Journal of Machine Learning Research 21 (2020) 1-32 Submitted 3/19; Revised 1/20; Published 3/20

Community-Based Group Graphical Lasso

Eugen Pircalabelu EUGEN.PIRCALABELU@UCLOUVAIN.BE
[1] *UCLouvain, Institute of Statistics, Biostatistics and Actuarial Sciences, Voie du Roman Pays 20, 1348 Louvain-la-Neuve,*
[2] *KU Leuven, ORSTAT and Leuven Statistics Research Center, Naamsestraat 69, 3000 Leuven, Belgium*

Gerda Claeskens GERDA.CLAESKENS@KULEUVEN.BE
KU Leuven, ORSTAT and Leuven Statistics Research Center, Naamsestraat 69, 3000 Leuven, Belgium

Editor: Karsten Borgwardt

Abstract

A new strategy for probabilistic graphical modeling is developed that draws parallels to community detection analysis. The method jointly estimates an undirected graph and homogenous communities of nodes. The structure of the communities is taken into account when estimating the graph and at the same time, the structure of the graph is accounted for when estimating communities of nodes. The procedure uses a joint group graphical lasso approach with community detection-based grouping, such that some groups of edges co-occur in the estimated graph. The grouping structure is unknown and is estimated based on community detection algorithms. Theoretical derivations regarding graph convergence and sparsistency, as well as accuracy of community recovery are included, while the method's empirical performance is illustrated in an fMRI context, as well as with simulated examples.

Keywords: community detection; graphical model; group penalty; joint graphical lasso

1. Introduction

Probabilistic graphical modeling (PGM) summarizes the information coming from multivariate data in a graphical format where nodes, corresponding to features, are linked by edges that indicate dependence relations between the nodes. The objective in PGM is to estimate the structure of the graph (which nodes connect to which other nodes) when data at the nodes are available. The problem can be characterized as a combinatorial problem where the researcher chooses one graph out of many possible graphs as the best performing one.

A graph is estimated based on the multivariate data available at the nodes. The edges between the nodes are estimated using a group penalty, where the grouping is defined based on estimated communities of similar nodes. More concretely, we use a penalized procedure where a group penalty is added to a Gaussian negative log-likelihood and where the group structure (which nodes are similar to each other) is informed by community detection algorithms. We denote throughout the manuscript our procedure as 'ComGGL' which stands for 'community-based group graphical lasso'. The proposed method is illustrated with a resting state (that is, subjects were not performing any tasks) functional magnetic reso-

nance imaging (rsfMRI) example. We analyze here the data for one subject, for which the brain activity for $p = 114$ regions of interest (ROIs) has been measured $n = 240$ times. The data correspond to a subset of the original data analyzed in Schmittmann et al. (2015) and have been kindly provided to us by one of the authors.

In network analysis, the network (graph) is always given or known at the beginning of the modeling process, while in the Gaussian PGM approach one constructs a graph based on an estimated inverse covariance matrix. The graph is thus unknown at the beginning of the modeling process, making the two approaches fundamentally different in spirit. An important part of the literature on networks deals with the estimation of hidden communities of nodes, by which it is meant that certain nodes are linked more often to other nodes which are similar, rather than to dissimilar nodes. This way the nodes form groups or communities of nodes that are more homogenous within the community than between communities where there is a larger degree of heterogeneity. For community detection on networks see Holland et al. (1983), Airoldi et al. (2008, 2013), Rohe et al. (2011), Chen et al. (2012), Amini et al. (2013), Qin and Rohe (2013), Arias-Castro and Verzelen (2014), Cai and Li (2015), Le and Levina (2015), Lei and Rinaldo (2015) and Amini and Levina (2018) among many others. For the estimation of penalized sparse undirected graphs we refer to Friedman et al. (2008), Ravikumar et al. (2008), Bickel and Levina (2008a,b), Boyd et al. (2011), Guo et al. (2011), Witten et al. (2011), Mazumder and Hastie (2012), Danaher et al. (2014) and Pircalabelu et al. (2016) among others.

In the last decade scientists in the field of neuroimaging have been actively investigating data-driven 'brain parcelations' by which it is meant that based on fMRI signals in the brain, one deploys a clustering procedure with the purpose of identifying groups of regions in the brain that act together and are similar enough to form a homogenous block. We refer to Arslan et al. (2018) for a recent systematic and through review on clustering methods applied to fMRI data. To give an example, Yeo et al. (2011) obtained a data-driven parcelation of the brain that contained just seven groups of homogeneous regions which span the entire brain. The rough equivalent of clustering, when one deals with networks is community detection, and so identifying groups of similar nodes in the graph is from an fMRI perspective appealing as it translates into coarse data-driven parcelations of the brain. Moreover, since probabilistic graphs are in spirit different from networks, one cannot simply use the estimated graph as an observed network, as this is not supported by a theoretical argument. To tackle this shortcoming, the main novel contribution of the procedure we develop here is to provide a valid framework that allows the joint estimation of the graph and latent communities of nodes. The new method is theoretically justified and its practical use is showcased via simulations and an fMRI data example.

The structure of the manuscript is as follows. In Section 2 we introduce the proposed model, Section 3 points to similarities and differences between our model and other existing models and in Section 4 we discuss the computational aspects of obtaining joint estimators for the graph and the community structure. In Section 5 we comment on the shortcomings of two-step approaches that first estimate the graph and then consider the estimated graph as an observed network. In Section 6 we investigate graph convergence and graph sparsistency properties. In Section 7 we complement the theoretical analysis by investigating community labeling consistency. In Section 8 we compare the ComGGL method with state-of-the-art two-step approaches in a simulation study, while in Section 9 we illustrate the ComGGL

method on the rsfMRI dataset introduced in Section 1. We finish in Section 10 with a discussion of the method and extensions.

2. Proposed Model and Estimation Method

Let $\boldsymbol{Y} = (Y_1, \ldots, Y_p)^\top \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$ be a random vector having a Gaussian distribution. We associate to each component in $\boldsymbol{Y}$ a node in an undirected graph $G(E, V)$, where $V = \{1, \ldots, p\}$ represents the set of nodes and E is the set of undirected edges of the form $a - b$ between a pair of nodes (a, b). Denote by $\boldsymbol{\Theta}_{a,b}$ the entry on row a and column b from the matrix $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$, where $a \neq b$. Under the Gaussian assumption, Lauritzen (1996, see Proposition 5.2 and Section 5.2) has shown that if $\boldsymbol{\Theta}_{a,b} \neq 0$ then this corresponds to an edge linking nodes a and b in the graph $G(E, V)$. There is a clear link between the concentration matrix $\boldsymbol{\Theta}$ and $G(E, V)$ since the edge set is defined as $E = \{(a, b) \in V \times V | a \neq b \,\&\, \boldsymbol{\Theta}_{a,b} \neq 0\}$. Note that self-loops $a - a$ are not allowed.

Assume further the existence of a sample of n iid vectors, $\boldsymbol{Y}_1, \ldots, \boldsymbol{Y}_n$. Under the stated assumptions, the log-likelihood of the data is proportional to

$$\mathcal{L}(\boldsymbol{\Theta}) = \log\det\boldsymbol{\Theta} - \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Theta}) \tag{1}$$

where $\boldsymbol{S} = (1/n)\sum_{i=1}^{n} \boldsymbol{Y}_i \boldsymbol{Y}_i^\top$ is the empirical covariance matrix. Throughout the manuscript we allow p to depend on n and denote this by p_n.

In the classical setting when $p_n < n$, maximizing (1) with respect to $\boldsymbol{\Theta}$ yields $\boldsymbol{S}^{-1}$ as the maximum likelihood estimator. However, if p_n is close to n or $p_n > n$, the maximum likelihood estimator might be unsatisfactory or ill-defined. For such cases, a new estimator for $\boldsymbol{\Theta}$ is obtained by maximizing a penalized log-likelihood function of the form

$$\mathcal{L}(\boldsymbol{\Theta}) = \log\det\boldsymbol{\Theta} - \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Theta}) - P_\lambda(\boldsymbol{\Theta})$$

under the constraint that $\boldsymbol{\Theta} \succ 0$ (positive definite) and where $P_\lambda(\boldsymbol{\Theta})$ is a suitable penalty function applied to the entries of $\boldsymbol{\Theta}$, which depends on a regularization parameter λ.

In this manuscript we assume further the existence of K_n communities of nodes and denote by $\mathcal{C}_k$ the subset of nodes from $G(E, V)$ that belong to the kth community. By $\#\mathcal{C}_k$ we denote the cardinality of the set $\mathcal{C}_k$. For each component Y_j of $\boldsymbol{Y}$, with $j = 1, \ldots, p_n$ there exists a labeling vector $\boldsymbol{Z}_j = (Z_{j,1}, \ldots, Z_{j,K_n})^\top$ with components either 0 or 1. The role of the vector $\boldsymbol{Z}_j$ is to assign a community (or a label) to each node in the graph. We assume that a node can belong to only one single community. For example, if $\boldsymbol{Z}_1 = (0, 0, 0, 1)^\top$ this implies that the first node in the graph $G(E, V)$ belongs to the fourth community out of a total of four communities of nodes. Hence the vector $\boldsymbol{Z}_j$ contains only a single 1 and all other components are 0. We concatenate all vectors $\boldsymbol{Z}_j$ into a *membership* matrix $\boldsymbol{Z} = [\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_{p_n}]^\top$ of dimension $p_n \times K_n$. For simplicity, we work most often with the matrix $\boldsymbol{Z}\boldsymbol{Z}^\top$ which has the advantage of having the same dimension as $\boldsymbol{\Theta}$ and $\boldsymbol{S}$.

Since we allow for settings where $p_n > n$ and where sparse graphs (that is, many entries $\boldsymbol{\Theta}_{a,b} = 0$) are desired, we consider the penalized negative log-likelihood and seek $\hat{\boldsymbol{\Theta}}$ minimizing $\ell(\boldsymbol{\Theta})$ where

$$\ell(\boldsymbol{\Theta}) = \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Theta}) - \log\det\boldsymbol{\Theta} + P_{\lambda_{n1},\lambda_{n2},\lambda_{n3}}(\boldsymbol{\Theta}), \tag{2}$$

$$P_{\lambda_{n1},\lambda_{n2},\lambda_{n3}}(\boldsymbol{\Theta}) = \underbrace{\lambda_{n1}\sum_{a\neq b}|\boldsymbol{\Theta}_{a,b}|}_{\mathcal{P}_1} + \underbrace{\lambda_{n2}\sum_{k=1}^{K_n}(\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{a,b})^2)^{1/2}}_{\mathcal{P}_2} + \underbrace{\lambda_{n3}\text{tr}(\boldsymbol{Z}\boldsymbol{Z}^\top\boldsymbol{\Theta})}_{\mathcal{P}_3},$$

such that $\boldsymbol{\Theta} \succ 0$. The regularization parameters λ_{n1}, λ_{n2} and λ_{n3} are assumed to be known. By $\boldsymbol{\Theta}^k_{a,b}$ we denote the entry on line a and column b from the matrix $\boldsymbol{\Theta}$, where $a \neq b \in \mathcal{C}_k$.

We detail the purpose of each component of the penalty $P_{\lambda_{n1},\lambda_{n2},\lambda_{n3}}(\boldsymbol{\Theta})$.

$\mathcal{P}_1$ is a classical ℓ_1-penalty that controls the sparsity level of $\boldsymbol{\Theta}$. It shrinks small entries of $\boldsymbol{\Theta}$ to 0, thus enforcing sparsity in $\boldsymbol{\Theta}$ and consequently in $G(E, V)$. The term is controlling the presence of edges between any two nodes irrespective of the community they belong to, with higher values for λ_{n1} forcing sparser estimators.

$\mathcal{P}_2$ is a 'grouping' term as in the spirit of Yuan and Lin (2006) and Danaher et al. (2014) that shrinks together the entries corresponding to a community. It quantifies the effect of the grouping of the nodes on the estimation of the graph as it encourages the entries of the $\boldsymbol{\Theta}$ matrix that correspond to a community to share similarity in terms of magnitude. If the regularization level λ_{n2} becomes large, then the penalty will tend to increase the shrinkage applied to the graph by introducing extra sparsity at the community level.

$\mathcal{P}_3$ links the graph information encoded by the concentration matrix $\boldsymbol{\Theta}$ with the clustering matrix $\boldsymbol{Z}\boldsymbol{Z}^\top$ through a 'trace' operator. The choice for this lies in the fact that the 'trace' part of the objective function can be written as $\text{tr}(\widetilde{\boldsymbol{S}}\boldsymbol{\Theta})$ where $\widetilde{\boldsymbol{S}} = \boldsymbol{S}+\boldsymbol{Z}\boldsymbol{Z}^\top$ which implies that the proposed procedure optimizes a *pseudo* log-likelihood where the total observed sample information consists of a linear combination between the empirical covariance matrix and the community membership matrix. Since $\boldsymbol{Z}\boldsymbol{Z}^\top$ contains the value 1 for *within*-community edges only and 0 everywhere else, in $\widetilde{\boldsymbol{S}}$ only the entries corresponding to within-community edges are perturbed, while all other entries are identical to those in the empirical covariance $\boldsymbol{S}$. In order to balance the contributions from $\boldsymbol{S}$ and $\boldsymbol{Z}\boldsymbol{Z}^\top$ an extra parameter λ_{n3} is introduced. Its role is to act as a 'weight' that balances the two trace quantities as the empirical covariance $\boldsymbol{S}$ might dominate the matrix $\boldsymbol{Z}\boldsymbol{Z}^\top$ which can happen when the entries in the two matrices are not of the same order of magnitude.

In practice, the labeling matrix $\boldsymbol{Z}$ is most often unknown and thus needs to be estimated, which renders the minimization of (2) not directly applicable. Since exact recovery of the membership matrix $\boldsymbol{Z}$ is known to be an NP-hard problem (see Leskovec et al., 2010, and references therein) and since $\boldsymbol{Z}\boldsymbol{Z}^\top$ might not be full rank, most often researchers perform a relaxation for computational reasons. For relaxation on community detection problems, see Cai and Li (2015) and Amini and Levina (2018). As a relaxation we replace $\boldsymbol{Z}\boldsymbol{Z}^\top$ by $\boldsymbol{X}$. The new objective function that we create using $\boldsymbol{X}$ reflects that (i) the estimation of hidden communities is of interest and (ii) the structure of the graph $G(E, V)$ depends on the homogeneity and structure of the subgraphs formed by the latent communities. In order to accomplish this we require that (i) the diagonal elements $\boldsymbol{X}_{a,a} = 1$, corresponding to knowing that a node necessarily belongs to one community, as there are no nodes without a community label, (ii) the off-diagonal elements $\boldsymbol{X}_{a,b} \in [0, 1]$, corresponding to relaxing

a hard 0/1 decision in favor of a 'majority vote' decision and (iii) $\boldsymbol{X}$ should be positive semi-definite.

Following the relaxation approach, we minimize $\ell(\boldsymbol{\Theta}, \boldsymbol{X})$ over $\boldsymbol{\Theta}$ and $\boldsymbol{X}$, where

$$\ell(\boldsymbol{\Theta}, \boldsymbol{X}) = \text{tr}(\boldsymbol{S\Theta}) - \log\det\boldsymbol{\Theta} + \lambda_{n1}\sum_{a\neq b}|\boldsymbol{\Theta}_{a,b}| + \lambda_{n2}\sum_{k=1}^{K_n}(\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}_{a,b}^k)^2)^{1/2} + \lambda_{n3}\text{tr}(\boldsymbol{X\Theta}) \tag{3}$$

such that $\boldsymbol{\Theta} \succ 0$, $\boldsymbol{X} \succeq 0$ (positive semi-definite), $0 \leq \boldsymbol{X}_{a,b} \leq 1$ and $\boldsymbol{X}_{a,a} = 1$, and the membership of nodes to the kth community depends on the matrix $\boldsymbol{X}$.

Equation (3) reflects that we are interested in jointly estimating the concentration matrix $\boldsymbol{\Theta}$ and the relaxed, unknown, labeling matrix $\boldsymbol{X}$. The number of communities K_n can be assumed known or can be estimated using an external procedure, as we do not consider it to be part of the optimization problem. The algorithm we propose to estimate both the graph and the communities uses the procedure of Le and Levina (2015) to estimate K_n at each iteration, but any other method for determining the number of communities can be used. A similar term to $\text{tr}(\boldsymbol{X\Theta})$ from (3) is used as an objective function in Cai and Li (2015) and Amini and Levina (2018) when estimating communities for networks. They directly use an adjacency matrix (or a function of it) of *observed* connections between the nodes, whereas our approach uses the unobserved concentration matrix to perform community detection of nodes on the undirected graphical model.

Alternatively, one can minimize the objective function over $\boldsymbol{\Theta}$ and $\boldsymbol{X}$:

$$\ell(\boldsymbol{\Theta}, \boldsymbol{X}) - \log\det\boldsymbol{X} \tag{4}$$

such that $\boldsymbol{\Theta} \succ 0$, $\boldsymbol{X} \succ 0$ and $0 \leq \boldsymbol{X}_{a,b} \leq 1$, $\boldsymbol{X}_{a,a} = 1$, and the membership of nodes to the kth community depends on the matrix $\boldsymbol{X}$. The term $-\log\det\boldsymbol{X}$ is introduced here for computational simplicity to ensure that $\boldsymbol{X}$ remains positive definite at each iteration of the optimization routine. Details are offered in Section 4.1.

When using (3) we abbreviate the method by ComGGL_1 and when using (4) by ComGGL_2.

To summarize, the objective is to estimate a graph where the grouping structure of the nodes is informed by the underlying communities of similar nodes. There is knowledge that an underlying community structure exists, but this structure is unknown and thus we estimate it. We optimize everything jointly and equations (3) and (4) can be seen as pseudo-loglikelihood functions which bring together the likelihood contribution of the data when estimating a Gaussian graphical model when the grouping structure is unknown and a convex relaxation that allows for community detection using the concentration matrix.

3. A Note on Similarities and Differences to Existing Literature

Tan et al. (2015) have proposed a two-step approach to detect homogeneous communities of nodes in graphical models. They first estimate disjoint clusters of nodes based on a clustering scheme on the empirical covariance matrix and then estimate a probabilistic subgraph using the nodes in each community. The final estimated graph is the union of all such subgraphs and thus there are no edges linking different communities to each other. This is a greedy strategy that divides a large graph estimation problem into many smaller

sized problems and as such gains in time complexity. Moreover, the communities of nodes are treated as a 'nuisance' parameter useful only to determine on which nodes one should estimate a subgraph, rather than as an integral part of the data generating process.

In contrast to Tan et al. (2015), we start from multivariate data observed at the level of the nodes and both the graph and the communities of nodes together with the labeling (which node belongs to which community) need to be estimated. Sometimes one may have expert information regarding the number of communities, or the labels of nodes, but generally it is much harder to have solid information on the topology of the graph, especially if the number of nodes is large. We propose here to *jointly* estimate the undirected Gaussian graph and to identify the latent communities of nodes, where the structure of the community has a *direct* effect on the topology of the estimated graph.

Recently in a time series context, Brownlees et al. (2018) proposed a generalized stochastic block model where the concentration matrix $\boldsymbol{\Theta}$ is a function of the Laplacian of a latent graph. Their approach focuses on obtaining an estimator for $\boldsymbol{\Sigma}$ and uses the empirical covariance matrix to find the communities, whereas our approach uses the unknown concentration matrix. This approach, like ours, uses a spectral decomposition. However, if $p_n \gg n$ then $\boldsymbol{S}$ is not positive definite and if K_n is large, it might create a problem in accurate recovery, especially so since Theorem 2 in their paper uses the sample concentration matrix. Another decisive point of departure is the fact that their approach estimates communities, *without* estimating the latent graph, while ComGGL estimates both quantities of interest.

There is a connection between our approach and the stochastic block model in the work of Lei and Rinaldo (2015) which recovers hidden communities from random adjacency matrices. We recover communities using information from an inverse covariance matrix estimated from random data at the node level. The fundamental difference between our approach and SBMs, lies in the fact that in order to estimate communities of similar nodes we use *estimated* probabilistic Gaussian graphs rather than *observed* networks.

Arroyo Relión et al. (2019) proposed a method based on a group penalty for classification purposes, but the differences with ComGGL are quite substantial. Firstly, those authors assume the graph as *given*, whereas ComGGL starts from data at the nodes and then estimates the graph. Secondly, the grouping structure used in their approach treats all edges connected to a node as forming a group (so in this sense it is fixed and known), whereas ComGGL estimates at each step an underlying unknown grouping structure. Lastly, the objective of Arroyo Relión et al. (2019) is to classify new subjects into a category using network information, whereas our objective is to classify nodes of the graph into a community of similar nodes.

Another connection is with the 'variable clustering' approach of Bunea et al. (2016) of which the G-models are most interesting as they obtain minimax optimal rates of exact partition recovery. The major difference between ComGGL and G-models is the fact that our approach uses the estimator of the true inverse covariance structure $\boldsymbol{\Theta}$ to classify nodes to communities whereas G-models use the empirical correlation matrix (which can perform unsatisfactorily when $p_n \gg n$) to cluster variables similar to the approach of Tan et al. (2015). Moreover, within a block the covariances are assumed equal. This assumption, despite its strictness, has the advantage that the CORD distance, introduced by Bunea et al. (2016), between two variables from the same cluster is always zero. Another difference

is the fact that our approach is a one-step only procedure where we simultaneously estimate the underlying graph as well as the labeling of the nodes.

4. Computational Aspects

Boyd et al. (2011) estimate undirected Gaussian graphical models using a computationally simple algorithm called 'alternating direction method of multipliers' (ADMM), while Cai and Li (2015) and Amini and Levina (2018) use the ADMM algorithm for community detection problems on networks. Here we present an ADMM algorithm that, when converged, outputs $\hat{\boldsymbol{\Theta}}$, an estimated number of communities (unless it is prespecified by the user), as well as the labeling of the nodes.

4.1. Algorithm Implementation

Figure 1 presents a schematic version of the steps in the proposed ADMM algorithm. Starting from data at the node level, the algorithm estimates at each iteration a concentration matrix $\boldsymbol{\Theta}$ which is used to construct an estimated undirected Gaussian graphical model. The estimated $\boldsymbol{\Theta}$ matrix is then used to estimate $\boldsymbol{X}$ and also to determine the number of hidden communities (if it is not fixed upfront by the researcher) on a function of the adjacency matrix (denoted by $\boldsymbol{A}$) as in Le and Levina (2015). Once the number of communities has been detected or specified, spectral clustering techniques are used to determine the community membership and to label the nodes pertaining to the communities. Based on the community membership, the grouping penalty is updated and as a final step we update the concentration matrix $\boldsymbol{\Theta}$. Note that the argument is circular: to estimate the community structure one needs $\boldsymbol{\Theta}$, but in order to estimate $\boldsymbol{\Theta}$ one needs the grouping structure. This shows how $\boldsymbol{\Theta}$ and the community structure depend on one another and are not separated as the structure of the communities is informative for the estimation of $\boldsymbol{\Theta}$.

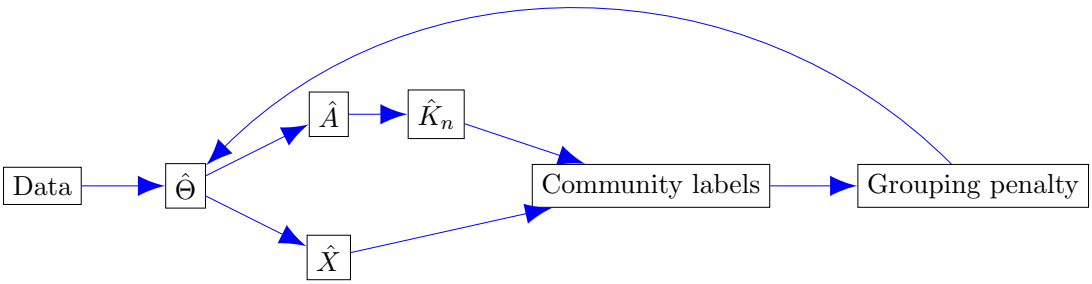


Figure 1: ComGGL procedure flow chart.

For the ADMM algorithm, optimizing (3) is equivalent to minimize over $\boldsymbol{\Theta}, \boldsymbol{X}, \widetilde{\boldsymbol{\Theta}}$ and $\widetilde{\boldsymbol{X}}$

$$\bar{\ell}(\boldsymbol{\Theta}, \boldsymbol{X}, \widetilde{\boldsymbol{\Theta}}, \widetilde{\boldsymbol{X}}) \equiv \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Theta}) - \log\det\boldsymbol{\Theta} + \lambda_{n1}\sum_{a\neq b}|\widetilde{\boldsymbol{\Theta}}_{ab}| + \lambda_{n2}\sum_{k=1}^{K_n}\{\sum_{a\neq b\in\mathcal{C}_k}(\widetilde{\boldsymbol{\Theta}}_{ab}^{k})^2\}^{1/2}$$

$$+\lambda_{n3}\text{tr}(\boldsymbol{X}\boldsymbol{\Theta}) + \tilde{I}(0 \leq \widetilde{\boldsymbol{X}}_{a,b} \leq 1) \tag{5}$$

such that $\boldsymbol{\Theta} \succ 0$, $\boldsymbol{X} \succeq 0$ plus the additional constraints that $\boldsymbol{\Theta} = \widetilde{\boldsymbol{\Theta}}$ and $\boldsymbol{X} = \widetilde{\boldsymbol{X}}$. The function $\tilde{I}(a \in B)$ is the indicator function defined to take the value 0 if $a \in B$ and ∞, otherwise. The diagonal elements of $\widetilde{\boldsymbol{X}}$ are by default set to 1.

Similarly, if $\boldsymbol{X} \succ 0$ is required, one can use

$$\min_{\boldsymbol{\Theta},\boldsymbol{X},\widetilde{\boldsymbol{\Theta}},\widetilde{\boldsymbol{X}}} \{\bar{\ell}(\boldsymbol{\Theta}, \boldsymbol{X}, \widetilde{\boldsymbol{\Theta}}, \widetilde{\boldsymbol{X}}) - \log\det \boldsymbol{X}\} \tag{6}$$

under the remaining constraints of (5).

Based on (5) and (6) we create the augmented Lagrangian functions as

$$\begin{aligned} L_{\text{ComGGL}_1}(\boldsymbol{\Theta}, \widetilde{\boldsymbol{\Theta}}, \widetilde{\boldsymbol{U}}_1, \boldsymbol{X}, \widetilde{\boldsymbol{X}}, \widetilde{\boldsymbol{U}}_2) = \text{tr}(\boldsymbol{S}\boldsymbol{\Theta}) - \log\det\boldsymbol{\Theta} + \lambda_{n1}\sum_{a\neq b}|\widetilde{\boldsymbol{\Theta}}_{a,b}| \\ + \lambda_{n2}\sum_{k=1}^{K_n}(\sum_{a\neq b\in\mathcal{C}_k}(\widetilde{\boldsymbol{\Theta}}^k_{a,b})^2)^{1/2} + \lambda_{n3}\text{tr}(\boldsymbol{X}\boldsymbol{\Theta}) + \tilde{I}(\mathbf{0} \leq \widetilde{\boldsymbol{X}} \leq \mathbf{1}) \\ + \frac{\rho}{2}\|\boldsymbol{\Theta} - \widetilde{\boldsymbol{\Theta}} + \widetilde{\boldsymbol{U}}_1\|_F^2 + \frac{\rho}{2}\|\boldsymbol{X} - \widetilde{\boldsymbol{X}} + \widetilde{\boldsymbol{U}}_2\|_F^2 \end{aligned}$$

and $L_{\text{ComGGL}_2}(\boldsymbol{\Theta}, \widetilde{\boldsymbol{\Theta}}, \widetilde{\boldsymbol{U}}_1, \boldsymbol{X}, \widetilde{\boldsymbol{X}}, \widetilde{\boldsymbol{U}}_2) = L_{\text{ComGGL}_1} - \log\det\boldsymbol{X}$, where $\rho > 0$ is a known constant, $\mathbf{0}$ is a matrix of dimension $p_n \times p_n$ with all elements equal to 0, $\mathbf{1}$ is a matrix of dimension $p_n \times p_n$ with all elements equal to 1, $\widetilde{\boldsymbol{U}}_1$ and $\widetilde{\boldsymbol{U}}_2$ are called 'dual' variables and $\|\cdot\|_F^2$ is the squared Frobenius norm.

We now present a step-by-step description of the algorithm.

Step 1: Initialize $\boldsymbol{\Theta} = \widetilde{\boldsymbol{\Theta}} = \widetilde{\boldsymbol{X}} = \widetilde{\boldsymbol{U}}_1 = \widetilde{\boldsymbol{U}}_2 = \boldsymbol{A} = \boldsymbol{I}$ each having the dimension of $\boldsymbol{S}$ and where $\boldsymbol{I}$ is the identity matrix.

Step 2: At iteration $(m+1)$, specify the number of hidden communities $K_{n(m+1)}$. Alternatively if K_n is not specified one can use the adjacency matrix $\boldsymbol{A}_{(m)}$ (see Step 9 for its definition) to estimate the number of hidden communities $K_{n(m+1)}$ as in Le and Levina (2015) or by other techniques.

Step 3: Update $\boldsymbol{X}_{(m+1)}$. If L_{ComGGL_1} is used, the solution is obtained by

$$\boldsymbol{X}_{(m+1)} = \arg\min_{\boldsymbol{X}}\{\lambda_{n3}\text{tr}(\boldsymbol{X}\boldsymbol{\Theta}_{(m)}) + \frac{\rho}{2}\|\boldsymbol{X} - \widetilde{\boldsymbol{X}}_{(m)} + \widetilde{\boldsymbol{U}}_{2(m)}\|_F^2\}.$$

Setting the gradient with respect to $\boldsymbol{X}$ to 0 yields

$$\lambda_{n3}\boldsymbol{\Theta}_{(m)} + \rho(\boldsymbol{X} - \widetilde{\boldsymbol{X}}_{(m)} + \widetilde{\boldsymbol{U}}_{2(m)}) = 0 \Leftrightarrow \rho\boldsymbol{X} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top,$$

where $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ is the eigen-decomposition of $\rho(\widetilde{\boldsymbol{X}}_{(m)} - \widetilde{\boldsymbol{U}}_{2(m)}) - \lambda_{n3}\boldsymbol{\Theta}_{(m)}$ with $\boldsymbol{\Lambda} = \text{diag}(\Lambda_1, \ldots, \Lambda_{p_n})$ and $\boldsymbol{Q}\boldsymbol{Q}^\top = \boldsymbol{Q}^\top\boldsymbol{Q} = \boldsymbol{I}$.

To ensure that $\boldsymbol{X} \succeq 0$, we create a diagonal matrix $\bar{\boldsymbol{\Lambda}}$ where $\bar{\boldsymbol{\Lambda}}_{a,a} = \Lambda_a$ if $\Lambda_a > 0$ and 0 otherwise. The update is obtained as $\boldsymbol{X}_{(m+1)} = (1/\rho)\boldsymbol{Q}\bar{\boldsymbol{\Lambda}}\boldsymbol{Q}^\top$.

If L_{ComGGL_2} is used, setting the gradient to 0 yields

$$\boldsymbol{\Theta}_{(m)} - \boldsymbol{X}^{-1} + \rho(\boldsymbol{X} - \widetilde{\boldsymbol{X}}_{(m)} + \widetilde{\boldsymbol{U}}_{2(m)}) = 0 \Leftrightarrow \rho\bar{\boldsymbol{X}} - \bar{\boldsymbol{X}}^{-1} = \boldsymbol{\Lambda},$$

where $\bar{\boldsymbol{X}} = \boldsymbol{Q}^\top \boldsymbol{X}\boldsymbol{Q}$. Since $\boldsymbol{\Lambda}$ is diagonal, we have that $\rho\bar{\boldsymbol{X}}_{a,a} + 1/\bar{\boldsymbol{X}}_{a,a} = \Lambda_a$ from which we derive that a solution is $\bar{\boldsymbol{X}}_{a,a} = (\Lambda_a + \sqrt{\Lambda_a^2 + 4\rho})/(2\rho)$ which is always positive since $\rho > 0$. The update is obtained as $\boldsymbol{X}_{(m+1)} = \boldsymbol{Q}\bar{\boldsymbol{X}}\boldsymbol{Q}^\top$ which is always positive definite.

Step 4: Perform approximate spectral K-means clustering to label which nodes belong to each of the K_n communities. The procedure goes as follows: retain first the leading $K_{n(m+1)}$ eigenvectors of $\boldsymbol{X}_{(m+1)}$ and stack them together into an $n \times K_{n(m+1)}$ matrix, apply then to the resulting matrix the K-means procedure to recover a labeling of the nodes.

Step 5: Update $\widetilde{\boldsymbol{X}}_{(m+1)}$ as

$$\widetilde{\boldsymbol{X}}_{(m+1)} = \arg\min_{\widetilde{\boldsymbol{X}}} \{\tilde{I}(\mathbf{0} \leq \widetilde{\boldsymbol{X}} \leq \mathbf{1}) + \frac{\rho}{2}\|\boldsymbol{X}_{(m+1)} - \widetilde{\boldsymbol{X}} + \widetilde{\boldsymbol{U}}_{2(m)}\|_F^2\}.$$

It has been shown in Cai and Li (2015) that

$$\widetilde{\boldsymbol{X}}_{(m+1)} = \min\{\max(\boldsymbol{X}_{(m+1)} + \widetilde{\boldsymbol{U}}_{2(m)}, \mathbf{0}), \mathbf{1}\}.$$

Step 6: Conditional on $\boldsymbol{X}_{(m+1)}$, update $\boldsymbol{\Theta}_{(m+1)}$ as:

$$\boldsymbol{\Theta}_{(m+1)} = \arg\min_{\boldsymbol{\Theta}} \{\text{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \lambda_{n3}\text{tr}(\boldsymbol{X}_{(m+1)}\boldsymbol{\Theta}) - \log\det\boldsymbol{\Theta} + \frac{\rho}{2}\|\boldsymbol{\Theta} - \widetilde{\boldsymbol{\Theta}}_{(m)} + \widetilde{\boldsymbol{U}}_{1(m)}\|_F^2\}.$$

The update is obtained in closed form following the same steps as for $\boldsymbol{X}_{(m+1)}$. Setting the gradient with respect to $\boldsymbol{\Theta}$ to 0 yields:

$$\begin{aligned} \boldsymbol{S} + \lambda_{n3}\boldsymbol{X}_{(m+1)} - \boldsymbol{\Theta}^{-1} + \rho(\boldsymbol{\Theta} - \widetilde{\boldsymbol{\Theta}}_{(m)} + \widetilde{\boldsymbol{U}}_{1(m)}) = 0, & \\ \rho\boldsymbol{\Theta} - \boldsymbol{\Theta}^{-1} \approx \rho(\widetilde{\boldsymbol{\Theta}}_{(m)} - \widetilde{\boldsymbol{U}}_{1(m)}) - \boldsymbol{S}, & \\ \rho\boldsymbol{\Theta} - \boldsymbol{\Theta}^{-1} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top, & \end{aligned}$$

where here $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ denotes the eigen-decomposition of $\rho(\widetilde{\boldsymbol{\Theta}}_{(m)} - \widetilde{\boldsymbol{U}}_{1(m)}) - \boldsymbol{S}$. Using the same reasoning as in Step 3, we have that $\rho\bar{\boldsymbol{\Theta}}_{a,a} + 1/\bar{\boldsymbol{\Theta}}_{a,a} = \Lambda_a$ (where $\bar{\boldsymbol{\Theta}} = \boldsymbol{Q}^\top\boldsymbol{\Theta}\boldsymbol{Q}$) from which we derive that a solution is $\bar{\boldsymbol{\Theta}}_{a,a} = (\Lambda_a + \sqrt{\Lambda_a^2 + 4\rho})/(2\rho)$ and the update takes the form $\boldsymbol{\Theta}_{(m+1)} = \boldsymbol{Q}\bar{\boldsymbol{\Theta}}\boldsymbol{Q}^\top$.

The approximation used when setting the gradient to 0, is used here for numerical stability reasons since the update $\boldsymbol{X}_{(m+1)}$ from Step 3 already uses the term $\boldsymbol{\Theta}_{(m)}$. Since the algorithm has as constraint that $\boldsymbol{\Theta} = \widetilde{\boldsymbol{\Theta}}$ (that is, at each iteration the entries in the two matrices are made more similar to each other), for later iterations the 'information' contained in $\widetilde{\boldsymbol{\Theta}}_{(m)}$ will be used twice when updating $\boldsymbol{\Theta}_{(m+1)}$: once directly through $\widetilde{\boldsymbol{\Theta}}_{(m)}$ and once indirectly through $\boldsymbol{X}_{(m+1)}$.

Step 7: Using the number of communities $K_{n(m+1)}$ and the labeling obtained using $\boldsymbol{X}_{(m+1)}$ in Step 4, update $\widetilde{\boldsymbol{\Theta}}_{(m+1)}$ as:

$$\widetilde{\boldsymbol{\Theta}}_{(m+1)} = \arg\min_{\widetilde{\boldsymbol{\Theta}}} \{\lambda_{n1} \sum_{a \neq b} |\widetilde{\boldsymbol{\Theta}}_{a,b}| + \lambda_{n2} \sum_{k=1}^{K_{n(m+1)}} \{ \sum_{a \neq b \in \mathcal{C}_k} (\widetilde{\boldsymbol{\Theta}}^k_{a,b})^2 \}^{1/2} + \frac{\rho}{2} \|\boldsymbol{\Theta}_{(m+1)} - \widetilde{\boldsymbol{\Theta}} + \widetilde{\boldsymbol{U}}_{1(m)}\|_F^2 \}.$$

Danaher et al. (2014) showed that the solution to the group graphical lasso optimization problem takes the form of an elementwise soft thresholding operation which applied to our problem becomes

$$\begin{aligned} \widetilde{\boldsymbol{\Theta}}_{a,b,(m+1)} &= \text{Soft}_{\lambda_{n1}/\rho}(\boldsymbol{\Theta}_{a,b,(m+1)} + \widetilde{\boldsymbol{U}}_{1,a,b,(m)}) \\ &\quad \times \left(1 - \frac{\lambda_{n2}}{\rho \sum_{k=1}^{K_{n(m+1)}} \sqrt{\sum_{a \neq b \in \mathcal{C}_k} \{\text{Soft}_{\lambda_{n1}/\rho}(\boldsymbol{\Theta}^k_{a,b,(m+1)} + \widetilde{\boldsymbol{U}}^k_{1,a,b,(m)})\}^2}}\right)_+ \end{aligned}$$

where $\text{Soft}_{\lambda_{n1}/\rho}(x) = \text{sgn}(x) \max\{(|x| - \lambda_{n1}/\rho), 0\}$ and $x_+ = \max(x, 0)$.

Step 8: Update $\widetilde{\boldsymbol{U}}_{1(m+1)}$ and $\widetilde{\boldsymbol{U}}_{2(m+1)}$ as

$$\begin{aligned} \widetilde{\boldsymbol{U}}_{1(m+1)} &= \widetilde{\boldsymbol{U}}_{1(m)} + \boldsymbol{\Theta}_{(m+1)} - \widetilde{\boldsymbol{\Theta}}_{(m+1)}, \\ \widetilde{\boldsymbol{U}}_{2(m+1)} &= \widetilde{\boldsymbol{U}}_{2(m)} + \boldsymbol{X}_{(m+1)} - \widetilde{\boldsymbol{X}}_{(m+1)}. \end{aligned}$$

Step 9: Conditional on $\widetilde{\boldsymbol{\Theta}}_{(m+1)}$, update $\boldsymbol{A}_{(m+1)}$ where $\boldsymbol{A}_{ab,(m+1)} = \begin{cases} 1, \text{if } \widetilde{\boldsymbol{\Theta}}_{a,b,(m+1)} \neq 0 \\ 0, \text{otherwise.} \end{cases}$

Step 10: Iterate Steps 2 to 9 until convergence.

The algorithm stops when the difference between the values of the objective function at two consecutive iterations is smaller than a fixed tolerance threshold, which was set throughout all numerical experiments to the value 10^{-4}.

4.2. Algorithm Convergence

The algorithm provided in Section 4.1 relies on profiling both $\boldsymbol{\Theta}$ and $\boldsymbol{X}$ at Steps 3 and 6. After conditioning, the functions are convex in their arguments and due to this biconvexity of the objective function, we can use an ADMM strategy at each of the two steps in the algorithm to obtain a minimizer. The convergence of the algorithm follows from Deng and Yin (2016) which, using a generalized version of the ADMM algorithm, propose optimizing a general constrained convex optimization problem of the form $f(x, y) = g(x) + h(y)$ where g and h are convex functions and where x and y satisfy constraints of the form $Ax + By = b$ for known A, B, and b. See Theorem 2.3 in their paper of which the algorithm in Section 4.1 can be seen as a special case where their matrices P and Q are set to 0 similar to their example in Section 5.2.

The heaviest computational cost is given by the eigen-decompositions needed to ensure the estimators are positive definite which makes the complexity of the algorithm to be of the order $O(p_n^3)$. This is in line with other ℓ_1 based procedures such as those of Friedman et al.

(2008), Rothman et al. (2008), Lam and Fan (2009), Danaher et al. (2014), Pircalabelu et al. (2016), Saegusa and Shojaie (2016) and Molstad and Rothman (2018) among others.

In Figure 6 in Appendix B we show the convergence of the ComGGL algorithms in practice using the rsfMRI example introduced in Section 1.

5. On the Separability Between the Concentration Matrix and the Community Labeling

As we have argued in Section 4.1, there is a feedback loop between $\boldsymbol{\Theta}$ and the grouping structure. The main justification for this is that it is important to know the communities of similar nodes and this information should be taken into account in a *direct way* when estimating the graph, because in an fMRI context the connectivity pattern of the regions within the community is sometimes different than the connectivity pattern between communities.

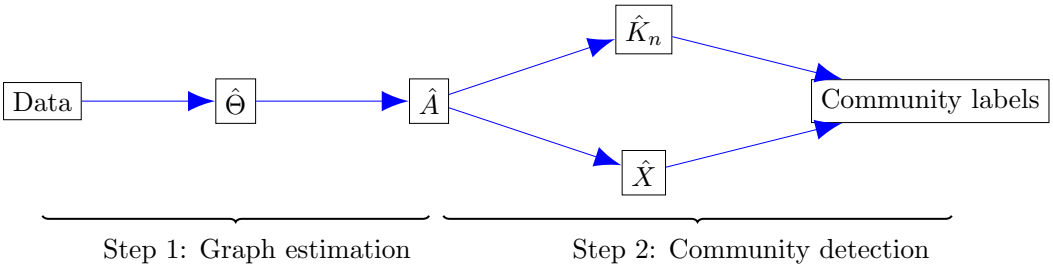


Figure 2: Two-step procedure flow chart.

On the other hand, one might envision a simpler two-step, sequential strategy where the communities are *not* taken into account when estimating the graph. That is, in the first step estimate $\boldsymbol{\Theta}$ (or $\boldsymbol{A}$) and in the second step estimate the community structure based on $\hat{\boldsymbol{\Theta}}$ (or $\hat{\boldsymbol{A}}$). In the literature, the closest application to ours is that of Pavlović (2015) which based on fMRI data, in a classical low dimensional setting where $n > p_n$, uses the empirical correlation matrix for the ROIs and thresholds it using an arbitrary cut-off, to obtain a network where a 0 thresholded value denotes a non-existing edge between two nodes. On the resulting network, they use SBM algorithms to estimate hidden communities. This approach has several limitations: (i) it uses directly the empirical covariance which when $n < p_n$ is not consistent, (ii) the underlying generative process is insensitive to the community structure, in the sense that whether nodes a and b belong to the same cluster or not, makes no difference for $\boldsymbol{\Theta}$ and (iii) the fact that $\hat{\boldsymbol{A}}$ is estimated rather than observed is not taken into account, which makes the SBM assumptions that edges occur based on a Bernoulli model invalid. In this sense, the communities as well as the labeling of nodes are not part of the model and the estimation of $\boldsymbol{A}$ is not accounted for.

Figure 2 presents the flow chart for the two-step, sequential estimation. The major difference with the ComGGL procedure presented in Figure 1 is the fact that the matrix $\boldsymbol{\Theta}$ is not sensitive to the communities. The updating of the concentration matrix based on the communities does not take place and as such the two estimation problems (estimating the graph and estimating the hidden communities) are *completely* separated of one another

and can be done *independently* of one another. The optimization problem can now be set as follows, where Step 2 depends on the results obtained in Step 1,

Step 1: Iterate to obtain

$$\min_{\boldsymbol{\Theta}}\{\text{tr}(\boldsymbol{S\Theta}) - \log\det\boldsymbol{\Theta} + \lambda_{n1}\sum_{a\neq b}|\boldsymbol{\Theta}_{a,b}|\} \text{ such that } \boldsymbol{\Theta} \succ 0. \tag{7}$$

Step 2: Iterate to obtain

$$\max_{\boldsymbol{X}}\{\text{tr}(\hat{\boldsymbol{A}}\boldsymbol{X})\} \text{ such that } \boldsymbol{X} \succeq 0,\ 0 \leq \boldsymbol{X}_{a,b},\ \boldsymbol{X}_{a,a} = 1,\ \boldsymbol{X}\mathbf{1} = (p_n/K_n)\mathbf{1}, \tag{8}$$

where $\hat{\boldsymbol{A}}_{a,b} = 1$ if $\hat{\boldsymbol{\Theta}}_{a,b} \neq 0$ and 0, otherwise; or iterate to obtain

$$\min_{\boldsymbol{X}}\{\text{tr}(\hat{\boldsymbol{E}}\boldsymbol{X})\} \text{ such that } \boldsymbol{X} \succeq 0,\ 0 \leq \boldsymbol{X}_{a,b}, \tag{9}$$

where $\hat{\boldsymbol{E}} = -(\boldsymbol{I}-\boldsymbol{D})^{1/2}\hat{\boldsymbol{A}}(\boldsymbol{I}-\boldsymbol{D})^{1/2} + \boldsymbol{D}^{1/2}(\mathbf{1}-\boldsymbol{I}-\hat{\boldsymbol{A}})\boldsymbol{D}^{1/2}$ and $\boldsymbol{D}$ is a diagonal matrix containing at its diagonal the degrees of the nodes in $\hat{\boldsymbol{A}}$.

In comparison with the optimization problem proposed in Section 2, we observe that

(i) the estimation in a two-step approach can be done in cascade with regular algorithms: Graphical Lasso (Friedman et al., 2008) for optimizing (7) in Step 1 and the sbmSDP procedure proposed in Amini and Levina (2018) for optimizing (8) and the GSBM procedure proposed in Cai and Li (2015) for optimizing (9) in Step 2,

(ii) there is no feedback from the communities to how the graph is estimated and

(iii) $\hat{\boldsymbol{A}}$ is used as an observed adjacency matrix, so no variability in estimating $\hat{\boldsymbol{A}}$ is taken into account.

The novelty of our approach comes from incorporating how the underlying community structure is influencing the estimation of the inverse covariance matrix. If one has knowledge that there are communities of nodes that group together due to functional resemblance and thus tend to be more homogenous and 'communicate' more intensely to the members of the group, the proposed method incorporates this into the modeling step and this is reflected in the estimation of $\boldsymbol{\Theta}$. In the two-step approach this information is ignored.

6. Theoretical Properties

We first introduce notation and discuss the needed technical conditions used throughout.

Denote by

- $\boldsymbol{\Sigma}_0 = (\boldsymbol{\Sigma}_{0,a,b})$: the true covariance matrix;
- $\boldsymbol{\Theta}_0 = (\boldsymbol{\Theta}_{0,a,b})$: the true inverse-covariance matrix;
- $\text{eig}_{\min}(\cdot)$ and $\text{eig}_{\max}(\cdot)$: the smallest and largest eigenvalues of a matrix;

- $\mathcal{S} = \{(a,b)|\ \boldsymbol{\Theta}_{0,a,b} \neq 0\}$: the set of pairs of nodes (a,b) for which the true values in the inverse-covariance matrix are non-zero (these correspond to true edges in the graph);
- $\mathcal{S}^c = \{(a,b)|\ \boldsymbol{\Theta}_{0,a,b} = 0\}$: the set of pairs of nodes (a,b) for which the true values in the inverse-covariance matrix are zero (these correspond to no edges in the graph);
- $s_n = \#\mathcal{S} - p_n$: the number of off-diagonal non-zero elements in the set $\mathcal{S}$ (the number of edges in the true underlying graph also referred to as the 'sparsity' of the graph);
- K_n: the number of hidden communities in the graph;
- $\|\cdot\|$: the spectral norm of a matrix;
- $\mathcal{C}_k = \{a|\ a \in V\ \&\ a$ belongs to the kth community$\}$: the set of nodes in the graph that belong to the kth community;
- $\boldsymbol{Z}_{\mathcal{C}_k *}$: the submatrix of the community membership matrix $\boldsymbol{Z}$ consisting of the rows indexed by the set $\mathcal{C}_k$.

Let $p_n,\ K_n,\ s_n,\ \lambda_{n1}$ and λ_{n2} be sequences that depend on the sample size n as follows: as $n \to \infty$ then $p_n \to \infty$, $s_n \to \infty$, $K_n \to \infty$, $\lambda_{n1} \to 0$ and $\lambda_{n2} \to 0$. For simplicity we assume $\lambda_{n3} = 1$. In words, this implies that as we get more and more cases, the number of nodes in the graph, the number of hidden communities (which depends on p_n) and the number of edges in the graph can grow (but slowly, see condition D), whereas the penalty sequences become less and less important, thus decay towards 0.

General regularity conditions:

(A) There exist constants $\tau_1,\ \tau_2,$ and τ_3 such that:

$$0 < \tau_1 < \text{eig}_{\min}(\boldsymbol{\Sigma}_0) < \text{eig}_{\max}(\boldsymbol{\Sigma}_0) < \tau_2 < \infty;$$

$$0 \leq \text{eig}_{\min}(\boldsymbol{Z}\boldsymbol{Z}^\top) < \text{eig}_{\max}(\boldsymbol{Z}\boldsymbol{Z}^\top) < \tau_3 < \infty;$$

(B) $\max_{(a,b)\in\mathcal{S}}(\mathcal{P}_1'(\boldsymbol{\Theta}_{0,a,b}) + \mathcal{P}_2'(\boldsymbol{\Theta}_{0,a,b})) = O\big(\big(\frac{p_n}{s_n} + 1\big)\big(\frac{\log p_n}{n}\big)^{1/2}\big)$;

(C) $K_n = O(p_n)$;

(D) $\big(\frac{p_n^2}{nK_n} - \frac{p_n}{n}\big) \to 0$, or equivalently, $\frac{p_n}{n}\big(\frac{p_n}{K_n} - 1\big) \to 0$;

(E) There exists a constant $\tau_4 > 0$ such that $\min_{k=1,\ldots,K_n} \min_{(a,b)\in\mathcal{S}} |\boldsymbol{\Theta}^k_{0,a,b}| \geq \tau_4$.

Assumption (A) guarantees that the eigenvalues of the true covariance matrix $\boldsymbol{\Sigma}_0$ and those of the true clustering matrix $\boldsymbol{Z}\boldsymbol{Z}^\top$ are well-behaving. Assumption (B) is a technical condition to get the desired rates. Assumption (C) specifies that the number of hidden communities can be at most of order p_n, we cannot have more communities than nodes; the most extreme case is the one where each node belongs to one community. Assumption (D) links how p_n and K_n can grow with n and it assumes that all communities have roughly the same size. Since each node can belong to only one community, one can permute the rows such that the clustering matrix $\boldsymbol{Z}\boldsymbol{Z}^\top$ can be partitioned as a block diagonal matrix with K_n blocks, where all $(p_n/K_n)^2$ entries in the block are equal to 1 and all other entries are set to 0. By convention in each block the elements on the diagonal (a total of p_n/K_n elements) are also fixed to 1. The most extreme cases when (D) is attained is when either

(i) $p_n/n \to 0$ or (ii) $p_n/K_n \to 1$. Note that (i) is stricter than the usual $\log(p_n)/n$ used with ℓ_1 penalties, implying that ComGGL needs generally a much larger sample than the simpler graphical lasso procedure and (ii) specifies that each node tends to belong to its own community (there is no grouping effect).

Proposition 1 specifies the order of the maximal estimation error of $\boldsymbol{\Theta}$ and $\boldsymbol{X}$ expressed as the Frobenius norm of the difference between the estimators and the true parameters, for a given number of communities K_n, which can grow with the sample size n, and a labeling of nodes. The first term is linked to the estimation of $\boldsymbol{\Theta}$, whereas the second one is coming from the estimation of $\boldsymbol{X}$.

Proposition 1 *Under regularity conditions (A) to (E), for a known sequence K_n, if (i) $\log(p_n)/n = O(\lambda_{n1}^2)$, (ii) $(p_n + s_n)(\log p_n)^{\zeta}/n = O(1)$ for some $\zeta > 1$ and (iii) $\lambda_{n2} = O(\sqrt{\log(p_n)/n})$ then there exist estimators $(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{X}})$ based on the objective function $\ell(\boldsymbol{\Theta}, \boldsymbol{X})$ such that*

$$\max(\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_0\|_F, \|\hat{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{Z}^\top\|_F) = O_p\Big(\max\big\{\sqrt{(p_n + s_n)\frac{\log p_n}{n}}, \sqrt{\frac{p_n^2}{nK_n} - \frac{p_n}{n}}\big\}\Big). \quad (10)$$

Proposition 2 specifies under what conditions the elements that are 0 in the true matrix $\boldsymbol{\Theta}_0$ are with high probability correctly estimated as 0 by the estimator $\hat{\boldsymbol{\Theta}}$. We stress here that sparsistency properties for the ComGGL procedure concern only $\hat{\boldsymbol{\Theta}}$ as due to the imposed relaxations for $\boldsymbol{X}$ in the objective functions (3) and (4), we are not guaranteed that the estimated $\hat{\boldsymbol{X}}$ is also sparse.

Proposition 2 *Under conditions of Proposition 1 for estimators $(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{X}})$ based on the objective function $\ell(\boldsymbol{\Theta}, \boldsymbol{X})$ that satisfy (i) equation (10), (ii) $\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_0\| = O_p(\sqrt{\eta_{n1}})$ and (iii) $\|\hat{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{Z}^\top\| = O_p(\sqrt{\eta_{n2}})$ for sequences $\eta_{n1}, \eta_{n2} \to 0$ if*

$$\sqrt{\frac{\log p_n}{n}} + \sqrt{\eta_{n1}} + \sqrt{\eta_{n2}} + \lambda_{n2}\boldsymbol{\Theta}_{a,b}^k \Big/ \sqrt{\sum_{a \neq b \in \mathcal{C}_k} (\boldsymbol{\Theta}_{a,b}^k)^2} = O(\lambda_{n1}), \quad (11)$$

we have that with probability tending to 1, $\hat{\boldsymbol{\Theta}}_{a,b} = 0$ for all $(a, b) \in \mathcal{S}^c$ from the k-th community.

7. On the Estimation and Consistency of the Community Labeling

Proposition 3 *Let $\boldsymbol{Z}$ be the community membership matrix and let $\boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^\top$ be the eigen-decomposition of $\boldsymbol{Z}\boldsymbol{Z}^\top$. There exists a matrix $\boldsymbol{W}_{K_n \times K_n}$ with real elements such that $\boldsymbol{Q} = \boldsymbol{Z}\boldsymbol{W}$ and where the Euclidean distance between vectors $\boldsymbol{W}_l$ and $\boldsymbol{W}_m$ (which represent the l-th and m-th row of the matrix $\boldsymbol{W}$) is $\|\boldsymbol{W}_l - \boldsymbol{W}_m\| = \{(\#\mathcal{C}_l)^{-1} + (\#\mathcal{C}_m)^{-1}\}^{1/2}$ for all $1 \leq l < m \leq K_n$.*

The proof of Proposition 3 follows from Lemma 2.1 of Lei and Rinaldo (2015) where their connectivity matrix $\boldsymbol{B}$ is replaced by the identity matrix. The proposition establishes that (i) the eigenvectors $\boldsymbol{Q}$ contain information about the community membership matrix

$\boldsymbol{Z}$ and (ii) that one can recover the community structure since two nodes belong to the same community if the rows of the matrix of eigenvectors $\boldsymbol{Q}$ are also the same.

Using Proposition 3 we use spectral clustering with the approximate k-means procedure as in Lei and Rinaldo (2015) and Amini and Levina (2018) to get

$$(\hat{\boldsymbol{Z}}, \hat{\boldsymbol{W}}) = \arg \min_{\boldsymbol{Z} \in \mathbb{M}_{p_n \times K_n}, \boldsymbol{W} \in \mathbb{R}_{K_n \times K_n}} \|\boldsymbol{Z}\boldsymbol{W} - \hat{\boldsymbol{Q}}\|_F^2, \tag{12}$$

where $\hat{\boldsymbol{Q}}\hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{Q}}^\top$ is the K_n-dimensional eigen-decomposition of the matrix $\boldsymbol{X}$ corresponding to the K_n largest absolute eigenvalues and $\mathbb{M}_{p_n \times K_n}$is the set of matrices of dimension $p_n \times K_n$ that have on each row only one value of 1, indicating the community to which the node belongs to, and all other values on the row set at 0, since a node belongs to only one community.

Proposition 3 offers information that using a k-means procedure is an appropriate strategy to identify the hidden communities. It does not however, provide bounds on the relative error for community reconstruction using the k-means procedure.

Proposition 4 similar to Theorem 3.1 of Lei and Rinaldo (2015), quantifies the errors when performing '$(1+\xi)$ k-means' clustering on the rows of $\hat{\boldsymbol{Q}}$ to estimate the community membership. The term '$(1+\xi)$' refers to the fact that there exists a polynomial time algorithm that obtains estimators $(\hat{\boldsymbol{Z}}, \hat{\boldsymbol{W}})$ such that $\|\hat{\boldsymbol{Z}}\hat{\boldsymbol{W}} - \hat{\boldsymbol{Q}}\|_F^2 \leq (1+\xi) \min_{\boldsymbol{Z} \in \mathbb{M}_{p_n \times K_n}, \boldsymbol{W} \in \mathbb{R}_{K_n \times K_n}} \|\boldsymbol{Z}\boldsymbol{W} - \hat{\boldsymbol{Q}}\|_F^2$.

Let S_k denote the sets of misclassified nodes from the kth community. By $\mathcal{C} = \bigcup_{k=1}^{K_n} (\mathcal{C}_k \setminus S_k)$ we denote the set of all nodes correctly classified across all communities and by $\boldsymbol{Z}_{\mathcal{C}*}$ we denote the submatrix of $\boldsymbol{Z}$ formed by retaining only the rows indexed by the set $\mathcal{C}$ of correctly classified nodes and all columns. The errors in Proposition 4 relate to the sizes of the sets of misclassified nodes for each community, $\#S_k$, and specify conditions on the interplay between n, p_n and K_n.

Proposition 4 *Let $\boldsymbol{Z}$ be the community membership matrix and $\hat{\boldsymbol{Z}}$ be the result of the spectral clustering in (12). There exists a constant $c > 0$ such that if $(2+\xi)n^{-1/2}(p_n^2 - K_n p_n)^{1/2} < c$ then with probability tending to 1 there exist subsets $S_k \subset \mathcal{C}_k$ for $k = 1, \ldots, K_n$ and a $K_n \times K_n$ permutation matrix $\boldsymbol{J}$ such that $\hat{\boldsymbol{Z}}_{\mathcal{C}*}\boldsymbol{J} = \boldsymbol{Z}_{\mathcal{C}*}$ where $\sum_{k=1}^{K_n} \#S_k / \#\mathcal{C}_k \leq c^{-1}(2+\xi)n^{-1/2}(p_n^2 - K_n p_n)^{1/2}$.*

8. Simulations

We generated data $\boldsymbol{Y}_i \sim N(\boldsymbol{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ is graph structured and where the sample sizes were $n = 100$ and 1000. More precisely, the graph underlying $\boldsymbol{\Theta}$ contained $K_n = 1, 3$ and 10 communities. The communities contained 20, 50 and 200 nodes with probability $\pi_w = .5$ of nodes being connected to other nodes *within* the community. The probabilities $\pi_b = .1$ and .2 of nodes being connected to nodes from other communities (edges *between* communities) have been used. To evaluate the robustness of the method to model misspecification, we have also generated data from a heavy-tailed multivariate t distribution with 2 degrees of freedom. A number of 48 different simulation settings were created and 100 repetitions per setting were generated.

Figure 3: Visual representation of a random $\boldsymbol{\Theta}$ matrix used in the data generating process. The colored dots indicate non-zero elements, while the white dots indicate elements set at 0. On the main diagonal three communities of nodes are illustrated. The left panel shows the representation when $\pi_b = .1$ and the right panel, the representation when $\pi_b = .2$.

Figure 3 graphically illustrates the structure of $\boldsymbol{\Theta}$. The elements of $\boldsymbol{\Sigma}$ have been generated as follows. First, we created a matrix $\tilde{\boldsymbol{\Sigma}}$ where the value .3 is placed for off-diagonal elements for which the corresponding value in the randomly generated adjacency matrix was 1. Let ϕ be the absolute value of the smallest eigenvalue of $\tilde{\boldsymbol{\Sigma}}$. We then replaced all diagonal elements of $\tilde{\boldsymbol{\Sigma}}$ by ϕ+.2 to ensure positive definiteness, and finally converted $\tilde{\boldsymbol{\Sigma}}$ to a correlation matrix which was further used as $\boldsymbol{\Sigma}$ to generate data. This resulted in values $\boldsymbol{\Sigma}_{a \neq b}$ in the range $-.4$ to .4.

The tested competitors were: ComGGL$_1$, ComGGL$_2$, the sequential estimation using graphical lasso and the sbmSDP, GSBM and CORD and the cluster graphical lasso procedure with single, average and complete linkages. For CORD the estimated covariance matrix $\hat{\boldsymbol{\Sigma}} = \hat{\boldsymbol{\Theta}}^{-1}$ has been converted to a correlation matrix, which was then used as input for the procedure. ComGGL$_1$ enforces positive semi-definiteness of $\boldsymbol{X}$, while ComGGL$_2$ enforces $\boldsymbol{X}$ to be positive definite. For the ComGGL and the sequential sbmSDP and GSBM we use the approach proposed in Le and Levina (2015) for determining an optimal value for K_n; while for cluster graphical lasso and CORD we use the approaches suggested by the corresponding authors. Regularization parameters λ_{n1} and λ_{n2} have been selected using 3-fold cross-validation on a grid of 10×10 values, while to give equal importance to the contribution of $\boldsymbol{S}$ and $\boldsymbol{X}$ in the objective function, we have set $\lambda_{n3} = 1$.

We evaluate all procedures with respect to: (i) the Frobenius norm of the difference between the true $\boldsymbol{\Theta}_0$ and $\hat{\boldsymbol{\Theta}}$ (lower is better), (ii) the F_1 score measuring the accuracy of recovering the edges of the graph (larger is better) and (iii) the Rand index measuring the accuracy of recovering the labelings of the nodes (higher is better). Tables 1–3 present a summary of the obtained results and for all methods and settings averages over 100 simulation runs are presented. More simulation output can be found in the Online Appendix 1.

K	$\#\mathcal{C}_k$	n	ComGGL$_2$	ComGGL$_1$	sbmSDP	GSBM	clustGL:Sin	Avg	Com	CORD
1	20	100	5.41 (0.73)	5.41 (0.73)	5.51 (0.74)	5.51 (0.74)	5.91 (0.79)	6.15 (0.85)	6.32 (0.82)	5.51 (0.74)
1	20	500	4.74 (0.77)	4.74 (0.77)	4.66 (0.77)	4.66 (0.77)	5.42 (0.92)	6.06 (0.88)	6.21 (0.87)	4.66 (0.77)
1	20	1000	4.69 (0.77)	4.69 (0.76)	4.65 (0.77)	4.65 (0.77)	5.45 (0.91)	6.05 (0.88)	6.25 (0.85)	4.65 (0.77)
1	200	100	23.33 (1.45)	23.33 (1.45)	23.34 (1.47)	23.34 (1.47)	23.34 (1.48)	23.59 (1.51)	23.64 (1.51)	23.34 (1.47)
1	200	500	22.37 (1.02)	22.37 (1.02)	22.44 (1.15)	22.44 (1.15)	22.42 (1.11)	22.97 (1.26)	23.18 (1.34)	22.44 (1.15)
1	200	1000	21.73 (1.14)	21.73 (1.14)	21.70 (1.13)	21.70 (1.13)	21.74 (1.13)	22.88 (1.23)	23.17 (1.27)	21.70 (1.13)
3	20	100	8.28 (0.71)	8.28 (0.71)	8.38 (0.75)	8.38 (0.75)	8.44 (0.76)	8.93 (0.85)	9.09 (0.85)	8.38 (0.75)
3	20	500	6.09 (0.60)	6.09 (0.60)	6.00 (0.60)	6.00 (0.60)	6.40 (0.65)	8.19 (0.93)	8.71 (0.83)	6.00 (0.60)
3	20	1000	5.90 (0.61)	5.91 (0.61)	5.84 (0.60)	5.84 (0.60)	6.28 (0.63)	8.08 (0.99)	8.69 (0.84)	5.84 (0.60)
10	20	100	15.11 (0.89)	15.11 (0.89)	15.19 (0.96)	15.19 (0.96)	15.18 (0.96)	15.56 (1.03)	15.70 (1.04)	15.19 (0.96)
10	20	500	12.60 (0.75)	12.59 (0.75)	12.57 (0.76)	12.57 (0.76)	12.58 (0.77)	14.05 (1.09)	14.79 (0.92)	12.57 (0.76)
10	20	1000	10.24 (0.65)	10.24 (0.65)	10.17 (0.65)	10.17 (0.65)	10.34 (0.66)	13.75 (1.24)	14.32 (1.17)	10.17 (0.65)
10	50	100	76.48 (0.63)	76.48 (0.63)	76.51 (0.63)	76.51 (0.63)	/	/	/	76.51 (0.63)
10	50	500	22.79 (1.03)	22.79 (1.03)	22.92 (0.95)	22.92 (0.95)	/	/	/	22.92 (0.95)
10	50	1000	19.99 (0.85)	19.98 (0.85)	19.93 (0.85)	19.93 (0.85)	/	/	/	19.93 (0.85)
1	20	100	8.13 (0.80)	8.13 (0.80)	8.08 (0.81)	8.08 (0.81)	8.26 (0.83)	8.47 (0.90)	8.57 (0.86)	8.08 (0.81)
1	20	500	8.73 (0.78)	8.73 (0.78)	8.70 (0.78)	8.70 (0.78)	8.85 (0.81)	9.02 (0.85)	9.12 (0.85)	8.70 (0.78)
1	20	1000	8.77 (0.77)	8.77 (0.78)	8.73 (0.77)	8.73 (0.77)	8.87 (0.79)	9.01 (0.82)	9.11 (0.81)	8.73 (0.77)
1	200	100	26.30 (1.55)	26.30 (1.55)	26.26 (1.54)	26.26 (1.54)	26.29 (1.54)	26.68 (1.55)	27.05 (1.63)	26.26 (1.54)
1	200	500	28.63 (1.57)	28.63 (1.57)	28.60 (1.57)	28.60 (1.57)	28.62 (1.57)	28.93 (1.59)	29.17 (1.61)	28.60 (1.57)
1	200	1000	29.09 (1.63)	29.10 (1.63)	29.06 (1.62)	29.06 (1.62)	29.08 (1.62)	29.42 (1.61)	29.66 (1.69)	29.06 (1.62)
3	20	100	11.72 (0.83)	11.73 (0.83)	11.67 (0.82)	11.67 (0.82)	11.75 (0.84)	12.37 (0.95)	12.64 (0.96)	11.67 (0.82)
3	20	500	13.02 (0.83)	13.02 (0.83)	12.97 (0.83)	12.97 (0.83)	13.03 (0.84)	13.37 (0.90)	13.57 (0.90)	12.97 (0.83)
3	20	1000	13.26 (0.81)	13.26 (0.81)	13.22 (0.82)	13.22 (0.82)	13.27 (0.82)	13.62 (0.87)	13.74 (0.86)	13.22 (0.82)
10	20	100	19.38 (1.21)	19.38 (1.21)	19.35 (1.21)	19.35 (1.21)	19.39 (1.22)	20.03 (1.43)	20.52 (1.61)	19.35 (1.21)
10	20	500	22.41 (1.17)	22.41 (1.17)	22.38 (1.17)	22.38 (1.17)	22.40 (1.17)	22.86 (1.28)	23.16 (1.24)	22.38 (1.17)
10	20	1000	23.18 (1.11)	23.18 (1.11)	23.13 (1.12)	23.13 (1.12)	23.15 (1.12)	23.62 (1.22)	23.87 (1.23)	23.13 (1.12)
10	50	100	31.95 (6.67)	31.95 (6.67)	31.48 (4.92)	31.48 (4.92)	/	/	/	31.48 (4.92)
10	50	500	36.40 (1.36)	36.41 (1.36)	36.37 (1.35)	36.37 (1.35)	/	/	/	36.37 (1.35)
10	50	1000	37.68 (1.44)	37.68 (1.44)	37.66 (1.44)	37.66 (1.44)	/	/	/	37.66 (1.44)

Table 1: Simulated data. Average and standard deviation of the Frobenius norm (smaller is better), when all competitors use an estimated K_n value and 3-fold CV is used to select the optimal tuning parameters, and $\pi_b = .1$. The symbol '/' denotes that the method has been omitted from the calculation due to computational complexity. Top part: Gaussian data; lower part: Student t_2.

K	$\#\mathcal{C}_k$	n	ComGGL$_2$	ComGGL$_1$	sbmSDP	GSBM	clustGL$_{\text{Sin}}$	clustGL$_{\text{Avg}}$	clustGL$_{\text{Com}}$	CORD
1	20	100	0.68 (0.07)	0.68 (0.07)	0.65 (0.08)	0.65 (0.08)	0.50 (0.09)	0.39 (0.11)	0.31 (0.08)	0.65 (0.08)
1	20	500	0.82 (0.02)	0.82 (0.02)	0.83 (0.02)	0.83 (0.02)	0.64 (0.08)	0.43 (0.11)	0.37 (0.11)	0.83 (0.02)
1	20	1000	0.84 (0.02)	0.84 (0.02)	0.84 (0.02)	0.84 (0.02)	0.65 (0.07)	0.45 (0.10)	0.36 (0.08)	0.84 (0.02)
1	200	100	0.13 (0.03)	0.13 (0.03)	0.13 (0.03)	0.13 (0.03)	0.13 (0.03)	0.07 (0.02)	0.06 (0.02)	0.13 (0.03)
1	200	500	0.37 (0.06)	0.37 (0.06)	0.36 (0.04)	0.36 (0.04)	0.36 (0.05)	0.25 (0.08)	0.21 (0.04)	0.36 (0.04)
1	200	1000	0.50 (0.00)	0.50 (0.00)	0.50 (0.00)	0.50 (0.00)	0.49 (0.01)	0.29 (0.06)	0.23 (0.04)	0.50 (0.00)
3	20	100	0.45 (0.03)	0.45 (0.03)	0.43 (0.04)	0.43 (0.04)	0.41 (0.05)	0.27 (0.06)	0.21 (0.06)	0.43 (0.04)
3	20	500	0.66 (0.01)	0.66 (0.01)	0.66 (0.01)	0.66 (0.01)	0.63 (0.02)	0.41 (0.08)	0.30 (0.04)	0.66 (0.01)
3	20	1000	0.71 (0.01)	0.71 (0.01)	0.71 (0.01)	0.71 (0.01)	0.67 (0.02)	0.44 (0.09)	0.31 (0.03)	0.71 (0.01)
10	20	100	0.22 (0.04)	0.22 (0.04)	0.21 (0.04)	0.21 (0.04)	0.21 (0.04)	0.12 (0.03)	0.09 (0.03)	0.21 (0.04)
10	20	500	0.55 (0.01)	0.55 (0.01)	0.55 (0.01)	0.55 (0.01)	0.54 (0.01)	0.34 (0.06)	0.25 (0.02)	0.55 (0.01)
10	20	1000	0.62 (0.01)	0.62 (0.01)	0.62 (0.01)	0.62 (0.01)	0.61 (0.01)	0.38 (0.07)	0.28 (0.05)	0.62 (0.01)
10	50	100	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	/	/	/	0.32 (0.00)
10	50	500	0.41 (0.01)	0.41 (0.01)	0.40 (0.03)	0.40 (0.03)	/	/	/	0.40 (0.03)
10	50	1000	0.52 (0.00)	0.52 (0.00)	0.52 (0.00)	0.52 (0.00)	/	/	/	0.52 (0.00)
1	20	100	0.67 (0.05)	0.66 (0.05)	0.68 (0.05)	0.68 (0.05)	0.58 (0.09)	0.44 (0.15)	0.37 (0.13)	0.68 (0.05)
1	20	500	0.74 (0.04)	0.71 (0.05)	0.74 (0.04)	0.74 (0.04)	0.63 (0.07)	0.45 (0.14)	0.35 (0.12)	0.74 (0.04)
1	20	1000	0.75 (0.04)	0.72 (0.05)	0.76 (0.04)	0.76 (0.04)	0.63 (0.08)	0.47 (0.15)	0.35 (0.11)	0.76 (0.04)
1	200	100	0.35 (0.03)	0.35 (0.03)	0.35 (0.03)	0.35 (0.03)	0.35 (0.03)	0.31 (0.06)	0.27 (0.08)	0.35 (0.03)
1	200	500	0.42 (0.03)	0.41 (0.03)	0.42 (0.03)	0.42 (0.03)	0.42 (0.03)	0.35 (0.07)	0.30 (0.09)	0.42 (0.03)
1	200	1000	0.44 (0.03)	0.43 (0.03)	0.45 (0.03)	0.45 (0.03)	0.44 (0.03)	0.35 (0.08)	0.29 (0.09)	0.45 (0.03)
3	20	100	0.45 (0.02)	0.45 (0.02)	0.46 (0.02)	0.46 (0.02)	0.44 (0.02)	0.33 (0.07)	0.27 (0.08)	0.46(0.02)
3	20	500	0.52 (0.02)	0.52 (0.02)	0.52 (0.02)	0.52 (0.02)	0.51 (0.02)	0.39 (0.08)	0.31 (0.09)	0.52(0.02)
3	20	1000	0.54 (0.02)	0.54 (0.02)	0.55 (0.02)	0.55 (0.02)	0.53 (0.02)	0.38 (0.08)	0.31 (0.08)	0.55(0.02)
10	20	100	0.31 (0.01)	0.31 (0.01)	0.31 (0.01)	0.31 (0.01)	0.31 (0.01)	0.27 (0.04)	0.24 (0.05)	0.31 (0.01)
10	20	500	0.40 (0.01)	0.40 (0.01)	0.40 (0.01)	0.40 (0.01)	0.39 (0.01)	0.32 (0.06)	0.28 (0.08)	0.40 (0.01)
10	20	1000	0.43 (0.01)	0.43 (0.01)	0.43 (0.01)	0.43 (0.01)	0.43 (0.01)	0.34 (0.07)	0.29 (0.08)	0.43 (0.01)
10	50	100	0.22 (0.02)	0.22 (0.02)	0.22 (0.02)	0.22 (0.02)	/	/	/	0.22 (0.02)
10	50	500	0.31 (0.01)	0.31 (0.01)	0.31 (0.01)	0.31 (0.01)	/	/	/	0.31 (0.01)
10	50	1000	0.34 (0.01)	0.34 (0.01)	0.34 (0.01)	0.34 (0.01)	/	/	/	0.34 (0.01)

Table 2: Simulated data. Average and standard deviation of the F_1 score (larger is better), when all competitors use an estimated K_n value and 3-fold CV is used to select the optimal tuning parameters, and $\pi_b = .1$. The symbol '/' denotes that the method has been omitted from the calculation due to computational complexity. Top part: Gaussian data; lower part: Student t_2.

K	$\#\mathcal{C}_k$	n	ComGGL$_2$	ComGGL$_1$	sbmSDP	GSBM	clustGL$_{\text{Sin}}$	clustGL$_{\text{Avg}}$	clustGL$_{\text{Com}}$	CORD
1	20	100	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.64 (0.14)	0.37 (0.18)	0.26 (0.11)	0.94 (0.12)
1	20	500	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.66 (0.13)	0.35 (0.16)	0.28 (0.18)	0.96 (0.10)
1	20	1000	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.64 (0.12)	0.35 (0.13)	0.25 (0.11)	0.97 (0.09)
1	200	100	0.51 (0.13)	0.28 (0.07)	0.27 (0.06)	0.29 (0.07)	0.97 (0.02)	0.35 (0.12)	0.23 (0.04)	0.99 (0.02)
1	200	500	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.97 (0.01)	0.38 (0.12)	0.25 (0.09)	0.78 (0.18)
1	200	1000	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.97 (0.02)	0.37 (0.10)	0.25 (0.09)	0.65 (0.17)
3	20	100	0.32 (0.00)	0.32 (0.00)	0.34 (0.06)	0.34 (0.06)	0.37 (0.02)	0.54 (0.05)	0.59 (0.05)	0.33 (0.02)
3	20	500	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	0.37 (0.01)	0.54 (0.05)	0.59 (0.02)	0.38 (0.05)
3	20	1000	0.33 (0.04)	0.33 (0.04)	0.36 (0.11)	0.36 (0.12)	0.37 (0.01)	0.54 (0.05)	0.59 (0.02)	0.37 (0.05)
10	20	100	0.43 (0.10)	0.68 (0.05)	0.69 (0.06)	0.67 (0.08)	0.12 (0.01)	0.61 (0.08)	0.72 (0.02)	0.10 (0.01)
10	20	500	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.12 (0.01)	0.59 (0.11)	0.72 (0.02)	0.22 (0.11)
10	20	1000	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.12 (0.01)	0.59 (0.10)	0.70 (0.07)	0.33 (0.14)
10	50	100	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	/	/	/	0.90 (0.00)
10	50	500	0.10 (0.00)	0.10 (0.00)	0.17 (0.19)	0.16 (0.19)	/	/	/	0.34 (0.14)
10	50	1000	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	/	/	/	0.49 (0.12)
1	20	100	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.71 (0.17)	0.45 (0.27)	0.31 (0.19)	0.53 (0.20)
1	20	500	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.70 (0.15)	0.41 (0.21)	0.27 (0.16)	0.59 (0.22)
1	20	1000	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.68 (0.16)	0.44 (0.27)	0.26 (0.18)	0.62 (0.22)
1	200	100	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.98 (0.02)	0.76 (0.26)	0.56 (0.31)	0.05 (0.05)
1	200	500	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.98 (0.02)	0.69 (0.25)	0.50 (0.30)	0.06 (0.05)
1	200	1000	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	1.00 (0.00)	0.98 (0.02)	0.60 (0.26)	0.41 (0.26)	0.08 (0.06)
3	20	100	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	0.35 (0.02)	0.50 (0.08)	0.56 (0.09)	0.59 (0.05)
3	20	500	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	0.35 (0.02)	0.49 (0.09)	0.56 (0.09)	0.56 (0.06)
3	20	1000	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	0.32 (0.00)	0.35 (0.02)	0.52 (0.08)	0.56 (0.07)	0.54 (0.07)
10	20	100	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.11 (0.01)	0.33 (0.21)	0.47 (0.23)	0.87 (0.03)
10	20	500	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.11 (0.01)	0.38 (0.21)	0.50 (0.24)	0.85 (0.05)
10	20	1000	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.11 (0.01)	0.41 (0.20)	0.53 (0.23)	0.84 (0.05)
10	50	100	0.33 (0.25)	0.33 (0.25)	0.32 (0.26)	0.32 (0.25)	/	/	/	0.89 (0.03)
10	50	500	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	/	/	/	0.89 (0.08)
10	50	1000	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	0.10 (0.00)	/	/	/	0.89 (0.02)

Table 3: Simulated data. Average and standard deviation of the Rand index (larger is better), when all competitors use an estimated K_n value and 3-fold CV is used to select the optimal tuning parameters, and $\pi_b = .1$. The symbol '/' denotes that the method has been omitted from the calculation due to computational complexity. Top part: Gaussian data; lower part: Student t_2.

Upon inspection, the tables reveal that:

(i) increasing the sample size from 100 to 500 and 1000 is beneficial in reducing the Frobenius norm of the difference between the true $\boldsymbol{\Theta}_0$ and $\hat{\boldsymbol{\Theta}}$ and the F_1 score measuring the accuracy of graph recovery for all methods;

(ii) quite consistenly across all tested scenarios the clusterGL with the average or complete linkage performed unsatisfactory with respect to the Frobenius norm and F_1 score; moreover due to the computational complexity of selecting the number of components, for larger graphs estimation was prohibitive;

(iii) across all tested scenarios if the data generating process is misspecified (that is, using a Student t distribution with 2 degrees of freedom) it leads to a similar deterioration of performance for all methods;

(iv) not knowing the true value of K_n has a larger impact on the accuracy of correctly assigning the nodes to their respective communities for ComGGL, sbmSDP and GSBM (which all use the same procedure for determining K_n) than for clusterGL (which especially with the average or complete linkage performed very well with respect to the criterion). This points to the crucial need of having a precise idea about the number of communities, or alternatively an accurate estimate for it;

(v) allowing all nodes to belong to one single community of nodes does not incur any sensible performance loss compared to the competitors;

When comparing the techniques among themselves, the ComGGL procedures, CORD and the two-step sbmSDP and GSBM procedures are the best performing competitors with respect to the proposed performance measures. The clusterGL procedure presents an interesting phenomenon where using the single linkage provided satisfactory Frobenius norm and F_1 performance, but unsatisfactory labeling recovery, whereas using the complete linkage provided good labeling recovery but unsatisfactory Frobenius norm and F_1 performance, regardless of K_n being known or not.

It is worth mentioning that none of the competitors is everywhere the best performing one, however the proposed ComGGL procedures seem to consistently be close to the best performing techniques. This is especially so with respect to graph recovery, pointing to the fact that taking into account the existence of communities of nodes when estimating the graph is a valid strategy that can improve the accuracy of estimating the graph and the concentration matrix. Moreover, under both a positive definiteness constraint and positive semi-definiteness constraint on $\boldsymbol{X}$, ComGGL provided similar performance.

From our experiments, we have observed that the method is relatively robust to the value of λ_{n3} even when $\boldsymbol{\Theta}$ dominates $\boldsymbol{Z}\boldsymbol{Z}^\top$ (by which we mean that the largest eigenvalues of the two matrices are on different scales), although a more precise tuning of this parameter can sometimes result in moderate improvements in performance. If the procedure is given knowledge about an optimal number of communities (see Online Appendix 2), it also performs accurate labeling, in line with other competitors. This points to the fact that estimating communities using concentration matrix information rather than binary adjacency matrices is also a competitive strategy. The added advantage of ComGGL is that it performs a one-step estimation of all quantities, whereas the two-step adjacency-based approaches cannot theoretically justify the extra Bernoulli assumption on the edges of the estimated graph which is used as input for community detection, although in the simulated experiment it seemed to work.

Figure 4 shows the average running time of each competitor for different sizes of the graphs. It suggests that (i) for fixed values of the tunning parameters, the proposed method is more time consuming than the two-step approaches, but comparable or slightly faster that the cluster graphical lasso (the search for an optimal K_n is the bottleneck of this competitor) and (ii) the computational efficiency suggested by (4) translates into faster runtimes for ComGGL_2 compared to ComGGL_1.

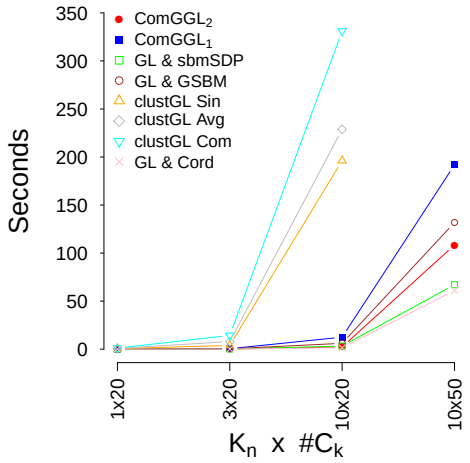


Figure 4: Simulated data. Average runtime in seconds (smaller is better) when the data are generated from the Gaussian model. For all settings we fix $\pi_b = .1$ and $\lambda_{n3} = 1$. The symbols represent averages over all different runs for different configuration of K_n and $\#\mathcal{C}_k$. The points are connected to facilitate comparison.

9. Application to rsfMRI Data

We retake the rsfMRI example introduced in Section 1. The dataset contains 114 ROIs (columns) and 240 BOLD measurements (rows). Each ROI is associated with a node in the graph and the purpose is to jointly estimate (i) the brain pathways (which ROIs connect with other ROIs) as well as (ii) the hidden community structure of the nodes (which nodes are similar enough to each other to form a homogenous cluster of brain ROIs).

We evaluate several procedures with respect to the accuracy of recovering the six lobes that span the entire brain, namely Frontal, Parietal, Temporal, Occipital, Insula and Cingulate. The ROIs are positioned in only one of the lobes. An overlap between the estimated community membership and the known lobe membership reveals how well the lobe structure of the brain is captured by the community structure. The procedures we evaluate are ComGGL$_1$, ComGGL$_2$, two-step sequential estimation using simple graphical lasso (no grouping) and the sbmSDP procedure of Amini and Levina (2018), the GSBM procedure of Cai and Li (2015) and the CORD procedure of Bunea et al. (2016). We add to the list also the cluster graphical lasso procedure of Tan et al. (2015) with single, average and complete linkages. The two-step procedures use first the graphical lasso algorithm (which is indifferent to the grouping structure of the nodes) to estimate the graph and in the second step, use community detection procedures on the estimated graph to detect clusters of similar nodes. The regularization levels were $(\lambda_{n1}, \lambda_{n2}, \lambda_{n3}) = (.4, .2, 1)$ for ComGGL$_1$ and ComGGL$_2$, $\lambda_{n1} = 0.395$ for the two-step procedures that use the output of the graphical lasso. For these regularization parameters, all estimated graphs had a sparsity coefficient of roughly 92%.

Table 4 presents the performance of the methods with respect to the accuracy of recovering the six lobes. We measure accuracy of lobe recovery using the (adjusted) Rand index and the F_1 index defined as

$$\text{Rand} = \frac{yy + nn}{N_T},$$
$$F_1 = \frac{2PR}{P+R} \text{ where } P = yy/(yy+ny); R = yy/(yy+yn),$$

				GL			clusterGL		
	K_n	ComGGL$_1$	ComGGL$_2$	sbmSDP	GSBM	CORD	Single	Average	Complete
F_1	4	.40	**.41**	**.41**	.39	/	.37	.32	.25
	5	.42	.37	**.44**	.40	/	.37	.26	.23
	6	**.39**	.38	.37	.37	/	.36	.25	.20
	7	**.50**	.38	.38	.37	/	.36	.22	.20
	8	**.47**	.34	.36	.39	/	.35	.19	.18
	9	**.42**	.30	.35	.37	.38	.36	.18	.17
	10	.35	.32	.32	**.36**	/	.35	.17	.16
	11	**.40**	.32	.29	.36	/	.35	.17	.14
	12	**.37**	.25	.27	**.37**	/	.35	.17	.13
Rand Index	4	**.57**	.42	**.57**	.52	/	.50	.40	.25
	5	**.37**	.31	.35	.33	/	.25	.14	.12
	6	.19	**.23**	.21	.21	/	.00	.01	.01
	7	**.36**	.24	.25	.23	/	.00	.01	.02
	8	**.30**	.19	.24	.27	/	.01	.02	.02
	9	**.25**	.16	.24	**.25**	.17	.00	.01	.02
	10	.16	.16	.22	**.24**	/	.01	.01	.02
	11	**.27**	.15	.19	.25	/	.01	.01	.01
	12	.20	.08	.17	**.27**	/	.01	.01	.01

Table 4: rsfMRI data. F_1 label accuracy index (larger is better; top panel) and Rand Index (larger is better; bottom panel) for various values of K_n ranging from 4 to 12. Largest values per K_n are presented in bold.

and where

- yy represents the number of pairs of regions that are assigned to the same community and at the same time belong to the same lobe;
- yn represents the number of pairs of regions that are assigned to the same community, but in reality belong to different lobes;
- ny represents the number of pairs of regions that are assigned to two different communities, but in reality belong to the same lobe;
- nn represents the number of pairs of regions that are assigned to two different communities and at the same time belong to different lobes;
- N_T represents the total number of pairs of regions.

For all tested competitors but CORD, one can fix the number of desired communities and the purpose of the analysis in Table 4 is to perform a sensitivity analysis when the number of communities is varied between 4 and 12. The main conclusion is that ComGGL$_1$ offers best performance in recovering the lobe partition of the brain among all competitors, followed closely by the two-step procedures and ComGGL$_2$. The performance of clusterGL seems to deteriorate severely when $K_n > 4$, so we dropped it in the sequential analysis.

Next, we evaluate the procedures with respect to the homogeneity of the six lobes, where the homogeneity score (higher is better) of a lobe is defined as:

$$\text{Homogeneity}(\text{Lobe}_j) = \frac{\#\text{ROIs} \in \text{Lobe}_j}{\#\text{communities estimated for all ROIs} \in \text{Lobe}_j}.$$

Lobe	ROIs	ComGGL$_1$	ComGGL$_2$	GL		
				sbmSDP	GSBM	CORD
Cingulate	8	2.00	2.00	**2.67**	**2.67**	**2.67**
Frontal	39	**9.75**	**9.75**	7.80	7.80	**9.75**
Insula	4	**4.00**	**4.00**	**4.00**	**4.00**	**4.00**
Occipital	13	**13.0**	**13.0**	6.50	6.50	4.33
Parietal	24	**6.00**	**6.00**	4.80	4.00	4.80
Temporal	26	8.67	5.20	5.20	6.50	**13.0**
Average over lobes	/	**7.24**	6.66	5.16	5.24	6.42
Average within LH over lobes		**4.56**	3.27	3.32	3.16	3.31
Average within RH over lobes		**3.79**	3.54	2.80	2.85	3.30

Table 5: rsfMRI data. Homogeneity index (larger is better) for the six brain lobes. Largest values per lobe are presented in bold. The number of ROIs per lobe is also presented.

To give a concrete example, we know that the Occipital lobe contains 13 ROIs out of the 114 ROIs under study. If all the nodes are estimated as being part of a single community, then Homogeneity(Occipital)=13. If some of the 13 nodes are estimated to be in one community and the rest into another community then Homogeneity(Occipital)=6.5, since two communities are estimated for these nodes.

Table 5 presents the obtained results when all methods are allowed to estimate K_n. All procedures identify correctly the insula lobe into one single community, but the ComGGL procedures also correctly identify the occipital lobe. The CORD method did not satisfactorily identify the occipital lobe as one single community, but identified the temporal lobe well. The sbmSDP and GSBM, compared to the other procedures, are only marginally better in reconstructing the cingulate lobe. Inspecting the right and left hemisphere separately, we can conclude that on both hemispheres ComGGL$_1$ provided communities that are closer to the lobe membership. In the left hemisphere all the other techniques provided similar homogeneity scores, but for the regions on the right hemisphere ComGGL$_1$, the sbmSDP and GSBM procedures estimated that the nodes are less homogeneous. ComGGL$_2$ estimates the right hemisphere as being slightly more homogenous than the left one, whereas the CORD procedure estimates both hemispheres as having roughly the same homogeneity score. Overall, the ComGGLs estimated communities that come closer to the lobe repartition, followed by the CORD procedure.

Figure 5 shows the estimated graphs when applying ComGGL$_1$ (panel a), ComGGL$_2$ (panel b), the two-step sequential estimation using simple graphical lasso and sbmSDP (panel c), GSBM (panel d) and the CORD procedure (panel e). The bottom panels display the estimated community labels for each of the competitors.

The procedures estimate between seven and nine communities of nodes and output similar graph and community configurations, however certain differences can be illustrated, especially with respect to the number of ROIs included in each community, as well as to the structure of the communities in terms of included regions. All estimated configurations agree on the identification of the community, labeled '2', that groups ROIs across the hemispheres, while most procedures agree on the identification of two communities, labeled '1' and '5' in the frontal part of the brain and two communities, '6' and '7' that span the posterior part. However similar the community structure across the solutions might be, differences can also be observed, most predominantly for ComGGL$_2$ that estimates the 5th and 7th communities consisting of single ROIs and CORD that prefers more fragmented communities with a few number of large components accompanied by a

	F_1				Rand Index			
	ComGGL$_2$	sbmSDP	GSBM	CORD	ComGGL$_2$	sbmSDP	GSBM	CORD
ComGGL$_1$	.81	.86	.86	.71	.74	.79	.80	.62
ComGGL$_2$		.84	.83	.69		.77	.77	.61
sbmSDP			.94	.71			.91	.62
GSBM				.71				.61

Table 6: rsfMRI data. Community structure agreement measured by the F_1 score (left panel) and Rand index (right panel). Larger values denote larger agreement between methods.

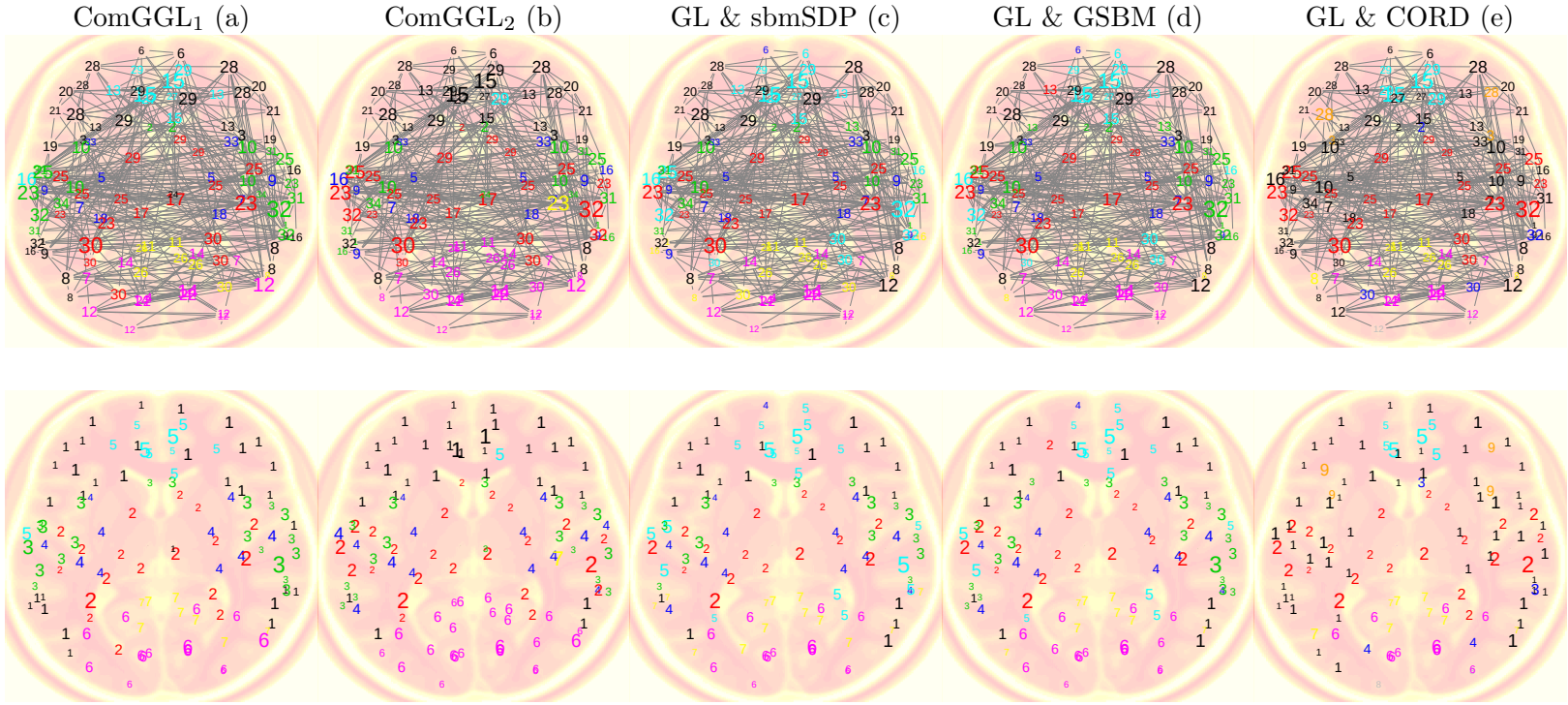
larger number of small communities. Table 6 quantifies using the F_1 and Rand index to what extend the five solutions agree on the latent community structure and it supports that visual inspection that ComGGL$_1$, sbmSDP and GSBM provide quite similar communities in terms of identified ROIs that form a community, number and size of the communities.

10. Discussion

We have introduced a new method that estimates an undirected graphical model and at the same time performs community detection of similar nodes. Our procedure takes the estimated communities into account when estimating the underlying concentration matrix. The application of the method to fMRI data shows a good performance and reveals (i) a clear functional separation between the communities of brain regions as well as (ii) homogenous communities. On simulated data the ComGGL procedure provided similar results to state-of-the-art two-step procedures. For future developments, one might offer the method more freedom in labeling the nodes by allowing certain hub nodes to belong to multiple clusters as in the overlapping clusters framework of Bing et al. (2020). Another possible extension is towards conditional graphical models as in Yin and Li (2011) where one estimates a graph conditional on external information at the level of the nodes. An interesting idea would be to quantify to what extent both the graph and the communities depend on external information at the node level.

Acknowledegements

The authors acknowledge support by the Research Foundation–Flanders and of KU Leuven grant GOA/12/14. The authors thank L.J. Waldorp for kindly providing the rsfMRI data and they thank the Editor, Associate Editor and the reviewers for their comments that resulted in an improved paper.



33; 18; 20; 10; 10; 15; 8 32; 23; 18; 14; 1; 25; 1 22; 15; 18; 13; 19; 14; 13 21; 17; 24; 13; 15; 16; 8 55; 22; 4; 2; 6; 11; 8; 2; 4

Figure 5: rsfMRI data. Top panels: estimated graphs using ComGGL$_1$ (panel a), ComGGL$_2$ (panel b), graphical lasso with the sbmSDP procedure (panel c), with GSBM (panel d) and with CORD (panel e). Bottom panels: estimated community membership for each node in the graph. The size of the labels is proportional to the degree of the node. The values under each figure represent the number of ROIs within each estimated community.

11. Appendix A

Proof [Proof of Proposition 1] The proof follows the idea of Theorem 1 of Lam and Fan (2009) which in turn follows the lines of Rothman et al. (2008) and Bickel and Levina (2008a), adapted to our objective function.
For any symmetric matrix $\boldsymbol{U}$ of dimension $p_n \times p_n$ with finite entries, let $\boldsymbol{D}_U$ be its diagonal matrix (a matrix of which the diagonal is equal to that of $\boldsymbol{U}$ and all other elements are 0) and $\boldsymbol{R}_U = \boldsymbol{U} - \boldsymbol{D}_U$ its off-diagonal matrix (this implies that $\boldsymbol{U} = \boldsymbol{D}_U + \boldsymbol{R}_U$) and let $\boldsymbol{\Delta}_U = \alpha_n \boldsymbol{R}_U + \beta_n \boldsymbol{D}_U$ where $\alpha_n = (s_n \log(p_n)/n)^{1/2} \to 0$ and $\beta_n = (p_n \log(p_n)/n)^{1/2} \to 0$. See page 4271 of Lam and Fan (2009). Let $\boldsymbol{V}$ be an arbitrary symmetric matrix of dimension $p_n \times p_n$ with finite entries and $\boldsymbol{\Delta}_V = \gamma_n \boldsymbol{V}$ where $\gamma_n = n^{-1/2}(p_n^2/K_n - p_n)^{1/2}$, which using assumptions (C) and (D) tends to 0.

The matrix $\boldsymbol{Z}\boldsymbol{Z}^\top$ is block structured with K_n blocks on the diagonal, each of dimension $p_n/K_n \times p_n/K_n$. The diagonals are by convention set to 1, thus p_n elements need not be estimated. The matrices $\boldsymbol{\Delta}_U$ and $\boldsymbol{\Delta}_V$ will be used as 'perturbation matrices' around the true $\boldsymbol{\Theta}_0$ and $\boldsymbol{Z}\boldsymbol{Z}^\top$, that will become smaller with increasing sample size and will determine the rate of convergence of the estimators. We make a distinction between the diagonal (that is, inverse variance entries) and off-diagonal entries (inverse covariance entries) for $\boldsymbol{\Theta}$.

Define $\mathcal{A}$ as a set of matrices

$$\mathcal{A} = \big\{(\boldsymbol{U}, \boldsymbol{V}) : \|\boldsymbol{\Delta}_U\|_F^2 = C_1^2\alpha_n^2 + C_2^2\beta_n^2 \text{ and } \|\boldsymbol{\Delta}_V\|_F^2 = C_3^2\gamma_n^2\big\},$$

where $\|\boldsymbol{\Delta}_U\|_F^2$ and $\|\boldsymbol{\Delta}_V\|_F^2$ are the squared Frobenius norms of the perturbation matrices, that is, $\|\boldsymbol{\Delta}_U\|_F^2 = \text{tr}(\boldsymbol{\Delta}_U^T\boldsymbol{\Delta}_U) = \sum_a \sum_b \boldsymbol{\Delta}_{U,a,b}^2$. This implies that $\|\alpha_n \boldsymbol{R}_U + \beta_n \boldsymbol{D}_U\|_F^2 = (C_1^2 s_n + C_2^2 p_n)\log(p_n)/n$.

Let $\boldsymbol{\Theta} = \boldsymbol{\Theta}_0 + \boldsymbol{\Delta}_U$, $\boldsymbol{X} = \boldsymbol{Z}\boldsymbol{Z}^\top + \boldsymbol{\Delta}_V$ and $\ell(\boldsymbol{\Theta}, \boldsymbol{X})$ be the objective function used in (2). It is sufficient to show that for sufficiently large constants C_1, C_2 and C_3 the probability

$$P\Big(\inf_{(\boldsymbol{U},\boldsymbol{V})\in\mathcal{A}} \ell(\boldsymbol{\Theta}_0 + \boldsymbol{\Delta}_U, \boldsymbol{Z}\boldsymbol{Z}^\top + \boldsymbol{\Delta}_V) > \ell(\boldsymbol{\Theta}_0, \boldsymbol{Z}\boldsymbol{Z}^\top)\Big) \to 1.$$

This implies that there exist minimizers $\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{X}}$ in the set

$$\big\{(\boldsymbol{\Theta}_0 + \boldsymbol{\Delta}_U, \boldsymbol{Z}\boldsymbol{Z}^\top + \boldsymbol{\Delta}_V) : \|\boldsymbol{\Delta}_U\|_F^2 \leq C_1^2\alpha_n^2 + C_2^2\beta_n^2 \text{ and } \|\boldsymbol{\Delta}_V\|_F^2 = C_3^2\gamma_n^2\big\}$$

such that $\max(\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_0\|_F, \|\hat{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{Z}^\top\|_F) = O_p\big(\max(\sqrt{(p_n + s_n)\log(p_n)/n}, \sqrt{p_n^2/(nK_n) - p_n/n})\big)$. Consider now the difference $\ell(\boldsymbol{\Theta}, \boldsymbol{X}) - \ell(\boldsymbol{\Theta}_0, \boldsymbol{Z}\boldsymbol{Z}^\top) = I_1 + I_2 + I_3 + I_4 + I_5$ where

$$\begin{aligned}
I_1 &= \text{tr}(\boldsymbol{S}\boldsymbol{\Theta}) - \log\det\boldsymbol{\Theta} - \{\text{tr}(\boldsymbol{S}\boldsymbol{\Theta}_0) - \log\det\boldsymbol{\Theta}_0\}\\
I_2 &= \lambda_{n1} \sum_{a,b\in\mathcal{S}^c} (|\boldsymbol{\Delta}_{U,a,b}|)\\
I_3 &= \lambda_{n1} \sum_{a,b\in\mathcal{S}} (|\boldsymbol{\Theta}_{a,b}| - |\boldsymbol{\Theta}_{0,a,b}|)\\
I_4 &= \text{tr}(\boldsymbol{\Theta}\boldsymbol{X}) - \text{tr}(\boldsymbol{\Theta}_0\boldsymbol{Z}\boldsymbol{Z}^\top)\\
I_5 &= \lambda_{n2} \sum_{k=1}^{K_n} \Bigg(\sqrt{\sum_{a\neq b\in\mathcal{C}_k} (\boldsymbol{\Theta}_{a,b}^k)^2} - \sqrt{\sum_{a\neq b\in\mathcal{C}_k} (\boldsymbol{\Theta}_{0,a,b}^k)^2}\Bigg).
\end{aligned}$$

Using assumptions (A) and (B) Lam and Fan (2009) showed that for constants C_1 and C_2 (i) the sum $I_1 + I_2 + I_3 > 0$ and (ii) the leading term in the sum is of order $O_p(C_1^2\alpha_n^2 + C_2^2\beta_n^2)$. It suffices to show that also $I_4 \geq 0$ and that it dominates the remaining term I_5.

We have that

$$I_4 = \text{tr}(\boldsymbol{\Theta}\boldsymbol{X}) - \text{tr}(\boldsymbol{\Theta}_0\boldsymbol{Z}\boldsymbol{Z}^\top) = \text{tr}((\boldsymbol{\Theta}_0 + \boldsymbol{\Delta}_U)\boldsymbol{X} - \boldsymbol{\Theta}_0\boldsymbol{Z}\boldsymbol{Z}^\top)$$
$$= \text{tr}(\boldsymbol{\Theta}_0(\boldsymbol{X} - \boldsymbol{Z}\boldsymbol{Z}^\top) + \boldsymbol{\Delta}_U\boldsymbol{X}) = \text{tr}(\boldsymbol{\Theta}_0\boldsymbol{\Delta}_V) + \text{tr}(\boldsymbol{\Delta}_U\boldsymbol{Z}\boldsymbol{Z}^\top) + \text{tr}(\boldsymbol{\Delta}_U\boldsymbol{\Delta}_V).$$

As in their value K_4, Supplement p.19, of Yin and Li, 2011,

$$\text{tr}(\boldsymbol{\Theta}_0\boldsymbol{\Delta}_V) \leq \text{eig}_{\max}(\boldsymbol{\Theta}_0)\|\text{vec}(\boldsymbol{\Delta}_V)\| = \text{eig}_{\max}(\boldsymbol{\Theta}_0)\sqrt{\text{vec}(\boldsymbol{\Delta}_V)^\top\text{vec}(\boldsymbol{\Delta}_V)}$$
$$\leq (1/\tau_1)\|\boldsymbol{\Delta}_V\|_F = (1/\tau_1)C_3\gamma_n$$

$$\text{tr}(\boldsymbol{\Delta}_U\boldsymbol{Z}\boldsymbol{Z}^\top) = \text{tr}(\boldsymbol{Z}\boldsymbol{Z}^\top\boldsymbol{\Delta}_U)$$
$$\leq \tau_3\|\boldsymbol{\Delta}_U\|_F = \tau_3\sqrt{C_1^2\alpha_n^2 + C_2^2\beta_n^2} = \tau_3 O_p(C_1\alpha_n + C_2\beta_n)$$

$$\text{tr}(\boldsymbol{\Delta}_U\boldsymbol{\Delta}_V) \leq \text{eig}_{\max}(\boldsymbol{\Delta}_U)\|\boldsymbol{\Delta}_V\|_F = o_p(C_3\gamma_n)$$

which implies that $I_4 = O_p(\alpha_n + \beta_n + \gamma_n)$.

We focus now on I_5.

$$|I_5| = \Big|\lambda_{n2}\sum_{k=1}^{K_n}\Big(\sqrt{\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{a,b})^2} - \sqrt{\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{0,a,b})^2}\Big)\Big|$$
$$\leq \lambda_{n2}\sum_{k=1}^{K_n}\frac{\big|\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{a,b})^2 - \sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{0,a,b})^2\big|}{\sqrt{\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{a,b})^2} + \sqrt{\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{0,a,b})^2}}$$
$$= \lambda_{n2}\sum_{k=1}^{K_n}\frac{\sum_{a\neq b\in\mathcal{C}_k}|(\boldsymbol{\Theta}^k_{0,a,b} + \boldsymbol{\Delta}^k_{U_{a,b}})^2 - (\boldsymbol{\Theta}^k_{0,a,b})^2|}{\sqrt{\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{a,b})^2} + \sqrt{\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{0,a,b})^2}}$$
$$\leq \frac{\lambda_{n2}}{\sqrt{\tau_4}}\sum_{k=1}^{K_n}\sum_{a\neq b\in\mathcal{C}_k}|(\boldsymbol{\Delta}^k_{U_{a,b}})^2 + 2\boldsymbol{\Theta}^k_{0,a,b}\boldsymbol{\Delta}^k_{U_{a,b}}| \quad \text{(using assumption E)}$$
$$\leq \frac{\lambda_{n2}}{\sqrt{\tau_4}}(\|\boldsymbol{\Delta}_U\|_F^2 + \sum_{k=1}^{K_n}\sum_{a\neq b\in\mathcal{C}_k}|2\boldsymbol{\Theta}^k_{0,a,b}||\boldsymbol{\Delta}^k_{U_{a,b}}|)$$
$$\leq \frac{\lambda_{n2}}{\sqrt{\tau_4}}(\|\boldsymbol{\Delta}_U\|_F^2 + \sum_{k=1}^{K_n}\sum_{a\neq b\in\mathcal{C}_k}|2\boldsymbol{\Theta}^k_{0,a,b}|\sqrt{s_n+p_n}\|\boldsymbol{\Delta}_U\|_F)$$
$$= \frac{\lambda_{n2}}{\sqrt{\tau_4}}(\|\boldsymbol{\Delta}_U\|_F^2 + M\sqrt{s_n+p_n}\|\boldsymbol{\Delta}_U\|_F),$$

for a general large constant M (that sums the off-diagonal elements of the true concentration matrix).

If $\lambda_{n2} = O(\sqrt{\log(p_n)/n})$ then

$$|I_5| \leq O_p\Big(\sqrt{\frac{\log p_n}{n}}(\alpha_n^2 + \beta_n^2) + \sqrt{\frac{\log p_n}{n}}\sqrt{s_n+p_n}\sqrt{\frac{(s_n+p_n)\log p_n}{n}}\Big)$$
$$= O_p\Big((\sqrt{\frac{\log p_n}{n}} + 1)(\alpha_n^2 + \beta_n^2)\Big).$$

For sufficiently large constants, $I_4 \geq 0$ dominates the term I_5 since the sequences involved in I_4 are tending towards 0 much slower than the sequences involved in I_5 and thus I_4 will dominate I_5.

■

Proof [Proof of Proposition 2]

Arguing as in Guo et al. (2011) and Lam and Fan (2009) it suffices to show that for all indices $(a,b) \in \mathcal{S}^c$, the derivative $\big(\partial/\partial \boldsymbol{\Theta}^k_{a,b}\big)\ell(\boldsymbol{\Theta}, \boldsymbol{X})$ evaluated at the sample realization of the estimators $\hat{\boldsymbol{\Theta}}^k_{a,b}$ and $\hat{\boldsymbol{X}}^k_{a,b}$ has with high probability, the same sign as the estimated value $\hat{\boldsymbol{\Theta}}^k_{a,b}$.

The partial derivative of $\ell(\boldsymbol{\Theta}, \boldsymbol{X})$ wrt $\boldsymbol{\Theta}^k_{a,b}$ is given as

$$\begin{aligned}\frac{\partial \ell(\boldsymbol{\Theta}, \boldsymbol{X})}{\partial \boldsymbol{\Theta}^k_{a,b}} &= 2\Bigg(\boldsymbol{S}^k_{a,b} - \boldsymbol{\Sigma}^k_{a,b} + \lambda_{n1}\text{sgn}(\boldsymbol{\Theta}^k_{a,b}) - \boldsymbol{X}^k_{a,b} + \frac{\lambda_{n2}\boldsymbol{\Theta}^k_{a,b}}{\sqrt{\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{a,b})^2}}\Bigg) \\ &= 2\Bigg(I_1 + I_2 + I_3 + \frac{\lambda_{n2}\boldsymbol{\Theta}^k_{a,b}}{\sqrt{\sum_{a\neq b\in\mathcal{C}_k}(\boldsymbol{\Theta}^k_{a,b})^2}}\Bigg),\end{aligned}$$

where $I_1 = \boldsymbol{S}^k_{a,b} - \boldsymbol{\Sigma}^k_{0,a,b}$, $I_2 = \boldsymbol{\Sigma}^k_{0,a,b} - \boldsymbol{\Sigma}^k_{a,b} - (\boldsymbol{Z}\boldsymbol{Z}^\top)^k_{a,b}$ and $I_3 = (\boldsymbol{Z}\boldsymbol{Z}^\top)^k_{a,b} - \boldsymbol{X}^k_{a,b}$.

Using the estimators $(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{X}})$ satisfying the necessary conditions stipulated in Propositions 1 and 2 follows, from Lam and Fan (2009) that $\max_{a,b}|I_1| = O_p(\sqrt{\log(p_n)/n})$.

We now consider I_2 and I_3. From Lemma 1, Lam and Fan, 2009,

$$\begin{aligned}|I_2| &= |\hat{\boldsymbol{\Sigma}}^k_{a,b} - \boldsymbol{\Sigma}^k_{0,a,b} + (\boldsymbol{Z}\boldsymbol{Z}^\top)^k_{a,b}| \leq \|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0 + \boldsymbol{Z}\boldsymbol{Z}^\top\| \\ &\leq \|\hat{\boldsymbol{\Sigma}} - \boldsymbol{\Sigma}_0\| + \|\boldsymbol{Z}\boldsymbol{Z}^\top\| = O_p(\sqrt{\eta_{n1}}) + O(1), \\ |I_3| &= |(\boldsymbol{Z}\boldsymbol{Z}^\top)^k_{a,b} - \hat{\boldsymbol{X}}^k_{a,b}| = |\hat{\boldsymbol{X}}^k_{a,b} - (\boldsymbol{Z}\boldsymbol{Z}^\top)^k_{a,b}| \leq \|\hat{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{Z}^\top\| = O_p(\sqrt{\eta_{n2}}).\end{aligned}$$

This implies that for sufficiently large constants $\max_{a,b}|I_1+I_2+I_3| = O_p(\sqrt{\log(p_n)/n}+\sqrt{\eta_{n1}}+\sqrt{\eta_{n2}})$. It can easily be seen that as long as (11) holds, the sign of the derivative evaluated at the estimated values $\hat{\boldsymbol{\Theta}}^k_{a,b}$ and $\hat{\boldsymbol{X}}^k_{a,b}$ will depend on the sign of $\hat{\boldsymbol{\Theta}}^k_{a,b}$ only. ■

Proof [Proof of Proposition 4] Following the version of 'Davis Kahan sin θ' theorem presented in Theorem 2 of Yu et al. (2015) we have that there exists an orthogonal matrix $\boldsymbol{O}$ such that

$$\frac{1}{\sqrt{2K_n}}\|\hat{\boldsymbol{U}} - \boldsymbol{U}\boldsymbol{O}\|_F \leq \frac{2^{3/2}\min(\sqrt{(s-r+1)}\|\hat{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{Z}^\top\|, \|\hat{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{Z}^\top\|_F)}{\min(\text{eig}_{r-1} - \text{eig}_r, \text{eig}_s - \text{eig}_{s+1})},$$

where s and r denote the positions of the ordered (from large to small) eigenvalues of the matrix $\boldsymbol{Z}\boldsymbol{Z}^\top$. Using Proposition 1 we have that $\|\hat{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{Z}^\top\| \leq \|\hat{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{Z}^\top\|_F = O_p(n^{-1/2}\sqrt{p_n^2/K_n - p_n})$. This implies that

$$\|\hat{\boldsymbol{U}} - \boldsymbol{U}\boldsymbol{O}\|_F \leq \kappa\sqrt{\frac{p_n^2}{n} - \frac{K_n p_n}{n}},$$

where κ is an unknown positive constant.

The rest of the proof follows as in the proof of Theorem 1 of Lei and Rinaldo (2015) since their Lemma 5.3 covers the k-means problem posed in (12) and since their Lemma 2.1 is replaced by our Proposition 3 for this specific context. As such, following the same reasoning yields for the ComGGL procedure that

$$\sum_{k=1}^{K_n}\frac{\#S_k}{\#\mathcal{C}_k} \leq 4(4+2\xi)\|\hat{\boldsymbol{X}} - \boldsymbol{Z}\boldsymbol{Z}^\top\|_F^2 \leq 4(4+2\xi)\kappa\sqrt{\frac{p_n^2}{n} - \frac{K_n p_n}{n}} = c^{-1}(2+\xi)\sqrt{\frac{p_n^2}{n} - \frac{K_n p_n}{n}},$$

where c is an unknown positive constant. ■

12. Appendix B

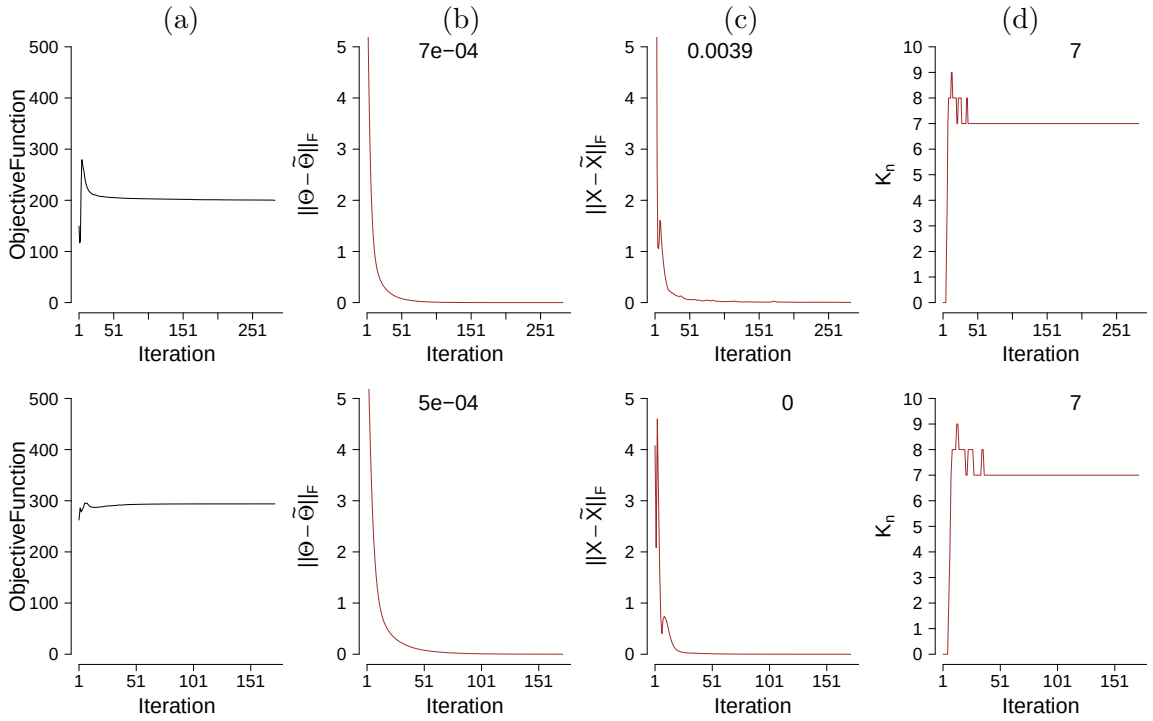


Figure 6: rsfMRI data, see Section 9. Assessing convergence of the ADMM algorithm for ComGGL$_1$ (top row) and ComGGL$_2$ (bottom row). The panels show at each iteration (a) the value of the objective function (should stabilize), (b) the Frobenius norm of $\hat{\mathbf{\Theta}} - \widetilde{\mathbf{\Theta}}$ (should approach 0), (c) the Frobenius norm of $\hat{\boldsymbol{X}} - \widetilde{\boldsymbol{X}}$ (should approach 0) and (d) the estimated number of communities K_n (should stabilize). The number on top of each graph gives the value at the last iteration of the algorithm.

It took ComGGL$_1$ for the rsfMRI example, on a standard laptop, 5.5 seconds (283 iterations) to converge when the tolerance threshold was 10^{-4}, 9.5 seconds (466 iterations) when the tolerance threshold was 10^{-5} and 25.8 seconds (1282 iterations) when the tolerance threshold was 10^{-6}. The computational advantage provided by (4) resulted in lower computational complexity and convergence was attained for ComGGL$_2$ much faster: 3.5 seconds (172 iterations), 4.8 seconds (226 iterations) and 5.8 seconds (282 iterations) for the same thresholds. The two-step GL & GSBM procedure needed 5.9 seconds (447 iterations), 6.7 seconds (464 iterations) and 14.1 seconds (897 iterations) to estimate both the graph and the communities, while GL & sbmSDP needed 4.8 seconds (855 iterations), 12.5 (2298 iterations) and 38.7 seconds (7122 iterations). This exercise illustrates that with respect to the competitor procedures, ComGGL can be more time consuming to reach the same accuracy level.

References

E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9:1981–2014, 2008.

E. M. Airoldi, T. B. Costa, and S. H. Chan. Stochastic blockmodel approximation of a graphon: Theory and consistent estimation. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 692–700, 2013.

A. A. Amini and E. Levina. On semidefinite relaxations for the block model. *The Annals of Statistics*, 46(1):149–179, 2018.

A. A. Amini, A. Chen, P. J. Bickel, and E. Levina. Pseudo-likelihood methods for community detection in large sparse networks. *The Annals of Statistics*, 41(4):2097–2122, 2013.

E. Arias-Castro and N. Verzelen. Community detection in dense random networks. *The Annals of Statistics*, 42(3):940–969, 2014.

J. D. Arroyo Relión, D. Kessler, E. Levina, and S. F. Taylor. Network classification with applications to brain connectomics. *The Annals of Applied Statistics*, 13(3):1648–1677, 09 2019.

S. Arslan, S. I. Ktena, A. Makropoulos, E. C. Robinson, D. Rueckert, and S. Parisot. Human brain mapping: A systematic comparison of parcellation methods for the human cerebral cortex. *NeuroImage*, 170:5–30, 2018.

P. J. Bickel and E. Levina. Covariance regularization by thresholding. *The Annals of Statistics*, 36 (6):2577–2604, 2008a.

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

X. Bing, F. Bunea, Y. Ning, and M. Wegkamp. Adaptive estimation in structured factor models with applications to overlapping clustering. *The Annals of Statistics*, to appear, 2020.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011.

C. Brownlees, G. S. Gudmundsson, and G. Lugosi. Community detection in partial correlation network models. Technical report, Universitat Pompeu Fabra, 2018.

F. Bunea, C. Giraud, and X. Luo. Minimax optimal variable clustering in G-models via Cord. *arXiv:*, 1508.01939v2, 2016. URL `https://arxiv.org/abs/1508.01939v2`.

T. T. Cai and X. Li. Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. *The Annals of Statistics*, 43(3):1027–1059, 2015.

Y. Chen, S. Sanghavi, and H. Xu. Clustering sparse graphs. In *Proceedings of the 25th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 2204–2212, 2012.

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

W. Deng and W. Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3):889–916, 2016.

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

J. Guo, E. Levina, G. Michailidis, and J. Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011.

P. W. Holland, K. B. Laskey, and S. Leinhardt. Stochastic blockmodels: first steps. *Social Networks*, 5(2):109–137, 1983.

C. Lam and J. Fan. Sparsistency and rates of convergence in large covariance matrix estimation. *The Annals of Statistics*, 37(6B):4254–4278, 2009.

S.L. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

C. M. Le and E. Levina. Estimating the number of communities in networks by spectral methods. *arXiv:*, 1507.00827, 2015. URL `http://arxiv.org/abs/1507.00827`.

J. Lei and A. Rinaldo. Consistency of spectral clustering in stochastic block models. *The Annals of Statistics*, 43(1):215–237, 2015.

J. Leskovec, K. J. Lang, and M. Mahoney. Empirical comparison of algorithms for network community detection. In *Proceedings of the 19th International Conference on World Wide Web*, pages 631–640. ACM, 2010.

R. Mazumder and T. Hastie. The graphical lasso: New insights and alternatives. *Electronic Journal of Statistics*, 6:2125–2149, 2012.

A.J. Molstad and A.J. Rothman. Shrinking characteristics of precision matrix estimators. *Biometrika*, 105(3):563–574, 2018.

D. M. Pavlović. *Generalised Stochastic Blockmodels and their Applications in the Analysis of Brain Networks*. PhD thesis, University of Warwick, 2015.

E. Pircalabelu, G. Claeskens, and L. J. Waldorp. Mixed scale joint graphical lasso. *Biostatistics*, 17 (4):793–806, 2016.

T. Qin and K. Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In C. J. C. Burges, L. Bottou, Z. Ghahramani, and K.Q. Weinberger, editors, *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS)*, pages 3120–3128, 2013.

P. D. Ravikumar, G. Raskutti, M. J. Wainwright, and B. Yu. Model selection in Gaussian graphical models: High-dimensional consistency of l_1-regularized MLE. In *Proceedings of the 22nd Annual Conference on Neural Information Processing Systems (NIPS)*, pages 1329–1336, 2008.

K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 2011.

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

T. Saegusa and A. Shojaie. Joint estimation of precision matrices in heterogeneous populations. *Electronic Journal of Statistics*, 10(1):1341–1392, 2016.

V. D. Schmittmann, S. Jahfari, D. Borsboom, A. O. Savi, and L. J. Waldorp. Making large-scale networks from fMRI data. *PloS one*, 10(9):e0129074, 2015.

K. M. Tan, D M. Witten, and A. Shojaie. The cluster graphical lasso for improved estimation of Gaussian graphical models. *Computational Statistics & Data Analysis*, 85:23–36, 2015.

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

B. T. T. Yeo, F. M. Krienen, J. Sepulcre, M. R. Sabuncu, D. Lashkari, M. Hollinshead, J. L. Roffman, J. W. Smoller, L. Zollei, J. R. Polimeni, B. Fischl, H. Liu, and R. L. Buckner. The organization of the human cerebral cortex estimated by functional correlation. *Journal of neurophysiology*, 106: 1125–1165, 2011.

J. Yin and H. Li. A sparse conditional Gaussian graphical model for analysis of genetical genomics data. *The Annals of Applied Statistics*, 5(4):2630–2650, 2011.

Y. Yu, T. Wang, and R. J. Samworth. A useful variant of the Davis-Kahan theorem for statisticians. *Biometrika*, 102(2):315–323, 2015.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68:49–67, 2006.