Journal of Machine Learning Research 13 (2012) 781-794 Submitted 8/11; Revised 1/12; Published 3/12

Exact Covariance Thresholding into Connected Components for Large-Scale Graphical Lasso

Rahul Mazumder RAHULM@STANFORD.EDU
Trevor Hastie* HASTIE@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305

Editor: Francis Bach

Abstract

We consider the sparse inverse covariance regularization problem or *graphical lasso* with regularization parameter λ. Suppose the sample *covariance graph* formed by thresholding the entries of the sample covariance matrix at λ is decomposed into connected components. We show that the *vertex-partition* induced by the connected components of the thresholded sample covariance graph (at λ) is *exactly* equal to that induced by the connected components of the estimated concentration graph, obtained by solving the graphical lasso problem for the *same* λ. This characterizes a very interesting property of a path of graphical lasso solutions. Furthermore, this simple rule, when used as a wrapper around existing algorithms for the graphical lasso, leads to enormous performance gains. For a range of values of λ, our proposal splits a large graphical lasso problem into smaller tractable problems, making it possible to solve an otherwise infeasible large-scale problem. We illustrate the graceful scalability of our proposal via synthetic and real-life microarray examples.

Keywords: sparse inverse covariance selection, sparsity, graphical lasso, Gaussian graphical models, graph connected components, concentration graph, large scale covariance estimation

1. Introduction

Consider a data matrix $\mathbf{X}_{n\times p}$ comprising of n sample realizations from a p dimensional Gaussian distribution with zero mean and positive definite covariance matrix $\Sigma_{p\times p}$ (unknown), that is, $\mathbf{x}_i \overset{\text{i.i.d}}{\sim} \text{MVN}(\mathbf{0},\Sigma),\ i=1,\ldots,n$. The task is to estimate the unknown Σ based on the n samples. ℓ_1 regularized Sparse Inverse Covariance Selection also known as *graphical lasso* (Friedman et al., 2007; Banerjee et al., 2008; Yuan and Lin, 2007) estimates the covariance matrix Σ, under the assumption that the inverse covariance matrix, that is, Σ^{-1} is sparse. This is achieved by minimizing the regularized negative log-likelihood function:

$$\underset{\Theta\succeq\mathbf{0}}{\text{minimize}} \quad -\log\det(\Theta)+\text{tr}(\mathbf{S}\Theta)+\lambda\sum_{i,j}|\Theta_{ij}|, \tag{1}$$

where $\mathbf{S}$ is the sample covariance matrix. Problem (1) is a convex optimization problem in the variable Θ (Boyd and Vandenberghe, 2004). Let $\widehat{\Theta}^{(\lambda)}$ denote the solution to (1). We note that (1) can also be used in a more non-parametric fashion for any positive semi-definite input matrix $\mathbf{S}$, not necessarily a sample covariance matrix of a MVN sample as described above.

∗. Also in the Department of Health, Research and Policy

A related criterion to (1) is one where the diagonals are not penalized—by substituting $\mathbf{S} \leftarrow \mathbf{S} + \lambda I_{p\times p}$ in the "unpenalized" problem we get (1). In this paper we concentrate on problem (1).

Developing efficient large-scale algorithms for (1) is an active area of research across the fields of Convex Optimization, Machine Learning and Statistics. Many algorithms have been proposed for this task (Friedman et al., 2007; Banerjee et al., 2008; Lu, 2009, 2010; Scheinberg et al., 2010; Yuan, 2009, for example). However, it appears that certain special properties of the solution to (1) have been largely ignored. This paper is about one such (surprising) property—namely establishing an equivalence between the *vertex-partition* induced by the connected components of the non-zero pattern of $\widehat{\Theta}^{(\lambda)}$ and the thresholded sample covariance matrix $\mathbf{S}$. This paper is *not* about a specific algorithm for the problem (1)—it focuses on the aforementioned observation that leads to a novel thresholding/screening procedure based on $\mathbf{S}$. This provides interesting insight into the path of solutions $\{\widehat{\Theta}^{(\lambda)}\}_{\lambda\geq 0}$ obtained by solving (1), over a path of λ values. The behavior of the connected-components obtained from the non-zero patterns of $\{\widehat{\Theta}^{(\lambda)}\}_{\lambda\geq 0}$ can be completely understood by simple screening rules on $\mathbf{S}$. This can be done without *even attempting* to solve (1)—arguably a very challenging convex optimization problem. Furthermore, this thresholding rule can be used as a *wrapper* to enormously boost the performance of existing algorithms, as seen in our experiments. This strategy becomes extremely effective in solving large problems over a range of values of λ—sufficiently restricted to ensure sparsity and the separation into connected components. Of course, for sufficiently small values of λ there will be no separation into components, and hence no computational savings.

At this point we introduce some notation and terminology, which we will use throughout the paper.

1.1 Notations and Preliminaries

For a matrix $\mathbf{Z}$, its $(i,j)^{\text{th}}$ entry is denoted by $\mathbf{Z}_{ij}$.

We also introduce some graph theory notations and definitions (Bollobas, 1998) sufficient for this exposition. A finite undirected graph $\mathcal{G}$ on p vertices is given by the ordered tuple $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ the collection of (undirected) edges. The edge-set is equivalently represented via a (symmetric) 0-1 matrix[1] (also known as the *adjacency* matrix) with p rows/columns. We use the convention that a node is not connected to itself, so the diagonals of the adjacency matrix are all zeros. Let $|\mathcal{V}|$ and $|\mathcal{E}|$ denote the number of nodes and edges respectively.

We say two nodes $u, v \in \mathcal{V}$ are *connected* if there is a *path* between them. A maximal connected *subgraph*[2] is a *connected component* of the graph $\mathcal{G}$. *Connectedness* is an equivalence relation that decomposes a graph $\mathcal{G}$ into its connected components $\{(\mathcal{V}_\ell, \mathcal{E}_\ell)\}_{1\leq\ell\leq K}$—with $\mathcal{G} = \cup_{\ell=1}^{K}(\mathcal{V}_\ell, \mathcal{E}_\ell)$, where K denotes the number of connected components. This decomposition partitions the vertices $\mathcal{V}$ of $\mathcal{G}$ into $\{\mathcal{V}_\ell\}_{1\leq\ell\leq K}$. Note that the labeling of the components is unique upto permutations on $\{1,\ldots,K\}$. Throughout this paper we will often refer to this partition as the *vertex-partition* induced by the components of the graph $\mathcal{G}$. If the size of a component is one, that is, $|\mathcal{V}_\ell| = 1$, we say that the node is *isolated*. Suppose a graph $\widehat{\mathcal{G}}$ defined on the set of vertices $\mathcal{V}$ admits the following decomposition into connected components: $\widehat{\mathcal{G}} = \cup_{\ell=1}^{\hat{K}}(\widehat{\mathcal{V}}_\ell, \widehat{\mathcal{E}}_\ell)$. We say the vertex-

1. 0 denotes absence of an edge and 1 denotes its presence.
2. $\mathcal{G}' = (\mathcal{V}', \mathcal{E}')$ is a *subgraph* of $\mathcal{G}$ if $\mathcal{V}' \subset \mathcal{V}$ and $\mathcal{E}' \subset \mathcal{E}$.

partitions induced by the connected components of $\mathcal{G}$ and $\widehat{\mathcal{G}}$ are *equal* if $\widehat{K} = K$ and there is a permutation π on $\{1,\ldots,K\}$ such that $\widehat{\mathcal{V}}_{\pi(\ell)} = \mathcal{V}_\ell$ for all $\ell \in \{1,\ldots,K\}$.

The paper is organized as follows. Section 2 describes the covariance graph thresholding idea along with theoretical justification and related work, followed by complexity analysis of the algorithmic framework in Section 3. Numerical experiments appear in Section 4, concluding remarks in Section 5 and the proofs are gathered in the Appendix A.

2. Methodology: *Exact* Thresholding of the Covariance Graph

The sparsity pattern of the solution $\widehat{\Theta}^{(\lambda)}$ to (1) gives rise to the symmetric edge matrix/skeleton $\in \{0,1\}^{p\times p}$ defined by:

$$\mathcal{E}_{ij}^{(\lambda)} = \begin{cases} 1 & \text{if } \widehat{\Theta}_{ij}^{(\lambda)} \neq 0,\, i \neq j; \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The above defines a symmetric graph $\mathcal{G}^{(\lambda)} = (\mathcal{V}, \mathcal{E}^{(\lambda)})$, namely the *estimated concentration graph* (Cox and Wermuth, 1996; Lauritzen, 1996) defined on the nodes $\mathcal{V} = \{1,\ldots,p\}$ with edges $\mathcal{E}^{(\lambda)}$.

Suppose the graph $\mathcal{G}^{(\lambda)}$ admits a decomposition into $\kappa(\lambda)$ connected components:

$$\mathcal{G}^{(\lambda)} = \cup_{\ell=1}^{\kappa(\lambda)} \mathcal{G}_\ell^{(\lambda)}, \tag{3}$$

where $\mathcal{G}_\ell^{(\lambda)} = (\widehat{\mathcal{V}}_\ell^{(\lambda)}, \mathcal{E}_\ell^{(\lambda)})$ are the components of the graph $\mathcal{G}^{(\lambda)}$. Note that $\kappa(\lambda) \in \{1,\ldots,p\}$, with $\kappa(\lambda) = p$ (large λ) implying that all nodes are isolated and for small enough values of λ, there is only one component, that is, $\kappa(\lambda) = 1$.

We now describe the simple screening/thresholding rule. Given λ, we perform a thresholding on the entries of the sample covariance matrix $\mathbf{S}$ and obtain a graph edge skeleton $\mathrm{E}^{(\lambda)} \in \{0,1\}^{p\times p}$ defined by:

$$\mathrm{E}_{ij}^{(\lambda)} = \begin{cases} 1 & \text{if } |\mathbf{S}_{ij}| > \lambda,\, i \neq j; \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

The symmetric matrix $\mathrm{E}^{(\lambda)}$ defines a symmetric graph on the nodes $\mathcal{V} = \{1,\ldots,p\}$ given by $\mathrm{G}^{(\lambda)} = (\mathcal{V}, \mathrm{E}^{(\lambda)})$. We refer to this as the *thresholded sample covariance graph*. Similar to the decomposition in (3), the graph $\mathrm{G}^{(\lambda)}$ also admits a decomposition into connected components:

$$\mathrm{G}^{(\lambda)} = \cup_{\ell=1}^{k(\lambda)} \mathrm{G}_\ell^{(\lambda)}, \tag{5}$$

where $\mathrm{G}_\ell^{(\lambda)} = (\mathcal{V}_\ell^{(\lambda)}, \mathrm{E}_\ell^{(\lambda)})$ are the components of the graph $\mathrm{G}^{(\lambda)}$.

Note that the components of $\mathcal{G}^{(\lambda)}$ require knowledge of $\widehat{\Theta}^{(\lambda)}$—the solution to (1). Construction of $\mathrm{G}^{(\lambda)}$ and its components require operating on $\mathbf{S}$—an operation that can be performed completely independent of the optimization problem (1), which is arguably more expensive (See Section 3). The surprising message we describe in this paper is that the *vertex-partition* of the connected components of (5) is *exactly* equal to that of (3).

This observation has the following consequences:

1. We obtain a very interesting property of the path of solutions $\{\widehat{\Theta}^{(\lambda)}\}_{\lambda\geq 0}$—the behavior of the connected components of the estimated concentration graph can be completely understood by simple screening rules on $\mathbf{S}$.

2. The cost of computing the connected components of the thresholded sample covariance graph (5) is orders of magnitude smaller than the cost of fitting graphical models (1). Furthermore, the computations pertaining to the covariance graph can be done off-line and are amenable to parallel computation (See Section 3).

3. The optimization problem (1) completely separates into $k(\lambda)$ separate optimization sub-problems of the form (1). The sub-problems have size equal to the number of nodes in each component $p_i := |\mathcal{V}_i|, i = 1,\ldots,k(\lambda)$. Hence for certain values of λ, solving problem (1) becomes feasible although it may be impossible to operate on the $p \times p$ dimensional (global) variable Θ on a single machine.

4. Suppose that for λ_0, there are $k(\lambda_0)$ components and the graphical model computations are distributed.[3] Since the vertex-partitions induced via (3) and (5) are nested with increasing λ (see Theorem 2), it suffices to operate independently on these separate machines to obtain the path of solutions $\{\widehat{\Theta}^{(\lambda)}\}_\lambda$ for all $\lambda \geq \lambda_0$.

5. Consider a distributed computing architecture, where every machine allows operating on a graphical lasso problem (1) of maximal size $p_{\max}$. Then with relatively small effort we can find the smallest value of $\lambda = \lambda_{p_{\max}}$, such that there are no connected components of size larger than $p_{\max}$. Problem (1) thus 'splits up' independently into manageable problems across the different machines. When this structure is not exploited the global problem (1) remains intractable.

The following theorem establishes the main technical contribution of this paper—the equivalence of the vertex-partitions induced by the connected components of the thresholded sample covariance graph and the estimated concentration graph.

Theorem 1 *For any $\lambda > 0$, the components of the estimated concentration graph $\mathcal{G}^{(\lambda)}$, as defined in (2) and (3) induce* exactly *the same vertex-partition as that of the thresholded sample covariance graph $\mathrm{G}^{(\lambda)}$, defined in (4) and (5). That is $\kappa(\lambda) = k(\lambda)$ and there exists a permutation π on $\{1,\ldots,k(\lambda)\}$ such that:*

$$\widehat{\mathcal{V}}_i^{(\lambda)} = \mathcal{V}_{\pi(i)}^{(\lambda)}, \ \forall i = 1,\ldots,k(\lambda). \tag{6}$$

Proof *The proof of the theorem appears in Appendix A.1.* ■

Since the decomposition of a symmetric graph into its connected components depends upon the ordering/ labeling of the components, the permutation π appears in Theorem 1.

Remark 1 *Note that the edge-structures* within *each block need not be preserved. Under a matching reordering of the labels of the components of $\mathcal{G}^{(\lambda)}$ and $\mathrm{G}^{(\lambda)}$:*
for every fixed ℓ such that $\widehat{\mathcal{V}}_\ell^{(\lambda)} = \mathcal{V}_\ell^{(\lambda)}$ the edge-sets $\mathcal{E}_\ell^{(\lambda)}$ and $\mathrm{E}_\ell^{(\lambda)}$ are not *necessarily equal.*

3. Distributing these operations depend upon the number of processors available, their capacities, communication lag, the number of components and the maximal size of the blocks across all machines. These of-course depend upon the computing environment. In the context of the present problem, it is often desirable to club smaller components into a single machine.

Theorem 1 leads to a special property of the path-of-solutions to (1), that is, the vertex-partition induced by the connected components of $\mathcal{G}^{(\lambda)}$ are nested with increasing λ. This is the content of the following theorem.

Theorem 2 *Consider two values of the regularization parameter such that $\lambda > \lambda' > 0$, with corresponding concentration graphs $\mathcal{G}^{(\lambda)}$ and $\mathcal{G}^{(\lambda')}$ as in (2) and connected components (3). Then the vertex-partition induced by the components of $\mathcal{G}^{(\lambda)}$ are* nested *within the partition induced by the components of $\mathcal{G}^{(\lambda')}$. Formally, $\kappa(\lambda) \geq \kappa(\lambda')$ and the vertex-partition $\{\widehat{\mathcal{V}}_\ell^{(\lambda)}\}_{1\leq\ell\leq\kappa(\lambda)}$ forms a finer resolution of $\{\widehat{\mathcal{V}}_\ell^{(\lambda')}\}_{1\leq\ell\leq\kappa(\lambda')}$.*
Proof *The proof of this theorem appears in the Appendix A.2.* ■

Remark 2 *It is worth noting that Theorem 2 addresses the nesting of the edges* across *connected components and not within a component. In general, the edge-set $\mathcal{E}^{(\lambda)}$ of the estimated concentration graph need not be nested as a function of λ:*
for $\lambda > \lambda'$, in general, $\mathcal{E}^{(\lambda)} \not\subset \mathcal{E}^{(\lambda')}$.

See Friedman et al. (2007, Figure 3), for numerical examples demonstrating the non-monotonicity of the edge-set across λ, as described in Remark 2.

2.1 Node-Thresholding

A simple consequence of Theorem 1 is that of *node-thresholding*. If $\lambda \geq \max_{j\neq i}|\mathbf{S}_{ij}|$, then the i^{th} node will be isolated from the other nodes, the off-diagonal entries of the i^{th} row/column are all zero, that is, $\max_{j\neq i}|\widehat{\Theta}_{ij}^{(\lambda)}| = 0$. Furthermore, the i^{th} diagonal entries of the estimated covariance and precision matrices are given by $(\mathbf{S}_{ii}+\lambda)$ and $\frac{1}{\mathbf{S}_{ii}+\lambda}$, respectively. Hence, as soon as $\lambda \geq \max_{i=1,\ldots,p}\{\max_{j\neq i}|\mathbf{S}_{ij}|\}$, the estimated covariance and precision matrices obtained from (1) are both diagonal.

2.2 Related Work

Witten et al. (2011) independently discovered block screening as described in this paper. At the time of our writing, an earlier version of their paper was available (Witten and Friedman, 2011); it proposed a scheme to detect isolated nodes for problem (1) via a simple screening of the entries of $\mathbf{S}$, but no block screening. Earlier, Banerjee et al. (2008, Theorem 4) made the same observation about isolated nodes. The revised manuscript (Witten et al., 2011) that includes block screening became available shortly after our paper was submitted for publication.

Zhou et al. (2011) use a thresholding strategy followed by re-fitting for estimating Gaussian graphical models. Their approach is based on the node-wise lasso-regression procedure of Meinshausen and Bühlmann (2006). A hard thresholding is performed on the ℓ_1-penalized regression coefficient estimates at every node to obtain the graph structure. A restricted MLE for the concentration matrix is obtained for the graph. The proposal in our paper differs since we are interested in solving the GLASSO problem (1).

3. Computational Complexity

The overall complexity of our proposal depends upon (a) the graph partition stage and (b) solving (sub)problems of the form (1). In addition to these, there is an unavoidable complexity associated with handling and/or forming $\mathbf{S}$.

The cost of computing the connected components of the thresholded covariance graph is fairly negligible when compared to solving a similar sized graphical lasso problem (1)—see also our simulation studies in Section 4. In case we observe samples $\mathbf{x}_i \in \Re^p, i = 1, \ldots, n$ the cost for creating the sample covariance matrix $\mathbf{S}$ is $O(n \cdot p^2)$. Thresholding the sample covariance matrix costs $O(p^2)$. Obtaining the connected components of the thresholded covariance graph costs $O(|\mathrm{E}^{(\lambda)}| + p)$ (Tarjan, 1972). Since we are interested in a region where the thresholded covariance graph is sparse enough to be broken into smaller connected components—$|\mathrm{E}^{(\lambda)}| \ll p^2$. Note that all computations pertaining to the construction of the connected components and the task of computing $\mathbf{S}$ can be computed off-line. Furthermore the computations are parallelizable. Gazit (1991, for example) describes parallel algorithms for computing connected components of a graph—they have a time complexity $O(\log(p))$ and require $O((|\mathrm{E}^{(\lambda)}| + p)/\log(p))$ processors with space $O(p + |\mathrm{E}^{(\lambda)}|)$.

There are a wide variety of algorithms for the task of solving (1). While an exhaustive review of the computational complexities of the different algorithms is beyond the scope of this paper, we provide a brief summary for a few algorithms below.

Banerjee et al. (2008) proposed a smooth accelerated gradient based method (Nesterov, 2005) with complexity $O(\frac{p^{4.5}}{\varepsilon})$ to obtain an ε accurate solution—the per iteration cost being $O(p^3)$. They also proposed a block coordinate method which has a complexity of $O(p^4)$.

The complexity of the GLASSO algorithm (Friedman et al., 2007) which uses a row-by-row block coordinate method is roughly $O(p^3)$ for reasonably sparse-problems with p nodes. For denser problems the cost can be as large as $O(p^4)$.

The algorithm SMACS proposed in Lu (2010) has a per iteration complexity of $O(p^3)$ and an overall complexity of $O(\frac{p^4}{\sqrt{\varepsilon}})$ to obtain an $\varepsilon > 0$ accurate solution.

It appears that most existing algorithms for (1), have a complexity of at least $O(p^3)$ to $O(p^4)$ or possibly larger, depending upon the algorithm used and the desired accuracy of the solution—making computations for (1) almost impractical for values of p much larger than 2000.

It is quite clear that the role played by covariance thresholding is indeed crucial in this context. Assume that we use a solver of complexity $O(p^J)$ with $J \in \{3,4\}$, along with our screening procedure. Suppose for a given λ, the thresholded sample covariance graph has $k(\lambda)$ components—the total cost of solving these smaller problems is then $\sum_{i=1}^{k(\lambda)} O(|\mathcal{V}_i^{(\lambda)}|^J) \ll O(p^J)$, with $J \in \{3,4\}$. This difference in practice can be enormous—see Section 4 for numerical examples. This is what makes large scale graphical lasso problems solvable!

4. Numerical Examples

In this section we show via numerical experiments that the screening property helps in obtaining many fold speed-ups when compared to an algorithm that does not exploit it. Section 4.1 considers synthetic examples and Section 4.2 discusses real-life microarray data-examples.

4.1 Synthetic Examples

Experiments are performed with two publicly available algorithm implementations for the problem (1):

GLASSO**:** The algorithm of Friedman et al. (2007). We used the MATLAB wrapper available at `http://www-stat.stanford.edu/~tibs/glasso/index.html` to the Fortran code. The specific criterion for convergence (lack of progress of the diagonal entries) was set to 10^{-5} and the maximal number of iterations was set to 1000.

SMACS**:** denotes the algorithm of Lu (2010). We used the MATLAB implementation `smooth_covsel` available at `http://people.math.sfu.ca/~zhaosong/Codes/SMOOTH_COVSEL/`. The criterion for convergence (based on duality gap) was set to 10^{-5} and the maximal number of iterations was set to 1000.

We will like to note that the convergence criteria of the two algorithms GLASSO and SMACS are not the same. For obtaining the connected components of a symmetric adjacency matrix we used the MATLAB function `graphconncomp`. All of our computations are done in MATLAB 7.11.0 on a 3.3 GhZ Intel Xeon processor.

The simulation examples are created as follows. We generated a block diagonal matrix given by $\tilde{\mathbf{S}} = \text{blkdiag}(\tilde{\mathbf{S}}_1, \ldots, \tilde{\mathbf{S}}_K)$, where each block $\tilde{\mathbf{S}}_\ell = \mathbf{1}_{p_\ell \times p_\ell}$—a matrix of all ones and $\sum_\ell p_\ell = p$. In the examples we took all p_ℓs to be equal to p_1 (say). Noise of the form $\sigma \cdot UU'$ (U is a $p \times p$ matrix with i.i.d. standard Gaussian entries) is added to $\tilde{\mathbf{S}}$ such that 1.25 times the largest (in absolute value) off block-diagonal (as in the block structure of $\tilde{\mathbf{S}}$) entry of $\sigma \cdot UU'$ equals the smallest absolute non-zero entry in $\tilde{\mathbf{S}}$, that is, one. The sample covariance matrix is $\mathbf{S} = \tilde{\mathbf{S}} + \sigma \cdot UU'$.

We consider a number of examples for varying K and p_1 values, as shown in Table 1. Sizes were chosen such that it is at-least 'conceivable' to solve (1) on the full dimensional problem, without screening. In all the examples shown in Table 1, we set $\lambda_I := (\lambda_{\max} + \lambda_{\min})/2$, where for all values of λ in the interval $[\lambda_{\min}, \lambda_{\max}]$ the thresh-holded version of the sample covariance matrix has exactly K connected components. We also took a larger value of λ, that is, $\lambda_{II} := \lambda_{\max}$, which gave sparser estimates of the precision matrix but the number of connected components were the same.

The computations across different connected blocks could be distributed into as many machines. This would lead to almost a K fold improvement in timings, however in Table 1 we report the timings by operating serially across the blocks. The serial 'loop' across the different blocks are implemented in MATLAB.

Table 1 shows the rather remarkable improvements obtained by using our proposed covariance thresholding strategy as compared to operating on the whole matrix. Timing comparisons between GLASSO and SMACS are not fair, since GLASSO is written in Fortran and SMACS in MATLAB. However, we note that our experiments are meant to demonstrate how the thresholding helps in improving the overall computational time over the baseline method of not exploiting screening. Clearly our proposed strategy makes solving larger problems (1), not only feasible but with quite attractive computational time. The time taken by the graph-partitioning step in splitting the thresholded covariance graph into its connected components is negligible as compared to the timings for the optimization problem.

K	p_1 / p	λ	**Algorithm**	**Algorithm Timings** (sec) with screen	without screen	**Ratio** Speedup factor	**Time** (sec) graph partition
2	200 / 400	λ_I	GLASSO	11.1	25.97	2.33	0.04
			SMACS	12.31	137.45	11.16	
		λ_{II}	GLASSO	1.687	4.783	2.83	0.066
			SMACS	10.01	42.08	4.20	
2	500 /1000	λ_I	GLASSO	305.24	735.39	2.40	0.247
			SMACS	175	2138*	12.21	
		λ_{II}	GLASSO	29.8	121.8	4.08	0.35
			SMACS	272.6	1247.1	4.57	
5	300 /1500	λ_I	GLASSO	210.86	1439	6.82	0.18
			SMACS	63.22	6062*	95.88	
		λ_{II}	GLASSO	10.47	293.63	28.04	0.123
			SMACS	219.72	6061.6	27.58	
5	500 /2500	λ_I	GLASSO	1386.9	-	-	0.71
			SMACS	493	-	-	
		λ_{II}	GLASSO	17.79	963.92	54.18	0.018
			SMACS	354.81	-	-	
8	300 /2400	λ_I	GLASSO	692.25	-	-	0.713
			SMACS	185.75	-	-	
		λ_{II}	GLASSO	9.07	842.7	92.91	0.023
			SMACS	153.55	-	-	

Table 1: Table showing (a) the times in seconds with screening, (b) without screening, that is, on the whole matrix and (c) the ratio (b)/(a)—'Speedup factor' for algorithms GLASSO and SMACS. Algorithms with screening are operated serially—the times reflect the total time summed across all blocks. The column 'graph partition' lists the time for computing the connected components of the thresholded sample covariance graph. Since $\lambda_{II} > \lambda_I$, the former gives sparser models. '*' denotes the algorithm did not converge within 1000 iterations. '-' refers to cases where the respective algorithms failed to converge within 2 hours.

4.2 Micro-array Data Examples

The graphical lasso is often used in learning connectivity networks in gene-microarray data (Friedman et al., 2007, see for example). Since in most real examples the number of genes p is around tens of thousands, obtaining an inverse covariance matrix by solving (1) is computationally impractical. The covariance thresholding method we propose easily applies to these problems—and as we

see gracefully delivers solutions over a large range of the parameter λ. We study three different micro-array examples and observe that as one varies λ from large to small values, the thresholded covariance graph splits into a number of non-trivial connected components of varying sizes. We continue till a small/moderate value of λ when the maximal size of a connected component gets larger than a predefined machine-capacity or the 'computational budget' for a single graphical lasso problem. Note that in relevant micro-array applications, since $p \gg n$ (n, the number of samples is at most a few hundred) heavy regularization is required to control the variance of the covariance estimates—so it does seem reasonable to restrict to solutions of (1) for large values of λ.

Following are the data-sets we used for our experiments:

(A) This data-set appears in Alon et al. (1999) and has been analyzed by Rothman et al. (2008, for example). In this experiment, tissue samples were analyzed using an Affymetrix Oligonucleotide array. The data were processed, filtered and reduced to a subset of $p = 2000$ gene expression values. The number of Colon Adenocarcinoma tissue samples is $n = 62$.

(B) This is an early example of an expression array, obtained from the Patrick Brown Laboratory at Stanford University. There are $n = 385$ patient samples of tissue from various regions of the body (some from tumors, some not), with gene-expression measurements for $p = 4718$ genes.

(C) The third example is the by now famous NKI data set that produced the 70-gene prognostic signature for breast cancer (Van-De-Vijver et al., 2002). Here there are $n = 295$ samples and $p = 24481$ genes.

Among the above, both (B) and (C) have few missing values—which we imputed by the respective global means of the observed expression values. For each of the three data-sets, we took $\mathbf{S}$ to be the corresponding sample correlation matrix. The *exact thresholding* methodolgy could have also been applied to the sample covariance matrix. Since it is a common practice to standardize the "genes", we operate on the sample correlation matrix.

Figure 1 shows how the component sizes of the thresholded covariance graph change across λ. We describe the strategy we used to arrive at the figure. Note that the connected components change *only* at the absolute values of the entries of $\mathbf{S}$. From the sorted absolute values of the off-diagonal entries of $\mathbf{S}$, we obtained the smallest value of λ, say $\lambda'_{\min}$, for which the size of the maximal connected component was 1500. For a grid of values of λ till $\lambda'_{\min}$, we computed the connected components of the thresholded sample-covariance matrix and obtained the size-distribution of the various connected components. Figure 1 shows how these components change over a range of values of λ for the three examples (A), (B) and (C). The number of connected components of a particular size is denoted by a color-scheme, described by the color-bar in the figures. With increasing λ: the larger connected components gradually disappear as they decompose into smaller components; the sizes of the connected components decrease and the frequency of the smaller components increase. Since these are all correlation matrices, for $\lambda \geq 1$ all the nodes in the graph become isolated. The range of λ values for which the maximal size of the components is smaller than 1500 differ across the three examples. For (C) there is a greater variety in the sizes of the components as compared to (A) and (B). Note that by Theorem 1, the pattern of the components appearing in Figure 1 are exactly the same as the components appearing in the solution of (1) for that λ.

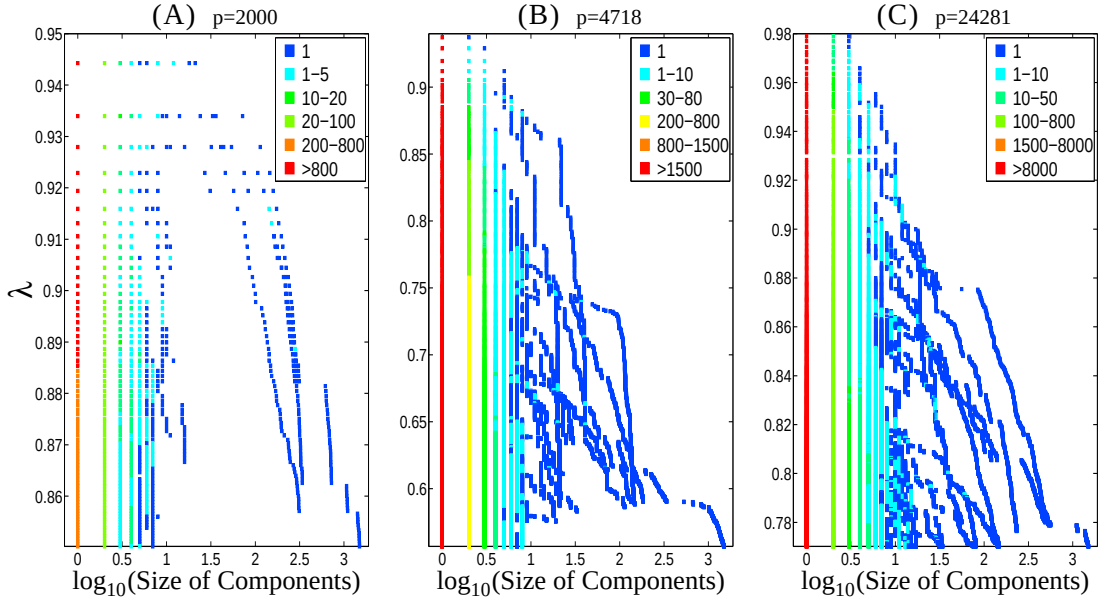

Figure 1: Figure showing the size distribution (in the log-scale) of connected components arising from the thresholded sample covariance graph for examples (A)-(C). For every value of λ (vertical axis), the horizontal slice denotes the sizes of the different components appearing in the thresholded covariance graph. The colors represent the number of components in the graph having that specific size. For every figure, the range of λ values is chosen such that the maximal size of the connected components do not exceed 1500.

For examples (B) and (C) we found that the full problem sizes are beyond the scope of GLASSO and SMACS—the screening rule is apparently the *only* way to obtain solutions for a reasonable range of λ-values as shown in Figure 1.

5. Conclusions

In this paper we present a novel property characterizing the family of solutions to the graphical lasso problem (1), as a function of the regularization parameter λ. The property is fairly surprising—the vertex partition induced by the connected components of the non-zero pattern of the estimated concentration matrix (at λ) and the thresholded sample covariance matrix $\mathbf{S}$ (at λ) are *exactly equal*. This property seems to have been unobserved in the literature. Our observation not only provides interesting insights into the properties of the graphical lasso solution-path but also opens the door to solving large-scale graphical lasso problems, which are otherwise intractable. This simple rule when used as a wrapper around existing algorithms leads to enormous performance boosts—on occasions by a factor of thousands!

Acknowledgments

This work is supported by NSF-DMS grant 1007719.

Appendix A. Proofs

Here we provide proofs of Theorems 1 and 2.

A.1 Proof of Theorem 1

Proof Suppose $\widehat{\Theta}$ (we suppress the superscript λ for notational convenience) solves problem (1), then standard KKT conditions of optimality (Boyd and Vandenberghe, 2004) give:

$$|\mathbf{S}_{ij} - \widehat{\mathbf{W}}_{ij}| \leq \lambda \quad \forall\, \widehat{\Theta}_{ij} = 0; \qquad \text{and} \tag{7}$$

$$\widehat{\mathbf{W}}_{ij} = \mathbf{S}_{ij} + \lambda \quad \forall\, \widehat{\Theta}_{ij} > 0; \quad \widehat{\mathbf{W}}_{ij} = \mathbf{S}_{ij} - \lambda \,\forall\, \widehat{\Theta}_{ij} < 0; \tag{8}$$

where $\widehat{\mathbf{W}} = (\widehat{\Theta})^{-1}$. The diagonal entries satisfy $\widehat{\mathbf{W}}_{ii} = \mathbf{S}_{ii} + \lambda$, for $i = 1, \ldots, p$.

Using (4) and (5), there exists an ordering of the vertices $\{1, \ldots, p\}$ of the graph such that $\mathrm{E}^{(\lambda)}$ is block-diagonal. For notational convenience, we will assume that the matrix is already in that order. Under this ordering of the vertices, the edge-matrix of the thresholded covariance graph is of the form:

$$\mathrm{E}^{(\lambda)} = \begin{pmatrix} \mathrm{E}_1^{(\lambda)} & 0 & \cdots & 0 \\ 0 & \mathrm{E}_2^{(\lambda)} & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \mathrm{E}_{k(\lambda)}^{(\lambda)} \end{pmatrix} \tag{9}$$

where the different components represent blocks of indices given by: $\mathcal{V}_\ell^{(\lambda)}, \ell = 1, \ldots, k(\lambda)$.

We will construct a matrix $\widehat{\mathbf{W}}$ having the same structure as (9) which is a solution to (1). Note that if $\widehat{\mathbf{W}}$ is block diagonal then so is its inverse. Let $\widehat{\mathbf{W}}$ and its inverse $\widehat{\Theta}$ be given by:

$$\widehat{\mathbf{W}} = \begin{pmatrix} \widehat{\mathbf{W}}_1 & 0 & \cdots & 0 \\ 0 & \widehat{\mathbf{W}}_2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \widehat{\mathbf{W}}_{k(\lambda)} \end{pmatrix}, \quad \widehat{\Theta} = \begin{pmatrix} \widehat{\Theta}_1 & 0 & \cdots & 0 \\ 0 & \widehat{\Theta}_2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \widehat{\Theta}_{k(\lambda)} \end{pmatrix}$$

Define the block diagonal matrices $\widehat{\mathbf{W}}_\ell$ or equivalently $\widehat{\Theta}_\ell$ via the following sub-problems

$$\widehat{\Theta}_\ell = \underset{\Theta_\ell}{\arg\min} \;\; \{-\log\det(\Theta_\ell) + \mathrm{tr}(\mathbf{S}_\ell \Theta_\ell) + \lambda \sum_{ij} |\,(\Theta_\ell)_{ij}\,|\} \tag{10}$$

for $\ell = 1, \ldots, k(\lambda)$, where $\mathbf{S}_\ell$ is a sub-block of $\mathbf{S}$, with row/column indices from $\mathcal{V}_\ell^{(\lambda)} \times \mathcal{V}_\ell^{(\lambda)}$. The same notation is used for Θ_ℓ. Denote the inverses of the block-precision matrices by $\{\widehat{\Theta}_\ell\}^{-1} = \widehat{\mathbf{W}}_\ell$. We will show that the above $\widehat{\Theta}$ satisfies the KKT conditions—(7) and (8).

Note that by construction of the thresholded sample covariance graph,
if $i \in \mathcal{V}_\ell^{(\lambda)}$ and $j \in \mathcal{V}_{\ell'}^{(\lambda)}$ with $\ell \neq \ell'$, then $|\mathbf{S}_{ij}| \leq \lambda$.

Hence, for $i \in \mathcal{V}_\ell^{(\lambda)}$ and $j \in \mathcal{V}_{\ell'}^{(\lambda)}$ with $\ell \neq \ell'$; the choice $\widehat{\Theta}_{ij} = \widehat{\mathbf{W}}_{ij} = 0$ satisfies the KKT conditions (7)

$$|\mathbf{S}_{ij} - \widehat{\mathbf{W}}_{ij}| \leq \lambda$$

for all the off-diagonal entries in the block-matrix (9).

By construction (10) it is easy to see that for every ℓ, the matrix $\widehat{\Theta}_\ell$ satisfies the KKT conditions (7) and (8) corresponding to the ℓ^{th} block of the $p \times p$ dimensional problem. Hence $\widehat{\Theta}$ solves problem (1).

The above argument shows that the connected components obtained from the estimated precision graph $\mathcal{G}^{(\lambda)}$ leads to a partition of the vertices $\{\widehat{\mathcal{V}}_\ell^{(\lambda)}\}_{1\le\ell\le\kappa(\lambda)}$ such that for every $\ell \in \{1,\ldots,k(\lambda)\}$, there is a $\ell' \in \{1,\ldots,\kappa(\lambda)\}$ such that $\widehat{\mathcal{V}}_{\ell'}^{(\lambda)} \subset \mathcal{V}_\ell^{(\lambda)}$. In particular $k(\lambda) \le \kappa(\lambda)$.

Conversely, if $\widehat{\Theta}$ admits the decomposition as in the statement of the theorem, then it follows from (7) that:
for $i \in \widehat{\mathcal{V}}_\ell^{(\lambda)}$ and $j \in \widehat{\mathcal{V}}_{\ell'}^{(\lambda)}$ with $\ell \neq \ell'$; $|\mathbf{S}_{ij} - \widehat{\mathbf{W}}_{ij}| \le \lambda$. Since $\widehat{\mathbf{W}}_{ij} = 0$, we have $|\mathbf{S}_{ij}| \le \lambda$. This proves that the connected components of $\mathrm{G}^{(\lambda)}$ leads to a partition of the vertices, which is finer than the vertex-partition induced by the components of $\mathcal{G}^{(\lambda)}$. In particular this implies that $k(\lambda) \ge \kappa(\lambda)$.

Combining the above two we conclude $k(\lambda) = \kappa(\lambda)$ and also the equality (6). The permutation π in the theorem appears since the labeling of the connected components is not unique. ■

A.2 Proof of Theorem 2

Proof This proof is a direct consequence of Theorem 1, which establishes that the vertex-partitions induced by the the connected components of the estimated precision graph and the thresholded sample covariance graph are equal.

Observe that, by construction, the connected components of the thresholded sample covariance graph, that is, $\mathrm{G}^{(\lambda)}$ are nested within the connected components of $\mathrm{G}^{(\lambda')}$. In particular, the vertex-partition induced by the components of the thresholded sample covariance graph at λ, is contained inside the vertex-partition induced by the components of the thresholded sample covariance graph at λ'. Now, using Theorem 1 we conclude that the vertex-partition induced by the components of the estimated precision graph at λ, given by $\{\widehat{\mathcal{V}}_\ell^{(\lambda)}\}_{1\le\ell\le\kappa(\lambda)}$ is contained inside the vertex-partition induced by the components of the estimated precision graph at λ', given by $\{\widehat{\mathcal{V}}_\ell^{(\lambda')}\}_{1\le\ell\le\kappa(\lambda')}$. The proof is thus complete. ■

References

U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750, June 1999. ISSN 0027-8424. doi: 10.1073/pnas.96.12.6745. URL `http://dx.doi.org/10.1073/pnas.96.12.6745`.

O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

B. Bollobas. *Modern Graph Theory*. Springer, New York, 1998.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

D.R Cox and N. Wermuth. *Multivariate Dependencies*. Chapman and Hall, London, 1996.

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

H. Gazit. An optimal randomized parallel algorithm for finding connected components in a graph. *SIAM Journal on Computing*, 20(6):1046–1067, 1991.

S. Lauritzen. *Graphical Models*. Oxford University Press, 1996.

Z. Lu. Smooth optimization approach for sparse covariance selection. *SIAM Journal on Optimization*, 19:1807–1827, February 2009. ISSN 1052-6234. doi: 10.1137/070695915. URL http://portal.acm.org/citation.cfm?id=1654243.1654257.

Z. Lu. Adaptive first-order methods for general sparse inverse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 31:2000–2016, May 2010. ISSN 0895-4798. doi: http://dx.doi.org/10.1137/080742531. URL http://dx.doi.org/10.1137/080742531.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *Annals of Statistics*, 34:1436–1462, 2006.

Y. Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming, Series A*, 103:127–152, 2005.

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

K. Scheinberg, S. Ma, and D. Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In *Neural Information Processing Systems*, pages 2101–2109, 2010.

R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*, 1(2): 146160, 1972.

M. J. Van-De-Vijver, Y. D. He, L. J. van't Veer, H. Dai, A. A. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, M. J. Marton, M. Parrish, D. Atsma, A. Witteveen, A. Glas, L. Delahaye, T. van der Velde, H. Bartelink, S. Rodenhuis, E. T. Rutgers, S. H. Friend, and R. Bernards. A gene-expression signature as a predictor of survival in breast cancer. *The New England Journal of Medicine*, 347:1999–2009, Dec 2002.

D. Witten and J. Friedman. A fast screening rule for the graphical lasso. *accepted for publication in Journal of Computational and Graphical Statistics*, 2011. Report dated 5-12-2011.

D. Witten, J. Friedman, and N. Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4):892–900, 2011. (Reference draft dated 9/8/2011).

M. Yuan and Y. Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

X. Yuan. Alternating direction methods for sparse covariance selection. *Methods*, (August):1–12, 2009. URL http://www.optimization-online.org/DB-FILE/2009/09/2390.pdf.

S. Zhou, P. Rütimann, M. Xu, and P. Bühlmann. High-dimensional covariance estimation based on gaussian graphical models. *Journal of Machine Learning Research*, 999888:2975–3026, November 2011. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2078183.2078201.