Modularized Gaussian Graphical Model

Jian Guo[a], Sijian Wang[b]

[a]*Department of Statistics, University of Michigan*
[b]*Department of Biostatistics and Medical Informatics, Department of Statistics, University of Wisconsin*

Abstract

Gaussian graphical models explore the partial correlation structures between random variables and illustrate them using networks. In this paper, we proposed to improve the estimation of Gaussian graphical models by incorporating the unknown modular structure existing in many real world networks. The proposed method simultaneously identifies the modules and estimates the network structure. Several numerical experiments on simulated and real data sets demonstrate that the estimation of the graphical models can be improved by appropriately utilizing prior modular information.

Keywords: Graphical model, Group penalty, Lasso, Modular network, Precision matrix

1. Introduction

Gaussian graphical model characterizes the partial correlation structures between continuous random variables using undirected networks, where each node represents a variable and each edge indicates the existence of partial correlation between the two variables connected by this edge. This model has been applied to estimate the the structure of many real networks such as information networks and biological networks. In the Gaussian graphical model, all variables are assumed to jointly follow a multivariate Gaussian distribution, which has a good property that each off-diagonal entry of its inverse covariance matrix (namely precision matrix) is proportional to the partial correlation between the two corresponding variables. Therefore, the structure of the underlying network is determined by the nonzero entries of the precision matrix. The estimation methods for Gaussian graphical model have been extensively studied in Meinshausen and Buhlmann (2006); Yuan

and Lin (2007); Banerjee et al. (2008); Friedman et al. (2008); Rothman et al. (2008); Fan et al. (2009); Rocha et al. (2008); Peng et al. (2009) and references therein.

Many real life networks exhibit modular structures. Specifically, the networks consist of a number of subnetworks, namely *modules* (also referred to as *communities* or *clusters*). The within-module connections‘ are usually much more popular than the between-module connections. Figure 1 illustrates a network with three modules. Examples of modular networks include gene transcriptional regulatory networks (Lee et al., 2002), internet and world wide webs, scientific citation and collaboration networks and social networks (Newman and Girvan, 2004). The estimation of Gaussian graphical

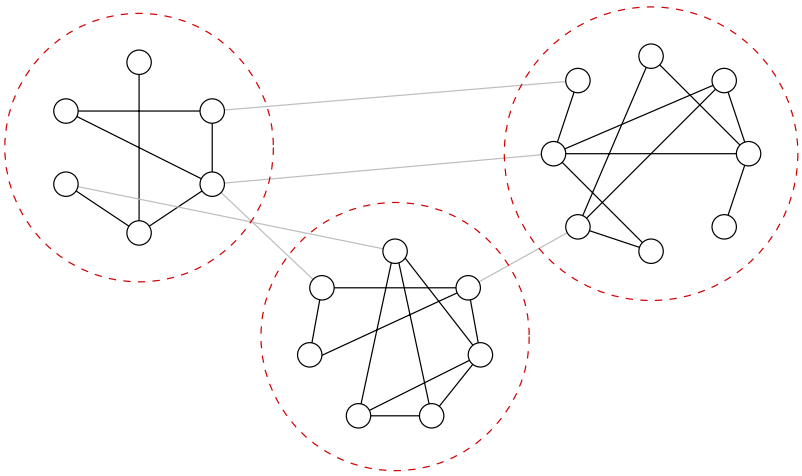

Figure 1: Illustration of a network with three modules. This figure was copied from Newman and Girvan (2004).

model may be improved by incorporating the prior knowledge of the modular structure. When the modules are known, Duchi et al. (2008) and Schmidt et al. (2009) proposed to penalize the ℓ_∞-norm and ℓ_2-norm of the block in the precision matrix associated with each module in the network. Although these two methods are more effective in removing unimportant links between modules, they involves all links within each block and this is not realistic in many practical problems. When the modular structure is unknown, Marlin and Murphy (2009) introduced a Bayesian approach which jointly discovers the modules and estimate the sparse precision matrix. Ambroise et al. (2009) proposed a variational EM algorithm to address this problem.

We proposed a new approach, modularized Gaussian graphical model (MGGM), to improve the estimation of the Gaussian graphs for the networks with the unknown modular structures. Furthermore, with the MGGM framework, we developed two methods, MGGM-DP and MGGM-GP, to encourage the within-module links and discourage the between-module links. Figure 2 illustrates the improved estimation achieved by MGGM-DP. The true graph (shown in Panel (A)) is composed of five disjointed modules, each of which is a clique (i.e., fully connected subgraph). Compare to GLasso (Friedman et al., 2008) (Panel (B)), the proposed method (Panel (C)) achieves a better structure estimation result.

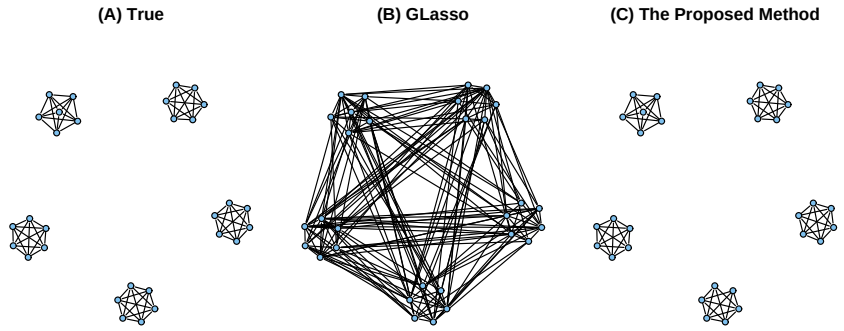

Figure 2: Illustration of the improved estimation achieved by the proposed MGGM-DP method which utilizes the modular information. The true graph (Panel (A)) is composed of five disjointed modules, each of which is a fully connected subgraph (clique). Panel (B) shows the graph estimated by GLasso. Panel (C) shows the graph estimated by MGGM-DP.

The remaining part of this paper is organized as follows. Section 2 introduces the models and discusses the algorithmic issues. The performance of the propose method on synthetic and real data is demonstrated in Section 3 and 4.

2. Methodology

In this section, we introduce the models and discuss the associated algorithms. Section 2.1 reviews the estimation method for sparse precision matrix in Gaussian graphical model, while Section 2.2 introduces the framework of the proposed MGGM method and provides two concrete MGGM estimators.

2.1. Preliminary and Review

Consider we have a data matrix $\boldsymbol{X} = (x_{i,j})_{n\times p}$ with n observations and p variables. Without loss of generality, we assume all variables (columns) are centered, i.e., $\sum_{i=1}^{n} x_{i,j} = 0$ for all $j = 1, 2, \ldots, p$. In the Gaussian graphical model, the observations in $\boldsymbol{X}$ are *i.i.d.* generated from a p-variate Gaussian distribution $\mathrm{N}(\mathbf{0}, \boldsymbol{\Omega}^{-1})$, where $\boldsymbol{\Omega} = (\omega_{j,j'})_{p\times p}$ is a precision matrix. The log-likelihood of the joint distribution is:

$$l(\boldsymbol{\Omega}) = -\frac{n}{2}\log(2\pi) + \frac{n}{2}[\log(\det(\boldsymbol{\Omega})) - \mathrm{trace}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Omega})] \ , \tag{1}$$

where $\log(\cdot)$ is the natural logarithm and $\det(\cdot)$ and $\mathrm{trace}(\cdot)$ are the determinant and the trace of a matrix, respectively.

To estimate a sparse precision matrix $\boldsymbol{\Omega}$, we consider the following penalized log-likelihood:

$$\max_{\boldsymbol{\Omega}\succ 0} \ \log(\det(\boldsymbol{\Omega})) - \mathrm{trace}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Omega}) - \lambda \sum_{j<j'} |\omega_{j,j'}| \ , \tag{2}$$

where $\boldsymbol{\Omega} \succ 0$ indicates $\boldsymbol{\Omega}$ is symmetric and positive definite. Due to the ℓ_1-norm penalty, the tuning parameter λ controls the sparsity of the estimates. When $\lambda = 0$, criterion (2) reduces to the maximum likelihood estimation (MLE) problem which has a closed form solution $\widehat{\boldsymbol{\Omega}} = (\widehat{\boldsymbol{\Sigma}})^{-1}$, where $\widehat{\boldsymbol{\Sigma}}$ is the sample covariance matrix. When $\lambda > 0$, the ℓ_1 penalty shrinks some off-diagonal elements in $\boldsymbol{\Omega}$ to be exactly zero and thus imposes the sparsity of the precision matrices. The solution of (2) has been studied extensively in literatures. In this paper, the GLasso algorithm (Friedman et al., 2008) was used to solve criterion (2) and the estimation results were used as a benchmark.

2.2. Modularized Gaussian Graphical Model

In this section, we introduce the MGGM method. For the convenience of description, we first list the framework of MGGM and then explain it step by step.

Framework of MGGM

Step 1 (Initialization). Initialize the sparse precision matrix $\widehat{\boldsymbol{\Omega}} = (\widehat{\omega}_{j,j'})_{p\times p}$ by solving (2) using the GLasso algorithm and compute an adjacency matrix $\boldsymbol{W} = (w_{j,j'})_{p\times p}$ as

$$w_{j,j'} = \begin{cases} |\widehat{\omega}_{j,j'}|, & \text{if } j \neq j'; \\ 0, & \text{if } j = j'. \end{cases}$$

Step 2 (Identification of modules). Given the adjacency matrix $\boldsymbol{W}$, partition the p variables into K disjointed clusters indexed by $\{g_k\}_{k=1}^K$ using the normalized cut algorithm.
Step 3 (Estimation of Graphs). Update the estimate of $\boldsymbol{\Omega}$ by solving the following criterion:

$$\max_{\boldsymbol{\Omega}\succ 0} \log(\det(\boldsymbol{\Omega})) - \text{trace}(\widehat{\boldsymbol{\Sigma}}\boldsymbol{\Omega}) - \text{P}(\boldsymbol{\Omega}; \{g_k\}_{k=1}^K) \ , \qquad (3)$$

where $\text{P}(\boldsymbol{\Omega}; \{g_k\}_{k=1}^K)$ is a penalty function on $\boldsymbol{\Omega}$ with respect to the partition $\{g_k\}_{k=1}^K$.

The algorithm is designed to simultaneously detect modules and estimate the sparse precision matrix. In Step 1, MGGM initializes a graph by estimating the sparse precision matrix using the GLasso algorithm. The edge weights in the graph are the magnitude of the elements in the precision matrix. In Step 2, MGGM identifies the modules in the graph, i.e., it partitions all nodes into K disjointed clusters. Many network modularizing/clustering algorithms can be applied here to discover the clusters. In this work, we applied the normalized cut algorithm proposed by Yu and Shi (2003). In Step 3, MGGM estimates the sparse precision matrix using the modular structure identified in Step 2. Specifically, it solves a regularized likelihood estimation problem with penalty $\text{P}(\boldsymbol{\Omega}; \{g_k\}_{k=1}^K)$, which depends on the modular partition $\{g_k\}_{k=1}^K$. This penalty encourages fewer connections between different modules, as well as more connections within the modules. Two concrete examples of this penalty function, the double regularization penalty (DP) and the group penalty (GP), are introduced in Section 2.2.1 and Section 2.2.2, respectively. The MGGM methods associated with penalties DP and GP are referred to as MGGM-DP and MGGM-GP, respectively.

Remark *Step 2 and 3 in the algorithm framework of the modularized Gaussian graphical model can be repeated to further update the estimates. However, numerical studies showed that the iterative process only marginally improved the estimates.*

2.2.1. Double Regularization Penalty

The double regularization penalty is defined as follows

$$\text{P}^{DP}(\boldsymbol{\Omega}; \{g_k\}_{k=1}^K) = \lambda_1 \sum_{k=1}^{K} \sum_{j,j'\in g_k; j<j'} |\omega_{j,j'}| + \lambda_2 \sum_{k<k'} \sum_{j\in g_k, j'\in g_{k'}} |\omega_{j,j'}| \ . \qquad (4)$$

This penalty function includes two tuning parameters λ_1 and λ_2, controlling the sparsity of the within-module elements and the between-module elements of $\mathbf{\Omega}$, respectively. The objective function (3) with the penalty (4) can be solved by the GLasso algorithm. Since we restrict $\lambda_1 < \lambda_2$ when selecting the tuning parameters, this penalty naturally encourages more sparsity on between-module connections.

2.2.2. Group Penalty

Another concrete example of $\mathrm{P}(\mathbf{\Omega}; \{g_k\}_{k=1}^K)$ is the group penalty (GP) defined as follows:

$$\mathrm{P}^{GP}(\mathbf{\Omega}; \{g_k\}_{k=1}^K) = \lambda(\sum_{k=1}^{K} \sqrt{\sum_{j,j' \in g_k; j<j'} |\omega_{j,j'}|} + \sum_{k<k'} \sum_{j \in g_k, j' \in g_{k'}} \sqrt{|\omega_{j,j'}|}) \ . \quad (5)$$

The first part of this penalty, $\sum_{k=1}^K \sqrt{\sum_{j,j' \in g_k; j<j'} |\omega_{j,j'}|}$, treats all $\omega_{j,j'}$'s in the same module as a group and shrinks them simultaneously, whereas the second part, $\sum_{k<k'} \sum_{j \in g_k, j' \in g_{k'}} \sqrt{|\omega_{j,j'}|}$, penalizes the between module elements individually. Due to the nature of the ℓ_1-norm under each square root, this penalty also shrinks some $\omega_{j,j'}$'s exactly to zero and thus it allows the flexibility of link selection within each module.

The objective function (3) with the group penalty (5) can be iteratively solved using the local linear approximation (LLA) algorithm proposed by Zou and Li (2008). Specifically, when $\omega_{j,j'} \approx \omega_{j,j'}^{(t^*)}$, we consider approximating the penalty functions

$$\begin{aligned} \sum_{k=1}^{K} \sqrt{\sum_{j,j' \in g_k; j<j'} |\omega_{j,j'}|} &\approx \sum_{k=1}^{K} \frac{\sum_{j,j' \in g_k; j<j'} |\omega_{j,j'}|}{\sqrt{\sum_{j,j' \in g_k; j<j'} |\omega_{j,j'}^{(t^*)}|}} \\ \sum_{k<k'} \sum_{j \in g_k, j' \in g_{k'}} \sqrt{|\omega_{j,j'}|} &\approx \sum_{k<k'} \sum_{j \in g_k, j' \in g_{k'}} \frac{|\omega_{j,j'}|}{\sqrt{|\omega_{j,j'}^{(t^*)}|}} \ . \end{aligned}$$

This approximation, together with substituting the estimate in step t^*, $\omega_{j,j'}^{(t^*)}$, in the denominator of the ratios, converts the maximization problem into a sparse precision matrix estimation problem:

$$\max_{\mathbf{\Omega} \succ 0} \quad \log(\det(\mathbf{\Omega})) - \mathrm{trace}(\widehat{\mathbf{\Sigma}}\mathbf{\Omega}) - \lambda \sum_{j<j'} \alpha_{j,j'} |\omega_{j,j'}| \ , \quad (6)$$

where

$$\alpha_{j,j'} = \begin{cases} 1/\sqrt{\sum_{j,j' \in g_k; j<j'} |\omega^{(t^*)}_{j,j'}|}, & \text{if } j, j' \in g_k; j < j'; \\ 1/\sqrt{|\omega^{(t^*)}_{j,j'}|}, & \text{if } j \in g_k, j' \in g_{k'}; k < k'. \end{cases}$$

Objective function (6) can be efficiently solved by the GLasso algorithm. For numerical stability, we threshold the absolute value of $\sqrt{\sum_{j,j' \in g_k; j<j'} |\omega^{(t^*)}_{j,j'}|}$ and $\sqrt{|\omega^{(t^*)}_{j,j'}|}$ at 10^{-10} over different iterations. Objective function (6) also explains the intuition that the group penalty (5) encourages more connections within the modules. In fact, for each within-module $\omega_{j,j'}$, the weight $\alpha_{j,j'}$ depends on the sum of the magnitudes of all updated $\widehat{\omega}_{j,j'}$ belonging to this module, and thus it is generally smaller than the weight of a between-module element $\omega_{j,j'}$, which only depends on the estimates of themselves. Note that the similar group penalties have been used in the the variable selection problems for linear regression (Zhou and Zhu, 2007) and survival analysis (Wang et al., 2009), as well as the joint estimation problems for multiple graphical models (Guo et al., 2010a,b).

3. Simulated Examples

In this section, we examine the performance of the proposed modularized Gaussian graphical model using three simulated examples, each consisting of several disjointed modules representing different types of graphs, such as the clique graph, the nearest-neighbor graph and the scale-free graph. The latter two graphs have been applied in the simulation study of in existing literatures (Li and Gui, 2006; Fan et al., 2009; Guo et al., 2010a,b).

Example 1: Clique Modules This example simulates a graph composed of five disjointed modules each being a six-node-clique (Panel (A) of Figure 3).

Example 2: Nearest-neighbor Modules This example considers a network consisting of several nearest-neighbor modules, each of which was generated by the mechanism introduced in Li and Gui (2006). Specifically, for a module with q nodes, we generate q points randomly on a unit square, calculate all pairwise distances, and find the nearest five neighbors of each point. Then any two points are connected if they are

nearest neighbors of each other. In this example, we consider two networks. The first one consists of two nearest neighbor subgraphs with sizes 10 and 20, respectively (Panel (A) of Figure 4). The second one consists of four subgraphs with sizes 10, 20, 30 and 40, respectively. From Panel (A) of Figure 5, we can see that the subgraph with 20 nodes splits into two disjointed parts with sizes 6 and 14, respectively, whereas the subgraph with 30 nodes splits into three parts with sizes 3, 5 and 22, respectively. For clarification, we refer to it as a network with seven modules.

Example 3: Scale-free Modules This example generates two networks, one consisting of two scale-free modules with sizes 10 and 20, respectively, and the other consisting of four scale-free modules with sizes 10, 20, 30 and 40, respectively. The scale-free models are generated using the Barabasi-Albert algorithm (Barabasi and Albert, 1999). The two networks are illustrated in Figure 6 (Panel (A)) and Figure 7 (Panel (A)), respectively.

The edges of the networks in Examples 1–3 were weighted using the mechanism in Fan et al. (2009). Specifically, each nonzero off-diagonal element in $\mathbf{\Omega}$ is drawn randomly from intervals $[-1, -0.5] \cup [0.5, 1]$. Each diagonal element of $\mathbf{\Omega}$ is set as a factor of the sum of the absolute values of all off-diagonal elements in the same row. The factor is selected to ensure the positive definiteness of $\mathbf{\Omega}$. Finally, $\mathbf{\Omega}$ is scaled to have unit diagonal elements.

For each example, we simulate 50 data sets each with $n = 80$ observations *i.i.d.* generated from multivariate normal distribution $N(\mathbf{0}, \mathbf{\Omega}^{-1})$. The performance of the modules is measured by several criterions, including the entropy loss (EL) and the Frobenius loss (FL):

$$\begin{aligned} \text{EL} &= \text{trace}(\mathbf{\Omega}^{-1}\widehat{\mathbf{\Omega}}) - \log(\det(\mathbf{\Omega}^{-1}\widehat{\mathbf{\Omega}})) - p \\ \text{FL} &= \|\mathbf{\Omega} - \widehat{\mathbf{\Omega}}\|_F^2 / \|\mathbf{\Omega}\|_F^2 \ , \end{aligned}$$

where $\|\cdot\|_F$ is a matrix *Frobenius* norm. Smaller EL and FL indicate better estimation result. The estimation performance is also measured by the false positive rate (FP, the proportion of the incorrectly identified nonzeros) and the false negative rate (FN, the proportion of the incorrectly identified zeros).

In MGGM, several arguments are required to be specified in advance. These include the tuning parameter of the GLasso algorithm in Step 1, the number of clusters K in Step 2 and the tuning parameters in Step 3 (the

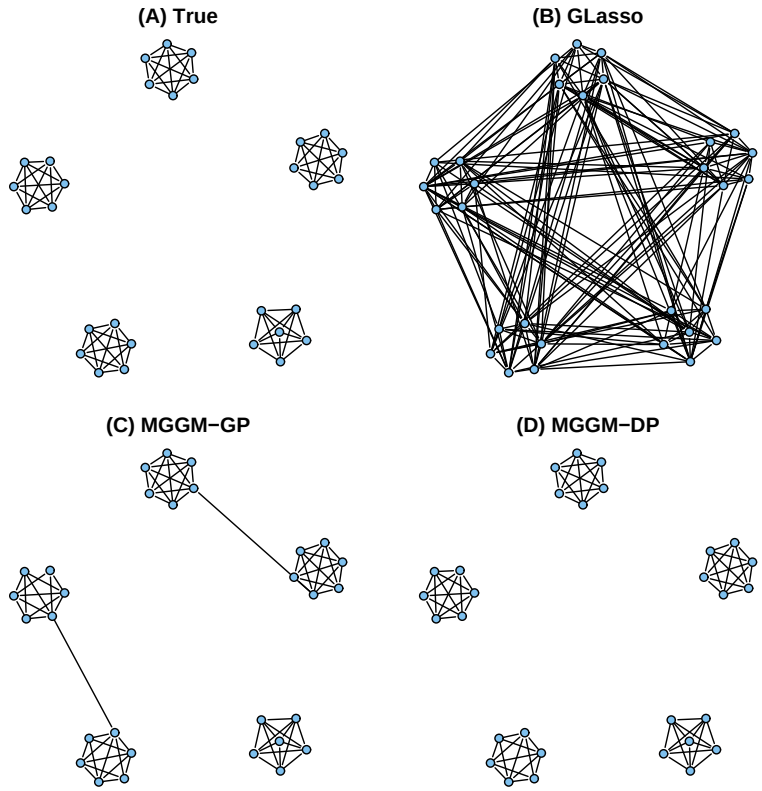

Figure 3: The true and the estimated networks randomly picked from one out of the 50 replications in Example 1.

tuning parameters λ_1 and λ_2 for the double regularization penalty and the tuning parameter λ for the group penalty). These arguments can be selected by the M-fold cross validation method. Specifically, all observations in the data set are randomly divided into M disjointed subsets with equal size, namely folds. Then the arguments are selected to maximize the cross validation score defined as

$$\mathrm{S}_{CV}(\mathbf{\Omega}) = \sum_{m=1}^{M} [\log(\det(\widehat{\mathbf{\Omega}}^{[-m]})) - \mathrm{trace}(\widehat{\mathbf{\Sigma}}^{[m]}\widehat{\mathbf{\Omega}}^{[-m]})] \ , \tag{7}$$

where $\widehat{\mathbf{\Sigma}}^{[m]}$ is the sample covariance matrix estimated by the observations in the m-th fold and $\widehat{\mathbf{\Omega}}^{[-m]}$ is the estimate of $\mathbf{\Omega}$ using all observations except those in the m-th fold.

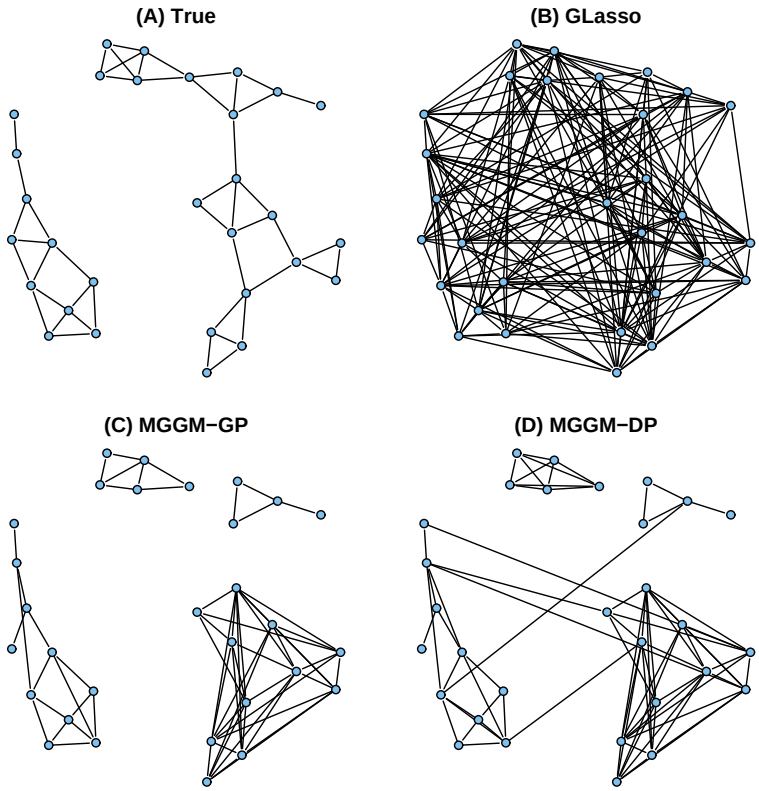

Figure 4: The true and the estimated networks in one out of the 50 replications in Example 2 (two-module network).

Table 1 summarizes the estimation results averaged over 50 replications. Compared with Glasso, MGGM-GP and MGGM-DP achieve lower entropy loss and Frobenius loss and they also effectively reduce the false positive rate. Figure 3–7 illustrate the estimated networks randomly picked from one out of 50 replications. In all these example, MGGM-GP and MGGM-DP conduct better estimation of the modular structures in the networks.

4. Real Example

In the section, we applied MGGM-DP and MGGM-GP to analyze a data set collecting the stock prices of S&P 100 companies. S&P 100 is a stock market index developed by the Chicago Board Options Exchange (CBOE), which later transferred the index to Standard & Poor's. The S&P 100 data

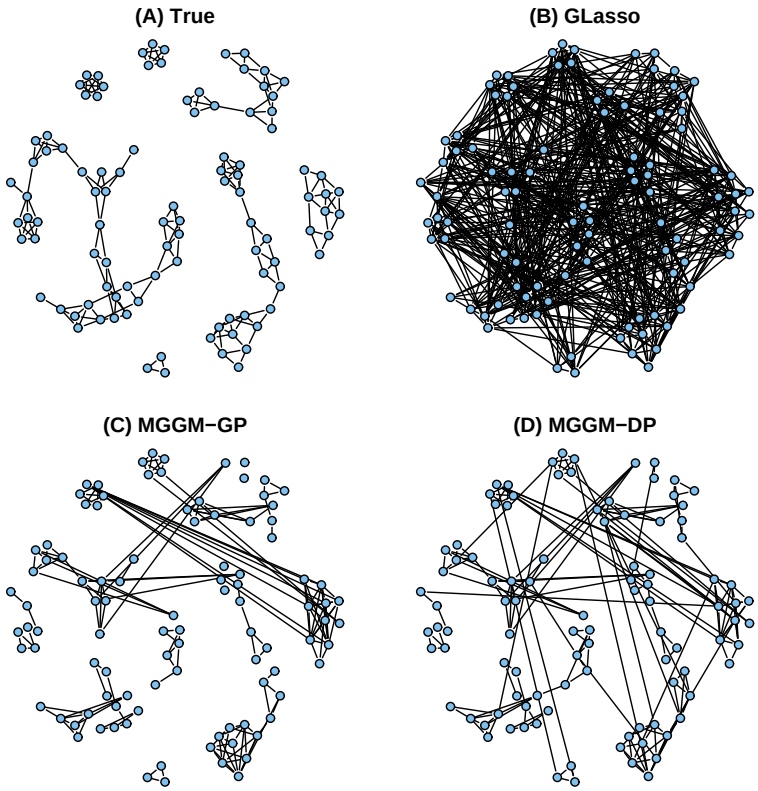

Figure 5: The true and the estimated networks in one out of the 50 replications in Example 2 (seven-module network).

set consists of the 100 leading U.S. stocks with exchange-listed options. The stocks in the S&P 100 are generally among the largest and most established companies in the famous S&P 500. The daily log-return is defined as the logarithm of the ratios of the stock prices between every two consecutive days and the log-returns are approximately assumed to be *i.i.d.* in financial market analysis. We collected the daily log-returns of the S&P 100 companies from January 1st, 2006 to December 31th, 2006. To remove the market environment effect, the log-returns of S&P 100 index (a linear combination of the log-returns of the S&P 100 companies) were regressed on the log-returns of the 100 companies and the residuals were collected in a new data set with 100 variables and 251 observations.

The networks estimated by MGGM-GP and MGGM-DP were illustrated in Figure 8 and 9, respectively. In Figure 8, we can see that the network

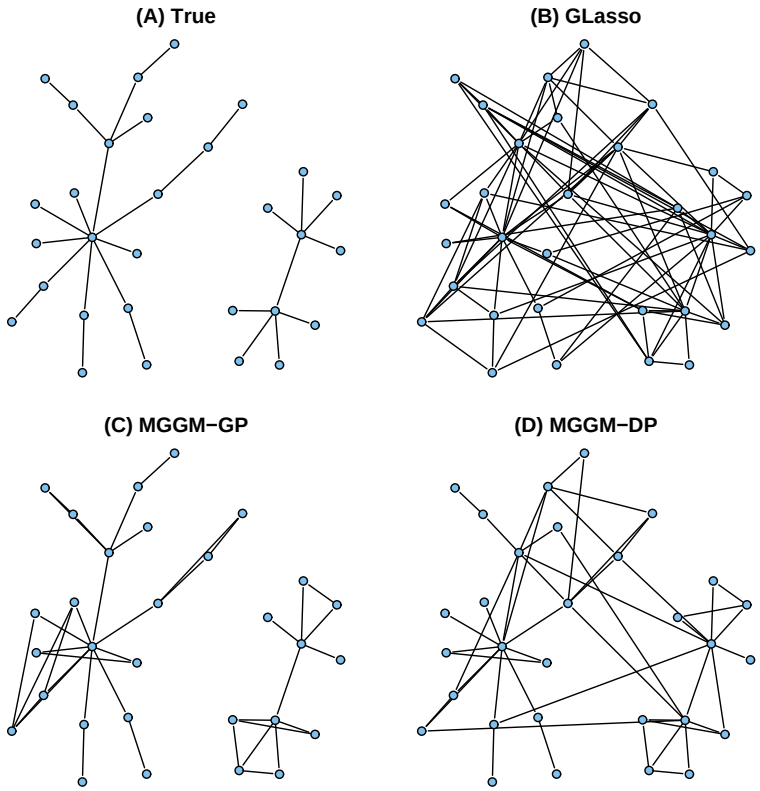

Figure 6: The true and the estimated networks in one out of the 50 replications in Example 3 (two-module network).

estimated by MGGM-GP includes three large modules. The first module includes several information technology companies (IBM, MSFT, SLB and WMB), as well as several finance companies (BK, C, HIG, JPM, RF, USB and WFC). The second module includes a few companies related to healthcare service and drug/food industry (CI, JNJ, KFT, PEP, PG and UNH), as well as a few companies about shipping and transportation service (FDX, NSC and UPS). The remaining module consists of four energy companies: AEP, ETR, EXC and SO. The network estimated by MGGM-DP exactly covers the three large modules discovered by MGGM-GP. In addition, MGGM-DP also detects another large module (the one consists of a few companies related to aerospace technology: BA, GD, HON, RTX and UTX), as well as a few small cliques. For example, one small clique includes three investment banks (GS, LEH and MS), while another includes two automakers (GM and

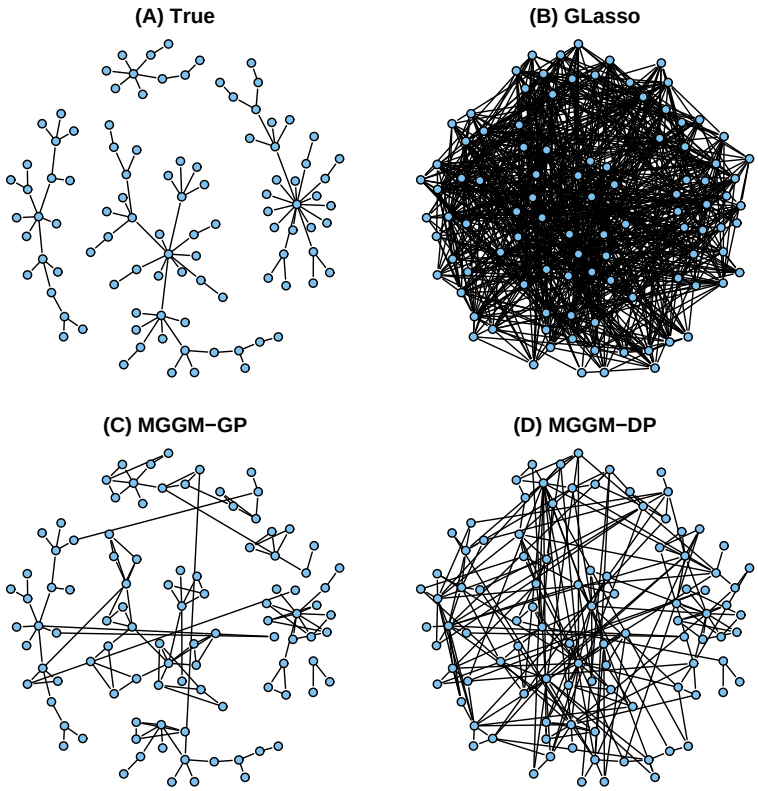

Figure 7: The true and the estimated networks in one out of the 50 replications in Example 3 (four-module network).

F).

Acknowledgements

The authors thank Prof. Elizaveta Levina, Prof. George Michailidis and Prof. Ji Zhu for their helpful suggestions.

References

Ambroise, C., Chiquet, J., Matias, C., 2009. Inferring sparse Gaussian graphical models with latent structure. Electronic Journal of Statistics 3, 205–238.

Table 1: Results of examples 1–3. The values in the table are averaged over 50 replications and their standard deviations are recorded in the corresponding parentheses.

Example (#modules)	Method	EL	FL	FN (%)	FP (%)
Clique (5)	GLasso	2.43 (0.21)	0.12 (0.02)	4.7 (1.6)	30 (4.1)
	MGGM-GP	2.00 (0.38)	0.11 (0.02)	5.3 (2.5)	3.9 (2.3)
	MGGM-DP	1.88 (0.37)	0.10 (0.02)	5.1 (1.9)	3.3 (2.9)
Nearest-neighbor (2)	GLasso	2.57 (0.28)	0.10 (0.01)	0.8 (0.8)	34.2 (5.0)
	MGGM-GP	1.93 (0.32)	0.08 (0.01)	1.5 (0.8)	5.1 (1.5)
	MGGM-DP	1.91 (0.26)	0.07 (0.01)	1.0 (0.4)	14.1 (3.0)
Nearest-neighbor (7)	GLasso	10.64 (0.61)	0.15 (0.00)	1.2 (0.1)	10.2 (0.6)
	MGGM-GP	9.36 (0.91)	0.13 (0.01)	1.5 (0.2)	1.9 (0.6)
	MGGM-DP	8.90 (0.88)	0.12 (0.01)	1.4 (0.2)	2.0 (0.6)
Scale-free (2)	GLasso	1.74 (0.26)	0.07 (0.01)	0.0 (0.0)	16.7 (1.5)
	MGGM-GP	1.30 (0.28)	0.05 (0.01)	0.2 (0.4)	4.8 (1.3)
	MGGM-DP	1.25 (0.21)	0.04 (0.01)	0.0 (0.0)	7.0 (2.7)
Scale-free (4)	GLasso	8.60 (0.47)	0.08 (0.01)	0.0 (0.0)	14.3 (0.6)
	MGGM-GP	5.51 (0.58)	0.06 (0.01)	0.2 (0.0)	1.3 (0.4)
	MGGM-DP	5.15 (0.50)	0.05 (0.01)	0.0 (0.0)	3.2 (0.3)

Banerjee, O., Ghaoui, L., d'Aspremont, A., 2008. Model selection through sparse maximum likelihood estimation. Journal of Machine Learning Research 9, 485–516.

Barabasi, A.-L., Albert, R., 1999. Emergence of scaling in random networks. Science 286, 509–512.

Duchi, J., Gould, S., Koller, D., 2008. Projected subgradient methods for learning sparse Gaussians. In: Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, Helsinki, Finland.

Fan, J., Feng, Y., Wu, Y., 2009. Network exploration via the adaptive LASSO and SCAD penalties. Annals of Applied Statistics 3, 521–541.

Friedman, J., Hastie, T., Tibshirani, R., 2008. Sparse inverse covariance estimation with the graphical lasso. Biostatistics 9, 432–441.

Guo, J., Levina, E., Michailidis, G., Zhu, J., 2010a. Estimating heterogenous graphical models for discrete data. Tech. rep., Department of Statistics, University of Michigan, Ann Arbor.

Guo, J., Levina, E., Michailidis, G., Zhu, J., 2010b. Joint estimation of multiple graphical models. Biometrika, To appear.

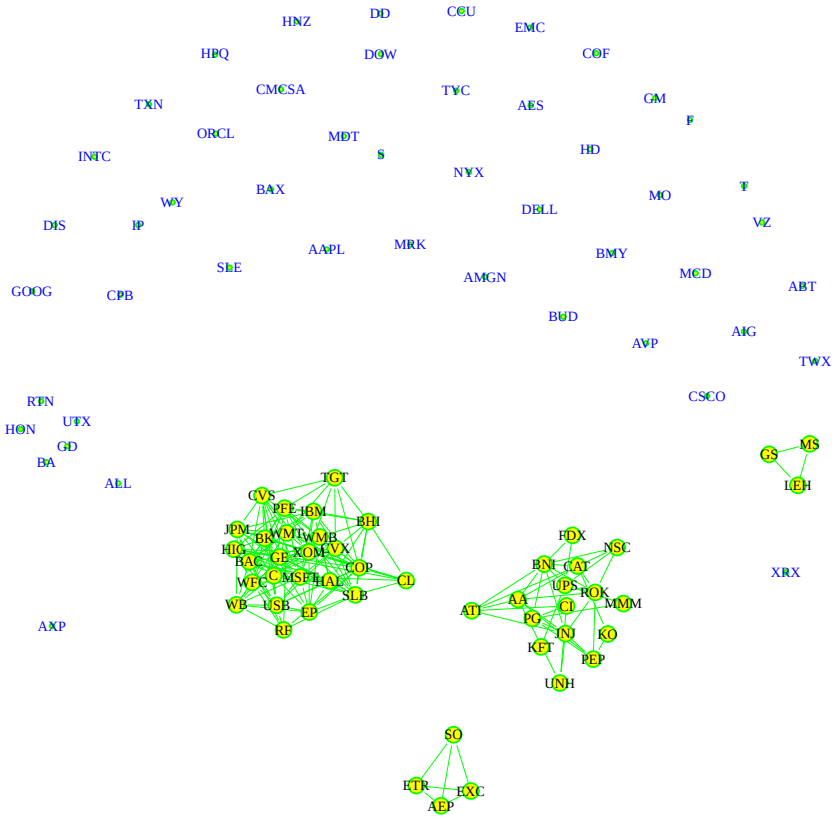

Figure 8: Illustration of the relationships between the S&P 100 companies estimated by MGGM-GP.

Lee, T., Rinaldi, N., Robert, F., Odom, D., Bar-Joseph, Z., Gerber, G., Hannett, N., Harbison, C., Thompson, C., Simon, I., Zeitlinger, J., Jennings, E., Murray, H., Gordon, D., Ren, B., Wyrick, J., Tagne, J., Volkert, T., Fraenkel, E., Gifford, D., Young, R., 2002. Transcriptional regulatory networks in Saccharomyces cerevisiae. Science 298, 799–804.

Li, H., Gui, J., 2006. Gradient directed regularization for sparse Gaussian concentration graphs, with applications to inference of genetic networks. Biostatistics 7, 302–317.

Marlin, B., Murphy, K., 2009. Sparse Gaussian graphical models with unknown block structure. In: Proceedings of the 26th International Conference on Machine Learning, Montreal, Canada.

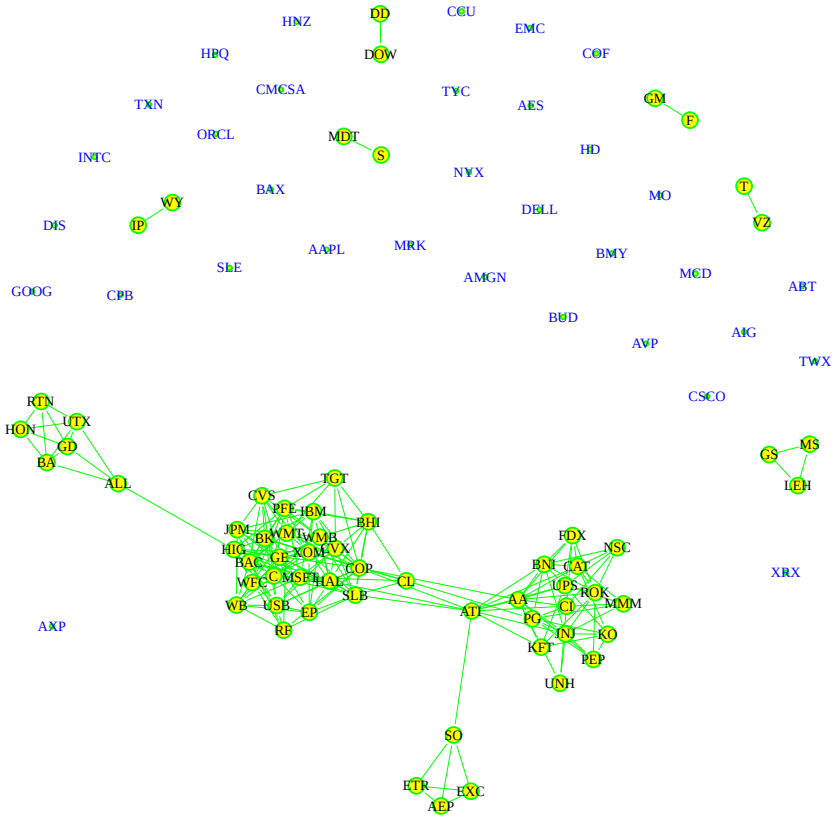

Figure 9: Illustration of the relationships between the S&P 100 companies estimated by MGGM-DP.

Meinshausen, N., Buhlmann, P., 2006. High-dimensional graphs with the lasso. Annals of Statistics 34, 1436–1462.

Newman, M., Girvan, M., 2004. Finding and evaluating community structure in networks. Physics Review E 69, 026113.

Peng, J., Wang, P., Zhou, N., Zhu, J., 2009. Partial correlation estimation by joint sparse regression model. Journal of the American Statistical Asscociation 104, 735–746.

Rocha, G., Zhao, P., Yu, B., 2008. A path following algorithm for Sparse Pseudo-Likelihood Inverse Covariance Estimation (SPLICE). Tech. rep., Department of Statistics, University of California, Berkeley.

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

Schmidt, M., van de Berg, E., Friedlander, M., Murphy, K., 2009. Optimizing costly functions with simple constraints: a limited-memory projected quasi-Newton algorithm. In: Proceedings of the 12th International Conference on Artificial Intelligence and Statistics, Clearwater Beach, Florida, USA.

Wang, S., Zhou, N., Nan, B., Zhu, J., 2009. Hierarchically penalized cox model for survival data with grouped variables and its oracle property. Biometrika 96, 307–322.

Yu, S., Shi, J., 2003. Multiclass spectral clustering. In: In International Conference on Computer Vision. pp. 313–319.

Yuan, M., Lin, Y., 2007. Model selection and estimation in the Gaussian graphical model. Biometrika 94, 19–35.

Zhou, N., Zhu, J., 2007. Group variable selection via a hierarchical lasso and its oracle property. In: Eprint arXiv:1006.2871.

Zou, H., Li, R., 2008. One-step sparse estimates in nonconcave penalized likelihood models. Annals of Statistics 36, 1108–1126.