Efficient Dimensionality Reduction for High-Dimensional Network Estimation

Safiye Celik SAFIYE@CS.WASHINGTON.EDU
Department of Computer Science and Engineering, University of Washington, Seattle, WA 98195

Benjamin A. Logsdon BLOGSDON@CS.WASHINGTON.EDU
Department of Genome Sciences, University of Washington, Seattle, WA 98195

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

Abstract

We propose *module graphical lasso* (MGL), an aggressive dimensionality reduction and network estimation technique for a high-dimensional Gaussian graphical model (GGM). MGL achieves scalability, interpretability and robustness by exploiting the *modularity* property of many real-world networks. Variables are organized into tightly coupled *modules* and a graph structure is estimated to determine the conditional independencies among modules. MGL iteratively learns the module assignment of variables, the *latent variables*, each corresponding to a module, and the parameters of the GGM of the latent variables. In synthetic data experiments, MGL outperforms the standard graphical lasso and three other methods that incorporate latent variables into GGMs. When applied to gene expression data from ovarian cancer, MGL outperforms standard clustering algorithms in identifying functionally coherent gene sets and predicting survival time of patients. The learned modules and their dependencies provide novel insights into cancer biology as well as identifying possible novel drug targets.

1. INTRODUCTION

Probabilistic graphical models provide a powerful framework to represent rich statistical dependencies among random variables, hence their broad application to biology, computer vision and robotics. An edge in a graphical model represents a conditional dependence between the two nodes the edge connects. In a Gaussian graphical model (GGM), edges are parameterized by elements of the inverse covariance matrix (precision matrix). Biologists are increasingly interested in understanding how thousands of genes interact, which has stimulated considerable research into structure estimation of high-dimensional GGM.

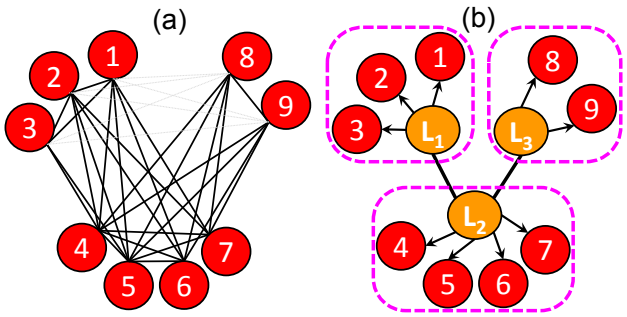


Figure 1. (a): GGM representation of $\mathbf{X} = \{X_1, \ldots, X_9\}$; (b) MGL representation of $\mathbf{X}$

A popular approach to estimating the graph structure of a high-dimensional GGM is the *graphical lasso* (Friedman et al., 2007) that independently penalizes each off-diagonal element of the inverse covariance matrix with an L_1 norm. However, the independence assumption is unrealistic for many real-world networks that are *structured*, where edges are not mutually independent. For example, in *gene regulatory networks*, genes involved in similar functional *modules* are more likely to interact with each other. In addition, there are often high-level interactions between functional modules, which can be difficult to identify in a standard GGM representation (see Fig. 1(a)). Importantly, how genes are organized into functional modules and how these modules interact with each other are scientifically relevant. In this paper, we propose a general framework to accommodate the modular nature of many real-world networks.

Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

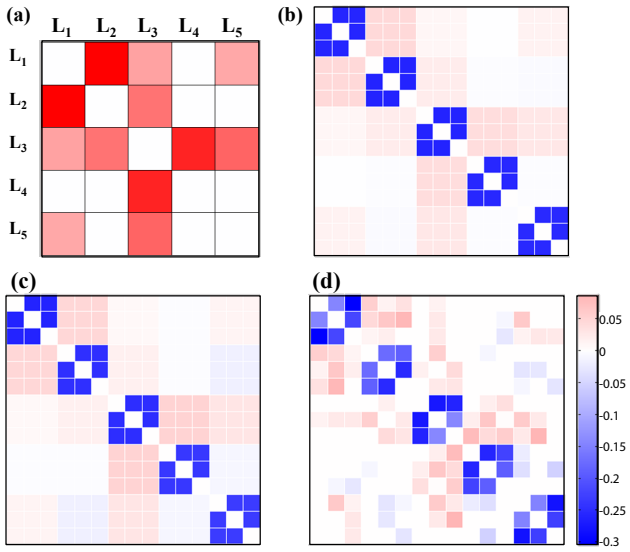


Figure 2. (a): Heatmap of $\mathbf{\Sigma_L^{-1}}$. White elements are zero and colored ones are nonzero; thus, colored elements correspond to edges in the graph. (b): Heatmap of $\mathbf{\Sigma_X^{-1}}$. (c): MGL estimate of $\mathbf{\Sigma_X^{-1}}$. (d): GGM estimate of $\mathbf{\Sigma_X^{-1}}$.

Our approach, called *module graphical lasso* (MGL), is characterized by the incorporation of latent variables into the GGM. Fig. 1(b) illustrates a toy example where three latent variables L_1, L_2 and L_3 have mutual dependencies in addition to connections to observed variables by directed edges. Each of L_1, L_2 and L_3 represents aggregate activity level of specific functional modules as defined by a core of tightly coupled genes. The undirected edges among latent variables determine the dependencies among these functional modules. As can be seen in Fig. 1, MGL provides a more compact representation of the conditional independence relationships compared to the equivalent GGM. By modeling the conditional independence relationships among k latent variables instead of p ($k \ll p$), we show that MGL scales better than standard graphical lasso when $p \gg n$, enabling us to efficiently learn a GGM with thousands of variables. We considered a toy example with 5 latent variables $\mathbf{L}$ and 15 observed variables $\mathbf{X}$ with the inverse covariance matrix of the latent variables ($\mathbf{\Sigma_L^{-1}}$) illustrated in Fig. 2. Given the same data consisting of 100 observations on $\mathbf{X}$, MGL almost perfectly estimates $\mathbf{\Sigma_X^{-1}}$, whereas graphical lasso fails to reveal the latent structure among the 5 groups of variables (Fig. 2).

The rest of the paper is organized as follows. In Sections 2 and 3, we provide the formulation and the learning algorithm for MGL. In Sec. 4, we present the results of our experiments on synthetic data and ovarian cancer gene expression data. We conclude with a discussion in Sec. 5. Derivations of the learning algorithms and proofs are available at http://leelab.cs.washington.edu/projects/MGL.

2. MODULE GRAPHICAL LASSO

2.1. Preliminaries

Assume that we wish to learn the Gaussian graphical model (GGM) with p variables based on n observations $\mathbf{x}[1], \ldots, \mathbf{x}[n] \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \mathbf{\Sigma})$, where $\mathbf{\Sigma}$ is a $p \times p$ covariance matrix. It is well known that the sparsity pattern of $\mathbf{\Sigma}^{-1}$ represents the conditional independence relationships among the variables (Mardia et al., 1979; Lauritzen, 1996). Specifically, $(\mathbf{\Sigma}^{-1})_{jj'} = 0$ for some $j \neq j'$ if and only if X_j and $X_{j'}$ are conditionally independent given X_k with $k = \{1, \ldots, p\} \setminus \{j, j'\}$. Hence, the nonzero pattern of $\mathbf{\Sigma}^{-1}$ corresponds to the graph structure of a GGM. In order to obtain a sparse estimate for the GGM, a number of authors (Yuan & Lin, 2007; Banerjee et al., 2008; Friedman et al., 2007) have considered maximizing the penalized log likelihood, a method called *graphical lasso*:

$$\max_{\mathbf{\Theta} \succ 0} \left\{ \log\det \mathbf{\Theta} - \mathrm{tr}(\mathbf{S\Theta}) - \lambda \sum_{j \neq j'} |\Theta_{jj'}| \right\}, \quad (1)$$

where $\mathbf{S}$ is empirical covariance matrix, λ is a positive tuning parameter, the constraint $\mathbf{\Theta} \succ 0$ restricts the solution to the space of positive definite matrices of size $p \times p$, and the last term is the element-wise L_1 penalty. We denote by $\mathbf{\Theta}$ the estimate of inverse covariance matrix throughout the paper. When λ is large, the resulting estimate will be sparse.

The probabilistic interpretation of the L_1 penalty term defines the optimization parameters $\mathbf{\Theta}$ as *random variables* rather than fixed parameters. This interpretation requires that we optimize the joint probability density:

$$\log P(\mathbf{S}, \mathbf{\Theta}) = \log P(\mathbf{S}|\mathbf{\Theta}) + \log P(\mathbf{\Theta}) \quad (2)$$

The use of the *Laplacian prior* $P(\Theta_{j,j'}) = \lambda/2 \cdot \exp(-\lambda|\Theta_{j,j'}|)$ leads to the optimization problem described in Eq. 1. The hyperparameter λ adjusts the sparsity of the optimization variable $\mathbf{\Theta}$.

2.2. Module Graphical Lasso Formulation

Let $\mathbf{L} = \{L_1, \ldots, L_k\}$ be a set of *latent variables*: $\mathbf{L} \sim N(\mathbf{0}, \mathbf{\Sigma_L})$, where $\mathbf{\Sigma_L}$ is a $k \times k$ covariance matrix. Let $\mathbf{X} = \{X_1, \ldots, X_p\}$ be a set of *observed variables*, each having the distribution: $X_i|L_{Z_i}, \sigma^2_{Z_i} \sim N(L_{Z_i}, \sigma^2_{Z_i})$, where Z_i refers to the index of the latent variable which X_i is associated with. Here, we refer to a set of observed variables that correspond to the same latent variable as a *module*. As an example, the jth module $\mathcal{M}_j$ can be defined as $\mathcal{M}_j = \{X_i|Z_i = j\}$ for $1 \leq j \leq k$. Thus, $\mathbf{Z} = \{Z_1, \ldots, Z_p\}$ defines the module assignment of p variables into k modules.

Then, the joint probability distribution function $P(\mathbf{X}, \mathbf{L}, \mathbf{Z}, \mathbf{\Sigma_L})$ of the MGL has the following form:

$$\begin{aligned} & P(\mathbf{X}, \mathbf{L}, \mathbf{Z}, \mathbf{\Sigma_L}) \qquad (3) \\ &= \prod_{i=1}^{p} P(X_i|L_{Z_i}) P(\mathbf{L}|\mathbf{\Sigma_L}) P({\mathbf{\Sigma_L}}^{-1}) P(\mathbf{Z}) \\ &= \prod_{i=1}^{p} \frac{1}{\sqrt{2\pi\sigma_{Z_i}^2}} \exp\left\{-\frac{(X_i - L_{Z_i})^2}{2\sigma_{Z_i}^2}\right\} \\ &\quad \cdot \frac{1}{\sqrt{(2\pi)^k |\mathbf{\Sigma_L}|}} \exp\left\{-\frac{1}{2}\mathbf{L}^\intercal {\mathbf{\Sigma_L}}^{-1} \mathbf{L}\right\} \\ &\quad \cdot \prod_{j \neq j'} \frac{\lambda}{2} \exp\left\{-\lambda |({\mathbf{\Sigma_L}}^{-1})_{jj'}|\right\} P(\mathbf{Z}). \end{aligned}$$

MGL can be seen as a generalized k-means clustering that takes into account the Mahalanobis distances between latent variables (2nd term in Eq. 3), in addition to the distances between each variable and the corresponding latent variable (1st term in Eq. 3).

Given n observations $\mathbf{x}[1], \ldots, \mathbf{x}[n] \in \mathbb{R}^p$ in $\mathbf{X}$, MGL aims to estimate values on the latent variables $\mathbf{L}$, module assignment variables $\mathbf{Z}$, and the inverse covariance matrix of the latent variables $\mathbf{\Sigma_L^{-1}}$. In order to estimate the inverse covariance matrix over the observed variables, $\mathbf{\Sigma_X^{-1}}$, we can use the relationship between $\mathbf{\Sigma_L^{-1}}$ and $\mathbf{\Sigma_X^{-1}}$, as described in *Lemma* 3.

2.3. Properties of Module Graphical Lasso

Lemma 1. *The joint distribution of* $\mathbf{X} = \{X_1, \ldots, X_p\}$ *and* $\mathbf{L} = \{L_1, \ldots, L_k\}$ *is Gaussian:* $(\mathbf{X}, \mathbf{L}) \sim N(\mathbf{0}, \mathbf{\Sigma_{XL}})$*, where* $\mathbf{\Sigma_{XL}}$ *is a* $(p+k) \times (p+k)$ *covariance matrix.*

Lemma 2. *The marginal probability distribution of the observed variables* $\mathbf{X} = \{X_1, \ldots, X_p\}$ *is Gaussian:* $\mathbf{X} \sim N(\mathbf{0}, \mathbf{\Sigma_X})$*, where* $\mathbf{\Sigma_X}$ *is a* $p \times p$ *covariance matrix.*

Lemma 3. *Let* $\mathbf{\Sigma_L}$ *be a* $k \times k$ *covariance matrix of* $\mathbf{L}$*. The relationship between* $\mathbf{\Sigma_X}$ *and* $\mathbf{\Sigma_L}$ *is as*

$$\mathbf{\Sigma_X} = \left\{\mathbf{A} - \mathbf{C}^\intercal \mathbf{B}^{-1} \mathbf{C}\right\}^{-1}, \qquad (4)$$

where $\mathbf{A}$ *is a* $p \times p$ *diagonal matrix whose element* $A_{ij} = 1/\sigma_{Z_i}^2$ *if* $i = j$ *and 0 otherwise;* $\mathbf{C}$ *is a* $k \times p$ *matrix whose element* $C_{ij} = -(1/\sigma_i^2)$ *if* $X_j \in \mathcal{M}_i$ *and 0 otherwise; and*

$$\mathbf{B} = {\mathbf{\Sigma_L}}^{-1} + \begin{bmatrix} |\mathcal{M}_1|/\sigma_1^2 & \ldots & 0 \\ \vdots & \vdots & \vdots \\ 0 & \ldots & |\mathcal{M}_k|/\sigma_k^2 \end{bmatrix}$$

$|\mathcal{M}_k|$ *meaning the number of* X *variables in the module* k*.*

2.4. Related Work

MGL jointly clusters variables into modules and learns a network among the modules through an iterative procedure. This key aspect differentiates MGL from previous approaches that can be organized into four categories:

The first category includes latent factor models, such as latent factor analysis or probabilistic PCA (Tipping & Bishop, 1999), which do not learn the network among latent factors.

Second, Toh & Horimoto (2002) clusters variables first and then learns the dependency structure among the cluster centroids, instead of jointly clustering and learning the network. This method can achieve improved scalability and interpretability; however, we showed through our extensive experiments that MGL outperforms this approach based on all of the evaluation criteria we incorporated.

Third, He et al. (2012) models each latent variable as a linear combination of variables and estimates the network among k latent variables. Although this approach also learns a network of k latent variables instead of p variables, it does not explicitly cluster variables, which results in a vastly different learning algorithm from MGL. Clustering of variables is a key feature of MGL reducing the number of parameters and increasing the model's interpretability, which enables interesting analyses shown in Sec. 4.2.2.

Finally, many authors attempted to incorporate latent variables into GGMs. However, they do not explicitly cluster variables into modules, and require the learning of Σ^{-1} of p variables instead of k latent variables ($k \ll p$), which drastically increase the number of parameters. Chandrasekaran et al. (2012) assume that Σ^{-1} of observed variables decomposes into a sparse matrix and a low-rank matrix, and the low-rank matrix represents the effect of unobserved latent variables. They proposed a convex optimization algorithm that utilizes both L_1 and nuclear norm as penalty terms. The SIMoNe (Ambroise et al., 2009) uses an Expectation-Maximization approach (Dempster et al., 1977) for variational estimation of the latent structure while inferring the network among the entire variables. In contrast, MGL performs a more aggressive dimensionality reduction by learning a network of k latent variables instead of p observed variables ($k << p$). Guo & Wang (2010) proposed an algorithm consisting of three steps: 1) apply the graphical lasso to compute an adjacency matrix of the variables; 2) partition variables into disjoint clusters; and 3) estimate a sparse Σ^{-1} with a modified penalty term such that within-cluster edges are less strongly penalized. Given the module assignment of variables, Duchi et al. (2008) proposed to penalize the infinity-norm and Schmidt et al. (2009) proposed to penalize the two-norm of the inverse covariance matrix block corresponding to each module in the network

of the variables. Marlin et al. (2009) and Marlin & Murphy (2009) make use of these methods (Duchi et al., 2008; Schmidt et al., 2009), after first identifying the groups of the variables when the modular structure is unknown.

3. LEARNING ALGORITHM

3.1. Overview

Here, we present our learning algorithm that optimizes the likelihood function based on the joint distribution described in Eq. 3. Given $X(\in \mathbb{R}^{p\times n})$ that contains n observations $\mathbf{x}[1], \ldots, \mathbf{x}[n] \in \mathbb{R}^p$ on $\mathbf{X}$, MGL aims to learn the following:

- L $(\in \mathbb{R}^{k\times n})$ containing the values on $\mathbf{L}$ in the n observations $\mathbf{l}[1], \ldots, \mathbf{l}[n] \in \mathbb{R}^k$;
- Z $(\in \{1, \ldots, k\}^p)$ specifying the module assignment of $X_1, \ldots, X_p$ into k modules; and
- Θ_L $(\in \mathbb{R}^{k\times k})$ denoting the estimate of the inverse covariance matrix $\mathbf{\Sigma_L^{-1}}$. Using the *Lemma* 3, we can obtain Θ_X $(\in \mathbb{R}^{p\times p})$, the precision matrix estimate of $\mathbf{X}$.

We choose to address our learning problem by finding the joint *maximum a posteriori (MAP)* assignment to all of the optimization variables – L, Z, and Θ_L. This means that we optimize the following objective function with respect to L, Z, and Θ_L $(\succ 0)$:

$$\begin{aligned}
&\log P(X, L, Z, \Theta_L; \lambda, \sigma) \qquad (5)\\
&= \log P(\Theta_L) + \log P(L|\Theta_L)\\
&\quad + \log P(X|L, Z) + \log P(Z)\\
&= \frac{n}{2}(\log\det \Theta_L - \mathrm{tr}(S_L\Theta_L))\\
&\quad -\lambda \sum_{j\neq j'} |(\Theta_L)_{jj'}| - \sum_{i=1}^{p} \frac{\| X_i - L_{Z_i} \|_2^2}{2\sigma_{Z_i}^2},
\end{aligned}$$

where $S_L = \frac{1}{n} LL^\intercal$ is the empirical estimate of the covariance matrix of L, X_i denotes the ith row of the matrix X, L_i denotes the ith row of the matrix L, and λ is a positive tuning parameter that adjusts the sparsity of Θ_L.

Throughout this paper, we choose hard assignment of variables to modules to reduce the number of parameters and to increase each module's biological interpretability, where interpretability is a key MGL design feature. Soft assignment is a straightforward extension. We also assume a uniform prior distribution over $\mathbf{Z}$.

We use a *coordinate ascent procedure* over the three sets of optimization variables – L, Z, and Θ_L. We iteratively estimate each of the optimization variables until convergence. Since our objective is continuous on a compact level set, based on Thm. 4.1 in Tseng (2001), the solution sequence from MGL is defined and bounded; every coordinate group reached by the iterates is a stationary point of the MGL objective function. And we observed that the value of the objective likelihood function monotonically increases.

3.2. Iterative estimation of L, Z and Θ_L

3.2.1. Estimation of L

To estimate L given Z and Θ_L, from Eq. 5, we solve the following problem:

$$\max_{L_1,\ldots,L_k} \left\{ -\mathrm{tr}\left(LL^\intercal \Theta_L\right) - \sum_{i=1}^{p} \frac{\| X_i - L_{Z_i} \|_2^2}{\sigma_{Z_i}^2} \right\}. \qquad (6)$$

Setting the derivative of the objective function in Eq. 6 to zero with respect to L_m leads to:

$$L_m = \frac{\sum_{X_i\in\mathcal{M}_m} X_i - \sigma_m^2 \sum_{i\neq m} (\Theta_L)_{im} L_i}{|\mathcal{M}_m| + \sigma_m^2 (\Theta_L)_{mm}}, \qquad (7)$$

where $\mathcal{M}_m$ means a set of X_i that belongs to the mth module: $\mathcal{M}_m = \{X_i | Z_i = m\}$, and $|\mathcal{M}_m|$ means the number of variables that belong to $\mathcal{M}_m$. We update L_m for each m $(1 \leq m \leq k)$, based on the current values of the other latent variables.

If all elements in Θ_L are equal to zero, L_m would be set to the centroid of the mth module. This leads to a nice interpretation of the MGL learning algorithm with respect to the k-*means* clustering. *The k-means clustering algorithm is the special case of the MGL when no network structure is assumed to exist among the latent variables (cluster centroids). More specifically, the MGL is a generalization of the k-means with the distance metric determined by the sparse estimate of the latent structure (Θ_L).*

3.2.2. Estimation of Z

In order to estimate Z given L and Θ_L, we solve the following:

$$\max_{Z_1,\ldots,Z_p} \left\{ -\sum_{i=1}^{p} \frac{\| X_i - L_{Z_i} \|_2^2}{\sigma_{Z_i}^2} \right\}, \qquad (8)$$

which, when $\sigma_1, \ldots, \sigma_k = 1$, finds the module for X_i that minimizes the Euclidean distance between X_i and the latent variable.

3.2.3. Estimation of Θ_L

To estimate Θ_L given L and Z, we solve the following optimization problem:

$$\max_{\Theta_L\succ 0} \left\{ \log\det \Theta_L - \mathrm{tr}\left(S_L\Theta_L\right) - \lambda \sum_{j\neq j'} |(\Theta_L)_{jj'}| \right\}, \qquad (9)$$

where $S_L = \frac{1}{n} LL^\intercal$ is the empirical estimate of the covariance matrix of L. Since L is given, the optimization problem in Eq. 9 can be solved by the standard graphical lasso algorithm applied to L.

4. EXPERIMENTAL RESULTS

We present our results on synthetic data (Sec. 4.1) and ovarian cancer gene expression data (Sec. 4.2).

4.1. Synthetic Data

We compared MGL algorithm with four other methods in terms of the performance of learning networks with latent variables: 1) the standard graphical lasso (Glasso) (Friedman et al., 2007), 2) the method proposed by Toh & Horimoto (2002) that clusters the variables and learns the network of cluster centroids (Toh), 3) the SIMoNe method proposed by Ambroise et al. (2009), and 4) the regularized maximum likelihood decomposition (RMLD) method proposed by Chandrasekaran et al. (2012).

For Glasso, we used CRAN R package *QUIC* (Hsieh et al., 2011); for SIMoNe, we used CRAN R package *simone*; and for RMLD, we used LogdetPPA (Wang et al., 2010), a MATLAB software for log-determinant SDP. We implemented MGL in C, and we used the C source code of CRAN R package *huge* (Zhao et al., 2012) to estimate the inverse covariance matrix of the latent variables (Sec. 3.2.3).

Toh & Horimoto (2002) originally uses hierarchical clustering for grouping the variables. In our interpretation of Toh, we used k-means algorithm for clustering the variables due to k-means' better cluster quality and scalability that we observed for high-dimensional data. Also, we used Glasso to learn the network of cluster centroids for Toh. So, throughout this paper, the method we refer by Toh is k-means followed by Glasso. In terms of module assignments and latent variables, k-means and Toh are identical.

We used $\sigma_1, \ldots, \sigma_k = 1$ for MGL throughout Sec. 4, such that we evaluate MGL in the simplest and efficient setting. When $\sigma_1, \ldots, \sigma_k = 1$, Eq. 8 is equal to the Euclidean distance objective of the k-means clustering algorithm, which we use as the first step for Toh.

4.1.1. Data Generation

We synthetically generated data based on the joint distribution described in Eq. 3. We first generate the inverse covariance matrix ${\boldsymbol{\Sigma}_\mathbf{L}}^{-1}$ by creating $\mathbf{A}$ as $A_{ii} = 0.5$ and

$$A_{ij}\ (i \neq j) \overset{\text{i.i.d.}}{\sim} \begin{cases} 0 & \text{w. prb. } 1-(b-a) \\ \text{Unif}([a,b]) & \text{w. prb. } b-a \end{cases}, \tag{10}$$

and setting ${\boldsymbol{\Sigma}_\mathbf{L}}^{-1} = \mathbf{A} + \mathbf{A}^\intercal$. We arranged the parameters a and b such that the resulting matrix ${\boldsymbol{\Sigma}_\mathbf{L}}^{-1}$ is positive definite. If it is still not positive definite, which happened only rarely, we regenerated the matrix $\mathbf{A}$. Then, we used *Lemma* 3 to generate $\boldsymbol{\Sigma}_\mathbf{X}$ based on $\boldsymbol{\Sigma}_\mathbf{L}$ and σ. We generated the data for $\mathbf{X}$ according to $\mathbf{x}_1, \ldots, \mathbf{x}_n \overset{\text{i.i.d.}}{\sim} N(\mathbf{0}, \boldsymbol{\Sigma}_\mathbf{X})$, which results in $X \in \mathbb{R}^{p \times n}$.

In order to evaluate these algorithms in varying degrees of high-dimensionality, we created three settings in terms of (P, K, N), where P is the number of variables, K is the number of latent variables, and N is the sample size.

Setting I - (100, 10, 10)

Setting II - (150, 10, 10): The difference from Setting I is the number of variables P, which increases the dimensionality of the data by 1.5 times.

Setting III - (150, 15, 10): We increased the number of latent variables K such that the sample size N is smaller than K.

By setting $a = 0.2$ and $b = 0.6$ in Eq. 10, we created two different data matrices (training and test datasets) in each of Settings I, II and III. The sparsity (i.e., ratio of the number of nonzero edges to the number of all potential edges) of the resulting data matrices was around 35%. We used one of the two data matrices for training MGL and its competitors, and the other one for testing.

4.1.2. Synthetic Test Log-Likelihoods

We measure the performance of MGL and four competing methods in terms of test log-likelihood using the training/test datasets described above. We chose cross-data test log-likelihood as an evaluation metric because it allows direct comparisons between methods that incorporate latent variables and methods that do not (given that each method estimates a p-dimensional precision matrix). Test log-likelihood allows us to evaluate how well the learned models fit unseen data.

We performed 5-fold cross validation tests within the training dataset in order to select λ that gives the best average test log-likelihood for each method. In this cross-validation for choosing λ, we used a wide range of the λ values such that the solutions for the inverse covariance matrices range from a full matrix to an empty matrix.

Fig. 3 shows the difference of the test log-likelihood between each method and the SIMoNe method in Settings I, II and III. For MGL and Toh, we present the results for 3 different k values representing the number of latent variables – $K/2$, K and $2K$, where K means the true number of modules, assuming that the true number of modules (K) is unknown by the methods.

It can be seen that MGL outperforms all of its competitors in all of the three simulation settings we considered. Although SIMoNe and RMLD are specific generalizations of Glasso, Ambroise et al. (2009) showed that Glasso outperforms SIMoNe when $p = n$ and $p = 2n$, and Giraud & Tsybakov (2012) argued that RMLD results are valid and meaningful only when $p < n$, consistently with our results.

We note that in Fig. 3a and Fig. 3c, the test log-likelihood was maximized when we used more latent variables than in the generating model. This is a result of the high-dimensionality of the data. But when we increased k further, test log-likelihood of MGL and Toh decreased, and for $k = p$, they both became equal to the one of Glasso as expected.

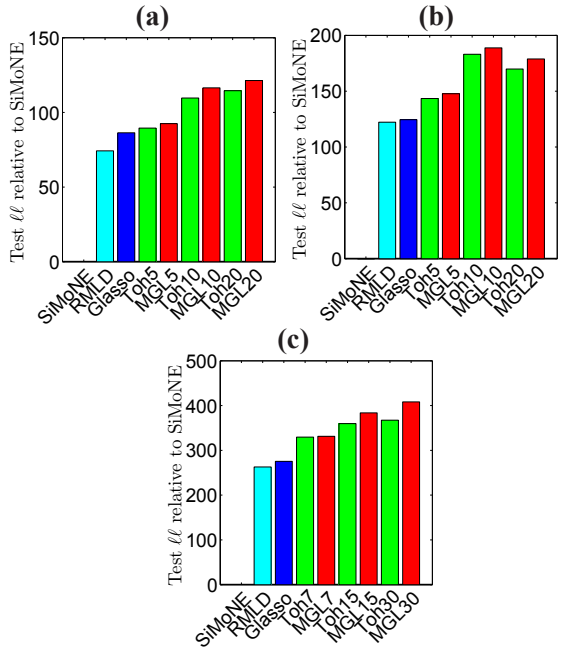


Figure 3. For $(P,\ K,\ N)$ (a) (100, 10, 10), (b) (150, 10, 10), and (c) (150, 15, 10), we considered SIMoNe as reference and computed the difference in cross-data test log-likelihood of each method compared to the one of SIMoNe (y-axis). Each bar corresponds to (1) SIMoNe, (2) RMLD (3) Glasso, (4) Toh for $k = K/2$, (5) MGL for $k = K/2$, (6) Toh for $k = K$, (7) MGL for $k = K$, (8) Toh for $k = 2K$, and (9) MGL for $k = 2K$.

4.2. Cancer Gene Expression Data

Ovarian cancer is the 5th leading cause of cancer death among US women and has a 5-year survival rate of 30% (Bast et al., 2009). Learning the gene regulatory network from expression data is an effective strategy to identify novel disease mechanisms (Akavia et al., 2010; TCGA, 2012). Thus, we experimented MGL on three gene expression datasets containing 10404 gene expression levels in a total of 909 patients with ovarian serous carcinoma – Tothill (269 samples) (Tothill et al., 2008), TCGA (560 samples) (TCGA, 2012), and Denkert (80 samples) (Denkert et al., 2009). We mainly used Tothill for training, and TCGA and Denkert for testing.

Given the data, MGL estimates Z, L and Θ_L (see Eq. 5), which describe a gene module network characterized by the assignments of genes to modules and the latent structure among the modules (Fig. 1(b)). We evaluated MGL based on: 1) how well the learned model fits unseen data (Sec. 4.2.1); 2) how significantly the inferred modules are coherent in terms of gene functions (Sec. 4.2.2); and 3) how well the inferred latent variables are predictive of survival time (Sec. 4.2.3). We also present some of the biologically interesting findings that we obtain from the MGL results (Sec. 4.2.4).

Since this application requires learning a network with >10K variables, the methods that attempt to learn the network of all individual variables do not scale. Therefore, we compared MGL with only Toh that first clusters the variables and then learns the network of cluster centroids.

4.2.1. Cross-validation test log-likelihoods

We applied k-means clustering and used the resulting clusters as a starting point for MGL and Toh. We compared between MGL and Toh in terms of the cross-validation (CV) test log-likelihood of the estimated p-dimensional precision matrices. We performed model selection using Bayesian Information Criterion (BIC) for k-means. Cluster count (k) was determined as 150 by BIC. Since the data is high-dimensional, we performed 2-fold CV. We used a wide range of the λ values such that the solutions for the module precision matrices range from a full matrix to an empty matrix. The results were averaged over 10 iterations due to non-deterministic nature of the k-means. Fig. 4 shows the test log-likelihoods of each method. MGL clearly outperforms Toh, meaning that the learned model by MGL fits unseen data better than the one by Toh. Moreover, the standard deviation of the test log-likelihoods of the folds is smaller for MGL than Toh, indicating the robustness of MGL. In the subsequent sets of experiments (Sections 4.2.2 and 4.2.3), we use $k = 150$ (as determined by BIC) and $\lambda = .004$ (as chosen by CV).

4.2.2. Functional enrichment of modules

Genes assigned to the same module are likely to share similar functions, and those in the connected modules are likely to be involved in similar cellular processes as well. We define a *super-module* (or a *super-cluster*) as the set of genes in two connected modules (or clusters). We compared super-clusters from the learned network by Toh to super-modules from the learned network by MGL in terms of functional coherence. For each of the 4722 Curated

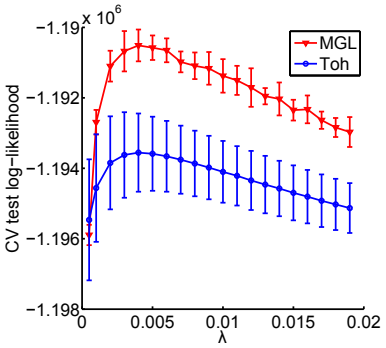


Figure 4. Comparison between MGL and Toh in terms of cross-validation test log-likelihood for varying λ values and for $k = 150$ (which was determined by BIC). Standard deviation between the test log-likelihoods of the folds are shown by the error bars.

GeneSets from the Molecular Signatures Database (Liberzon et al., 2011), we computed the significance of the overlap between the GeneSet and super-modules (or super-clusters). We applied Bonferroni correction to the p-values and only considered the GeneSets with $p < 0.05$ in either MGL or Toh. We repeated this process 50 times with different random initial points for k-means. As can be seen in Fig. 5(a), for each of 50 runs, there are a larger number of GeneSets that are more significantly overlapped with MGL super-modules than with Toh super-clusters. Thus, MGL improves the initial network of Toh, resulting in far more shared processes between modules that are connected in the estimated network. Additionally, we observed that in each independent run, MGL improves the actual p-values. In Fig. 5(b), for each of 4722 functional categories, the smallest p-value achieved by MGL super-modules is plotted (y-axis) against that achieved by Toh super-clusters (x-axis). The results for all 50 runs were aggregated in Fig. 5(b). Most of the dots in Fig. 5(b) lie above the diagonal, meaning that for most of the functional categories, MGL super-modules achieve better enrichment than Toh super-clusters. Moreover, 6 GeneSets were observed to be enriched by MGL super-modules with p-values not only smaller than 10^{-90}, but also smaller than the best p-values for Toh super-clusters (10^{-20}). These GeneSets were related to cell differentiation and increased cell growth, which are core processes relevant to cancer progression. This shows MGL's power to detect core cancer modules. We also performed the experiment explained above for learned modules without considering the latent network among them, and observed that the functional enrichment results for modules were consistent with the ones for super-modules. As can be seen in Fig. 6(a), for each of 50 runs, there are a larger number of GeneSets that are more significantly overlapped with MGL modules than with Toh clusters. An additional interesting observation is that MGL learns much sparser module networks than Toh for any attempted λ value in a wide range. For $\lambda = .004$ and $k = 150$, the average number of the edges was 6324.7 for the Toh network, and was 4626.6 for the MGL network. MGL removes a handful of dependencies from the initial Toh network and adds a number of new dependencies while improving the module assignments meanwhile. Sparsity of MGL networks is plausible in terms of genetic robustness. We compared the enrichment of module pairs whose dependencies are removed by MGL to that of module pairs between which new dependencies are added. Interestingly, the former was smaller than the latter for 49 runs out of 50 as displayed in Fig. 6(b).

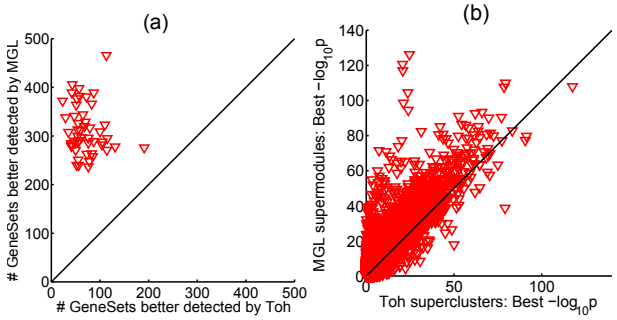


Figure 5. (a) Each dot represents a run with a random k-means starting point. For each of 50 runs, the number of GeneSets more significantly overlapped with MGL super-modules (y-axis) is compared to that with Toh super-clusters (x-axis). (b) Each dot represents a GeneSet. For each GeneSet, the smallest enrichment p-value achieved by Toh (x-axis) vs. MGL (y-axis) is compared. We note that the plot for each run was consistent with the aggregated plot.

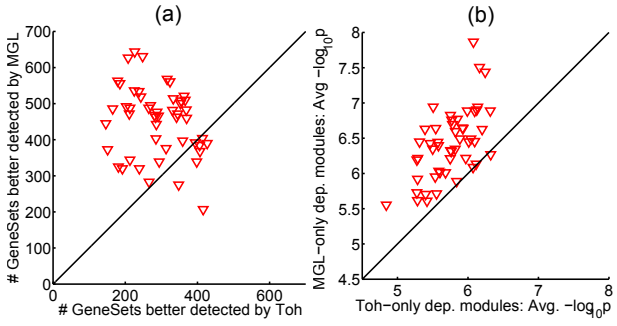


Figure 6. Each dot represents a run with a random k-means starting point. (a) For each of 50 runs, the number of GeneSets more significantly overlapped with MGL modules (y-axis) is compared to that with Toh clusters (x-axis).(b) For each of 50 runs, the average enrichment p-value for Toh-only dependent modules (x-axis) is compared to MGL-only dependent modules (y-axis).

4.2.3. Survival Prediction Using Latent Variables as Features

The latent variables could represent activity levels of pathways relevant to the disease process and clinical outcomes.

We evaluated how well the inferred latent variables learned from Tothill are predictive of survival time of ovarian cancer patients in TCGA and Denkert datasets. After learning Toh and MGL on Tothill dataset, we trained the Cox regression model using the inferred latent variables as features in Tothill dataset, and then tested the model on a separate test dataset. In the test dataset, we computed the concordance index (c-Index) which is considered a standard evaluation metric estimating the accuracy of survival prediction based on the 'censored' survival data. Fig. 7 shows the c-Index achieved for varying sparsity levels for the Cox regression model (x-axis) on the two training-test dataset pairs. It compares latent variables (modules) from MGL to clusters from Toh and individual genes. In both of the settings, the c-Index values for modules are larger than those for Toh clusters or individual genes, for a wide range of sparsity levels. The maximum c-Index for modules is also higher than those for clusters and individual genes. The c-Index values were averaged over 50 runs for MGL and Toh.

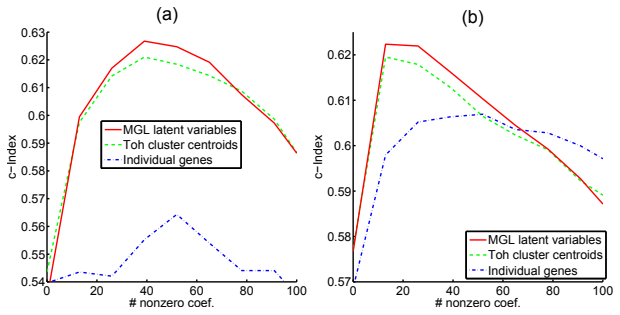
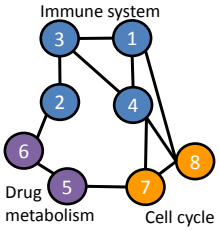


Figure 7. Comparison among MGL latent variables (modules), k-means clusters and individual genes based on survival prediction performance. x-axis gives the number of nonzero coefficients selected by penalized Cox regression model and y-axis gives c-Index values. Two pairs of training-test data are considered: (a) Tothill-Denkert, (b) Tothill-TCGA.

4.2.4. Interesting Findings

A handful of modules identified by MGL are enriched for processes relevant to tumor biology, drug metabolism, and response to drug therapy. Fig. 8 shows a small portion of the module network learned on Tothill dataset. It is a network among immune system, cell cycle and drug metabolism processes. Edges between modules 1 through 4 indicate conditional dependencies among cytokines, inflammation, and immune signaling, which play important roles in tumor biology (Coussens & Werb, 2002). There are suggestive edges between module 4 and modules 7 and 8 (cell cycle modules), since innate immune response can stimulate cell division in neoplastic cells. (Coussens & Werb, 2002). Finally, module 5 is significantly enriched for PDGF for signaling. PDGF receptor agonists, such as the popular drug Gleevec, have succeeded in treating chronic myelogenous leukemia patients (Pietras et al., 2003).



Mod	Detected Pathway	Mod	Detected Pathway
1	Cytokine receptor interaction/chemo kine signaling	5	Signaling by PDGF
2	Inflammatory response	6	Drug metabolism
3	Natural killer cell	7	Mitotic cell cycle
4	Immune system signaling	8	Oocyte meiosis and cell cycle

Figure 8. As small portion of the pathway structure identified by MGL on Tothill dataset.

5. DISCUSSION

We proposed the *module graphical lasso*, a novel high-dimensional GGM representation of conditional independencies among tightly coupled sets of variables (*modules*). The MGL algorithm is a novel high-dimensional clustering algorithm that is a generalization of k-means clustering, with Mahalanobis distances between variables. The full joint probability distribution function Eq. 3 defines a non-Euclidean distance metric between the latent variables $\mathbf{L}$ based on Θ_L.

There are several possible extensions. First, MGL could be extended to other graphical models, such as Markov random fields, with novel distance metrics and clustering properties. Second, the assumptions about relationships between latent and observed variables could be relaxed. For instance, we could apply soft assignments of variables to modules, and learn sub-networks within modules. Third, we could add learning of module variances (σ) as an inference step to the MGL algorithm. And finally, we plan to apply MGL to gene expression data across multiple healthy and cancerous tissues to identify conserved and differential latent molecular networks driving tumor biology.

Acknowledgments

The authors acknowledge funding from the following sources: American Association of Univ. Women International Doctoral Fellowship to SC, NIH T32 HL 007312 to BAL, and Univ. Washington Royalty Research Fund to SL.

References

Akavia, U.D., Litvin, O., Kim, J., Sanchez-Garcia, F., Kotliar, D., Causton, H.C., Pochanard, P., Mozes, E, Garraway, L.A., and Pe'er, D. An integrated approach to uncover drivers of cancer. *Cell*, 143(6):1005–17, 2010.

Ambroise, C., Chiquet, J., and Matias, C. Inferring sparse gaussian graphical models with latent structure. *Electron. J. Statist.*, 3:205–238, 2009.

Banerjee, O., El Ghaoui, L., and d'Aspremont, A. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *JMLR*, 9:485–516, 2008.

Bast, R.C., Hennessy, B., and Mills, G.B. The biology of ovarian cancer: new opportunities for translation. *Nature Reviews Cancer*, 9(6):415–428, 2009.

Chandrasekaran, V., Parrilo, P.A., and Willsky, A.S. Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40:1935–1967, 2012.

Coussens, L.M. and Werb, Z. Inflammation and cancer. *Nature*, 420(6917):860–867, 2002.

Dempster, A.P., Laird, N.M., and Rubin, D.B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1): 1–38, 1977.

Denkert, C., J., Budczies, S., Darb-Esfahani, Gyrffy, B., Sehouli, J., Knsgen, D., Zeillinger, R., Weichert, W., Noske, A., Buckendahl, A.C., Mller, B.M., Dietel, M., and Lage, H. A prognostic gene expression index in ovarian cancer - validation across different independent data sets. *J. Pathol.*, 218(2):273–80, 2009.

Duchi, J., Gould, S., and Koller, D. Projected subgradient methods for learning sparse gaussians. *UAI*, 2008.

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

Giraud, C. and Tsybakov, A.S. Discussion: Latent variable graphical model selection via convex optimization. *The Annals of Statistics*, 40(4):1984–1988, 2012.

Guo, J. and Wang, S. Modularized gaussian graphical model. *submitted to Computational Statistics and Data Analysis*, 2010.

He, Y., Kavukcuoglu, K., Qi, Y., and Park, H. Structured latent factor analysis. *NIPS*, 2012.

Hsieh, Cho-Jui, Sustik, Mátyás A., Dhillon, Inderjit S., and Ravikumar, Pradeep K. Sparse inverse covariance matrix estimation using quadratic approximation. *NIPS*, 2011.

Lauritzen, S.L. *Graphical Models*. Oxford Science Publications, 1996.

Liberzon, A., Subramanian, A., Pinchback, R., Thorvaldsdottir, H., Tamayo, P., and Mesirov, J.P. Molecular signatures database (msigdb) 3.0. *Bioinformatics*, 27(12): 1739–1740, 2011.

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

Marlin, B.M. and Murphy, K. Sparse gaussian graphical models with unknown block structure. *ICML*, 2009.

Marlin, B.M., Schmidt, M., and Murphy, K. Group sparse priors for covariance estimation. *UAI*, 2009.

Pietras, K., Sjoblom, T., Rubin, K., Heldin, C.H., and Ostman, A. Pdgf receptors as cancer drug targets. *Cancer cell*, 3:439–444, 2003.

Schmidt, M., van den Berg, E., Friedlander, M.P., and Murphy, K. Optimizing costly functions with simple constraints: A limited-memory projected quasi-newton algorithm. *AISTATS*, 2009.

TCGA, Cancer Genome Atlas Research Network. Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609–15, 2012.

Tipping, M.E. and Bishop, C.M. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61(3):611–622, 1999.

Toh, H. and Horimoto, K. Inference of a genetic network by a combined approach of cluster analysis and graphical gaussian modeling. *Bioinformatics*, 18(2):287–297, 2002.

Tothill, R.W., Tinker, A.V., George, J., Brown, R., Fox, S.B., Lade, S., Johnson, D.S., Trivett, M.K., Etemadmoghadam, D., Locandro, B., Traficante, N., Fereday, S., Hung, J.A., Chiew, Y.E., Haviv, I., Group, Australian Ovarian Cancer Study, Gertig, D., DeFazio, A., and Bowtell, D.D. Novel molecular subtypes of serous and endometrioid ovarian cancer linked to clinical outcome. *Clin. Cancer Res.*, 14(16):5198–208, 2008.

Tseng, P. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications*, 109(3):475494, 2001.

Wang, Chengjing, Sun, Defeng, and Toh, Kim-Chuan. Solving log-determinant optimization problems by a newton-cg primal proximal point algorithm. *SIAM J. Optimization*, (20):2994–3013, 2010.

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

Zhao, Tuo, Liu, Han, Roeder, Kathryn, Lafferty, John, and Wasserman, Larry. The huge package for high-dimensional undirected graph estimation in r. *JMLR*, (13):1059–1062, 2012.