Journal of the Royal Statistical Society
SERIES B Statistical Methodology

J. R. Statist. Soc. B (2014)
76, *Part* 2, *pp.* 373–397

The joint graphical lasso for inverse covariance estimation across multiple classes

Patrick Danaher,

University of Washington, Seattle, USA

Pei Wang,

Fred Hutchinson Cancer Research Center, Seattle, USA

and Daniela M. Witten

University of Washington, Seattle, USA

[Received October 2011. Final revision April 2013]

Summary. We consider the problem of estimating multiple related Gaussian graphical models from a high dimensional data set with observations belonging to distinct classes. We propose the *joint graphical lasso*, which borrows strength across the classes to estimate multiple graphical models that share certain characteristics, such as the locations or weights of non-zero edges. Our approach is based on maximizing a penalized log-likelihood. We employ generalized fused lasso or group lasso penalties and implement a fast alternating directions method of multipliers algorithm to solve the corresponding convex optimization problems. The performance of the method proposed is illustrated through simulated and real data examples.

Keywords: Alternating directions method of multipliers; Gaussian graphical model; Generalized fused lasso; Graphical lasso; Group lasso; High dimensional data; Network estimation

1. Introduction

In recent years, much interest has focused on estimating an undirected graphical model on the basis of an $n \times p$ data matrix $\mathbf{X}$, where n is the number of observations and p is the number of features. Suppose that the observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$ 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. Then 0s in the inverse covariance matrix $\boldsymbol{\Sigma}^{-1}$ correspond to pairs of features that are conditionally independent—i.e. pairs of variables that are independent of each other, given all of the other variables in the data set. These conditional dependence relationships can be represented by a graph in which nodes represent features and edges connect conditionally dependent pairs of features (Lauritzen, 1996).

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

$$\frac{n}{2}[\log\{\det(\boldsymbol{\Sigma}^{-1})\} - \mathrm{tr}(\mathbf{S}\boldsymbol{\Sigma}^{-1})]. \tag{1}$$

Maximizing expression (1) with respect to $\boldsymbol{\Sigma}^{-1}$ yields the maximum likelihood estimate $\mathbf{S}^{-1}$.

Address for correspondence: Pei Wang, Public Health Sciences Division, Fred Hutchinson Cancer Research Center, M2-B230, 1100 Fairview Avenue North, Seattle, WA 98109, USA.
E-mail: pwang@fhcrc.org

© 2013 Royal Statistical Society 1369–7412/14/76373

However, two problems can arise in using this maximum likelihood approach to estimate $\mathbf{\Sigma}^{-1}$. First, in the high dimensional setting where the number of features p is larger than the number of observations n, the empirical covariance matrix $\mathbf{S}$ is singular and so cannot be inverted to yield an estimate of $\mathbf{\Sigma}^{-1}$. If $p \approx n$, then even if $\mathbf{S}$ is not singular the maximum likelihood estimate for $\mathbf{\Sigma}^{-1}$ will suffer from very high variance. Second, one often is interested in identifying pairs of variables that are unconnected in the graphical model, i.e. that are conditionally independent; these correspond to 0s in $\mathbf{\Sigma}^{-1}$. But maximizing the log-likelihood (1) will in general yield an estimate of $\mathbf{\Sigma}^{-1}$ with no elements that are exactly equal to 0.

In recent years, various proposals have been made for estimating $\mathbf{\Sigma}^{-1}$ in the high dimensional setting in such a way that the resulting estimate is *sparse*. Meinshausen and Bühlmann (2006) proposed to do this via a penalized regression approach, which was extended by Peng *et al.* (2009). Various other researchers have instead taken a penalized log-likelihood approach (among others, Yuan and Lin (2006), Friedman *et al.* (2007b) and Rothman *et al.* (2008)): rather than maximizing expression (1), one can instead solve the problem

$$\text{maximize}_{\mathbf{\Theta}}[\log\{\det(\mathbf{\Theta})\} - \text{tr}(\mathbf{S\Theta}) - \lambda\|\mathbf{\Theta}\|_1], \tag{2}$$

where λ is a non-negative tuning parameter, and $\|\mathbf{\Theta}\|_1$ denotes the sum of the absolute values of the elements of $\mathbf{\Theta}$. This is a convex optimization problem, and its solution provides an estimate for $\mathbf{\Sigma}^{-1}$. The use of an l_1- or *lasso* (Tibshirani, 1996) penalty on the elements of $\mathbf{\Theta}$ has the effect that, when the tuning parameter λ is large, some elements of the resulting precision matrix estimate will be exactly equal to 0. Moreover, problem (2) can be solved even if $p \gg n$. The solution to problem (2) is referred to as the *graphical lasso*. Some researchers have proposed applying the l_1-penalty in problem (2) only to the off-diagonal elements of $\mathbf{\Theta}$.

Graphical models are especially of interest in the analysis of gene expression data, since it is believed that genes operate in pathways, or networks. Graphical models based on gene expression data can provide a useful tool for visualizing the relationships between genes and for 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 data sets the observations may correspond to several distinct classes, so the assumption that all observations are drawn from the same distribution is inappropriate. For instance, suppose that a cancer researcher collects gene expression measurements for a set of cancer tissue samples and a set of normal tissue samples. In this case, one might want to estimate a graphical model for the cancer samples and a graphical model for the normal samples. One would expect the two graphical models to be similar to each other, since both are based on the same type of tissue, but also to have important differences stemming from the fact that gene networks are often dysregulated in cancer. Estimating separate graphical models for the cancer and normal samples does not exploit the similarity between the true graphical models. And estimating a single graphical model for the cancer and normal samples ignores the fact that we do not expect the true graphical models to be identical, and that the differences between the graphical models may be of interest.

In this paper, we propose the *joint graphical lasso* (JGL), a technique for jointly estimating multiple graphical models corresponding to distinct but related conditions, such as cancer and normal tissue. Our approach is an extension of the graphical lasso (2) to the case of multiple data sets. It is based on a penalized log-likelihood approach, where the choice of penalty depends on the characteristics of the graphical models that we expect to be shared across conditions.

We illustrate our method with a small toy example that consists of observations from two classes. Within each class, the observations are independent and identically distributed according to a normal distribution. The two classes have distinct covariance matrices. When we apply

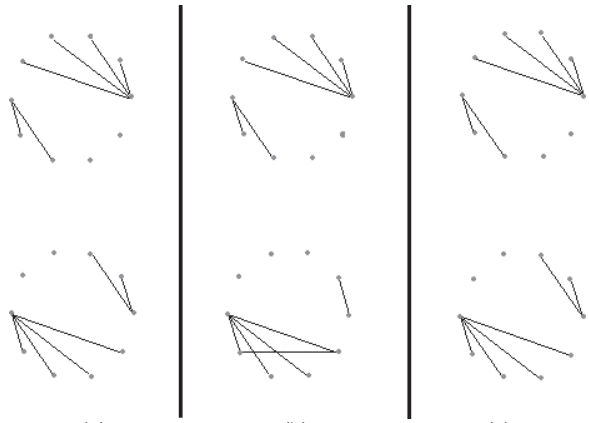

Fig. 1. Comparison of the graphical lasso with our JGL in a toy example with two conditions, $p = 10$ variables and $n = 200$ observations per condition: (a) true networks; (b) networks estimated by applying the graphical lasso separately to each class; (c) networks estimated by applying our JGL proposal

the graphical lasso separately to the observations in each class, the resulting graphical model estimates are less accurate than when we use our JGL approach. Results are shown in Fig. 1.

The rest of this paper is organized as follows. In Section 2, we present the JGL optimization problem. Section 3 contains an alternating directions method of multipliers (ADMM) algorithm for its solution. In Section 4, we present theoretical results that lead to massive gains in the algorithm's computational efficiency. Section 5 contains a discussion of related approaches from the literature, and in Section 6 we discuss tuning parameter selection. In Section 7, we illustrate the performance of our proposal in a simulation study. Section 8 contains an application to a lung cancer gene expression data set. The discussion is in Section 9.

The programs that were used to generate the tables and figures shown in this paper can be obtained from

`http://wileyonlinelibrary.com/journal/rss-datasets`

and the R package `JGL` that implements the proposed approaches is available from the Comprehensive R Archive Network.

2. Joint graphical lasso

We briefly introduce some notation that will be used throughout this paper.

We let K denote the number of classes in our data and let $\mathbf{\Sigma}_k^{-1}$ denote the true precision matrix for the kth class. We shall seek to estimate $\mathbf{\Sigma}_1^{-1}, \ldots, \mathbf{\Sigma}_K^{-1}$ by formulating convex optimization problems with arguments $\{\mathbf{\Theta}\} = \mathbf{\Theta}^{(1)}, \ldots, \mathbf{\Theta}^{(K)}$. The solutions $\{\hat{\mathbf{\Theta}}\} = \hat{\mathbf{\Theta}}^{(1)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ to these optimization problems will constitute estimates of $\{\mathbf{\Sigma}^{-1}\} = \mathbf{\Sigma}_1^{-1}, \ldots, \mathbf{\Sigma}_K^{-1}$.

We shall index matrix elements by using $i = 1, \ldots, p$ and $j = 1, \ldots, p$, and we shall index classes by using $k = 1, \ldots, K$. $\|\mathbf{A}\|_F$ will denote the Frobenius norm of matrix $\mathbf{A}$, i.e. $\|\mathbf{A}\|_F = \sqrt{(\sum_i \sum_j A_{ij}^2)}$.

2.1. General formulation for the joint graphical lasso

Suppose that we are given K data sets $\mathbf{Y}^{(1)}, \ldots, \mathbf{Y}^{(K)}$, with $K \geqslant 2$. $\mathbf{Y}^{(k)}$ is an $n_k \times p$ matrix consisting of n_k observations with measurements on a set of p features, which are common to

all K data sets. Furthermore, we assume that the $\sum_{k=1}^{K} n_k$ observations are independent, and that the observations within each data set are identically distributed: $\mathbf{y}_1^{(k)}, \ldots, \mathbf{y}_{n_k}^{(k)} \sim N(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$. Without loss of generality, we assume that the features within each data set are centred such that $\boldsymbol{\mu}_k = \mathbf{0}$. We let $\mathbf{S}^{(k)} = (1/n_k)(\mathbf{Y}^{(k)})^{\mathrm{T}}\mathbf{Y}^{(k)}$, the empirical covariance matrix for $\mathbf{Y}^{(k)}$. The log-likelihood for the data takes the form (up to a constant)

$$l(\{\boldsymbol{\Theta}\}) = \frac{1}{2}\sum_{k=1}^{K} n_k[\log\{\det(\boldsymbol{\Theta}^{(k)})\} - \mathrm{tr}(\mathbf{S}^{(k)}\boldsymbol{\Theta}^{(k)})]. \tag{3}$$

Maximizing equation (3) with respect to $\boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(K)}$ yields the maximum likelihood estimate $(\mathbf{S}^{(1)})^{-1}, \ldots, (\mathbf{S}^{(K)})^{-1}$.

However, depending on the application, the maximum likelihood estimates that result from equation (3) may not be satisfactory. When p is smaller than but close to n_k, the maximum likelihood estimates can have very high variance, and no elements of $(\mathbf{S}^{(1)})^{-1}, \ldots, (\mathbf{S}^{(K)})^{-1}$ will be 0, leading to difficulties in interpretation. In addition, when $p > n_k$, the maximum likelihood estimates become ill defined. Moreover, if the K data sets correspond to observations collected from K distinct but related classes, then we might wish to borrow strength across the K classes to estimate the K precision matrices, rather than estimating each precision matrix separately.

Therefore, instead of estimating $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$ by maximizing equation (3), we take a penalized log-likelihood approach and seek $\{\hat{\boldsymbol{\Theta}}\}$ solving

$$\mathrm{maximize}_{\{\boldsymbol{\Theta}\}}\left(\sum_{k=1}^{K} n_k[\log\{\det(\boldsymbol{\Theta}^{(k)})\} - \mathrm{tr}(\mathbf{S}^{(k)}\boldsymbol{\Theta}^{(k)})] - P(\{\boldsymbol{\Theta}\})\right) \tag{4}$$

subject to the constraint that $\boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(K)}$ are positive definite. Here $P(\{\boldsymbol{\Theta}\})$ denotes a convex penalty function, so that the objective in problem (4) is strictly concave in $\{\boldsymbol{\Theta}\}$. We propose to choose a penalty function P that will encourage $\hat{\boldsymbol{\Theta}}^{(1)}, \ldots, \hat{\boldsymbol{\Theta}}^{(K)}$ to share certain characteristics, such as the locations or values of the non-zero elements; moreover, we would like the estimated precision matrices to be sparse. In particular, we shall consider penalty functions that take the form

$$P(\{\boldsymbol{\Theta}\}) = \lambda_1 \sum_{k}\sum_{i \neq j} |\theta_{ij}^{(k)}| + \tilde{P}(\{\boldsymbol{\Theta}\}),$$

where λ_1 is a non-negative tuning parameter and $\tilde{P}$ is a convex function. When $\tilde{P}(\{\boldsymbol{\Theta}\}) = 0$, problem (4) amounts to performing K uncoupled graphical lasso optimization problems (2). The $\tilde{P}$-penalty is chosen to encourage similarity across the K estimated precision matrices; therefore, we refer to the solution to problem (4) as the joint graphical lasso (JGL). We discuss specific forms of the penalty function in problem (4) in the next section.

2.2. Two useful penalty functions

In this subsection, we introduce two particular choices of the convex penalty function P in problem (4) that lead to useful graphical model estimates. In Appendix A, we further extend these proposals to work on the scale of partial correlations.

2.2.1. Fused graphical lasso

The *fused graphical lasso* (FGL) is the solution to problem (4) with the penalty

$$P(\{\boldsymbol{\Theta}\}) = \lambda_1 \sum_{k=1}^{K}\sum_{i \neq j} |\theta_{ij}^{(k)}| + \lambda_2 \sum_{k<k'}\sum_{i,j} |\theta_{ij}^{(k)} - \theta_{ij}^{(k')}|, \tag{5}$$

where λ_1 and λ_2 are non-negative tuning parameters. This is a *generalized fused lasso* penalty (Hoefling, 2010a) and results from applying l_1-penalties to

(a) each off-diagonal element of the K precision matrices and
(b) differences between corresponding elements of each pair of precision matrices.

Like the graphical lasso, the FGL results in sparse estimates $\hat{\mathbf{\Theta}}^{(1)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ when the tuning parameter λ_1 is large. In addition, many elements of $\hat{\mathbf{\Theta}}^{(1)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ will be identical across classes when the tuning parameter λ_2 is large (Tibshirani *et al*., 2005). Thus the FGL borrows information aggressively across classes, encouraging not only similar network structure but also similar edge values.

2.2.2. Group graphical lasso

We define the *group graphical lasso* (GGL) to be the solution to problem (4) with

$$P(\{\mathbf{\Theta}\}) = \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |\theta_{ij}^{(k)}| + \lambda_2 \sum_{i \neq j} \left(\sum_{k=1}^{K} \theta_{ij}^{(k)2} \right)^{1/2}. \tag{6}$$

Again, λ_1 and λ_2 are non-negative tuning parameters. A lasso penalty is applied to the elements of the precision matrices and a *group lasso* penalty is applied to the (i, j) element across all K precision matrices (Yuan and Lin, 2007). This group lasso penalty encourages a similar pattern of sparsity across all the precision matrices—i.e. there will be a tendency for the 0s in the K estimated precision matrices to occur in the same places. Specifically, on the one hand, 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 $\hat{\mathbf{\Theta}}^{(k)}$.

The GGL encourages a weaker form of similarity across the K precision matrices than does the FGL: the latter encourages shared edge values across the K matrices, whereas the former encourages only a shared pattern of sparsity.

3. Algorithm for the joint graphical lasso problem

3.1. An alternating directions method of multipliers algorithm

We solve problem (4) by using an ADMM algorithm. We refer the reader to Boyd *et al*. (2010) for a thorough exposition of ADMM algorithms as well as their convergence properties, and to Simon and Tibshirani (2011) and Mohan *et al*. (2012) for recent applications of ADMM algorithms to related problems.

To solve problem (4) subject to the constraint that $\mathbf{\Theta}^{(k)}$ is positive definite for $k = 1, \ldots, K$ using the ADMM algorithm, we note that the problem can be rewritten as

$$\underset{\{\mathbf{\Theta}\},\{\mathbf{Z}\}}{\text{minimize}} \left(- \sum_{k=1}^{K} n_k [\log\{\det(\mathbf{\Theta}^{(k)})\} - \text{tr}(\mathbf{S}^{(k)}\mathbf{\Theta}^{(k)})] + P(\{\mathbf{Z}\}) \right), \tag{7}$$

subject to the positive definiteness constraint as well as the constraint that $\mathbf{Z}^{(k)} = \mathbf{\Theta}^{(k)}$ for $k = 1, \ldots, K$, where $\{\mathbf{Z}\} = \mathbf{Z}^{(1)}, \ldots, \mathbf{Z}^{(K)}$. The scaled augmented Lagrangian (Boyd *et al*., 2010) for this problem is given by

$$\begin{aligned} L_\rho(\{\mathbf{\Theta}\}, \{\mathbf{Z}\}, \{\mathbf{U}\}) = & - \sum_{k=1}^{K} n_k [\log\{\det(\mathbf{\Theta}^{(k)})\} - \text{tr}(\mathbf{S}^{(k)}\mathbf{\Theta}^{(k)})] + P(\{\mathbf{Z}\}) \\ & + \frac{\rho}{2} \sum_{k=1}^{K} \|\mathbf{\Theta}^{(k)} - \mathbf{Z}^{(k)} + \mathbf{U}^{(k)}\|_{\text{F}}^2 - \frac{\rho}{2} \sum_{k=1}^{K} \|\mathbf{U}^{(k)}\|_{\text{F}}^2, \end{aligned} \tag{8}$$

where $\{\mathbf{U}\} = \mathbf{U}^{(1)}, \ldots, \mathbf{U}^{(K)}$ are dual variables and ρ serves as a 'penalty parameter'. Roughly speaking, an ADMM algorithm corresponding to equation (8) results from iterating three simple steps. At the ith iteration, they are as follows.

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

We now present the ADMM algorithm in greater detail for solving the JGL problem.

(a) Initialize the variables: $\boldsymbol{\Theta}^{(k)} = \mathbf{I}$, $\mathbf{U}^{(k)} = \mathbf{0}$ and $\mathbf{Z}^{(k)} = \mathbf{0}$ for $k = 1, \ldots, K$.
(b) Select a scalar $\rho > 0$.
(c) For $i = 1, 2, 3, \ldots$ until convergence update as follows.
  (i) For $k = 1, \ldots, K$, update $\boldsymbol{\Theta}^{(k)}_{(i)}$ as the minimizer (with respect to $\boldsymbol{\Theta}^{(k)}$) of

$$-n_k[\log\{\det(\boldsymbol{\Theta}^{(k)})\} - \operatorname{tr}(\mathbf{S}^{(k)}\boldsymbol{\Theta}^{(k)})] + \frac{\rho}{2}\|\boldsymbol{\Theta}^{(k)} - \mathbf{Z}^{(k)}_{(i-1)} + \mathbf{U}^{(k)}_{(i-1)}\|^2_{\mathrm{F}}.$$

  Letting $\mathbf{V}\mathbf{D}\mathbf{V}^{\mathrm{T}}$ denote the eigendecomposition of $\mathbf{S}^{(k)} - \rho\mathbf{Z}^{(k)}_{(i-1)}/n_k + \rho\mathbf{U}^{(k)}_{(i-1)}/n_k$, the solution is given (Witten and Tibshirani, 2009) by $\mathbf{V}\tilde{\mathbf{D}}\mathbf{V}^{\mathrm{T}}$, where $\tilde{\mathbf{D}}$ is the diagonal matrix with jth diagonal element

$$\frac{n_k}{2\rho}\{-D_{jj} + (D^2_{jj} + 4\rho/n_k)^{1/2}\}.$$

  (ii) Update $\{\mathbf{Z}_{(i)}\}$ as the minimizer (with respect to $\{\mathbf{Z}\}$) of

$$\frac{\rho}{2}\sum_{k=1}^{K}\|\mathbf{Z}^{(k)} - (\boldsymbol{\Theta}^{(k)}_{(i)} + \mathbf{U}^{(k)}_{(i-1)})\|^2_{\mathrm{F}} + P(\{\mathbf{Z}\}). \tag{9}$$

  (iii) For $k = 1, \ldots, K$, update $\mathbf{U}^{(k)}_{(i)}$ as $\mathbf{U}^{(k)}_{(i-1)} + \boldsymbol{\Theta}^{(k)}_{(i)} - \mathbf{Z}^{(k)}_{(i)}$.

The final $\hat{\boldsymbol{\Theta}}^{(1)}, \ldots, \hat{\boldsymbol{\Theta}}^{(K)}$ that result from this algorithm are the JGL estimates of $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$. This algorithm is guaranteed to converge to the global optimum (Boyd *et al.*, 2010). We note that the positive definiteness constraint on the estimated precision matrices is naturally enforced by the update in step (c)(i).

This algorithm requires specification of a penalty ρ controlling the step size and a convergence criterion. In the examples throughout this paper, we use $\rho = 1$ and declare convergence when

$$\sum_k \|\boldsymbol{\Theta}^{(k)}_{(i)} - \boldsymbol{\Theta}^{(k)}_{(i-1)}\|_1 \Big/ \sum_k \|\boldsymbol{\Theta}^{(k)}_{(i-1)}\|_1 < 10^{-5}.$$

Details of the minimization of expression (9) will depend on the form of the convex penalty function P. We note that the task of minimizing equation (9) can be rewritten as

$$\underset{\{\mathbf{Z}\}}{\text{minimize}}\left[\frac{\rho}{2}\sum_{k=1}^{K}\|\mathbf{Z}^{(k)} - \mathbf{A}^{(k)}\|^2_{\mathrm{F}} + P(\{\mathbf{Z}\})\right], \tag{10}$$

where

$$\mathbf{A}^{(k)} = \boldsymbol{\Theta}^{(k)}_{(i)} + \mathbf{U}^{(k)}_{(i-1)}. \tag{11}$$

We shall see in Section 3.2 that, for the FGL and GGL penalties, solving problem (10) is a simple task, regardless of the value of K.

The algorithm given above involves computing the eigendecomposition of a $p \times p$ matrix, which can be computationally demanding when p is large. However, in Section 4, we shall present

two theorems that reveal that, when the values of the tuning parameters λ_1 and λ_2 are large, we can obtain the *exact* solution to the JGL optimization problem without ever computing the eigendecomposition of a $p \times p$ matrix. Therefore, solving the JGL problem is fast even when p is quite large. In Section 8, we shall see that one can perform the FGL with $K = 2$ classes and almost 18000 features in under 2 min.

3.2. Solving problem (10) for the joint graphical lasso

We now consider the problem of solving problem (10) if P is a generalized fused lasso or group lasso penalty.

3.2.1. Solving problem (10) for the fused graphical lasso

If P is the penalty given in equation (5), then problem (10) takes the form

$$\underset{\{\mathbf{Z}\}}{\text{minimize}}\left(\frac{\rho}{2} \sum_{k=1}^{K} \|\mathbf{Z}^{(k)} - \mathbf{A}^{(k)}\|_F^2 + \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |Z_{ij}^{(k)}| + \lambda_2 \sum_{k<k'} \sum_{i,j} |Z_{ij}^{(k)} - Z_{ij}^{(k')}| \right). \tag{12}$$

Now problem (12) is completely separable with respect to each pair of matrix elements (i, j), i.e. one can simply solve, for each (i, j),

$$\underset{Z_{ij}^{(1)},\ldots,Z_{ij}^{(K)}}{\text{minimize}}\left\{ \frac{\rho}{2} \sum_{k=1}^{K} (Z_{ij}^{(k)} - A_{ij}^{(k)})^2 + \lambda_1 \mathbf{1}_{i \neq j} \sum_{k=1}^{K} |Z_{ij}^{(k)}| + \lambda_2 \sum_{k<k'} |Z_{ij}^{(k)} - Z_{ij}^{(k')}| \right\}. \tag{13}$$

This is a special case of the *fused lasso signal approximator* (Hoefling, 2010a) in which there is a fusion between each pair of variables. A very efficient algorithm for this special case, which can be performed in $O\{K \log(K)\}$ operations, is available (Hocking *et al.*, 2011; Hoefling, 2010b; Tibshirani, 2012).

In fact, when $K = 2$, problem (13) has a very simple closed form solution. When $\lambda_1 = 0$, it is easy to verify that the solution to problem (13) takes the form

$$(\hat{Z}_{ij}^{(1)}, \hat{Z}_{ij}^{(2)}) = \begin{cases} (A_{ij}^{(1)} - \lambda_2/\rho, A_{ij}^{(2)} + \lambda_2/\rho) & \text{if } A_{ij}^{(1)} > A_{ij}^{(2)} + 2\lambda_2/\rho, \\ (A_{ij}^{(1)} + \lambda_2/\rho, A_{ij}^{(2)} - \lambda_2/\rho) & \text{if } A_{ij}^{(2)} > A_{ij}^{(1)} + 2\lambda_2/\rho, \\ \left(\dfrac{A_{ij}^{(1)} + A_{ij}^{(2)}}{2}, \dfrac{A_{ij}^{(1)} + A_{ij}^{(2)}}{2} \right) & \text{if } |A_{ij}^{(1)} - A_{ij}^{(2)}| \leqslant 2\lambda_2/\rho. \end{cases} \tag{14}$$

And, when $\lambda_1 > 0$, the solution to problem (13) can be obtained through soft thresholding expression (14) by λ_1/ρ (see Friedman *et al.* (2007)). Here the soft thresholding operator is defined as $S(x, c) = \text{sgn}(x)(|x| - c)_+$, where $a_+ = \max(a, 0)$.

3.2.2. Solving problem (10) for the group graphical lasso

If P is the group lasso penalty (6), then problem (10) takes the form

$$\underset{\{\mathbf{Z}\}}{\text{minimize}}\left\{ \frac{\rho}{2} \sum_{k=1}^{K} \|\mathbf{Z}^{(k)} - \mathbf{A}^{(k)}\|_F^2 + \lambda_1 \sum_{k=1}^{K} \sum_{i \neq j} |Z_{ij}^{(k)}| + \lambda_2 \sum_{i \neq j} \left(\sum_k Z_{ij}^{(k)2} \right)^{1/2} \right\}. \tag{15}$$

First, for all $j = 1, \ldots, p$ and $k = 1, \ldots, K$, it is easy to see that the solution to problem (15) has $\hat{Z}_{jj}^{(k)} = A_{jj}^{(k)}$. And one can show that the off-diagonal elements take the form (Friedman *et al.*, 2010)

$$\hat{Z}_{ij}^{(k)} = S(A_{ij}^{(k)}, \lambda_1/\rho)\left(1 - \frac{\lambda_2}{\rho\left\{\sum_{k=1}^{K} S(A_{ij}^{(k)}, \lambda_1/\rho)^2\right\}^{1/2}}\right)_+, \tag{16}$$

where S denotes the soft thresholding operator.

4. Faster computations for the fused graphical lasso and group graphical lasso

We now present two theorems that lead to substantial computational improvements to the JGL algorithm that was presented in Section 3. Using these theorems, we can inspect the empirical covariance matrices $\mathbf{S}^{(1)}, \ldots, \mathbf{S}^{(K)}$ to determine whether the solution to the JGL optimization problem is block diagonal after some permutation of the features. Then we can simply perform the JGL algorithm on the features within each block separately, to obtain *exactly* the same solution that would have been obtained by applying the algorithm to all p features. This leads to huge speed improvements since it obviates the need ever to compute the eigendecomposition of a $p \times p$ matrix. Our results mirror recent improvements in algorithms for solving the graphical lasso problem (Witten *et al.*, 2011; Mazumder and Hastie, 2012).

For instance, suppose that, for a given choice of λ_1 and λ_2, we determine that the estimated inverse covariance matrices $\hat{\mathbf{\Theta}}^{(1)}, \ldots, \hat{\mathbf{\Theta}}^{(K)}$ are block diagonal, each with the same R blocks, the rth of which contains p_r features, $\Sigma_{r=1}^{R} p_r = p$. Then in each iteration of the JGL algorithm, rather than having to compute the eigendecomposition of K $p \times p$ matrices, we need only to compute eigendecompositions of matrices of dimension $p_1 \times p_1, \ldots, p_R \times p_R$. This leads to a potentially massive reduction in computational complexity from $O(p^3)$ to $\Sigma_{r=1}^{R} O(p_r^3)$.

We begin with a very simple lemma for which the proof follows by inspection of problem (4). The lemma can be extended by induction to any number of blocks.

Lemma 1. Suppose that the solution to the FGL or GGL optimization problem is block diagonal with known blocks, i.e. the features can be reordered in such a way that each estimated inverse covariance matrix takes the form

$$\hat{\mathbf{\Theta}}^{(k)} = \begin{pmatrix} \hat{\mathbf{\Theta}}_1^{(k)} & 0 \\ 0 & \hat{\mathbf{\Theta}}_2^{(k)} \end{pmatrix} \tag{17}$$

where each of $\hat{\mathbf{\Theta}}_1^{(1)}, \ldots, \hat{\mathbf{\Theta}}_1^{(K)}$ has the same dimension. Then, $\hat{\mathbf{\Theta}}_1^{(1)}, \ldots, \hat{\mathbf{\Theta}}_1^{(K)}$ and $\hat{\mathbf{\Theta}}_2^{(1)}, \ldots, \hat{\mathbf{\Theta}}_2^{(K)}$ can be obtained by solving the FGL or GGL optimization problem on just the corresponding set of features.

We now present the key results. Theorems 1 and 2 outline necessary and sufficient conditions for the presence of block diagonal structure in the FGL and GGL optimization problems and are proven in Appendix B.

Theorem 1. Consider the FGL optimization problem with $K = 2$ classes. Let C_1 and C_2 be a non-overlapping partition of the p variables, such that $C_1 \cap C_2 = \emptyset$, $C_1 \cup C_2 = \{1, \ldots, p\}$. The following conditions are necessary and sufficient for the variables in C_1 to be completely disconnected from those in C_2 in each of the resulting network estimates:

(a) $|n_1 S_{ij}^{(1)}| \leqslant \lambda_1 + \lambda_2$ for all $i \in C_1$ and $j \in C_2$,
(b) $|n_2 S_{ij}^{(2)}| \leqslant \lambda_1 + \lambda_2$ for all $i \in C_1$ and $j \in C_2$, and
(c) $|n_1 S_{ij}^{(1)} + n_2 S_{ij}^{(2)}| \leqslant 2\lambda_1$ for all $i \in C_1$ and $j \in C_2$.

Furthermore, if $K > 2$, then

$$|n_k S_{ij}^{(k)}| \leqslant \lambda_1 \qquad \text{for all } i \in C_1, j \in C_2, k = 1, \ldots, K \tag{18}$$

is a sufficient condition for the variables in C_1 to be completely disconnected from those in C_2.

Theorem 2. Consider the GGL optimization problem with $K \geqslant 2$ classes. Let C_1 and C_2 be a non-overlapping partition of the p variables, such that $C_1 \cap C_2 = \emptyset$, $C_1 \cup C_2 = \{1, \ldots, p\}$. Then the following condition is necessary and sufficient for the variables in C_1 to be completely disconnected from those in C_2 in each of the resulting network estimates:

$$\sum_{k=1}^{K} (|n_k S_{ij}^{(k)}| - \lambda_1)_+^2 \leqslant \lambda_2^2 \text{ for all } i \in C_1, j \in C_2. \tag{19}$$

Theorems 1 and 2 allow us to check quickly whether, given a partition of the features C_1 and C_2, the solution to the JGL optimization problem is block diagonal with one block corresponding to features in C_1 and one block corresponding to features in C_2. In practice, for any given (λ_1, λ_2), we can quickly perform the following two-step procedure to identify any block structure in the FGL or GGL solution.

(a) Create $\mathbf{M}$, a $p \times p$ matrix with $M_{jj} = 1$ for $j = 1, \ldots, p$. For $i \neq j$, let $M_{ij} = 0$ if the conditions that are specified in theorem 1 are met for that pair of variables and the FGL penalty is used, or if the condition of theorem 2 is met for that pair of variables and the GGL penalty is used. Otherwise, set $M_{ij} = 1$.
(b) Identify the connected components of the undirected graph whose adjacency matrix is given by $\mathbf{M}$. Note that this can be performed in $O(|\mathbf{M}|)$ operations, where $|\mathbf{M}|$ is the number of non-zero elements in $\mathbf{M}$ (Tarjan, 1972).

Theorems 1 and 2 guarantee that the connected components that are identified in step (b) correspond to distinct blocks in the FGL or GGL solutions. Therefore, we can quickly obtain these solutions by solving the FGL or GGL optimization problems on the submatrices of these K $p \times p$ empirical covariance matrices that correspond to that block diagonal structure. Consequently, we can obtain the *exact* solution to the JGL optimization problem on extremely high dimensional data sets that would otherwise be computationally intractable. For instance, in Section 8 we performed FGL on a gene expression data set with almost 18000 features in under 2 min.

As pointed out by a reviewer, theorems 1 and 2 lead to improvements in speed only if the tuning parameters λ_1 and λ_2 are sufficiently large. We argue that this will in fact be so in most practical applications of the JGL. When network estimation is performed for data exploration and when p is large, only a very sparse network estimate will be useful; otherwise, interpretation of the estimate will be impossible. Even when data exploration is not the end goal of the analysis, large values of λ_1 and λ_2 will generally be used, since most data sets cannot reasonably support estimation of $Kp(p+1)/2$ non-zero parameters when $n \ll p$.

5. Relationship to previous proposals

Several past proposals have been made to estimate graphical models jointly on the basis of observations drawn from distinct conditions. Some proposals have used time series data to define time varying networks in the context of continuous or binary data (Zhou *et al.*, 2008; Song *et al.*, 2009a, b; Ahmed and Xing, 2009; Kolar and Xing, 2009; Kolar *et al.*, 2010).

Guo *et al.* (2011) instead described a likelihood-based method for estimating precision matrices across multiple related classes simultaneously. They employed a hierarchical penalty that forces similar patterns of sparsity across classes, an approach that is similar in spirit to the GGL.

Our FGL and GGL proposals have several advantages over these existing approaches. Methods for estimating time varying networks cannot be easily extended to the setting where the classes lack a natural ordering. The proposal of Guo *et al.* (2011) is a closer precursor to our method and can in fact be stated as an instance of problem (4) with a *hierarchical group lasso penalty*

$$P(\{\boldsymbol{\Theta}\}) = \lambda \sum_{i \neq j} \left(\sum_{k} |\theta_{ij}^{(k)}| \right)^{1/2} \tag{20}$$

that encourages a shared pattern of sparsity across the K classes. But the approach of Guo *et al.* (2011) has some disadvantages relative to the FGL and GGL.

(a) The penalty (20) is not convex, so algorithmic convergence to the global optimum is not guaranteed.
(b) Because penalty (20) is not convex, it is not possible to achieve the improvements in speed that were described in Section 4. Consequently, the proposal of Guo *et al.* (2011) is quite slow relative to our approach, as seen later in Figs 2(e), 4(e) and 5(e), and essentially cannot be applied to very high dimensional data sets.
(c) Unlike the FGL and GGL, it uses just one tuning parameter and cannot control separately the sparsity level and the extent of network similarity.
(d) In cases where we expect edge values as well as network structure to be similar between classes, the FGL is much better suited than the GGL and the proposal of Guo *et al.* (2011), both of which encourage shared patterns of sparsity but do not encourage similarity in the signs and values of the non-zero edges.

The proposal of Guo *et al.* (2011) is included in the simulation study in Section 7.

6. Tuning parameter selection

Network estimation is usually performed to aid exploratory data analysis and hypothesis generation. For these purposes, approaches such as the Akaike information criterion (AIC), Bayesian information criterion and cross-validation may tend to choose models that are too large to be useful, and model selection is better guided by practical considerations, such as network interpretability and stability and the desire for an edge set with a low false discovery rate (Meinshausen and Bühlmann, 2010; Li *et al.*, 2013). Thus, in most cases, we recommend an application-driven selection of tuning parameters to achieve a model that is biologically plausible, sufficiently complex to be interesting and sufficiently sparse to be interpretable and extremely well supported by the data. In fact, network estimation methods will often prove most descriptively useful when run over a variety of tuning parameters, giving the researcher a sense of how easily various edges overcome increasing values of the sparsity penalty and how readily they become shared across networks as the similarity penalty increases. Ideally, the final model would be accompanied by a p-value on each edge or an overall estimate of the edge false discovery rate (FDR), a difficult problem that was addressed by Li *et al.* (2013) in the partial correlation-based network estimation framework and an important goal for research in likelihood-based network estimation methods.

When an objective method of selecting tuning parameters is desirable, one can select tuning parameters for the JGL by using an approximation of the AIC:

$$\mathrm{AIC}(\lambda_1, \lambda_2) = \sum_{k=1}^{K} [n_k \mathrm{tr}(\mathbf{S}^{(k)} \hat{\boldsymbol{\Theta}}^{(k)}_{\lambda_1,\lambda_2}) - n_k \log\{\det(\hat{\boldsymbol{\Theta}}^{(k)}_{\lambda_1,\lambda_2})\} + 2E_k], \tag{21}$$

where $\hat{\boldsymbol{\Theta}}^{(k)}_{\lambda_1,\lambda_2}$ is the inverse covariance matrix estimated on the kth class of data by using the tuning parameters λ_1 and λ_2, and E_k is the number of non-zero elements in $\hat{\boldsymbol{\Theta}}^{(k)}_{\lambda_1,\lambda_2}$. A grid search can then be performed to select λ_1 and λ_2 that minimize the $\mathrm{AIC}(\lambda_1, \lambda_2)$ score. The simulation study in Section 7 suggests that this criterion tends to select models whose Kullback–Leibler divergence dKL from the true model is low. When the number of variables p is very large, computing $\mathrm{AIC}(\lambda_1, \lambda_2)$ over a range of values for λ_1 and λ_2 may prove computationally onerous. If this is so, we suggest a dense search over λ_1 while holding λ_2 at a fixed low value, followed by a quick search over λ_2, holding λ_1 at the selected value.

7. Simulation study

We compare the performances of the FGL and GGL with two existing methods, the graphical lasso and the proposal of Guo *et al.* (2011), in Section 7.1. When applying the graphical lasso, networks are fitted for each class separately. We investigate the effects of n and p on the FGL and GGL's performances in Section 7.2. Additional simulation results are presented in Appendix C.

The effects of the FGL and GGL penalties vary with the sample size. For ease of presentation of the simulation study results, we multiply the reported tuning parameters λ_1 and λ_2 by the sample size of each class before performing the JGL.

To ease interpretation, we reparameterize the GGL penalties in our simulation study. The motivation is to summarize the regularization for 'sparsity' and for 'similarity' separately. In the FGL, this is nicely achieved by just using λ_1 and λ_2, as the former drives network sparsity and the latter drives network similarity. In contrast, in the GGL, both tuning parameters contribute to sparsity: λ_1 drives individual network edges to zero whereas λ_2 simultaneously drives network edges to zero across all K network estimates. We reparameterize our simulation results for the GGL in terms of

$$\omega_1 = \lambda_1 + \frac{1}{\sqrt{2}}\lambda_2$$

and

$$\omega_2 = \frac{1}{\sqrt{2}}\lambda_2 \Big/ \left(\lambda_1 + \frac{1}{\sqrt{2}}\lambda_2\right),$$

which we found respectively to reflect the levels of sparsity and similarity regularization.

7.1. Performance as a function of tuning parameters

7.1.1. Simulation set-up

In this simulation, we consider a three-class problem. We first generate three networks with $p = 500$ features belonging to 10 equally sized unconnected subnetworks, each with a power law degree distribution. Power law degree distributions are thought to mimic the structure of biological networks (Chen and Sharp, 2004) and are generally more difficult to estimate than simpler structures (Peng *et al.*, 2009). Of the 10 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 topology of the networks generated is shown in Fig. 6 in Appendix D.

Given a network structure, we generate a covariance matrix for the first class as follows (Peng *et al.*, 2009). We create a $p \times p$ matrix with 1s on the diagonal, 0s on elements not corresponding to network edges and values from a uniform distribution with support on $\{[-0.4, -0.1] \cup [0.1, 0.4]\}$ on elements corresponding to edges. To ensure positive definiteness, we divide each off-diagonal element by 1.5 times the sum of the absolute values of off-diagonal elements in its row. Finally, we average the matrix with its transpose, achieving a symmetric, positive definite matrix $\mathbf{A}$. We then create the (i, j) element of $\mathbf{\Sigma}_1$ as

$$d_{ij}(\mathbf{A}^{-1})_{ij}\{(\mathbf{A}^{-1})_{ii}(\mathbf{A}^{-1})_{jj}\}^{-1/2},$$

where $d_{ij} = 0.6$ if $i \neq j$ and $d_{ij} = 1$ if $i = j$. We create $\mathbf{\Sigma}_2$ equal to $\mathbf{\Sigma}_1$; then we reset one of its 10 subnetwork blocks to the identity. We create $\mathbf{\Sigma}_3$ equal to $\mathbf{\Sigma}_2$, and reset an additional subnetwork block to the identity. Finally, for each class we generate independent, identically distributed samples from an $N(\mathbf{0}, \mathbf{\Sigma}_k)$ distribution.

We present two additional simulations studies involving two-class data sets in Appendix C. The first additional simulation uses the same network structure as described above, and the second uses a single power law network with no block structure.

7.1.2. Simulation results

Our first set of simulations illustrates the effect of varying tuning parameters on the performances of the FGL and GGL. We generated 100 three-class data sets with $p = 500$ features and $n = 150$ observations per class, as described in Section 7.1. Class 1's network had 490 edges, class 2's network is missing 49 of those edges, and class 3's network is missing an additional 49 edges. Fig. 2 characterizes the average performance of the methods over the 100 data sets. In each plot, the curves for the FGL and GGL indicate the results obtained with a single value of the similarity tuning parameters λ_2 and ω_2. The graphical lasso and the proposal of Guo *et al.* (2011) are included in the comparisons.

Fig. 2(a) displays the number of true edges selected against the number of false edges selected. We say that an edge (i, j) in the kth network is selected if $\hat{\theta}_{ij}^{(k)} \neq 0$, and we say that the edge is true if $(\mathbf{\Sigma}_k^{-1})_{ij} \neq 0$ and false if $(\mathbf{\Sigma}_k^{-1})_{ij} = 0$. As the sparsity tuning parameters λ_1 and ω_1 decrease, the number of edges selected increases. At many values of the similarity tuning parameter λ_2, the FGL dominates the other methods. At some choices of the similarity tuning parameter ω_2, the GGL performs as well as the method of Guo *et al.* (2011). The FGL, GGL and the proposal of Guo *et al.* (2011) dominate the graphical lasso.

Fig. 2(b) displays the sum of squared errors between estimated edge values and true edge values: $\Sigma_{k=1}^{K} \Sigma_{i \neq j} \{\hat{\theta}_{ij}^{(k)} - (\mathbf{\Sigma}_k^{-1})_{ij}\}^2$. Unlike the proposal of Guo *et al.* (2011), the FGL, GGL and the graphical lasso tend to overshrink edge values towards zero owing to the use of convex penalty functions. Thus, although the FGL and GGL attain sum of squared errors values that are as low as those of Guo *et al.* (2011), they do so when estimating much larger networks. When simultaneous edge selection and estimation are desired, it may be useful to run the FGL or GGL once and then to rerun them with smaller penalties on the selected edges, as in Meinshausen (2007).

Fig. 2(c) evaluates each method's success in detecting *differential edges*, or edges that differ between classes. We say that the (i, j)th edge is estimated to be differential between the kth and k'th networks if $\hat{\theta}_{ij}^{(k)} \neq \hat{\theta}_{ij}^{(k')}$, and we say that it is truly differential if $(\mathbf{\Sigma}_k^{-1})_{ij} \neq (\mathbf{\Sigma}_{k'}^{-1})_{ij}$

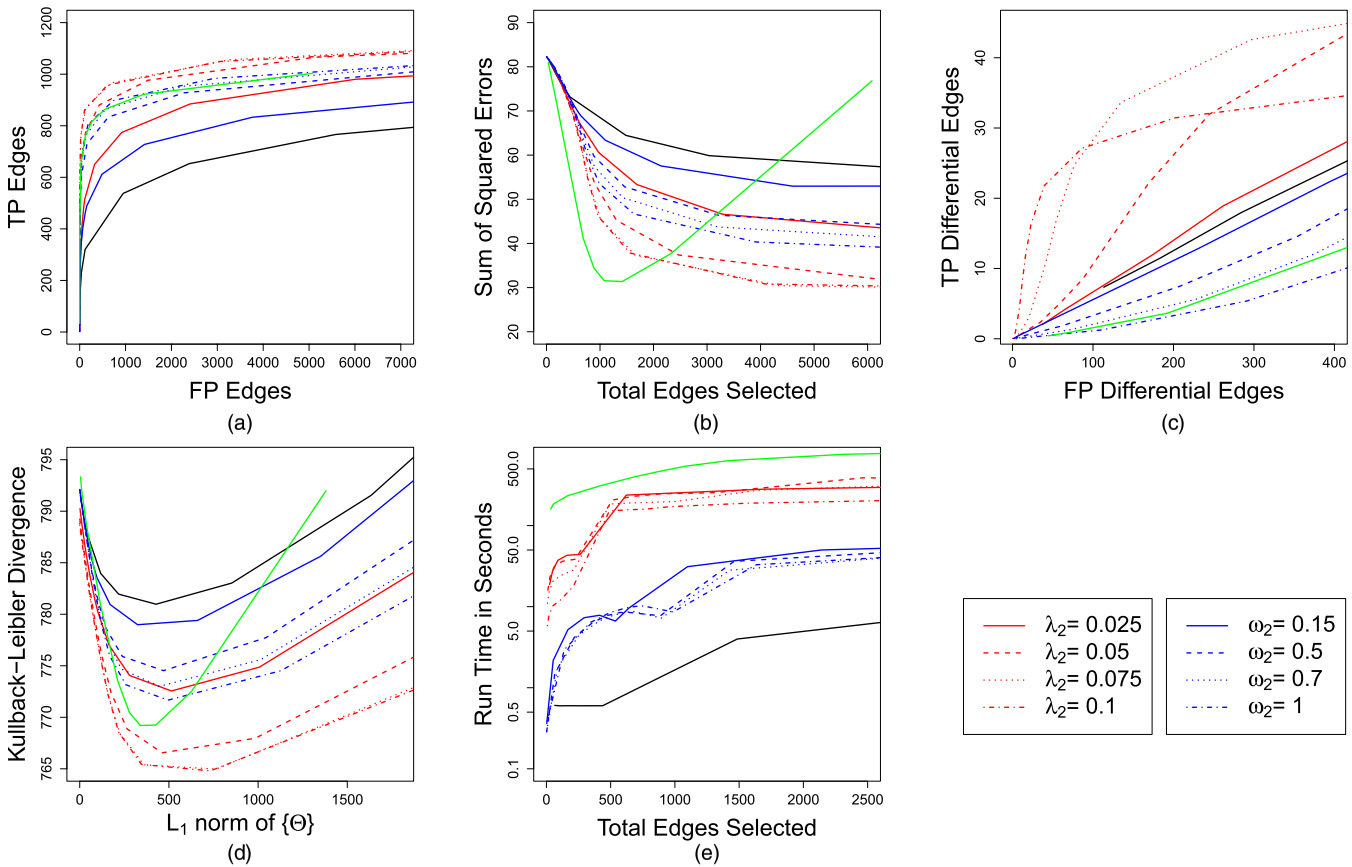

Fig. 2. Performance of the FGL (———), GGL (———), the method of Guo *et al.* (2011) (———) and the graphical lasso (———) on simulated data with 150 observations in each of three classes and 500 features: (a) number of edges correctly identified to be non-zero (true positive edges) *versus* number of edges incorrectly identified to be non-zero (false positive edges); (b) sum of squared errors in edge values *versus* the total number of edges estimated to be non-zero; (c) number of edges correctly found to have values differing between classes (true positive differential edges) *versus* the number of edges incorrectly found to have values differing between classes (false positive differential edges); (d) dKL for the estimated models from the true models *versus* the l_1-norm of the off-diagonal entries of the estimated precision matrices; (e) running time *versus* the number of non-zero edges estimated (note the use of a log-scale on the vertical axis)

and falsely differential if $(\Sigma_k^{-1})_{ij} = (\Sigma_{k'}^{-1})_{ij}$. For the FGL, the number of differential edges is computed as the number of pairs $k < k'$, $i < j$, such that $\hat{\theta}_{ij}^{(k)} \neq \hat{\theta}_{ij}^{(k')}$. Since the GGL, the proposal of Guo *et al.* (2011) and the graphical lasso cannot yield edges that are exactly identical across classes, for those approaches the number of differential edges is computed as the number of pairs $k < k'$, $i < j$, such that $|\hat{\theta}_{ij}^{(k)} - \hat{\theta}_{ij}^{(k')}| > 10^{-2}$. The number of true positive differential edges is plotted against the number of false positive differential edges. By controlling the total number of non-zero edges, the sparsity tuning parameters λ_1 and ω_1 have a large effect on the number of edges that are estimated to differ between the two networks. The FGL yields fewer false positive results than the competing methods, since it shrinks between-class differences in edge values to 0. Since neither the GGL nor the method of Guo *et al.* (2011) is designed to shrink edge values towards each other, by this measure neither method outperforms even the graphical lasso.

Fig. 2(d) displays the sum of the dKLs of the estimated distributions from the true distributions, as a function of the l_1-norm of the off-diagonal elements of the estimated precision matrices, i.e. $\Sigma_k \Sigma_{i \neq j} |\hat{\theta}_{ij}^{(k)}|$. The dKL from the multivariate normal model with inverse covariance estimates $\hat{\boldsymbol{\Theta}}^{(1)}, \ldots, \hat{\boldsymbol{\Theta}}^{(k)}$ to the multivariate normal model with the true precision matrices $\boldsymbol{\Sigma}_1^{-1}, \ldots, \boldsymbol{\Sigma}_K^{-1}$ is

$$\frac{1}{2} \sum_{k=1}^{K} [-\log\{\det(\hat{\boldsymbol{\Theta}}^{(k)} \boldsymbol{\Sigma}_k)\} + \operatorname{tr}(\hat{\boldsymbol{\Theta}}^{(k)} \boldsymbol{\Sigma}_k)].$$

At most values of λ_2, the FGL attains a lower dKL than the other methods, followed by the method of Guo *et al.* (2011) and then by the GGL. The graphical lasso has the worst performance, since it estimates each network separately.

Fig. 2(e) compares the methods' running times. The computation time (in seconds) is plotted against the total number of non-zero edges estimated. The graphical lasso is fastest, but the FGL and GGL are much faster than the proposal of Guo *et al.* (2011), owing to the results from Section 4. Timing comparisons were performed on an Intel Xeon x5680 3.3 GHz processor. It is worth mentioning that the FGL algorithm is much faster in problems with only two classes, since in that case there is a closed form solution to the generalized fused lasso problem (Section 3.2). This can be seen in Figs 4(e) and 5(e) in Appendix C.

We examined the details of the models from the FGL, GGL, the method of Guo *et al.* (2011) and the graphical lasso with tuning parameters selected as described in Section 6. The performance of these models is detailed in Table 1. The AIC-selected FGL and GGL models outperform the AIC-selected models from the earlier methods. The AIC selects a larger model

Table 1. Performance of models selected by the AIC†

Method	*Tuning parameters*	*AIC*	*dKL*	*TPE*	*FPE*	*TPDE*	*FPDE*
FGL	$\lambda_1 = 0.175$, $\lambda_2 = 0.025$	1465	774	884	2406	77	4977
GGL	$\omega_1 = 0.225$, $\omega_2 = 1$	1470	776	898	736	53	1456
Graphical lasso	$\lambda = 0.2$	1471	781	766	5578	85	11609
Guo *et al.* (2011)	$\lambda = 0.4$	1338	791	1003	5080	89	8992

†For each method, the tuning parameters selected by the AIC are displayed, as are the average values over 100 iterations of the following performance metrics: the AIC, the Kullback–Leibler divergence from the true model, dKL, numbers of true and false positive edges, TPE and FPE, and numbers of true and false positive differential edges, TPDE and FPDE.

for the method of Guo *et al*. (2011) than it does for the FGL and GGL. For all methods, the AIC appears to select models with low Kullback–Leibler divergences from the truth but with greater numbers of edges than would be ideal for accurate hypothesis generation. The AIC selected a much smaller model for the GGL than for the other methods, achieving by far the best edge estimation performance.

7.2. Performance as a function of n and p

We now evaluate the effect of sample size n and dimension p on the performances of the FGL and GGL.

7.2.1 Simulation set-up

We generate a pair of networks with $p = 500$ much as described in Section 7.1.1, but with $K = 2$ instead of $K = 3$. The first network has 10 equal-sized components with power law degree distributions, and the second network is identical to the first in both edge identity and value, but with two components removed.

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. We generate covariance matrices from the networks exactly as described in Section 7.1.1.

7.2.2. Simulation results

For both the $p = 500$ and the $p = 1000$ networks, we simulate 100 data sets with $n = 50$, $n = 200$ and $n = 500$ samples in each class. We run the FGL with $\lambda_1 = 0.2$ and $\lambda_2 = 0.1$, and the GGL with $\lambda_1 = 0.05$ and $\lambda_2 = 0.25$. These tuning parameter values were chosen because they were in the range of successful values in the similar simulation set-up of Section 7.1.2 (see Fig. 2) and therefore provide a good setting under which to evaluate the effects of n and p on the FGL and GGL. Table 2 displays the dKL of each estimated model as well as the sensitivity and FDR

Table 2. Performances as a function of n and p†

Method	p	n	*Mean dKL*	*Means for the following types of detection:*			
				Edge sensitivity	*Edge FDR*	*Differential edge sensitivity*	*Differential edge 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

†Means over 100 replicates are shown for dKL, and for sensitivity and the FDR of detection of edges and differential edge detection.

in both edge detection and differential edge detection. In this simulation setting, the accuracy of covariance estimation (as measured by dKL) improves significantly from $n = 50$ to $n = 200$, and it improves only marginally with a further increase to $n = 500$. Detection of edges improves throughout the range of ns sampled: for both the FGL and the GGL, sensitivity improves slightly with increased sample size, and the FDR decreases dramatically. Accurate detection of edge differences is a more difficult goal, though the FGL succeeds in it at higher sample sizes.

8. Analysis of lung cancer microarray data

We applied the FGL to a data set containing 22283 microarray-derived gene expression measurements from large airway epithelial cells sampled from 97 patients with lung cancer and 90 controls (Spira *et al.*, 2007). The data are publicly available from the *Gene Expression Omnibus* (Barrett *et al.*, 2005) at accession number GDS2771. We omitted genes with standard deviations in the bottom 20% since a greater share of their variance is probably attributable to non-biological noise. The remaining genes were standardized to have mean 0 and standard deviation 1 within each class. To avoid disparate levels of sparsity between the classes and to prevent the larger class from dominating the estimated networks, we weighted each class equally instead of by sample size in problem (4). Since our goal was data visualization and hypothesis generation, we chose a high value for the sparsity tuning parameter, $\lambda_1 = 0.95$, to yield very sparse network estimates. We ran the FGL with a range of λ_2-values to identify the edges that differed most strongly, and we settled on $\lambda_2 = 0.005$ as providing the most interpretable results. This application-driven choice of tuning parameters is appropriate when the goal of network estimation is description and hypothesis generation. A full analysis of this data set would involve examination of network estimates across a range of tuning parameters. Application of theorem 1 revealed that only 278 genes were connected to any other gene by using the chosen tuning parameters. Identification of block diagonal structure by using theorem 1 and application of the FGL algorithm took less than 2 min. (Note that this data set is so large that it would be computationally prohibitive to apply the proposal of Guo *et al.* (2011)!) The FGL estimated 134 edges shared between the two networks, 202 edges present only in the cancer network and 18 edges present only in the normal tissue network. The results are displayed in Fig. 3.

The estimated networks contain many two-gene subnetworks that are common to both classes, a few small subnetworks and one large subnetwork specific to tumour cells. Reassuringly, 45% of edges, including almost all of the two-gene subnetworks, connect multiple probes for the same gene. Many other edges connect genes that are obviously related, that are involved in the same biological process or that even code for components of the same enzyme. Examples include TUBA1B and TUBA1C, PABPC1 and PABPC3, HLA-B and HLA-G, and SERPINB3 and SERPINB4. Recovery of these pairs suggests that the FGL (and other network analysis tools) can generate high quality hypotheses about gene coregulation and functional interactions. This increases our confidence that some of the non-obvious two-gene subnetworks that are detected in this analysis may merit further investigation. Examples include DAZAP2 and TCP1, PRKAR1A and CALM3, and BCLAF1 and SERPB1. A complete list of subnetworks detected is available in the on-line supplementary materials.

The small black and green network in Fig. 3 suggests an interesting phenomenon. It contains multiple probes for two haemoglobin genes, HBA2 and HBB. In the normal tissue network, the probes for these genes are heavily interconnected both within and between the genes. In the tumour cells, although edges between HBA2 probes and between HBB probes are preserved, no edges connect the two genes. The abundance of connections between the two genes in healthy

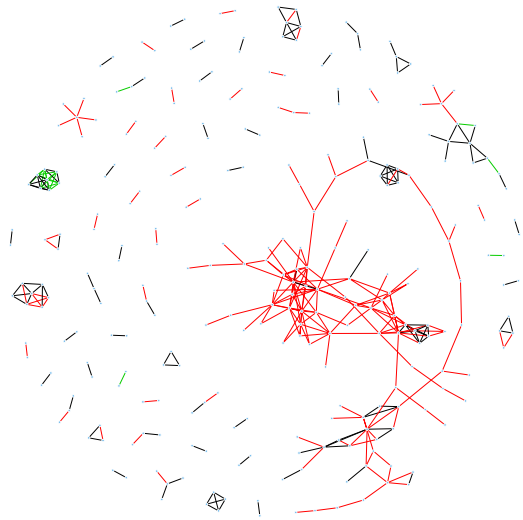

Fig. 3. Conditional dependence networks inferred from 17772 genes in normal and cancerous lung cells (278 genes have non-zero edges in at least one of the two networks): ———, edges common to both classes; ———, tumour-specific edges; ———, normal-specific edges

cells and the absence of connections in tumour cells may indicate a possible direction of future investigation.

The most promising results of this analysis arise from the large subnetwork (104 nodes for 84 unique genes) that is unique to tumour cells. Many of the subnetwork's genes are involved in constructing ribosomes, including RPS8, RPS23, RPS24, RPS7P11, RPL3, RPL5, RPL10A, RPL14P1, RPL15, RPL17, RPL30 and RPL31. Other genes in the subnetwork further involve ribosome functioning: SRP14 and SRP9L1 are involved in recruiting proteins from ribosomes into the endoplasmic reticulum and NACA inhibits the SRP pathway. Thus this subnetwork portrays a detailed web of relationships that is consistent with known biology. More interestingly, this network also contains two genes in the RAS oncogene family: RAB1A and RAB11A. Genes in this family have been linked to many types of cancer and are considered promising targets for therapeutics (Adjei, 2008). These genes' connections with ribosome activity in the tumour samples may indicate a relationship that is common to an important subset of cancers. Many other genes belong to this network, each indicating a potentially novel interaction in cancer biology.

9. Discussion

We have introduced the JGL, a method for estimating sparse inverse covariance matrices on the basis of observations drawn from distinct but related classes. We describe an ADMM algorithm for the solution to the JGL problem with any convex penalty function, and we provide explicit and efficient solutions for two useful penalty functions. Our algorithm is tractable on very large data sets (more than 20 000 features) and usually converges in seconds for smaller problems (500 features). Our joint estimation methods outperform competing approaches over a range of simulated data sets.

In the JGL optimization problem (4), the contribution of each class to the penalized log-likelihood is weighted by its size; consequently, the largest class can have outsize influence on

the estimated networks. By omitting the n_k-term in problem (4), it is possible to weight the classes equally to prevent a single class from dominating estimation.

We note that the FGL and GGL's reliance on two tuning parameters is a strength rather than a drawback: unlike the proposal of Guo *et al*. (2011), which involves a single tuning parameter that controls both sparsity and similarity, in performing the FGL and GGL one can vary separately the amount of similarity and sparsity to enforce in the network estimates.

The JGL has potential applications beyond those discussed in this paper. For instance, one could use it to shrink multiple classes' precision matrices towards each other to define a classifier that is intermediate between quadratic discriminant analysis and linear discriminant analysis (Hastie *et al*., 2009). In fact, a similar approach has been taken in recent work (Simon and Tibshirani, 2011). In the unsupervised setting, it can be used in the maximization step of Gaussian model-based clustering to reduce the variance that is associated with estimating a separate covariance matrix for each cluster.

The FGL and GGL are implemented in the R package `JGL`, which is available on the Comprehensive R Archive Network: `http://cran.r-project.org/`.

Acknowledgements

We thank two reviewers, the Associate Editor and the Joint Editor for useful comments that substantially improved this paper; Noah Simon, Holger Hoefling, Jacob Bien and Ryan Tibshirani for helpful conversations and for sharing with us unpublished results, and Jian Guo and Ji Zhu for providing software for the proposal in Guo *et al*. (2011). We thank Karthik Mohan, Mike Chung, Su-In Lee, Maryam Fazel and Seungyeop Han for helpful comments. PD and PW are supported by National Institutes of Health grant 1R01GM082802. PW is also supported by National Institutes of Health grants P01CA53996 and U24CA086368. DW is supported by National Institutes of Health grant DP5OD009145.

Appendix A: Modifying the joint graphical lasso to work on the scale of partial correlations

A reviewer suggested that, under some circumstances, it may be preferable to encourage the K networks to have shared partial correlations, rather than shared precision matrices. Below, we describe a simple approach for extending our FGL proposal to work on the scale of partial correlations. A similar approach can be taken to extend the GGL. The extension relies on two insights.

(a) $\rho_{ij} = -\sigma^{ij}/\sqrt{(\sigma^{ii}\sigma^{jj})}$, where ρ_{ij} is the true partial correlation between the ith and jth features, and where σ^{ij} is the (i, j)th entry of the true precision matrix.

(b) The algorithm for solving the FGL optimization problem can easily be modified to make use of the penalty function

$$P(\{\boldsymbol{\Theta}\}) = \sum_{k=1}^{K} \sum_{i \neq j} \lambda_{1,ij} |\theta_{ij}^{(k)}| + \sum_{k<k'} \sum_{i,j} \lambda_{2,ij} |\theta_{ij}^{(k)} - \theta_{ij}^{(k')}|, \tag{22}$$

where $\lambda_{t,ij} = \lambda_t/\sqrt{(\hat{\sigma}^{ii}\hat{\sigma}^{jj})}$, $t = 1, 2$, and where $\hat{\sigma}^{ii}$ is an estimate of the ith diagonal element of the K precision matrices. (Here, we assume that the K precision matrices have shared diagonal elements.) The estimate $\{\hat{\sigma}^{ii}\}$ can be obtained in various ways, for instance by performing the graphical lasso on the samples from all K data sets together. Then this approach will effectively result in applying a generalized fused lasso penalty to the partial correlations for the K classes.

Appendix B: Proofs of theorems 1 and 2

B.1. Preliminaries to proofs of theorems 1 and 2

We begin with a few comments on subgradients. The subgradient of $|\theta_{ij}^{(k)}|$ with respect to $\theta_{ij}^{(k)}$ equals

$$\begin{cases} 1 & \text{if } \theta_{ij}^{(k)} > 0, \\ -1 & \text{if } \theta_{ij}^{(k)} < 0, \\ a & \text{if } \theta_{ij}^{(k)} = 0 \end{cases}$$

for some $a \in [-1, 1]$. The subgradient of $|\theta_{ij}^{(k)} - \theta_{ij}^{(k')}|$ with respect to $(\theta_{ij}^{(k)}, \theta_{ij}^{(k')})$ for $k \neq k'$ equals $(d, -d)$, where

$$d = \begin{cases} 1 & \text{if } \theta_{ij}^{(k)} > \theta_{ij}^{(k')}, \\ -1 & \text{if } \theta_{ij}^{(k)} < \theta_{ij}^{(k')}, \\ a & \text{if } \theta_{ij}^{(k)} = \theta_{ij}^{(k')} \end{cases}$$

for some $a \in [-1, 1]$. Finally, the subgradient of $\{\sum_{k=1}^{K} (\theta_{ij}^{(k)})^2\}^{1/2}$ with respect to $(\theta_{ij}^{(1)}, \ldots, \theta_{ij}^{(K)})$ is given by

$$\begin{aligned} &(\theta_{ij}^{(1)}, \ldots, \theta_{ij}^{(K)}) \Big\{ \sum_{k=1}^{K} (\theta_{ij}^{(k)})^2 \Big\}^{-1/2} && \text{if } \sum_{k=1}^{K} (\theta_{ij}^{(k)})^2 > 0, \\ &(\Upsilon_{1,ij}, \ldots, \Upsilon_{K,ij}) && \text{if } \theta_{ij}^{(1)} = \ldots = \theta_{ij}^{(K)} = 0 \end{aligned}$$

for some $\Upsilon_{1,ij}, \ldots, \Upsilon_{K,ij}$ such that $\Sigma_{k=1}^{K} \Upsilon_{k,ij}^2 \leqslant 1$.

To prove theorem 1, we shall use the following lemma.

Lemma 2. The following two sets of conditions are equivalent.

(a) $|n_1 S_1| \leqslant \lambda_1 + \lambda_2$, $|n_2 S_2| \leqslant \lambda_1 + \lambda_2$ and $|n_1 S_1 + n_2 S_2| \leqslant 2\lambda_1$.
(b) There exist $\Gamma_1, \Gamma_2, \Upsilon \in [-1, 1]$ such that $-n_1 S_1 - \lambda_1 \Gamma_1 - \lambda_2 \Upsilon = 0$, and $-n_2 S_2 - \lambda_1 \Gamma_2 + \lambda_2 \Upsilon = 0$.

Proof. We shall begin by proving that set (b) implies set (a), and then prove that set (a) implies set (b).

Proof that set (b) implies set (a): first, $-n_1 S_1 - \lambda_1 \Gamma_1 - \lambda_2 \Upsilon = 0$ implies that $|n_1 S_1| \leqslant \lambda_1 + \lambda_2$, since $\Gamma_1, \Upsilon \in [-1, 1]$. Similarly, $-n_2 S_2 - \lambda_1 \Gamma_2 + \lambda_2 \Upsilon = 0$ implies that $|n_2 S_2| \leqslant \lambda_1 + \lambda_2$. Finally, summing the two equations in set (b) reveals that $n_1 S_1 + n_2 S_2 = -\lambda_1(\Gamma_1 + \Gamma_2)$, which implies that $|n_1 S_1 + n_2 S_2| \leqslant 2\lambda_1$.

Proof that set (a) implies set (b): without loss of generality, assume that $n_1 S_1 \geqslant n_2 S_2$. We split the proof into two cases.

(a) *Case 1*, $n_1 S_1 - n_2 S_2 < 2\lambda_2$: let $\Gamma_1 = \Gamma_2 = (-n_1 S_1 - n_2 S_2)/2\lambda_1$, and $\Upsilon = (-n_1 S_1 + n_2 S_2)/2\lambda_2$.
First, note that by conditions (a) we know that $|n_1 S_1 + n_2 S_2| \leqslant 2\lambda_1$. Therefore, $\Gamma_1, \Gamma_2 \in [-1, 1]$. Second, note that case 1's assumption that $n_1 S_1 - n_2 S_2 < 2\lambda_2$ implies that $\Upsilon \in [-1, 1]$. Finally, we see by inspection that $-n_1 S_1 - \lambda_1 \Gamma_1 - \lambda_2 \Upsilon = 0$, and $-n_2 S_2 - \lambda_1 \Gamma_2 + \lambda_2 \Upsilon = 0$.

(b) *Case 2*, $n_1 S_1 - n_2 S_2 \geqslant 2\lambda_2$: let $\Gamma_1 = (-n_1 S_1 + \lambda_2)/\lambda_1$, $\Gamma_2 = (-n_2 S_2 - \lambda_2)/\lambda_1$, and $\Upsilon = -1$. Then, by inspection, $-n_1 S_1 - \lambda_1 \Gamma_1 - \lambda_2 \Upsilon = 0$, and $-n_2 S_2 - \lambda_1 \Gamma_2 + \lambda_2 \Upsilon = 0$.
It remains to show that $\Gamma_1, \Gamma_2, \Upsilon \in [-1, 1]$. Trivially, $\Upsilon = -1 \in [-1, 1]$. From our assumption that $|n_1 S_1| \leqslant \lambda_1 + \lambda_2$, we know that $\Gamma_1 = (-n_1 S_1 + \lambda_2)/\lambda_1 \geqslant -1$. Moreover, by the assumptions that $n_1 S_1 - n_2 S_2 \geqslant 2\lambda_2$ and $|n_1 S_1 + n_2 S_2| \leqslant 2\lambda_1$, we have that

$$\Gamma_1 = \frac{-n_1 S_1 + \lambda_2}{\lambda_1} \leqslant \frac{-n_1 S_1 + \lambda_2 \{(n_1 S_1 - n_2 S_2)/2\lambda_2\}}{\lambda_1} = \frac{-n_1 S_1 - n_2 S_2}{2\lambda_1} \leqslant 1. \tag{23}$$

Therefore $\Gamma_1 \in [-1, 1]$.
By the assumption that $|n_2 S_2| \leqslant \lambda_1 + \lambda_2$, we know that $\Gamma_2 = (-n_2 S_2 - \lambda_2)/\lambda_1 \leqslant 1$. From the assumptions that $n_1 S_1 - n_2 S_2 \geqslant 2\lambda_2$ and $|n_1 S_1 + n_2 S_2| \leqslant 2\lambda_1$, we have that

$$\Gamma_2 = \frac{-n_2 S_2 - \lambda_2}{\lambda_1} \geqslant \frac{-n_2 S_2 - \lambda_2 \{(n_1 S_1 - n_2 S_2)/2\lambda_2\}}{\lambda_1} = \frac{-n_1 S_1 - n_2 S_2}{2\lambda_1} \geqslant -1. \tag{24}$$

Therefore $\Gamma_2 \in [-1, 1]$.

Thus we conclude that set (a) implies set (b), and our proof of lemma 2 is complete. □

We shall make use of the following lemma to prove theorem 2.

Lemma 3. The following two conditions are equivalent.

(a) There exist scalars $a_1, \ldots, a_K$ such that $\Sigma_{k=1}^{K} a_k^2 \leqslant 1$ and $n_k |S_k| \leqslant \lambda_1 + \lambda_2 a_k$ for all $k = 1, \ldots, K$.

(b) There exist scalars $\Gamma_1, \ldots, \Gamma_K \in [-1, 1]$ and $\Upsilon_1, \ldots, \Upsilon_K$ such that $\sum_{k=1}^K \Upsilon_k^2 \leqslant 1$ and $n_k S_k + \lambda_1 \Gamma_k + \lambda_2 \Upsilon_k = 0$ for $k = 1, \ldots, K$.

Proof. We shall begin by proving that condition (b) implies condition (a) and then show that condition (a) implies condition (b).

Proof that condition (b) implies condition (a): by condition (b), $n_k|S_k| = |\lambda_1\Gamma_k + \lambda_2\Upsilon_k| \leqslant \lambda_1|\Gamma_k| + \lambda_2|\Upsilon_k| \leqslant \lambda_1 + \lambda_2|\Upsilon_k|$. Letting $a_k = |\Upsilon_k|$, the result holds.

Proof that condition (a) implies condition (b): let Γ_k and Υ_k take the following forms, for $k = 1, \ldots, K$,

$$\Gamma_k = \begin{cases} -1 & \text{if } n_k S_k \geqslant \lambda_1, \\ -n_k S_k/\lambda_1 & \text{if } -\lambda_1 < n_k S_k < \lambda_1, \\ 1 & \text{if } n_k S_k \leqslant -\lambda_1, \end{cases} \tag{25}$$

$$\Upsilon_k = \begin{cases} (-n_k S_k + \lambda_1)/\lambda_2 & \text{if } n_k S_k > \lambda_1, \\ 0 & \text{if } -\lambda_1 \leqslant n_k S_k \leqslant \lambda_1, \\ (-n_k S_k - \lambda_1)/\lambda_2 & \text{if } n_k S_k < -\lambda_1. \end{cases} \tag{26}$$

First, we note by inspection that $\Gamma_k \in [-1, 1]$ and that $n_k S_k + \lambda_1\Gamma_k + \lambda_2\Upsilon_k = 0$ for $k = 1, \ldots, K$. It remains to show that $\sum_{k=1}^K \Upsilon_k^2 \leqslant 1$. Specifically, we shall show that $\Upsilon_k^2 \leqslant a_k^2$ for $k = 1, \ldots, K$. To see why this is so, note that if $-\lambda_1 < n_k S_k < \lambda_1$ then $0 = \Upsilon_k^2 \leqslant a_k^2$. And, if $n_k S_k > \lambda_1$ or $n_k S_k < -\lambda_1$, then

$$\Upsilon_k^2 = \left(\frac{n_k|S_k| - \lambda_1}{\lambda_2} \right)^2 \leqslant a_k^2.$$

B.2. Proof of theorem 1

We first consider the claim for the case $K = 2$. By the Karush–Kuhn–Tucker (see for example Boyd and Vandenberghe (2004)) conditions, a necessary and sufficient set of conditions for $\{\boldsymbol{\Theta}\}$ to be the solution to the JGL problem is that

$$\begin{aligned} 0 &= n_1(\boldsymbol{\Theta}^{(1)})^{-1} - n_1\mathbf{S}^{(1)} - \lambda_1\boldsymbol{\Gamma}_1 - \lambda_2\boldsymbol{\Upsilon}, \\ 0 &= n_2(\boldsymbol{\Theta}^{(2)})^{-1} - n_2\mathbf{S}^{(2)} - \lambda_1\boldsymbol{\Gamma}_2 + \lambda_2\boldsymbol{\Upsilon}, \end{aligned} \tag{27}$$

where $\Gamma_{1,ij}$ is the subgradient of $|\theta_{ij}^{(1)}|$ with respect to $\theta_{ij}^{(1)}$, $\Gamma_{2,ij}$ is the subgradient of $|\theta_{ij}^{(2)}|$ with respect to $\theta_{ij}^{(2)}$ and Υ_{ij} is the subgradient of $|\theta_{ij}^{(1)} - \theta_{ij}^{(2)}|$ with respect to $\theta_{ij}^{(1)}$.

Let C_1 and C_2 be a partition of the p variables into two non-overlapping sets, with $C_1 \cap C_2 = \emptyset$ and $C_1 \cup C_2 = \{1, \ldots, p\}$. Consider the matrices

$$\begin{aligned} \boldsymbol{\Theta}^{(1)} &= \begin{pmatrix} \boldsymbol{\Theta}_1^{(1)} & 0 \\ 0 & \boldsymbol{\Theta}_2^{(1)} \end{pmatrix}, \\ \boldsymbol{\Theta}^{(2)} &= \begin{pmatrix} \boldsymbol{\Theta}_1^{(2)} & 0 \\ 0 & \boldsymbol{\Theta}_2^{(2)} \end{pmatrix}, \end{aligned} \tag{28}$$

where $\boldsymbol{\Theta}_1^{(1)}$ and $\boldsymbol{\Theta}_1^{(2)}$ solve the JGL problem on the features in C_1, and $\boldsymbol{\Theta}_2^{(1)}$ and $\boldsymbol{\Theta}_2^{(2)}$ solve the JGL problem on the features in C_2. By inspection of equations (27), $\boldsymbol{\Theta}^{(1)}$ and $\boldsymbol{\Theta}^{(2)}$ solve the entire JGL optimization problem if and only if, for all $i \in C_1$ and $j \in C_2$, there exist $\Gamma_{1,ij}, \Gamma_{2,ij}, \Upsilon_{ij} \in [-1, 1]$ such that

$$\begin{aligned} -n_1 S_{ij}^{(1)} - \lambda_1\Gamma_{1,ij} - \lambda_2\Upsilon_{ij} &= 0, \\ -n_2 S_{ij}^{(2)} - \lambda_1\Gamma_{2,ij} + \lambda_2\Upsilon_{ij} &= 0. \end{aligned} \tag{29}$$

Therefore, by lemma 2, the proof of the claim for the case $K = 2$ is complete.

The derivation of the sufficient condition for the case $K > 2$ is simple and we omit it here.

B.3. Proof of theorem 2

We note that theorem 2's condition (19) is equivalent to

$$|n_k S_{ij}^{(k)}| \leqslant \lambda_1 + \lambda_2 a_{ij,k} \qquad \text{for all } i \in C_1, j \in C_2, k = 1, \ldots, K \tag{30}$$

where $a_{ij,1}, \ldots, a_{ij,K}$ are scalars that satisfy $\sum_{k=1}^K a_{ij,k}^2 \leqslant 1$. We shall prove that condition (30) is necessary

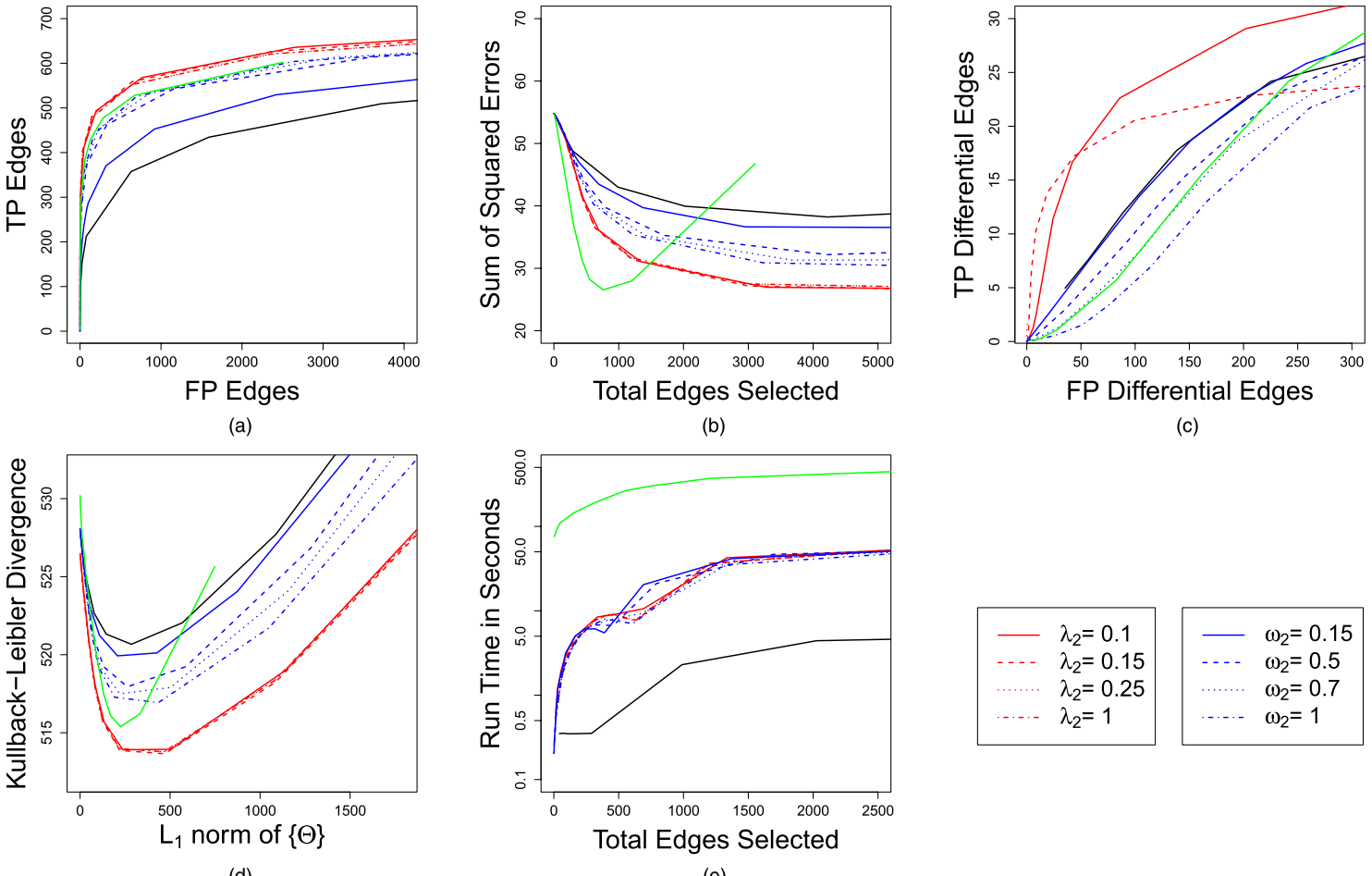

Fig. 4. Performance of the FGL (———), GGL (———), the method of Guo *et al.* (2011) (———) and the graphical lasso (———) on simulated data with 150 observations in each of two classes and 500 features corresponding to 10 equally sized unconnected subnetworks drawn from a power law distribution (the details are as given in Fig. 2)

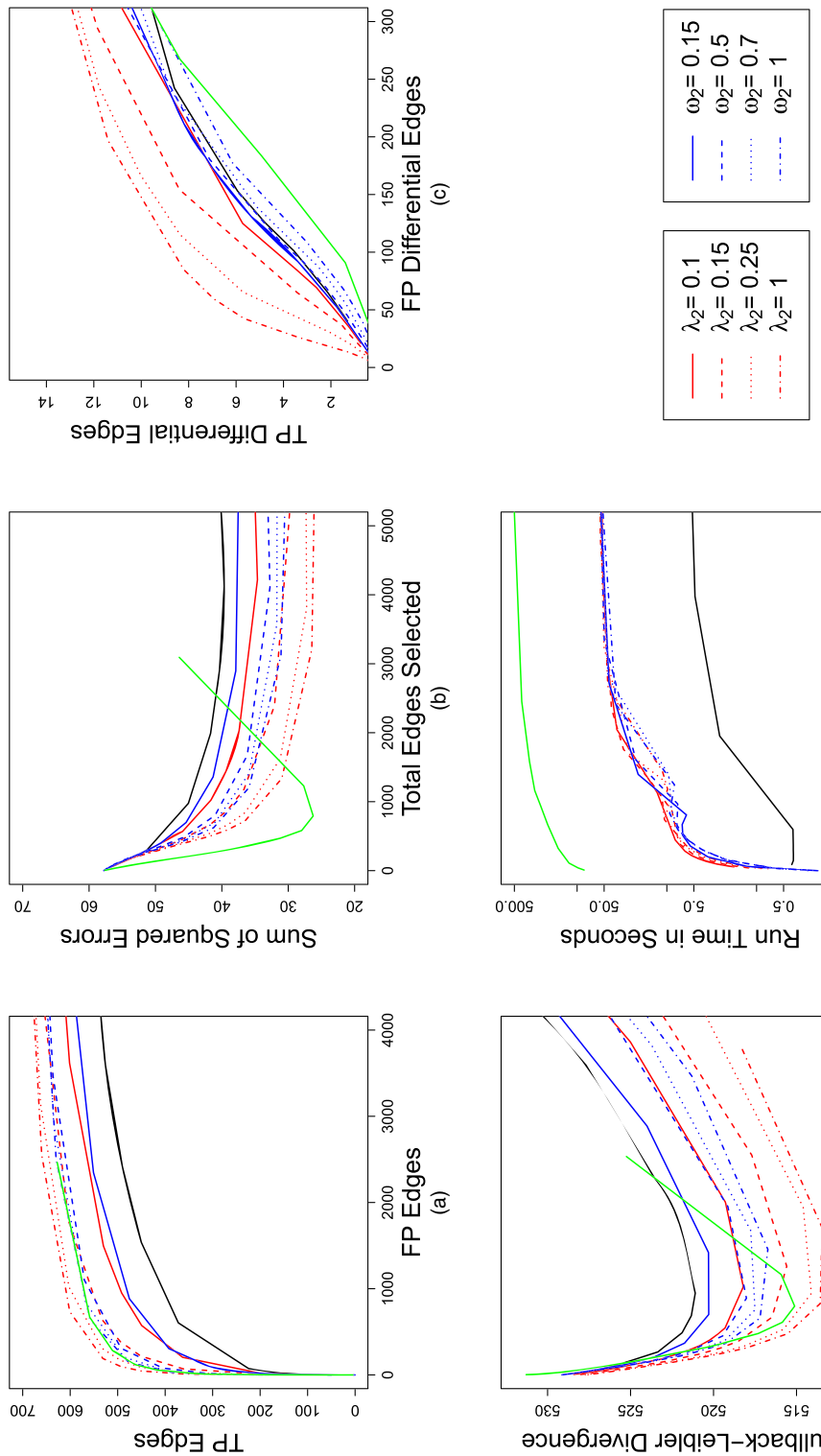

Fig. 5. Performance of the FGL (——), GGL (——), the method of Guo *et al.* (2011) (——) and the graphical lasso (——) on simulated data with 150 observations in each of two classes and 500 features corresponding to a single large power law network (the details are as given in Fig. 2)

and sufficient for the variables in C_1 to be completely disconnected from those in C_2 in each of the resulting network estimates.

By the Karush–Kuhn–Tucker conditions, a necessary and sufficient set of conditions for $\{\boldsymbol{\Theta}\}$ to be the solution to the JGL problem is that

$$0 = n_k(\boldsymbol{\Theta}^{(k)})^{-1} - n_k\mathbf{S}^{(k)} - \lambda_1\boldsymbol{\Gamma}_k - \lambda_2\boldsymbol{\Upsilon}_k \tag{31}$$

for $k = 1, \ldots, K$. In equation (31), $\Gamma_{k,ij}$ is the subgradient of $|\theta_{ij}^{(k)}|$ with respect to $\theta_{ij}^{(k)}$, and $(\Upsilon_{1,ij}, \ldots, \Upsilon_{K,ij})$ is the subgradient of $\sqrt{\Sigma_{k=1}^{K}(\theta_{ij}^{(k)})^2}$ with respect to $(\theta_{ij}^{(1)}, \ldots, \theta_{ij}^{(K)})$.

Let C_1 and C_2 be a partition of the p variables into two non-overlapping sets, with $C_1 \cap C_2 = \emptyset$ and $C_1 \cup C_2 = \{1, \ldots, p\}$. Consider the matrices of the form

$$\boldsymbol{\Theta}^{(k)} = \begin{pmatrix} \boldsymbol{\Theta}_1^{(k)} & 0 \\ 0 & \boldsymbol{\Theta}_2^{(k)} \end{pmatrix} \tag{32}$$

for $k = 1, \ldots, K$, where $\boldsymbol{\Theta}_1^{(1)}, \ldots, \boldsymbol{\Theta}_1^{(k)}$ solve the JGL problem on the features in C_1, and $\boldsymbol{\Theta}_2^{(1)}, \ldots, \boldsymbol{\Theta}_2^{(k)}$ solve the JGL problem on the features in C_2. By inspection of equation (31), $\boldsymbol{\Theta}^{(1)}, \ldots, \boldsymbol{\Theta}^{(k)}$ solve the entire JGL optimization problem if and only if, for all $i \in C_1$ and $j \in C_2$, there exist $\Gamma_{1,ij}, \ldots, \Gamma_{K,ij} \in [-1, 1]$ and $\Upsilon_{1,ij}, \ldots, \Upsilon_{K,ij}$ satisfying $\Sigma_{k=1}^{K}\Upsilon_{k,ij}^2 \leqslant 1$ such that

$$-n_k S_{ij}^{(k)} - \lambda_1\Gamma_{k,ij} - \lambda_2\Upsilon_{k,ij} = 0. \tag{33}$$

Therefore, by lemma 3, the proof is complete.

Appendix C: Additional simulations for two-class data sets

We first present results for a simulation study similar to that in Section 7.1, but with only two classes. Taking an approach similar to that described in Section 7.1, we defined two networks with $p = 500$ features belonging to 10 equally sized unconnected subnetworks, each with a power law degree distribution. Of the 10 subnetworks, eight have the same structure and edge values in both classes, and two are present in only one class. Class 1's network has 490 edges, 94 of which are not present in class 2. We generated covariance matrices as described in Section 7.1.1. Again, we simulated 100 data sets with $n = 150$ observations per class. The results that are shown in Fig. 4 are similar to the results in Section 7.1.2.

We also simulated data with an entirely different network structure. Instead of the block diagonal network structure that was used in the previous simulations, in this simulation we generated data drawn from a single large power law network. We defined class 1's network to be a single power law network with only one component and generated $\boldsymbol{\Sigma}_1$ as described in Section 7.1.1. We then identified a branch in this

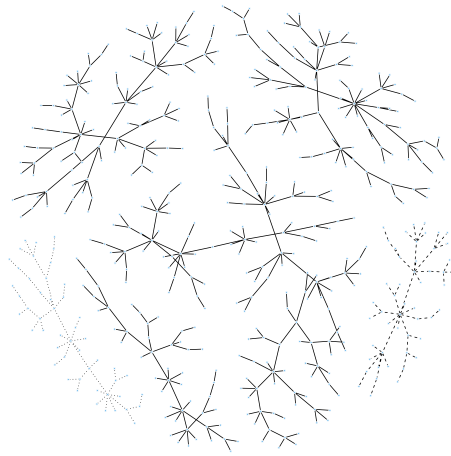

Fig. 6. Network used to generate simulated data sets for Fig. 2 in Section 7.1.2: full edges are common to all three classes; broken edges are present only in classes 1 and 2, and dotted edges are present only in class 1

network connected to the rest of the network through only one edge. We then let Σ_2^{-1} equal Σ_1^{-1}, except for the elements corresponding to the edges in the selected branch, which were set to be 0 instead. Finally, we defined Σ_2 by inverting Σ_2^{-1}, and generated the two classes' data by using Σ_1 and Σ_2. This yielded distributions based on two power law networks that were identical except for a missing branch in class 2. Class 1's network has 499 edges, 104 of which are not present in class 2. We simulated 100 data sets with $n = 150$ observations per class. Fig. 5 shows the results, averaged over the 100 data sets. Again, the FGL and GGL were superior to or competitive with the other methods.

Appendix D: Network structure used in simulations

The network structure for the simulations that were described in Section 7.1 is displayed in Fig. 6. Black edges are shared between all three classes' networks, green edges are present only in classes 1 and 2 and red edges are present only in class 1.

References

Adjei, A. (2008) K-ras as a target for lung cancer therapy. *J. Thor. Oncol.*, **3**, no. 6, S160–S163.

Ahmed, A. and Xing, E. (2009) Tesla: recovering time-varying networks of dependencies in social and biological studies. *Proc. Natn. Acad. Sci. USA*, **29**, 11878–11883.

Barrett, T., Suzek, T., Troup, D., Wilhite, S., Ngau, W., Ledoux, P., Rudnev, D., Lash, A., Fujibuchi, W. and Edgar, R. (2005) NCBI GEO: mining millions of expression profiles—database and tools. *Nucleic Acids Res.*, **33**, D562–D566.

Boyd, S., Parikh, N., Chu, E., Peleato, B. and Eckstein, J. (2010) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundns Trends Mach. Learn.*, **3**, 1–122.

Boyd, S. and Vandenberghe, L. (2004) *Convex Optimization*. New York: Cambridge University Press.

Chen, H. and Sharp, B. (2004) Content-rich biological network constructed by mining pubmed abstracts. *BMC Bioinform.*, **5**, article 147.

Friedman, J., Hastie, T., Hoefling, H. and Tibshirani, R. (2007a) Pathwise coordinate optimization. *Ann. Appl. Statist.*, **1**, 302–332.

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

Friedman, J., Hastie, T. and Tibshirani, R. (2010) A note on the group lasso and a sparse group lasso. *Technical Report*. Department of Statistics, Stanford University, Stanford.

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

Hastie, T., Tibshirani, R. and Friedman, J. (2009) *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. New York: Springer.

Hocking, L., Joulin, A. and Bach, F. (2011) Clusterpath: an algorithm for clustering using convex fusion penalties. In *Proc. 28th Int. Conf. Machine Learning*. New York: Omnipress.

Hoefling, H. (2010a) A path algorithm for the fused lasso signal approximator. *J. Computnl Graph. Statist.*, **19**, 984–1006.

Hoefling, H. (2010b) Personal communication.

Kolar, M., Song, L., Ahmed, A. and Xing, E. (2010) Estimating time-varying networks. *Ann. Appl. Statist.*, **4**, 94–123.

Kolar, M. and Xing, E. (2009) Sparsistent estimation of time-varying discrete markov random fields. Carnegie Mellon University, Pittsburgh. *Preprint arXiv:0907.2337*.

Lauritzen, S. (1996) *Graphical Models*. Oxford: Oxford Science Publications.

Li, S., Hsu, L., Peng, J. and Wang, P. (2013) Bootstrap inference for network construction. *Ann. Appl. Statist.*, to be published.

Mazumder, R. and Hastie, T. (2012) Exact covariance-thresholding into connected components for large-scale graphical lasso. *J. Mach. Learn. Res.*, **13**, 781–794.

Meinshausen, N. (2007) Relaxed lasso. *Computnl Statist. Data Anal.*, **52**, 374–393.

Meinshausen, N. and Bühlmann, P. (2006) High dimensional graphs and variable selection with the lasso. *Ann. Statist.*, **34**, 1436–1462.

Meinshausen, N. and Bühlmann, P. (2010) Stability selection (with discussion). *J. R. Statist. Soc.* B, **72**, 417–473.

Mohan, K., Chung, M., Han, S., Witten, D., Lee, S. and Fazel, M. (2012) Structured learning of Gaussian graphical models. *Adv. Neur. Inform. Process.*, **25**, 629–637.

Peng, J., Wang, P., Zhou, N. and Zhu, J. (2009) Partial correlation estimation by joint sparse regression model. *J. Am. Statist. Ass.*, **104**, 735–746.

Rothman, A., Levina, E. and Zhu, J. (2008) Sparse permutation invariant covariance estimation. *Electron. J. Statist.*, **2**, 494–515.
Simon, N. and Tibshirani, R. (2011) Discriminant analysis with adaptively pooled covariance. *Preprint arXiv:1111.1687*.
Song, L., Kolar, M. and Xing, E. (2009a) Keller: estimating time-evolving interactions between genes. *Bioinformatics*, **25**, i128–i136.
Song, L., Kolar, M. and Xing, E. (2009b) Time-varying dynamic Bayesian networks In *Proc. 23rd Neural Information Processing Systems Conf*. Red Hook: Curran Associates.
Spira, A., Beane, J., Shah, V., Steiling, K., Liu, G., Schembri, F., Gilman, S., Dumas, Y., Calner, P., Sebastiani, P., Sridhar, S., Beamis, J., Lamb, C., Anderson, T., Gerry, N., Keane, J., Lenburg, M. and Brody, J. (2007) Airway epithelial gene expression in the diagnostic evaluation of smokers with suspect lung cancer. *Nat. Med.*, **13**, 361–366.
Tarjan, R. (1972) Depth-first search and linear graph algorithms. *SIAM J. Comput.*, **1**, 146–160.
Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.* B, **58**, 267–288.
Tibshirani, R. (2012) Personal communication.
Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2005) Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc.* B, **67**, 91–108.
Witten, D., Friedman, J. and Simon, N. (2011) New insights and faster computations for the graphical lasso. *J. Computnl Graph. Statist.*, **20**, 892–900.
Witten, D. M. and Tibshirani, R. (2009) Covariance-regularized regression and classification for high-dimensional problems. *J. R. Statist. Soc.* B, **71**, 615–636.
Yuan, M. and Lin, Y. (2006) Model selection and estimation in regression with grouped variables. *J. R. Statist. Soc.* B, **68**, 49–67.
Yuan, M. and Lin, Y. (2007) Model selection and estimation in the Gaussian graphical model. *Biometrika*, **94**, 19–35.
Zhou, S., Lafferty, J. and Wasserman, L. (2008) Time varying undirected graphs. *21st A. Conf. Learning Theory*, Helsinki.

Supporting information

Additional 'supporting information' may be found in the on-line version of this article:

'Benchmarking the two MCMC strategies for sampling the Bayesian bridge posterior distiution' and
'An empirical study of mixing rates in parameter-expanded Gibbs samplers for sparse Bayesian regression models'