Journal of Machine Learning Research 18 (2018) 1-48 Submitted 9/16; Revised 4/17; Published 4/18

Estimation of Graphical Models through Structured Norm Minimization

Davoud Ataee Tarzanagh TARZANAGH@UFL.EDU
Department of Mathematics
UF Informatics Institute
University of Florida
Gainesville, FL 32611-8105, USA

George Michailidis GMICHAIL@UFL.EDU
Department of Statistics
UF Informatics Institute
University of Florida
Gainesville, FL 32611-8545, USA

Editor: Bert Huang

Abstract

Estimation of Markov Random Field and covariance models from high-dimensional data represents a canonical problem that has received a lot of attention in the literature. A key assumption, widely employed, is that of *sparsity* of the underlying model. In this paper, we study the problem of estimating such models exhibiting a more intricate structure comprising simultaneously of *sparse, structured sparse* and *dense* components. Such structures naturally arise in several scientific fields, including molecular biology, finance and political science. We introduce a general framework based on a novel structured norm that enables us to estimate such complex structures from high-dimensional data. The resulting optimization problem is convex and we introduce a linearized multi-block alternating direction method of multipliers (ADMM) algorithm to solve it efficiently. We illustrate the superior performance of the proposed framework on a number of synthetic data sets generated from both random and structured networks. Further, we apply the method to a number of real data sets and discuss the results.

Keywords: Markov Random Fields, Gaussian covariance graph model, structured sparse norm, regularization, alternating direction method of multipliers (ADMM), convergence.

1. Introduction

There is a substantial body of literature on methods for estimating network structures from high-dimensional data, motivated by important biomedical and social science applications; see Barabási and Albert (1999); Liljeros et al. (2001); Robins et al. (2007); Guo et al. (2011a); Danaher et al. (2014); Friedman et al. (2008); Tan et al. (2014); Guo et al. (2015). Two powerful formalisms have been employed for this task, the Markov Random Field (MRF) model and the Gaussian covariance graph model (GCGM). The former captures statistical conditional dependence relationships amongst random variables that correspond to the network nodes, while the latter to marginal associations. Since in most applications the number of model parameters to be estimated far exceeds the available sample size, the assumption

of sparsity is made and imposed through regularization. An ℓ_1 penalty on the parameters encoding the network edges is the most common choice; see Friedman et al. (2008); Karoui (2008); Cai and Liu (2011); Xue et al. (2012), which can also be interpreted from the Bayesian perspective as using an independent double-exponential prior distribution on each edge parameter. Consequently, this approach encourages sparse uniform network structures that may not be the most suitable choice for many real world applications, which in turn have *hub* nodes or *dense subgraphs*. As argued in Barabási and Albert (1999); Liljeros et al. (2001); Newman (2001); Li et al. (2005); Fortunato (2010); Newman (2012) many networks exhibit different structures at different scales. An example includes a densely connected subgraph, also known as a *community* in the social networks literature. Such structures in social interaction networks may correspond to groups of people sharing common interests or being co-located (Traud et al., 2011; Newman and Girvan, 2004), while in biological systems to groups of proteins responsible for regulating or synthesizing chemical products (Guimera and Amaral, 2005; Lewis et al., 2010; see, Figure 3 for an example). Hence, in many applications, simple sparsity or alternatively, a dense structure fails to capture salient features of the true underlying mechanism that gave rise to the available data.

In this paper, we introduce a framework based on a novel structured sparse norm that allows us to recover such complex structures. Specifically, we consider Markov Random Field and covariance models where the parameter of interest, Θ can be expressed as the superposition of sparse, structured sparse and dense components as follows:

$$\Theta \;=\; \underbrace{Z_1 + Z_1^\top}_{\text{sparse part}} \;+\; \underbrace{Z_2 + Z_2^\top + \cdots + Z_n + Z_n^\top}_{\text{structured sparse part}} \;+\; \underbrace{E}_{\text{dense part}}\;, \tag{1}$$

where Z_1 is a sparse matrix, $Z_2, \ldots, Z_n$ are the set of $n-1$ structured sparse matrices (see, Figure 3 for an example of such structured matrices), and E is a dense matrix having possibly very many small, non-zero entries. As shown in Figure 3, the elements of Z_1 represent edges between non-structured nodes, and the non-zero parts of structured matrices $Z_2, \ldots, Z_n$ correspond to densely connected subgraphs (communities).

We elaborate more on the decomposition proposed above. We start by discussing on the sparse and structured sparse component and then elaborate on the dense component. Traditional sparse (lasso Tibshirani, 1996; Friedman et al., 2008) and group sparse (group lasso Yuan and Lin, 2007; Jacob et al., 2009; Obozinski et al., 2011) are tailor-made to estimate and recover sparse and structured sparse model structures, respectively. However, these methods can not accommodate different structures, unless users specify a *priori* the structure of interest (e.g. hub nodes and sparse components), thus severely limiting their application scope. On the other hand, the general framework introduced, is capable of estimating from high-dimensional data, *groups with overlaps*, *hubs* and *dense subgraphs*, with the size and location of such structures *not known a priori*.

Next, we discuss the role of the dense component E. In many applications, the data generation mechanism may correspond to a true sparse or structured sparse structure, "corrupted" by a dense component comprising of possible many small entries. A simple example of such a generating mechanism in linear models would have the regression coefficient being sparse with a few large entries and a more dense component having possibly many small, nonzero entries. In such instances, a pure sparse model formulation may not perform particularly well due to the presence of the dense component and may require very careful

tuning to recover the sparse component of interest. This line of reasoning is also adopted in Chernozhukov et al. (2017). Note however, that the model may also be used in settings where there is a significant dense component; however, as discussed in Chernozhukov et al. (2017) recovery of the individual component is not guaranteed. Hence, in this work we adopt the viewpoint that E represents a small "perturbation" of the sparse+structured sparse structure. To achieve these goals, it leverages a new structured norm that is used as the regularization term of the corresponding objective function.

The resulting optimization problem is solved through a multi-block ADMM algorithm. A key technical innovation is the development of a linearized ADMM algorithm that avoids introducing auxiliary variables which is a common strategy in the literature. We establish the global convergence of the proposed algorithm and illustrate its efficiency through numerical experimentation. The algorithm takes advantage of the special structure of the problem formulation and thus is suitable for large instances of the problem. To the best of our knowledge, this is the first work that gives global convergence guarantees for linearized multi-block ADMM with Gauss-Seidel updates, which is of interest in its own accord.

The remainder of the paper is organized as follows: In Section 2, we present the new structured norm used as the regularization term in the objective function of the Markov Random Field, covariance graph, regression and vector auto-regression models. In Section 3, we introduce an efficient multi-block ADMM algorithm to solve the problem, and provide the convergence analysis of the algorithm. In Section 4, we illustrate the proposed framework on a number of synthetic and real data sets, while some concluding remarks are drawn in Section 5.

2. A General Framework for Learning under Structured Sparsity

We start by introducing key definitions and associated notation.

2.1 Symmetric Structured Overlap Norm

Let X be an $m \times p$ data matrix, Θ be a $p \times p$ symmetric matrix containing the parameters of interest of the statistical loss function $\mathcal{G}(X, \Theta)$. The most popular assumption used in the literature is that Θ is *sparse* and can be successfully recovered from high-dimensional data by solving the following optimization problem

$$\underset{\Theta \in \mathcal{S}}{\text{minimize}} \quad \mathcal{G}(X, \Theta) + \lambda \big\|\Theta\big\|_1, \tag{2}$$

where $\mathcal{S}$ is some set depending on the loss function; λ is s a non-negative regularization constant; and $\|.\|_1$ denotes the ℓ_1 norm or the sum of the absolute values of the matrix elements.

To explicitly model different structures in the parameter Θ, we introduce the following *symmetric structured overlap norm (SSON)*:

Definition 1 ***(Symmetric Structured Overlap Norm)****. Let Θ be a $p \times p$ symmetric matrix containing the model parameters of interest. The symmetric structured overlap norm*

for a set of partitioned matrices $Z_1, \ldots, Z_n$ is given by,

$$\begin{aligned}
\underset{Z_1,\ldots,Z_n,\ E}{\text{minimize}} \quad \Omega(\Theta, Z_1, \ldots, Z_n, E) \ &:= \ \lambda_1 \|Z_1 - \text{diag}(Z_1)\|_1 \\
&+ \ \sum_{i=2}^{n} \hat{\lambda}_i \|Z_i - \text{diag}(Z_i)\|_1 + \lambda_i \sum_{j=1}^{l_i} \|(Z_i - \text{diag}(Z_i))_j\|_F \\
&+ \ \frac{\lambda_e}{2} \|E\|_F^2, \\
\Theta \ &= \ \sum_{i=1}^{n} \big(Z_i + Z_i^\top\big) + E, \qquad (3)
\end{aligned}$$

where $\{\lambda_i\}_{i=1}^n$ and $\{\hat{\lambda}_i\}_{i=2}^n$ are nonnegative regularization constants; l_i is the number of blocks of the partitioned matrix Z_i; $(Z_i - \text{diag}(Z_i))_j$ is the jth block of the partitioned matrix Z_i; E is an unstructured noise matrix; $\|.\|_1$ denotes the ℓ_1 norm or the sum of the absolute values of the matrix elements; and $\|.\|_F$ the Frobenius norm.

We note that the overlap norm defined by Mohan et al. (2012); Tan et al. (2014) encourages the recovery of matrices that can be expressed as a union of few rows and the corresponding columns (i.e. hub nodes). However, SSON represents a new symmetric and significantly more general variant of the overlap norm that promotes matrices that can be expressed as the sum of symmetric structured matrices. Moreover, unlike the previous group sparsity and the latent group lasso discussed in Yuan and Lin (2007); Jacob et al. (2009); Obozinski et al. (2011) that require users to specify structures of interest *a priori*, the SSON achieves a similar objective in an *agnostic manner*, relying only on how well such structures fit the observed data.

In many applications, such as regression models, we are interested in modeling different structures in a parameter vector θ. In these cases, we have the following definition as a special case of SSON:

Definition 2 *Let θ be a $p \times 1$ vector containing the model parameters of interest. The structured overlap norm for a set of partitioned vectors $z_1, \ldots, z_n$ is given by,*

$$\begin{aligned}
\underset{z_1,\ldots,z_n,\ e}{\text{minimize}} \quad \omega(\theta, z_1, \ldots, z_n, e) \ &:= \ \lambda_1 \|z_1\|_1 + \sum_{i=2}^{n} \hat{\lambda}_i \|z_i\|_1 + \lambda_i \sum_{j=1}^{l_i} \|z_{ij}\|_2 + \frac{\lambda_e}{2} \|e\|_2^2, \\
\theta \ &= \ z_1 + z_2 + \cdots + z_n + e, \qquad (4)
\end{aligned}$$

where $\{\lambda_i\}_{i=1}^n$ and $\{\hat{\lambda}_i\}_{i=2}^n$ are nonnegative regularization constants; l_i is the number of blocks of the partitioned vector z_i; z_{i_j} is the jth block of the partitioned vector z_i (see, Figure 1); e is an unstructured noise vector; $\|.\|_1$ denotes the ℓ_1 norm or the sum of the absolute values of the vector elements; and $\|.\|_2$ the two norm.

Remark 3 *In the formulation of the problem, λ_1, $\{\hat{\lambda}_2, \ldots, \hat{\lambda}_n, \lambda_2, \ldots, \lambda_n\}$, and λ_e are tuning parameters corresponding to the sparse component Z_1, the structured components $\{Z_2, \ldots Z_n\}$ and the dense (noisy) component E, respectively. While the nonzero components may be clustered into groups, the nonzero groups may also be sparse. The latter can be achieved by (3) when $\{\hat{\lambda}_2, \ldots, \hat{\lambda}_n\}$ are positive constants.*

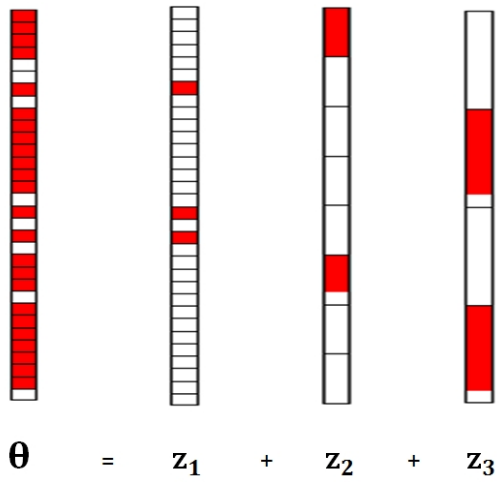


Figure 1: Decomposition of a vector θ into partitioned vectors z_1, z_2 and z_3, where z_1 is sparse, z_2 and z_3 are structured sparse vectors. White and red elements are zero and non-zero in the model parameter vector θ, respectively.

Remark 4 *The SSON admits the lasso (Tibshirani, 1996), the group lasso with overlaps (Jacob et al., 2009; Obozinski et al., 2011) and the ridge shrinkage (Hoerl and Kennard, 1970) methods as three extreme cases, by respectively setting* $\{\hat{\lambda}_2, \ldots, \hat{\lambda}_n, \lambda_2, \ldots, \lambda_n, \lambda_e\} \to \infty$, $\{\lambda_1, \hat{\lambda}_2, \ldots, \hat{\lambda}_n, \lambda_e\} \to \infty$, *and* $\{\lambda_1, \ldots, \lambda_n, \hat{\lambda}_2, \ldots, \hat{\lambda}_n\} \to \infty$[1].

Note that SSON is rather different from the sparse group lasso, which also uses a combination of ℓ_1 and ℓ_G penalization, where $\|.\|_G$ is the group lasso norm. The sparse group lasso penalty is $\bar{\omega}(\theta) = \lambda_1\|\theta\|_1 + \lambda_2\|\theta\|_G$, and thus the includes lasso and group lasso as extreme cases corresponding to $\lambda_2 = 0$ and $\lambda_1 = 0$, respectively. However, $\bar{\omega}(\theta)$ does not split θ into a sparse and a group sparse part and will produce a sparse solution as long as $\lambda_1 > 0$. Hence, the sparse group lasso method can be thought of as a sparsity-based method with additional shrinkage by $\|\theta\|_G$. The group sparsity processes data very differently from SSON and consequently has very different prediction risk behavior. The same argument illustrates the advantages of the proposed SSON penalty over the well-known elastic net penalty. The elastic net is a combination of lasso and ridge penalties (Zou and Hastie, 2005). However, the elastic net does not split θ into a sparse and a dense component. Our results show that SSON tends to perform no worse than, and often performs significantly better than ridge, lasso, group lasso or elastic net with penalty levels chosen by cross-validation.

Remark 5 *In order to encourage different structures in the parameter matrix* Θ*, we consider the Frobenius norm of blocks of partitioned matrices, which leads to recovery of dense subgraphs. Other values for the norm of such blocks are also possible; e.g. the* ℓ_∞ *norm.*

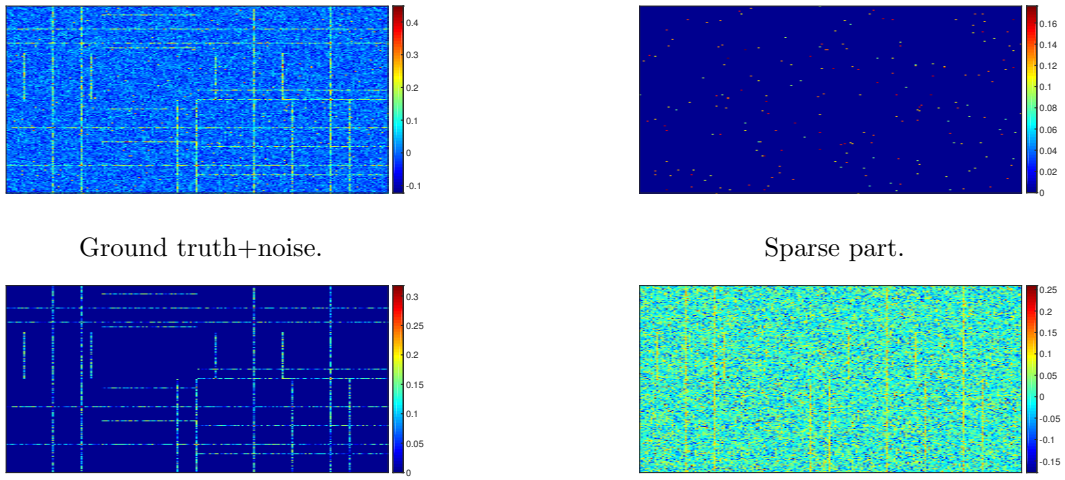
Remark 6 *The matrix* E *is an important component of the SSON framework.*

It enables to develop a convergent multi-block ADMM *to solve the problem of estimating a structured Markov Random Field or covariance model. Note that in general, a direct*

1. For example, with $\lambda_e \to \infty$, we set $\frac{\lambda_e}{2}\|E\|_F^2 = 0$ when $E = 0$, so the problem is well-defined.

extension of ADMM to multi-block convex minimization problems is not *necessarily convergent even without linearization of the corresponding subproblems as shown in Chen et al. (2016).*

From a performance standpoint, our results show that adding a ridge penalty term $\frac{\lambda_e}{2}\|E\|_F^2$ *to the structured norm is provably effective in correctly identifying the underlying structures in the presence of noisy data (Zou and Hastie, 2005; Chernozhukov et al., 2017) (see, Figure 2 for an example of decomposition* (1) *in the presense of noise for covariance matrix estimation.)*



Ground truth+noise. Sparse part.

Structured sparse part. Noisy part.

Figure 2: Heat map of the covariance matrix Θ_3 decomposed into sparse and structured sparse parts in the presence of noise, estimated by SSON using problem (11).

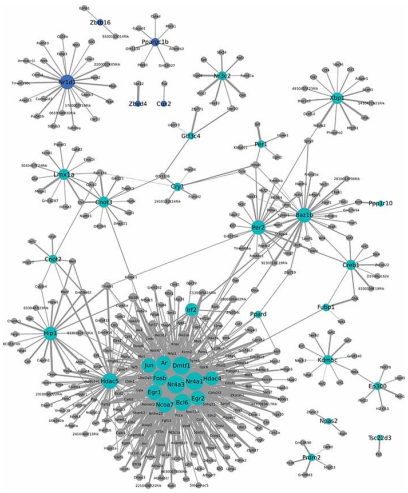
Next, we discuss the use of the SSON as a regularizer for maximum likelihood estimation of the following popular statistical models: (i) members of the Markov Random Field family including the Gaussian graphical model, the Gaussian graphical model with latent variables and the binary Ising model, (ii) the Gaussian covariance graph model and (iii) the classical regression and the vector auto-regression models. For the sake of completeness, we provide a complete, but succinct description of the corresponding models and the proposed regularization.

2.2 Structured Gaussian Graphical Models

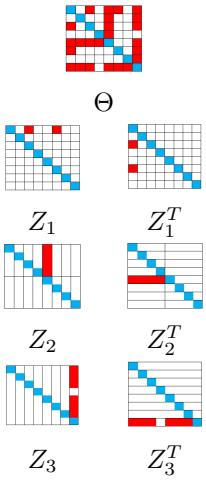
Let X be a data matrix consisting of p-dimensional samples from a zero mean Gaussian distribution,

$$x_1, \ldots, x_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma).$$

In order to obtain a sparse and interpretable estimate of the precision matrix Σ^{-1} that captures conditional dependence relationships, many authors have considered the well-known



(a) Postpartum NAC Gene Network (Zhao et al., 2014).



(b) Examples of partitioned matrices for the underlying network in (a).

Figure 3: The figure illustrates that block partitions through structured matrices could be set based on a desire for interpretability of the resulting estimated network structure. Panel (a) shows example of structured gene network, while panel (b) provides decomposition into structured matrices for the network in (a). Blue elements are diagonal ones, white elements are zero and red elements are non-zero in the model parameter matrix Θ. The structured penalty function (3) is then applied to each block for matrices $\{Z_i\}_{i=1}^n$.

graphical lasso problem (Friedman et al., 2008; Rothman et al., 2008) in the form of (2) with loss function

$$\mathcal{G}_1(X, \Theta_1) := \text{trace}(\hat{\Sigma}\Theta_1) - \log\det\Theta_1, \qquad \Theta_1 \in \mathcal{S}, \tag{5}$$

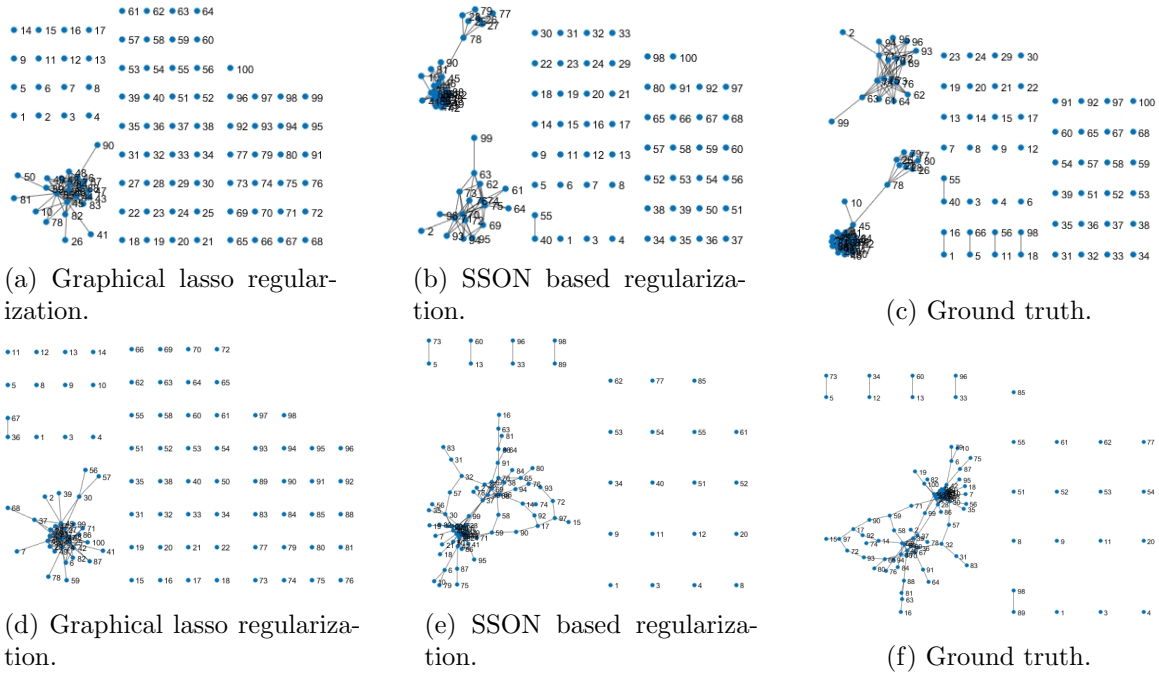
where $\hat{\Sigma}$ is the empirical covariance matrix of X; Θ_1 is the estimate of the precision matrix Σ^{-1}; and $\mathcal{S}$ is the set of $p \times p$ symmetric positive definite matrices.

As is well known, the norm penalty in (2) encourages zeros (sparsity) in the solution. However, as previously argued, many biological and social network applications exhibit more complex structures than mere sparsity. Using the proposed SSON, we define the following objective function for the problem at hand:

$$\begin{aligned} \underset{\Theta_1, Z_1, \ldots, Z_n \in \mathcal{S},\ E}{\text{minimize}} \quad & \mathcal{G}_1(X, \Theta_1) + \Omega(\Theta_1, Z_1, \ldots, Z_n, E), \\ \Theta_1 \quad = \quad & \sum_{i=1}^{n} \left(Z_i + Z_i^\top\right) + E, \end{aligned} \tag{6}$$

where Θ_1 is the model parameter matrix and $\Omega(\Theta_1, Z_1, \ldots, Z_n, E)$ the corresponding SSON defined in (3).

Formulation (6) allows us to obtain more accurate and compact network estimates than conventional methods whenever the network exhibits different structures. Moreover, our formulation does not require *a priori* knowledge of the underlying network structure (i.e. which nodes in the network form densely connected subgraphs (see, Figure 4)).



(a) Graphical lasso regularization.

(b) SSON based regularization.

(c) Ground truth.

(d) Graphical lasso regularization.

(e) SSON based regularization.

(f) Ground truth.

Figure 4: Estimates from the SSON based regularization on two examples of Gaussian graphical models comprising of $p = 100$ nodes, using in (4b) three structured matrices and in (4e) four structured matrices.

In Figure 4, the performance of our proposed approach is illustrated on two simulated data sets exhibiting different structures (sub-figures (4c) and (4f)); it can be seen that the proposed SSON based graphical lasso (sub-figures (4b) and (4e)) can recover the network structure much better than the popular graphical lasso based estimator (Friedman et al., 2008) (subfigures (4a) and (4d)).

2.3 Structured Ising Model

Another popular graphical model, suitable for binary or categorical data, is the Ising one (Ising, 1925). It is assumed that observations $x_1, \ldots, x_m$ are independent and identically distributed from

$$f(x, \Theta_2) = \frac{1}{\mathbb{W}(\Theta_2)} \exp\Big(\sum_{j=1}^{p} \theta_{jj} x_j + \sum_{1 \le j < j' \le p} \theta_{jj'} x_j x_{j'} \Big), \tag{7}$$

where $\mathbb{W}(\Theta_2)$ is the partition function, which ensures that the density sums to one. Here, Θ_2 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 given the remaining ones.

Several papers proposing estimation procedures for this model have been published. Lee et al. (2007) considered maximizing an ℓ_1-penalized log-likelihood for this model. Due to the difficulty in computing the log-likelihood with the expensive partition function, several authors have considered alternative approaches. For instance, Ravikumar et al. (2011)

proposed a neighborhood selection approach. The latter proposal involves solving p logistic regressions separately (one for each node in the network), which leads to an estimated parameter matrix that is in general not symmetric. In contrast, several authors considered maximizing an ℓ_1-penalized pseudo-likelihood with a symmetric constraint on Θ_2 (Guo et al., 2011a,b). Under the model (7), the log-pseudo-likelihood for m observations takes the form

$$\mathcal{G}_2(X, \Theta_2) := \sum_{j=1}^{p}\sum_{j'=1}^{p} \theta_{jj'}(X^T X)_{jj'} - \sum_{i=1}^{m}\sum_{j=1}^{p} \log\Big(1 + \exp\big(\theta_{jj} + \sum_{j'\neq j} \theta_{jj'} x_{ij'}\big)\Big), \tag{8}$$

We propose instead to impose the SSON on Θ_2 in (8) in order to estimate a binary network with different structures. This leads to the following optimization problem

$$\begin{aligned} \underset{\Theta_2, Z_1,\ldots,Z_n\in\mathcal{S},\ E}{\text{minimize}} \quad & \mathcal{G}_2(X, \Theta_2) + \Omega(\Theta_2, Z_1, \ldots, Z_n, E), \\ \Theta_2 \quad &= \quad \sum_{i=1}^{n}\big(Z_i + Z_i^\top\big) + E, \end{aligned} \tag{9}$$

where Θ_2 is the model parameter matrix and $\Omega(\Theta_2, Z_1, \ldots, Z_n, E)$ the corresponding SSON defined in (3).

An interesting connection can be drawn between our technique and the Ising block model discussed in Berthet et al. (2016), which is a perturbation of the mean field approximation of the Ising model known as the Curie-Weiss model: the sites are partitioned into two blocks of equal size and the interaction between those within the same block is stronger than across blocks, to account for more order within each block. However, one can easily seen that the Ising block model is a special case of (9).

2.4 Structured Gaussian Covariance Graphical Models

Next, we consider estimation of a covariance matrix under the assumption that

$$x_1, \ldots, x_m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma).$$

This is of interest because the sparsity pattern of Σ specifies the structure of the marginal independence graph (Drton and Richardson, 2002, 2008).

Let Θ_3 be a $p \times p$ symmetric matrix containing the parameters of interest. Setting the loss function $\mathcal{G}_3(X, \Theta_3) := \frac{1}{2}\|\Theta_3 - \hat{\Sigma}\|_F^2$, Xue et al. (2012) proposed to estimate the positive definite covariance matrix, Σ by solving

$$\underset{\Theta_3\in\mathcal{S}}{\text{minimize}} \quad \mathcal{G}_3(X, \Theta_3) + \lambda\|\Theta_3\|_1, \tag{10}$$

where $\hat{\Sigma}$ is the empirical covariance matrix, $\mathcal{S} = \{\Theta_3 : \Theta_3 \succeq \varepsilon I \text{ and } \Theta_3 = \Theta_3^T\}$, and ε is a small positive constant. We extend (10) to accommodate structures of the covariance graph by imposing the SSON on Θ_3. This results in the following optimization problem

$$\begin{aligned} \underset{\Theta_3, Z_1,\ldots,Z_n\in\mathcal{S},\ E}{\text{minimize}} \quad & \mathcal{G}_3(X, \Theta_3) + \Omega(\Theta_3, Z_1, \ldots, Z_n, E), \\ \Theta_3 \quad &= \quad \sum_{i=1}^{n}\big(Z_i + Z_i^\top\big) + E. \end{aligned} \tag{11}$$

where Θ_3 is the model parameter matrix and $\Omega(\Theta_3, Z_1, \ldots, Z_n, E)$ the corresponding SSON defined in (3).

2.5 Structured Gaussian Graphical Models with latent variables

In many applications throughout science and engineering, it is often the case that some relevant variables are not observed. For the Gaussian Graphical model, Chandrasekaran et al. (2010) proposed a convex optimization problem to estimate it in the presence of latent variables. Let Θ_4 be a $p \times p$ symmetric matrix containing the parameters of interest. Setting $\mathcal{G}_4(X, \Theta_4) := \langle \Theta_4, \Sigma_O \rangle - \log\det \Theta_4$, their objective function is given by

$$\begin{aligned} \underset{\Theta_4, Z_1, Z_2 \in \mathcal{S}}{\text{minimize}} \quad & \mathcal{G}_4(X, \Theta_4) + \alpha \|Z_1\|_1 + \beta \operatorname{trace}(Z_{n+1}) + \mathbb{1}_{Z_{n+1} \succeq 0}, \\ \Theta_4 \quad &= \quad Z_1 - Z_{n+1}, \end{aligned} \tag{12}$$

where Σ_O is the sample covariance matrix of the observed variables; α and β are positive constants; and the indicator function $\mathbb{1}_{Z_{n+1} \succeq 0}$ is defined as

$$\mathbb{1}_{Z_{n+1} \succeq 0} := \begin{cases} 0, & \text{if } Z_{n+1} \succeq 0, \\ +\infty, & \text{otherwise.} \end{cases}$$

This convex optimization problem aims to estimate an inverse covariance matrix that can be decomposed into a sparse matrix Z_1 minus a low-rank matrix Z_{n+1} based on high-dimensional data.

Next, we extend the SSON to solve the latent variable graphical model selection. Problem (12) can be rewritten in the following equivalent form by introducing new variables $\{Z_i\}_{i=1}^n$:

$$\begin{aligned} \underset{\Theta_4, Z_1, \ldots, Z_n \in \mathcal{S},\ E}{\text{minimize}} \quad & \mathcal{G}_4(X, \Theta_4) + \Omega(\Theta_4, Z_1, \ldots, Z_n, E) + \lambda_{n+1} \operatorname{trace}(Z_{n+1}) + \mathbb{1}_{Z_{n+1} \succeq 0}, \\ \Theta_4 \quad &= \quad \sum_{i=1}^{n} \big(Z_i + Z_i^\top\big) - Z_{n+1} + E, \end{aligned} \tag{13}$$

where Θ_4 is the model parameter matrix and $\Omega(\Theta_4, Z_1, \ldots, Z_n, E)$ the corresponding SSON defined in (3).

2.6 Structured Linear Regression and Vector Auto-Regression

The proposed SSON is also applicable to structured regression problems. Although this is not the main focus on this paper, nevertheless, we include a brief discussion, especially for lag selection in vector autoregressive models that are of prime interest in the analysis of high-dimensional time series data. The canonical formulation of the regularized regression problem is given by:

$$\min_{\theta \in \mathbb{R}^p} \|y - X\theta\|_2 + \lambda \Psi(\theta). \tag{14}$$

where $\{(y_i, x_i)\}_{i=1}^m$, $y_i \in \mathbb{R}$, $x_i \in \mathbb{R}^p$, with $y = [y_1, \ldots, y_m]^\top$ being the response variable and $X = [x_1^\top, \ldots x_m^\top]$ a set of p-predictors that are assumed to be independently and identically distributed (i.i.d.); $\lambda > 0$ is a regularization parameter and $\Psi(\theta)$ is a suitable norm. Specific

choices of $\Psi(.)$ lead to popular regularizers including the lasso -$\Psi(\theta) = \|\theta\|_1$- and the group lasso.

We propose instead to impose the SSON on θ in (14) in order to solve structured regression problems. Problem (14) can be rewritten in the following form by introducing new variables $\{z_i\}_{i=1}^n$ and e:

$$\begin{aligned} \underset{\theta, z_1, \ldots, z_n, e}{\text{minimize}} \quad & \mathcal{G}(X, \theta) + \omega(\theta, z_1, \ldots, z_n, e), \\ \theta_4 \quad = \quad & z_1 + z_2 + \cdots + z_n + e, \end{aligned} \tag{15}$$

where $\mathcal{G}(X, \theta) = \|y - X\theta\|_2$; θ is the model parameter vector and $\omega(\theta, z_1, \ldots, z_n, e)$ the corresponding structured norm defined in (4).

Problem (15) can equivalently be thought of as a generalization of subspace clustering (Elhamifar and Vidal, 2009). Indeed, in order to segment the data into their respective subspaces, we need to compute an affinity vector θ that encodes the pairwise affinities between data vectors.

An interesting application of the SSON for multivariate regression problems is on structured estimation of *vector autoregression* (VAR) models (Lütkepohl, 2005), a popular model for economic and financial time series data (Tsay, 2005), dynamical systems (Ljung, 1998) and more recently brain function connectivity (Valdés-Sosa et al., 2005). The model captures both temporal and cross-dependencies between stationary time series. Formally, let $\{x_1, \ldots, x_m\}$ be a p-dimensional time series set of observations that evolve over time according to a lag-d model:

$$x_{t+1} = \sum_{k=1}^{d} \Theta_k^\top x_{t-k} + \epsilon_t, \qquad \epsilon_1, \ldots, \epsilon_{m-1} \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \Sigma), \qquad t = 1, \ldots, m-1,$$

where $\{\Theta\}_{k=1}^d \in \mathbb{R}^{p \times p}$ are *transition matrices* for different lags, and $\{\epsilon_1, \ldots, \epsilon_{m-1}\}$ independent multivariate Gaussian *white noise* processes. The VAR process is assumed to be stable and stationary (bounded spectral density), while the noise covariance matrix Σ is assumed to be positive definite with bounded largest eigenvalue (Basu and Michailidis, 2015).

Given m observations $\{x_1, x_2, \ldots, x_m\}$ from a stationary VAR process, the lag-m VAR can be written is given by

$$\underbrace{\begin{bmatrix} x_m \\ x_{m-1} \\ \vdots \\ x_2 \end{bmatrix}}_{\mathbf{Y}} = \underbrace{\begin{bmatrix} {x_{m-1}}^\top \\ {x_{m-2}}^\top \\ \vdots \\ {x_1}^\top \end{bmatrix}}_{\mathbf{X}} \Theta + \underbrace{\begin{bmatrix} {\epsilon_{m-1}}^\top \\ {\epsilon_{m-2}}^\top \\ \vdots \\ {\epsilon_1}^\top \end{bmatrix}}_{\varepsilon}. \tag{16}$$

It can be seen that to estimate Θ one can solve the following least squares problem

$$\min_{\Theta \in \mathbb{R}^{p \times p}} \|Y - X\Theta\|_F. \tag{17}$$

However, as the number of component time series increases, the number of parameters to be estimated grows as dp^2 ; hence, structural assumptions are imposed to estimate them from

limited sample size. A popular choice is the lasso (Basu and Michailidis, 2015), that leads to sparse estimates. However, it does not incorporate the notion of lag selection, which could lead to certain spurious coefficients coming from further lags in the past. To address this problem, Basu et al. (2015) proposed a thresholded lasso estimate. However, our SSON can be used for lag selection, that guarantees that more recent lags are favored over further in the past ones.

Let Θ_5 be a $mp \times mp$ symmetric matrix containing the parameters of interest for all m lages of the problem. Setting the loss function $\mathcal{G}_5(X, \Theta_5) := \|Y - X\Theta_5\|$, we propose to estimate the transition matrix, Θ by solving the following optimization problem:

$$\begin{aligned} \min_{\Theta_5, Z_1, \ldots, Z_n, E \in \mathbb{R}^{p\times p}} \quad & \mathcal{G}_5(X, \Theta_5) + \Omega(\Theta_5, Z_1, \ldots, Z_n, E), \\ \Theta_5 \quad = \quad & \sum_{i=1}^{n} \big(Z_i + Z_i^\top\big) + E, \end{aligned} \tag{18}$$

where Θ_5 is the estimate of the covariance matrix and $\Omega(\Theta_5, Z_1, \ldots, Z_n, E)$ the corresponding SSON defined in (3).

3. Multi-Block ADMM for Estimating Structured Network Models

Objective functions (6), (9), (11), (13), (15), and (18) involve separable convex functions, while the constraint is simply linear, and therefore they are suitable for ADMM based algorithms. We next introduce a linearized multi-block ADMM algorithm to solve these problems and establish its global convergence properties.

The alternating direction method of multipliers (ADMM) is widely used in solving structured convex optimization problems due to its superior performance in practice; see Scheinberg et al. (2010); Boyd et al. (2011); Hong and Luo (2017); Lin et al. (2015, 2016); Sun et al. (2015); Davis and Yin (2015); Hajinezhad and Hong (2015); Hajinezhad et al. (2016). On the theoretical side, Chen et al. (2016) provided a counterexample showing that the ADMM may *fail to converge* when the number of blocks exceeds *two*. Hence, many authors reformulate the problem of estimating a Markov Random Field model to a two block ADMM algorithm by grouping the variables and introducing auxiliary variables (Ma et al., 2013; Mohan et al., 2012; Tan et al., 2014). However, in the context of large-scale optimization problems, the grouping ADMM method becomes expensive due to its high memory requirements. Moreover, despite lack of convergence guarantees under standard convexity assumptions, it has been observed by many researchers that the unmodified multi-block ADMMs with Gauss-Seidel updates often outperform all its modified versions in practice (Wang et al., 2013; Sun et al., 2015; Davis and Yin, 2015).

Next, we present a *convergent multi-block* ADMM with Gauss-Seidel updates to solve convex problems (6), (9), (11), (13), and (18). The ADMM is constructed for an augmented Lagrangian function defined by

$$\begin{aligned} \mathcal{L}_\gamma(\Theta, Z_1, \ldots, Z_n, E; \Lambda) \quad = \quad & \mathcal{G}(X, \Theta) + f_1(Z_1) + \cdots + f_n(Z_n) + \frac{\lambda_e}{2}\|E\|_F^2 \\ - \quad & \langle \Lambda, \Theta - \sum_{i=1}^{n} Z_i + Z_i^\top - E\rangle + \frac{\gamma}{2}\|\Theta - \sum_{i=1}^{n} Z_i + Z_i^\top - E\|_F^2, \end{aligned} \tag{19}$$

where Λ is the Lagrange multiplier, γ a penalty parameter, $\mathcal{G}(X,\Theta)$ the loss function of interest and

$$\begin{aligned} f_1(Z_1) &:= \lambda_1\|Z_1 - \mathrm{diag}(Z_1)\|_1, \\ f_i(Z_i) &:= \hat{\lambda}_i\|Z_i - \mathrm{diag}(Z_i)\|_1 + \lambda_i \sum_{j=1}^{l_i} \|(Z_i - \mathrm{diag}(Z_i))_j\|_F, \quad i = 2,\ldots,n. \end{aligned} \tag{20}$$

In a typical iteration of the ADMM for solving (19), the following updates are implemented:

$$\Theta^{k+1} = \underset{\Theta}{\mathrm{argmin}} \quad \mathcal{G}(X,\Theta) + \frac{\gamma}{2}\|\Theta - B_0\|_F^2, \tag{21}$$

$$Z_i^{k+1} = \underset{Z_i}{\mathrm{argmin}} \quad f_i(Z_i) + \frac{\gamma}{2}\|Z_i + Z_i^\top - B_i\|_F^2, \qquad i = 1,\ldots n, \tag{22}$$

$$E^{k+1} = \underset{E}{\mathrm{argmin}} \quad f_e(E) + \frac{\gamma}{2}\|E - B_{n+1}\|_F^2, \tag{23}$$

$$\Lambda^{k+1} = \Lambda^k - \gamma(\Theta^{k+1} - \sum_{i=1}^{n} Z_i^{k+1} + {Z_i^{k+1}}^\top - E^{k+1}). \tag{24}$$

where

$$\begin{aligned} B_0 &= \sum_{i=1}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k, \\ B_1 &= \Theta^{k+1} - (\sum_{i=2}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k), \\ B_i &= \Theta^{k+1} - (\sum_{j=1}^{i-1} Z_j^{k+1} + {Z_j^{k+1}}^\top \\ &\quad + \sum_{j=i+1}^{n} Z_j^k + {Z_j^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k), \qquad i = 2,\ldots n-1, \\ B_n &= \Theta^{k+1} - (\sum_{i=1}^{n-1} Z_i^{k+1} + {Z_i^{k+1}}^\top + E^k + \frac{1}{\gamma}\Lambda^k), \\ B_{n+1} &= \Theta^{k+1} - (\sum_{i=1}^{n} Z_i^{k+1} + {Z_i^{k+1}}^\top + \frac{1}{\gamma}\Lambda^k). \end{aligned} \tag{25}$$

To avoid introducing auxiliary variables and still solve subproblems (22) efficiently, we propose to approximate the subproblems (22) by linearizing the quadratic term of its objective function (see also Bolte et al., 2014; Lin et al., 2011; Yang and Yuan, 2013). With this linearization, the resulting approximation to (22) is then simple enough to have a closed-form solution. More specifically, letting $H_i(Z_i) = \frac{\gamma}{2}\|Z_i + Z_i^\top - B_i\|_F^2$, we define the following majorant function of $H_i(Z_i)$ at point Z_i^k,

$$H_i(Z_i) \le \gamma\Big(\frac{1}{2}\|Z_i^k + {Z_i^k}^\top - B_i\|_F^2 + \langle \nabla H_i(Z_i^k), Z_i - Z_i^k\rangle + \frac{\varrho}{2}\|Z_i - Z_i^k\|_F^2\Big), \tag{26}$$

where ϱ is a proximal parameter, and

$$\nabla H_i(Z_i^k) := 2(Z_i^k + {Z_i^k}^\top) - (B_i + B_i^\top), \tag{27}$$

Plugging (26) into (22), with simple algebraic manipulations, we obtain:

$$Z_i^{k+1} \quad = \quad \underset{Z_i}{\operatorname{argmin}} \quad f_i(Z_i) + \frac{\varrho\gamma}{2}\|Z_i - C_i\|_F^2, \qquad i = 1, \dots n, \tag{28}$$

where $C_i = Z_i^k - \frac{1}{\varrho}\nabla H_i(Z_i^k)$.

The next result establishes the sufficient decrease property of the objective function given in (22), after a proximal map step computed in (28).

Lemma 7 *(Sufficient decrease property). Let $\varrho > \frac{L_{H_i}}{\gamma}$, where L_{H_i} is a Lipschitz constant of the gradient $\nabla H_i(Z_i)$ and γ is a penalty parameter defined in* (19)*. Then, we have*

$$f_i(Z_i^{k+1}) + H_i(Z_i^{k+1}) \quad \leq \quad f_i(Z_i^k) + H_i(Z_i^k) - \frac{(\varrho\gamma - L_{H_i})}{2}\|Z_i^{k+1} - Z_i^k\|_F^2, \qquad i = 1, \dots, n,$$

where $Z_i^{k+1} \in \mathbb{R}^{n\times n}$ defined by (28)*.*

Proof. The proof of this Lemma follows along similar lines to the proof of Lemma 3.2 in Bolte et al. (2014).

It is well known that (28) has a closed-form solution that is given by the shrinkage operation (Boyd et al., 2011):

$$\begin{aligned} Z_1^{k+1} \quad &= \quad \text{Shrink}\Big(C_1, \frac{\lambda_1}{\varrho\gamma}\Big), \\ Z_{i_j}^{k+1} \quad &= \quad \max\big(1 - \frac{\lambda_i}{\varrho\gamma\|\text{Shrink}(C_{i_j}, \frac{\hat{\lambda}_i}{\varrho\gamma})\|_F}, 0\big) \cdot \text{Shrink}(C_{i_j}, \frac{\hat{\lambda}_i}{\varrho\gamma}), \quad \substack{i=2,\dots n,\\ j=1,\dots,l_i,} \end{aligned} \tag{29}$$

where $\text{Shrink}(\cdot,\cdot)$ in (29) denotes the soft-thresholds operator, applied element-wise to a matrix A (Boyd et al., 2011):

$$\text{Shrink}(A_{ij}, b) := \text{sign}(A_{ij})\max\big(|A_{ij}| - b, 0\big) \qquad \substack{i=1,\dots p,\\ j=1,\dots,p,}.$$

Remark 8 *Note that in the case of solving problem* (13)*, one needs to add another block function $f_{n+1}(Z_{n+1}) := \lambda_{n+1}\,\text{trace}(Z_{n+1}) + \mathbb{1}_{Z_{n+1}\succeq 0}$ to the augmented Lagrangian function* (19) *and update $\{C_i\}_{i=1}^n$. In this case, the proximal mapping of f_{n+1} is*

$$prox(f_{n+1}, \gamma, Z_{n+1}) := \underset{Z_{n+1}}{\operatorname{argmin}}\, f_{n+1}(Z_{n+1}) + \frac{\gamma}{2}\|Z_{n+1} - C_{n+1}\|_F^2, \tag{30}$$

where $C_{n+1} = \Theta^{k+1} - (\sum_{i=1}^n Z_i^{k+1} + {Z_i^{k+1}}^\top + E^k + \frac{1}{\gamma}\Lambda^k)$. It is easy to verify that (30) *has a closed-form solution given by*

$$Z_{n+1} = U\ \max(D - \frac{\lambda_{n+1}}{\gamma}, 0)\ U^T,$$

where UDU^T is the eigenvalue decomposition of C_{n+1} (see, Chandrasekaran et al., 2010; Ma et al., 2013 for more details).

The discussions above suggest that the following unmodified ADMM for solving (19) gives rise to an efficient algorithm.

Algorithm 1 Multi-Block ADMM Algorithm for Solving (19).

1: **Initialize** The parameters:

(a) Primal variables Θ, $Z_1, \ldots, Z_n$, E, to the $p \times p$ identity matrix.

(b) Dual variable Λ to the $p \times p$ zero matrix.

(c) Constants $\varrho, \lambda_e, \tau > 0$, and $\gamma \geq \sqrt{2}\lambda_e$.

(d) Nonnegative regularization constants $\lambda_1, \ldots, \lambda_n$, $\hat{\lambda}_2, \ldots, \hat{\lambda}_n$.

2: **Iterate** Until the stopping criterion $\|\Theta^k - \Theta^{k-1}\|_F^2 / \|\Theta^{k-1}\|_F \leq \tau$ is met:

(a) Update Θ:

$$\Theta^{k+1} = \underset{\Theta \in \mathcal{S}}{\operatorname{argmin}} \ \mathcal{G}(X, \Theta) + \frac{\gamma}{2}\|\Theta - B_0\|_F^2,$$

where B_0 is defined in (25).

(b) Update Z_i:

i. $Z_1^{k+1} = \text{Shrink}\Big(C_1, \frac{\lambda_1}{\varrho\gamma}\Big)$,

ii. $Z_{i_j}^{k+1} = \max\big(1 - \dfrac{\lambda_i}{\varrho\gamma\|\text{Shrink}(C_{i_j}, \frac{\hat{\lambda}_i}{\varrho\gamma})\|_F}, 0\big) \cdot \text{Shrink}(C_{i_j}, \frac{\hat{\lambda}_i}{\varrho\gamma}), \quad \begin{smallmatrix} i=2,\ldots n, \\ j=1,\ldots,l_i, \end{smallmatrix}$

where C_i is defined in (28).

(c) Update E:

$$E^{k+1} = \underset{E}{\operatorname{argmin}} \ \tfrac{\lambda_e}{2}\|E\|_F^2 + \tfrac{\gamma}{2}\|E - B_{n+1}\|_F^2$$

where B_{n+1} is defined in (25).

(d) Update Λ:

$$\Lambda^{k+1} = \Lambda^k - \gamma(\Theta^{k+1} - \textstyle\sum_{i=1}^n Z_i^{k+1} + {Z_i^{k+1}}^\top - E^{k+1})$$

Remark 9 *The complexity of Algorithm 1 is of the same order as the graphical lasso (Friedman et al., 2008), the method in Tan et al. (2014) for hub node discovery and the algorithm used for estimation of sparse covariance matrices introduced by Xue et al. (2012). Indeed, one can easily see that with any set of structured matrices $\{Z_i\}_{i=1}^n$, the complexity of Algorithm 1 is equal to $O(p^3)$, which is the complexity of the eigen-decomposition for updating Θ in step 2(a).*

Since both the objective function and constraints of (19) become separable after using the linearization technique introduced in (26), the problem can be decomposed into $n + 2$ smaller subproblems; the latter can be solved in a parallel and distributed manner with a small modification in Algorithm 1. Indeed, we can apply a Jacobian ADMM to solve (19)

with the following updates,

$$\begin{aligned}
\Theta^{k+1} &= \underset{\Theta}{\operatorname{argmin}} \quad \mathcal{G}(X,\Theta) + \frac{\gamma}{2}\|\Theta - B_0\|_F^2, \\
Z_i^{k+1} &= \underset{Z_i}{\operatorname{argmin}} \quad f_i(Z_i) + \frac{\varrho\gamma}{2}\|Z_i - C_i\|_F^2, \qquad i = 1,\dots n, \\
E^{k+1} &= \underset{E}{\operatorname{argmin}} \quad f_e(E) + \frac{\gamma}{2}\|E - B_{n+1}\|_F^2, \\
\Lambda^{k+1} &= \Lambda^k - \gamma(\Theta^{k+1} - \sum_{i=1}^{n} Z_i^{k+1} + {Z_i^{k+1}}^\top - E^{k+1}).
\end{aligned} \tag{31}$$

where C_i is defined in (28) with

$$\begin{aligned}
B_0 &= \sum_{i=1}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k, \\
B_i &= \Theta^k - (\sum_{j=1}^{i-1} Z_j^k + {Z_j^k}^\top \\
&+ \sum_{j=i+1}^{n} Z_j^k + {Z_j^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k), \qquad i = 2,\dots n-1, \\
B_n &= \Theta^k - (\sum_{i=1}^{n-1} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k), \\
B_{n+1} &= \Theta^k - (\sum_{i=1}^{n} Z_i^k + {Z_i^k}^\top + \frac{1}{\gamma}\Lambda^k).
\end{aligned} \tag{32}$$

Intuitively, the performance of the Jacobian ADMM should be worse than the Gauss-Seidel version, because the latter always uses the latest information of the primal variables in the updates. We refer to Liu et al. (2015); Lin et al. (2015) for a detailed discussion on the convergence analysis of the Jacobian ADMM and its variants. On the positive side, we obtain a parallelizable version of the multi-block ADMM algorithm.

3.1 Convergence analysis

The next result establishes the global convergence of the standard multi-block ADMM for solving SSON based statistical learning problems, by using the Kurdyka- Lojasiewicz (KL) property of the objective function in (19).

Theorem 10 *The sequence $U^k := (\Theta^k, Z_1^k, \dots, Z_n^k, E^k, \Lambda^k)$ generated by Algorithm 1 from any starting point converges to a stationary point of the problem given in* (19).

Proof. A detailed exposition is given in Appendix B.

4. Experimental Results

In this section, we present numerical results for Algorithm 1 (henceforth called SSONA), on both synthetic and real data sets. The results are organized in the following three sub-sections: in Section 4.1, we present numerical results on synthetic data comparing the performance of SSONA to that of grouping variables ADMM and also for assessing the accuracy in recovering a multi-layered structure in Markov Random Field and covariance graph models that constitute the prime focus in this paper. In Section 4.2 we use the proposed SSONA for feature selection in classification problems involving two real data sets in order to calibrate SSON performance with respect to an independent validation set. Finally, in Section 4.3, we analyze using SSONA on some other interesting real data sets from the social and biological sciences.

4.1 Experimental results for the SSON algorithm on graphical models based on synthetic data

Next, we evaluate the performance of SSONA on ten synthetic graphical model problems, comprising of $p = 100$, 500 and 1000 variables. The underlying network structure corresponds to an Erdős-Rényi model graph, a nearest neighbor graph and a scale-free random graph, respectively. The CONTEST [1] package is used to generate the synthetic graphs, and the UGM [2] package to implement Gibbs sampling for estimating the Ising Model. Based on the generated graph topologies, we consider the following settings for generating synthetic data sets:

I. **Gaussian graphical models**:
For a given number of variables p, we first create a symmetric matrix $E \in \mathbb{R}^{p\times p}$ by using CONTEST in a MATLAB environment. Given matrix E, we set Σ^{-1} equal to $E + (0.1 - \bar{\Lambda}_{\min}(E))\ I$, where $\bar{\Lambda}_{\min}(E)$ is the smallest eigenvalue of E and I denotes the identity matrix. We then draw $N = 5p$ i.i.d. vectors $x_1, \ldots, x_m$ from the gasserian distribution $\mathcal{N}(0, \Sigma)$ by using the *mvnrnd* function in MATLAB, and then compute a sample covariance matrix of the variables.

II. **Gaussian graphical models with latent variables:**
For a given number of variables p, we first create a matrix $\Sigma^{-1} \in \mathbb{R}^{(p+r)\times(p+r)}$ by using CONTEST as described in I. We then choose the sub-matrix $\Theta_O = \Sigma^{-1}(1:p, 1:p)$ as the ground truth matrix of the matrix Θ_4 and chose

$$\begin{aligned}\Theta_U &= \Sigma^{-1}(1:p,\ p+1:p+r)\big(\Sigma^{-1}(p+1:p+r,\ p+r:p+r)\big)^{-1} \\ &\quad\ \Sigma^{-1}(p+1:p+r,\ 1:p)\end{aligned}$$

as the ground truth matrix of the low rank matrix U. We then draw $N = 5p$ i.i.d. vectors $x_1, \ldots, x_m$ from the Gaussian distribution $\mathcal{N}(0, (\Theta_O - \Theta_U)^{-1})$, and compute the sample covariance matrix of the variables Σ_O.

III. **The Binary Network**:
To generate the parameter matrix Σ, we create an adjacency matrix as in Setup I by using

1. CONTEST is available at http://www.mathstat.strath.ac.uk/outreach/contest/
2. UGM is available at http://www.di.ens.fr/ mschmidt/Software/UGM.html

CONTEST. Then, each of $N = 5p$ observations is generated through Gibbs sampling. We take the first 100000 iterations as our burn-in period, and then collect observations, so that they are nearly independent.

We compare SSONA to the following competing methods:

- **CovSel**, designed to estimate a sparse Gaussian graphical model (Friedman et al., 2008);
- **HGL**, focusing on learning a Gaussian graphical model having hub nodes (Tan et al., 2014);
- **PGADM**, designed to learn a Gaussian graphical model with some latent nodes (Ma et al., 2013);
- **Pseudo-Exact**, designed to learn a binary Ising graphical model (Höfling and Tibshirani, 2009);
- **glasso-SF**, Learning Scale Free Networks by reweighted ℓ_1 Regularization (Liu and Ihler, 2011);
- **GADMM**, A two block ADMM method with grouping variables.

All the algorithms have been implemented in the MATLAB R2015b environment on a PC with a 1.8 GHz processor and 6GB RAM memory. Further, all the algorithms are being terminated either when

$$\frac{\|\Theta^k - \Theta^{k-1}\|_F^2}{\|\Theta^{k-1}\|_F^2} \leq \tau, \qquad \tau = 1e-5,$$

or the number of iterations and CPU times exceed 1,000 and 10 minutes, respectively.

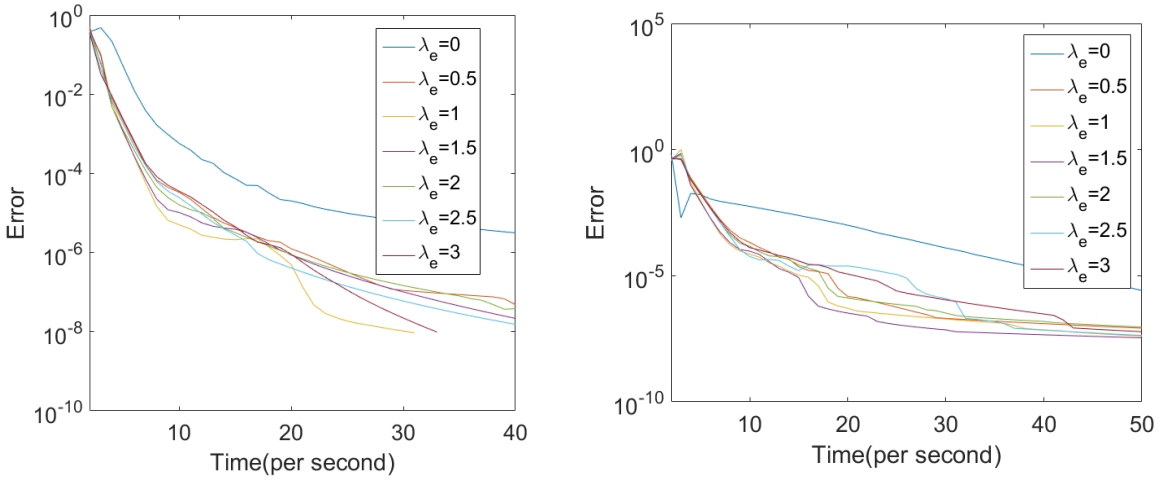
We found that in practice the computation cost for SSONA increases with the size of structured matrices. Therefore, we use a limited memory version of SSONA in our experimental results to obtain good accuracy. Block sizes in Figure 3 could be set based on a desire for interpretability of the resulting estimates. In this section, we choose four structured matrices with blocks of size

$$\begin{aligned}
(Z_2)_j &= [1, \frac{p}{2}], \qquad j = 1 \ldots, l_2, \\
(Z_3)_j &= [1, \frac{p}{5}], \qquad j = 1 \ldots, l_3, \\
(Z_4)_j &= [1, \frac{p}{10}], \qquad j = 1 \ldots, l_4, \\
(Z_5)_j &= [1, \frac{p}{20}], \qquad j = 1 \ldots, l_5,
\end{aligned}$$

where l_i is determined based on size of the adjacency matrix, p (see, Figure 3).

The penalty parameters λ_e and $\{\lambda_i\}_{i=1}^n$ play an important rule for the convex decomposition to be successful. We learn them through numerical experimentation (see Figures 5 and 6) and set them respectively to

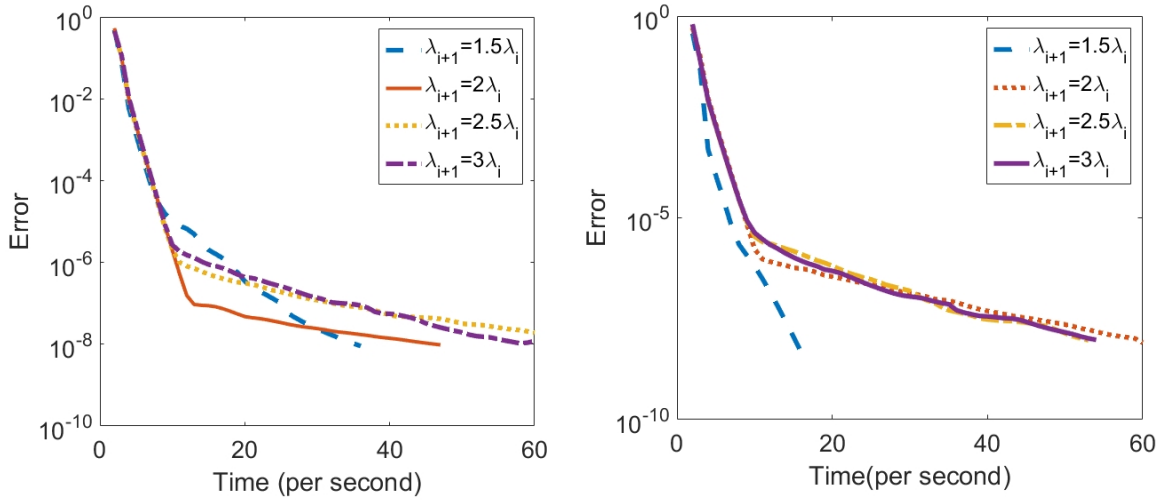
$$\varrho = 4, \quad \lambda_e = 1, \quad \lambda_1, \lambda_2 = 0.5\lambda_e, \quad \hat{\lambda}_i = 0.25\lambda_e, \quad \text{and} \quad \lambda_{i+1} = 2\lambda_i \quad \text{for} \quad i = 2, \ldots, n.$$



(a) (b)

Figure 5: Learning turning parameter λ_e for two covariance estimation problems. Comparison of the absolute errors produced by the algorithms based on CPU time for different choices of λ_e.

It can be seen from Figure 5 that with the addition of the ridge penalty term $\frac{\lambda_e}{2}\|E\|_F^2$ the algorithm clearly outperforms its unmodified counterpart in terms of CPU time for any fixed number of iterations. Indeed, when the model becomes more dense, SSONA is more effective to recover the network structure.



(a) (b)

Figure 6: Learning turning parameter λ_i for $i = 2, \ldots, n$ for two covariance estimation problems for different choices of λ_i for $i = 2, \ldots, n$.

Next, we conduct experiments to assess the performance of the developed multi-block ADMM algorithm (SSONA) vis-a-vis the GADMM for solving two covariance graph estimation problems of dimension 1000 in the presence of noise. Figure 7 depicts the absolute error of the objective function for different choices of the regularization parameter γ of the augmented Lagrangian and that of the dense noisy component λ_e; note that the latter is key for the convergence of the proposed algorithm.

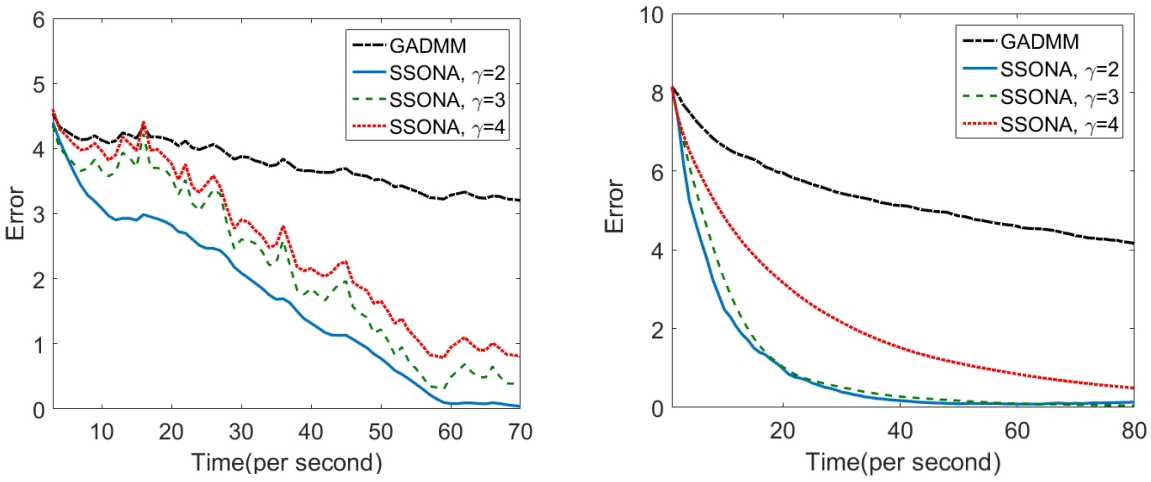


Figure 7: Comparison of the absolute errors produced by the algorithms based on CPU time for different choices of γ.

We define the following two performance measures, as proposed in Tan et al. (2014):

- Number of correctly estimated edges, n_e :

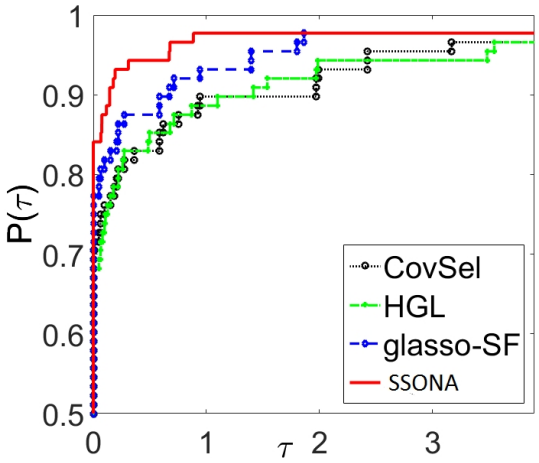
$$\sum_{j<j'} \left(1_{\{|\hat{\Theta}|>1e-4 \text{ and } |\Theta_{jj'}|\neq 0\}}\right).$$

- Sum of squared errors, s_e:

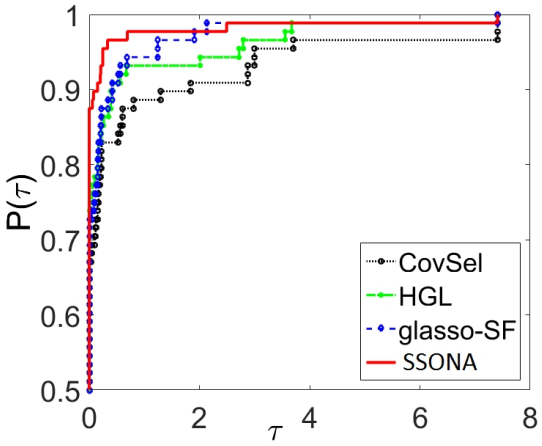
$$\sum_{j<j'} \left(|\hat{\Theta}_{jj'} - \Theta_{jj'}|\right)^2.$$

The experiment is repeated ten times and the average number of correctly estimated edges, n_e and sum of squared errors, s_e are considered for comparison. We have used the *performance profile*, as proposed in Dolan and Moré (2002), to display the efficiency of the algorithms considered, in terms of n_e and s_e. As stated in Dolan and Moré (2002), this profile provides a wealth of information such as solver efficiency, robustness and probability of success in compact form and eliminates the influence of a small number of problems on the evaluating process and the sensitivity of results associated with the ranking of solvers. Indeed, the performance profile plots the fraction of problem instances for which any given method is within a factor of the best solver. The horizontal axis of the figure gives the percentage of the test problems for which a method is efficient, while the vertical axis gives

the percentage of the test problems that were successfully solved by each method (robustness). The performance profiles of the considered algorithms in log2 scale are depicted in Figures 8,9 and 10.

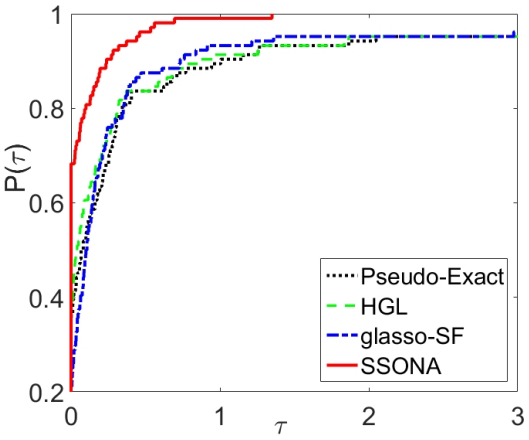


(a) Performance based on n_e.

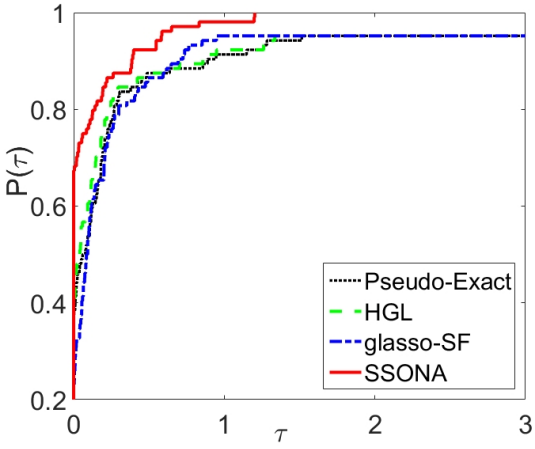


(b) Performance based on s_e.

Figure 8: Performance profiles of CovSel, HGL, glasso-SF and SSONA



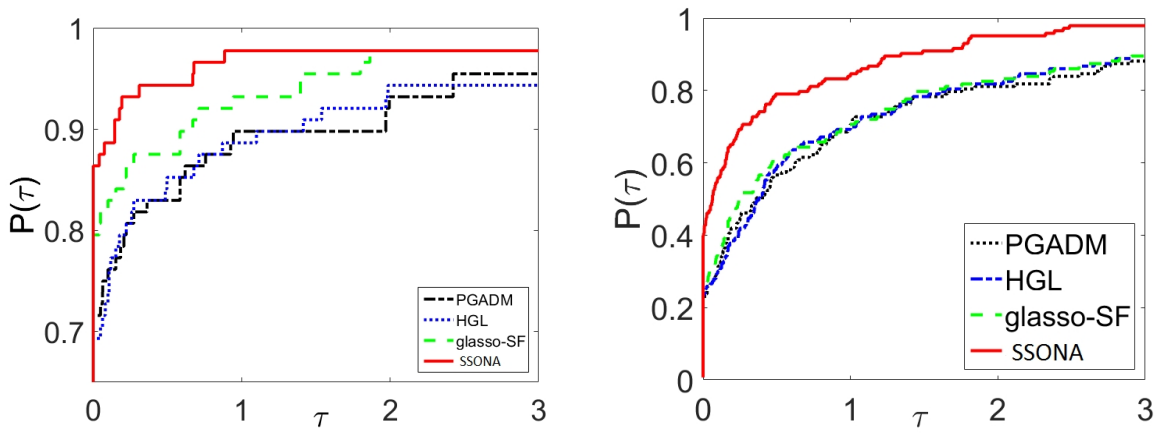
(a) Performance based on n_e.



(b) Performance based on s_e.

Figure 9: Performance profiles of Pseudo-Exact, HGL, glasso-SF and SSONA.

Figures 8,9 and 10 show the performance profiles of the considered algorithms for estimation of graphical models in terms of number of correctly estimated edges and sum of squared errors, respectively. The left and right panel are drawn in terms of n_e and s_e, respectively. The results in these figures clearly demonstrate the superior performance of the proposed method, since it solves all test problems without exhibiting any failure. Moreover, the SSONA algorithm is the best algorithm among the considered ones, as it solves more than 80 % of the test problems achieving the maximum number of correctly estimated edge



(a) Performance based on n_e.

(b) Performance based on s_e.

Figure 10: Performance profiles of PGADM, HGL, glasso-SF and SSONA.

n_e and minimum value of estimation loss s_e. Further, the performance index of SSONA grows up rapidly in comparison with the other considered algorithms. The latter implies that whenever SSONA is not the best algorithm, its performance index is close to the index of the best one.

4.1.1 Experiments on structured graphical models

In this section, we present numerical results on structured graphical models to demonstrate the efficiency of SSONA. We compare the behavior of SSONA for a fixed value of $p = 100$ with a lasso version of our algorithm. Results provided in Figures 11, 12, 13 and 14 indicate the efficiency of algorithm 1 on structured graphical models. These results also show how the structure of the network returned by the two algorithms changes with growing m (note that λ_i and $\hat{\lambda}_i$ are kept fixed for each value of m). It can be easily seen from these figures (comparing Row I and II) that SSONA is less sensitive to the number of samples and shows a better approximation of the network structure even for small sample sizes.

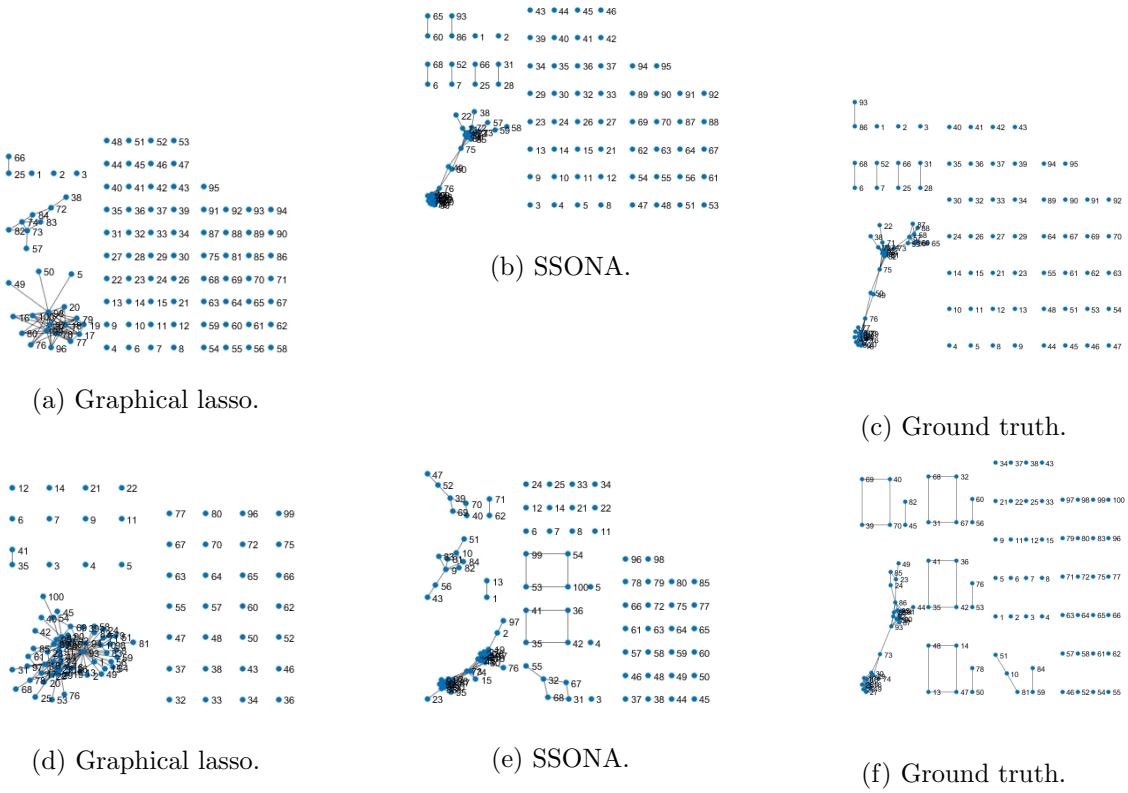
4.2 Classification and clustering accuracy based on SSONA

In this section, we evaluate the efficiency of SSONA on real data sets in recovering complex structured sparsity patterns and subsequently evaluate them on a classification task. The two data sets deal with applications in cancer genomic and document classification.

4.2.1 SSONA for Gene Selection Task

Classification with a sparsity constraint has become a standard tool in applications involving Omics data, due to the large number of available features and the small number of samples. The data set under study considers gene expression profiles of lung cancer tumors. Specifically, the data[2] consist of gene expression profiles of 12,626 genes for 197 lung tissue

2. `http://www.broadinstitute.org/cgibin/cancer/publications/view/87`.



(a) Graphical lasso.

(b) SSONA.

(c) Ground truth.

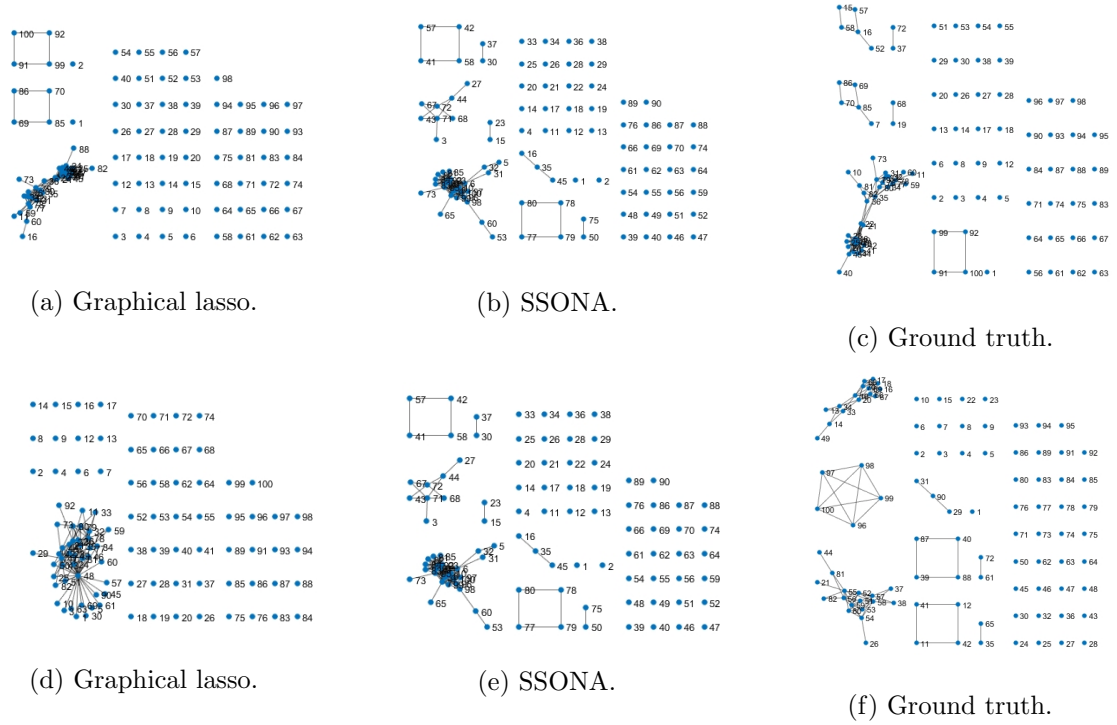
(d) Graphical lasso.

(e) SSONA.

(f) Ground truth.

Figure 11: Simulation for the Gaussian graphical model. Row I: Results for $p = 100$ and $m = 200$. Row II: Results for $p = 100$ and $m = 100$.

samples, with 139 adenocarcinomas(AD), 21 squamous cell carcinomas(SQ), 20 carcinoids (COID) and 17 normal lung tissue (NL). To distinguish lung adenocarcinomas from the normal lung tissues, we consider the diagnosis of lung cancer as a binary classification problem. Let the 17 normal lung comprise the positive class and the 139 lung adenocarcinomas the negative class. Following the workflow in Monti et al. (2003), we reserve the 1000 most significant genes after a preprocessing step. In the numerical experiment, we compare group lasso (Yuan and Lin, 2007), group lasso with overlap (Obozinski et al., 2011) and SSONA according to the following two criteria: average classification accuracy and gene selection performance. The experiment is repeated ten times and the average accuracy and performance are depicted in Table 1.



(a) Graphical lasso. (b) SSONA. (c) Ground truth.

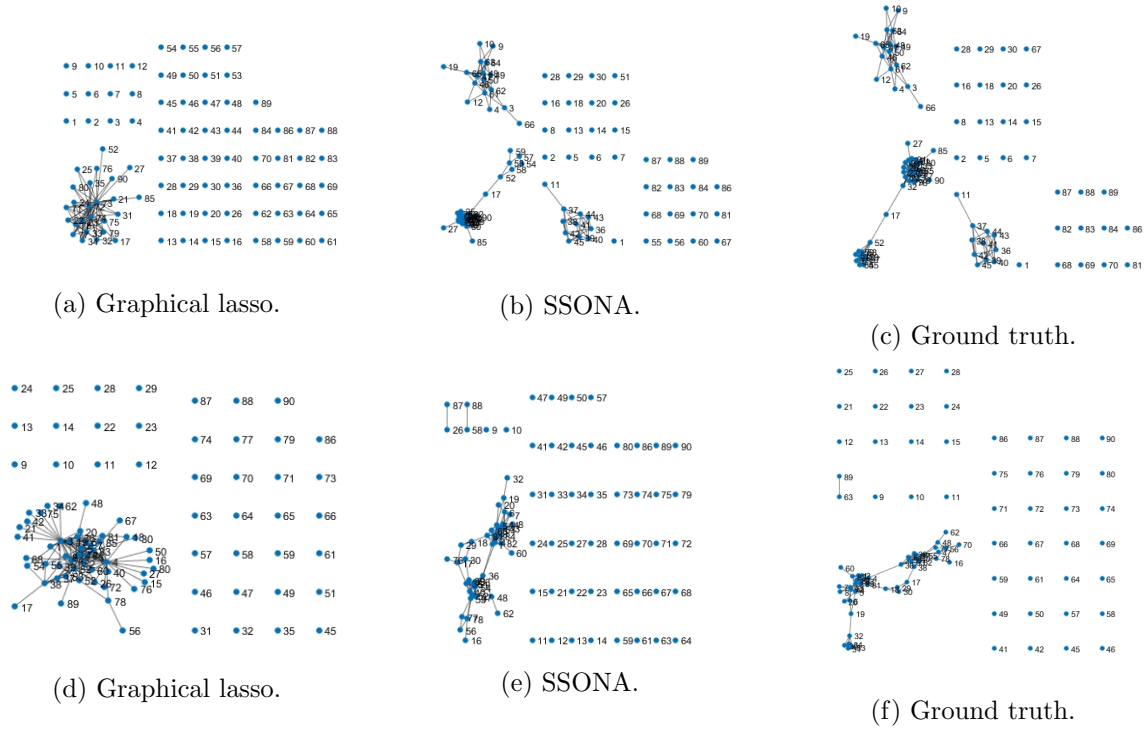
(d) Graphical lasso. (e) SSONA. (f) Ground truth.

Figure 12: Simulation for the Covariance graph model. Row I: Results for $p = 100$ and $m = 200$. Row II: Results for $p = 100$ and $m = 100$.

Method	**Average classification accuracy**	**Average number of genes selected**
Group lasso (Yuan and Lin, 2007)	0.815(0.046)	69.11(3.23)
Group lasso with overlap (Obozinski et al., 2011)	0.834(0.035)	57.30(2.71)
SSONA (4 structured matrices)	0.807(0.028)	61.44(2.80)
SSONA (6 structured matrices)	0.839(0.022)	56.111(2.100)

Table 1: Experimental results on lung cancer data over 10 replications (the standard deviations are reported in parentheses).

As is shown in Table 1, SSONA achieves higher classification accuracy than the group lasso and lower classification accuracy than the latent group lasso, although the performance of all three methods is very similar and within the variability induced by the replicates. However, our SSON based lasso does not require a priori knowledge of group structures, which is a prerequisite for the other two methods. One can easily improve the the classification accuracy and gene selection performance of SSONA by adding more structured matrices.



(a) Graphical lasso. (b) SSONA. (c) Ground truth.

(d) Graphical lasso. (e) SSONA. (f) Ground truth.

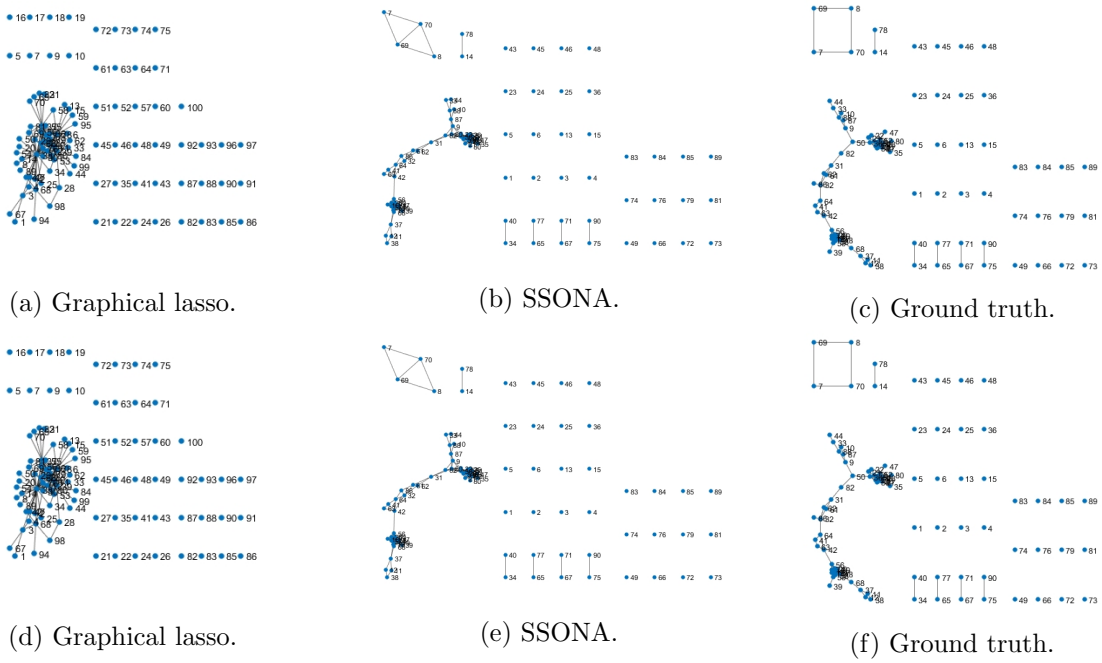
Figure 13: Simulation for the Gaussian graphical model with 10 latent variables. Row I: Results for $p = 100$ and $m = 200$. Row II: Results for $p = 100$ and $m = 100$.

In our experiments, SSONA selects the least number of genes and achieves the smallest standard deviation of average number of genes *without any priori knowledge*. Due to the different number of randomly selected genes, the average number of gene sometimes will be a non-integer.

4.2.2 SSONA for Document Classification Task

The next example involves a data set [3] containing 1427 documents with a corpus of size 17785 words. We randomly partition the data into 999 training, 214 validation and 214 test examples, corresponding to a 70/15/15 split (Rao et al., 2016). We first train a Latent Dirichlet Allocation based topics model (Blei et al., 2003) to assign the words to 100 "topics". These correspond to our groups, and since a single word can be assigned to multiple topics, the groups overlap. We then train a lasso logistic model using as outcome variable indicating whether the document discusses atheism or not , together with an overlapping group lasso and a SSON based lasso model where the tuning parameters are selected based on cross validation. Table 2 shows that the variants of the SSON yield almost the same misclassification rate compared to the other two methods, while it does not require a priori knowledge of group structures.

3. `http://qwone.com/jason/ 20Newsgroups/`



(a) Graphical lasso. (b) SSONA. (c) Ground truth.

(d) Graphical lasso. (e) SSONA. (f) Ground truth.

Figure 14: Simulation for the binary Ising Markov random field. Row I: Results for $p = 100$ and $m = 200$. Row II: Results for $p = 100$ and $m = 100$.

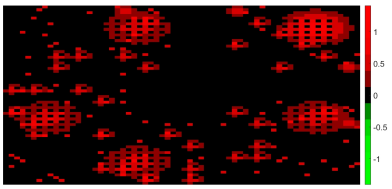
Method	**Misclassification Rate**
Group lasso (Yuan and Lin, 2007)	0.445
Group lasso with overlap (Obozinski et al., 2011)	0.390
SSONA (5 structured matrices)	0.435
SSONA (6 structured matrices)	0.421
SSONA (7 structured matrices)	0.401

Table 2: Misclassification rate on the test set for document classification.

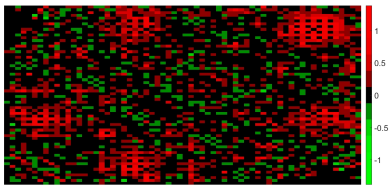
4.2.3 SSONA for structured subspace clustering

Our last example focuses on data clustering. The data come from multiple low-dimensional linear or affine subspaces embedded in a high-dimensional space. Our method is based on (11), wherein each point in a union of subspaces has a representation with respect to a dictionary formed by all other data points. In general, finding such a representation is NP hard. We apply our subspace clustering algorithm to a structured data in the presence of noise. The segmentation of the data is obtained by applying SSONA to the adjacency matrix built from the data. Our method can handle noise and missing data and is effective to detect the clusters.

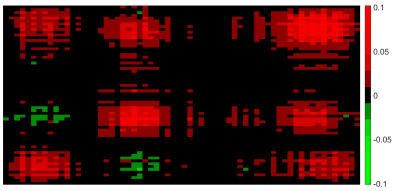
Figure 15 shows that our approach significantly outperforms state-of-the-art methods.



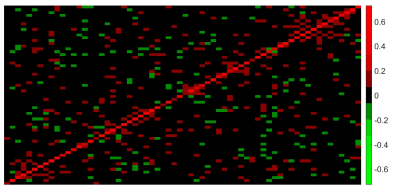
Ground Truth.



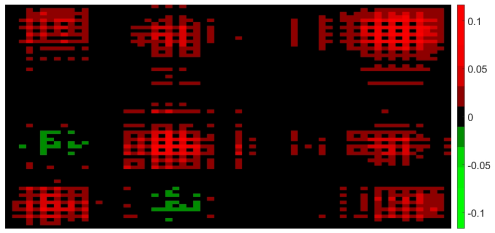
Ground Truth+ Noise.



LRR (Liu et al., 2013).



SSC (Elhamifar and Vidal, 2009).



SSONA.

Figure 15: Heatmap of different algorithms for detecting clusters in data.

4.3 Application to real data sets

Next, we use the SSON framework to analyze three data sets from molecular and social science domains. Although there is no known ground truth, the proposed framework recovers interesting patterns and highly interpretable structures.

Analysis of connectivity in the financial sector. We applied the SSON methodology to analyze connectivity in the financial sector. We use monthly stock returns data from August, 2001 to July, 2016 for three financial sectors, namely banks (BA), primary broker/dealers (PB), and insurance companies (INS). The data are obtained from the University of Chicago's Center for Research in Security Prices database (CRSP).

Our final sample covers 75 different institutions spanning a 16-year period. Figure 16 shows the mean (in %) of monthly stock returns across different sectors in each 3-year long rolling windows. As expected, the average returns are significantly lower during the financial 2007-2009 crisis period, compared to any other period in our sample. Indeed, looking across the sectors, all three sectors experienced diminished performance during the 2007-2009 crisis. Further, the almost linear ramp-up following 2009 clearly captures the recovery of financial stocks and the broader market.

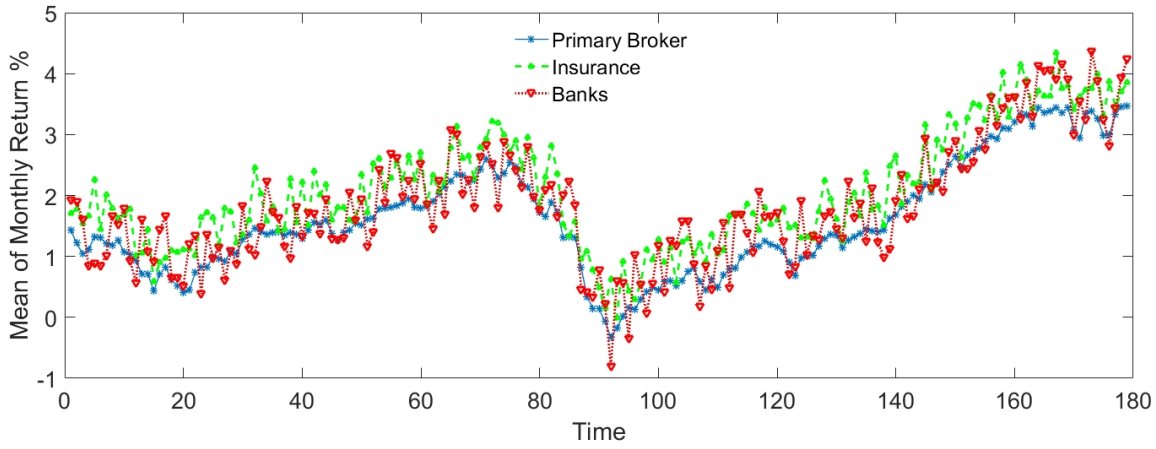
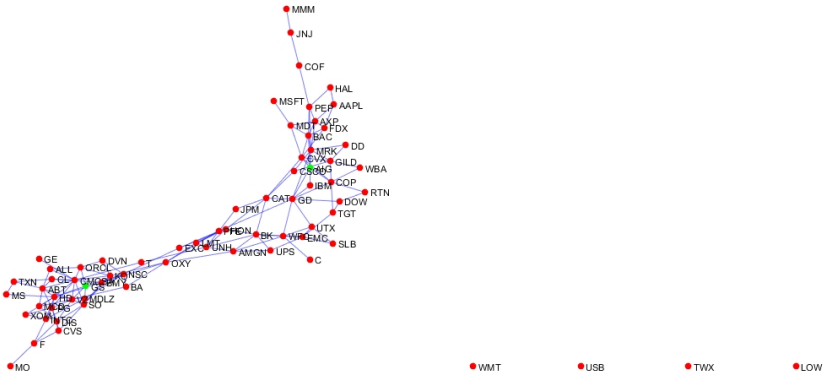


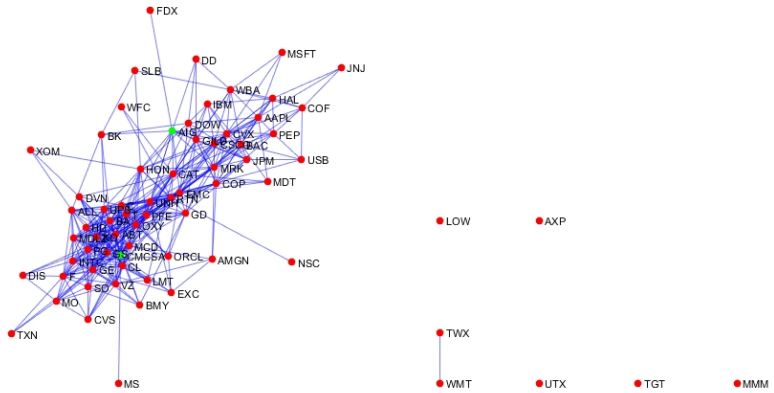
Figure 16: Average monthly return of firms in the three sectors- Bank, primary broker-dealer and insurance firms, in different 3-year rolling windows during 180 months. The figure shows diminished performance during the 2007-2009 crisis (time step : 80-100) and also clearly captures the strong recovery of stock performance starting in 2009.

Next, we estimate a measure of network connectivity for a sample of the 71 components of the SP100 index that were present during the entire 2001-16 period under consideration. Figure 17 depicts the network estimates of the transition (lead-lag) matrices using straight lasso VAR and SSONA based VAR for the January 2007 to Oct 2009 period. It can be seen that the lasso VAR estimates produce a more highly connected network, while the SSONA ones identify two more connected components. Both methods highlight the key role played by AIG and GS (Goldman Sachs), but the SSONA based network indicates that one dense connected component is centered around the former, while the other dense connected component around the latter. In summary, both methods capture the main connectivity patterns during the crisis period, but SSONA provides a more nuanced picture.

US House voting data set. We applied SSONA to describe the relationships amongst House Representatives in the U.S. Congress during the 2005-2006 period (109th Congress). The variables correspond to the 435 representatives, and the observations to the 1210 votes that the House deliberated and voted on during that period, which include bills, resolutions, motions, debates and roll call votes. The assumption of our model is that bills are i.i.d. sample from the same underlying Ising model. The votes are recorded as "yes" (encoded as "1") and "no" (encoded as "0"). Missing observations were replaced with the majority vote of the House members party on that particular vote. Following Guo et al. (2015), we used a bootstrap procedure with the proposed SSONA estimator to evaluate the confidence of the estimated edges. Specifically, we estimated the network for multiple bootstrap samples of the same size, and only retained the edges that appeared more that ω percent of the time. The goal of the analysis is to understand the type of relationships that existed among the House members in the 109th Congress. In particular, we wish to identify and interpret the presence of densely connected components, as well of sparse components. The heatmap of



(a) SSONA VAR



(b) Lasso VAR

Figure 17: Networks estimate by SSONA and Lasso VAR during crisis period of Jan 2007 to Oct 2009.

the adjacency matrix of the estimated network by using SSONA is depicted in Figure 18. It can be easily seen that there exist densely connected components in the network, a fact that the glasso algorithm (Friedman et al., 2008) fails to recover (see, Figure 19).

The network representation of subgraphs, with a cut-off value of 0.6, is given in Figures 20, 21 and 22. We only plot the edges associated with the subgraphs to enhance the visual reading of densely correlated areas. An interesting result of applying SSONA on this data set is the clear separation between members of the Democratic and Republican parties, as expected (see, Figures 20, 21 and 22). Moreover, voting relationships within the two parties exhibit a clustering structure, which a closer inspection of the votes and subsequent analysis showed was mainly driven by the position of the House member on the ideological/political spectrum.

Other interesting patterns emerging from the analysis is that SSONA recovers members of opposite parties as a sparse component in each subgraph (see, Figures 20, 21 and 22). For instance, Figure 21 shows that Republican members such as Simpson, Kirk and Hyde are sparsely connected in a clustered group of Democratic members. This is possibly due to

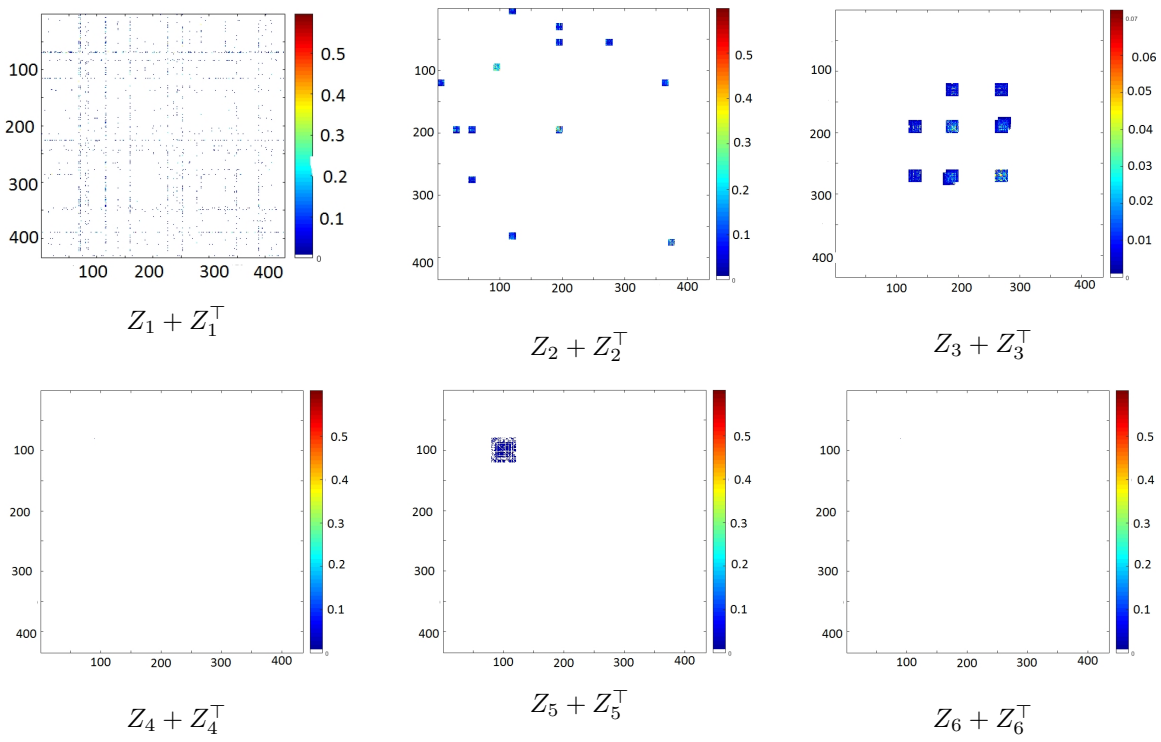


Figure 18: Heatmap of the structured precision matrix Θ decomposed into $Z_1 + Z_1^\top + \cdots + Z_6 + Z_6^\top$ in the House voting data, estimated by SSONA.

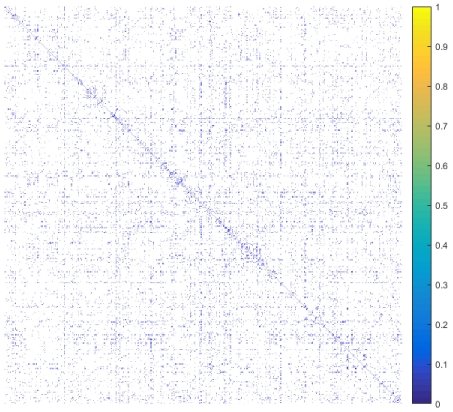


Figure 19: Heatmap of the inverse covariance matrix in the voting record of the U.S. House of Representatives, estimated by the graphical lasso method (Friedman et al., 2008).

the overall centrist record of Kirk and alignment of Hyde and Simpson on selected issues. Similarly, Figure 21 indicates that Democratic members Bishop, Hastings and Meek are approximately sparsely connected to a subgraph of Republican members. Bishop from Georgia has compiled a fairly conservative voting record. The same conclusion can be derived from Figure 21. Indeed, Figures 20, 21 and 22 reveals that there are strong positive

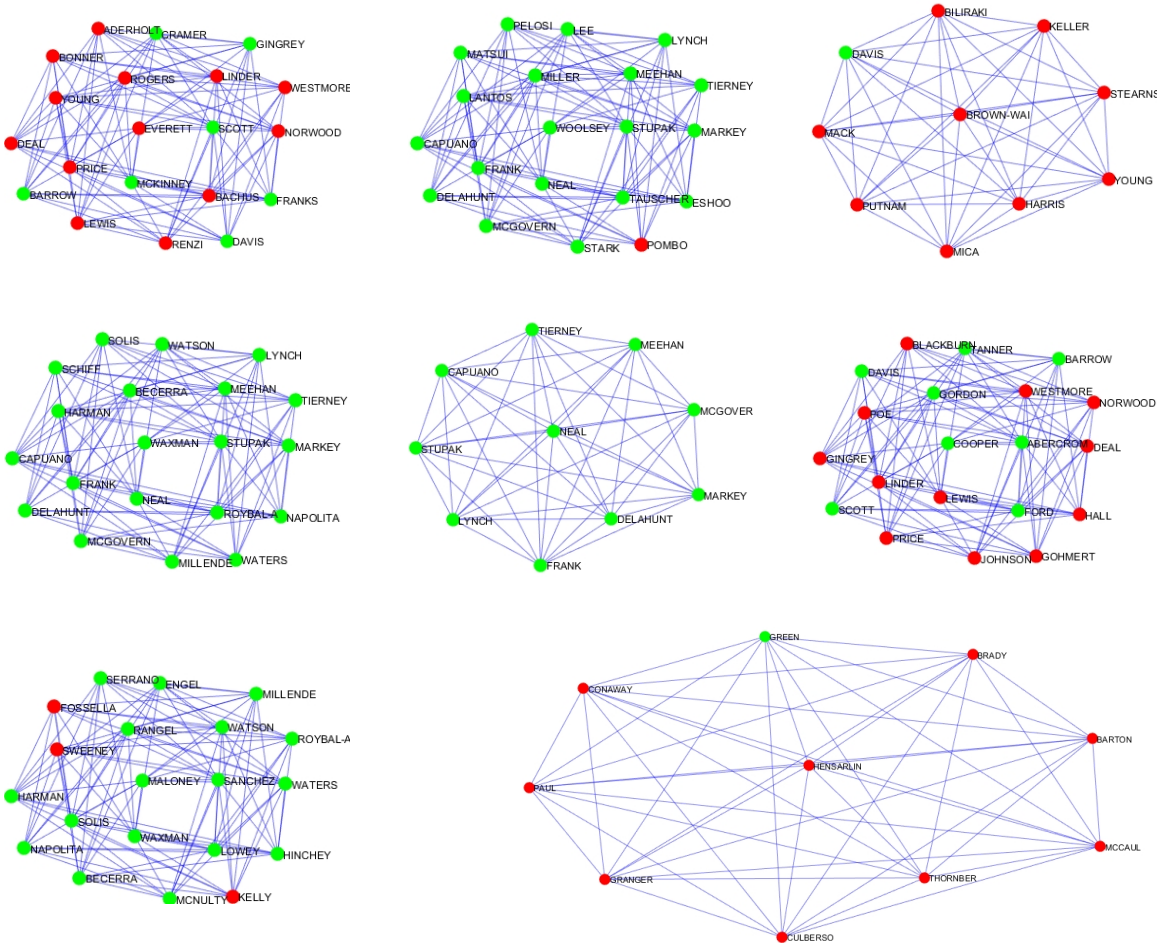


Figure 20: Dense subgraphs identified by SSONA for the House voting data with an inclusion cutoff value of 0.6. Subfigures correspond to a densely connected area in Figure 18 for the symmetric structured matrix $Z_2 + Z_2^\top$. The nodes represent House members, with red and green colored nodes corresponding to Republicans and Democrats, respectively. A blue line corresponds to an edge between two nodes.

associations between members of the same party and negative associations between members of opposite parties. Obviously, at the higher cutoff value the dependence structure between members of opposite parties becomes sparser.

Other patterns of interest include a strong dependence between members of two opposite parties in selected subgraphs when the members come from the same state, as is the case for New York state members Jerrold Nadler (D), Anthony D. Weiner (D), Ed Towns (D), Major Owens (D), Nydia Velzquez (D), Vito Fossella (R), Carolyn B. Maloney (D), Charles B. Rangel (D), Jos Serrano (D), Eliot L. Engel (D), Nita Lowey (D), Sue W. Kelly (R), John E. Sweeney (R), Michael R. McNulty (D), Maurice Hinchey (D), John M. McHugh (R), Sherwood Boehlert (R), Jim Walsh (R), Tom Reynolds (R), Brian Higgins (D) -see

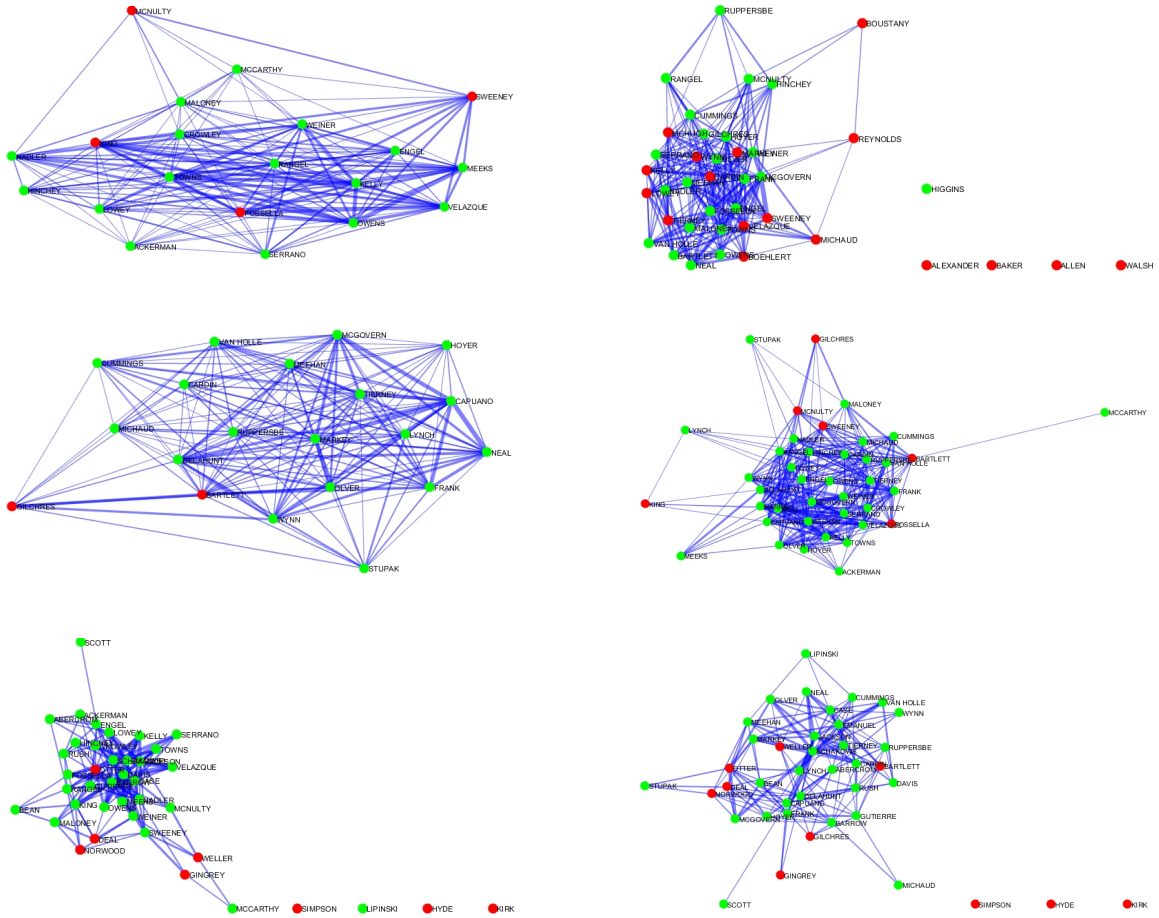


Figure 21: Dense subgraphs identified by SSONA for the House voting data with an inclusion cutoff value of 0.6. Subfigures correspond to a densely connected area in Figure 18 for the symmetric structured matrix $Z_3 + Z_3^\top$. The nodes represent House members, with red and green colored nodes corresponding to Republicans and Democrats, respectively. A blue line corresponds to an edge between two nodes.

Figure 21. However, in this instance, there is also a cluster of positive associations between Democrats.

In summary, SSONA provides deeper insights into relationships between House members, going beyond the obvious separation into two parties, according to their voting record.

Analysis of a breast cancer data set. We applied SSONA to a data set containing 800 gene expression measurements from large epithelial cells obtained from 255 patients with breast cancer. The goal is to capture regulatory interactions amongst the genes, as well as to identify genes that tend to have interactions with other genes in a group and hence act as master regulators, thus providing insights into the molecular circuitry of the disease. Figure 23 depicts the heat map of the estimated adjacency matrix for the breast cancer

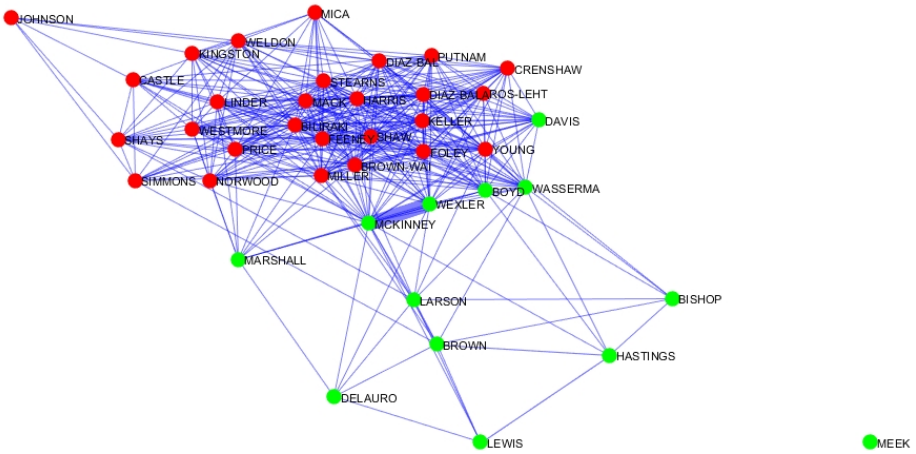


Figure 22: Dense subgraph identified by SSONA for the House voting data with an inclusion cutoff value of 0.6. Subfigures corresponds to a densely connected area in Figure 18 for the symmetric structured matrix $Z_5 + Z_5^\top$. The nodes represent House members, with red and blue node colors corresponding to Republicans and Democrats, respectively. A blue line corresponds to an edge between two nodes.

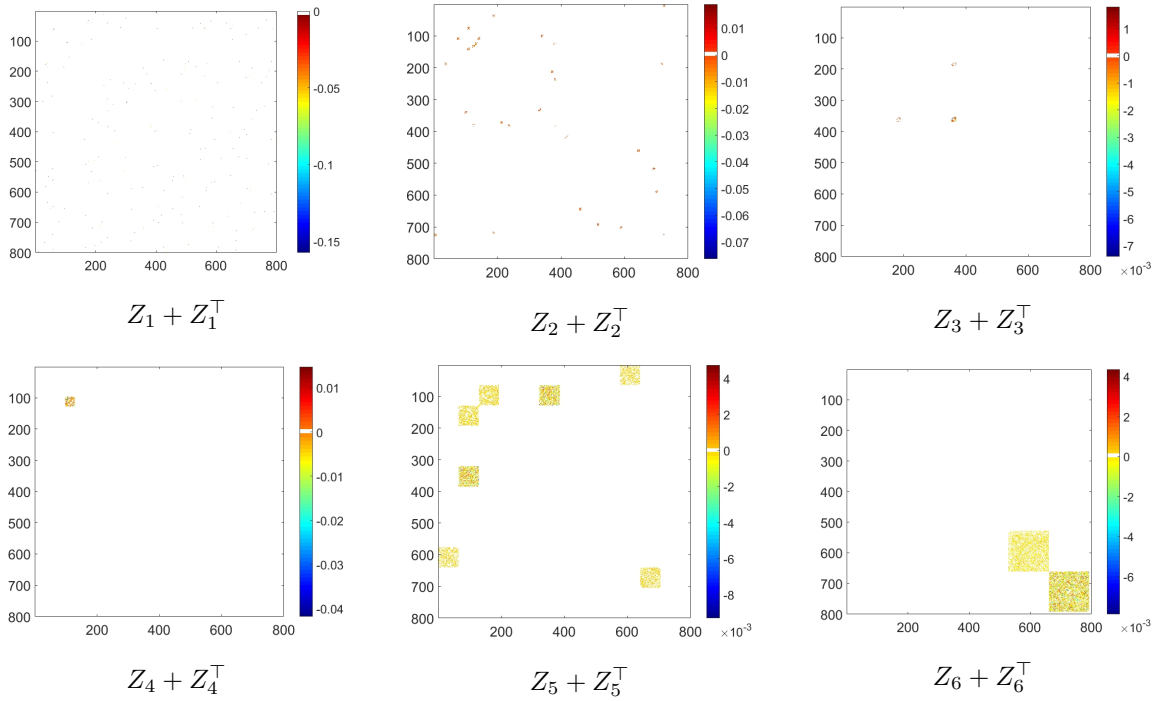


Figure 23: Heat map of the structured precision matrix Θ decomposed into $Z_1 + Z_1^\top + \cdots + Z_6 + Z_6^\top$ in the breast cancer data set, estimated by SSONA.

data set. As it is clear in Figure 23, $Z_2 + Z_2^\top, \ldots, Z_5 + Z_5^\top$ and $Z_6 + Z_6^\top$ show that selected

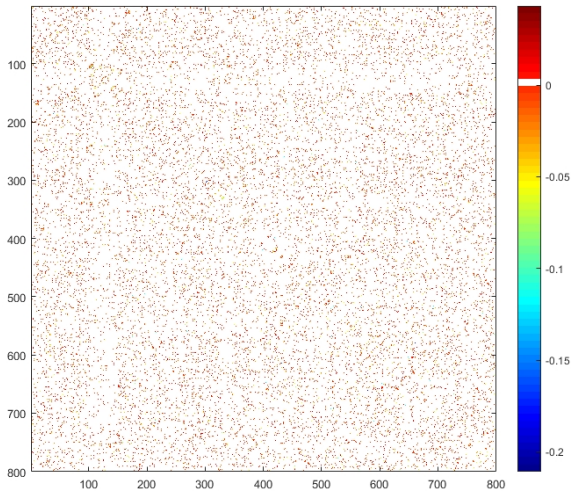


Figure 24: Heatmap of the inverse covariance matrix in the breast cancer data sets, estimated from graphical lasso (Friedman et al., 2008).

genes are densely connected, which is not the case when employing the the graphical lasso algorithm (see, Figure 24). Therefore, SSONA can provide an intuitive explanation of the relationships among the genes in the breast cancer data set (see, Figure 25 and 26 for two examples). These genes connectivity in the tumor samples may indicate a relationship that is common to an important subset of cancers. Many other genes belong to this network, each indicating a potentially interesting interaction in cancer biology. We omit the full list of densely connected genes in our estimated network and provide a complete list in the on-line supplementary materials available in the first author's homepage.

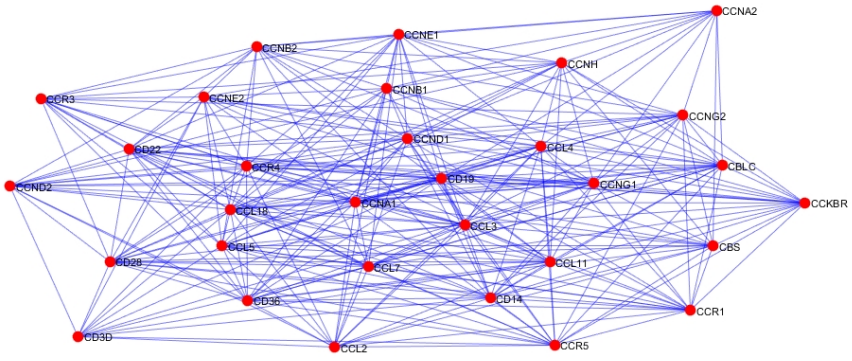


Figure 25: Network layout of grouped genes identified by SSONA for the breast cancer data set. Subfigure corresponds to a densely connected component in Figure 23 for the structured matrix $Z_4 + Z_4^\top$.

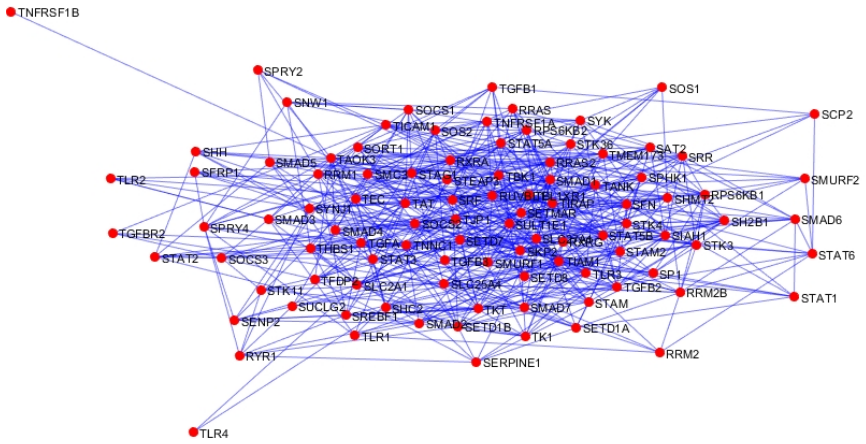


Figure 26: Network layout of grouped genes identified by SSONA for the breast cancer data set. Subfigure corresponds to a densely connected component in Figure 23 for the structured matrix $Z_6 + Z_6^\top$.

5. Conclusion

In this paper, a new structured norm minimization method for solving multi-structure graphical model selection problems is proposed. Using the proposed SSON, we can efficiently and accurately recover the underlying network structure. Our method utilizes a class of sparse structured norms in order to achieve higher order accuracy in approximating the decomposition of the parameter matrix in Markov Random Field and Gaussian Covariance Graph models. We also provide a brief discussion of its application to regression and classification problems. Further, we introduce a linearized multi-block ADMM algorithm to solve the resulting optimization problem. The global convergence of the algorithm is established without any upper bound on the penalty parameter. We applied the proposed methodology to a number of real and synthetic data sets that establish its overall usefulness and superior performance to competing methods in the literature.

Acknowledgments

The authors would like to thank the Editor and three anonymous referees for many constructive comments and suggestions that improved significantly the structure and readability of the paper. This work was supported in part by NSF grants DMS-1545277, DMS-1632730, NIH grant 1R01-GM1140201A1 and by the UF Informatics Institute.

Appendix A. Update for Θ

In each iteration of Algorithm 1 the update for Θ depends on the form of the loss function $g(\Theta)$. We consider the following cases to update Θ:

1. The update for Θ_1 in Algorithm 1 (step 2(a)) can be obtained by minimizing

$$\text{trace}(\hat{\Sigma}\Theta_1) - \log\det\Theta_1 + \frac{\gamma}{2}\|\Theta_1 - (\sum_{i=1}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k)\|_F^2,$$

with respect to Θ_1 (note that the constraint $\Theta_1 \in \mathcal{S}$ 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

$$\Theta_1 = \frac{1}{2}U\Big(D + \sqrt{D^2 + \frac{4}{\gamma}I}\Big)U^T,$$

where UDU^T stands for the eigen-decomposition of $\sum_{i=1}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k - \frac{1}{\gamma}\hat{\Sigma}$.

2. Update for Θ_2 in Step 2(a) of Algorithm 1 leads to the following optimization problem

$$\begin{aligned}\underset{\Theta_3\in\mathcal{S}}{\text{minimize}}\ \Phi(\Theta_2) &= \sum_{j=1}^{p}\sum_{j'=1}^{p}\theta_{jj'}(X^TX)_{jj'} - \sum_{i=1}^{m}\sum_{j=1}^{p}\log\Big(1+\exp[\theta_{jj}+\sum_{j'\neq j}\theta_{jj'}x_{ij'}]\Big)\\ &+ \frac{\gamma}{2}\|\Theta_2 - (\sum_{i=1}^{n} Z_i^k + {Z_i^k}^\top + E^k + \frac{1}{\gamma}\Lambda^k)\|_F^2. \qquad (33)\end{aligned}$$

We use a novel non-monotone version of the Barzilai-Borwein method (Barzilai and Borwein, 1988; Raydan, 1997; Fletcher, 2005; Ataee Tarzanagh et al., 2014) to solve (33). The details are given in Algorithm 2.

Algorithm 2 Non-monotone Barzilai Borwein Method for solving (33)

Initialize The parameters:

(a) $\Theta^0 = I$, $\Theta^1 = 2\Theta^0$, $\alpha^1 = 1$ and $t^0 = 10$.

(b) A positive sequence $\{\eta^t\}$ satisfying $\sum_{k=1}^{\infty}\eta^t = \eta < \infty$.

(c) Constants $\sigma > 0$, $\epsilon > 0$, and $\nu \in (0,1)$.

Iterate Until the stopping criterion $\dfrac{\|\Theta^t - \Theta^{t-1}\|_F^2}{\|\Theta^{t-1}\|_F^2} \leq \epsilon$ is met:

1. $\mathbb{G}^t = -\alpha^t\nabla\Phi(\Theta^t)$.
2. Set $\rho = 1$.
3. **If** $t > t^0$, **then**
   While $\|\Phi(\Theta^t + \rho^t\mathbb{G}^t)\|_F \leq \Phi(\Theta^t) + \eta^t - \sigma\rho^2{\alpha^t}^2\|\mathbb{G}^t\|_F^2$, **do**
   Set $\rho = \nu\rho$;
   EndWhile
   EndIf
4. Define $\rho^t = \rho$ and $\Theta^{t+1} = \Theta^t + \rho^t\mathbb{G}^t$.
5. Define $\alpha^{t+1} = \dfrac{\text{trace}\Big((\Theta^t - \Theta^{t+1})^T(\Theta^t - \Theta^{t+1})\Big)}{\text{trace}\Big((\nabla\Phi(\Theta^t) - \nabla\Phi(\Theta^{t+1}))^T(\Theta^t - \Theta^{t+1})\Big)}$

3. To update Θ_3 in step 2(a), using (11), we have that

$$\begin{aligned}\underset{\Theta_3}{\text{minimize}}\ \frac{1}{2}\|\Theta_3-\hat{\Sigma}\|_F^2 \quad &+ \quad \frac{\gamma}{2}\|\Theta_3-\Big(\sum_{i=1}^{n}Z_i^k+{Z_i^k}^\top+E^k+\frac{1}{\gamma}\Lambda^k\Big)\|_F^2\\ &= \quad \Big(\frac{1}{1+\gamma}(\hat{\Sigma}+\gamma(\sum_{i=1}^{n}Z_i^k+{Z_i^k}^\top+E^k)+\Lambda^k)\Big)_+\end{aligned}$$

where $V_+ = U_\dagger D_+ U_\dagger$ such that

$$UDU=\begin{pmatrix}U_\dagger & U_\ddagger\end{pmatrix}\begin{pmatrix}D_+ & 0\\ 0 & D_-\end{pmatrix}\begin{pmatrix}U_\dagger\\ U_\ddagger\end{pmatrix},$$

is the eigen-decomposition of the matrix V , and D_+ and D_- are the nonnegative and negative eigenvalues of V.

Appendix B. Convergence Analysis

Before establishing the main result on global convergence of the proposed ADMM algorithm, we provide the necessary definitions used in the proofs (for more details see Bolte et al. (2014)):

Definition 11 *(Kurdyka- Lojasiewicz property).*
The function f is said to have the Kurdyka- Lojasiewicz (K-L) property at point Z_0, if there exist $c_1>0$, $c_2>0$ and $\phi\in\Gamma_{c_2}$ such that for all

$$Z\in B(Z_0,c_1)\cap\{Z: f(Z_0)<f(Z)<f(Z_0)+c_2\},$$

the following inequality holds

$$\phi'\big(f(Z)-f(Z_0)\big)\,dist\big(0,\partial f(Z)\big)\geq 1,$$

where Γ_{c_2} stands for the class of functions $\phi:[0,c_2]\to\mathbb{R}^+$ with the properties:

(i) ϕ is continuous on $[0,c_2)$;

(ii) ϕ is smooth concave on $(0,c_2)$;

(iii) $\phi(0)=0$, $\nabla\phi(s)>0$, $\forall\ s\in(0,c_2)$.

Definition 12 *(Semi-algebraic sets and functions).*

(i) A subset $C\in\mathbb{R}^{n\times n}$ is semi-algebraic, if there exists a finite number of real polynomial functions h_{ij}, $s_{ij}:\mathbb{R}^{n\times n}\to\mathbb{R}$ such that

$$C=\cup_{i=1}^{\bar{p}}\cap_{j=1}^{\bar{q}}\{Z\in\mathbb{R}^{n\times n}:\ g_{ij}(Z)=0\quad and\quad s_{ij}(Z)<0\}.$$

(ii) A function $h:\mathbb{R}^{n\times n}\to(-\infty,+\infty]$ is called semi-algebraic, if its graph

$$\mathbb{G}(h):=\{(Z,y)\in\mathbb{R}^{n\times n+1}: h(Z)=y\},$$

is a semi-algebraic set in $R^{n\times n+1}$.

Definition 13 *(Sub-analytic sets and functions).*

(i) A subset $C \in \mathbb{R}^{n\times n}$ is sub-analytic, if there exists a finite number of real analytic functions h_{ij}, $s_{ij} : \mathbb{R}^{n\times n} \to \mathbb{R}$ such that

$$C = \cup_{i=1}^{\bar{p}} \cap_{j=1}^{\bar{q}} \{Z \in \mathbb{R}^d : g_{ij}(Z) = 0 \quad \text{and} \quad s_{ij}(Z) < 0\}.$$

(ii) A function h: $R^{n\times n} \to (-\infty, +\infty]$ is called sub-analytic, if its graph

$$\mathbb{G}(h) := \{(Z, y) \in \mathbb{R}^{n\times n+1} : h(Z) = y\}$$

is a sub-analytic set in $R^{n\times n+1}$.

It can be easily seen that both real analytic and semi-algebraic functions are sub-analytic. In general, the sum of two sub-analytic functions is not necessarily sub-analytic. However, it is easy to show that for two sub-analytic functions, if at least one function maps bounded sets to bounded sets, then their sum is also sub-analytic (Bolte et al., 2014).

Remark 14 *Each f_i in* (19) *is a convex semi-algebraic function (see, example 5.3 in (Bolte et al., 2014)), while the loss function $\mathcal{G}$ in* (6), (9), (11), (13), *and* (18) *is sub-analytic (even analytic). Since each function f_i maps bounded sets to bounded sets, we can conclude that the augmented Lagrangian function*

$$\begin{aligned}\mathcal{L}_\gamma(\Theta, Z_1, \ldots, Z_n, E; \Lambda) &= \mathcal{G}(X, \Theta) + f_1(Z_1) + \cdots + f_n(Z_n) + f_e(E) \\ &- \langle \Lambda, \Theta - \sum_{i=1}^{n} Z_i + Z_i^\top - E\rangle \\ &+ \frac{\gamma}{2}\|\Theta - \sum_{i=1}^{n} Z_i + Z_i^\top - E\|_F^2,\end{aligned}$$

which is the summation of sub-analytic functions is itself sub-analytic. All sub-analytic functions which are continuous over their domain satisfy a K-L inequality, as well as some, but not all, convex functions (see Bolte et al., 2014 for details and a counterexample). Therefore, the augmented Lagrangian function $\mathcal{L}_\gamma$ satisfies the K-L property.

Next, we establish a series of lemmas used in the proof of Theorem 10.

Lemma 15 *Let $U^k := (\Theta^k, Z_1^k, \ldots, Z_n^k, E^k; \Lambda^k)$ be a sequence generated by Algorithm 1, then there exists a positive constant ϑ such that*

$$\begin{aligned}\mathcal{L}_\gamma(U^{k+1}) &\leq \mathcal{L}_\gamma(U^k) - \frac{\vartheta}{2}\Big(\|\Theta^k - \Theta^{k+1}\|_F \\ &+ \sum_{i=1}^{n} \|Z_i^k - Z_i^{k+1}\|_F + \|E^k - E^{k+1}\|_F + \|\Lambda^k - \Lambda^{k+1}\|_F\Big). \qquad (34)\end{aligned}$$

Proof. Using the first-order optimality conditions for (21) and the convexity of $\mathcal{G}(X,\Theta)$, we obtain

$$\begin{aligned}
0 &= \langle \Theta^k - \Theta^{k+1}, \nabla\mathcal{G}(X,\Theta^{k+1}) - \Lambda^k + \gamma(\Theta^{k+1} - \sum_{i=1}^n Z_i^k + {Z_i^k}^\top - E^k)\rangle \\
&\leq \mathcal{G}(X,\Theta^k) - \mathcal{G}(X,\Theta^{k+1}) - \langle \Theta^k - \Theta^{k+1}, \Lambda^k\rangle \\
&+ \gamma\langle \Theta^k - \Theta^{k+1}, \Theta^{k+1} - \sum_{i=1}^n Z_i^k + {Z_i^k}^\top - E^k\rangle \\
&= \mathcal{G}(X,\Theta^k) - \langle \Theta^k, \Lambda^k\rangle + \frac{\gamma}{2}\sum_{i=1}^n \|\Theta^k - \sum_{i=1}^n Z_i^k + {Z_i^k}^\top - E^k\|_F^2 - \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2 \\
&- \Big(\mathcal{G}(X,\Theta^{k+1}) - \langle \Theta^{k+1}, \Lambda^k\rangle + \frac{\gamma}{2}\|\Theta^{k+1} - \sum_{i=1}^n Z_i^k + {Z_i^k}^\top - E^k\|_F^2\Big) \\
&= \mathcal{L}_\gamma(U^k) - \mathcal{L}_\gamma(\Theta^{k+1}, Z_1^k, \ldots, Z_n^k, E^k; \Lambda^k) - \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2, \qquad (35)
\end{aligned}$$

where the second equality follows from the fact that

$$(u_1 - u_2)^T(u_3 - u_1) = \frac{1}{2}\Big(\|u_2 - u_3\|_F^2 - \|u_1 - u_2\|_F^2 - \|u_1 - u_3\|_F^2\Big).$$

Using (22), (23) and Lemma 7, we have that

$$\begin{aligned}
\mathcal{L}_\gamma(\Theta^{k+1}, Z_1^k, Z_2^k, \ldots, E^k; \Lambda^k) &- \mathcal{L}_\gamma(\Theta^{k+1}, Z_1^{k+1}, Z_2^k, \ldots, E^k; \Lambda^k) \\
&- \frac{(\gamma\varrho - L_{H_1})}{2}\|Z_1^k - Z_1^{k+1}\|_F^2 \\
&\geq 0, \\
\mathcal{L}_\gamma(\Theta^{k+1}, \ldots, Z_{i-1}^{k+1}, Z_i^k, \ldots, E^k; \Lambda^k) &- \mathcal{L}_\gamma(\Theta^{k+1}, \ldots, Z_i^{k+1}, Z_{i+1}^k, \ldots, E^k; \Lambda^k) \\
&- \frac{(\gamma\varrho - L_{H_i})}{2}\|Z_i^k - Z_i^{k+1}\|_F^2 \\
&\geq 0, \qquad i = 2, \ldots, n, \qquad (36)
\end{aligned}$$

where L_{H_i} is a Lipschitz constant of the gradient $\nabla H_i(Z_i)$, and $\varrho \geq \frac{L_{H_i}}{\gamma}$, $(i = 1, \ldots, n)$ is a proximal parameter.

Following the same steps as (35), we have that

$$\begin{aligned}
\mathcal{L}_\gamma(\Theta^k, Z_1^{k+1}, \ldots, Z_n^{k+1}, E^k; \Lambda^k) &- \mathcal{L}_\gamma(\Theta^{k+1}, Z_1^{k+1}, \ldots, Z_n^{k+1}, E^{k+1}; \Lambda^k) \\
&- \frac{\gamma}{2}\|E^k - E^{k+1}\|_F^2 \\
&\geq 0, \qquad (37)
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{L}_\gamma(\Theta^{k+1}, Z_1^{k+1}, \ldots, Z_n^{k+1}, E^{k+1}; \Lambda^k) &- \mathcal{L}_\gamma(\Theta^{k+1}, Z_1^{k+1}, \ldots, Z_n^{k+1}, E^{k+1}; \Lambda^{k+1}) \\
&- \frac{\lambda_e^2}{\gamma}\|E^k - E^{k+1}\|_F^2 \\
&\geq 0. \qquad (38)
\end{aligned}$$

Let

$$\hat{\gamma} := \max(\gamma\varrho - L_{H_1}, \ldots, \gamma\varrho - L_{H_n}), \quad \bar{\gamma} := \frac{\gamma^2 - 2\lambda_e^2}{\gamma(1+\lambda_e^2)}, \quad \vartheta := \max(\hat{\gamma}, \bar{\gamma}, \gamma).$$

Then, using (35)– (38), and $\gamma \geq \sqrt{2}\lambda_e$, we have

$$\begin{aligned}
&\mathcal{L}_\gamma(U^k) - \mathcal{L}_\gamma(U^{k+1}) \geq \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2 \\
+ \;& \frac{\hat{\gamma}}{2}\sum_{i=1}^n \|Z_i^k - Z_i^{k+1}\|_F^2 + \frac{\gamma^2 - 2\lambda_e^2}{2\gamma}\|E^k - E^{k+1}\|_F^2, \\
= \;& \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2 + \frac{\hat{\gamma}}{2}\sum_{i=1}^n \|Z_i^k - Z_i^{k+1}\|_F^2 + \frac{\bar{\gamma}}{2}\|E^k - E^{k+1}\|_F^2 + \frac{\lambda_e^2\bar{\gamma}}{2}\|E^k - E^{k+1}\|_F^2, \\
= \;& \frac{\gamma}{2}\|\Theta^k - \Theta^{k+1}\|_F^2 + \frac{\hat{\gamma}}{2}\sum_{i=1}^n \|Z_i^k - Z_i^{k+1}\|_F^2 + \frac{\bar{\gamma}}{2}\Big(\|E^k - E^{k+1}\|_F^2 + \|\Lambda^k - \Lambda^{k+1}\|_F^2\Big), \\
\geq \;& \frac{\vartheta}{2}\Big(\|\Theta^k - \Theta^{k+1}\|_F^2 + \sum_{i=1}^n \|Z_i^k - Z_i^{k+1}\|_F^2 + \|E^k - E^{k+1}\|_F^2 + \|\Lambda^k - \Lambda^{k+1}\|_F^2\Big).
\end{aligned}$$

□

Lemma 16 *Let $U^k = (\Theta^k, Z_1^k, \ldots X_n^k, E^k, \Lambda^k)$ be a sequence generated by Algorithm 1. Then, there exists a subsequence U^{k_s} of $\{U^k\}$, such that*

$$\lim_{s\to\infty} \mathcal{G}(X, \Theta^{k_s}) = g(\Theta^*), \quad \lim_{s\to\infty} f_i(Z_i^{k_s}) = f_i(Z_i^*), \quad \lim_{s\to\infty} f_e(E_i^{k_s}) = f_e(E_i^*),$$

where

$$\lim_{s\to\infty} U^{k_s} = (\Theta^*, Z_1^*, \ldots, Z_n^*, E^*, \Lambda^*).$$

Proof. Let $\Upsilon^{k+1} = \Theta^{k+1} - \sum_{i=1}^n Z_i^{k+1} + Z_i^{k+1^\top} - E^{k+1}$. Using the quadratic function $f_e(E) = \frac{\lambda_e}{2}\|E\|_F^2$, we have that

$$\begin{aligned}
f_e(E^{k+1} - \Upsilon^{k+1}) &= \frac{\lambda_e}{2}\|E^{k+1} - \Upsilon^{k+1}\|_F^2 \\
&= \frac{\lambda_e}{2}\|E^{k+1}\|^2 - \lambda_e\langle E^{k+1}, \Upsilon^{k+1}\rangle + \frac{\lambda_e}{2}\|\Upsilon^{k+1}\|_F^2. \qquad (39)
\end{aligned}$$

Using (39) and the fact that each function f_i is lower bounded, there exists $\underline{\mathcal{L}}$, such that

$$\begin{aligned}
\mathcal{L}_\gamma(U^{k+1}) &= \mathcal{G}(X, \Theta^{k+1}) + f_1(Z_1^{k+1}) + \ldots f_n(Z_n^{k+1}) + \frac{\lambda_e}{2}\|E^{k+1} - \Upsilon^{k+1}\|_F^2 \\
&+ \frac{\gamma - \lambda_e}{2}\|\Upsilon^{k+1}\|_F^2 \geq \underline{g} + \underline{f_1} + \cdots + \underline{f_n} \geq \underline{\mathcal{L}}, \qquad (40)
\end{aligned}$$

since $\mathcal{G}(X, \Theta^{k+1})$ and $f_i(Z_i^{k+1})(i = 1, \ldots, n)$ are all lower bounded.

Now, using Lemma 15, we have that

$$\frac{\vartheta}{2}\sum_{k=0}^{K}\Big(\|\Theta^k-\Theta^{k+1}\|_F^2+\sum_{i=1}^{n}\|Z_i^k-Z_i^{k+1}\|_F^2 \quad + \quad \|E^k-E^{k+1}\|_F^2+\|\Lambda^k-\Lambda^{k+1}\|_F^2\Big)$$
$$\leq \quad \mathcal{L}_\gamma(U^0)-\underline{\mathcal{L}}. \tag{41}$$

Lemma 15 together with (41) shows that $\mathcal{L}_\gamma(U^k)$ converges to $\mathcal{L}_\gamma(U^*)$. Note that (41) and the coerciveness of $\mathcal{G}(X,\Theta)$ and f_i $(i=1,\ldots,n)$ imply that $\{(\Theta^k,Z_1^k,\ldots,Z_n^k)\}$ is a bounded sequence. This together with the updating formula of Λ^{k+1} and (41) yield the boundedness of E^{k+1}. Moreover, the fact that $\Lambda^k=-\lambda_e E^k$, gives the boundedness of Λ^k, which implies that the entire sequence $\{U^k\}$ is a bounded one. Therefore, there exists a subsequence

$$U^{k_s}=(\Theta^{k_s},Z_1^{k_s},\ldots,Z_n^{k_s},E^{k_s};\Lambda^{k_s}),\qquad s=0,1,\ldots$$

such that $U^{k_s}\to U^*$ as $s\to\infty$.

Now, using the fact that $\mathcal{G}(X,\Theta)$, $f_i(Z_i)$ $(i=1,\ldots,n)$ and $f_e(E)$ are continuous functions, we have that

$$\lim_{s\to\infty}\mathcal{G}(X,\Theta^{k_s})=g(\Theta^*),\quad \lim_{s\to\infty}f_i(Z_i^{k_q})=f_i(Z_i^*),\quad \lim_{s\to\infty}f_e(E_i^{k_q})=f_e(E_i^*).$$

□

Lemma 17 *Algorithm 1 either stops at a stationary point of the problem* (19) *or generates an infinite sequence* $\{U^k\}$*, so that any limit point of* $\{U^k\}$ *is a critical point of* $\mathcal{L}_\gamma(U^k)$ (19)*.*

Proof. From the definition of the augmented Lagrangian function in (19), we have that

$$\begin{aligned}
&\nabla\mathcal{G}(X,\Theta^{k+1})-\Lambda^{k+1}+\gamma\Upsilon^{k+1}=\nabla_\Theta\mathcal{L}_\gamma(U^{k+1}),\\
&\partial f_i(Z_i^{k+1})-\Lambda^{k+1}-{\Lambda^{k+1}}^\top-\gamma(\Upsilon^{k+1}+{\Upsilon^{k+1}}^\top)\in\partial_{Z_i}\mathcal{L}_\gamma(U^{k+1}),\quad i=1,\ldots,n,\\
&\lambda_e E^{k+1}+\Lambda^{k+1}-\gamma\Upsilon^{k+1}=\nabla_E\mathcal{L}_\gamma(U^{k+1}),\\
&\gamma\Upsilon^{k+1}=-\nabla_\Lambda\mathcal{L}_\gamma(U^{k+1}),
\end{aligned}\tag{42}$$

where $\Upsilon^{k+1}=\Theta^{k+1}-\sum_{i=1}^n Z_i^{k+1}+{Z_i^{k+1}}^\top-E^{k+1}$.

Moreover, the updating formula of Λ^{k+1}, (20) and (28) yields that

$$
\begin{aligned}
\nabla \mathcal{G}(X, \Theta^{k+1}) - \Lambda^{k+1} &= \gamma\Big(\Theta^{k+1} - \Theta^k \\
&+ \sum_{i=1}^n Z_i^k - Z_i^{k+1} + (Z_i^k - Z_i^{k+1})^\top + E^k - E^{k+1}\Big) \\
\partial f_1(Z_1^{k+1}) - \Lambda^{k+1} - {\Lambda^{k+1}}^\top &= \gamma\varrho(Z_1^k - Z_1^{k+1}) + \gamma\Big(\Theta^{k+1} - \Theta^k \\
&+ (\Theta^{k+1} - \Theta^k)^\top + \sum_{i=1}^n Z_i^k - Z_i^{k+1} \qquad (43) \\
&+ (Z_i^k - Z_i^{k+1})^\top + E^k - E^{k+1} + (E^k - E^{k+1})^\top\Big) \\
\partial f_i(Z_i^{k+1}) - \Lambda^{k+1} - {\Lambda^{k+1}}^\top &= \gamma\varrho(Z_i^k - Z_i^{k+1}) \\
&+ \gamma\Big(\Theta^{k+1} - \Theta^k + (\Theta^{k+1} - \Theta^k)^\top \\
&+ \sum_{j=i}^n Z_i^k - Z_i^{k+1} + (Z_i^k - Z_i^{k+1})^\top \\
&+ E^k - E^{k+1} + (E^k - E^{k+1})^\top\Big) \qquad i = 2, \ldots, n, \\
\lambda_e E^{k+1} &+ \Lambda^{k+1} = 0. \qquad (44)
\end{aligned}
$$

Combining (42), (43), and the updating formula of Λ^{k+1}, we have that

$$
(\hbar_\Theta^{k+1}, \hbar_1^{k+1}, \ldots, \hbar_n^{k+1}, \hbar_E^{k+1}, \hbar_\Lambda^{k+1}) \in \partial \mathcal{L}_\gamma(U^{k+1}), \qquad (45)
$$

where

$$
\begin{aligned}
\hbar_\Theta^{k+1} &:= \Lambda^k - \Lambda^{k+1} + \gamma\Big(\Theta^{k+1} - \Theta^k + \sum_{i=1}^n Z_i^k - Z_i^{k+1} + (Z_i^k - Z_i^{k+1})^\top + E^k - E^{k+1}\Big) \\
\hbar_{Z_1}^{k+1} &:= \Lambda^k - \Lambda^{k+1} + (\Lambda^k - \Lambda^{k+1})^\top + \gamma\varrho(Z_1^k - Z_1^{k+1}) \\
&+ \gamma\Big(\Theta^{k+1} - \Theta^k + (\Theta^{k+1} - \Theta^k)^\top + \sum_{i=1}^n Z_i^k - Z_i^{k+1} + (Z_i^k - Z_i^{k+1})^\top \\
&+ E^k - E^{k+1} + (E^k - E^{k+1})^\top\Big) \\
\hbar_{Z_i}^{k+1} &:= \Lambda^k - \Lambda^{k+1} + (\Lambda^k - \Lambda^{k+1})^\top + \gamma\varrho(Z_i^k - Z_i^{k+1}) \\
&+ \gamma\Big(\Theta^{k+1} - \Theta^k + (\Theta^{k+1} - \Theta^k)^\top + \sum_{j=i}^n Z_i^k - Z_i^{k+1} + (Z_i^k - Z_i^{k+1})^\top \\
&+ E^k - E^{k+1} + (E^k - E^{k+1})^\top\Big), \qquad i = 2, \ldots, n, \\
\hbar_E^{k+1} &:= \Lambda^k - \Lambda^{k+1}, \\
\hbar_\Lambda^{k+1} &:= \frac{1}{\gamma}(\Lambda^{k+1} - \Lambda^k), \qquad (46)
\end{aligned}
$$

Now, using (41), we obtain that

$$\lim_{k\to\infty}(\|\hbar_{\Theta}^{k+1}\|_F, \|\hbar_{Z_1}^{k+1}\|_F, \ldots, \|\hbar_{Z_n}^{k+1}\|_F, \|\hbar_{E}^{k+1}\|_F; \|R_{\Lambda}^{k+1}\|_F) = (0, \ldots, 0). \tag{47}$$

Suppose that Algorithm 1 does not stop at a stationary point. Using Lemma 16, there exists a subsequence U^{k_s}, such that $U^{k_s} \to U^*$ as $s \to \infty$. Using (45) and (47), we conclude that $(0, \ldots, 0) \in \partial\mathcal{L}_\gamma(U^*)$. □

Proof of Theorem 10. Lemmas 16 and 17 imply that $\{U^k\}$ is a bounded sequence and the set of limit points of $\{U^k\}$ starting from U^0 is non-empty, respectively. Moreover, Lemma 5 and Remark 5 of (Bolte et al., 2014) imply that the set of limit points of $\{U^k\}$ starting from U^0 is compact. The remainder of the proof of this Theorem follows along similar lines to the proof of Theorem 1 in (Bolte et al., 2014), by utilizing the K-L property of the problem (19) (see, Remark 14). □

References

D Ataee Tarzanagh, M Reza Peyghami, and H Mesgarani. A new nonmonotone trust region method for unconstrained optimization equipped by an efficient adaptive radius. *Optimization Methods and Software*, 29(4):819–836, 2014.

Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

Jonathan Barzilai and Jonathan M Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

Sumanta Basu and George Michailidis. Regularized estimation in sparse high-dimensional time series models. *The Annals of Statistics*, 43(4):1535–1567, 2015.

Sumanta Basu, Ali Shojaie, and George Michailidis. Network granger causality with inherent grouping structure. *The Journal of Machine Learning Research*, 16(1):417–453, 2015.

Quentin Berthet, Philippe Rigollet, and Piyush Srivastava. Exact recovery in the Ising block model. *arXiv:1612.03880*, 2016.

David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2): 459–494, 2014.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

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

Venkat Chandrasekaran, Pablo A Parrilo, and Alan S Willsky. Latent variable graphical model selection via convex optimization. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 1610–1613. IEEE, 2010.

Caihua Chen, Bingsheng He, Yinyu Ye, and Xiaoming Yuan. The direct extension of admm for multi-block convex minimization problems is not necessarily convergent. *Mathematical Programming*, 155(1-2):57–79, 2016.

Victor Chernozhukov, Christian Hansen, and Yuan Liao. A lava attack on the recovery of sums of dense and sparse signals. *The Annals of Statistics*, 45(1):39–76, 2017.

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

Damek Davis and Wotao Yin. A three-operator splitting scheme and its optimization applications. *Set-Valued and Variational Analysis*, pages 1–30, 2015.

Elizabeth D Dolan and Jorge J Moré. Benchmarking optimization software with performance profiles. *Mathematical programming*, 91(2):201–213, 2002.

Mathias Drton and Thomas S Richardson. A new algorithm for maximum likelihood estimation in gaussian graphical models for marginal independence. In *Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence*, pages 184–191. Morgan Kaufmann Publishers Inc., 2002.

Mathias Drton and Thomas S Richardson. Graphical methods for efficient likelihood inference in gaussian covariance models. *Journal of Machine Learning Research*, 9(May): 893–914, 2008.

Ehsan Elhamifar and René Vidal. Sparse subspace clustering. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2790–2797. IEEE, 2009.

Roger Fletcher. On the barzilai-borwein method. *Optimization and control with applications*, pages 235–256, 2005.

Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

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

Roger Guimera and Luis A Nunes Amaral. Functional cartography of complex metabolic networks. *Nature*, 433(7028):895–900, 2005.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011a.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Asymptotic properties of the joint neighborhood selection method for estimating categorical markov networks. *arXiv preprint math.PR/0000000*, 2011b.

Jian Guo, Jie Cheng, Elizaveta Levina, George Michailidis, and Ji Zhu. Estimating heterogeneous graphical models for discrete data with an application to roll call voting. *The Annals of Applied Statistics*, 9(2):821, 2015.

Davood Hajinezhad and Mingyi Hong. Nonconvex alternating direction method of multipliers for distributed sparse principal component analysis. In *Signal and Information Processing (GlobalSIP), 2015 IEEE Global Conference on*, pages 255–259. IEEE, 2015.

Davood Hajinezhad, Mingyi Hong, Tuo Zhao, and Zhaoran Wang. Nestt: A nonconvex primal-dual splitting method for distributed and stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 3215–3223, 2016.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Holger Höfling and Robert Tibshirani. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods. *Journal of Machine Learning Research*, 10(Apr):883–906, 2009.

Mingyi Hong and Zhi-Quan Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162(1-2):165–199, 2017.

Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

Laurent Jacob, Guillaume Obozinski, and Jean-Philippe Vert. Group lasso with overlap and graph lasso. In *Proceedings of the 26th annual international conference on machine learning*, pages 433–440. ACM, 2009.

Noureddine El Karoui. Operator norm consistent estimation of large-dimensional sparse covariance matrices. *The Annals of Statistics*, pages 2717–2756, 2008.

Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of Markov networks using ℓ_1-regularization. In *Advances in neural Information processing systems*, pages 817–824, 2007.

Anna CF Lewis, Charlotte M Deane, Mason A Porter, and Nick S Jones. The function of communities in protein interaction networks at multiple scales. *BMC systems biology*, 4 (1):100, 2010.

Lun Li, David Alderson, John C Doyle, and Walter Willinger. Towards a theory of scale-free graphs: Definition, properties, and implications. *Internet Mathematics*, 2(4):431–523, 2005.

Fredrik Liljeros, Christofer R Edling, Luis A Nunes Amaral, H Eugene Stanley, and Yvonne Åberg. The web of human sexual contacts. *Nature*, 411(6840):907–908, 2001.

Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. On the global linear convergence of the ADMM with multiblock variables. *SIAM Journal on Optimization*, 25(3):1478–1497, 2015.

Tianyi Lin, Shiqian Ma, and Shuzhong Zhang. Iteration complexity analysis of multi-block ADMM for a family of convex minimization without strong convexity. *Journal of Scientific Computing*, 69(1):52–81, 2016.

Zhouchen Lin, Risheng Liu, and Zhixun Su. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Advances in neural information processing systems*, pages 612–620, 2011.

Guangcan Liu, Zhouchen Lin, Shuicheng Yan, Ju Sun, Yong Yu, and Yi Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):171–184, 2013.

Qiang Liu and Alexander Ihler. Learning scale free networks by reweighted l1 regularization. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 40–48, 2011.

Risheng Liu, Zhouchen Lin, and Zhixun Su. Linearized alternating direction method with parallel splitting and adaptive penalty for separable convex programs in machine learning. *Machine Learning*, 99(2), 2015.

Lennart Ljung. System identification. In *Signal analysis and prediction*, pages 163–173. Springer, 1998.

Helmut Lütkepohl. *New introduction to multiple time series analysis*. Springer Science & Business Media, 2005.

Shiqian Ma, Lingzhou Xue, and Hui Zou. Alternating direction methods for latent variable gaussian graphical model selection. *Neural computation*, 25(8):2172–2198, 2013.

Karthik Mohan, Mike Chung, Seungyeop Han, Daniela Witten, Su-In Lee, and Maryam Fazel. Structured learning of Gaussian graphical models. In *Advances in neural information processing systems*, pages 620–628, 2012.

Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine learning*, 52(1):91–118, 2003.

Mark EJ Newman. The structure of scientific collaboration networks. *Proceedings of the National Academy of Sciences*, 98(2):404–409, 2001.

Mark EJ Newman. Communities, modules and large-scale structure in networks. *Nature Physics*, 8(1):25–31, 2012.

Mark EJ Newman and Michelle Girvan. Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113, 2004.

Guillaume Obozinski, Laurent Jacob, and Jean-Philippe Vert. Group lasso with overlaps: the latent group lasso approach. *arXiv preprint arXiv:1110.0413*, 2011.

Nikhil Rao, Robert Nowak, Christopher Cox, and Timothy Rogers. Classification with the sparse group lasso. *IEEE Transactions on Signal Processing*, 64(2):448–463, 2016.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Marcos Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM Journal on Optimization*, 7(1):26–33, 1997.

Garry Robins, Pip Pattison, Yuval Kalish, and Dean Lusher. An introduction to exponential random graph $p*$ models for social networks. *Social networks*, 29(2):173–191, 2007.

Adam J Rothman, Peter J Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

Katya Scheinberg, Shiqian Ma, and Donald Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In *Advances in Neural Information Processing Systems*, pages 2101–2109, 2010.

Defeng Sun, Kim-Chuan Toh, and Liuqin Yang. A convergent 3-block semiproximal alternating direction method of multipliers for conic programming with 4-type constraints. *SIAM journal on Optimization*, 25(2):882–915, 2015.

Kean Ming Tan, Palma London, Karthik Mohan, Su-In Lee, Maryam Fazel, and Daniela M Witten. Learning graphical models with hubs. *Journal of Machine Learning Research*, 15(1):3297–3331, 2014.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Amanda L Traud, Eric D Kelsic, Peter J Mucha, and Mason A Porter. Comparing community structure to characteristics in online collegiate social networks. *SIAM review*, 53(3): 526–543, 2011.

Ruey S Tsay. *Analysis of financial time series*, volume 543. John Wiley & Sons, 2005.

Pedro A Valdés-Sosa, Jose M Sánchez-Bornot, Agustín Lage-Castellanos, Mayrim Vega-Hernández, Jorge Bosch-Bayard, Lester Melie-García, and Erick Canales-Rodríguez. Estimating brain functional connectivity with sparse multivariate autoregression. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 360(1457): 969–981, 2005.

Xiangfeng Wang, Mingyi Hong, Shiqian Ma, and Zhi-Quan Luo. Solving multiple-block separable convex minimization problems using two-block alternating direction method of multipliers. *arXiv preprint arXiv:1308.5294*, 2013.

Lingzhou Xue, Shiqian Ma, and Hui Zou. Positive-definite 1-penalized estimation of large covariance matrices. *Journal of the American Statistical Association*, 107(500):1480–1491, 2012.

Junfeng Yang and Xiaoming Yuan. Linearized augmented Lagrangian and alternating direction methods for nuclear norm minimization. *Mathematics of computation*, 82(281): 301–329, 2013.

Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Changjiu Zhao, Brian Earl Eisinger, Terri M Driessen, and Stephen C Gammie. Addiction and reward-related genes show altered expression in the postpartum nucleus accumbens. *Frontiers in behavioral neuroscience*, 8, 2014.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.