Supplementary materials for this article are available online.
Please click the JCGS link at http://pubs.amstat.org.

New Insights and Faster Computations for the Graphical Lasso

Daniela M. WITTEN, Jerome H. FRIEDMAN, and Noah SIMON

We consider the *graphical lasso* formulation for estimating a Gaussian graphical model in the high-dimensional setting. This approach entails estimating the inverse covariance matrix under a multivariate normal model by maximizing the ℓ_1-penalized log-likelihood. We present a very simple necessary and sufficient condition that can be used to identify the connected components in the graphical lasso solution. The condition can be employed to determine whether the estimated inverse covariance matrix will be block diagonal, and if so, then to identify the blocks. This in turn can lead to drastic speed improvements, since one can simply apply a standard graphical lasso algorithm to each block separately. Moreover, the necessary and sufficient condition provides insight into the graphical lasso solution: the set of connected nodes at any given tuning parameter value is a superset of the set of connected nodes at any larger tuning parameter value. This article has supplementary material online.

Key Words: Convex optimization; Gene expression; Inverse covariance estimation; ℓ_1 penalty; Networks; Sparsity.

1. INTRODUCTION

In recent years, there has been a great deal of interest in inverse covariance estimation in the high-dimensional setting, in which the number of features p greatly exceeds the number of observations n. Particular interest has focused upon estimating a *sparse inverse covariance matrix*—that is, obtaining an estimate of the inverse covariance matrix in which some elements are exactly equal to zero. This is of interest because under the simple setting where the rows of an $n \times p$ data matrix $\mathbf{X}$ are independent and distributed $N(\mathbf{0}, \mathbf{\Sigma})$, a zero in an off-diagonal element of $\mathbf{\Sigma}^{-1}$ corresponds to a pair of variables that are conditionally independent. Thus, under the multivariate Gaussian assumption, estimation of a sparse inverse covariance matrix can be thought of as a way to estimate a *graphical model* for the data. In the graphical model interpretation, each feature is represented by a node, and each

Daniela M. Witten is Assistant Professor, Department of Biostatistics, University of Washington, Box 357232, Seattle, WA 98195-7232 (E-mail: *dwitten@u.washington.edu*). Jerome H. Friedman is Professor, and Noah Simon is Ph.D. Student, Department of Statistics, Stanford University, Sequoia Hall, Stanford, CA 94305.

Journal of Computational and Graphical Statistics, Volume 20, Number 4, Pages 892–900
DOI: 10.1198/jcgs.2011.11051a

nonzero off-diagonal element in the inverse covariance matrix is represented by an edge between the corresponding pair of nodes.

A natural way to estimate $\mathbf{\Sigma}^{-1}$ is by maximizing the log-likelihood of the data. Under the Gaussian model, the log-likelihood takes the form

$$\log\det\mathbf{\Sigma}^{-1} - \operatorname{tr}(\mathbf{S}\mathbf{\Sigma}^{-1}), \tag{1.1}$$

where $\mathbf{S} = \mathbf{X}^T\mathbf{X}/n$ is an estimate of the covariance matrix of the data. Define $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$. Then maximizing (1.1) with respect to $\mathbf{\Theta}$ leads to the maximum likelihood estimate $\hat{\mathbf{\Theta}} = \mathbf{S}^{-1}$, which will in general contain no elements exactly equal to zero. Moreover, when $p > n$, $\mathbf{S}$ will be singular and so the maximum likelihood estimate cannot be computed.

Yuan and Lin (2007) proposed that rather than maximizing the log-likelihood (1.1), one should instead maximize the penalized log-likelihood

$$\log\det\mathbf{\Theta} - \operatorname{tr}(\mathbf{S}\mathbf{\Theta}) - \lambda\|\mathbf{\Theta}\|_1 \tag{1.2}$$

over nonnegative definite matrices $\mathbf{\Theta}$. Here, λ is a nonnegative tuning parameter. We will refer to the problem (1.2) as the *graphical lasso* (Friedman, Hastie, and Tibshirani 2007). Using (1.2) has two major advantages over (1.1): the solution is positive definite for all $\lambda > 0$ even if $\mathbf{S}$ is singular, and also when λ is sufficiently large, the estimate $\hat{\mathbf{\Theta}}$ will be *sparse* due to the *lasso*-type penalty on the elements of $\mathbf{\Theta}$ (Tibshirani 1996). A great number of algorithms have been proposed for solving the graphical lasso problem (1.2) (among others, Friedman, Hastie, and Tibshirani 2007; Yuan and Lin 2007; Banerjee, El Ghaoui, and D'Aspremont 2008; D'Aspremont, Banerjee, and El Ghaoui 2008; Rothman, Levina, and Zhu 2008; Yuan 2008; Lu 2009; Scheinberg, Ma, and Goldfarb 2010). We note that some authors have considered a slight variant of (1.2) in which the diagonal elements of $\mathbf{\Theta}$ are not penalized.

Here, we present a necessary and sufficient condition for a set of nodes to form a connected component in the graphical model—that is, for the solution to the graphical lasso problem to be block diagonal, subject to some rearrangement of the features. To our surprise, this condition appears to have been overlooked in the extensive literature on solving problem (1.2). However, it was independently discovered by Mazumder and Hastie (2011). The condition directly results in a very simple check that can be employed before solving (1.2), in order to achieve massive computational gains. We derive the necessary and sufficient condition in Section 2. Two algorithms based on this condition are given in Section 3, and we present timing results in Section 4. The discussion is in Section 5.

2. A NECESSARY AND SUFFICIENT CONDITION

By the Karush–Kuhn–Tucker conditions (KKT conditions; see, e.g., Boyd and Vandenberghe 2004), a necessary and sufficient condition for $\mathbf{\Theta}$ to maximize (1.2) is that it satisfies

$$\mathbf{\Theta}^{-1} - \mathbf{S} - \lambda\Gamma(\mathbf{\Theta}) = \mathbf{0}, \tag{2.1}$$

where $\Gamma(x)$ is the subgradient of $|x|$, applied componentwise to the elements of a matrix. That is, $\Gamma(\boldsymbol{\Theta})$ is a $p \times p$ matrix whose (i, j) element is $\Gamma(\Theta_{ij})$, the subgradient of $|\Theta_{ij}|$: $\Gamma(\Theta_{ij}) = 1$ if $\Theta_{ij} > 0$, $\Gamma(\Theta_{ij}) = -1$ if $\Theta_{ij} < 0$, and $\Gamma(\Theta_{ij})$ takes on a value between -1 and 1 if $\Theta_{ij} = 0$.

Theorem 1 states that if it is known a priori that the solution to the graphical lasso problem is block diagonal, then the solution can be obtained by solving a smaller graphical lasso problem on each individual block. The proof follows by inspection of (1.2).

Theorem 1. *If the solution to* (1.2) *takes the form* $\boldsymbol{\Theta} = \begin{pmatrix} \boldsymbol{\Theta}_1 & \\ & \boldsymbol{\Theta}_2 \end{pmatrix}$, *then* (1.2) *can be solved by separately maximizing*

$$\log\det\boldsymbol{\Theta}_1 - \operatorname{tr}(\mathbf{S}_1\boldsymbol{\Theta}_1) - \lambda\|\boldsymbol{\Theta}_1\|_1 \tag{2.2}$$

with respect to $\boldsymbol{\Theta}_1$, *and*

$$\log\det\boldsymbol{\Theta}_2 - \operatorname{tr}(\mathbf{S}_2\boldsymbol{\Theta}_2) - \lambda\|\boldsymbol{\Theta}_2\|_1 \tag{2.3}$$

with respect to $\boldsymbol{\Theta}_2$, *where* $\mathbf{S}_1$ *and* $\mathbf{S}_2$ *are the submatrices of* $\mathbf{S}$ *corresponding to* $\boldsymbol{\Theta}_1$ *and* $\boldsymbol{\Theta}_2$.

Clearly, one can extend Theorem 1 by induction to any number of blocks.

Now, let $C_1, C_2, \ldots, C_K$ denote a partition of the p features, i.e. $C_k \cap C_{k'} = \emptyset$ for all $k \neq k'$, and $C_1 \cup C_2 \cup \cdots \cup C_K = \{1, \ldots, p\}$. Without loss of generality, assume that the features are ordered such that if $i \in C_k, i' \in C_{k'}, k < k'$, then $i < i'$. Furthermore, we let $|C_k|$ denote the number of features in C_k. We now present our main theorem.

Theorem 2. *A necessary and sufficient condition for the solution to the graphical lasso problem to be block diagonal with blocks* $C_1, C_2, \ldots, C_K$ *is that* $|S_{ii'}| \leq \lambda$ *for all* $i \in C_k, i' \in C_{k'}, k \neq k'$.

Proof: First, we will show that if the solution to the graphical lasso problem is block diagonal with blocks $C_1, C_2, \ldots, C_K$, then $|S_{ii'}| \leq \lambda$ for all $i \in C_k, i' \in C_{k'}, k \neq k'$. To see this, note that if the solution is block diagonal and the ith and i'th features are in separate blocks, then $\Theta_{ii'} = (\boldsymbol{\Theta}^{-1})_{ii'} = 0$ since the inverse of a block diagonal matrix is also block diagonal with the same block structure. By inspection of (2.1), this implies that $|S_{ii'}| \leq \lambda$.

Now we must show that if $|S_{ii'}| \leq \lambda$ for all $i \in C_k, i' \in C_{k'}, k \neq k'$, then the solution to the graphical lasso problem is block diagonal. Consider the matrix

$$\tilde{\boldsymbol{\Theta}} = \begin{pmatrix} \boldsymbol{\Theta}_1 & & & \\ & \boldsymbol{\Theta}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{\Theta}_K \end{pmatrix}; \tag{2.4}$$

that is, $\tilde{\boldsymbol{\Theta}}$ is a block diagonal matrix with K blocks. We construct the kth block, $\boldsymbol{\Theta}_k$, to solve the graphical lasso problem (1.2) applied only to the $|C_k| \times |C_k|$ symmetric submatrix of $\mathbf{S}$ consisting of the features in C_k. We claim that if $|S_{ii'}| \leq \lambda$ for all $i \in C_k, i' \in C_{k'}$, $k \neq k'$, then $\tilde{\boldsymbol{\Theta}}$ satisfies the KKT conditions (2.1). This can be seen by inspection: the (i, i') equation of (2.1) is satisfied by the (i, i') element of $\tilde{\boldsymbol{\Theta}}$, for $i \in C_k, i' \in C_{k'}, k \neq k'$,

since $\tilde{\Theta}_{ii'} = (\tilde{\mathbf{\Theta}}^{-1})_{ii'} = 0$. Moreover, the kth block of $\tilde{\mathbf{\Theta}}$ satisfies the corresponding system of equations within (2.1), by construction. Since the KKT conditions are necessary and sufficient for a solution, it follows that $\tilde{\mathbf{\Theta}}$ does indeed solve the graphical lasso problem. □

The following corollary is a direct consequence of Theorem 2, applied with $C_1 = \{i\}$ and $C_2 = \{1, 2, \ldots, i-1, i+1, \ldots, p\}$.

Corollary 1. *A necessary and sufficient condition for the ith node to be fully unconnected from all other nodes in the solution to (1.2) is that $|S_{ii'}| \leq \lambda$ for all $i' \neq i$.*

This corollary implies that one can simply screen the off-diagonal elements in a given column of $\mathbf{S}$ in order to determine whether the corresponding node in the solution to the graphical lasso problem is unconnected from all other nodes. The sufficiency of the condition in Corollary 1 was reported in theorem 4 of Banerjee, El Ghaoui, and D'Aspremont (2008).

3. TWO NEW ALGORITHMS FOR THE GRAPHICAL LASSO

We can combine Corollary 1 with Theorem 1 in order to obtain Algorithm 1. Standard algorithms for solving the graphical lasso problem require $\mathcal{O}(p^3)$ operations (Friedman, Hastie, and Tibshirani 2007). Note that Step 1 of Algorithm 1 can be performed in at most $\mathcal{O}(p^2)$ operations. Then Step 3 can be performed in $\mathcal{O}(p-q)^3$ operations. In other words, by applying Algorithm 1, the total computational burden has been reduced from $O(p^3)$ to at most $O(p^2 + (p-q)^3)$. This is a great improvement if $p-q$ is substantially smaller than p, that is, if many of the nodes are fully unconnected from all other nodes in the graphical lasso solution.

Algorithm 1 A fast algorithm based on Corollary 1

1. Identify the fully unconnected nodes in the graphical lasso solution, that is, $\{i : |S_{ii'}| \leq \lambda, i' = 1, \ldots, i-1, i+1, \ldots, p\}$. Let q denote the number of fully unconnected nodes.
2. Without loss of generality, assume that the features are ordered such that the q fully unconnected features precede the $p-q$ other features.
3. The solution to the graphical lasso problem (1.2) takes the form

$$\mathbf{\Theta} = \begin{pmatrix} \frac{1}{S_{11}+\lambda} & & & \\ & \ddots & & \\ & & \frac{1}{S_{qq}+\lambda} & \\ & & & \mathbf{\Theta}_{q+1} \end{pmatrix}, \tag{2.5}$$

where $\mathbf{\Theta}_{q+1}$ solves the graphical lasso problem applied only to the square symmetric $(p-q) \times (p-q)$ submatrix of $\mathbf{S}$ consisting of the features that are *not* fully unconnected from all other features.

Algorithm 2 A fast algorithm based on Theorem 2

1. Let $\mathbf{A}$ denote a $p \times p$ matrix whose off-diagonal elements are of the form $A_{ii'} = 1_{|S_{ii'}|>\lambda}$ and whose diagonal elements equal one.
2. Identify the $K \geq 1$ connected components of the graph for which $\mathbf{A}$ is the adjacency matrix. For each $k = 1, \ldots, K$, let C_k denote the set of indices of the features in the kth connected component.
3. Without loss of generality, assume that the features are ordered such that if $i \in C_k$, $i' \in C_{k'}$, and $k < k'$, then $i < i'$.
4. The solution to the graphical lasso problem (1.2) takes the form

$$\boldsymbol{\Theta} = \begin{pmatrix} \boldsymbol{\Theta}_1 & & & \\ & \boldsymbol{\Theta}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{\Theta}_K \end{pmatrix}, \tag{3.1}$$

where $\boldsymbol{\Theta}_k$ solves the graphical lasso problem applied only to the square symmetric submatrix of $\mathbf{S}$ consisting of the features whose indices are in C_k. Note that if $C_k = \{i\}$—that is, the ith node is completely unconnected from all other nodes—then $\boldsymbol{\Theta}_k$ is a scalar equal to $1/(S_{ii} + \lambda)$.

An even faster algorithm can be obtained by combining Theorem 2 with Theorem 1, and is given in Algorithm 2.

In Step 1 of Algorithm 2, the adjacency matrix $\mathbf{A}$ can be computed in $\mathcal{O}(p^2)$ operations. Then in Step 2, the connected components of the graph can be computed in at most $\mathcal{O}(p^2)$ operations. In Step 4, the graphical lasso problem must be solved K times, once on each of the $|C_k| \times |C_k|$ submatrices of $\mathbf{S}$ corresponding to features whose indices are in C_k. This will require $\mathcal{O}(\sum_{k=1}^{K} |C_k|^3)$ operations. Therefore, applying Algorithm 2 leads to a reduction in computational complexity from $\mathcal{O}(p^3)$ to $\mathcal{O}(p^2 + \sum_{k=1}^{K} |C_k|^3)$. This is potentially a massive improvement if K is large, or if $|C_1|, \ldots, |C_K|$ are small relative to p.

In general, we expect Algorithm 2 to be faster than Algorithm 1, since the former exploits *all* block diagonal structure in the graphical lasso solution rather than only the fully unconnected nodes. However, in certain cases, the solution to the graphical lasso may be block diagonal with all but the largest block composed of individual fully unconnected nodes. In this case, Algorithm 1 may be faster than Algorithm 2, since Step 1 of Algorithm 1 is faster than Steps 1 and 2 of Algorithm 2, and the remaining steps are the same.

4. TIMING RESULTS

In a small simulation study, we explored the computational improvements that can result from Algorithms 1 and 2. We generated data with n independent observations distributed $N(0, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is a $p \times p$ matrix that takes one of three forms: (1) $\boldsymbol{\Sigma} = \mathbf{I}$, (2) $\boldsymbol{\Sigma}$ is a block diagonal matrix, with a single $(p/2) \times (p/2)$ block with all off-diagonal elements

equal to 0.5, and all diagonal elements of $\mathbf{\Sigma}$ equal 1, and (3) all off-diagonal elements of $\mathbf{\Sigma}$ equal 0.5, and all diagonal elements equal 1. We used $n = 20$ and considered three values of p: $p = 500, 1000, 2000$. We chose the tuning parameter in each simulation in order to achieve a desired "sparsity level," measured by the fraction of completely unconnected nodes in the graphical lasso solution. The three sparsity levels considered were 0.2, 0.5, and 0.9.

In order to solve (1.2), we used the `glasso` package in R (Friedman, Hastie, and Tibshirani 2011; R Development Core Team 2011), which is based upon the algorithm proposed by Friedman, Hastie, and Tibshirani (2007). Three versions of the `glasso` package were compared. Version 1.4 does not make use of the results in this article. Version 1.6 employs Algorithm 1, and version 1.7 employs Algorithm 2.

We also included in our comparisons the lasso-based neighborhood selection approach of Meinshausen and Bühlmann (2006) for estimating a graphical model. This approach entails estimating the nonzero edges of the ith variable by simply performing an ℓ_1-penalized regression of the ith column of $\mathbf{X}$ onto all of the other columns. It was shown by Yuan and Lin (2007), and discussed further by Friedman, Hastie, and Tibshirani (2007), that this can be seen as an approximation to the problem (1.2). In the timing comparisons reported by Friedman, Hastie, and Tibshirani (2007), the Meinshausen and Bühlmann (2006) approximation is substantially faster than the exact solution to the graphical lasso problem (1.2); however, those timing results were performed without the speed-ups to the graphical lasso algorithm described here. We used the implementation of the Meinshausen and Bühlmann (2006) algorithm contained in version 1.4 of the `glasso` package for our timing comparisons. Note that the same tuning parameter was used to perform the exact graphical lasso algorithm and the approximate Meinshausen and Bühlmann (2006) algorithm.

Timing results for the three simulations are shown in Tables 1–3. All timings were carried out on a AMD Opteron 848 2.20 GHz processor.

We first compare the speed of Algorithm 1 to the standard graphical lasso algorithm that does not use the results in this article (version 1.6 versus version 1.4). Using Algorithm 1 resulted in massive reductions in computational time. In general, the speed-up factor due to Algorithm 1 will increase as the tuning parameter is increased, and will also increase as the number of nodes in the model (p) is increased for a given fraction of unconnected nodes.

Algorithm 2 led to substantial speed increases over Algorithm 1 in Simulation 1 when 20% or 50% of nodes were unconnected, and slightly smaller speed increases over Algorithm 1 in Simulation 2 when 20% or 50% of nodes were unconnected. It did not lead to an improvement over Algorithm 1 in Simulation 3, since in that case the true $\mathbf{\Sigma}$ is dense and so the connected nodes in the graphical lasso solution mostly belong to a single connected component rather than multiple connected components. (Recall that Algorithm 2 provides an improvement over Algorithm 1 only if the connected nodes in the graphical lasso solution belong to distinct connected components.) Regardless of the simulation set-up, when 90% of nodes were unconnected then Algorithm 1 tended to be a bit faster than Algorithm 2. This is because computing the graphical lasso solution on the 10% of nodes that are not unconnected is so fast that the time required to identify all connected components in the graphical lasso solution, as in Algorithm 2, is not worthwhile.

Table 1. Timing comparisons for computing the graphical lasso solution using three versions of the `glasso` package: without the results in this article (version 1.4), using Algorithm 1 (version 1.6), and using Algorithm 2 (version 1.7). The tuning parameter λ was chosen so that 20% of nodes in the graphical lasso solution are unconnected. Means (and standard errors) were computed over 20 replicates, and are reported in seconds. Timings for the Meinshausen and Bühlmann (2006) (MB) approximation are also reported.

		Simulation 1	Simulation 2	Simulation 3
$p = 500$	Version 1.4	5.601 (0.24)	11.254 (0.727)	10.229 (0.284)
	Version 1.6	3.145 (0.133)	4.947 (0.241)	5.999 (0.243)
	Version 1.7	1.198 (0.139)	4.728 (0.39)	6.714 (0.238)
	MB approximation	1.498 (0.069)	1.234 (0.035)	1.217 (0.036)
$p = 1000$	Version 1.4	45.039 (1.905)	91.831 (5.286)	81.783 (2.843)
	Version 1.6	25.311 (1.33)	44.638 (2.82)	58.465 (2.879)
	Version 1.7	11.975 (0.901)	39.111 (2.337)	60.656 (2.602)
	MB approximation	11.294 (0.36)	8.687 (0.085)	9.29 (0.282)
$p = 2000$	Version 1.4	338.926 (7.859)	787.872 (44.312)	676.375 (21.802)
	Version 1.6	186.753 (7.366)	409.218 (22.067)	346.913 (9.922)
	Version 1.7	85.123 (4.178)	343.652 (23.57)	404.491 (4.934)
	MB approximation	70.177 (2.479)	71.561 (1.264)	73.773 (1.922)

Strikingly, when the graphical lasso is performed with a large tuning parameter value, then the algorithms proposed in this article lead to such massive speed improvements that computing the *exact* graphical lasso solution is much faster even than computing the Meinshausen and Bühlmann (2006) approximate solution (Tables 2 and 3).

5. DISCUSSION

We have presented a necessary and sufficient condition for the solution to the graphical lasso problem to be block diagonal. This condition leads to a simple check that can be

Table 2. As in Table 1, but with λ chosen so that 50% of nodes are unconnected.

		Simulation 1	Simulation 2	Simulation 3
$p = 500$	Version 1.4	5.865 (0.129)	7.779 (0.389)	7.934 (0.32)
	Version 1.6	0.553 (0.02)	0.762 (0.046)	0.64 (0.024)
	Version 1.7	0.162 (0.005)	0.944 (0.107)	0.874 (0.025)
	MB approximation	1.133 (0.029)	1.211 (0.041)	1.155 (0.031)
$p = 1000$	Version 1.4	46.614 (1.615)	73.371 (4.123)	77.58 (3.674)
	Version 1.6	7.282 (0.17)	14.068 (0.708)	9.779 (0.42)
	Version 1.7	0.615 (0.015)	14.829 (1.611)	11.149 (0.539)
	MB approximation	8.679 (0.223)	9.458 (0.318)	8.982 (0.251)
$p = 2000$	Version 1.4	330.943 (7.859)	583.787 (37.084)	468.915 (17.78)
	Version 1.6	58.494 (1.179)	99.675 (7.591)	76.579 (3.042)
	Version 1.7	2.302 (0.032)	92.461 (12.207)	81.758 (2.363)
	MB approximation	70.876 (1.691)	73.523 (1.799)	70.707 (1.431)

Table 3. As in Table 1, but with λ chosen so that 90% of nodes are unconnected.

		Simulation 1	Simulation 2	Simulation 3
$p = 500$	Version 1.4	5.503 (0.171)	6.594 (0.497)	4.262 (0.21)
	Version 1.6	0.102 (0.002)	0.11 (0.002)	0.099 (0.002)
	Version 1.7	0.104 (0.002)	0.112 (0.002)	0.131 (0.002)
	MB approximation	1.498 (0.038)	1.701 (0.075)	1.839 (0.087)
$p = 1000$	Version 1.4	43.346 (2.205)	48.706 (2.96)	39.047 (1.245)
	Version 1.6	0.398 (0.004)	0.433 (0.005)	0.403 (0.003)
	Version 1.7	0.46 (0.006)	0.499 (0.01)	0.535 (0.007)
	MB approximation	10.564 (0.272)	11.221 (0.573)	12.548 (0.675)
$p = 2000$	Version 1.4	215.078 (5.11)	351.06 (9.346)	301.123 (7.548)
	Version 1.6	1.808 (0.012)	1.95 (0.02)	1.792 (0.013)
	Version 1.7	1.888 (0.024)	2.251 (0.032)	2.342 (0.015)
	MB approximation	73.3 (1.784)	75.687 (2.695)	70.516 (2.165)

applied to the empirical covariance matrix before a standard graphical lasso algorithm is employed, in order to drastically reduce computations.

Using this quick check, one can estimate graphical models on datasets that were previously prohibitively large, such as large genomic datasets. For instance, it now takes only a few minutes to estimate a graphical model from a gene expression dataset with $p = 30{,}000$ genes, provided that one is interested in a model in which most of the genes are not connected (as one would likely want for the sake of interpretability). In this case, the main computational burden simply involves scanning through the elements of the $p \times p$ empirical covariance matrix in order to identify off-diagonal elements that are less than λ in absolute value.

The necessary and sufficient condition presented here also leads to new insights into the graphical lasso solution. Given two tuning parameters λ_1 and λ_2, $\lambda_1 < \lambda_2$, the set of unconnected nodes (i.e., individual nodes that are connected to no other node) with tuning parameter λ_1 is a subset of the set of unconnected nodes with tuning parameter λ_2. (However, no such claim can be made about the set of nonzero edges—i.e., an edge that is nonzero in the graphical model with tuning parameter λ_1 may equal zero in the graphical model with tuning parameter λ_2, and vice-versa.) Furthermore, the nodes in a connected component of the graphical lasso solution with tuning parameter λ_1 will remain unconnected from all nodes not in that connected component as the tuning parameter increases.

SUPPLEMENTARY MATERIALS

R package for graphical lasso without results in this paper: R package `glasso1.4` is version 1.4 of the `glasso` package available on CRAN, which does not make use of the results in this article. (glasso1.4_1.0.tar.gz)

R package for graphical lasso with Algorithm 1: R package `glasso1.6` is version 1.6 of the `glasso` package available on CRAN, which makes use of Algorithm 1. (glasso1.6_1.0.tar.gz)

R package for graphical lasso with Algorithm 2: R-package `glasso1.7` is version 1.7 of the `glasso` package available on CRAN, which makes use of Algorithm 2. (glasso1.7_1.0.tar.gz)

R script to perform timing comparisons: R script to perform timing comparisons reported in Tables 1–3. (TimingComparisons.R)

ACKNOWLEDGMENTS

The authors thank Robert Tibshirani for useful suggestions and for integrating the algorithms described here in the `glasso` R package. We also thank Pei Wang for helpful comments, and Patrick Danaher for spotting an error in an earlier version of this article.

[Received April 2011. Revised May 2011.]

REFERENCES

Banerjee, O., El Ghaoui, L. E., and D'Aspremont, A. (2008), "Model Selection Through Sparse Maximum Likelihood Estimation for Multivariate Gaussian or Binary Data," *Journal of Machine Learning Research*, 9, 485–516. [893,895]

Boyd, S., and Vandenberghe, L. (2004), *Convex Optimization*, Cambridge, U.K.: Cambridge University Press. [893]

D'Aspremont, A., Banerjee, O., and El Ghaoui, L. (2008), "First-Order Methods for Sparse Covariance Selection," *SIAM Journal on Matrix Analysis and Applications*, 30, 56–66. [893]

Friedman, J., Hastie, T., and Tibshirani, R. (2007), "Sparse Inverse Covariance Estimation With the Graphical Lasso," *Biostatistics*, 9, 432–441. [893,895,897]

——— (2011), "glasso: Graphical Lasso—Estimation of Gaussian Graphical Models," R package version 1.4, available at *cran.r-project.org*. [897]

Lu, Z. (2009), "Smooth Optimization Approach for Sparse Covariance Selection," *SIAM Journal on Optimization*, 19, 1807–1827. [893]

Mazumder, R., and Hastie, T. (2011), "Exact Covariance Thresholding Into Connected Components for Large-Scale Graphical Lasso," available at *http://arxiv.org/abs/1108.3829*. [893]

Meinshausen, N., and Bühlmann, P. (2006), "High-Dimensional Graphs and Variable Selection With the Lasso," *The Annals of Statistics*, 34, 1436–1462. [897,898]

R Development Core Team (2011), *R: A Language and Environment for Statistical Computing*, Vienna, Austria: R Foundation for Statistical Computing. Available at *http://www.R-project.org*. [897]

Rothman, A., Levina, E., and Zhu, J. (2008), "Sparse Permutation Invariant Covariance Estimation," *Electronic Journal of Statistics*, 2, 494–515. [893]

Scheinberg, K., Ma, S., and Goldfarb, D. (2010), "Sparse Inverse Covariance Selection via Alternating Linearization Methods," in *Advances in Neural Information Processing Systems*, Vancouver, Canada: MIT Press. Available at *http://books.nips.cc/papers/files/nips23/NIPS2010_0109.pdf*. [893]

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society, Ser. B*, 58, 267–288. [893]

Yuan, M. (2008), "Efficient Computation of ℓ_1 Regularized Estimates in Gaussian Graphical Models," *Journal of Computational and Graphical Statistics*, 17, 809–826. [893]

Yuan, M., and Lin, Y. (2007), "Model Selection and Estimation in the Gaussian Graphical Model," *Biometrika*, 94, 19–35. [893,897]