are given in Appendix F.

5.2 Simulation Study

Here we compare the performance of HBN to the proposal of Höfling and Tibshirani (2009), implemented using the `R` package `BMN`.

We simulated a binary network with $p = 50$ and $|\mathcal{H}| = 5$ hub nodes. To generate the parameter matrix $\mathbf{\Theta}$, we created an adjacency matrix $\mathbf{A}$ as in Set-up I of Section 3.5.2 with five hub nodes. Then $\bar{\mathbf{E}}$ was generated as in Section 3.5.2, and we set $\mathbf{\Theta} = \bar{\mathbf{E}}$.

Each of $n = 100$ observations was generated using Gibbs sampling (Ravikumar et al., 2010; Guo et al., 2010). Suppose that $x_1^{(t)}, \ldots, x_p^{(t)}$ is obtained at the tth iteration of the Gibbs sampler. Then, the $(t+1)$th iteration is obtained according to

$$x_j^{(t+1)} \sim \text{Bernoulli}\left(\frac{\exp(\theta_{jj}+\sum_{j\neq j'}\theta_{jj'}x_{j'}^{(t)})}{1+\exp(\theta_{jj}+\sum_{j\neq j'}\theta_{jj'}x_{j'}^{(t)})}\right) \qquad \text{for } j = 1, \ldots, p.$$

We took the first 10^5 iterations as our burn-in period, and then collected an observation every 10^4 iterations, such that the observations were nearly independent (Guo et al., 2010).

The results, averaged over 100 data sets, are shown in Figure 7. We used a fine grid of values for the ℓ_1 tuning parameter for Höfling and Tibshirani (2009), resulting in curves shown in each panel of the figure. For HBN, we fixed $\lambda_1 = 5$, considered $\lambda_3 = \{15, 25, 30\}$, and used a fine grid of values of λ_2. The proportion of correctly estimated hub nodes was calculated using the definition in Section 3.5.1 with $r = 20$. Figure 7 indicates that HBN consistently outperforms the proposal of Höfling and Tibshirani (2009).

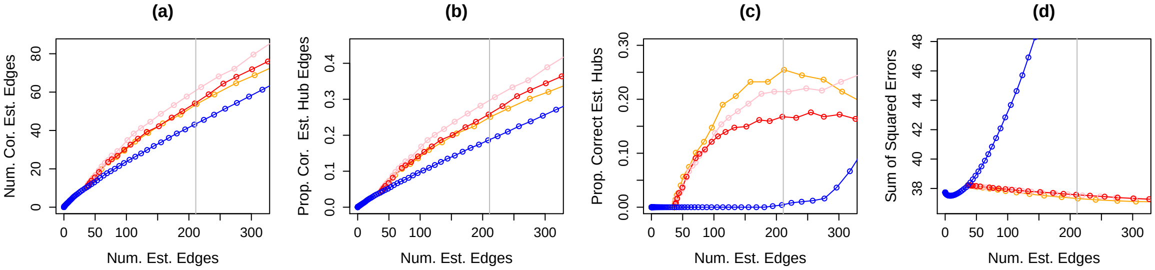

Figure 7: Binary network simulation with $n = 100$ and $p = 50$. Details of the axis labels are as in Figure 3. The colored lines correspond to the ℓ_1-penalized pseudo-likelihood proposal of Höfling and Tibshirani (2009) (——); and HBN with $\lambda_3 = 15$ (——), $\lambda_3 = 25$ (——), and $\lambda_3 = 30$ (——).

6. Real Data Application

We now apply HGL to a university webpage data set, and a brain cancer data set.

6.1 Application to University Webpage Data

We applied HGL to the university webpage data set from the "World Wide Knowledge Base" project at Carnegie Mellon University. This data set was pre-processed by Cardoso-Cachopo (2009). The data set consists of the occurrences of various terms (words) on webpages from four computer science departments at Cornell, Texas, Washington and Wisconsin. We consider only the 544 student webpages, and select 100 terms with the largest entropy for our analysis. In what follows, we model these 100 terms as the nodes in a Gaussian graphical model.

The goal of the analysis is to understand the relationships among the terms that appear on the student webpages. In particular, we wish to identify terms that are hubs. We are not interested in identifying edges between non-hub nodes. For this reason, we fix the tuning parameter that controls the sparsity of $\mathbf{Z}$ at $\lambda_1 = 0.45$ such that the matrix $\mathbf{Z}$ is sparse. In the interest of a graph that is interpretable, we fix $\lambda_3 = 1.5$ to obtain only a few hub nodes, and then select a value of λ_2 ranging from 0.1 to 0.5 using the BIC-type criterion presented in Section 3.4. We performed HGL with the selected tuning parameters $\lambda_1 = 0.45$, $\lambda_2 = 0.25$, and $\lambda_3 = 1.5$.[6] The estimated matrices are shown in Figure 8.

Figure 8(a) indicates that six hub nodes are detected: *comput*, *research*, *scienc*, *software*, *system*, and *work*. For instance, the fact that *comput* is a hub indicates that many terms'

[6] The results are qualitatively similar for different values of λ_1.

occurrences are explained by the occurrence of the word *comput*. From Figure 8(b), we see that several pairs of terms take on non-zero values in the matrix $(\mathbf{Z}-\text{diag}(\mathbf{Z}))$. These include *(depart, univers)*; *(home, page)*; *(institut, technolog)*; *(graduat, student)*; *(univers, scienc)*, and *(languag,program)*. These results provide an intuitive explanation of the relationships among the terms in the webpages.

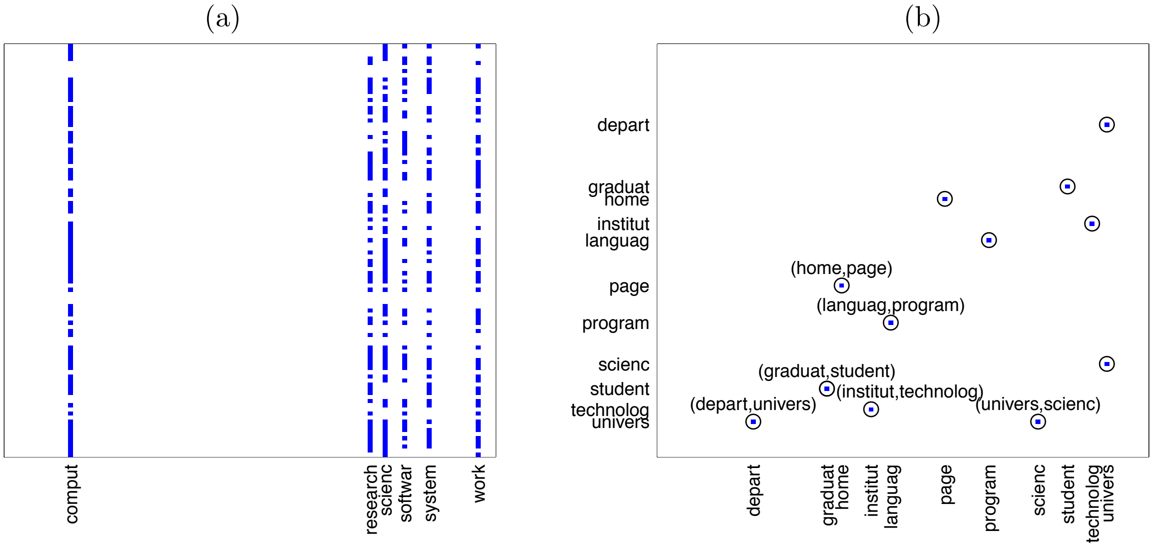

Figure 8: Results for HGL on the webpage data with tuning parameters selected using BIC: $\lambda_1 = 0.45$, $\lambda_2 = 0.25$, $\lambda_3 = 1.5$. Non-zero estimated values are shown, for *(a):* $(\mathbf{V} - \text{diag}(\mathbf{V}))$, and *(b):* $(\mathbf{Z} - \text{diag}(\mathbf{Z}))$.

6.2 Application to Gene Expression Data

We applied HGL to a publicly available cancer gene expression data set (Verhaak et al., 2010). The data set consists of mRNA expression levels for 17,814 genes in 401 patients with glioblastoma multiforme (GBM), an extremely aggressive cancer with very poor patient prognosis. Among 7,462 genes known to be associated with cancer (Rappaport et al., 2013), we selected 500 genes with the highest variance.

We aim to reconstruct the gene regulatory network that represents the interactions among the genes, as well as to identify hub genes that tend to have many interactions with other genes. Such genes likely play an important role in regulating many other genes in the network. Identifying such regulatory genes will lead to a better understanding of brain cancer, and eventually may lead to new therapeutic targets. Since we are interested in identifying hub genes, and not as interested in identifying edges between non-hub nodes, we fix $\lambda_1 = 0.6$ such that the matrix $\mathbf{Z}$ is sparse. We fix $\lambda_3 = 6.5$ to obtain a few hub nodes, and we select λ_2 ranging from 0.1 to 0.7 using the BIC-type criterion presented in Section 3.4.

We applied HGL with this set of tuning parameters to the empirical covariance matrix corresponding to the 401×500 data matrix, after standardizing each gene to have variance one. In Figure 9, we plotted the resulting network (for simplicity, only the 438 genes with at least two neighbors are displayed). We found that five genes are identified as hubs. These genes are TRIM48, TBC1D2B, PTPN2, ACRC, and ZNF763, in decreasing order of estimated edges.

Interestingly, some of these genes have known regulatory roles. PTPN2 is known to be a signaling molecule that regulates a variety of cellular processes including cell growth, differentiation, mitotic cycle, and oncogenic transformation (Maglott et al., 2004). ZNF763 is a DNA-binding protein that regulates the transcription of other genes (Maglott et al., 2004). These genes do not appear to be highly-connected to many other genes in the estimate that results from applying the graphical lasso (5) to this same data set (results not shown). These results indicate that HGL can be used to recover known regulators, as well as to suggest other potential regulators that may be targets for follow-up analysis.

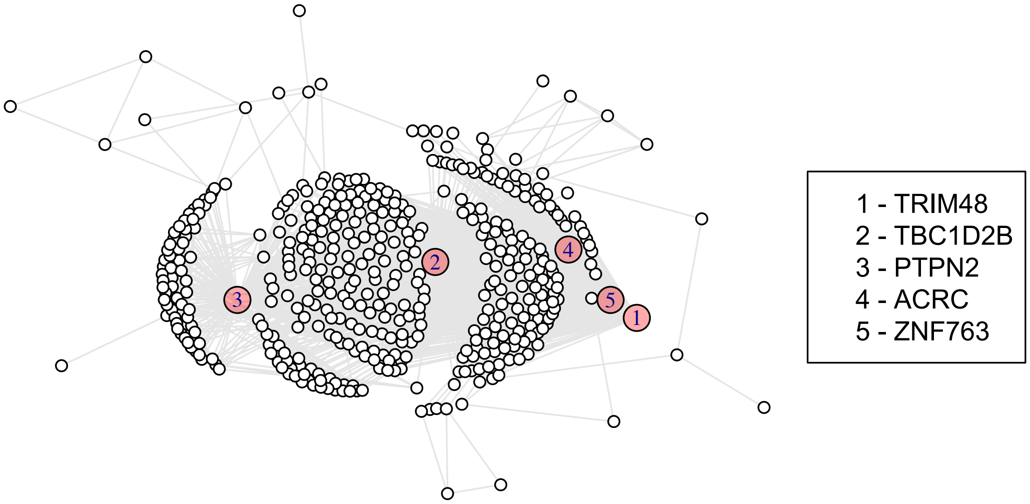

Figure 9: Results for HGL on the GBM data with tuning parameters selected using BIC: $\lambda_1 = 0.6$, $\lambda_2 = 0.4$, $\lambda_3 = 6.5$. Only nodes with at least two edges in the estimated network are displayed. Nodes displayed in pink were found to be hubs by the HGL algorithm.

7. Discussion

We have proposed a general framework for estimating a network with hubs by way of a convex penalty function. The proposed framework has three tuning parameters, so that it can flexibly accommodate different numbers of hubs, sparsity levels within a hub, and connectivity levels among non-hubs. We have proposed a BIC-type quantity to select tuning parameters for our proposal. We note that tuning parameter selection in unsupervised settings remains a challenging open problem (see, e.g., Foygel and Drton, 2010; Meinshausen and Bühlmann, 2010). In practice, tuning parameters could also be set based on domain knowledge or a desire for interpretability of the resulting estimates.

The framework proposed in this paper assumes an underlying model involving a set of edges between non-hub nodes, as well as a set of hub nodes. For instance, it is believed that such hub nodes arise in biology, in which "super hubs" in transcriptional regulatory networks may play important roles (Hao et al., 2012). We note here that the underlying model of hub nodes assumed in this paper differs fundamentally from a scale-free network in which the degree of connectivity of the nodes follows a power law distribution—scale-free networks simply do not have such very highly-connected hub nodes. In fact, we have shown that existing techniques for estimating a scale-free network, such as Liu and Ihler (2011) and Defazio and Caetano (2012), cannot accommodate the very dense hubs for which our proposal is intended.

As discussed in Section 2, the hub penalty function involves decomposing a parameter matrix $\mathbf{\Theta}$ into $\mathbf{Z}+\mathbf{V}+\mathbf{V}^T$, where $\mathbf{Z}$ is a sparse matrix, and $\mathbf{V}$ is a matrix whose columns are entirely zero or (almost) entirely non-zero. In this paper, we used an ℓ_1 penalty on $\mathbf{Z}$ in order to encourage it to be sparse. In effect, this amounts to assuming that the non-hub nodes obey an Erdős-Rényi network. But our formulation could be easily modified to accommodate a different network prior for the non-hub nodes. For instance, we could assume that the non-hub nodes obey a scale-free network, using the ideas developed in Liu and Ihler (2011) and Defazio and Caetano (2012). This would amount to modeling a scale-free network with hub nodes.

In this paper, we applied the proposed framework to the tasks of estimating a Gaussian graphical model, a covariance graph model, and a binary network. The proposed framework can also be applied to other types of graphical models, such as the Poisson graphical model (Allen and Liu, 2012) or the exponential family graphical model (Yang et al., 2012a).

In future work, we will study the theoretical statistical properties of the HGL formulation. For instance, in the context of the graphical lasso, it is known that the rate of statistical convergence depends upon the maximal degree of any node in the network (Ravikumar et al., 2011). It would be interesting to see whether HGL theoretically outperforms the graphical lasso in the setting in which the true underlying network contains hubs. Furthermore, it will be of interest to study HGL's hub recovery properties from a theoretical perspective.

An `R` package `hglasso` is publicly available on the authors' websites and on `CRAN`.

Acknowledgments

We thank three reviewers for helpful comments that improved the quality of this manuscript. We thank Qiang Liu helpful responses to our inquiries regarding Liu and Ihler (2011). The authors acknowledge funding from the following sources: NIH DP5OD009145 and NSF CAREER DMS-1252624 and Sloan Research Fellowship to DW, NSF CAREER ECCS-0847077 to MF, and Univ. Washington Royalty Research Fund to DW, MF, and SL.

Appendix A. Derivation of Algorithm 1

Recall that the scaled augmented Lagrangian for (4) takes the form

$$\begin{aligned} L(\mathbf{B}, \tilde{\mathbf{B}}, \mathbf{W}) = \ell(\mathbf{X}, \mathbf{\Theta}) + \lambda_1 \|\mathbf{Z} - \text{diag}(\mathbf{Z})\|_1 + \lambda_2 \|\mathbf{V} - \text{diag}(\mathbf{V})\|_1 \\ + \lambda_3 \sum_{j=1}^{p} \|(\mathbf{V} - \text{diag}(\mathbf{V}))_j\|_2 + g(\tilde{\mathbf{B}}) + \frac{\rho}{2}\|\mathbf{B} - \tilde{\mathbf{B}} + \mathbf{W}\|_F^2. \end{aligned} \tag{13}$$

The proposed ADMM algorithm requires the following updates:

1. $\mathbf{B}^{(t+1)} \leftarrow \underset{\mathbf{B}}{\text{argmin}}\ L(\mathbf{B}, \tilde{\mathbf{B}}^t, \mathbf{W}^t)$,
2. $\tilde{\mathbf{B}}^{(t+1)} \leftarrow \underset{\tilde{\mathbf{B}}}{\text{argmin}}\ L(\mathbf{B}^{(t+1)}, \tilde{\mathbf{B}}, \mathbf{W}^t)$,
3. $\mathbf{W}^{(t+1)} \leftarrow \mathbf{W}^t + \mathbf{B}^{(t+1)} - \tilde{\mathbf{B}}^{(t+1)}$.

We now proceed to derive the updates for $\mathbf{B}$ and $\tilde{\mathbf{B}}$.

Updates for B

To obtain updates for $\mathbf{B} = (\mathbf{\Theta}, \mathbf{V}, \mathbf{Z})$, we exploit the fact that (13) is separable in $\mathbf{\Theta}, \mathbf{V}$, and $\mathbf{Z}$. Therefore, we can simply update with respect to $\mathbf{\Theta}, \mathbf{V}$, and $\mathbf{Z}$ one-at-a-time. Update for $\mathbf{\Theta}$ depends on the form of the convex loss function, and is addressed in the main text. Updates for $\mathbf{V}$ and $\mathbf{Z}$ can be easily seen to take the form given in Algorithm 1.

Updates for $\tilde{\mathbf{B}}$

Minimizing the function in (13) with respect to $\tilde{\mathbf{B}}$ is equivalent to

$$\begin{aligned} &\underset{\tilde{\mathbf{\Theta}}, \tilde{\mathbf{V}}, \tilde{\mathbf{Z}}}{\text{minimize}} && \left\{ \frac{\rho}{2}\|\mathbf{\Theta} - \tilde{\mathbf{\Theta}} + \mathbf{W}_1\|_F^2 + \frac{\rho}{2}\|\mathbf{V} - \tilde{\mathbf{V}} + \mathbf{W}_2\|_F^2 + \frac{\rho}{2}\|\mathbf{Z} - \tilde{\mathbf{Z}} + \mathbf{W}_3\|_F^2 \right\} \\ &\text{subject to} && \tilde{\mathbf{\Theta}} = \tilde{\mathbf{Z}} + \tilde{\mathbf{V}} + \tilde{\mathbf{V}}^T. \end{aligned} \tag{14}$$

Let $\mathbf{\Gamma}$ be the $p \times p$ Lagrange multiplier matrix for the equality constraint. Then, the Lagrangian for (14) is

$$\frac{\rho}{2}\|\mathbf{\Theta} - \tilde{\mathbf{\Theta}} + \mathbf{W}_1\|_F^2 + \frac{\rho}{2}\|\mathbf{V} - \tilde{\mathbf{V}} + \mathbf{W}_2\|_F^2 + \frac{\rho}{2}\|\mathbf{Z} - \tilde{\mathbf{Z}} + \mathbf{W}_3\|_F^2 + \langle \mathbf{\Gamma}, \tilde{\mathbf{\Theta}} - \tilde{\mathbf{Z}} - \tilde{\mathbf{V}} - \tilde{\mathbf{V}}^T \rangle.$$

A little bit of algebra yields

$$\tilde{\mathbf{\Theta}} = \mathbf{\Theta} + \mathbf{W}_1 - \frac{1}{\rho}\mathbf{\Gamma},$$

$$\tilde{\mathbf{V}} = \frac{1}{\rho}(\mathbf{\Gamma} + \mathbf{\Gamma}^T) + \mathbf{V} + \mathbf{W}_2,$$

$$\tilde{\mathbf{Z}} = \frac{1}{\rho}\mathbf{\Gamma} + \mathbf{Z} + \mathbf{W}_3,$$

where $\mathbf{\Gamma} = \frac{\rho}{6}[(\mathbf{\Theta} + \mathbf{W}_1) - (\mathbf{V} + \mathbf{W}_2) - (\mathbf{V} + \mathbf{W}_2)^T - (\mathbf{Z} + \mathbf{W}_3)]$.

Appendix B. Conditions for HGL Solution to be Block-Diagonal

We begin by introducing some notation. Let $\|\mathbf{V}\|_{u,v}$ be the ℓ_u/ℓ_v norm of a matrix $\mathbf{V}$. For instance, $\|\mathbf{V}\|_{1,q} = \sum_{j=1}^{p} \|\mathbf{V}_j\|_q$. We define the support of a matrix $\mathbf{\Theta}$ as follows: $\text{supp}(\mathbf{\Theta}) = \{(i,j) : \Theta_{ij} \neq 0\}$. We say that $\mathbf{\Theta}$ is supported on a set $\mathcal{G}$ if $\text{supp}(\mathbf{\Theta}) \subseteq \mathcal{G}$. Let $\{C_1, \ldots, C_K\}$ be a partition of the index set $\{1, \ldots, p\}$, and let $\mathcal{T} = \cup_{k=1}^{K} \{C_k \times C_k\}$. We let $\mathbf{A}_{\mathcal{T}}$ denote the restriction of the matrix $\mathbf{A}$ to the set $\mathcal{T}$: that is, $(\mathbf{A}_{\mathcal{T}})_{ij} = 0$ if $(i,j) \notin \mathcal{T}$ and $(\mathbf{A}_{\mathcal{T}})_{ij} = A_{ij}$ if $(i,j) \in \mathcal{T}$. Note that any matrix supported on $\mathcal{T}$ is block-diagonal with K blocks, subject to some permutation of its rows and columns. Also, let $S_{\max} = \max\limits_{(i,j)\in\mathcal{T}^c} |S_{ij}|$. Define

$$\begin{aligned} \tilde{\mathbf{P}}(\mathbf{\Theta}) \quad = \quad & \min_{\mathbf{V},\mathbf{Z}} \quad \|\mathbf{Z} - \text{diag}(\mathbf{Z})\|_1 + \hat{\lambda}_2 \|\mathbf{V} - \text{diag}(\mathbf{V})\|_1 + \hat{\lambda}_3 \|\mathbf{V} - \text{diag}(\mathbf{V})\|_{1,q} \\ & \text{subject to} \quad \mathbf{\Theta} = \mathbf{Z} + \mathbf{V} + \mathbf{V}^T, \end{aligned} \tag{15}$$

where $\hat{\lambda}_2 = \frac{\lambda_2}{\lambda_1}$ and $\hat{\lambda}_3 = \frac{\lambda_3}{\lambda_1}$. Then, optimization problem (6) is equivalent to

$$\underset{\mathbf{\Theta}\in\mathcal{S}}{\text{minimize}} \quad -\log\det(\mathbf{\Theta}) + \langle \mathbf{\Theta}, \mathbf{S} \rangle + \lambda_1 \tilde{\mathbf{P}}(\mathbf{\Theta}), \tag{16}$$

where $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} \succ 0, \mathbf{\Theta} = \mathbf{\Theta}^T\}$.

Proof of Theorem 1 (Sufficient Condition)

Proof First, we note that if $(\mathbf{\Theta}, \mathbf{V}, \mathbf{Z})$ is a feasible solution to (6), then $(\mathbf{\Theta}_{\mathcal{T}}, \mathbf{V}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{T}})$ is also a feasible solution to (6). Assume that $(\mathbf{\Theta}, \mathbf{V}, \mathbf{Z})$ is not supported on $\mathcal{T}$. We want to show that the objective value of (6) evaluated at $(\mathbf{\Theta}_{\mathcal{T}}, \mathbf{V}_{\mathcal{T}}, \mathbf{Z}_{\mathcal{T}})$ is smaller than the objective value of (6) evaluated at $(\mathbf{\Theta}, \mathbf{V}, \mathbf{Z})$. By Fischer's inequality (Horn and Johnson, 1985),

$$-\log\det(\mathbf{\Theta}) \geq -\log\det(\mathbf{\Theta}_{\mathcal{T}}).$$

Therefore, it remains to show that

$$\begin{aligned} \langle \mathbf{\Theta}, \mathbf{S} \rangle + \lambda_1 \|\mathbf{Z} - \text{diag}(\mathbf{Z})\|_1 + \lambda_2 \|\mathbf{V} - \text{diag}(\mathbf{V})\|_1 + \lambda_3 \|\mathbf{V} - \text{diag}(\mathbf{V})\|_{1,q} \quad > \\ \langle \mathbf{\Theta}_{\mathcal{T}}, \mathbf{S} \rangle + \lambda_1 \|\mathbf{Z}_{\mathcal{T}} - \text{diag}(\mathbf{Z}_{\mathcal{T}})\|_1 + \lambda_2 \|\mathbf{V}_{\mathcal{T}} - \text{diag}(\mathbf{V}_{\mathcal{T}})\|_1 + \lambda_3 \|\mathbf{V}_{\mathcal{T}} - \text{diag}(\mathbf{V}_{\mathcal{T}})\|_{1,q}, \end{aligned}$$

or equivalently, that

$$\langle \mathbf{\Theta}_{\mathcal{T}^c}, \mathbf{S} \rangle + \lambda_1 \|\mathbf{Z}_{\mathcal{T}^c}\|_1 + \lambda_2 \|\mathbf{V}_{\mathcal{T}^c}\|_1 + \lambda_3 (\|\mathbf{V} - \text{diag}(\mathbf{V})\|_{1,q} - \|\mathbf{V}_{\mathcal{T}} - \text{diag}(\mathbf{V}_{\mathcal{T}})\|_{1,q}) > 0.$$

Since $\|\mathbf{V} - \text{diag}(\mathbf{V})\|_{1,q} \geq \|\mathbf{V}_{\mathcal{T}} - \text{diag}(\mathbf{V}_{\mathcal{T}})\|_{1,q}$, it suffices to show that

$$\langle \mathbf{\Theta}_{\mathcal{T}^c}, \mathbf{S} \rangle + \lambda_1 \|\mathbf{Z}_{\mathcal{T}^c}\|_1 + \lambda_2 \|\mathbf{V}_{\mathcal{T}^c}\|_1 > 0. \tag{17}$$

Note that $\langle \mathbf{\Theta}_{\mathcal{T}^c}, \mathbf{S} \rangle = \langle \mathbf{\Theta}_{\mathcal{T}^c}, \mathbf{S}_{\mathcal{T}^c} \rangle$. By the sufficient condition, $S_{\max} < \lambda_1$ and $2S_{\max} < \lambda_2$.

In addition, we have that

$$\begin{aligned}|\langle \mathbf{\Theta}_{\mathcal{T}^c}, \mathbf{S}\rangle| &= |\langle \mathbf{\Theta}_{\mathcal{T}^c}, \mathbf{S}_{\mathcal{T}^c}\rangle| \\ &= |\langle \mathbf{V}_{\mathcal{T}^c} + \mathbf{V}_{\mathcal{T}^c}^T + \mathbf{Z}_{\mathcal{T}^c}, \mathbf{S}_{\mathcal{T}^c}\rangle| \\ &= |\langle 2\mathbf{V}_{\mathcal{T}^c} + \mathbf{Z}_{\mathcal{T}^c}, \mathbf{S}_{\mathcal{T}^c}\rangle| \\ &\leq (2\|\mathbf{V}_{\mathcal{T}^c}\|_1 + \|\mathbf{Z}_{\mathcal{T}^c}\|_1) S_{\max} \\ &< \ \lambda_2 \|\mathbf{V}_{\mathcal{T}^c}\|_1 + \lambda_1 \|\mathbf{Z}_{\mathcal{T}^c}\|_1,\end{aligned}$$

where the last inequality follows from the sufficient condition. We have shown (17) as desired.

■

Proof of Theorem 2 (Necessary Condition)

We first present a simple lemma for proving Theorem 2. Throughout the proof of Theorem 2, $\|\cdot\|_\infty$ indicates the maximal absolute element of a matrix and $\|\cdot\|_{\infty,s}$ indicates the dual norm of $\|\cdot\|_{1,q}$.

Lemma 7 *The dual representation of* $\tilde{\mathbf{P}}(\mathbf{\Theta})$ *in (15) is*

$$\begin{array}{rcll}\tilde{\mathbf{P}}^*(\mathbf{\Theta}) & = & \underset{\mathbf{X},\mathbf{Y},\mathbf{\Lambda}}{\max} & \langle \mathbf{\Lambda}, \mathbf{\Theta}\rangle \\ & & \text{subject to} & \mathbf{\Lambda} + \mathbf{\Lambda}^T = \hat{\lambda}_2 \mathbf{X} + \hat{\lambda}_3 \mathbf{Y} \\ & & & \|\mathbf{X}\|_\infty \leq 1, \|\mathbf{\Lambda}\|_\infty \leq 1, \|\mathbf{Y}\|_{\infty,s} \leq 1 \\ & & & X_{ii} = 0, Y_{ii} = 0, \Lambda_{ii} = 0 \textit{ for } i = 1, \ldots, p,\end{array} \quad (18)$$

where $\frac{1}{s} + \frac{1}{q} = 1$.

Proof We first state the dual representations for the norms in (15):

$$\begin{array}{rcll}\|\mathbf{Z} - \text{diag}(\mathbf{Z})\|_1 & = & \underset{\mathbf{\Lambda}}{\max} & \langle \mathbf{\Lambda}, \mathbf{Z}\rangle \\ & & \text{subject to} & \|\mathbf{\Lambda}\|_\infty \leq 1, \Lambda_{ii} = 0 \text{ for } i = 1, \ldots, p,\end{array}$$

$$\begin{array}{rcll}\|\mathbf{V} - \text{diag}(\mathbf{V})\|_1 & = & \underset{\mathbf{X}}{\max} & \langle \mathbf{X}, \mathbf{V}\rangle \\ & & \text{subject to} & \|\mathbf{X}\|_\infty \leq 1, X_{ii} = 0 \text{ for } i = 1, \ldots, p,\end{array}$$

$$\begin{array}{rcll}\|\mathbf{V} - \text{diag}(\mathbf{V})\|_{1,q} & = & \underset{\mathbf{Y}}{\max} & \langle \mathbf{Y}, \mathbf{V}\rangle \\ & & \text{subject to} & \|\mathbf{Y}\|_{\infty,s} \leq 1, Y_{ii} = 0 \text{ for } i = 1, \ldots, p.\end{array}$$

Then,

$$\begin{aligned}
\tilde{\mathbf{P}}(\boldsymbol{\Theta}) &= \underset{\mathbf{V},\mathbf{Z}}{\min} && \|\mathbf{Z}-\mathrm{diag}(\mathbf{Z})\|_1 + \hat{\lambda}_2\|\mathbf{V}-\mathrm{diag}(\mathbf{V})\|_1 + \hat{\lambda}_3\|\mathbf{V}-\mathrm{diag}(\mathbf{V})\|_{1,q} \\
& \quad\text{subject to} && \boldsymbol{\Theta} = \mathbf{Z}+\mathbf{V}+\mathbf{V}^T \\
&= \underset{\mathbf{V},\mathbf{Z}}{\min} && \underset{\boldsymbol{\Lambda},\mathbf{X},\mathbf{Y}}{\max}\langle\boldsymbol{\Lambda},\mathbf{Z}\rangle + \hat{\lambda}_2\langle\mathbf{X},\mathbf{V}\rangle + \hat{\lambda}_3\langle\mathbf{Y},\mathbf{V}\rangle \\
& \quad\text{subject to} && \|\boldsymbol{\Lambda}\|_\infty \le 1, \|\mathbf{X}\|_\infty \le 1, \|\mathbf{Y}\|_{\infty,s} \le 1 \\
& && \Lambda_{ii}=0, X_{ii}=0, Y_{ii}=0 \text{ for } i=1,\ldots,p \\
& && \boldsymbol{\Theta} = \mathbf{Z}+\mathbf{V}+\mathbf{V}^T \\
&= \underset{\boldsymbol{\Lambda},\mathbf{X},\mathbf{Y}}{\max} && \underset{\mathbf{V},\mathbf{Z}}{\min}\langle\boldsymbol{\Lambda},\mathbf{Z}\rangle + \hat{\lambda}_2\langle\mathbf{X},\mathbf{V}\rangle + \hat{\lambda}_3\langle\mathbf{Y},\mathbf{V}\rangle \\
& \quad\text{subject to} && \|\boldsymbol{\Lambda}\|_\infty \le 1, \|\mathbf{X}\|_\infty \le 1, \|\mathbf{Y}\|_{\infty,s} \le 1 \\
& && \Lambda_{ii}=0, X_{ii}=0, Y_{ii}=0 \text{ for } i=1,\ldots,p \\
& && \boldsymbol{\Theta} = \mathbf{Z}+\mathbf{V}+\mathbf{V}^T \\
&= \underset{\boldsymbol{\Lambda},\mathbf{X},\mathbf{Y}}{\max} && \langle\boldsymbol{\Lambda},\boldsymbol{\Theta}\rangle \\
& \quad\text{subject to} && \boldsymbol{\Lambda}+\boldsymbol{\Lambda}^T = \hat{\lambda}_2\mathbf{X}+\hat{\lambda}_3\mathbf{Y} \\
& && \|\mathbf{X}\|_\infty \le 1, \|\boldsymbol{\Lambda}\|_\infty \le 1, \|\mathbf{Y}\|_{\infty,s} \le 1 \\
& && X_{ii}=0, Y_{ii}=0, \Lambda_{ii}=0 \text{ for } i=1,\ldots,p.
\end{aligned}$$

The third equality holds since the constraints on $(\mathbf{V},\mathbf{Z})$ and on $(\boldsymbol{\Lambda},\mathbf{X},\mathbf{Y})$ are both compact convex sets and so by the minimax theorem, we can swap max and min. The last equality follows from the fact that

$$\begin{aligned}
&\underset{\mathbf{V},\mathbf{Z}}{\min} && \langle\boldsymbol{\Lambda},\mathbf{Z}\rangle + \hat{\lambda}_2\langle\mathbf{X},\mathbf{V}\rangle + \hat{\lambda}_3\langle\mathbf{Y},\mathbf{V}\rangle \\
&\text{subject to} && \boldsymbol{\Theta} = \mathbf{Z}+\mathbf{V}+\mathbf{V}^T \\
&= && \begin{cases} \langle\boldsymbol{\Lambda},\boldsymbol{\Theta}\rangle & \text{if } \boldsymbol{\Lambda}+\boldsymbol{\Lambda}^T = \hat{\lambda}_2\mathbf{X}+\hat{\lambda}_3\mathbf{Y} \\ -\infty & \text{otherwise.} \end{cases}
\end{aligned}$$

■

We now present the proof of Theorem 2.

Proof The optimality condition for (16) is given by

$$\mathbf{0} = -\boldsymbol{\Theta}^{-1} + \mathbf{S} + \lambda_1\boldsymbol{\Lambda}, \qquad (19)$$

where $\boldsymbol{\Lambda}$ is a subgradient of $\tilde{\mathbf{P}}(\boldsymbol{\Theta})$ in (15) and the left-hand side of the above equation is a zero matrix of size $p \times p$.

Now suppose that $\boldsymbol{\Theta}^*$ that solves (19) is supported on $\mathcal{T}$, i.e., $\boldsymbol{\Theta}^*_{\mathcal{T}^c} = 0$. Then for any $(i,j) \in \mathcal{T}^c$, we have that

$$0 = S_{ij} + \lambda_1\Lambda^*_{ij}, \qquad (20)$$

where $\boldsymbol{\Lambda}^*$ is a subgradient of $\tilde{\mathbf{P}}(\boldsymbol{\Theta}^*)$. Note that $\boldsymbol{\Lambda}^*$ must be an optimal solution to the optimization problem (18). Therefore, it is also a feasible solution to (18), implying that

$$\begin{aligned}
|\Lambda^*_{ij} + \Lambda^*_{ji}| &\le \hat{\lambda}_2 + \hat{\lambda}_3, \\
|\Lambda^*_{ij}| &\le 1.
\end{aligned}$$

From (20), we have that $\Lambda^*_{ij} = -\frac{S_{ij}}{\lambda_1}$ and thus,

$$\begin{aligned} \lambda_1 &\geq \lambda_1 \max_{(i,j)\in\mathcal{T}^c} |\Lambda^*_{ij}| \\ &= \lambda_1 \max_{(i,j)\in\mathcal{T}^c} \frac{|S_{ij}|}{\lambda_1} \\ &= S_{\max}. \end{aligned}$$

Also, recall that $\hat{\lambda}_2 = \frac{\lambda_2}{\lambda_1}$ and $\hat{\lambda}_3 = \frac{\lambda_3}{\lambda_1}$. We have that

$$\begin{aligned} \lambda_2 + \lambda_3 &\geq \lambda_1 \max_{(i,j)\in\mathcal{T}^c} |\Lambda^*_{ij} + \Lambda^*_{ji}| \\ &= \lambda_1 \max_{(i,j)\in\mathcal{T}^c} \frac{2|S_{ij}|}{\lambda_1} \\ &= 2S_{\max}. \end{aligned}$$

Hence, we obtain the desired result.

■

Appendix C. Some Properties of HGL

Proof of Lemma 3

Proof Let $(\mathbf{\Theta}^*, \mathbf{Z}^*, \mathbf{V}^*)$ be the solution to (6) and suppose that $\mathbf{Z}^*$ is not a diagonal matrix. Note that $\mathbf{Z}^*$ is symmetric since $\mathbf{\Theta} \in \mathcal{S} \equiv \{\mathbf{\Theta} : \mathbf{\Theta} \succ 0 \text{ and } \mathbf{\Theta} = \mathbf{\Theta}^T\}$. Let $\hat{\mathbf{Z}} = \text{diag}(\mathbf{Z}^*)$, a matrix that contains the diagonal elements of the matrix $\mathbf{Z}^*$. Also, construct $\hat{\mathbf{V}}$ as follows,

$$\hat{\mathbf{V}}_{ij} = \begin{cases} \mathbf{V}^*_{ij} + \frac{\mathbf{Z}^*_{ij}}{2} & \text{if } i \neq j \\ \mathbf{V}^*_{jj} & \text{otherwise.} \end{cases}$$

Then, we have that $\mathbf{\Theta}^* = \hat{\mathbf{Z}} + \hat{\mathbf{V}} + \hat{\mathbf{V}}^T$. Thus, $(\mathbf{\Theta}^*, \hat{\mathbf{Z}}, \hat{\mathbf{V}})$ is a feasible solution to (6). We now show that $(\mathbf{\Theta}^*, \hat{\mathbf{Z}}, \hat{\mathbf{V}})$ has a smaller objective than $(\mathbf{\Theta}^*, \mathbf{Z}^*, \mathbf{V}^*)$ in (6), giving us a contradiction. Note that

$$\begin{aligned} \lambda_1\|\hat{\mathbf{Z}} - \text{diag}(\hat{\mathbf{Z}})\|_1 + \lambda_2\|\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}})\|_1 &= \lambda_2\|\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}})\|_1 \\ &= \lambda_2 \textstyle\sum_{i\neq j} |\mathbf{V}^*_{ij} + \frac{\mathbf{Z}^*_{ij}}{2}| \\ &\leq \lambda_2\|\mathbf{V}^* - \text{diag}(\mathbf{V}^*)\|_1 + \frac{\lambda_2}{2}\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1, \end{aligned}$$

and

$$\begin{aligned} & \lambda_3 \textstyle\sum_{j=1}^p \|(\hat{\mathbf{V}} - \text{diag}(\hat{\mathbf{V}}))_j\|_q \\ \leq\ & \lambda_3 \textstyle\sum_{j=1}^p \|(\mathbf{V}^* - \text{diag}(\mathbf{V}^*))_j\|_q + \frac{\lambda_3}{2} \textstyle\sum_{j=1}^p \|(\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*))_j\|_q \\ \leq\ & \lambda_3 \textstyle\sum_{j=1}^p \|(\mathbf{V}^* - \text{diag}(\mathbf{V}^*))_j\|_q + \frac{\lambda_3}{2}\|\mathbf{Z}^* - \text{diag}(\mathbf{Z}^*)\|_1, \end{aligned}$$

where the last inequality follows from the fact that for any vector $\mathbf{x} \in \mathbb{R}^p$ and $q \geq 1$, $\|\mathbf{x}\|_q$ is a nonincreasing function of q (Gentle, 2007).

Summing up the above inequalities, we get that

$$\begin{array}{rl} \lambda_1\|\hat{\mathbf{Z}} - \mathrm{diag}(\hat{\mathbf{Z}})\|_1 + \lambda_2\|\hat{\mathbf{V}} - \mathrm{diag}(\hat{\mathbf{V}})\|_1 + \lambda_3 \sum_{j=1}^p \|(\hat{\mathbf{V}} - \mathrm{diag}(\hat{\mathbf{V}}))_j\|_q & \leq \\ \frac{\lambda_2+\lambda_3}{2}\|\mathbf{Z}^* - \mathrm{diag}(\mathbf{Z}^*)\|_1 + \lambda_2\|\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*)\|_1 + \lambda_3 \sum_{j=1}^p \|(\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*))_j\|_q & < \\ \lambda_1\|\mathbf{Z}^* - \mathrm{diag}(\mathbf{Z}^*)\|_1 + \lambda_2\|\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*)\|_1 + \lambda_3 \sum_{j=1}^p \|(\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*))_j\|_q, & \end{array}$$

where the last inequality uses the assumption that $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$. We arrive at a contradiction and therefore the result holds. ■

Proof of Lemma 4

Proof Let $(\mathbf{\Theta}^*, \mathbf{Z}^*, \mathbf{V}^*)$ be the solution to (6) and suppose $\mathbf{V}^*$ is not a diagonal matrix. Let $\hat{\mathbf{V}} = \mathrm{diag}(\mathbf{V}^*)$, a diagonal matrix that contains the diagonal elements of $\mathbf{V}^*$. Also construct $\hat{\mathbf{Z}}$ as follows,

$$\hat{\mathbf{Z}}_{ij} = \begin{cases} \mathbf{Z}^*_{ij} + \mathbf{V}^*_{ij} + \mathbf{V}^*_{ji} & \text{if } i \neq j \\ \mathbf{Z}^*_{ij} & \text{otherwise.} \end{cases}$$

Then, we have that $\mathbf{\Theta}^* = \hat{\mathbf{V}} + \hat{\mathbf{V}}^T + \hat{\mathbf{Z}}$. We now show that $(\mathbf{\Theta}^*, \hat{\mathbf{Z}}, \hat{\mathbf{V}})$ has a smaller objective value than $(\mathbf{\Theta}^*, \mathbf{Z}^*, \mathbf{V}^*)$ in (6), giving us a contradiction. We start by noting that

$$\begin{aligned} \lambda_1\|\hat{\mathbf{Z}} - \mathrm{diag}(\hat{\mathbf{Z}})\|_1 + \lambda_2\|\hat{\mathbf{V}} - \mathrm{diag}(\hat{\mathbf{V}})\|_1 &= \lambda_1\|\hat{\mathbf{Z}} - \mathrm{diag}(\hat{\mathbf{Z}})\|_1 \\ &\leq \lambda_1\|\mathbf{Z}^* - \mathrm{diag}(\mathbf{Z}^*)\|_1 + 2\lambda_1\|\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*)\|_1. \end{aligned}$$

By Holder's Inequality, we know that $\mathbf{x}^T\mathbf{y} \leq \|\mathbf{x}\|_q\|\mathbf{y}\|_s$ where $\frac{1}{s} + \frac{1}{q} = 1$ and $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{p-1}$. Setting $\mathbf{y} = \mathrm{sign}(\mathbf{x})$, we have that $\|\mathbf{x}\|_1 \leq (p-1)^{\frac{1}{s}}\|\mathbf{x}\|_q$. Consequently,

$$\frac{\lambda_3}{(p-1)^{\frac{1}{s}}}\|\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*)\|_1 \leq \lambda_3 \sum_{j=1}^p \|(\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*))_j\|_q.$$

Combining these results, we have that

$$\begin{aligned} &\lambda_1\|\hat{\mathbf{Z}} - \mathrm{diag}(\hat{\mathbf{Z}})\|_1 + \lambda_2\|\hat{\mathbf{V}} - \mathrm{diag}(\hat{\mathbf{V}})\|_1 + \lambda_3 \sum_{j=1}^p \|(\hat{\mathbf{V}} - \mathrm{diag}(\hat{\mathbf{V}}))_j\|_q \\ &\leq \lambda_1\|\mathbf{Z}^* - \mathrm{diag}(\mathbf{Z}^*)\|_1 + 2\lambda_1\|\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*)\|_1 \\ &< \lambda_1\|\mathbf{Z}^* - \mathrm{diag}(\mathbf{Z}^*)\|_1 + \left(\lambda_2 + \frac{\lambda_3}{(p-1)^{\frac{1}{s}}}\right)\|\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*)\|_1 \\ &\leq \lambda_1\|\mathbf{Z}^* - \mathrm{diag}(\mathbf{Z}^*)\|_1 + \lambda_2\|\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*)\|_1 + \lambda_3 \sum_{j=1}^p \|(\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*))_j\|_q, \end{aligned}$$

where we use the assumption that $\lambda_1 < \frac{\lambda_2}{2} + \frac{\lambda_3}{2(p-1)^{\frac{1}{s}}}$. This leads to a contradiction. ■

Proof of Lemma 6

In this proof, we consider the case when $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$. A similar proof technique can be used to prove the case when $\lambda_1 < \frac{\lambda_2+\lambda_3}{2}$.

Proof Let $f(\mathbf{\Theta}, \mathbf{V}, \mathbf{Z})$ denote the objective of (6) with $q = 1$, and $(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*)$ the optimal solution. By Lemma 3, the assumption that $\lambda_1 > \frac{\lambda_2+\lambda_3}{2}$ implies that $\mathbf{Z}^*$ is a diagonal matrix. Now let $\hat{\mathbf{V}} = \frac{1}{2}\left(\mathbf{V}^* + (\mathbf{V}^*)^T\right)$. Then

$$\begin{aligned}
& f(\mathbf{\Theta}^*, \hat{\mathbf{V}}, \mathbf{Z}^*) \\
= \; & -\log\det \mathbf{\Theta}^* + \langle \mathbf{\Theta}^*, \mathbf{S} \rangle + \lambda_1 \|\mathbf{Z}^* - \mathrm{diag}(\mathbf{Z}^*)\|_1 + (\lambda_2 + \lambda_3)\|\hat{\mathbf{V}} - \mathrm{diag}(\hat{\mathbf{V}})\|_1 \\
= \; & -\log\det \mathbf{\Theta}^* + \langle \mathbf{\Theta}^*, \mathbf{S} \rangle + \frac{\lambda_2 + \lambda_3}{2}\|\mathbf{V}^* + \mathbf{V}^{*T} - \mathrm{diag}(\mathbf{V}^* + \mathbf{V}^{*T})\|_1 \\
\leq \; & -\log\det \mathbf{\Theta}^* + \langle \mathbf{\Theta}^*, \mathbf{S} \rangle + (\lambda_2 + \lambda_3)\|\mathbf{V}^* - \mathrm{diag}(\mathbf{V}^*)\|_1 \\
= \; & f(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*) \\
\leq \; & f(\mathbf{\Theta}^*, \hat{\mathbf{V}}, \mathbf{Z}^*),
\end{aligned}$$

where the last inequality follows from the assumption that $(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*)$ solves (6). By strict convexity of f, this means that $\mathbf{V}^* = \hat{\mathbf{V}}$, i.e., $\mathbf{V}^*$ is symmetric. This implies that

$$\begin{aligned}
f(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*) &= -\log\det \mathbf{\Theta}^* + \langle \mathbf{\Theta}^*, \mathbf{S} \rangle + \frac{\lambda_2 + \lambda_3}{2}\|\mathbf{V}^* + \mathbf{V}^{*T} - \mathrm{diag}(\mathbf{V}^* + \mathbf{V}^{*T})\|_1 \\
&= -\log\det \mathbf{\Theta}^* + \langle \mathbf{\Theta}^*, \mathbf{S} \rangle + \frac{\lambda_2 + \lambda_3}{2}\|\mathbf{\Theta}^* - \mathrm{diag}(\mathbf{\Theta}^*)\|_1 \qquad (21) \\
&= g(\mathbf{\Theta}^*),
\end{aligned}$$

where $g(\mathbf{\Theta})$ is the objective of the graphical lasso optimization problem, evaluated at $\mathbf{\Theta}$, with tuning parameter $\frac{\lambda_2+\lambda_3}{2}$. Suppose that $\tilde{\mathbf{\Theta}}$ minimizes $g(\mathbf{\Theta})$, and $\mathbf{\Theta}^* \neq \tilde{\mathbf{\Theta}}$. Then, by (21) and strict convexity of g, $g(\mathbf{\Theta}^*) = f(\mathbf{\Theta}^*, \mathbf{V}^*, \mathbf{Z}^*) \leq f(\tilde{\mathbf{\Theta}}, \tilde{\mathbf{\Theta}}/2, \mathbf{0}) = g(\tilde{\mathbf{\Theta}}) < g(\mathbf{\Theta}^*)$, giving us a contradiction. Thus it must be that $\tilde{\mathbf{\Theta}} = \mathbf{\Theta}^*$.

■

Appendix D. Simulation Study for Hub Covariance Graph

In this section, we present the results for the simulation study described in Section 4.2 with $n = 100$, $p = 200$, and $|\mathcal{H}| = 4$. We calculate the proportion of correctly estimated hub nodes with $r = 40$. The results are shown in Figure 10. As we can see from Figure 10, our proposal outperforms Bien and Tibshirani (2011). In particular, we can see from Figure 10(c) that Bien and Tibshirani (2011) fails to identify hub nodes.

Appendix E. Run Time Study for the ADMM algorithm for HGL

In this section, we present a more extensive run time study for the ADMM algorithm for HGL. We ran experiments with $p = 100, 200, 300$ and with $n = p/2$ on a 2.26GHz Intel Core

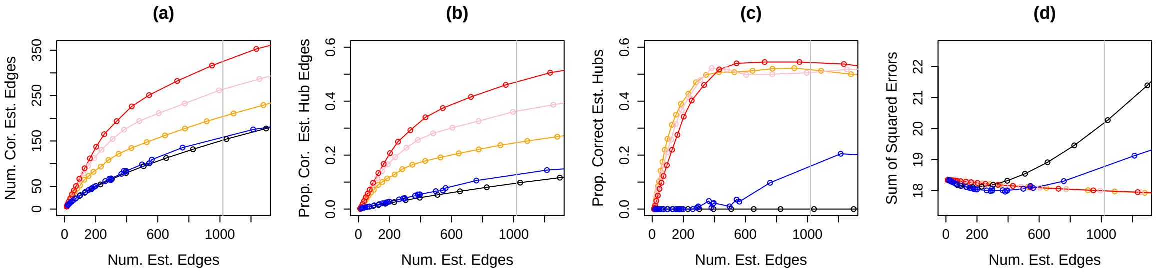

Figure 10: Covariance graph simulation with $n = 100$ and $p = 200$. Details of the axis labels are as in Figure 3. The colored lines correspond to the proposal of Xue et al. (2012) (——); HCG with $\lambda_3 = 1$ (——), $\lambda_3 = 1.5$ (——), and $\lambda_3 = 2$ (——); and the proposal of Bien and Tibshirani (2011) (——).

2 Duo machine. Results averaged over 10 replications are displayed in Figures 11(a)-(b), where the panels depict the run time and number of iterations required for the algorithm to converge, as a function of λ_1, with $\lambda_2 = 0.5$ and $\lambda_3 = 2$ fixed. The number of iterations required for the algorithm to converge is computed as the total number of iterations in Step 2 of Algorithm 1. We see from Figure 11(a) that as p increases from 100 to 300, the run times increase substantially, but never exceed several minutes. Note that these results are without using the block diagonal condition in Theorem 1.

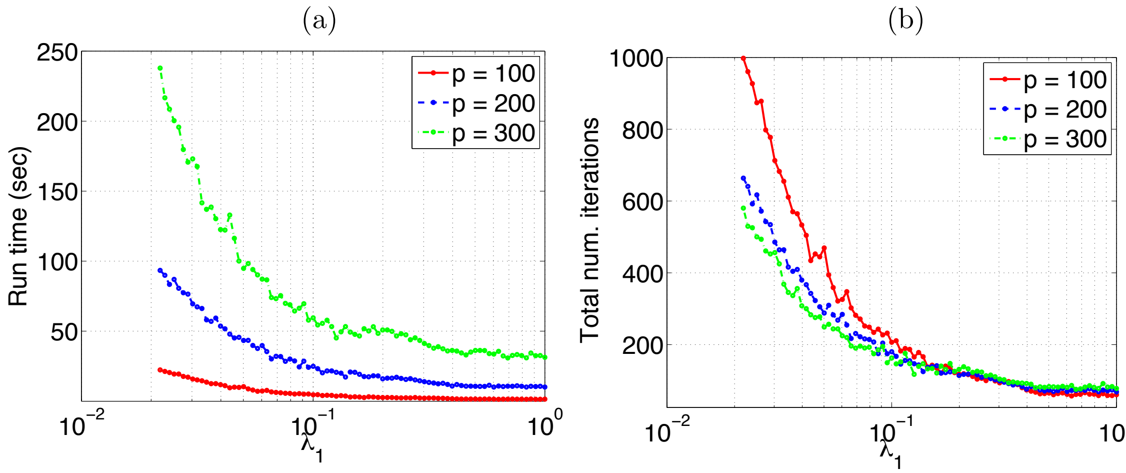

Figure 11: (a): Run time (in seconds) of the ADMM algorithm for HGL, as a function of λ_1, for fixed values of λ_2 and λ_3. (b): The total number of iterations required for the ADMM algorithm for HGL to converge, as a function of λ_1. All results are averaged over 10 simulated data sets. These results are without using the block diagonal condition in Theorem 1.

Appendix F. Update for $\mathbf{\Theta}$ in Step 2(a)i for Binary Ising Model using Barzilai-Borwein Method

We consider updating $\mathbf{\Theta}$ in Step 2(a)i of Algorithm 1 for binary Ising model. Let

$$h(\mathbf{\Theta}) = -\sum_{j=1}^{p}\sum_{j'=1}^{p} \theta_{jj'}(\mathbf{X}^T\mathbf{X})_{jj'} + \sum_{i=1}^{p}\sum_{j=1}^{p} \log\left(1+\exp\left[\theta_{jj} + \sum_{j'\neq j}\theta_{jj'}x_{ij'}\right]\right)$$
$$+\frac{\rho}{2}\|\mathbf{\Theta} - \tilde{\mathbf{\Theta}} + \mathbf{W}_1\|_F^2.$$

Then, the optimization problem for Step 2(a)i of Algorithm 1 is

$$\underset{\mathbf{\Theta}\in\mathcal{S}}{\text{minimize}} \quad h(\mathbf{\Theta}), \tag{22}$$

where $\mathcal{S} = \{\mathbf{\Theta} : \mathbf{\Theta} = \mathbf{\Theta}^T\}$. In solving (22), we will treat $\mathbf{\Theta} \in \mathcal{S}$ as an implicit constraint.

The Barzilai-Borwein method is a gradient descent method with the step-size chosen to mimic the secant condition of the BFGS method (see, e.g., Barzilai and Borwein, 1988; Nocedal and Wright, 2006). The convergence of the Barzilai-Borwein method for unconstrained minimization using a non-monotone line search was shown in Raydan (1997). Recent convergence results for a quadratic cost function can be found in Dai (2013). To implement the Barzilai-Borwein method, we need to evaluate the gradient of $h(\mathbf{\Theta})$. Let $\nabla h(\mathbf{\Theta})$ be a $p \times p$ matrix, where the (j, j') entry is the gradient of $h(\mathbf{\Theta})$ with respect to $\theta_{jj'}$, computed under the constraint $\mathbf{\Theta} \in \mathcal{S}$, that is, $\theta_{jj'} = \theta_{j'j}$. Then,

$$(\nabla h(\mathbf{\Theta}))_{jj} = -(\mathbf{X}^T\mathbf{X})_{jj} + \sum_{i=1}^{n}\left[\frac{\exp(\theta_{jj} + \sum_{j'\neq j}\theta_{jj'}x_{ij'})}{1+\exp(\theta_{jj} + \sum_{j'\neq j}\theta_{jj'}x_{ij'})}\right] + \rho(\theta_{jj} - \tilde{\theta}_{jj} + (\mathbf{W}_1)_{jj}),$$

and

$$(\nabla h(\mathbf{\Theta}))_{jj'} = -2(\mathbf{X}^T\mathbf{X})_{jj} + 2\rho(\theta_{jj'} - \tilde{\theta}_{jj'} + (\mathbf{W}_1)_{jj'})$$
$$+\sum_{i=1}^{n}\left[\frac{x_{ij'}\exp(\theta_{jj} + \sum_{j'\neq j}\theta_{jj'}x_{ij'})}{1+\exp(\theta_{jj} + \sum_{j'\neq j}\theta_{jj'}x_{ij'})} + \frac{x_{ij}\exp(\theta_{j'j'} + \sum_{j\neq j'}\theta_{jj'}x_{ij})}{1+\exp(\theta_{j'j'} + \sum_{j\neq j'}\theta_{jj'}x_{ij})}\right].$$

A simple implementation of the Barzilai-Borwein algorithm for solving (22) is detailed in Algorithm 2. We note that the Barzilai-Borwein algorithm can be improved (see, e.g., Barzilai and Borwein, 1988; Wright et al., 2009). We leave such improvement for future work.

Algorithm 2 Barzilai-Borwein Algorithm for Solving (22).

```
1. Initialize the parameters:
    (a) $\mathbf{\Theta}_1 = \mathbf{I}$ and $\mathbf{\Theta}_0 = 2\mathbf{I}$.
    (b) constant $\tau > 0$.
2. Iterate until the stopping criterion $\frac{\|\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1}\|_F^2}{\|\mathbf{\Theta}_{t-1}\|_F^2} \leq \tau$ is met, where $\mathbf{\Theta}_t$ is the value of $\mathbf{\Theta}$ obtained at the $t$th iteration:
    (a) $\alpha_t = \text{trace}\left[(\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1})^T(\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1})\right] / \text{trace}\left[(\mathbf{\Theta}_t - \mathbf{\Theta}_{t-1})^T(\nabla h(\mathbf{\Theta}_t) - \nabla h(\mathbf{\Theta}_{t-1}))\right]$.
    (b) $\mathbf{\Theta}_{t+1} = \mathbf{\Theta}_t - \alpha_t \nabla h(\mathbf{\Theta}_t)$.
```

References

G.I. Allen and Z. Liu. A log-linear graphical model for inferring genetic networks from high-throughput sequencing data. *IEEE International Conference on Bioinformatics and Biomedicine*, 2012.

A.L. Barabási. Scale-free networks: A decade and beyond. *Science*, 325:412–413, 2009.

A.L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286: 509–512, 1999.

J. Barzilai and J.M. Borwein. Two-point step size gradient methods. *IMA Journal of Numerical Analysis*, 8:141–148, 1988.

P.J. Bickel and E. Levina. Regularized estimation of large covariance matrices. *Annals of Statistics*, 36(1):199–227, 2008.

J. Bien and R. Tibshirani. Sparse estimation of a covariance matrix. *Biometrika*, 98(4): 807–820, 2011.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the ADMM. *Foundations and Trends in Machine Learning*, 3(1): 1–122, 2010.

T. Cai and W. Liu. Adaptive thresholding for sparse covariance matrix estimation. *Journal of the American Statistical Association*, 106(494):672–684, 2011.

A. Cardoso-Cachopo. 2009. "http://web.ist.utl.pt/acardoso/datasets/".

S. Chaudhuri, M. Drton, and T. Richardson. Estimation of a covariance matrix with zeros. *Biometrika*, 94:199–216, 2007.

Y. Dai. A new analysis on the Barzilai-Borwein gradient method. *Journal of the Operations Research Society of China*, 1(2):187–198, 2013.

P. Danaher, P. Wang, and D.M. Witten. The joint graphical lasso for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B*, 76 (2):373–397, 2014.

A. Defazio and T.S. Caetano. A convex formulation for learning scale-free network via submodular relaxation. *Advances in Neural Information Processing Systems*, 2012.

M. Drton and T.S. Richardson. A new algorithm for maximum likelihood estimation in Gaussian graphical models for marginal independence. *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence*, pages 184–191, 2003.

M. Drton and T.S. Richardson. Graphical methods for efficient likelihood inference in Gaussian covariance models. *Journal of Machine Learning Research*, 9:893–914, 2008.

J. Eckstein. Augmented Lagrangian and alternating direction methods for convex optimization: A tutorial and some illustrative computational results. *RUTCOR Research Reports*, 32, 2012.

J. Eckstein and D.P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(3, Ser. A):293–318, 1992.

N. El Karoui. Operator norm consistent estimation of large-dimensional sparse covariance matrices. *The Annals of Statistics*, 36(6):2717–2756, 2008.

P. Erdős and A. Rényi. On random graphs I. *Publ. Math. Debrecen*, 6:290–297, 1959.

H. Firouzi and A.O. Hero. Local hub screening in sparse correlation graphs. *Proceedings of SPIE, volume 8858, Wavelets and Sparsity XV, 88581H*, 2013.

R. Foygel and M. Drton. Extended Bayesian information criteria for Gaussian graphical models. *Advances in Neural Information Processing Systems*, 2010.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9:432–441, 2007.

J. E. Gentle. *Matrix Algebra: Theory, Computations, and Applications in Statistics.* Springer, New York, 2007.

J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint structure estimation for categorical Markov networks. Submitted, available at http://www.stat.lsa.umich.edu/˜elevina, 2010.

J. Guo, E. Levina, G. Michailidis, and J. Zhu. Asymptotic properties of the joint neighborhood selection method for estimating categorical Markov networks. `arXiv: math.PR/0000000`, 2011.

D. Hao, C. Ren, and C. Li. Revisiting the variation of clustering coefficient of biological networks suggests new modular structure. *BMC System Biology*, 6(34):1–10, 2012.

A. Hero and B. Rajaratnam. Hub discovery in partial correlation graphs. *IEEE Transactions on Information Theory*, 58:6064–6078, 2012.

H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods. *Journal of Machine Learning Research*, 10:883–906, 2009.

R.A. Horn and C.R. Johnson. *Matrix Analysis.* Cambridge University Press, New York, NY, 1985.

H. Jeong, S.P. Mason, A.L. Barabási, and Z.N. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41–42, 2001.

S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using ℓ_1-regularization. *Advances in Neural Information Processing Systems*, 2007.

L. Li, D. Alderson, J.C. Doyle, and W. Willinger. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics*, 2(4):431–523, 2005.

F. Liljeros, C.R. Edling, L.A.N. Amaral, H.E. Stanley, and Aberg Y. The web of human sexual contacts. *Nature*, 411:907–908, 2001.

Q. Liu and A.T. Ihler. Learning scale free networks by reweighed ℓ_1 regularization. *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics*, 15: 40–48, 2011.

S. Ma, L. Xue, and H. Zou. Alternating direction methods for latent variable Gaussian graphical model selection. *Neural Computation*, 2013.

Maglott et al. Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, 33(D):54–58, 2004.

K.V. Mardia, J. Kent, and J.M. Bibby. *Multivariate Analysis.* Academic Press, 1979.

R. Mazumder and T. Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13:781–794, 2012.

N. Meinshausen and P. Bühlmann. High dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34(3):1436–1462, 2006.

N. Meinshausen and P. Bühlmann. Stability selection (with discussion). *Journal of the Royal Statistical Society, Series B*, 72:417–473, 2010.

K. Mohan, P. London, M. Fazel, D.M. Witten, and S.-I. Lee. Node-based learning of Gaussian graphical models. *Journal of Machine Learning Research*, 15:445–488, 2014.

M.E.J. Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of the United States of America*, 98:404–409, 2000.

J. Nocedal and S.J. Wright. *Numerical Optimization.* Springer, 2006.

J. Peng, P. Wang, N. Zhou, and J. Zhu. Partial correlation estimation by joint sparse regression model. *Journal of the American Statistical Association*, 104(486):735–746, 2009.

Rappaport et al. MalaCards: an integrated compendium for diseases and their annotation. *Database (Oxford)*, 2013.

P. Ravikumar, M.J. Wainwright, and J.D. Lafferty. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 38(3):1287–1319, 2010.

P. Ravikumar, M.J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

M. Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7:26–33, 1997.

A. Rothman, P.J. Bickel, E. Levina, and J. Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

A. Rothman, E. Levina, and J. Zhu. Generalized thresholding of large covariance matrices. *Journal of the American Statistical Association*, 104:177–186, 2009.

N. Simon, J.H. Friedman, T. Hastie, and R. Tibshirani. A sparse-group lasso. *Journal of Computational and Graphical Statistics*, 22(2):231–245, 2013.

R. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1 (2):146–160, 1972.

Verhaak et al. Integrated genomic analysis identifies clinically relevant subtypes of glioblastoma characterized by abnormalities in PDGFRA, IDH1, EGFR, and NF1. *Cancer Cell*, 17(1):98–110, 2010.

D.M. Witten, J.H. Friedman, and N. Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900, 2011.

S.J. Wright, R.D. Nowak, and M. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Transactions on Signal Processing*, 57(7):2479–2493, 2009.

L. Xue, S. Ma, and H. Zou. Positive definite ℓ_1 penalized estimation of large covariance matrices. *Journal of the American Statistical Association*, 107(500):1480–1491, 2012.

E. Yang, G.I. Allen, Z. Liu, and P.K. Ravikumar. Graphical models via generalized linear models. *Advances in Neural Information Processing Systems*, 2012a.

S. Yang, Z. Pan, X. Shen, P. Wonka, and J. Ye. Fused multiple graphical lasso. `arXiv:1209.2139 [cs.LG]`, 2012b.

M. Yuan. Efficient computation of ℓ_1 regularized estimates in Gaussian graphical models. *Journal of Computational and Graphical Statistics*, 17(4):809–826, 2008.

M. Yuan and Y. Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(10):19–35, 2007a.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2007b.

H. Zou, T. Hastie, and R. Tibshirani. On the "degrees of freedom" of the lasso. *The Annals of Statistics*, 35(5):2173–2192, 2007.