The Joint Graphical Lasso for Inverse Covariance Estimation Across Multiple Classes

P. Danaher, P. Wang & D. Witten

Tavis Abrahamsen
Syed Rahman

Department of Statistics
University of Florida

April 16, 2015

Background

Suppose that observations $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^P$ are independent and identically distributed $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}$ is a positive definite $p \times p$ matrix.

Background

Suppose that observations $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^P$ are independent and identically distributed $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}$ is a positive definite $p \times p$ matrix.

The zeros in the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ correspond to pairs of features that are conditionally independent - that is, pairs of variables that are independent of each other, given all the other variables in the data set.

Background

Suppose that observations $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n \in \mathbb{R}^P$ are independent and identically distributed $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ where $\boldsymbol{\mu} \in \mathbb{R}^p$ and $\boldsymbol{\Sigma}$ is a positive definite $p \times p$ matrix.

The zeros in the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ correspond to pairs of features that are conditionally independent - that is, pairs of variables that are independent of each other, given all the other variables in the data set.

In a Gaussian graphical model, the conditional dependence relationships are represented by a graph in which nodes represent the features and edges connect conditionally dependent pairs of features.

Background

A natural way to estimate the precision (or concentration) matrix $\mathbf{\Sigma}^{-1}$ is via maximum likelihood. Letting $\boldsymbol{S}$ denote the empirical covariance matrix of X, the Gaussian log likelihood takes the form (up to a constant)

$$\frac{n}{2} \log \det \mathbf{\Sigma}^{-1} - \text{trace}\left(\boldsymbol{S}\, \mathbf{\Sigma}^{-1}\right)$$

Background

A natural way to estimate the precision (or concentration) matrix $\mathbf{\Sigma}^{-1}$ is via maximum likelihood. Letting $\boldsymbol{S}$ denote the empirical covariance matrix of X, the Gaussian log likelihood takes the form (up to a constant)

$$\frac{n}{2} \log \det \mathbf{\Sigma}^{-1} - \text{trace}\left(\boldsymbol{S}\, \mathbf{\Sigma}^{-1}\right)$$

Maximizing this function with respect to $\mathbf{\Sigma}^{-1}$ yields the maximum likelihood estimate S^{-1}.

Background

A natural way to estimate the precision (or concentration) matrix $\boldsymbol{\Sigma}^{-1}$ is via maximum likelihood. Letting $\boldsymbol{S}$ denote the empirical covariance matrix of X, the Gaussian log likelihood takes the form (up to a constant)

$$\frac{n}{2} \log \det \boldsymbol{\Sigma}^{-1} - \text{trace}\left(\boldsymbol{S}\,\boldsymbol{\Sigma}^{-1}\right)$$

Maximizing this function with respect to $\boldsymbol{\Sigma}^{-1}$ yields the maximum likelihood estimate S^{-1}.

Problems with using the MLE of $\boldsymbol{\Sigma}^{-1}$:

1. In the high dimensional setting where $p > n$ the matrix $\boldsymbol{S}$ is singular and cannot be inverted to yield an estimate of $\boldsymbol{\Sigma}^{-1}$.

2. Even when $\boldsymbol{S}^{-1}$ exists, this estimator will typically not be sparse.

Background

One method for obtaining sparse estimates of $\boldsymbol{\Sigma}^{-1}$ is to instead solve the *graphical lasso* problem

$$\max_{\boldsymbol{\Theta}} \log \det \boldsymbol{\Theta} - \text{trace}\,(\boldsymbol{S}\,\boldsymbol{\Theta}) - \lambda ||\boldsymbol{\Theta}||_1,$$

where λ is a nonnegative tuning parameter.

Background

One method for obtaining sparse estimates of $\mathbf{\Sigma}^{-1}$ is to instead solve the *graphical lasso* problem

$$\max_{\mathbf{\Theta}} \log \det \mathbf{\Theta} - \text{trace}\,(\boldsymbol{S}\,\mathbf{\Theta}) - \lambda||\mathbf{\Theta}||_1,$$

where λ is a nonnegative tuning parameter.

Advantages of using the graphical lasso estimator:

1. The l_1 penalty term yields sparse estimates of $\mathbf{\Sigma^{-1}}$ when λ is "large".

2. This problem can be solved even if $p \gg n$.

Motivation - Genetics

Graphical models are of particular interest in the analysis of gene expression data since they can provide a useful tool for visualizing the relationships among genes and generating biological hypotheses.

Motivation - Genetics

Graphical models are of particular interest in the analysis of gene expression data since they can provide a useful tool for visualizing the relationships among genes and generating biological hypotheses.

The standard formulation for estimating a Gaussian graphical model assumes that each observation is drawn from the same distribution. However, in many datasets the observations may correspond to several distinct classes.

Motivation - Genetics

Graphical models are of particular interest in the analysis of gene expression data since they can provide a useful tool for visualizing the relationships among genes and generating biological hypotheses.

The standard formulation for estimating a Gaussian graphical model assumes that each observation is drawn from the same distribution. However, in many datasets the observations may correspond to several distinct classes.

For example, suppose a cancer researcher collects gene expression measurements for a set of cancer tissue samples and a set of normal tissue samples. One might want to estimate a graphical model for the cancer samples and a graphical model for the normal samples.

Motivation - Genetics

Graphical models are of particular interest in the analysis of gene expression data since they can provide a useful tool for visualizing the relationships among genes and generating biological hypotheses.

The standard formulation for estimating a Gaussian graphical model assumes that each observation is drawn from the same distribution. However, in many datasets the observations may correspond to several distinct classes.

For example, suppose a cancer researcher collects gene expression measurements for a set of cancer tissue samples and a set of normal tissue samples. One might want to estimate a graphical model for the cancer samples and a graphical model for the normal samples.

One might expect the two graphical models to be similar to each other since both are based on the same type of tissue, but also have important differences resulting from the fact that gene networks are often dysregulated in cancer.

The Joint Graphical Lasso

The authors propose the *joint graphical lasso* as a technique for jointly estimating multiple graphical models corresponding to distinct but related conditions, such as cancer and normal tissue.

The Joint Graphical Lasso

The authors propose the *joint graphical lasso* as a technique for jointly estimating multiple graphical models corresponding to distinct but related conditions, such as cancer and normal tissue.

Suppose we have data from $K \geq 2$ distinct classes. Instead of estimating $\mathbf{\Sigma}_1^{-1}, \mathbf{\Sigma}_2^{-1}, \ldots, \mathbf{\Sigma}_K^{-1}$ separately, the authors propose estimating estimating these values jointly by maximizing the following penalized log-likelihood function

$$\max_{\{\mathbf{\Theta}\}} \sum_{k=1}^{K} n_k \left[\log \det \mathbf{\Theta}^{(k)} - \text{trace}\left(\boldsymbol{S}^{(k)} \mathbf{\Theta}^{(k)} \right) \right] - P(\{\mathbf{\Theta}\}),$$

subject to the constraint that $\mathbf{\Theta}^{(1)}, \mathbf{\Theta}^{(2)}, \ldots, \mathbf{\Theta}^{(K)}$ are positive definite, where $\{\mathbf{\Theta}\} = \{\mathbf{\Theta}^{(1)}, \mathbf{\Theta}^{(2)}, \ldots, \mathbf{\Theta}^{(k)}\}$.

The Joint Graphical Lasso

The authors propose the *joint graphical lasso* as a technique for jointly estimating multiple graphical models corresponding to distinct but related conditions, such as cancer and normal tissue.

Suppose we have data from $K \geq 2$ distinct classes. Instead of estimating $\boldsymbol{\Sigma}_1^{-1}, \boldsymbol{\Sigma}_2^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$ separately, the authors propose estimating estimating these values jointly by maximizing the following penalized log-likelihood function

$$\max_{\{\boldsymbol{\Theta}\}} \sum_{k=1}^{K} n_k \left[\log \det \boldsymbol{\Theta}^{(k)} - \text{trace}\left(\boldsymbol{S}^{(k)} \boldsymbol{\Theta}^{(k)} \right) \right] - P(\{\boldsymbol{\Theta}\}),$$

subject to the constraint that $\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \ldots, \boldsymbol{\Theta}^{(K)}$ are positive definite, where $\{\boldsymbol{\Theta}\} = \{\boldsymbol{\Theta}^{(1)}, \boldsymbol{\Theta}^{(2)}, \ldots, \boldsymbol{\Theta}^{(k)}\}$.

$P(\{\boldsymbol{\Theta}\})$ denotes a convex penalty function, so the objective function is strictly concave.

Joint Graphical Lasso

The authors propose choosing a penalty function P that will encourage $\hat{\mathbf{\Theta}}^{(1)}, \hat{\mathbf{\Theta}}^{(2)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ to share certain characteristics, such as the locations or values of the nonzero elements, in addition to providing sparse estimates of the precision matrices.

Joint Graphical Lasso

The authors propose choosing a penalty function P that will encourage $\hat{\mathbf{\Theta}}^{(1)}, \hat{\mathbf{\Theta}}^{(2)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ to share certain characteristics, such as the locations or values of the nonzero elements, in addition to providing sparse estimates of the precision matrices.

Two penalty functions suggested by the authors are the *fused graphical lasso* and *group graphical lasso* penalties.

Fused Graphical Lasso

The fused graphical lasso (FGL) penalty is given by

$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{k} \sum_{i \neq j} |\theta_{ij}|^{(k)} + \lambda_2 \sum_{k<k'} \sum_{i,j} |\theta_{ij}^{(k)} - \theta_{ij}^{(k)}|,$$

where λ_1 and λ_2 are nonnegative tuning parameters.

Fused Graphical Lasso

The fused graphical lasso (FGL) penalty is given by

$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{k} \sum_{i \neq j} |\theta_{ij}|^{(k)} + \lambda_2 \sum_{k<k'} \sum_{i,j} |\theta_{ij}^{(k)} - \theta_{ij}^{(k)}|,$$

where λ_1 and λ_2 are nonnegative tuning parameters.

1. Like the graphical lasso, FGL results in sparse estimates $\hat{\mathbf{\Theta}}^{(1)}, \hat{\mathbf{\Theta}}^{(2)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ when the tuning parameter λ_1 is large.

Fused Graphical Lasso

The fused graphical lasso (FGL) penalty is given by

$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{k} \sum_{i \neq j} |\theta_{ij}|^{(k)} + \lambda_2 \sum_{k<k'} \sum_{i,j} |\theta_{ij}^{(k)} - \theta_{ij}^{(k)}|,$$

where λ_1 and λ_2 are nonnegative tuning parameters.

1. Like the graphical lasso, FGL results in sparse estimates $\hat{\mathbf{\Theta}}^{(1)}, \hat{\mathbf{\Theta}}^{(2)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ when the tuning parameter λ_1 is large.

2. Many of the elements $\hat{\mathbf{\Theta}}^{(1)}, \hat{\mathbf{\Theta}}^{(2)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ will be identical across classes when the tuning parameter λ_2 is large.

Fused Graphical Lasso

The fused graphical lasso (FGL) penalty is given by

$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{k} \sum_{i \neq j} |\theta_{ij}|^{(k)} + \lambda_2 \sum_{k<k'} \sum_{i,j} |\theta_{ij}^{(k)} - \theta_{ij}^{(k)}|,$$

where λ_1 and λ_2 are nonnegative tuning parameters.

1. Like the graphical lasso, FGL results in sparse estimates $\hat{\mathbf{\Theta}}^{(1)}, \hat{\mathbf{\Theta}}^{(2)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ when the tuning parameter λ_1 is large.

2. Many of the elements $\hat{\mathbf{\Theta}}^{(1)}, \hat{\mathbf{\Theta}}^{(2)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ will be identical across classes when the tuning parameter λ_2 is large.

3. FGL borrows information aggressively across classes, encouraging not only similar network structure but also similar edge values.

Group Graphical Lasso

The group graphical lasso (GGL) penalty function is given by

$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{k} \sum_{i \neq j} |\theta_{ij}|^{(k)} + \lambda_2 \sum_{i \neq j} \sqrt{\sum_{k=1}^{k} \theta_{ij}^{(k)2}},$$

where λ_1 and λ_2 are nonnegative tuning parameters.

Group Graphical Lasso

The group graphical lasso (GGL) penalty function is given by

$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{k} \sum_{i \neq j} |\theta_{ij}|^{(k)} + \lambda_2 \sum_{i \neq j} \sqrt{\sum_{k=1}^{k} \theta_{ij}^{(k)2}},$$

where λ_1 and λ_2 are nonnegative tuning parameters.

1. The group lasso penalty encourages a similar pattern of sparsity across all of the precision matrices - there will be a tendency for the zeros in the K estimated precision matrices to occur in the same places.

Group Graphical Lasso

The group graphical lasso (GGL) penalty function is given by

$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{k} \sum_{i \neq j} |\theta_{ij}|^{(k)} + \lambda_2 \sum_{i \neq j} \sqrt{\sum_{k=1}^{k} \theta_{ij}^{(k)2}},$$

where λ_1 and λ_2 are nonnegative tuning parameters.

1. The group lasso penalty encourages a similar pattern of sparsity across all of the precision matrices - there will be a tendency for the zeros in the K estimated precision matrices to occur in the same places.

2. When $\lambda_1 = 0$ and $\lambda_2 > 0$, each $\hat{\mathbf{\Theta}}^{(k)}$ will have an identical pattern of non-zero elements. On the other hand, the lasso penalty encourages further sparsity within each $\hat{\mathbf{\Theta}}^{(k)}$.

Group Graphical Lasso

The group graphical lasso (GGL) penalty function is given by

$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{k} \sum_{i \neq j} |\theta_{ij}|^{(k)} + \lambda_2 \sum_{i \neq j} \sqrt{\sum_{k=1}^{k} \theta_{ij}^{(k)2}},$$

where λ_1 and λ_2 are nonnegative tuning parameters.

1. The group lasso penalty encourages a similar pattern of sparsity across all of the precision matrices - there will be a tendency for the zeros in the K estimated precision matrices to occur in the same places.

2. When $\lambda_1 = 0$ and $\lambda_2 > 0$, each $\hat{\mathbf{\Theta}}^{(k)}$ will have an identical pattern of non-zero elements. On the other hand, the lasso penalty encourages further sparsity within each $\hat{\mathbf{\Theta}}^{(k)}$.

3. GGL encourages a weaker form of similarity across the K precision matrices than FGL in that GGL only encourages a shared pattern of sparsity and not shared edge values.

Algorithm for the Joint Graphical Lasso

The author solve the joint graphical lasso problem using an *alternating directions method of multipliers* (ADMM) algorithm.

Algorithm for the Joint Graphical Lasso

The author solve the joint graphical lasso problem using an *alternating directions method of multipliers* (ADMM) algorithm.

The original problem can be rewritten as

$$\min_{\{\boldsymbol{\Theta}\},\{\boldsymbol{Z}\}} -\sum_{k=1}^{K} n_k \left[\log\,\det \boldsymbol{\Theta}^{(k)} - \text{trace}\left(\boldsymbol{S}^{(k)}\boldsymbol{\Theta}^{(k)}\right)\right] + P(\{\boldsymbol{Z}\})$$

subject to the positive-definiteness constraint as well as the constraint that $\boldsymbol{Z}^{(k)} = \boldsymbol{\Theta}^{(k)}$, where $\{\boldsymbol{Z}\} = \{\boldsymbol{Z}^{(1)}, \boldsymbol{Z}^{(2)}, \ldots, \boldsymbol{Z}^{(k)}\}$.

Algorithm for the Joint Graphical Lasso

The scaled augmented Lagrangian for this problem is given by

$$L_\rho(\{\boldsymbol{\Theta}\},\{\boldsymbol{Z}\},\{\boldsymbol{U}\}) = -\sum_{k=1}^{K} n_k \left[\log\,\det \boldsymbol{\Theta}^{(k)} - \text{trace}\left(\boldsymbol{S}^{(k)}\boldsymbol{\Theta}^{(k)}\right)\right] + P(\{\boldsymbol{Z}\}) + \frac{\rho}{2}\sum_{k=1}^{K} ||\boldsymbol{\Theta}^{(k)} - \boldsymbol{Z}^{(k)} + \boldsymbol{U}^{(k)}||_F^k,$$

where $\{\boldsymbol{U}\} = \{\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(k)}\}$.

Algorithm for the Joint Graphical Lasso

The scaled augmented Lagrangian for this problem is given by

$$L_\rho(\{\boldsymbol{\Theta}\},\{\boldsymbol{Z}\},\{\boldsymbol{U}\}) = -\sum_{k=1}^{K} n_k \left[\log\,\det \boldsymbol{\Theta}^{(k)} - \text{trace}\left(\boldsymbol{S}^{(k)}\boldsymbol{\Theta}^{(k)}\right)\right] + P(\{\boldsymbol{Z}\}) + \frac{\rho}{2}\sum_{k=1}^{K} ||\boldsymbol{\Theta}^{(k)} - \boldsymbol{Z}^{(k)} + \boldsymbol{U}^{(k)}||_F^k,$$

where $\{\boldsymbol{U}\} = \{\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(k)}\}$.

The ADMM corresponding to the above problem results in iterating three simple steps.

(a) $\{\boldsymbol{\Theta}_{(i)}\} \leftarrow \arg\min_{\{\boldsymbol{\Theta}\}}\{L_\rho(\{\boldsymbol{\Theta}\},\{\boldsymbol{Z}_{(i-1)}\},\{\boldsymbol{U}_{i-1}\})\}$

(b) $\{\boldsymbol{Z}_{(i)}\} \leftarrow \arg\min_{\{\boldsymbol{\Theta}_i\}}\{L_\rho(\{\boldsymbol{\Theta}\},\{\boldsymbol{Z}\},\{\boldsymbol{U}_{i-1}\})\}$

(c) $\{\boldsymbol{U}_{(i)}\} \leftarrow \{\boldsymbol{U}_{(i-1)}\} + (\{\boldsymbol{\Theta}_{(i)}\} - \{\boldsymbol{Z}_{(i)}\})$.

Algorithm for the Joint Graphical Lasso

The scaled augmented Lagrangian for this problem is given by

$$L_\rho(\{\boldsymbol{\Theta}\},\{\boldsymbol{Z}\},\{\boldsymbol{U}\}) = -\sum_{k=1}^{K} n_k \left[\log\,\det \boldsymbol{\Theta}^{(k)} - \text{trace}\left(\boldsymbol{S}^{(k)}\boldsymbol{\Theta}^{(k)}\right)\right] + P(\{\boldsymbol{Z}\}) + \frac{\rho}{2}\sum_{k=1}^{K} ||\boldsymbol{\Theta}^{(k)} - \boldsymbol{Z}^{(k)} + \boldsymbol{U}^{(k)}||_F^k,$$

where $\{\boldsymbol{U}\} = \{\boldsymbol{U}^{(1)}, \boldsymbol{U}^{(2)}, \ldots, \boldsymbol{U}^{(k)}\}$.

The ADMM corresponding to the above problem results in iterating three simple steps.

(a) $\{\boldsymbol{\Theta}_{(i)}\} \leftarrow \arg\min_{\{\boldsymbol{\Theta}\}}\{L_\rho(\{\boldsymbol{\Theta}\},\{\boldsymbol{Z}_{(i-1)}\},\{\boldsymbol{U}_{i-1}\})\}$

(b) $\{\boldsymbol{Z}_{(i)}\} \leftarrow \arg\min_{\{\boldsymbol{\Theta}_i\}}\{L_\rho(\{\boldsymbol{\Theta}\},\{\boldsymbol{Z}\},\{\boldsymbol{U}_{i-1}\})\}$

(c) $\{\boldsymbol{U}_{(i)}\} \leftarrow \{\boldsymbol{U}_{(i-1)}\} + (\{\boldsymbol{\Theta}_{(i)}\} - \{\boldsymbol{Z}_{(i)}\})$.

The update in step (a) preserves positive-definiteness, thus this constraint can be satisfied simply by initializing $\boldsymbol{\Theta}_{(0)}^{(k)} = \boldsymbol{I}$.

Simulation Study

The data for main simulation study consisted of generating three networks with $p = 500$ features belonging to ten equally sized unconnected subnetworks, each with a power law degree distribution, i.e. a scale-free network. Of the ten subnetworks, eight have the same structure and edge values in all three classes, one is identical between the first two classes and missing in the third (i.e. the corresponding features are singletons in the third network), and one is present in only the first class. The corresponding graph is shown in Figure 1.

Simulation Study

The data for main simulation study consisted of generating three networks with p = 500 features belonging to ten equally sized unconnected subnetworks, each with a power law degree distribution, i.e. a scale-free network. Of the ten subnetworks, eight have the same structure and edge values in all three classes, one is identical between the first two classes and missing in the third (i.e. the corresponding features are singletons in the third network), and one is present in only the first class. The corresponding graph is shown in Figure 1.

In addition to the 500-feature network pair, we generate a pair of networks with p = 1000 features, each of which is block diagonal with 500×500 blocks corresponding to two copies of the 500-feature networks just described.

Figure 1

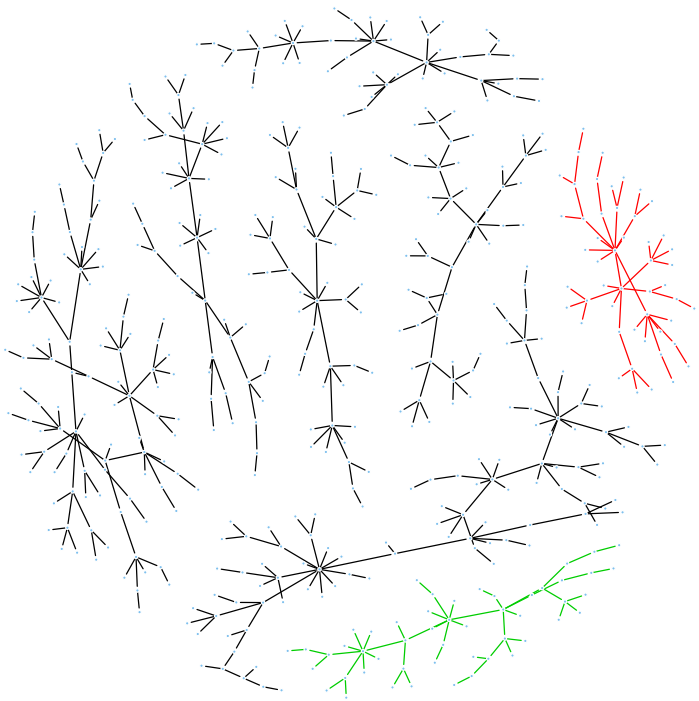

Figure: This shows the graph corresponding to the concentration matrix for the main simulation study. Black edges are common to all three classes, green edges are present only in classes 1 and 2, and red edges are present only in class 1.

Basic Result

As we will see in the next slides, when $p = 500$ and $n = 150$, FGL performs the best in the simulations conducted by Danaher et. al. although it is much slower than $glasso$ (the fastest) and the group lasso penalty case. The graphical lasso and the proposal of Guo et al. (2011) - to be discussed next week by André and Ray Bai - are included in the comparisons as well.

Basic Result

As we will see in the next slides, when $p = 500$ and $n = 150$, FGL performs the best in the simulations conducted by Danaher et. al. although it is much slower than $glasso$ (the fastest) and the group lasso penalty case. The graphical lasso and the proposal of Guo et al. (2011) - to be discussed next week by André and Ray Bai - are included in the comparisons as well.

In addition, FGL also seems to do better than GGL asymptotically when we hold p to be fixed and let n increase.

In terms of parameters:

FGL is presented in terms of λ_2 but for GGL they reparametrize the results in terms $\omega_2 = \frac{1}{\sqrt{2}}\lambda_2/(\lambda_1 + \frac{1}{\sqrt{2}}\lambda_2)$. Each line corresponds to a different value of λ_2 and ω_2.

In terms of parameters:

FGL is presented in terms of λ_2 but for GGL they reparametrize the results in terms $\omega_2 = \frac{1}{\sqrt{2}}\lambda_2/(\lambda_1 + \frac{1}{\sqrt{2}}\lambda_2)$. Each line corresponds to a different value of λ_2 and ω_2.

As λ_1 or $\omega_1 = \lambda_1 + \frac{1}{\sqrt{2}}\lambda_2$ increases, sparsity increases, or equivalently, the number of edges selected decreases.

In terms of parameters:

FGL is presented in terms of λ_2 but for GGL they reparametrize the results in terms $\omega_2 = \frac{1}{\sqrt{2}}\lambda_2/(\lambda_1 + \frac{1}{\sqrt{2}}\lambda_2)$. Each line corresponds to a different value of λ_2 and ω_2.

As λ_1 or $\omega_1 = \lambda_1 + \frac{1}{\sqrt{2}}\lambda_2$ increases, sparsity increases, or equivalently, the number of edges selected decreases.

According to the authors, approaches such as AIC, BIC, and cross-validation tend to choose models too large to be useful. In this setting, model selection should be guided by practical considerations, such as network interpretability, stability, and the desire for an edge set with a low false discovery rate.

Some measures of error

$$SSE = \sum_{k=1}^{K} \sum_{i \neq j} (\hat{\theta}_{ij}^{(k)} - (\Sigma^{(k)})_{ij}^{-1})^2$$

Some measures of error

$$SSE = \sum_{k=1}^{K} \sum_{i \neq j} (\hat{\theta}_{ij}^{(k)} - (\Sigma^{(k)})_{ij}^{-1})^2$$

Differential edges are defined as the edges that differ between classes. For FGL, it is computed as the number of pairs $k < k', i < j$ such that $\hat{\theta}_{ij}^{(k)} \neq \hat{\theta}_{ij}^{(k')}$. For GGL, the proposal of Guo et al. (2011), and the graphical lasso it is computed as the number of pairs $k < k', i < j$ such that $|\hat{\theta}_{ij}^{(k)} - \hat{\theta}_{ij}^{(k')}| > 10^{-2}$.

Some measures of error

$$SSE = \sum_{k=1}^{K} \sum_{i \neq j} (\hat{\theta}_{ij}^{(k)} - (\Sigma^{(k)})_{ij}^{-1})^2$$

Differential edges are defined as the edges that differ between classes. For FGL, it is computed as the number of pairs $k < k', i < j$ such that $\hat{\theta}_{ij}^{(k)} \neq \hat{\theta}_{ij}^{(k')}$. For GGL, the proposal of Guo et al. (2011), and the graphical lasso it is computed as the number of pairs $k < k', i < j$ such that $|\hat{\theta}_{ij}^{(k)} - \hat{\theta}_{ij}^{(k')}| > 10^{-2}$.

The Kullback-Leibler Divergence (dKL) from the multivariate normal model with inverse covariance estimates $\Theta^{(1)}, ..., \Theta^{(k)}$to the multivariate normal model with the true precision matrices $\Sigma^{(1)}, ..., \Sigma^{(k)}$ is

$$\frac{1}{2} \sum_{k=1}^{K} (-\log \det(\Theta^{(k)} \Sigma^{(k)}) + \text{trace}(\Theta^{(k)} \Sigma^{(k)}))$$

Figure 2 from Danaher et. al

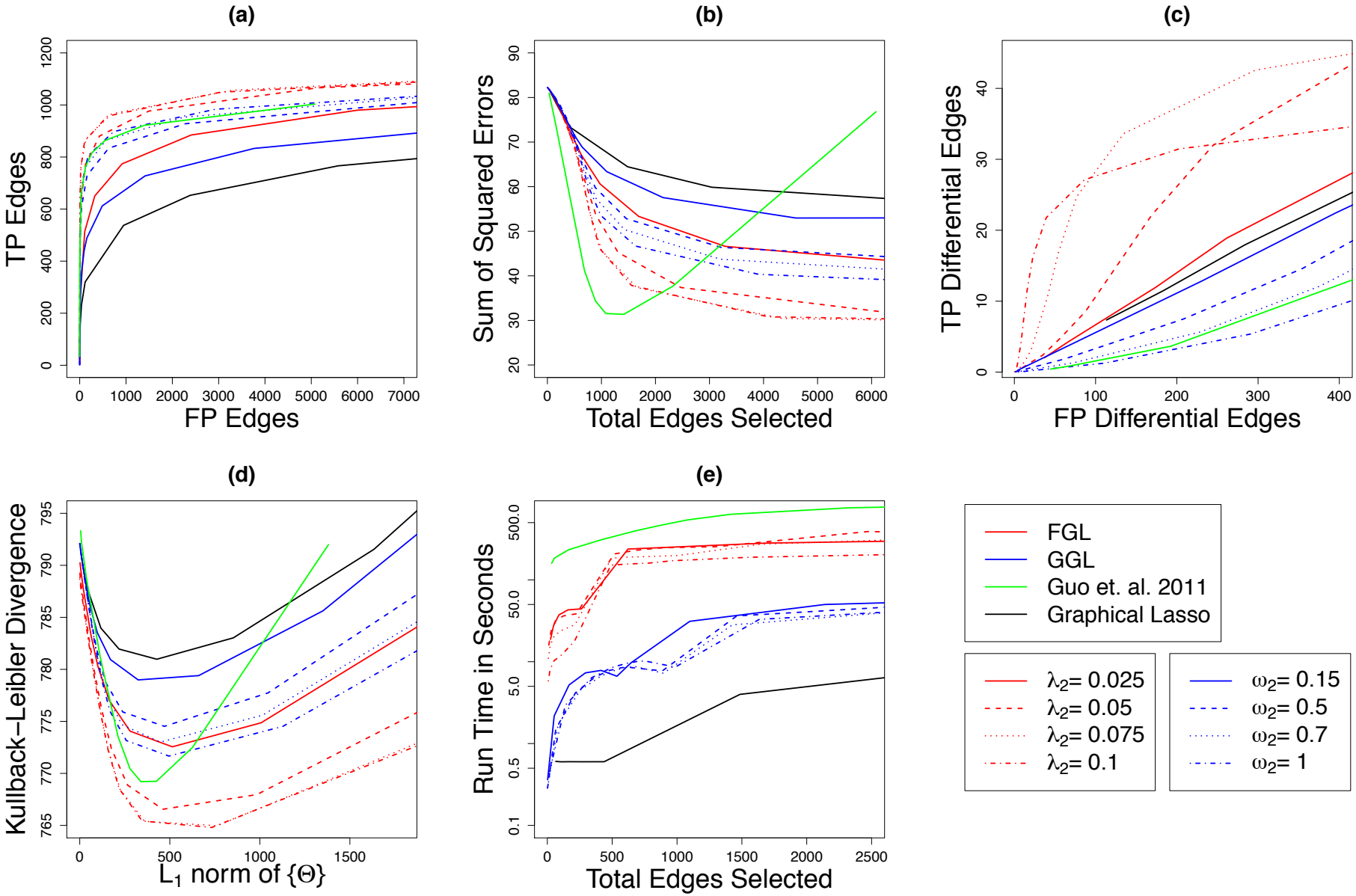

Figure 3 from Danaher et. al - Analysis of lung cancer microarray data

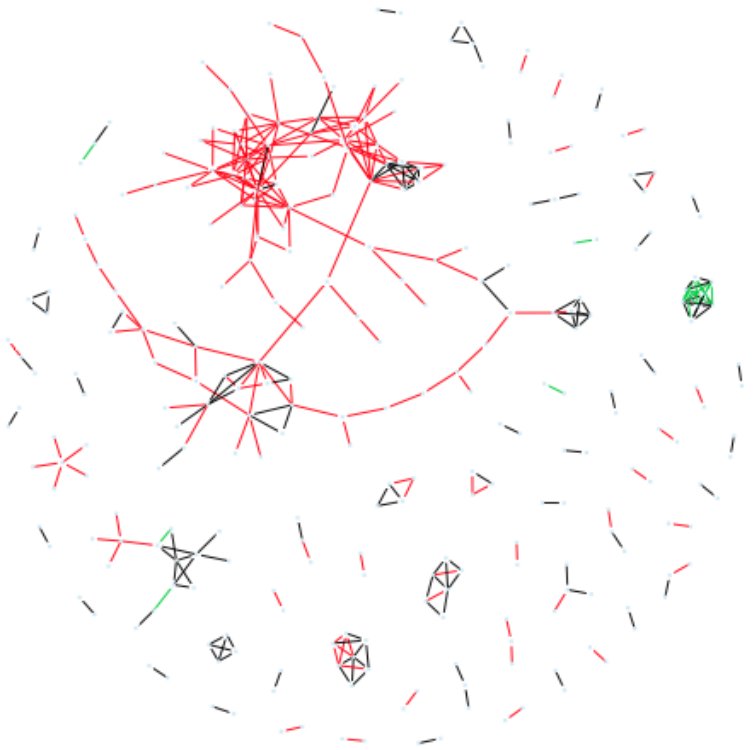

Figure: Conditional dependency networks inferred from 17,772 genes in normal and cancerous lung cells. 278 genes have nonzero edges in at least one of the two networks. Black lines denote edges common to both classes. Red and green lines denote tumor-specific and normal-specific edges, respectively. The parameters used were $\lambda_1 = 0.95$ and $\lambda_2 = 0.005$.

Table 1 from Danaher et. al

Table 1. *Performances as a function of n and p. Means over 100 replicates are shown for dKL, and for sensitivity (Sens.) and false discovery rate (FDR) of detection of edges (DE) and differential edge detection (DED).*

	p	n	dKL	DE Sens.	DE FDR	DED Sens.	DED FDR
FGL	500	50	545.1	0.502	0.966	0.262	0.996
		200	517.5	0.570	0.053	0.228	0.485
		500	516.6	0.590	0.001	0.192	0.036
	1000	50	1119.3	0.600	0.970	0.245	0.998
		200	1035.0	0.666	0.063	0.223	0.557
		500	1033.3	0.681	0.000	0.194	0.025
GGL	500	50	549.8	0.490	0.973	0.337	0.996
		200	520.8	0.505	0.060	0.244	0.903
		500	519.7	0.524	0.010	0.194	0.921
	1000	50	1127.9	0.587	0.976	0.316	0.998
		200	1041.7	0.615	0.061	0.239	0.908
		500	1039.4	0.629	0.007	0.197	0.920

Table 1 indicates that

for fixed p, as n increases, FGL seems to do much better in terms of edge detection (as expected, sensitivity increases and false discovery rate decreases as n increases) and differential edge detection (false discovery rate decreases as n increases) than GGL. Oddly enough, in all cases, DED Sens declines as n increases . Hence for smaller n the model is better at predicting when edges differ across classes.

Nearest Neighbor Network (n = 100, p =25)

	Group	
	$\lambda_1 = 0.07, \lambda_2 = 0.05$	$\lambda_1 = 0.05, \lambda_2 = 0.25$
F	0.05642315	0.06010037
FP	0.3391197	0.02918924
FN	0.3035487	0.8517427

Table: Nearest Neighbor Network for Group Lasso Penalty

	Fused	
	$\lambda_1 = 0.05, \lambda_2 = 0.05$	$\lambda_1 = 0.2, \lambda_2 = 0.1$
F	0.06335103	0.06954624
FP	0.4269327	0.002245857
FN	0.3139659	0.9723261

Table: Nearest Neighbor Network for Fused Lasso Penalty

Additional notes

In the appendix, they show that if you go down to 2 classes, FGL performs as fast as GGL while still maintaining the results best overall.

Additional notes

In the appendix, they show that if you go down to 2 classes, FGL performs as fast as GGL while still maintaining the results best overall.

We thought it was odd that the authors were much more interested in having a low false positive rate at the cost of a high false negative instead of a balance between the two.