RC25543 (WAT1509-0064) September 16, 2015
Mathematics

IBM Research Report

Scalable Computation of Regularized Precision Matrices via Stochastic Optimization

Yves F. Atchadé
University of Michigan
Ann Arbor, MI USA

Rahul Mazumder
MIT
Cambridge, MA USA

Jie Chen
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598 USA

Research Division
Almaden – Austin – Beijing – Brazil – Cambridge – Dublin – Haifa – India – Kenya – Melbourne – T.J. Watson – Tokyo – Zurich

Scalable Computation of Regularized Precision Matrices via Stochastic Optimization

Yves F. Atchadé yvesa@umich.edu
Department of Statistics,
University of Michigan,
Ann Arbor, MI, USA

Rahul Mazumder rahulmaz@mit.edu
MIT Sloan School of Management and Operations Research Center,
Massachusetts Institute of Technology,
Cambridge, MA, USA

Jie Chen chenjie@us.ibm.com
IBM Thomas J. Watson Research Center,
Yorktown Heights, NY, USA.

Editor:

Abstract

We consider the problem of computing a positive definite $p \times p$ inverse covariance matrix aka precision matrix $\theta = (\theta_{ij})$ which optimizes a regularized Gaussian maximum likelihood problem, with the elastic-net regularizer $\sum_{i,j=1}^{p} \lambda(\alpha|\theta_{ij}| + \frac{1}{2}(1-\alpha)\theta_{ij}^2)$, with regularization parameters $\alpha \in [0,1]$ and $\lambda > 0$. The associated convex semidefinite optimization problem is notoriously difficult to scale to large problems and has demanded significant attention over the past several years. We propose a new algorithmic framework based on stochastic proximal optimization (on the primal problem) that can be used to obtain near optimal solutions with substantial computational savings over deterministic algorithms. A key challenge of our work stems from the fact that the optimization problem being investigated does not satisfy the usual assumptions required by stochastic gradient methods. Our proposal has (a) computational guarantees and (b) scales well to large problems, even if the solution is not too sparse; thereby, enhancing the scope of regularized maximum likelihood problems to many large-scale problems of contemporary interest. An important aspect of our proposal is to bypass the *deterministic* computation of a matrix inverse by drawing random samples from a suitable multivariate Gaussian distribution.

Keywords: Graphical Lasso, Ridge Regularization, ℓ_1-regularization, Gaussian Maximum Likelihood, Precision Matrices, Stochastic Optimization, Proximal Gradient Descent

1. Introduction

We consider the problem of estimating an inverse covariance matrix aka precision matrix (Lauritzen, 1996) θ, from a data matrix $X_{n \times p}$ comprised of n samples from a p dimensional multivariate Gaussian distribution with mean zero and covariance matrix $\Sigma = \theta^{-1}$, i.e., $\mathbf{x}_i \overset{\text{i.i.d.}}{\sim} \mathbf{N}(0, \Sigma)$ for $i = 1, \ldots, n$. If $n < p$ it is a well known fact that the Maximum Likelihood Estimate (MLE) does not exist, and even if it does exist ($n \geq p$) the MLE can be poorly behaved and regularization is often called for. Various forms of regularization are used to improve the statistical behavior of covariance matrix estimates (Pourahmadi, 2013; Bühlmann and Van De Geer, 2011; Hastie et al., 2009) and is a topic of significant interest in the statistics and machine learning communities. This paper deals with the problem of computing such regularized matrices, in the settings where p is much larger than n or both p and n are large. To motivate the reader, we briefly review two popular forms of precision matrix regularization schemes under a likelihood framework: sparse precision matrix estimation via ℓ_1-norm regularization, and its dense counterpart, using an ℓ_2-norm regularization (ridge penalty); both on the entries of the matrix θ.

Sparse precision matrix estimation — the Graphical Lasso

One of the most popular regularization approaches and the main motivation behind this paper is the Graphical Lasso (Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2007b) procedure aka Glasso. Here, we estimate θ under the assumption that it is sparse, with a few number of non-zeros. Under the multivariate Gaussian modeling set up, $\theta_{ij} = 0$ (for $i \neq j$) is equivalent to the conditional independence of x_i and x_j given the remaining variables, where, $\mathbf{x} = (x_1, \ldots, x_p) \sim \mathbf{N}(0, \Sigma)$. Glasso minimizes the negative log-likelihood subject to a penalty on the ℓ_1 norm of the entries of the precision matrix θ. This leads to the following convex optimization problem (Boyd and Vandenberghe, 2004):

$$\underset{\theta \in \mathcal{M}_+}{\text{minimize}} \quad \underbrace{-\log\det\theta + \mathsf{Tr}(\theta S)}_{:=f(\theta)} + \lambda \sum_{i,j} |\theta_{ij}|, \tag{1}$$

where, $S = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i'$ is the sample covariance matrix, $\mathcal{M}_+$ denotes the set of positive definite matrices and $\lambda > 0$ is a tuning parameter that controls the degree of regularization[1]. In passing, we note that the Glasso criterion, though motivated as a regularized negative log-likelihood problem, can be used more generally for any positive semidefinite (PSD) matrix S.

In modern statistical applications we frequently encounter examples where Problem (1) needs to be solved for p of the order of several thousands. Thus there is an urgent need to develop fast and scalable algorithms for Problem (1). In this vein, the past several years have witnessed a flurry of interesting work in developing fast and efficient solvers for Problem (1). We present a very brief overview of the main approaches used for the Glasso problem, with further additional details presented in the Appendix, Section A. A representative list of popular algorithmic approaches include (a) block (where, each row/column is a block) coordinate methods (Banerjee et al., 2008; Friedman et al., 2007a; Mazumder and Hastie, 2012b); (b) proximal gradient descent type methods (Banerjee et al., 2008; Lu, 2009; Rolfs et al., 2012); (c) methods based on Alternating Direction Method of Multipliers (Scheinberg et al., 2010; Boyd et al., 2011; Yuan, 2012); (d) specialized interior point methods (Li and Toh, 2010); and (e) proximal Newton type methods (Hsieh et al., 2014; Oztoprak et al., 2012). All the aforementioned methods are deterministic in nature. Precise (global) computational

[1] As long as $\lambda > 0$, the minimum of Problem (1) is finite (see Lemma 2) and there is a unique minimizer. In some variants of Problem (1), the diagonal entries of θ are not penalized—such an estimator can infact be written as a version of Problem (1), with $S \leftarrow S - \lambda \mathbb{I}_{p\times p}$ where, $\mathbb{I}$ is a $p \times p$ identity matrix. The minimum of this problem need not be finite. In this paper, however, we will consider formulation (1) where the diagonals are penalized.

guarantees are available for some of them. It appears that most of the aforementioned computational approaches for Problem (1), have a (worst-case) cost of at least $O(p^3)$ or possibly larger—this is perhaps not surprising, since for $\lambda = 0$, finding the MLE requires computing S^{-1} (assuming that the inverse exists), with cost $O(p^3)$. Many of the state-of-the art algorithms for Glasso (Hsieh et al., 2014; Friedman et al., 2007a) (for example) make clever use of the fact that solutions to Problem (1) are sparse, for large values of λ. Another important structural property of Glasso, that enables the scalable computation of Problem (1) is the exact thresholding property (Mazumder and Hastie, 2012a; Witten et al., 2011). The method is particularly useful for large values of λ, whenever the solution to the Glasso problem decomposes into smaller connected components; and becomes less effective when the solution to the Glasso problem is not sufficiently sparse. In short, computing solutions to Problem (1) become increasingly difficult as soon as p exceeds a few thousand.

All existing algorithms proposed for Glasso, to the best of our knowledge, are deterministic batch algorithms. To improve the computational scalability of Problem (1), we consider a different approach in this paper. Our approach, uses for the first time, ideas from stochastic convex optimization for the Glasso problem.

From sparse to dense regularization

We consider another traditionally important regularization scheme, given via the following optimization problem:

$$\underset{\theta \in \mathcal{M}_+}{\text{minimize}} \quad -\log\det\theta + \mathsf{Tr}(\theta S) + \frac{\lambda}{2}\sum_{i,j}\theta_{ij}^2, \tag{2}$$

for some value of $\lambda > 0$. This can be thought of as the ridge regularized version[2] of Problem (1). We will see in Section 6 that Problem (2) admits an analytic solution which requires computing the eigen-decomposition of S, albeit difficult when both n and p are large. Note that many of the tricks employed by modern solvers for Glasso, anticipating a sparse solution, no longer apply here. The stochastic convex optimization framework that we develop in this paper also applies to Problem (2), thereby enabling the computation of near-optimal solutions for problem-sizes where the exact solution becomes impractical to compute.

In this paper, we study a general version of Problems (1) and (2) by taking a convex combination of the ridge and ℓ_1 penalties:

$$\underset{\theta \in \mathcal{M}_+}{\text{minimize}} \quad \underbrace{-\log\det\theta + \mathsf{Tr}(\theta S)}_{:=f(\theta)} + \underbrace{\sum_{i,j}\left(\alpha\lambda|\theta_{ij}| + \frac{(1-\alpha)}{2}\lambda\theta_{ij}^2\right)}_{:=g_\alpha(\theta)}, \tag{3}$$

with $\alpha \in [0,1]$. Following Zou and Hastie (2005), we dub the above problem as the *elastic net* regularized version of the negative log-likelihood. Notice that for $\alpha = 1$ we get Glasso and $\alpha = 0$ corresponds to Problem (2). We propose a novel, scalable framework for computing near-optimal solutions to Problem (3) via techniques in stochastic convex optimization.

[2] Note that some authors (Warton, 2008) refer to a different problem as a ridge regression problem, namely one where one penalizes the trace of θ instead of the frobenius norm of θ. Such regularizers are often used in the context of regularized discriminant analysis (Friedman, 1989; Hastie et al., 1995). However, in this paper we will denote Problem (2) as the ridge regularized version of the Gaussian maximum likelihood problem.

1.1 Organization of the paper

The remainder of the paper is organized as follows. Section 2 provides an outline of the methodology and our contributions in this paper. We study deterministic proximal gradient algorithms in Section 3. We present the stochastic algorithms, proposed herein—Algorithm 2 and Algorithm 3 in Section 4. We describe the exact thresholding rule for Problem (3) in Section 5. The application of the stochastic algorithm (Algorithm 2) to the ridge regularized problem (Problem 2) is presented in Section 6. We present some numerical results that illustrate our theory in Section 7. The proofs are collected in Section 8, and some additional material are presented in the appendix.

2. Outline of the paper and our contributions

Deterministic Algorithms

The starting point of our analysis, is the study of a (deterministic) proximal gradient (Nesterov (2013); Beck and Teboulle (2009); Becker et al. (2011); Parikh and Boyd (2013)) algorithm (Algorithm 1) for solving Problem (3). A direct application of the proximal gradient algorithm (Nesterov (2013); Beck and Teboulle (2009), for example) to Problem (3) has some issues. Firstly, the basic assumption of Lipschitz continuity of the gradient $\nabla f(\theta)$, demanded by the proximal gradient algorithm, is not satisfied here. Secondly, the proximal operator associated with Problem (3) is difficult to compute, as it involves minimizing an ℓ_1 regularized quadratic function over the cone $\mathcal{M}_+$. We show that these hurdles may be overcome by controlling the step-size. Loosely speaking, we also establish that $\nabla f(\theta)$ satisfies a Lipschitz condition (and $f(\theta)$ satisfies a strong convexity condition) across the iterations of the algorithm—a notion that we make precise in Section 3. Using these key aspects of our algorithm, we derive a global linear convergence rate of Algorithm 1, even though the objective function is not strongly convex on the whole feasible set $\mathcal{M}_+$. Furthermore, the algorithm has an appealing convergence behavior that we highlight: its convergence rate is dictated by the condition number[3] of $\hat{\theta}$, a solution to Problem (3). For a given accuracy $\delta > 0$, our analysis implies that Algorithm 1 has a computational cost complexity of $O\left(p^3 \mathsf{cond}(\hat{\theta})^2 \log(\delta^{-1})\right)$ to reach a δ-accurate solution, where $\mathsf{cond}(\hat{\theta})$ is the condition number of $\hat{\theta}$. The computational bottleneck of the algorithm is the evaluation of the gradient of the smooth component at every iteration, which in this problem is $\nabla f(\theta) = -\theta^{-1} + S$. Computing the gradient requires performing a matrix inversion, an operation that scales with p as $O(p^3)$—we refer the reader to Figure 1 for an idea about the scalability behavior of direct dense matrix inversion for a $p \times p$ matrix, for different sizes of p.

Proximal gradient descent methods on the primal of the Glasso problem has been studied by Rolfs et al. (2012). Our approaches however, have some differences—our analysis hinges heavily on basic tools and techniques made available by the general theory of proximal methods; and we analyze a generalized version: Problem (3). The main motivation behind our analysis of Algorithm 1 is that it lays the foundation for the stochastic algorithms, our primary object of study in this paper.

[3] defined as the ratio of the largest eigenvalue over the smallest eigenvalue

Stochastic Algorithms

For large values of p (larger than a few thousand), Algorithm 1 slows down considerably, due to repeated computation of the inverse: θ^{-1} (See also Figure 1) across the proximal gradient iterations. Even if the matrix θ is sparse and sparse numerical linear algebra methods are used for computing θ^{-1}, the computational cost depends quite heavily upon the sparsity pattern of θ and the re-ordering algorithm used to reduce fill-ins; and need not be robust[4] across different problem instances. Thus, our key strategy in the paper is to develop a stochastic method that completely bypasses the exact computation (via direct matrix inversion) of the gradient $\nabla f(\theta) = S - \theta^{-1}$. We propose to draw N_k samples $z_1, \ldots, z_{N_k}$ (at iteration k) from $\mathbf{N}(0, \theta_{k-1}^{-1})$ to form a noisy estimate $S - N_k^{-1} \sum_{k=1}^{N_k} z_i z_i'$ of the gradient $S - \theta_{k-1}^{-1}$. This scheme forms the main workhorse of our stochastic proximal gradient algorithm, which we call Algorithm 2.

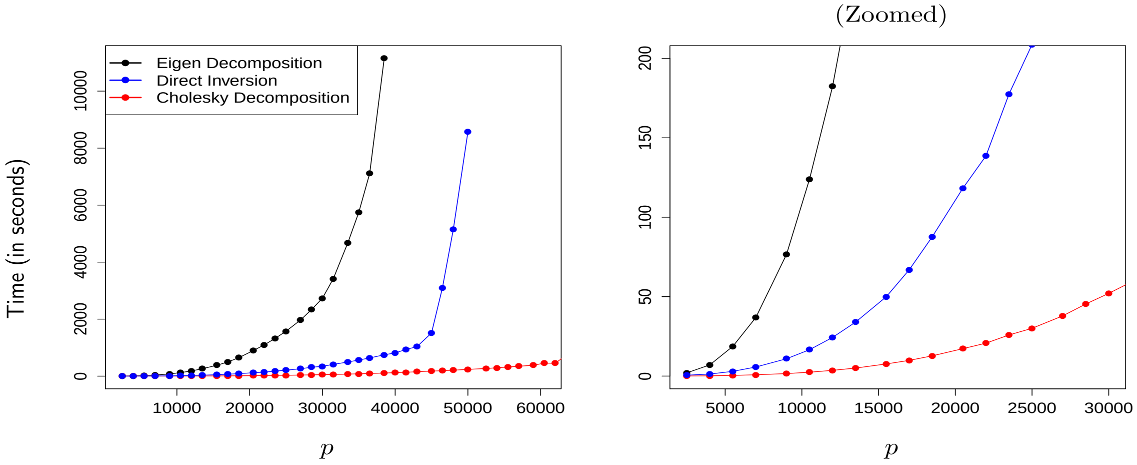

Figure 1: Figure showing the times in seconds to perform a direct eigen-decomposition, inversion and Cholesky decomposition using dense direct numerical linear algebra methods, for real symmetric matrices with size of upto $p = 65,000$. Eigen decompositions and matrix inversions are less memory friendly, when compared to Cholesky decompositions for large problem sizes. The timings displayed in the graphs support the practical feasibility of using Cholesky decomposition methods for large matrices—a main workhorse for the stochastic optimization algorithms proposed in the paper. [Right panel] displays a zoomed in version of the left panel plot, showing that Cholesky decompositions are significantly faster than inversion and eigen-decomposition methods even for smaller problems $p \leq 30,000$. The tail of the direct inversion curve on the left deviates from the $O(p^3)$ trend because the storage requirement has exceeded the capacity of main memory. Thus, the extra time is consumed by the slower virtual memory access. [The matrices used here were sparse with proportion of non-zeros $10/p$, positive definite with the reciprocal of the condition number given by 0.2—we used the Matlab function `sprandsym` to generate the matrices.]

[4] In fact, in our experiments we observed that Matlab performs dense Cholesky decomposition more efficiently than sparse Cholesky decomposition, even when the matrix is sparse. This is due in part to multithreading: the dense Cholesky decomposition is automatically multithreaded in Matlab, but the sparse Cholesky decomposition is not. Another reason is the difficulty of finding a good re-ordering algorithm to limit fill-ins when performing sparse Cholesky decomposition.

Stochastic optimization algorithms based on noisy estimates of the gradient have a long history that goes back to the pioneering works of Robbins and Monro (1951); Kiefer and Wolfowitz (1952). As datasets encountered by statisticians in the modern day grow larger and the optimization problems associated with statistical estimation tasks become increasingly challenging, the importance of stochastic algorithms to deliver scalable solvers is being progressively recognized in recent years. See for instance, the recent works in the optimization and machine learning communities (Bertsekas (2011); Duchi et al. (2012); Shalev-Shwartz and Zhang (2013); Konečný and Richtárik (2013); Xiao and Zhang (2014); Atchade et al. (2014), and the references therein). We note however, that our stochastic optimization formulation of Problem (3) differs from the usual stochastic optimization problem (for instance as in Bertsekas (2011)) which solves problems of the form

$$\underset{\theta}{\text{minimize}} \quad \int f(\theta; x)\pi(\mathrm{d}x) + g(\theta), \tag{4}$$

for an intractable integral $\int f(\theta; x)\pi(\mathrm{d}x)$, where, the map $\theta \mapsto f(\theta; x)$ is smooth, and g is possibly non-smooth. A special instance of (4) is when π is a discrete probability distribution over a very large set, making the integral $\int f(\theta; x)\pi(\mathrm{d}x) = \frac{1}{N}\sum_{i=1}^{N} f(\theta; x_i)$ a large sum and difficult to work with. We make the following remarks that highlight the differences between our approach and generic approaches for Problem (4):

- Problem (3) does not admit a straightforward representation of the form (4).
- The gradient $\nabla f(\theta) = S - \theta^{-1}$ has the integral representation $S - \int xx'\pi_\theta(\mathrm{d}x)$, where π_θ is the density of $\mathbf{N}(0, \theta^{-1})$, which depends on θ — a distinctive feature that sets our stochastic optimization framework apart from Problem (4).
- Last, but not least, the gradient map $\theta \mapsto \nabla f(\theta)$ is not Lipschitz continuous on $\mathcal{M}_+$, the feasible set of Problem (3).

A main contribution of our paper is to address the above challenges in the context of the stochastic optimization framework being proposed herein. In fact, our stochastic optimization framework is more in sync with the Robbins-Monro algorithm (Robbins and Monro (1951)) and can be viewed as a large-scale and non-smooth variant of the Robbins-Monro algorithm, along the lines of Atchade et al. (2014). Note however, that the theory of Atchade et al. (2014) cannot be directly applied here, as it requires the classical Lipschitz-continuity assumption of the smooth component of the objective function, and the ability to compute the proximal map of the non-smooth component. As explained above, these properties are not readily available in our case.

The main cost of Algorithm 2 lies with generating multivariate Gaussian random variables from $\mathbf{N}(0, \theta^{-1})$. A given iteration of Algorithm 2 is more cost-effective than an iteration of the deterministic algorithm, if the Monte Carlo sample size used in that iteration is smaller than p. This is because the cost of approximating θ^{-1} using p random samples from $\mathbf{N}(0, \theta^{-1})$ is similar to the cost of computing θ^{-1} by direct matrix inversion. We show that with an appropriate choice of the Monte Carlo batch size sequence $\{N_k\}$ (see Section 4.1 for details), Algorithm 2 reaches a solution with accuracy δ, before the Monte Carlo sample size becomes larger than p if $p \geq \mathsf{cond}(\hat{\theta})^2\delta^{-1}$. This result implies that Algorithm 2 is more cost-effective than Algorithm 1 in finding δ-accurate solutions in cases when p is large, the solution $\hat{\theta}$ is well-conditioned, and we seek a low-accuracy approximation of $\hat{\theta}$. The total cost of Algorithm 2 is then $O\left(p^3 \mathsf{cond}(\hat{\theta})^2 \log(\delta^{-1})\right)$. While on the surface, the cost looks similar to Algorithm 1 which performs a direct matrix inversion at every iteration, the constant involved in the big-O notation favors Algorithm 2 —see for example, Figure 1 showing the differences in computation times between a dense Cholesky decomposition and a direct dense matrix inversion. This is further substantiated in our numerical experiments (Section 7) where we do systematic comparisons between Algorithms 1 and 2.

A deeper investigation of our stochastic optimization scheme (Algorithm 2) outlined above, reveals the following. At each iteration k, all the random variables (samples) used to estimate θ_{k-1}^{-1} are discarded, and new random variables are generated to approximate θ_k^{-1}. We thus ask, is there a modified algorithm that makes clever use of the information associated with an approximate θ_{k-1}^{-1} to approximate θ_k^{-1}? In this vein, we propose a new algorithm: Algorithm 3 which recycles previously generated samples. Algorithm 3 has a per-iteration cost of $O(Np^3)$ when a Cholesky factorization is used to generate the Gaussian random variables, and where N is the Monte Carlo batch-size. The behavior of the algorithm is more complex, and thus developing a rigorous convergence guarantee with associated computational guarantees analogous to Algorithm 2 is beyond the scope of the current paper. We however, present some global convergence results on the algorithm. In particular, we show that when the sequence produced by Algorithm 3 converges, it necessarily converges to the solution of Problem (3).

Dense problems

We emphasize that a sizable component of our work relies on the speed and efficiency of modern dense numerical linear algebra methods for scalability, and thus our approach is relatively agnostic to the sparsity level of $\hat{\theta}$, a solution to Problem (3). In other words, our approach adapts to Problem (2) for large n and p, a problem which is perhaps not favorable for several current specialized implementations for Problem (1).

Exact covariance thresholding

We also extend the exact thresholding rule (Mazumder and Hastie, 2012a) originally proposed for the Glasso problem, to the more general case of Problem (3). Our result established herein, implies that the connected components of the graph $(1(|s_{ij}| > \lambda\alpha))$ are *exactly* equal to the connected components of the graph induced by the non-zeros of $\hat{\theta}$, a solution to Problem (3). This can certainly be used as a wrapper around any algorithm to solve Problem (3); and leads to dramatic performance gains whenever the size of the largest connected component of $(1\left(|s_{ij}| > \lambda\alpha\right))$ is sufficiently smaller than p.

We note that developing the fastest algorithmic implementation for Problem (3) or its special case, Glasso, is neither the intent nor focus of this paper. We view our work as one that proposes a new framework based on stochastic optimization that enables the *scalable computation* for the general class of Problems (3), across a wide range of the regularization parameters. The scalability properties of our proposal seem to be favorable over deterministic batch methods and in particular, proximal gradient descent methods tailored for Problem (3).

2.1 Notation

Throughout the paper, the regularization parameters λ and $\alpha \in (0,1]$, appearing in Problem (3) are assumed fixed and given. Let $\mathcal{M}$ denote the set of $p \times p$ symmetric matrices with inner product $\langle A, B\rangle = \mathsf{Tr}(A'B)$ and the Frobenius norm $\|A\|_{\mathsf{F}} \overset{\text{def}}{=} \sqrt{\langle A, A\rangle}$. $\mathcal{M}_+$ denotes the set of positive definite elements of $\mathcal{M}$. Let f be the function $\mathcal{M} \to (0,\infty]$ defined by

$$f(\theta) = \begin{cases} -\log\det\theta + \mathsf{Tr}(\theta S) & \text{if } \theta \in \mathcal{M}_+ \\ +\infty & \text{if } \ \theta \in \mathcal{M} \setminus \mathcal{M}_+. \end{cases}$$

We shall write the regularization term in Problem (3) as

$$g_\alpha(\theta) \overset{\text{def}}{=} \sum_{ij} \left(\alpha\lambda|\theta_{ij}| + \frac{(1-\alpha)}{2}\lambda\theta_{ij}^2 \right),$$

and

$$\phi_\alpha(\theta) \overset{\text{def}}{=} f(\theta) + g_\alpha(\theta), \ \ \theta \in \mathcal{M}. \tag{5}$$

For a matrix $A \in \mathcal{M}$, $\|A\|_2$ denotes the spectral norm of A, $\lambda_{\mathsf{min}}(A)$ (respectively $\lambda_{\mathsf{max}}(A)$) denotes the smallest (respectively, the largest) eigenvalue of A, and $\|A\|_1 \overset{\text{def}}{=} \sum_{i,j} |A_{ij}|$. For a subset $\mathcal{D} \subseteq \mathcal{M}$, $\iota_{\mathcal{D}}$ denotes the indicator function of $\mathcal{D}$, i.e.

$$\iota_D(u) \overset{\text{def}}{=} \begin{cases} 0 & \text{if } u \in \mathcal{D} \\ +\infty & \text{otherwise.} \end{cases}$$

For $\theta \in \mathcal{M}_+$, and $\gamma > 0$, we denote the proximal operator associated with Problem (3) as

$$\bar{T}_\gamma(\theta;\alpha) \overset{\text{def}}{=} \underset{u\in\mathcal{M}_+}{\mathsf{Argmin}} \left\{ g_\alpha(u) + \frac{1}{2\gamma} \left\| u - \theta + \gamma(S - \theta^{-1}) \right\|_{\mathsf{F}}^2 \right\}. \tag{6}$$

For $0 < \ell \le \psi$, we define

$$\mathcal{M}_+(\ell,\psi) \overset{\text{def}}{=} \left\{ \theta \in \mathcal{M}_+ : \ \lambda_{\mathsf{min}}(\theta) \ge \ell, \ \text{ and } \ \lambda_{\mathsf{max}}(\theta) \le \psi \right\}.$$

3. A proximal gradient algorithm for Problem (3)

We begin this section with a brief review of proximal gradient algorithms, following Nesterov (2013), which concerns the minimization of the following generic convex optimization problem:

$$\min_{\omega\in\Omega} \ \left\{ \bar{\phi}(\omega) \overset{\text{def}}{=} \bar{f}(\omega) + \bar{g}(\omega) \right\}, \tag{7}$$

where, Ω is a convex subset of a Euclidean space with norm $\|\cdot\|$; $\bar{g}(\cdot)$ is a closed convex function and $\bar{f}(\cdot)$ is convex, smooth on Ω satisfying:

$$\|\nabla\bar{f}(\omega) - \nabla\bar{f}(\omega')\| \le \bar{L}\|\omega - \omega'\|, \tag{8}$$

for $\omega, \omega' \in \Omega$ and $0 < \bar{L} < \infty$. The main ingredient in proximal gradient descent methods is the efficient computation of the proximal-operator ("prox-operator" for short), given by:

$$\bar{T}_\gamma(\bar{\omega}) \overset{\text{def}}{=} \underset{\omega\in\Omega}{\mathsf{Argmin}} \ \ \|\omega - (\bar{\omega} - \gamma\nabla f(\bar{\omega}))\|^2 + \bar{g}(\omega), \tag{9}$$

for some choice of $0 < \gamma \leq 1/\bar{L}$. The following simple recursive rule:

$$\omega_{k+1} = \bar{T}_\gamma(\omega_k), \quad k \geq 1,$$

for some initial choice of $\omega_1 \in \Omega$ and $\gamma = 1/\bar{L}$, then leads to a solution of Problem (7) (See for example, Nesterov (2013)).

Problem (3) has striking similarities to an optimization problem of the form (7), with $\bar{f}(\cdot) = f(\cdot)$, $\Omega = \mathcal{M}_+$ endowed with the Frobenius norm, $\bar{g}(\cdot) = g_\alpha(\cdot)$, and with $\bar{T}_\gamma$ given by (6). However, the use of the proximal gradient algorithm for Problem (3) presents some immediate challenges since:

- The gradient of the smooth component, namely, $\nabla f(\theta) = -\theta^{-1} + S$ is not Lipschitz on the entire domain $\mathcal{M}_+$ (as required in (8)), due to the unboundedness of the map $\theta \mapsto \theta^{-1}$.

- The corresponding proximal map $\bar{T}_\gamma(\cdot\, ; \alpha)$ defined in (6) need not be simple to compute.

Our first task in this paper, is to show how each of the above problems can be alleviated. We note that Rolfs et al. (2012) also analyze a proximal gradient descent algorithm for the case $\alpha = 1$. We present here a self-contained analysis: our proofs have some differences with that of Rolfs et al. (2012); and lays the foundation for the stochastic optimization scheme that we analyze subsequently. Loosely speaking, we will show that even if the function $f(\theta)$ does not have Lipschitz continuous gradient on the *entire* feasible set $\mathcal{M}_+$, it *does* satisfy (8) across the iterations of the proximal gradient algorithm. In addition, we demonstrate that by appropriately choosing the step-size γ, the proximal map $\bar{T}_\gamma(\cdot\, ; \alpha)$ can be computed by "dropping" the constraint $\theta \in \mathcal{M}_+$. We formalize the above in the following discussion.

For $\alpha \in [0, 1]$, $\gamma > 0$, and $\theta \in \mathcal{M}$ (i.e., the set of $p \times p$ symmetric matrices), the proximal operator associated with the function g_α is defined as

$$\mathrm{Prox}_\gamma(\theta; \alpha) \overset{\text{def}}{=} \underset{u \in \mathcal{M}}{\mathsf{Argmin}} \left\{ g_\alpha(u) + \frac{1}{2\gamma} \|u - \theta\|_{\mathsf{F}}^2 \right\}.$$

This operator has a very simple form. It is a matrix whose (i, j)th entry is given by:

$$\left(\mathrm{Prox}_\gamma(\theta; \alpha)\right)_{ij} = \begin{cases} 0 & \text{if } |\theta_{ij}| < \alpha\lambda\gamma \\ \frac{\theta_{ij} - \alpha\lambda\gamma}{1 + (1-\alpha)\lambda\gamma} & \text{if } \theta_{ij} \geq \alpha\lambda\gamma \\ \frac{\theta_{ij} + \alpha\lambda\gamma}{1 + (1-\alpha)\lambda\gamma} & \text{if } \theta_{ij} \leq -\alpha\lambda\gamma\,. \end{cases} \tag{10}$$

For $\gamma > 0$, and $\theta \in \mathcal{M}_+$, we consider a seemingly minor modification of the operator (6), given by:

$$\begin{aligned} T_\gamma(\theta; \alpha) &\overset{\text{def}}{=} \underset{u \in \mathcal{M}}{\mathsf{Argmin}} \left\{ g_\alpha(u) + \tfrac{1}{2\gamma} \left\| u - \theta + \gamma(S - \theta^{-1}) \right\|_{\mathsf{F}}^2 \right\} \\ &= \mathrm{Prox}_\gamma\left(\theta - \gamma(S - \theta^{-1}); \alpha\right). \end{aligned} \tag{11}$$

Compared to (6), one can notice that in (11) the positive definiteness constraint is relaxed. It follows from (10) that $T_\gamma(\theta; \alpha)$ is straightforward to compute. Notice that if $T_\gamma(\theta; \alpha)$ is positive definite, then $T_\gamma(\theta; \alpha) = \bar{T}_\gamma(\theta; \alpha)$. We will show that if γ is not too large then indeed $T_\gamma(\theta; \alpha) = \bar{T}_\gamma(\theta; \alpha)$ for all θ in certain subsets of $\mathcal{M}_+$.

At the very onset, we present a result which provides bounds on the spectrum of $\hat{\theta}$, a solution to Problem (3). The following lemma can be considered as a generalization of the result of Lu (2009) obtained for the Glasso problem (with $\alpha = 1$). Let us define the following quantities: $\lambda_1 \overset{\text{def}}{=} \alpha\lambda$, $\lambda_2 \overset{\text{def}}{=} (1-\alpha)\lambda/2$, $\mu \overset{\text{def}}{=} \|S\|_2 + \lambda_1 p$,

$$\begin{aligned}
\ell_\star &\overset{\text{def}}{=} \begin{cases} \frac{-\mu+\sqrt{\mu^2+8\lambda_2}}{4\lambda_2} & \text{if } \alpha \in [0,1) \\ \frac{1}{\mu} & \text{if } \alpha = 1, \end{cases} \\
U_1 &\overset{\text{def}}{=} \frac{1}{\lambda_1}\left(p - \ell_\star \mathrm{T}r(S) - 2p\lambda_2\ell_\star^2\right) \\
c(t) &\overset{\text{def}}{=} \frac{1}{\lambda_1(1-t)}\left(\lambda_1\|\theta(t)\|_1 - t\lambda_1 \mathrm{T}r(\theta(t)) + \lambda_2\|\theta(t)\|_{\mathsf{F}}^2\right) - \frac{\lambda_2\ell_\star^2 p}{\lambda_1(1-t)} \\
U_2 &\overset{\text{def}}{=} \inf_{t\in(0,1)} c(t),
\end{aligned} \tag{12}$$

where, we take $t \in (0,1)$ and $\theta(t) \overset{\text{def}}{=} (S + t\lambda_1 I)^{-1}$.

Lemma 1 *If $\hat{\theta}$ is a solution to Problem* (3) *with $\lambda > 0$, then $\hat{\theta}$ is unique, and $\hat{\theta} \in \mathcal{M}_+(\ell_\star, \psi_{UB})$, with $\psi_{UB} = \min\{U_1, U_2\}$, where, U_1, U_2 are as defined in* (12)*. In other words, we have the following bounds on the spectrum of $\hat{\theta}$*

$$\lambda_{\min}(\hat{\theta}) \ge \ell_\star, \quad \lambda_{\max}(\hat{\theta}) \le \psi_{UB}.$$

Proof *The proof is presented in Section 8.1.* ■

We make a few remarks about the bounds in (12).

• Computing U_2 requires performing a one dimensional minimization which can be carried out quite easily. Conservative but *valid* bounds can be obtained by replacing U_2 by evaluations of $c(\cdot)$ at some values of $t \in (0,1)$ for example: $t = \frac{1}{2}$ and $t = 0+$ (provided S is invertible).

• Since the condition number of $\hat{\theta}$ is $\mathsf{cond}(\hat{\theta}) = \lambda_{\mathsf{max}}(\hat{\theta})/\lambda_{\mathsf{min}}(\hat{\theta})$, the result above implies that $\mathsf{cond}(\hat{\theta}) \le \psi_{UB}/\ell_\star$. We note, however, that this upper bound $\psi_{UB}/\ell_\star$ may not be an accurate estimate of $\mathsf{cond}(\hat{\theta})$.

We now present an important property (Lemma 2) of the proximal gradient update step, for our problem. Towards this end, we define $\nu \overset{\text{def}}{=} \lambda_{\mathsf{min}}(S) - \lambda_1 p$ and

$$\psi_\star^1 \overset{\text{def}}{=} \begin{cases} \frac{-\nu+\sqrt{\nu^2+8\lambda_2}}{4\lambda_2} & \text{if } \alpha \in [0,1), \\ \frac{1}{\nu} & \text{if } \alpha = 1 \text{ and } \nu > 0 \\ +\infty & \text{if } \alpha = 1 \text{ and } \nu \le 0. \end{cases}$$

It is obvious that $0 < \ell_\star \le \psi_\star^1 \le \infty$. We also define

$$\psi_\star \overset{\text{def}}{=} \min\left(\psi_\star^1, \psi_{UB} + \sqrt{p}\left(\psi_{UB} - \ell_\star\right)\right).$$

Lemma 2 *Take $\gamma \in (0, \ell_\star^2]$ and let $\{\theta_j,\ j \geq 0\}$ be a sequence such that $\theta_j = T_\gamma(\theta_{j-1}; \alpha)$, for $j \geq 1$. If $\theta_0 \in \mathcal{M}_+(\ell_\star, \min\{\psi_{UB}, \psi_\star^1\})$, then $\theta_j \in \mathcal{M}_+(\ell_\star, \psi_\star)$ for all $j \geq 0$.*

Proof See Section 8.2. ■

In the special case of Glasso ($\alpha = 1$), the results of Lemma 2 correspond to those obtained by Rolfs et al. (2012). Our proof, however, has differences since we rely more heavily on basic properties of proximal maps.

Lemma 2 shows that for appropriate choices of $\gamma > 0$, the two proximal maps T_γ and $\bar{T}_\gamma$ produce the identical sequences that remain in the set $\mathcal{M}_+(\ell_\star, \psi_\star)$. This suggests that one can solve Problem (3) using the proximal operator T_γ, as the next result shows.

Theorem 3 *Fix arbitrary $0 < \ell < \psi < \infty$. For $k \geq 1$, let $\{\theta_j,\ 0 \leq j \leq k\}$ be a sequence obtained via the map T_γ: $\theta_{j+1} = T_\gamma(\theta_j; \alpha)$, for some $\gamma \in (0, \ell^2]$. Suppose that $\hat{\theta}, \theta_j \in \mathcal{M}_+(\ell, \psi)$, $0 \leq j \leq k$. Then*

$$\left\|\theta_k - \hat{\theta}\right\|_F^2 \leq \rho^k \left\|\theta_0 - \hat{\theta}\right\|_F^2, \quad \textit{and} \quad \left\{\phi_\alpha(\theta_k) - \phi_\alpha(\hat{\theta})\right\} \leq \frac{\left\|\theta_0 - \hat{\theta}\right\|_F^2}{2\gamma} \min\left\{\frac{1}{k}, \rho^k\right\}, \qquad (13)$$

where $\rho = 1 - \frac{\gamma}{\psi^2}$.

Proof See Section 8.3. ■

Remark 4 *If $\ell = \ell_\star$ and $\psi = \psi_\star$, and $\theta_0 \in \mathcal{M}_+(\ell_\star, \min\{\psi_{UB}, \psi_\star^1\})$, then the assumption that $\hat{\theta}, \theta_j \in \mathcal{M}_+(\ell, \psi)$, $0 \leq j \leq k$ is redundant, as shown in Lemma 1-2, and (13) holds.* □

An appealing feature of the iteration $\theta_{k+1} = T_\gamma(\theta_k; \alpha)$ is that its convergence rate is *adaptive*, i.e., the algorithm automatically adapts itself to the fastest possible convergence rate dictated by the condition number of $\hat{\theta}$. This is formalized in the following corollary:

Corollary 5 *Let $0 < \ell_{\star\star} < \psi_{\star\star} < \infty$ be such that $\lambda_{\mathsf{min}}(\hat{\theta}) > \ell_{\star\star}$, and $\lambda_{\mathsf{max}}(\hat{\theta}) < \psi_{\star\star}$. Let $\{\theta_k,\ k \geq 0\}$ be a sequence obtained via the map T_γ: $\theta_{j+1} = T_\gamma(\theta_j; \alpha)$, for some $\gamma \in (0, \ell_{\star\star}^2]$. If $\lim_k \theta_k = \hat{\theta}$, then there exists $k_0 \geq 0$, such that for all $k \geq k_0$,*

$$\left\|\theta_k - \hat{\theta}\right\|_F^2 \leq \left(1 - \frac{\gamma}{\psi_{\star\star}^2}\right)^{k-k_0} \left\|\theta_{k_0} - \hat{\theta}\right\|_F^2.$$

Proof By assumption, $\hat{\theta}$ belongs to the interior of $\mathcal{M}_+(\ell_{\star\star}, \psi_{\star\star})$. Since $\theta_k \to \hat{\theta}$, there exists $k_0 \geq 0$, such that $\theta_k \in \mathcal{M}_+(\ell_{\star\star}, \psi_{\star\star})$ for $k \geq k_0$. Then we apply the bound (13), and the lemma follows. ■

The analysis above suggests the following practical algorithm for Problem (3). Let $\{\gamma_k\}$ denote a sequence of positive step-sizes with $\lim_k \gamma_k = 0$. An example of such a sequence is $\gamma_k = \gamma_0/2^k$, for some $\gamma_0 > 0$. For convenience, we summarize in Algorithm 1, the deterministic proximal gradient algorithm for Problem (3).

Algorithm 1 (Deterministic Proximal Gradient)
Set $\mathsf{r} = 0$.

1. *Choose* $\theta_0 \in \mathcal{M}_+$.

2. *Given* θ_k*, compute:* $\theta_{k+1} = T_{\gamma_{\mathsf{r}}}(\theta_k; \alpha)$.

3. *If* $\lambda_{min}(\theta_{k+1}) \leq 0$*, then restart: set* $k \leftarrow 0$*,* $\mathsf{r} \leftarrow \mathsf{r} + 1$*, and go back to (1). Otherwise, set* $k \leftarrow k + 1$ *and go back to (2).*

We present a series of remarks about Algorithm 1:

• *Positive Definiteness.* In Step 3, positive definiteness is tested and the algorithm is restarted with a smaller step-size, if θ_{k+1} is no longer positive definite. The smallest eigenvalue of θ_{k+1}, i.e., $\lambda_{\mathsf{min}}(\theta_{k+1})$ can be efficiently computed by several means: (a) it can be computed via the Lanczos process (see e.g. Golub and Van Loan (2013) Theorem 10.1.2); (b) it may also be computed as a part of the step that approximates the spectral interval of θ_{k+1} using the procedure of Chen et al. (2011) (c) a Cholesky decomposition of θ_{k+1} also returns information about whether θ_{k+1} is positive definite or not.

An efficient implementation of the algorithm is possible by making Step 3 implicit. For instance the positive definiteness of θ_{k+1} can be checked as part of the computation of the gradient $\nabla f(\theta_{k+1}) = S - \theta_{k+1}^{-1}$ in Step 2.

• *Step Size.* If the initial step-size satisfies $\gamma_0 \leq \ell_\star^2$ and $\theta_0 \in \mathcal{M}_+(\ell_\star, \psi_\star)$, the algorithm is never re-initialized according to Lemma 2, and Theorem 3 holds. However, it is important to notice that Lemma 2 and Theorem 3 present a worst case analysis scenario and in practice the choice $\gamma_0 = \ell_\star^2$ can be overly conservative. In fact, Corollary 5 dictates that a better choice of step-size is $\gamma_0 = \lambda_{\mathsf{min}}(\hat{\theta})^2$. Obviously $\lambda_{\mathsf{min}}(\hat{\theta})$ is rarely known, but what this implies is that, in practice, one should initialize the algorithm with a large step-size and rely on the re-start trick (Step 3) to reduce the step-size, when θ_{k+1} is not positive definite.

• *Adaptive Convergence Rate.* We have seen in Corollary 5 that the convergence rate of the sequence $\{\theta_k\}$ improves with the iterations. This adaptive convergence rate behavior makes the cost-complexity analysis of Algorithm 1 more complicated. However, to settle ideas, if we set θ_0 close to $\hat{\theta}$, and the step-size obeys $\gamma \approx \lambda_{\mathsf{min}}(\hat{\theta})^2$, Theorem 3 and Corollary 5 imply that the number of iterations of Algorithm 1 needed to reach the precision δ (that is $\left\|\theta_k - \hat{\theta}\right\|_{\mathsf{F}}^2 \leq \delta$) is

$$O\left(-\frac{\psi_{\star\star}^2}{\ell_{\star\star}^2}\log\delta\right) \approx O\left(-\mathsf{cond}(\hat{\theta})^2 \log\delta\right).$$

• *Computational Cost.* The bottleneck of Algorithm 1 is the computation of the inverse θ_k^{-1}, which in general entails a computational cost of $O(p^3)$—See Figure 1 showing the computation times of matrix inversions for real symmetric $p \times p$ matrices, in practice. It follows that in the setting considered above, the computational cost of Algorithm 1 to achieve a δ-accurate solution is $O\left(p^3\mathsf{cond}(\hat{\theta})^2\log(1/\delta)\right)$.

4. Stochastic Optimization Based Algorithms

When p is large (for example, $p = 5,000$ or larger), the computational cost of Algorithm 1 becomes prohibitively expensive due to the associated matrix inversions—this is a primary motivation behind the stochastic optimization methods that we develop in this section. For $\theta \in \mathcal{M}_+$, let π_θ denote the density of $\mathbf{N}(0, \theta^{-1})$, the mean-zero normal distribution on $\mathbb{R}^p$ with covariance matrix θ^{-1}. We begin with the elementary observation that

$$\theta^{-1} = \int zz' \pi_\theta(\mathrm{d}z).$$

This suggests that on $\mathcal{M}_+$, we can approximate the gradient $\nabla f(\theta) = S - \theta^{-1}$ by $S - N^{-1} \sum_{j=1}^N z_j z_j'$, where $z_{1:N} \overset{\text{i.i.d.}}{\sim} \pi_\theta$; here, the notation $z_{1:N}$ denotes a collection of random vectors $z_i, i \leq N$.

To motivate the stochastic algorithm we will first establish an analog of Lemma 2, showing that iterating the stochastic maps obtained by replacing θ_{j-1}^{-1} in computing $T_\gamma(\theta_{j-1}; \alpha)$ in (11) by the Monte Carlo estimate described above, produces sequences that remain positive definite with high probability. Towards this end, fix $\gamma > 0$; a sequence of (positive) Monte Carlo batch-sizes: $\{N_k,\ k \geq 1\}$; and consider the stochastic process $\{\theta_k,\ k \geq 0\}$ defined as follows. First, we fix $\theta_0 \in \mathcal{M}_+$. For $k \geq 1$, and given the sigma-algebra $\mathcal{F}_{k-1} \overset{\text{def}}{=} \sigma(\theta_0, \ldots, \theta_{k-1})$:

$$\text{generate } z_{1:N_k} \overset{\text{i.i.d.}}{\sim} \mathbf{N}(0, \theta_{k-1}^{-1}), \quad \text{compute} \quad \Sigma_k = \frac{1}{N_k} \sum_{j=1}^{N_k} z_j z_j', \tag{14}$$

and set:

$$\theta_k = \mathrm{Prox}_\gamma \left(\theta_{k-1} - \gamma \left(S - \Sigma_k \right) \right). \tag{15}$$

For any $0 < \ell \leq \psi \leq \infty$, we set

$$\tau(\ell, \psi) \overset{\text{def}}{=} \inf \left\{ k \geq 0 :\ \theta_k \notin \mathcal{M}_+(\ell, \psi) \right\},$$

with the convention that $\inf \emptyset = \infty$. For a random variable $\Psi \geq \ell$, we define $\tau(\ell, \Psi)$ as equal to $\tau(\ell, \psi)$ on $\{\Psi = \psi\}$.

Given $\epsilon > 0$, we define $\mu_\epsilon \overset{\text{def}}{=} \|S\|_2 + (\lambda_1 + \epsilon)p$,

$$\ell_\star(\epsilon) \overset{\text{def}}{=} \begin{cases} \frac{-\mu_\epsilon + \sqrt{\mu_\epsilon^2 + 8\lambda_2}}{4\lambda_2} & \text{if } \alpha \in [0, 1) \\ \frac{1}{\mu_\epsilon} & \text{if } \alpha = 1. \end{cases}$$

Similarly, define $\nu_\epsilon \overset{\text{def}}{=} \lambda_{\mathsf{min}}(S) - (\lambda_1 + \epsilon)p$,

$$\psi_\star^1(\epsilon) \overset{\text{def}}{=} \begin{cases} \frac{-\nu_\epsilon + \sqrt{\nu_\epsilon^2 + 8\lambda_2}}{4\lambda_2} & \text{if } \alpha \in [0, 1), \\ \frac{1}{\nu_\epsilon} & \text{if } \alpha = 1 \text{ and } \nu_\epsilon > 0 \\ +\infty & \text{if } \alpha = 1 \text{ and } \nu_\epsilon \leq 0. \end{cases}$$

It is easy to check that $0 < \ell_\star(\epsilon) \leq \ell_\star \leq \psi_\star^1 \leq \psi_\star^1(\epsilon) \leq \infty$.

The following theorem establishes the convergence of the stochastic process θ_k, produced via the stochastic optimization scheme (15).

Theorem 6 *Let $\{\theta_k,\ k \geq 0\}$ be the stochastic process defined by the rules (14-15). Fix $\epsilon > 0$. Suppose that $\theta_0 \in \mathcal{M}_+(\ell_\star(\epsilon), \min(\psi_{UB}, \psi_\star^1(\epsilon)))$. Then there exists a random variable $\Psi_\star(\epsilon) \geq \ell_\star(\epsilon)$ such that*

$$\mathbb{P}\left[\tau\left(\ell_\star(\epsilon), \Psi_\star(\epsilon)\right) = \infty\right] \geq 1 - 4p^2 \sum_{j\geq 1} \exp\left(-\min\left(1, \frac{\epsilon^2 \ell_\star^2(\epsilon)}{16}\right) N_{j-1}\right).$$

If $\sum_j N_j^{-1} < \infty$, then $\mathbb{E}(\Psi_\star(\epsilon)^2) < \infty$ (hence $\Psi_\star(\epsilon)$ is finite almost surely), and on $\{\tau\left(\ell_\star(\epsilon), \Psi_\star(\epsilon)\right) = \infty\}$, $\lim_{k\to\infty} \theta_k = \hat{\theta}$.

Proof See Section 8.4. ■

Growth Condition on the Monte Carlo batch size

If we let the Monte Carlo sample size N_k increase as

$$N_k \geq \frac{3 \log p}{\min\left(1, \ell_\star^2(\epsilon)\epsilon^2/16\right)} + \alpha k^q,$$

for some $q > 1$, then $\sum_j N_j^{-1} < \infty$, and the bound in Theorem 3 above, becomes

$$\mathbb{P}\left[\tau\left(\ell_\star(\epsilon), \Psi_\star(\epsilon)\right) = \infty\right] \geq 1 - \frac{4\mu}{p},$$

where $\mu = \sum_{j\geq 0} \exp\left(-\alpha \min(1, \ell_\star^2(\epsilon)\epsilon^2/16) j^q\right) < \infty$. Hence for high-dimensional problems, and for moderately large Monte Carlo sample sizes, $\mathbb{P}\left[\tau\left(\ell_\star(\epsilon), \Psi_\star(\epsilon)\right) = \infty\right]$ can be made very close to one—this guarantees that positive definiteness of the process $\{\theta_k,\ k \geq 0\}$ is maintained and the sequence converges to $\hat{\theta}$, with high probability. The convergence rate of the process is quantified by the following theorem:

Theorem 7 *Let $\{\theta_k,\ k \geq 0\}$ be the stochastic process defined by (14-15). For some $0 < \ell \leq \psi \leq +\infty$, suppose that $\theta_0, \hat{\theta} \in \mathcal{M}_+(\ell, \psi)$, and $\gamma \leq \ell^2$. Then*

$$\mathbb{E}\left[\mathbf{1}_{\{\tau(\ell,\psi)>k\}} \left\|\theta_k - \hat{\theta}\right\|_{\mathsf{F}}^2\right] \leq \left(1 - \frac{\gamma}{\psi^2}\right)^k \left\|\theta_0 - \hat{\theta}\right\|_{\mathsf{F}}^2 + 2\gamma^2 \ell^{-2}(p+p^2) \sum_{j=1}^{k} N_j^{-1} \left(1 - \frac{\gamma}{\psi^2}\right)^{k-j}. \quad (16)$$

Proof See Section 8.5. ■

As with the deterministic sequence, the convergence rate of the stochastic sequence $\{\theta_k\}$ is determined by the condition number of $\hat{\theta}$. To see this, take $0 < \ell_{\star\star} < \psi_{\star\star} < \infty$, such that $\ell_{\star\star} < \lambda_{\mathsf{min}}(\hat{\theta})$, and $\lambda_{\mathsf{max}}(\hat{\theta}) < \psi_{\star\star}$. It is easy to show that a conditional version of (16) holds almost surely: for $0 \leq k_0 \leq k$, and for $\tau^{k_0}(\ell, \psi) \overset{\text{def}}{=} \inf\{k \geq k_0 : \theta_k \notin \mathcal{M}_+(\ell, \psi)\}$,

$$\mathbf{1}_{\{\theta_{k_0} \in \mathcal{M}_+(\epsilon_{\star\star}, \psi_{\star\star})\}} \mathbb{E}\left[\mathbf{1}_{\{\tau^{k_0}(\ell_{\star\star}, \psi_{\star\star}) > k\}} \left\|\theta_k - \hat{\theta}\right\|_{\mathsf{F}}^2 |\mathcal{F}_{k_0}\right] \leq \left(1 - \frac{\gamma}{\psi_{\star\star}^2}\right)^{k-k_0} \left\|\theta_{k_0} - \hat{\theta}\right\|_{\mathsf{F}}^2 + 2\gamma^2 \ell_{\star\star}^{-2}(p + p^2) \sum_{j=k_0+1}^{k} N_j^{-1} \left(1 - \frac{\gamma}{\psi_{\star\star}^2}\right)^{k-k_0-j}. \quad (17)$$

Therefore, as $\theta_k \to \hat{\theta}$ almost surely, and since $\hat{\theta} \in \mathcal{M}_+(\epsilon_{\star\star}, \psi_{\star\star})$, one can find k_0 such that with high probability, and for all $k \geq k_0$, the following holds:

$$\mathbf{1}_{\{\theta_{k_0} \in \mathcal{M}_+(\epsilon_{\star\star}, \psi_{\star\star})\}} \mathbf{1}_{\{\tau^{k_0}(\ell_{\star\star}, \psi_{\star\star}) > k\}} = 1. \quad (18)$$

If we make the (strong) assumption that (18) holds with probability one, then one can deduce from (17) that for $k \geq k_0$,

$$\mathbb{E}\left[\left\|\theta_k - \hat{\theta}\right\|_{\mathsf{F}}^2\right] \leq \left(1 - \frac{\gamma}{\psi_{\star\star}^2}\right)^{k-k_0} \mathbb{E}\left[\left\|\theta_{k_0} - \hat{\theta}\right\|_{\mathsf{F}}^2\right] + 2\gamma^2 \ell_{\star\star}^{-2}(p + p^2) \sum_{j=k_0+1}^{k} N_j^{-1} \left(1 - \frac{\gamma}{\psi_{\star\star}^2}\right)^{k-k_0-j}, \quad (19)$$

which is an analogue of Corollary 5. As in the deterministic case, this adaptive behavior complicates the complexity analysis of the algorithm. In the discussion below, we consider the idealized case where $\ell_{\star\star} = \lambda_{\mathsf{min}}(\hat{\theta})$, $\psi_{\star\star} = \lambda_{\mathsf{max}}(\hat{\theta})$, and $\theta_0 \in \mathcal{M}_+(\ell_{\star\star}, \psi_{\star\star})$.

Implications of Theorem 7 and choice of N_j

We now look at some of the implications of Theorem 7 and (19) in the ideal setting where $k_0 = 0$. If N_j is allowed to increase as $N_j = \lceil N + j^q \rceil$ for some $q > 0$, then $\sum_{j=1}^{k} \left(1 - \frac{\gamma}{\psi^2}\right)^{k-j} \frac{1}{N_j} \sim \frac{\psi^2}{\gamma} \frac{1}{N_k}$, as $k \to \infty$; then the implication of Theorem 7 and (19) is that, as $k \to \infty$,

$$\mathbb{E}\left[\left\|\theta_k - \hat{\theta}\right\|_{\mathsf{F}}^2\right] = O\left(\left(1 - \frac{\gamma}{\psi_{\star\star}^2}\right)^k + \frac{\psi_{\star\star}^2}{\gamma N_k}\right) = O\left(\rho^k + \frac{\psi_{\star\star}^2}{\gamma k^q}\right), \quad (20)$$

with $\rho = 1 - \frac{\gamma}{\psi_{\star\star}^2}$. Notice that the best choice of the step-size is $\gamma = \ell_{\star\star}^2$. Setting $\gamma = \ell_{\star\star}^2$, it follows that the number of iterations to guarantee that the left-hand side of (20) is smaller than $\delta \in (0, 1)$ is

$$k_\star = \left(\frac{\psi_{\star\star}^2}{\ell_{\star\star}^2} \frac{1}{\delta}\right)^{\frac{1}{q}} \vee \frac{\log(\delta^{-1})}{\log(\rho^{-1})}, \quad (21)$$

where $\rho = 1 - \frac{\ell_{\star\star}^2}{\psi_{\star\star}^2}$, and $a \vee b = \max(a, b)$. This implies that in choosing $N_j = \lceil N + j^q \rceil$, one should choose $q > 0$ such that

$$\left(\frac{\psi_{\star\star}^2}{\ell_{\star\star}^2} \frac{1}{\delta}\right)^{\frac{1}{q}} = \frac{\log(\delta^{-1})}{\log(\rho^{-1})} = O\left(\frac{\psi_{\star\star}^2}{\ell_{\star\star}^2} \log(\delta^{-1})\right) = O\left(\mathsf{cond}(\hat{\theta})^2 \log(\delta^{-1})\right), \quad (22)$$

where $\mathsf{cond}(\hat{\theta}) = \lambda_{\mathsf{max}}(\hat{\theta})/\lambda_{\mathsf{min}}(\hat{\theta})$ is the condition number of $\hat{\theta}$. Incidentally, (22) shows that one should choose $q > 1$, as also needed in Theorem 6.

The results developed above suggest the following stochastic version of Algorithm 1. As above, let $\{\gamma_k,\ k \geq 0\}$ be a sequence of positive step-sizes decreasing to zero, and let $\{N_k,\ k \geq 0\}$ be a sequence of Monte Carlo sample sizes. That is, N_k is the number of Monte Carlo sample draws from π_{θ_k} at iteration k. Algorithm 2 is summarized below:

Algorithm 2
Set $\mathsf{r} = 0$.

1. *Choose* $\theta_0 \in \mathcal{M}_+$.
2. *Given* θ_k, *generate* $z_{1:N_k} \overset{i.i.d.}{\sim} \pi_{\theta_k}$, *i.e., the density of* $\boldsymbol{N}(0, \theta_k^{-1})$, *and set*
$$\Sigma_{k+1} = \frac{1}{N_k} \sum_{j=1}^{N_k} z_j z_j'.$$
3. *Compute*
$$\theta_{k+1} = \mathrm{Prox}_{\gamma_{\mathsf{r}}}\left(\theta_k - \gamma_{\mathsf{r}}(S - \Sigma_{k+1}); \alpha\right).$$
4. *If* $\lambda_{min}(\theta_{k+1}) \leq 0$, *then restart: set* $k \leftarrow 0$, $\mathsf{r} \leftarrow \mathsf{r} + 1$, *and go back to (1). Otherwise, set* $k \leftarrow k+1$ *and go to (2).*

Remark 8 *As in Algorithm 1, the actual implementation of Step 4 can be avoided. For instance if the simulation of the Gaussian random variables in Step 2 uses the Cholesky decomposition, it* returns *the information whether* $\lambda_{min}(\theta_{k+1}) \leq 0$. *In this case, we restart the algorithm from* θ_0 *(or from* θ_k*), and with a smaller step-size, and a larger Monte Carlo batch size.* □

4.1 Sampling via dense Cholesky decomposition

The main computational cost of Algorithm 2 lies in generating multivariate Gaussian random variables. The standard scheme for simulating such random variables is to decompose the precision matrix θ as
$$\theta = R'R, \tag{23}$$
for some nonsingular matrix $R \in \mathbb{R}^{p \times p}$. Then a random sample from $\mathbf{N}(0, \theta^{-1})$ is obtained by simulating $u \sim \mathbf{N}(0, I_p)$ and returning $R^{-1}u$. The most common but remarkably effective approach to achieve the above decomposition (23) is via the Cholesky decomposition, which leads to R being triangular. This approach entails a total cost of $O(p^2 m + p^3/3)$ to generate a set of m independent Gaussian random variables and computing the outer-product matrix, which forms an approximation to θ^{-1}. The term $p^3/3$ accounts for the cost of the Cholesky decomposition; and $p^2 m$ accounts for doing m back-solves $R^{-1}u_i$ for m many standard Gaussian random vectors $u_i, i = 1, \ldots, m$; and subsequently computing $\frac{1}{m}\sum_{i=1}^{m}(R^{-1}u_i)(R^{-1}u_i)'$ — note that each back-solve $R^{-1}u_i$ can be performed with $O(p^2)$ cost since R is triangular. This shows that an iteration of Algorithm 2, implemented via Cholesky decomposition, is more cost-effective than an iteration of Algorithm 1, if the number of Gaussian random

samples generated in that iteration is less than p. Since $k_\star$ iterations (as defined in (21)) are needed to reach the precision δ, and $N_k = N + k^q$ (we assume that q is chosen as in (22)), we see that the number of samples per iteration of Algorithm 2 remains below p, if $p \geq \mathsf{cond}(\hat{\theta})^2\delta^{-1}$. In this case the overall computational cost of Algorithm 2, to obtain a δ-accurate solution is

$$O\left(p^3 \frac{\log(\delta^{-1})}{\log(\rho^{-1})}\right) = O\left(p^3 \mathsf{cond}(\hat{\theta})^2 \log(\delta^{-1})\right).$$

We caution the reader that, on the surface, the above cost seems to be of the same order as that of the deterministic algorithm (Algorithm 1), as seen from Theorem 3. However, the constants in the big-O notation differ, and are much better for the Cholesky decomposition than for inverting a matrix—see Figure 1 for a compelling illustration of this observation. In addition, as the problem sizes become much larger (i.e., larger than $p \approx 35,000$) matrix inversions become much more memory intensive than Cholesky decompositions; leading to prohibitely increased computation times—see Figure 1.

4.2 Sampling via specialized sparse numerical linear algebra methods

As an alternative to the above approach, note that equation (23) is also solved by $R = \theta^{1/2}$. If θ is sparse and very large, specialized numerical linear algebra methods can be used to compute $\theta^{-1/2}b$ for a vector or matrix b, with matching dimensions. These methods include Krylov space methods (Hale et al. (2008); Eiermann and Ernst (2006)), or matrix function approximation methods (Chen et al. (2011)). These methods heavily exploit sparsity and typically scale better than the Cholesky decomposition when dealing with very large sparse problems. For instance, the matrix function approximation method of Chen et al. (2011) has a computational cost of $O(m(p + C_p))$ to generate a set of m samples from $\mathbf{N}(0, \theta^{-1})$, where C_p is the cost of performing a matrix-vector product θb for some $b \in \mathbb{R}^p$. As comparison, Figure 2 shows the time for generating $1,000$ random samples from $\mathbf{N}(0, \theta^{-1})$, using dense Cholesky factorization, and using the matrix function approximation approach of (Chen et al. (2011)), for varying values of p. The value of p around which the matrix approximation method becomes better than the Cholesky decomposition depends on the sparsity of θ, and the implementations of the methods. These specialized sparse methods, however, need to be used with caution. For one thing, these methods are quite sensitive to the sparsity level of the iterates θ_k, and ultimately to the sparsity level $\hat{\theta}$, the solution to Problem (3) — the methods are useful only when the solutions are sufficiently sparse. This behavior should be contrasted to that of dense Cholesky decomposition based methods, which are less sensitive to the sparsity level of $\hat{\theta}$. Based on our experiments (not reported here), we recommend the use of dense Cholesky decomposition methods in the initial stages of the algorithm, when the iterates θ_k are relatively dense. As the number of iterations progresses and the estimates become more sparse, we recommend the use of specialized sparse numerical linear algebra methods for sampling from the Gaussian distributions. Since the use of dense Cholesky decomposition methods amply substantiates the main message of our paper—the effectiveness of stochastic gradient methods as a computationally scalable alternative to their deterministic counterparts, our experimental results reported in Section 7 focus on dense numerical linear algebra methods.

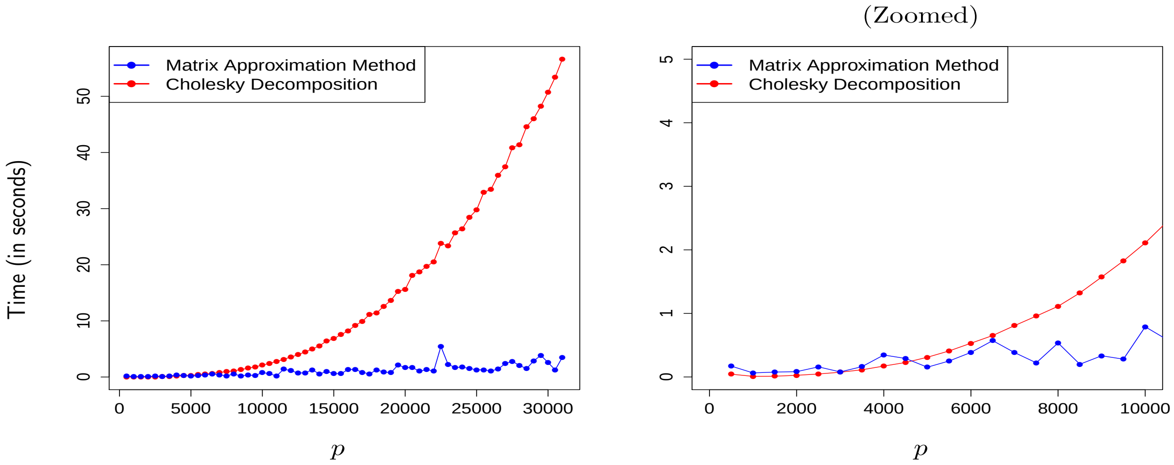

Figure 2: Figure showing the times in seconds to generate 1,000 Gaussian random samples from $\mathbf{N}(0, \theta^{-1})$, where $\theta \in \mathbb{R}^{p\times p}$ is constructed as explained in Section 7.1.1 with the proportion of non-zeros entries approximately set at $5/p$.

4.3 Borrowing information across iterations

A main limitation of Algorithm 2 is that at each iteration k, all the Monte Carlo samples used to estimate θ_k^{-1} are discarded, and new samples are generated to approximate θ_{k+1}^{-1}. We thus ask, is there a modified algorithm that makes clever use of the information associated with an approximate θ_k^{-1} to approximate θ_{k+1}^{-1}? In this vein, we propose herein a new stochastic algorithm: Algorithm 3 which recycles previously generated Monte Carlo samples in a novel fashion, to update its approximation for $\Sigma_{k+1} := \theta_{k+1}^{-1}$ from $\Sigma_k := \theta_k^{-1}$.

This new algorithm relies on the following algorithm parameters (a) N, where $N \geq 1$ is a given integer, and (b) $\{\zeta_k,\ k \geq 1\}$ which is a sequence of positive numbers such that

$$\sum_{k\geq 1} \zeta_k = \infty, \quad \text{and} \quad \sum_{k\geq 1} \zeta_k^2 < \infty. \tag{24}$$

The algorithm is summarized below:

Algorithm 3 *Set* $r = 0$.

1. *Choose* $\theta_0 \in \mathcal{M}_+$, *and* $\Sigma_0 \in \mathcal{M}_+$.

2. *Given* θ_k, *and* Σ_k, *generate* $z_{1:N} \overset{i.i.d.}{\sim} \pi_{\theta_k} = \boldsymbol{N}(0, \theta_k^{-1})$, *and compute*

$$\Sigma_{k+1} = \Sigma_k + \zeta_{k+1}\left(\frac{1}{N}\sum_{k=1}^{N} z_k z_k' - \Sigma_k\right). \tag{25}$$

3. *Compute*

$$\theta_{k+1} = \operatorname{Prox}_{\gamma_r}\left(\theta_k - \gamma_r(S - \Sigma_{k+1}); \alpha\right). \tag{26}$$

4. *If* $\lambda_{min}(\theta_{k+1}) \leq 0$, *then restart: set* $k \leftarrow 0$, $r \leftarrow r + 1$, *and go back to (1). Otherwise, set* $k \leftarrow k + 1$ *and go to (2).*

Notice that in Algorithm 3, the number of Monte Carlo samples is held fixed at N. Hence its cost per iteration is constant.

Algorithm 3 is more difficult to analyze because the two recursive equations (25) and (26) are intimately coupled. However, the next result gives some theoretical guarantees by showing that when the sequence $\{\theta_k,\ k \geq 0\}$ converges, it necessarily converges to the minimizer of Problem (3), i.e., $\hat{\theta}$.

Theorem 9 *Let $\{\theta_k,\ k \geq 0\}$ be the stochastic process generated by Algorithm 3 where, the sequence $\{\zeta_k\}$ satisfies (24). Fix $0 < \ell \leq \psi < \infty$. Suppose that $\hat{\theta}, \theta_0 \in \mathcal{M}_+(\ell, \psi)$, and $\gamma \leq \ell^2$. Then, on the event*

$$\{\tau(\ell,\psi) = +\infty, \quad \textit{and} \quad \{\theta_k\} \textit{ converges}\},$$

we have that $\lim_{k\to\infty} \theta_k = \hat{\theta}$.

Proof See Section 8.6. ■

5. Exact Thresholding into connected components

As mentioned in Section 1, the exact covariance thresholding rule (Mazumder and Hastie, 2012a), originally developed for the Glasso problem plays a crucial role in the scalability of Glasso to large values of p, for large values of λ. One simply requires that the largest connected component of the graph $((\mathbf{1}(|s_{ij}| > \lambda)))$, is of a size that can be handled by an algorithm for solving Glasso of that size. In this section, we extend this result to the more general case of Problem (3).

Consider the symmetric binary matrix $\mathcal{E} := ((\mathcal{E}_{ij}))$ with $\mathcal{E}_{ij} = \mathbf{1}(|s_{ij}| > \alpha\lambda)$, which defines a graph on the nodes $\mathcal{V} = \{1, \ldots, p\}$. Let $(\mathcal{V}_j, \mathcal{E}_j), j = 1, \ldots, J$ denote the J connected components of the graph $(\mathcal{V}, \mathcal{E})$. Let $\hat{\theta}$ be a minimizer of Problem (3) and consider the graph $\widehat{\mathcal{E}}$ induced by the sparsity pattern of $\hat{\theta}$, namely, $\widehat{\mathcal{E}}_{ij} = \mathbf{1}(|\hat{\theta}_{ij}| \neq 0)$. Let the connected components of $(\mathcal{V}, \widehat{\mathcal{E}})$ be denoted by $(\widehat{\mathcal{V}}_j, \widehat{\mathcal{E}}_j), j = 1, \ldots, \widehat{J}$. The following theorem states that these connected components are essentially the same.

Theorem 10 *Let $(\mathcal{V}_j, \mathcal{E}_j), j = 1, \ldots, J$ and $(\widehat{\mathcal{V}}_j, \widehat{\mathcal{E}}_j), j = 1, \ldots, \widehat{J}$ denote the connected components, as defined above.*

Then, $J = \widehat{J}$ and there exists a permutation Π on $\{1, \ldots, J\}$ such that $\widehat{\mathcal{V}}_{\Pi(j)} = \mathcal{V}_j$ and $\widehat{\mathcal{E}}_{\Pi(j)} = \mathcal{E}_j$ for all $j = 1, \ldots, J$.

Proof *See Appendix, Section 8.7 for the proof.* ■

Note that the permutation Π arises since the labelings of two connected component decompositions may be different.

Theorem 10 is appealing because the connected components of the graph $\mathcal{E}_{ij} = \mathbf{1}(|s_{ij}| > \alpha\lambda)$ are fairly easy to compute even for massive sized graphs—see also Mazumder and Hastie (2012a) for additional discussions pertaining to similar observations for the Glasso problem. A simple but powerful consequence of Theorem 10 is that, once the connected components

$(\mathcal{V}_j, \mathcal{E}_j), j = 1, \ldots, J$ are obtained, Problem (3) can be solved independently for each of the J different connected component blocks. In concluding, we note that Theorem 10 is useful if the maximum size of the connected components is small compared to p, which of course depends upon S and λ, α.

6. Special Case: Ridge regularization

In this section, we focus our attention to a special instance of Problem (3), namely, the ridge regularized version, i.e., Problem (2) for some value of $\lambda > 0$. Interestingly, the solution to this problem can be computed analytically as presented in the following lemma:

Lemma 11 *Let $S = UDU'$ denote the full eigendecomposition of S where, $D = \mathrm{diag}(d_1, \ldots, d_p)$. For any $\lambda > 0$ and $\alpha = 0$ the solution to Problem (3) is given by: $\hat{\theta} = U\mathrm{diag}(\widehat{\sigma})U'$, where, $\mathrm{diag}(\widehat{\sigma})$ is a diagonal matrix with the ith diagonal entry given by*

$$\widehat{\sigma}_i = \frac{-d_i + \sqrt{d_i^2 + 4\lambda}}{2\lambda}, \quad \textit{for} \quad i = 1, \ldots, p.$$

Proof *For the proof see Section 8.8* ■

We make the following remarks:

- Performing the eigen-decomposition of S is clearly the most expensive part in computing a solution to Problem (2); for a general real $p \times p$ symmetric matrix this has cost $O(p^3)$ and can be significantly more expensive than computing a direct matrix inverse or a Cholesky decomposition, as reflected in Figure 1.

- When $p \gg n$ and n is small, a minimizer for Problem (2) can be computed for large p, by observing that $S = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i\mathbf{x}_i' = \frac{1}{n}X'X$; thus the eigendecomposition of S can be done efficiently via a SVD of the $n \times p$ rectangular matrix X with $O(n^2p)$ cost, which reduces to $O(p)$ for values of $p \gg n$ with n small.

- However, computing the solution to Problem (2) becomes quite difficult when both p and n are large. In this case, both our stochastic algorithms: Algorithms 2 and 3 are seen to be very useful to get an approximate solution within a fraction of the total computation time. Section 7.2 presents some numerical experiments.

7. Numerical experiments

We performed some experiments to demonstrate the practical merit of our algorithm on some synthetic and real datasets.

Software Specifications

All our computations were performed in Matlab (R2014a (8.3.0.532) 64-bit (maci64)) on a OS X 10.8.5 (12F45) operating system with a 3.4 GHz Intel Core i5 processor with 32 GB Ram, processor speed 1600 MHz and DDR3 SDRAM.

7.1 Studying sparse problems

7.1.1 Simulated data

We test Algorithms 1, 2 and 3 with $p = 10^3, 5 \times 10^3$, and $p = 10^4$ for some synthetic examples. The data matrix $S \in \mathbb{R}^{p\times p}$ is generated as $S = n^{-1}\sum_{j=1}^{n} \mathbf{x}_j\mathbf{x}_j'$, where $n = p/2$, and $X_{1:n} \overset{\text{i.i.d.}}{\sim} \mathbf{N}_p(0, \theta_\star^{-1})$, for a "true" precision matrix $\theta_\star$ generated as follows. First we generate a symmetric sparse matrix B such that the proportion of non-zeros entries is $10/p$. We magnified the signal by adding 4 to all the non-zeros entries of B (subtracting 4 for negative non-zero entries). Then we set $\theta_\star = B + (\ell - \lambda_{\min}(B))I_p$, where $\lambda_{\min}(B)$ is the smallest eigenvalue of B, with $\ell = 1$.

Given S, we solve Problem (3) with $\alpha \approx 0.9$ and $\lambda \propto \sqrt{\log(p)/n}$ such that the sparsity (i.e., the number of non-zeros) of the solution is roughly $10/p$. In all the examples, we ran the deterministic algorithm (Algorithm 1) for a large number of iterations (one thousand) with a step-size $\gamma = 3.5$ to obtain a high-accuracy approximation of $\hat{\theta}$, the solution to Problem (3) (we take this estimate as $\hat{\theta}$ in what follows). Algorithms 1, 2 and 3 were then evaluated as how they progress towards the optimal solution $\hat{\theta}$ (recall that the optimization problem has a unique minimizer), as a function of time. All the algorithms were ran for a maximum of 300 iterations. Further details in setting up the solvers and parameter specifications are gathered in Section B (appendix). To measure the quality of the solution, we used the following metric:

$$\text{Relative Error} = \|\theta_k - \hat{\theta}\|_F / \|\hat{\theta}\|_F,$$

as a function of the number of iterations of the algorithms. Since the work done per iteration by the different algorithms are different, we monitored the progress of the algorithms as a function of time. The results are shown in Figure 3.

We also compared the performance of our algorithms with the exact thresholding scheme (Section 5) switched "on" — this offered marginal improvements since the size of the largest component was comparable to the size of the original matrix — see Section B for additional details on the sizes of the connected components produced. We also compared our method with a state-of-the algorithm: Quic (Hsieh et al., 2014), the only method that seemed to scale to *all* the problem sizes that have been considered in our computational experiments. We used the `R` package `QUIC`, downloaded from `CRAN` for our experiments. The results are shown in Table 1.

We note that it is not fair to compare our methods versus Quic due to several reasons. Firstly the available implementation of Quic works for the Glasso problem and the experiments we consider are for the generalized elastic net problem (3). Furthermore, Quic is a fairly advanced implementation written in `C++`, whereas our method is implemented entirely in Matlab. In addition, the default convergence criterion used by Quic is different than what we use. However, we do report the computational times of Quic simply to give an idea of where we are in terms of the state-of-the art algorithms for Glasso. Towards this end, we ran Quic for the Glasso problem with $\lambda = \alpha\lambda$ for a large tolerance parameter (we took the native tolerance parameter, based on relative errors used in the algorithm Quic by setting its convergence threshold (tol) as 10^{-10}), the solution thus obtained was denoted by $\hat{\theta}$. We ran Quic for a sequence of twenty tolerance values of the form 0.5×0.9^r for $r = 1, \ldots, 20$; and then obtained the solution for which the relative error $\|\theta_\text{r} - \hat{\theta}\|_F / \|\hat{\theta}\|_F \leq$ Tol with

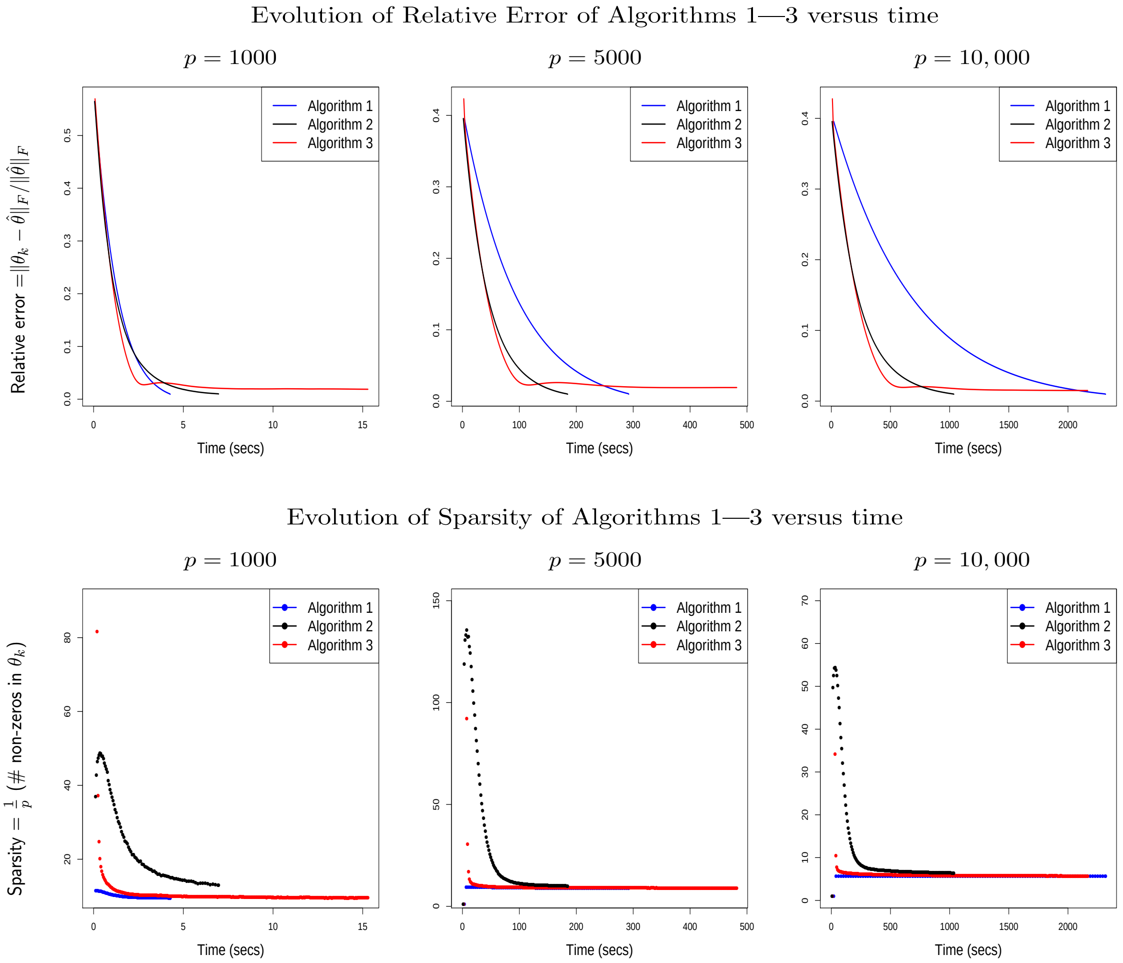

Figure 3: Evolution of relative error [top panel] and sparsity [bottom panel] of Algorithms 1-3 versus time (in secs); for three different problem sizes: $p \in \{10^3, 5 \times 10^3, 10^4\}$ for the examples described in Section 7.1.1. We observe that for larger values of $p \geq 5 \times 10^3$, the new stochastic algorithms proposed in this paper: Algorithms 2 and 3 reach moderate accuracy solutions in times significantly smaller than the deterministic counterpart: Algorithm 1. Algorithm 3 reaches a low accuracy solution quicker, but is dominated by Algorithm 2 in obtaining a solution with higher accuracy. For small values of p ($p = 1000$) the different algorithms are comparable because direct matrix inversions are computationally less expensive, the situation changes quickly however, with larger values of p (See also Figure 1).

Tol $\in \{0.1, 0.02\}$. For reference, the times taken by Quic to converge to its "default" convergence threshold (given by its relative error convergence threshold: tol$= 10^{-4}$) were 501 seconds for for $p = 5,000$ and 3020 seconds for $p = 10,000$.

7.1.2 Real dataset

The Patrick Brown dataset is an early example of an expression array, obtained from the Patrick Brown Laboratory at Stanford University and was studied in Mazumder and Hastie

(2012a). There are $n = 385$ patient samples of tissues from various regions of the body (some from tumors, some not), with gene-expression measurements for $p = 4718$ genes. For this example, the values of the regularization parameters were taken as $(\alpha, \lambda) = (0.99, 0.16)$. Here, splitting led to minor improvements since the size of the largest component was 4709, with all others having size one. We report the performance of our methods *without* using the splitting method. We computed $\hat{\theta}$ by running the deterministic algorithm for 1000 iterations, using a step-size $\gamma = 5 \times 10^{-5}$. Unlike the synthetic experiments, in this case, we considered relative changes in objective values to determine the progress of the algorithm, namely, $(\phi_\alpha(\theta_k) - \hat{\phi}_\alpha)/|\hat{\phi}_\alpha|$, where, we define $\hat{\phi}_\alpha = \phi_\alpha(\hat{\theta})$; and recall that $\phi_\alpha(\cdot)$ is defined in (5).

In this case, we also compared our method with Quic but the latter took a very long time in converging to even a moderate accuracy solution, so we took the solution delivered by its default mode as the reference solution $\hat{\theta}$. Quic took 5.9 hours to produce its default solution. Taking the objective value of this problem as the reference, we found that Quic took 6080.207 and 10799.769 secs to reach solutions with relative error 0.74 and 0.50 respectively. We summarize the results in Table 2.

Our empirical findings confirm the theoretical results that for large p, the stochastic algorithms reach low-accuracy solutions much faster than the deterministic algorithms. We also see that the splitting rule helps, as it should — major improvements are expected if the sizes of the connected components are significantly smaller than the original problem. The sparsity plot (Figure 3) shows that the solution provided by Algorithm 2 tends to be noisy. The averaging step in estimating θ_k^{-1} in Algorithm 3 makes these estimates much smoother, which results in solutions with good sparsity properties.

7.2 Studying dense problems

We performed some experiments to demonstrate the performance of our method on dense inverse covariance estimation problems. Here, we took a sample of size $n = p$ with $p \in \{10^4, 1.5 \times 10^4\}$, from a Gaussian density with independent covariates and mean zero. As described in Section 6, it is indeed possible to obtain a closed form solution to this problem, but it requires performing a large scale eigen-decomposition on S, which can be quite expensive. In this application, proximal gradient algorithms and in particular the stochastic algorithms presented in this paper, become particularly useful. They deliver approximate solutions to Problem 2 in times that are orders of magnitude smaller than that taken to obtain an exact solution.

In the experiments considered herein, we found the following scheme to be quite useful. We took a subsample of size $m \ll n$ from the original n samples and solved Problem (2) with a covariance matrix obtained from that subsample. This is indeed quite efficient since it requires computing the SVD of an $m \times p$ matrix, with $m \ll p$. We took the precision matrix and the covariance matrix associated with this subsample as a warm-start to the deterministic proximal gradient method, i.e., Algorithm 1 and Algorithm 2. This was seen to improve the overall run-time of the solution versus an initialization with a diagonal matrix.

We summarize our results in Table 3. For the case $p = 10,000$ our Monte Carlo batch size was of $N_k = 1,000 + \lceil k^{1.4} \rceil$ and we took $\gamma = 0.1/\lambda^2_{\max}(S)$. The algorithms were warm-started with the solution of Problem (2) for a subsample of size $m = 100$, which took 0.1 seconds to compute. For the case, $p = 15,000$ we took $N_k = 2,000 + \lceil k^{1.4} \rceil$ and γ as before. As a

	Time (in secs) taken by algorithms						
Accuracy	**Algorithm 1**		**Algorithm 2**		**Algorithm 3**		Quic
Tol	No Splitting	With Splitting	No Splitting	With Splitting	No Splitting	With Splitting	
			$p = 5000$				
10^{-1}	125.78	122.16	62.73	60.84	56.78	53.31	300.94
2×10^{-2}	251.92	241.49	161.18	142.83	292.34	271.67	350.29
			$p = 10,000$				
10^{-1}	921.52	612.11	317.35	155.33	289.58	192.28	2046.73
2×10^{-2}	1914.65	1305.65	766.69	463.66	647.27	563.42	2373.03

Table 1: Table showing the times (in secs) to reach an Accuracy of "Tol" for different algorithms, where, Accuracy refers to $\|\theta_k - \hat{\theta}\|_F / \|\hat{\theta}\|_F$. Algorithms 2 and 3 clearly shine over the deterministic method (Algorithm 1) for delivering moderate accuracy solutions. Algorithm 3 reaches a solution of moderate accuracy faster than Algorithm 2 and Algorithm 1; for smaller values of "Tol" Algorithm 2 wins. Here, splitting, which refers to the notion of covariance thresholding described in Section 5 is found to help, though not substantially — the regularization parameters in this problem lead to connected components of sizes comparable to the original problem. The timings of Quic are shown for reference purposes only, to get an idea of the times taken by state-of-the art algorithms. For reference, the times taken by Quic to converge to its default convergence criteria were 501 seconds for $p = 5,000$ and 3020 seconds for $p = 10,000$.

warm-start we took the solution of Problem (2) with a subsample of size $m = 500$ which was obtained in 1 second.

Accuracy	Time (in secs) taken by algorithms		
Tol	**Algorithm 1**	**Algorithm 2**	**Algorithm 3**
0.1	881.995	366.864	451.337
0.02	2030.405	942.924	> 654

Table 2: Results on the Patrick Brown microarray dataset (here, $n = 385$ and $p = 4718$). Algorithm 3 reached a solution of relative accuracy 0.06 within the first 500 iterations which took a total time of 654 seconds. Here, we use "Accuracy" to denote the relative error: $(\phi_\alpha(\theta_k) - \hat{\phi}_\alpha)/|\hat{\phi}_\alpha|$, where, $\hat{\phi}_\alpha$ is the optimal objective value for the problem. For comparison, Quic for the same dataset when set to optimize the corresponding graphical lasso problem with the same tuning parameter, took 5.9 hours to converge to a solution with the native (default) tolerance criterion. Taking the objective value of this problem as the reference, we found that Quic took approximately, 6080 secs ($\sim$ 1.7 hours) and 10800 secs ($\sim$ 3 hours) to reach solutions with relative errors 0.74 and 0.50 respectively. The algorithms presented in this paper show impressive performance for the particular tasks at hand.

8. Proofs

This section gathers the proofs and technical details appearing in the paper.

Accuracy		Time (in secs) taken by algorithms	
Tol	p	**Algorithm 1**	**Algorithm 2**
0.1	10^4	15.42	4.67
0.05	10^4	93.46	48.490
0.1	1.5×10^4	50.78	15.70
0.05	1.5×10^4	408.87	176.11

Table 3: Results for ridge regression. Here, we use "Accuracy" to denote the measure: $(\phi_\alpha(\theta_k) - \hat{\phi}_\alpha)/|\hat{\phi}_\alpha|$, where, $\hat{\phi}_\alpha$ is the optimal objective value for the problem. For $p = 10,000$ computing the exact solution (using a full eigen-decomposition) took 140 secs, for $p = 15,000$ the exact solution was computed in 500 secs. Both Algorithms 1 and 2 obtained approximate solutions in times significantly smaller than computing the exact solution to the problem. For details see Section 7.2.

8.1 Proof of Lemma 1

Proof

Uniqueness of $\hat{\theta}$:

If $\lambda_2 > 0$ then Problem (3) is strongly convex due to the presence of the quadratic regularizer, hence $\hat{\theta}$ is unique. If $\lambda_2 = 0$ and $\lambda_1 > 0$ then Problem (3) becomes equivalent to Glasso for which uniqueness of $\hat{\theta}$ was established in Banerjee et al. (2008); Lu (2009).

Spectral bounds on $\hat{\theta}$:

Consider the stationary conditions of Problem (3):

$$-\hat{\theta}^{-1} + S + \lambda_1 Z + 2\lambda_2\hat{\theta} = 0, \tag{27}$$

where, we use the notation: $Z = \text{sgn}(\hat{\theta})$, $\lambda_1 = \alpha\lambda$ and $\lambda_2 = (1-\alpha)\lambda/2$. It follows from (27) that

$$\begin{aligned}\hat{\theta}^{-1} - 2\lambda_2\hat{\theta} =& S + \lambda_1 Z \\ \leq& \|S + \lambda_1 Z\|_2 \mathbf{I} \\ \leq& (\|S\|_2 + \lambda_1\|Z\|_2)\,\mathbf{I} \\ \leq& (\|S\|_2 + \lambda_1 p)\,\mathbf{I} \quad (\text{since, } z_{ij} \in [-1,1] \text{ implies } \|Z\|_2 \leq p)\end{aligned} \tag{28}$$

If σ_i's denote the eigen-values of $\hat{\theta}$ then it follows from (28):

$$1/\sigma_i - 2\lambda_2\sigma_i \leq \|S\|_2 + \lambda_1 p = \mu.$$

Using elementary algebra, the above provides us a lower bound on all the eigen-values of the optimal solution $\hat{\theta}$: $\sigma_i \geq (-\mu + \sqrt{\mu^2 + 8\lambda_2})/(4\lambda_2)$ for $\lambda_2 \neq 0$, for all $i = 1, \ldots, p$. Note that for the case, $\lambda_2 = 0$ we have $\sigma_i \geq 1/\mu$ for all i. Combining these results we have the following:

$$\lambda_{\min}(\hat{\theta}) \geq \ell_\star := \begin{cases} \frac{-\mu + \sqrt{\mu^2 + 8\lambda_2}}{4\lambda_2} & \text{if } \lambda_2 \neq 0 \\ \frac{1}{\mu} & \text{otherwise,} \end{cases}$$

which completes the proof of the lower bound on the spectrum of $\hat{\theta}$.

We now proceed towards deriving upper bound on the eigen-values of $\hat{\theta}$.

From (27) we have:

$$0 = \langle \hat{\theta}, -\hat{\theta}^{-1} + S + \lambda_1 Z + 2\lambda_2 \hat{\theta} \rangle \implies \lambda_1 \|\hat{\theta}\|_1 = p - \langle \hat{\theta}, S \rangle - 2\lambda_2 \left\|\hat{\theta}\right\|_{\mathsf{F}}^2 \tag{29}$$

Now observe that:

$$\langle \hat{\theta}, S \rangle \geq \lambda_{\min}(\hat{\theta}) \mathrm{T}r(S) \quad \text{and} \quad \left\|\hat{\theta}\right\|_{\mathsf{F}}^2 \geq p\lambda_{\min}^2(\hat{\theta}). \tag{30}$$

We use $\ell_\star$ as a lower bound for $\lambda_{\min}(\hat{\theta})$ and use (30) in (29) to arrive at:

$$\|\hat{\theta}\|_1 \leq \frac{1}{\lambda_1}\left(p - \ell_\star \mathrm{T}r(S) - 2p\lambda_2 \ell_\star^2\right) := U_1 \tag{31}$$

The above bound can be tightened by adapting the techniques appearing in Lu (2009) for the special case $\lambda_2 = 0$; as we discuss below. Let $\hat{\theta}(t) := (S + t\lambda_1 \mathbf{I})^{-1}$ be a family of matrices defined on $t \in (0, 1)$. It is easy to see that

$$\hat{\theta}(t) \in \underset{\theta}{\mathsf{Argmin}} \left\{-\log\det(\theta) + \langle S + t\lambda_1 \mathbf{I}, \theta \rangle\right\},$$

which leads to

$$\begin{aligned} -\log\det(\hat{\theta}(t)) + \langle S + t\lambda_1 \mathbf{I}, \hat{\theta}(t) \rangle \leq & \ -\log\det(\hat{\theta}) + \langle S + t\lambda_1 \mathbf{I}, \hat{\theta} \rangle \\ -\log\det(\hat{\theta}) + \langle S, \hat{\theta} \rangle + \lambda_1 \|\hat{\theta}\|_1 + \lambda_2 \left\|\hat{\theta}\right\|_{\mathsf{F}}^2 \leq & \ -\log\det(\hat{\theta}(t)) + \langle S, \hat{\theta}(t) \rangle \\ & +\lambda_1 \|\hat{\theta}(t)\|_1 + \lambda_2 \left\|\hat{\theta}(t)\right\|_{\mathsf{F}}^2, \end{aligned} \tag{32}$$

where, the second inequality in (32) follows from the definition of $\hat{\theta}$. Adding the two inequalities in (32) and doing some simplification, we have:

$$\lambda_1 \|\hat{\theta}(t)\|_1 - t\lambda_1 \mathrm{T}r(\hat{\theta}(t)) + \lambda_2 \left\|\hat{\theta}(t)\right\|_{\mathsf{F}}^2 - \lambda_2 \left\|\hat{\theta}\right\|_{\mathsf{F}}^2 \geq \lambda_1 \|\hat{\theta}\|_1 - t\lambda_1 \mathrm{T}r(\hat{\theta}) \geq (\lambda_1 - t\lambda_1)\|\hat{\theta}\|_1,$$

where, the rhs of the above inequality was obtained by using the simple observation $\mathrm{T}r(\hat{\theta}) \leq \|\hat{\theta}\|_1$. Dividing both sides of the above inequality by $\lambda_1 - t\lambda_1$ we have:

$$\|\hat{\theta}\|_1 \leq \underbrace{\frac{1}{\lambda_1(1-t)}\left(\lambda_1 \|\hat{\theta}(t)\|_1 - t\lambda_1 \mathrm{T}r(\hat{\theta}(t)) + \lambda_2 \left\|\hat{\theta}(t)\right\|_{\mathsf{F}}^2\right)}_{:=a(t)} - \underbrace{\frac{\lambda_2 \left\|\hat{\theta}\right\|_{\mathsf{F}}^2}{\lambda_1(1-t)}}_{:=b(t)}. \tag{33}$$

Observing that $\left\|\hat{\theta}\right\|_{\mathsf{F}}^2 \geq \ell_\star^2 p$ and applying it to (33) we obtain:

$$\|\hat{\theta}\|_1 \leq \left(a(t) - \tilde{b}(t)\right), \tag{34}$$

where, $a(t) = \frac{1}{\lambda_1(1-t)}\left(\lambda_1\|\hat{\theta}(t)\|_1 - t\lambda_1 \mathrm{T}r(\hat{\theta}(t)) + \lambda_2 \left\|\hat{\theta}(t)\right\|_{\mathsf{F}}^2\right)$ and $\tilde{b}(t) := \frac{\lambda_2 \ell_\star^2 p}{\lambda_1(1-t)}$.

Inequality (34) in particular implies:

$$\|\hat{\theta}\|_1 \leq \inf_{t\in(0,1)} \left(a(t) - \tilde{b}(t)\right) := U_2 \tag{35}$$

where, the minimization problem appearing above is a one dimensional optimization and can be approximated quite easily. While a closed form solution to the minimization problem in (35) may not be available, $\|\hat{\theta}\|_1$ can be (upper) bounded by specific evaluations of $a(t)-\tilde{b}(t)$ at different values of $t \in (0,1)$. In particular, note that if S is invertible then, taking $t \approx 0+$ we get:

$$\|\hat{\theta}\|_1 \leq \left(\|S^{-1}\|_1 + \frac{\lambda_2}{\lambda_1}\left\|S^{-1}\right\|_{\mathsf{F}}^2\right) - \frac{\lambda_2 \ell_{LB}^2}{\lambda_1},$$

otherwise, taking $t = \frac{1}{2}$ leads to: $\|\hat{\theta}\|_1 \leq a(\frac{1}{2}) - \tilde{b}(\frac{1}{2})$.

Combining (31) and (35), we arrive at the following bound:

$$\|\hat{\theta}\|_1 \leq \min\{U_1, U_2\} \tag{36}$$

Now observe that:

$$\lambda_{\max}(\hat{\theta}) := \|\hat{\theta}\|_2 \leq \left\|\hat{\theta}\right\|_{\mathsf{F}} \leq \|\hat{\theta}\|_1 \leq \min\{U_1, U_2\} := \psi_{UB}.$$

■

8.2 Proof of Lemma 2

Proof
First Part: Lower bound on $\lambda_{\min}(\theta_j)$

Set $\bar{\theta} = \theta - \gamma(S - \theta^{-1})$. By definition, $T_\gamma(\theta;\alpha) = \mathsf{Argmin}_{u\in\mathcal{M}}\left[g_\alpha(u) + \frac{1}{2\gamma}\left\|u - \bar{\theta}\right\|_{\mathsf{F}}^2\right]$. By the optimality condition of this optimization problem, there exists $Z \in \mathcal{M}$ in the sub-differential of the function $\theta \mapsto \|\theta\|_1$ at $T_\gamma(\theta;\alpha)$ such that $Z_{ij} \in [-1,1]$, $\langle Z, T_\gamma(\theta;\alpha)\rangle = \|T_\gamma(\theta;\alpha)\|_1$, and

$$\frac{1}{\gamma}(T_\gamma(\theta;\alpha) - \bar{\theta}) + \alpha\lambda Z + (1-\alpha)\lambda T_\gamma(\theta;\alpha) = 0. \tag{37}$$

The fact that $Z_{ij} \in [-1,1]$ implies that $\|Z\|_2 \leq \|Z\|_{\mathsf{F}} \leq p$. Hence,

$$\lambda_{\mathsf{min}}(Z) \geq -p, \quad \text{and} \quad \lambda_{\mathsf{max}}(Z) \leq p. \tag{38}$$

Using $\bar{\theta} = \theta - \gamma(S - \theta^{-1})$, we expand (37) to

$$T_\gamma(\theta;\alpha) = (1 + (1-\alpha)\lambda\gamma)^{-1}\left(\theta - \gamma(S - \theta^{-1} + \alpha\lambda Z)\right).$$

We write $\theta - \gamma(S - \theta^{-1} + \alpha\lambda Z) = \theta + \gamma\theta^{-1} - \gamma(S + \alpha\lambda Z)$. We will use the fact that for any symmetric matrices A, B, $\lambda_{\mathsf{min}}(A+B) \geq \lambda_{\mathsf{min}}(A) + \lambda_{\mathsf{min}}(B)$, $\lambda_{\mathsf{max}}(A+B) \leq \lambda_{\mathsf{max}}(A) + \lambda_{\mathsf{max}}(B)$ (see e.g. Golub and Van Loan (2013) Theorem 8.1.5). In view of (38) we have:

$$\lambda_{\mathsf{min}}\left(\theta - \gamma(S - \theta^{-1} + \alpha\lambda Z)\right) \geq \lambda_{\mathsf{min}}(\theta + \gamma\theta^{-1}) - \gamma\left(\lambda_{\mathsf{max}}(S) + \alpha\lambda p\right). \tag{39}$$

Notice that the function $x \mapsto x + \frac{\gamma}{x}$ is increasing on $[\sqrt{\gamma}, \infty)$, and by assumption $\ell_\star \geq \sqrt{\gamma}$. Therefore, if $\lambda_{\min}(\theta) \geq \ell_\star$, we use the eigen-decomposition of θ to conclude that

$$\lambda_{\mathsf{min}}\left(\theta + \gamma\theta^{-1}\right) = \lambda_{\mathsf{min}}(\theta) + \frac{\lambda}{\lambda_{\mathsf{min}}(\theta)} \geq \ell_\star + \frac{\gamma}{\ell_\star}. \tag{40}$$

Hence

$$\lambda_{\mathsf{min}}\left(T_\gamma(\theta;\alpha)\right) \geq (1 + (1-\alpha)\lambda\gamma)^{-1}\left[\ell_\star + \frac{\gamma}{\ell_\star} - \gamma(\lambda_{\mathsf{max}}(S) + \alpha\lambda p)\right] = \ell_\star, \tag{41}$$

where the last equality uses the fact that $\ell_\star$ satisfies the equation

$$(1-\alpha)\lambda\ell_\star^2 + (\lambda_{\mathsf{max}}(S) + \alpha\lambda p)\ell_\star - 1 = 0.$$

Second Part: Upper bound on $\lambda_{\max}(\theta_j)$

We will first show that if $\psi_\star^1 \leq \psi_{UB}$, then $\lambda_{\max}(\theta_j) \leq \psi_\star^1$ for all $j \geq 1$. Following arguments similar to that used to arrive at (39), we have:

$$\lambda_{\mathsf{max}}\left(\theta - \gamma(S - \theta^{-1} + \alpha\lambda Z)\right) \leq \lambda_{\mathsf{max}}(\theta + \gamma\theta^{-1}) - \gamma\left(\lambda_{\mathsf{min}}(S) - \alpha\lambda p\right). \tag{42}$$

Using $\lambda_{\max}(\theta) \leq \psi_\star^1$; and following arguments used to arrive at (40), (41) we have:

$$\lambda_{\mathsf{max}}\left(\theta + \gamma\theta^{-1}\right) = \lambda_{\mathsf{max}}(\theta) + \frac{\lambda}{\lambda_{\mathsf{max}}(\theta)} \leq \psi_\star^1 + \frac{\gamma}{\psi_\star^1}.$$

Hence

$$\lambda_{\mathsf{max}}\left(T_\gamma(\theta;\alpha)\right) \leq (1 + (1-\alpha)\lambda\gamma)^{-1}\left[\psi_\star^1 + \frac{\gamma}{\psi_\star^1} - \gamma(\lambda_{\mathsf{min}}(S) - \alpha\lambda p)\right] = \psi_\star^1,$$

where the last equality uses the fact that when $\psi_\star^1 < \infty$, it satisfies the equation

$$(1-\alpha)\lambda\psi_\star^2 + (\lambda_{\mathsf{min}}(S) - \alpha\lambda p)\,\psi_\star - 1 = 0.$$

We now consider the case where, $\psi_\star^1 > \psi_{UB}$, and $\theta_0 \in \mathcal{M}_+(\ell_\star, \psi_{UB})$. The first part of the proof guarantees that $\theta_j \in \mathcal{M}_+(\ell_\star, +\infty)$ for all $j \geq 0$. For $j \geq 1$, by Lemma 14 applied with $\ell = \ell_\star$, $\psi = +\infty$, $\theta = \theta_{j-1}$, and $H = \theta_{j-1}^{-1}$, we get

$$\left\|\theta_j - \hat\theta\right\|_{\mathsf{F}} \leq \left\|\theta_{j-1} - \hat\theta\right\|_{\mathsf{F}}.$$

This implies that for any $j \geq 1$,

$$\begin{aligned}
\|\theta_j\|_2 &\leq \|\hat\theta\|_2 + \|\theta_j - \hat\theta\|_2 \\
&\leq \|\hat\theta\|_2 + \left\|\theta_0 - \hat\theta\right\|_{\mathsf{F}} \\
&\leq \psi_{UB} + \sqrt{p}(\psi_{UB} - \ell_\star).
\end{aligned}$$

where the last inequality uses Weyl's inequality since $\theta_0, \hat\theta \in \mathcal{M}_+(\ell_\star, \psi_{UB})$.

■

8.3 Proof of Theorem 3

Proof We follow closely the proof of Theorem 3.1. of Beck and Teboulle (2009). Suppose that the sequence $\{\theta_i,\ 0 \leq i \leq k\}$ belongs to $\mathcal{M}_+(\ell, \psi)$. For any $i \geq 0$, since $\theta_{i+1} = \mathrm{Prox}_\gamma(\theta_i - \gamma(S - \theta_i^{-1}); \alpha)$, we apply Lemma 14 with $H = \theta_i^{-1}$ to obtain

$$\left\|\theta_{i+1} - \hat{\theta}\right\|_{\mathsf{F}}^2 \leq 2\gamma\left(\phi_\alpha(\theta_{i+1}) - \phi_\alpha(\hat{\theta})\right) + \left\|\theta_{i+1} - \hat{\theta}\right\|_{\mathsf{F}}^2 \leq \left(1 - \frac{\gamma}{\psi^2}\right)\left\|\theta_i - \hat{\theta}\right\|_{\mathsf{F}}^2,$$

which implies that

$$2\gamma\left(\phi_\alpha(\theta_k) - \phi_\alpha(\hat{\theta})\right) + \left\|\theta_k - \hat{\theta}\right\|_{\mathsf{F}}^2 \leq \left(1 - \frac{\gamma}{\psi^2}\right)^k \left\|\theta_0 - \hat{\theta}\right\|_{\mathsf{F}}^2. \tag{43}$$

Again, from (59), we have

$$\phi_\alpha(\theta_{i+1}) - \phi_\alpha(\hat{\theta}) \leq \frac{1}{2\gamma}\left[\left\|\theta_i - \hat{\theta}\right\|_{\mathsf{F}}^2 - \left\|\theta_{i+1} - \hat{\theta}\right\|_{\mathsf{F}}^2\right].$$

We then sum for $i = 0$ to $k-1$ to obtain

$$2\gamma\sum_{i=1}^{k}\left\{\phi_\alpha(\theta_i) - \phi_\alpha(\hat{\theta})\right\} + \left\|\theta_k - \hat{\theta}_0\right\|_{\mathsf{F}}^2 \leq \left\|\theta_0 - \hat{\theta}\right\|_{\mathsf{F}}^2. \tag{44}$$

We now use Lemma 13 to write

$$\begin{aligned} g_\alpha(\theta_{i+1}) - g_\alpha(\theta_i) &\leq \frac{1}{\gamma}\left\langle \theta_i - \theta_{i+1}, \theta_{i+1} - \left(\theta_i - \gamma(S - \theta_i^{-1})\right)\right\rangle, \\ &= -\frac{1}{\gamma}\left\|\theta_{i+1} - \theta_i\right\|_{\mathsf{F}}^2 + \left\langle \theta_i - \theta_{i+1}, S - \theta_i^{-1}\right\rangle. \end{aligned}$$

This last inequality together with (55) applied with $\bar{\theta} = \theta_{i+1}$ and $\theta = \theta_i$, yields

$$\left\{\phi_\alpha(\theta_{i+1}) - \phi_\alpha(\hat{\theta})\right\} \leq \left\{\phi_\alpha(\theta_i) - \phi_\alpha(\hat{\theta})\right\} - \frac{1}{2\gamma}\left\|\theta_i - \theta_{i+1}\right\|_{\mathsf{F}}^2. \tag{45}$$

By multiplying both sides of the last inequality by i and summing from 0 to $k-1$, we obtain

$$\begin{aligned} k\left\{\phi_\alpha(\theta_k) - \phi_\alpha(\hat{\theta})\right\} &\leq \sum_{i=1}^{k}\left\{\phi_\alpha(\theta_i) - \phi_\alpha(\hat{\theta})\right\} - \frac{1}{2}\sum_{i=0}^{k-1}\frac{i}{\gamma}\left\|\theta_i - \theta_{i+1}\right\|_{\mathsf{F}}^2 \\ &\leq \sum_{i=1}^{k}\left\{\phi_\alpha(\theta_i) - \phi_\alpha(\hat{\theta})\right\}. \end{aligned}$$

Hence, given (44), we have

$$\left\{\phi_\alpha(\theta_k) - \phi_\alpha(\hat{\theta})\right\} \leq \frac{1}{2\gamma k}\left\|\theta_0 - \hat{\theta}\right\|_{\mathsf{F}}^2,$$

which together with (43) yields the stated bound. ■

8.4 Proof of Theorem 6

Proof Write $\tau_\epsilon = \tau(\ell_\star(\epsilon), \psi_\star^1(\epsilon))$.

$$\mathbb{P}\left[\tau_\epsilon = \infty\right] = 1 - \sum_{j=1}^{\infty} \mathbb{P}\left[\tau_\epsilon = j\right],$$

and

$$\mathbb{P}\left[\tau_\epsilon = j\right] = \mathbb{P}\left[\left(\lambda_{\mathsf{min}}(\theta_j) < \ell_\star(\epsilon) \text{ or } \lambda_{\mathsf{max}}(\theta_j) > \psi_\star^1(\epsilon)\right),\ \tau_\epsilon > j-1\right].$$

Now we proceed as in the proof of Lemma 2. Given θ_{j-1}, the optimality condition (37) becomes: there exists a matrix Δ_j, all entries of which belong to $[-1, 1]$ (that can be taken as $\mathsf{sign}(\theta_j)$), such that

$$\theta_j = (1 + (1-\alpha)\lambda\gamma)^{-1}\left(\theta_{j-1} + \gamma\theta_{j-1}^{-1} - \gamma(S + (\theta_{j-1}^{-1} - G_j) + \alpha\lambda\Delta_j)\right).$$

As in the proof of Lemma 2 we have,

$$\begin{aligned}\lambda_{\mathsf{max}}(S + (\theta_{j-1}^{-1} - G_j) + \lambda\Delta_j) &\leq \lambda_{\mathsf{max}}(S) + p\|\theta_{j-1}^{-1} - G_j\|_\infty + p\lambda, \\ \text{and} \quad \lambda_{\mathsf{min}}(S + (\theta_{j-1}^{-1} - G_j) + \lambda\Delta_j) &\geq \lambda_{\mathsf{min}}(S) - p\|\theta_{j-1}^{-1} - G_j\|_\infty - p\lambda.\end{aligned}$$

where for $A \in \mathcal{M}$, $\|A\|_\infty \overset{\text{def}}{=} \max_{i,j}|A_{ij}|$. Therefore, with the same steps as in the proof of Lemma 2, we see that on the event $\{\tau > j-1, \|G_j - \theta_{j-1}^{-1}\|_\infty \leq \epsilon\}$, $\lambda_{\mathsf{min}}(\theta_j) \geq \ell_\star(\epsilon)$, and $\lambda_{\mathsf{max}}(\theta_j) \leq \psi_\star^1(\epsilon)$. We conclude that,

$$\mathbb{P}\left[\tau_\epsilon = j\right] \leq \mathbb{P}\left[\tau_\epsilon = j | \tau_\epsilon > j-1\right] \leq \mathbb{P}\left[\|G_j - \theta_{j-1}^{-1}\|_\infty > \epsilon | \tau_\epsilon > j-1\right].$$

We prove in Lemma 15 the exponential bound

$$\mathbb{P}\left[\|G_j - \theta_{j-1}^{-1}\|_\infty > \epsilon | \lambda_{\mathsf{min}}(\theta_{j-1}) \geq \ell_\star(\epsilon)\right] \leq 8p^2 \exp\left(-\min(1, \ell_\star^2(\epsilon)\epsilon^2/16)N_{j-1}\right).$$

Hence

$$\mathbb{P}\left[\tau_\epsilon = \infty\right] \geq 1 - 8p^2 \sum_{j\geq 1} \exp\left(-\min(1, \ell_\star^2(\epsilon)\epsilon^2/16)N_{j-1}\right). \tag{46}$$

We will now show that there exists a random variable $\Psi_\star(\epsilon)$ such that on $\{\tau_\epsilon = +\infty\}$, $\lambda_{\mathsf{max}}(\theta_j) \leq \Psi_\star(\epsilon)$ for all $j \geq 0$.

We first note that on $\{\tau_\epsilon > k\}$, $\theta_0, \ldots, \theta_k \in \mathcal{M}_+(\ell, \psi)$, and $\theta_j = \text{Prox}_\gamma\left(\theta_{j-1} - \gamma(S - \Sigma_j; \alpha)\right)$ for $j = 1, \ldots, k$. We then apply Lemma 14 with $\theta = \theta_{j-1}$, $\bar{\theta} = \theta_j$ and $H = \Sigma_j$, to write

$$\begin{aligned}\left\|\theta_j - \hat{\theta}\right\|_{\mathsf{F}}^2 &\leq \left\|\theta_j - \hat{\theta}\right\|_{\mathsf{F}}^2 + 2\gamma\left\{\phi_\alpha(\theta_j) - \phi_\alpha(\hat{\theta})\right\} \\ &\leq \left(1 - \frac{\gamma}{\psi^2}\right)\left\|\theta_{j-1} - \hat{\theta}\right\|_{\mathsf{F}}^2 - 2\gamma\left\langle\hat{\theta} - \theta_j, \Sigma_j - \theta_{j-1}^{-1}\right\rangle.\end{aligned}$$

We multiply both sides by $\mathbf{1}_{\{\tau_\epsilon > j-1\}}$ and uses the fact that $\mathbf{1}_{\{\tau_\epsilon > j-1\}} = \mathbf{1}_{\{\tau_\epsilon = j\}} + \mathbf{1}_{\{\tau_\epsilon > j\}}$ to write

$$\mathbf{1}_{\{\tau_\epsilon > j\}} \left\|\theta_j - \hat\theta\right\|_{\mathsf{F}}^2 \leq \left(1 - \frac{\gamma}{\psi^2}\right) \mathbf{1}_{\{\tau_\epsilon > j-1\}} \left\|\theta_{j-1} - \hat\theta\right\|_{\mathsf{F}}^2 - 2\gamma \mathbf{1}_{\{\tau_\epsilon > j-1\}} \left\langle \hat\theta - \theta_j, \Sigma_j - \theta_{j-1}^{-1} \right\rangle. \quad (47)$$

Recall that $\theta_j = \mathrm{Prox}_\gamma\left(\theta_{j-1} - \gamma(S - \Sigma_j); \alpha\right)$, and split $\hat\theta - \theta_j$ as

$$\hat\theta - \theta_j = \hat\theta - T_\gamma(\theta_{j-1}; \alpha) + T_\gamma(\theta_{j-1}; \alpha) - \theta_j, \quad (48)$$

where $T_\gamma(\theta_{j-1}; \alpha) = \mathrm{Prox}_\gamma\left(\theta_{j-1} - \gamma(S - \theta_{j-1}^{-1}); \alpha\right)$. It is well known that the proximal operator is non-expansive—see (Bauschke and Combettes, 2011, Propositions 12.26 and 12.27). Hence

$$\left|\left\langle T_\gamma(\theta_{j-1}; \alpha) - \theta_j, \Sigma_j - \theta_{j-1}^{-1} \right\rangle\right| \leq \|T_\gamma(\theta_{j-1}; \alpha) - \theta_j\|_{\mathsf{F}} \left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}} \leq \gamma \left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2.$$

We then set $V_j \overset{\text{def}}{=} \mathbf{1}_{\{\tau_\epsilon > j-1\}} \left\langle \hat\theta - T_\gamma(\theta_{j-1}; \alpha), \Sigma_j - \theta_{j-1}^{-1} \right\rangle$, and use the last inequality, (48), and (47) to deduce that

$$\mathbf{1}_{\{\tau_\epsilon > j\}} \left\|\theta_j - \hat\theta\right\|_{\mathsf{F}}^2 \leq \left(1 - \frac{\gamma}{\psi^2}\right) \mathbf{1}_{\{\tau_\epsilon > j-1\}} \left\|\theta_{j-1} - \hat\theta\right\|_{\mathsf{F}}^2 - 2\gamma V_j + 2\gamma^2 \mathbf{1}_{\{\tau_\epsilon > j-1\}} \left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2. \quad (49)$$

Summing (49) for $j = 1$ to k yields

$$\begin{aligned}\sup_{k\geq 0} \mathbf{1}_{\{\tau_\epsilon > k\}} \left\|\theta_k - \hat\theta\right\|_{\mathsf{F}}^2 &\leq \left\|\theta_0 - \hat\theta\right\|_{\mathsf{F}}^2 + 2\gamma \sup_{k\geq 1}\left|\sum_{j=1}^k V_j\right| \\ &\quad + 2\gamma^2 \sum_{j=1}^\infty \mathbf{1}_{\{\tau_\epsilon > j-1\}} \left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2, \\ &= \left\|\theta_0 - \hat\theta\right\|_{\mathsf{F}}^2 + \zeta, \qquad (50)\end{aligned}$$

where $\zeta \overset{\text{def}}{=} 2\gamma \sup_{k\geq 1}\left|\sum_{j=1}^k V_j\right| + 2\gamma^2 \sum_{j=1}^\infty \mathbf{1}_{\{\tau_\epsilon > j-1\}} \left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2$. The bound (50) in turn means that on the event $\{\tau_\epsilon = \infty\}$, for all $j \geq 0$,

$$\|\theta_j\|_2 \leq \|\hat\theta\|_2 + \left\|\theta_j - \hat\theta\right\|_{\mathsf{F}} \leq \psi_{UB} + \sqrt{p(\psi_{UB} - \ell_\star(\epsilon))^2 + \zeta}.$$

Hence, with $\Psi_\star(\epsilon) \overset{\text{def}}{=} \min\left(\psi_\star^1(\epsilon), \psi_{UB} + \sqrt{p(\psi_{UB} - \ell_\star(\epsilon))^2 + \zeta}\right)$, we have shown that $\{\tau_\epsilon = \infty\} \subset \{\tau(\ell_\star(\epsilon), \Psi_\star(\epsilon)) = \infty\}$, and the first part of the lemma follows from the bound (46).

`Bound on` $\mathbb{E}(\Psi_\star(\epsilon)^2)$ Clearly it suffices to bound $\mathbb{E}(\zeta)$. Recall that $\Sigma_j = \frac{1}{N_j}\sum_{k=1}^{N_j} z_k z_k'$, where $z_{1:N_j} \overset{i.i.d.}{\sim} \mathbf{N}(0, \theta_{j-1}^{-1})$. We easily calculate (See Lemma 15 for details) that on the event $\{\tau_\epsilon > j-1\}$,

$$\mathbb{E}\left(\left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2 |\mathcal{F}_{j-1}\right) = \frac{1}{N_j}\left(\mathsf{Tr}(\theta_{j-1}^{-1})^2 + \left\|\theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2\right),$$

and for $\theta_j \in \mathcal{M}_+(\ell_\star(\epsilon), \psi_\star^1(\epsilon))$, $\mathsf{Tr}(\theta_j^{-1})^2 + \left\|\theta_j^{-1}\right\|_{\mathsf{F}}^2 \leq \ell_\star(\epsilon)^{-2}(p+p^2)$. Hence

$$\begin{aligned}\mathbb{E}\left[\sum_{j=1}^\infty \mathbf{1}_{\{\tau > j-1\}} \left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2\right] &= \sum_{j=1}^\infty \mathbb{E}\left[\mathbf{1}_{\{\tau > j-1\}} \mathbb{E}\left(\left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2 |\mathcal{F}_{j-2}\right)\right] \\ &\leq \ell_\star(\epsilon)^{-2}(p+p^2)\sum_{j=1}^\infty \frac{1}{N_j} < \infty,\end{aligned}$$

by assumption. By Doob's inequality (Hall and Heyde (1980) Theorem 2.2) applied to the martingale $\{\sum_{j=1}^k V_k\}$,

$$\begin{aligned}\mathbb{E}\left[\sup_{k\geq 1}\left|\sum_{j=1}^k V_j\right|\right] &= \lim_{N\to\infty} \mathbb{E}\left[\sup_{1\leq k\leq N}\left|\sum_{j=1}^k V_j\right|\right] \leq 2\lim_{N\to\infty} \mathbb{E}^{1/2}\left[\left|\sum_{j=1}^N V_j\right|^2\right] \\ &= 2\left\{\sum_{j=1}^\infty \mathbb{E}(V_j^2)\right\}^{1/2}.\end{aligned}$$

Using again the facts that the proximal operator is non-expansive and $\hat\theta = T_\gamma(\hat\theta;\alpha)$, we have $|V_j| \leq \mathbf{1}_{\{\tau_\epsilon > j-1\}} \left\|\theta_{j-1} - \hat\theta\right\|_{\mathsf{F}} \left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}}$. Therefore, with similar calculations as above, we have

$$\mathbb{E}(V_j^2) = \mathbb{E}\left[\mathbb{E}(V_j^2|\mathcal{F}_{j-1})\right] \leq \ell_\star(\epsilon)^{-2}(p+p^2)N_j^{-1}\mathbb{E}\left(\mathbf{1}_{\{\tau_\epsilon > j-1\}}\left\|\theta_{j-1} - \hat\theta\right\|_{\mathsf{F}}^2\right).$$

On $\{\tau_\epsilon > j-1\}$, $\left\|\theta_{j-1} - \hat\theta\right\|_{\mathsf{F}} \leq \sqrt{p}\|\theta_{j-1} - \hat\theta\|_2 \leq \sqrt{p}(\psi_{UB} - \ell_\star(\epsilon))$. Hence

$$\mathbb{E}(V_j^2) \leq \frac{p(p+p^2)(\psi_{UB} - \ell_\star(\epsilon))^2}{\ell_\star(\epsilon)^2}\frac{1}{N_j},$$

which together with the assumption $\sum_j N_j^{-1} < \infty$, and the above calculation show that $\mathbb{E}\left[\sup_{k\geq 1}\left|\sum_{j=1}^k V_j\right|\right] < \infty$.

`Convergence of` θ_n We sum (49) from $j=1$ to k, which gives, for all $k \geq 1$:

$$\begin{aligned}&\mathbf{1}_{\{\tau_\epsilon > k\}}\left\|\theta_k - \hat\theta\right\|_{\mathsf{F}}^2 + \frac{\gamma}{\psi^2}\sum_{j=1}^k \mathbf{1}_{\{\tau_\epsilon > j-1\}}\left\|\theta_{j-1} - \hat\theta\right\|_{\mathsf{F}}^2 \\ &\qquad\qquad \leq 2\gamma\sup_{k\geq 1}\left|\sum_{j=1}^k V_j\right| + 2\gamma^2\sum_{j=1}^\infty \mathbf{1}_{\{\tau_\epsilon > j-1\}}\left\|\Sigma_j - \theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2.\end{aligned}$$

We have seen above that the term on the right-hand side of this inequality has a finite expectation. This implies the series $\sum_{j=1}^{\infty}\mathbf{1}_{\{\tau_\epsilon>j-1\}}\left\|\theta_{j-1}-\hat{\theta}\right\|_{\mathsf{F}}^2$ is finite almost surely, which in turn implies that on $\{\tau_\epsilon=\infty\}$, we necessarily have $\lim_k \theta_k=\hat{\theta}$, as claimed.

■

8.5 Proof of Theorem 7

Proof Taking the expectation on both sides on (49) yields

$$\mathbb{E}\left[\mathbf{1}_{\{\tau>j\}}\left\|\theta_j-\hat{\theta}\right\|_{\mathsf{F}}^2\right]\leq\left(1-\frac{\gamma}{\psi^2}\right)\mathbb{E}\left[\mathbf{1}_{\{\tau>j-1\}}\left\|\theta_{j-1}-\hat{\theta}\right\|_{\mathsf{F}}^2\right]$$
$$+2\gamma^2\mathbb{E}\left[\mathbf{1}_{\{\tau>j-1\}}\mathbb{E}\left(\left\|\Sigma_j-\theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2|\mathcal{F}_{j-1}\right)\right].$$

Iterating this inequality yields

$$\mathbb{E}\left[\mathbf{1}_{\{\tau>k\}}\left\|\theta_k-\hat{\theta}\right\|_{\mathsf{F}}^2\right]\leq\left(1-\frac{\gamma}{\psi^2}\right)^k\left\|\theta_0-\hat{\theta}\right\|_{\mathsf{F}}^2$$
$$+2\gamma^2\sum_{j=1}^{k}\left(1-\frac{\gamma}{\psi^2}\right)^{k-j}\mathbb{E}\left[\mathbf{1}_{\{\tau>j-1\}}\mathbb{E}\left(\left\|\Sigma_j-\theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2\right)\right].$$

Recall that $\Sigma_j=\frac{1}{N_j}\sum_{k=1}^{N_j}z_kz_k'$, where $z_{1:N_j}\overset{i.i.d.}{\sim}\mathbf{N}(0,\theta_{j-1}^{-1})$. We easily calculate (See Lemma 15 for details) that on the event $\{\tau>j-1\}$,

$$\mathbb{E}\left(\left\|\Sigma_j-\theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2|\mathcal{F}_{j-1}\right)=\frac{1}{N_j}\left(\mathsf{Tr}(\theta_{j-1}^{-1})^2+\left\|\theta_{j-1}^{-1}\right\|_{\mathsf{F}}^2\right),$$

and for $\theta_j\in\mathcal{M}_+(\ell,\psi)$, $\mathsf{Tr}(\theta_j^{-1})^2+\left\|\theta_j^{-1}\right\|_{\mathsf{F}}^2\leq\ell^{-2}(p+p^2)$. The stated bound on the term $\mathbb{E}\left[\mathbf{1}_{\{\tau>k\}}\left\|\theta_k-\hat{\theta}\right\|_{\mathsf{F}}^2\right]$ then follows.

■

8.6 Proof of Theorem 9

Proof We write $\tau=\tau(\ell,\psi)$. On $\{\tau>k\}$, $\theta_0,\ldots,\theta_k\in\mathcal{M}_+(\ell,\psi)$, and $\theta_{i+1}=\mathrm{Prox}_\gamma(\theta_i-\gamma(S-\Sigma_{i+1};\alpha)$ for $i\geq 0$. We apply Lemma 14 with $H=\Sigma_{i+1}$ to write

$$\left\|\theta_{i+1}-\hat{\theta}\right\|_{\mathsf{F}}^2\leq\left(1-\frac{\gamma}{\psi^2}\right)\left\|\theta_i-\hat{\theta}\right\|_{\mathsf{F}}^2+2\gamma\left\langle\theta_{i+1}-\hat{\theta},\Sigma_{i+1}-\theta_i^{-1}\right\rangle.$$

By iterating this bound, we obtain

$$\left\|\theta_k-\hat{\theta}\right\|_{\mathsf{F}}^2 \leq \left(1-\frac{\gamma}{\psi^2}\right)^k\left\|\theta_0-\hat{\theta}\right\|_{\mathsf{F}}^2$$
$$+2\gamma\sup_{k\geq 0}\left\|\theta_k-\hat{\theta}\right\|_{\mathsf{F}}^2\sum_{j=1}^{k}\left(1-\frac{\gamma}{\psi^2}\right)^{k-j}\left\|\Sigma_{j+1}-\theta_j^{-1}\right\|_{\mathsf{F}}. \quad (51)$$

On $\{\tau(\ell,\psi)=\infty\}$, $\sup_{i\geq 0}\left\|\theta_i-\hat{\theta}\right\|_{\mathsf{F}}^2$ is finite and if $\lim_j \left\|\Sigma_{j+1}-\theta_j^{-1}\right\|_{\mathsf{F}}=0$, the bound (51) would easily imply that $\lim_k \left\|\theta_k-\hat{\theta}\right\|_{\mathsf{F}}^2=0$. Hence the theorem is proved by showing that on $\{\tau=\infty\}$, $\lim_k \left\|\Sigma_{k+1}-\theta_k^{-1}\right\|_{\mathsf{F}}=0$. From (25), we write

$$\Sigma_{k+1}-\theta_k^{-1}=(1-\zeta_{k+1})\left(\Sigma_k-\theta_{k-1}^{-1}\right)+(1-\zeta_{k+1})(\theta_{k-1}^{-1}-\theta_k^{-1})+\zeta_{k+1}\eta_{k+1},$$

where

$$\eta_{k+1}\stackrel{\text{def}}{=}\frac{1}{N}\sum_{k=1}^{N} z_k z_k'-\theta_k^{-1},\quad z_{1:N}\stackrel{i.i.d.}{\sim}\mathbf{N}(0,\theta_k^{-1}).$$

We expand this into

$$\mathbf{1}_{\{\tau>k\}}\left(\Sigma_{k+1}-\theta_k^{-1}\right)=(1-\zeta_{k+1})\mathbf{1}_{\{\tau>k-1\}}\left(\Sigma_k-\theta_{k-1}^{-1}\right)+R_{k+1}^{(1)}+R_{k+1}^{(2)}+R_{k+1}^{(3)}+R_{k+1}^{(4)},$$

where the remainders are given by

$$R_{k+1}^{(1)}\stackrel{\text{def}}{=}-\mathbf{1}_{\{\tau=k\}}(1-\zeta_{k+1})\Sigma_k,$$

$$R_{k+1}^{(2)}\stackrel{\text{def}}{=}(1-\zeta_k)\mathbf{1}_{\{\tau>k-1\}}\theta_{k-1}^{-1}-(1-\zeta_{k+1})\mathbf{1}_{\{\tau>k\}}\theta_k^{-1},$$

$$R_{k+1}^{(3)}\stackrel{\text{def}}{=}(\zeta_k-\zeta_{k+1})\mathbf{1}_{\{\tau>k-1\}}\theta_{k-1}^{-1},$$

and

$$R_{k+1}^{(4)}\stackrel{\text{def}}{=}\zeta_{k+1}\mathbf{1}_{\{\tau>k\}}\eta_{k+1}.$$

Since $\mathbf{1}_{\{\tau>k,\tau=\infty\}}=\mathbf{1}_{\{\tau=\infty\}}$, and $\mathbf{1}_{\{\tau=k,\tau=\infty\}}=0$, it follows that for all $n\geq 0$,

$$\mathbf{1}_{\{\tau=\infty\}}\left(\Sigma_{k+1}-\theta_k^{-1}\right)=\mathbf{1}_{\{\tau=\infty\}}\prod_{k=1}^{k}(1-\zeta_{k+1})(\Sigma_1-\theta_0^{-1})$$
$$+\mathbf{1}_{\{\tau=\infty\}}\sum_{j=1}^{k}\left(R_j^{(2)}+R_j^{(3)}+R_j^{(4)}\right)\prod_{i=j+1}^{k}(1-\zeta_{i+1}).$$

Clearly, we have $\prod_{k=1}^{k}(1-\zeta_{k+1})\leq\exp\left(-\sum_{k=1}^{k}\zeta_{k+1}\right)\to 0$ as $k\to\infty$ by (24), and if the series $\sum_{j\geq 1}\left(R_j^{(2)}+R_j^{(3)}+R_j^{(4)}\right)$ is finite on $\{\tau=\infty\}$, then by Kronecker lemma, it would follow that $\sum_{j=1}^{k}\left(R_j^{(2)}+R_j^{(3)}+R_j^{(4)}\right)\prod_{i=j+1}^{k}(1-\zeta_{i+1})\to 0$, as $k\to\infty$ on $\{\tau=\infty\}$. Hence, it suffices to prove that the series $\sum_{j\geq 1}\left(R_j^{(2)}+R_j^{(3)}+R_j^{(4)}\right)$ is finite on $\{\tau=\infty\}$.

We have $\sum_{k=1}^{k}R_k^{(2)}=(1-\zeta_1)\mathbf{1}_{\{\tau>0\}}\theta_0^{-1}-(1-\zeta_{k+1})\mathbf{1}_{\{\tau>k\}}\theta_k^{-1}$. The assumption that θ_k has a limit and $\theta_k\in\mathcal{M}_+(\ell,\psi)$ easily implies that $\sum_k R_k^{(2)}$ is finite. Similarly, we have $\sum_k\left\|R_k^{(3)}\right\|_{\mathsf{F}}\leq\ell^{-1}\zeta_0<\infty$, and

$$\mathbb{E}\left(\left\|\sum_k R_k^{(4)}\right\|_{\mathsf{F}}^2\right)=\sum_k\zeta_k^2\mathbb{E}\left(\mathbf{1}_{\{\tau>k\}}\left\|\frac{1}{N}\sum_{k=1}^{N}z_kz_k'-\theta_k^{-1}\right\|_{\mathsf{F}}^2\right)\leq\ell^{-2}(p+p^2)\sum_k\zeta_k^2<\infty.$$

■

8.7 Proof of Theorem 10

Proof The proof follows (Mazumder and Hastie, 2012a) with appropriate modifications, and we provide a brief sketch here.
First Part:
We start with the connected component decomposition of the non-zeros of $\hat{\theta}$. Let us assume that the rows/columns of the matrix $\hat{\theta}$ have been arranged such that it is block diagonal. We proceed by writing the KKT conditions of Problem (3):

$$-\hat{\theta}^{-1} + S + \tau_1 \operatorname{sgn}(\widehat{\theta}) + 2\tau_2\widehat{\theta} = 0, \tag{52}$$

where, $\tau_1 = \alpha\lambda_1$ and $\tau_2 = \frac{1-\alpha}{2}\lambda_2$ and $\operatorname{sgn}(\widehat{\theta})$ is a matrix where $\operatorname{sgn}(\cdot)$ is applied componentwise to every entry of $\widehat{\theta}$. Since $\hat{\theta}$ is block diagonal so is $\hat{\theta}^{-1}$. If we take the (i,j)th entry of the matrix appearing in (52) such that i and j belong to two different connected components then: $-(\hat{\theta}^{-1})_{ij} + 2\tau\hat{\theta}_{ij} = 0$ which implies that $s_{ij} + \tau_1 \operatorname{sgn}(\widehat{\theta}_{ij}) = 0$. Thus we have: $|s_{ij}| \leq \tau_1$ for all pairs i, j such that they belong to two different connected components. Thus the binary matrix $((\mathbf{1}(|s_{ij}| > \tau_1)))$ will have zeros for all i, j belonging to two different components $\widehat{\mathcal{V}}_r$ and $\widehat{\mathcal{V}}_s$ for $r \neq s$. The connected components of $((\mathbf{1}(|s_{ij}| > \tau_1)))$ have a finer resolution than $\widehat{\mathcal{V}}_j, j = 1, \ldots, \widehat{J}$ and in particular $\widehat{J} \leq J$.
Second Part:
For the other part, let us start by assuming that the symmetric binary matrix $((\mathbf{1}(|s_{ij}| > \tau_1)))$ breaks down into J many connected components; and let $\widetilde{\theta} = \operatorname{diag}(\hat{\theta}_1, \ldots, \hat{\theta}_J)$ be a block diagonal matrix, where, each $\hat{\theta}_r$ is obtained by solving Problem (3) restricted to the rth connected component $\mathcal{V}_r$ where, $r = 1, \ldots, J$. For any i, j belonging to two different components $\mathcal{V}_r$ and $\mathcal{V}_s$ with $r \neq s$ we have that $|s_{ij}| \leq \tau_1$ and in addition, $\widetilde{\theta}_{ij} = 0$ and $(\widetilde{\theta}^{-1})_{ij} = 0$. This implies that $\widetilde{\theta}$ satisfies the KKT conditions (52) and is hence a solution to Problem (3). This in particular, implies that $\widehat{J} \geq J$ and the connected components of $\widehat{\mathcal{V}}_j, j = 1, \ldots, \widehat{J}$ are a finer resolution than $\mathcal{V}_r, r = 1, \ldots, J$.

Combining the above two parts, we conclude that the connected components of the two binary matrices $((\mathbf{1}(|s_{ij}| > \tau_1)))$ and $((\mathbf{1}(|\hat{\theta}_{ij} \neq 0)))$ are indeed equal.

■

8.8 Proof of Lemma 11

Proof To see this we take the derivative of the objective function wrt θ and set it to zero:

$$-\theta^{-1} + S + \lambda\theta = 0. \tag{53}$$

Suppose that the sample covariance matrix S can be written as:

$$S = UDU',$$

where the above denotes the full eigen-value decomposition of S which is a $p \times p$ matrix. Let d_i denote the diagonals of D. We will show that the solution to Problem (2) is of the form $\hat{\theta} = U\operatorname{diag}(\sigma)U'$, where, $\operatorname{diag}(\sigma)$ is a diagonal matrix with the ith diagonal entry being σ_i.

Let us multiply both sides of (53) by U' and U on the left and right respectively. It is then easy to see that the optimal values of σ can be computed as follows:

$$-1/\sigma_i + d_i + \lambda\sigma_i = 0$$

for all $i = 1, \ldots, p$. The above can be solved for every i separately leading to:

$$\hat{\sigma}_i = \frac{-d_i + \sqrt{d_i^2 + 4\lambda}}{2\lambda}, \quad \forall i$$

Thus we have the statement of Lemma 11. ■

Acknowledgements

Yves F. Atchadé is partly supported by NSF grant DMS 1228164. Rahul Mazumder was supported by ONR grant ONR - N00014-15-1-2342, Columbia University's start-up fund and an interface grant from the Betty-Moore Sloan Foundation. R.M. will like to thank Robert Freund for helpful comments and encouragement.

References

Y. F. Atchade, G. Fort, and E. Moulines. On stochastic proximal gradient algorithms. *ArXiv e-prints*, February 2014.

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.

Heinz H. Bauschke and Patrick L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces.* CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC. Springer, New York, 2011.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

Stephen R Becker, Emmanuel J Candès, and Michael C Grant. Templates for convex cone problems with applications to sparse signal recovery. *Mathematical Programming Computation*, 3(3):165–218, 2011.

Dimitri P. Bertsekas. Incremental proximal methods for large scale convex optimization. *Mathematical Programming*, 129(2, Ser. B):163–195, 2011.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization.* Cambridge University Press, Cambridge, 2004.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning*, (3(1)), 2011.

Peter Bühlmann and Sara Van De Geer. *Statistics for high-dimensional data: methods, theory and applications.* Springer Science & Business Media, 2011.

Jie Chen, Mihai Anitescu, and Yousef Saad. Computing $f(A)b$ via least squares polynomial approximations. *SIAM Journal on Scientific Computing*, 33(1):195–222, 2011.

John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. Randomized smoothing for stochastic optimization. *SIAM Journal on Optimization*, 22(2):674–701, 2012. ISSN 1052-6234.

Michael Eiermann and Oliver G. Ernst. A restarted krylov subspace method for the evaluation of matrix functions. *SIAM Journal on Numerical Analysis*, 44(6):2481–2504, 2006.

J. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84:165–175, 1989.

Jerome Friedman, Trevor Hastie, Holger Hoefling, and Robert Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 2(1):302–332, 2007a.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9:432–441, 2007b.

Gene H. Golub and Charles F. Van Loan. *Matrix Computations.* Johns Hopkins Studies in the Mathematical Sciences. Johns Hopkins University Press, Baltimore, MD, fourth edition, 2013.

Nicholas Hale, Nicholas J. Higham, and Lloyd N. Trefethen. Computing A^{α}, $\log(A)$, and related matrix functions by contour integrals. *SIAM Journal of Numerical Analysis*, 46 (5):2505–2523, 2008.

P. Hall and C. C. Heyde. *Martingale Limit theory and its application.* Academic Press, New York, 1980.

Trevor Hastie, Andreas Buja, and Robert Tibshirani. Penalized discriminant analysis. *The Annals of Statistics*, pages 73–102, 1995.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning, Second Edition: Data Mining, Inference, and Prediction (Springer Series in Statistics).* Springer New York, 2 edition, 2009.

Cho-Jui Hsieh, Mátyás A. Sustik, Inderjit S. Dhillon, and Pradeep Ravikumar. Quic: Quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research*, 15:2911–2947, 2014.

J. Kiefer and J. Wolfowitz. Stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, 23:462–466, 1952.

J. Konečný and P. Richtárik. Semi-Stochastic Gradient Descent Methods. *ArXiv e-prints*, December 2013.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *The Annals of Statistics*, 28(5):1302–1338, 2000.

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

Lu Li and Kim-Chuan Toh. An inexact interior point method for l1-regularized sparse covariance selection. *Mathematical Programming Computation*, 31:2000–2016, May 2010.

Zhaosong Lu. Smooth optimization approach for sparse covariance selection. *SIAM Journal on Optimization*, 19:1807–1827, 2009.

Rahul Mazumder and Trevor Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *The Journal of Machine Learning Research*, 13: 781–794, 2012a.

Rahul Mazumder and Trevor Hastie. The graphical lasso: New insights and alternatives. *Electronic Journal of Statistics*, 6:2125–2149, 2012b.

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

Yu. Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

Figen Oztoprak, Jorge Nocedal, Steven Rennie, and Peder A Olsen. Newton-like methods for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems*, pages 755–763, 2012.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):123–231, 2013.

Mohsen Pourahmadi. *High-Dimensional Covariance Estimation: With High-Dimensional Data*. John Wiley & Sons, 2013.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 1951.

Benjamin Rolfs, Bala Rajaratnam, Dominique Guillot, Ian Wong, and Arian Maleki. Iterative thresholding algorithm for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems*, pages 1574–1582, 2012.

Katya Scheinberg, Shiqian Ma, and Donald Goldfarb. Sparse inverse covariance selection via alternating linearization methods. In *Advances in Neural Information Processing Systems*, pages 2101–2109, 2010.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *Journal of Machine Learning Research*, 14:567–599, 2013.

Lieven Vandenberghe and Stephen Boyd. Semidefinite programming. *SIAM review*, 38(1): 49–95, 1996.

Lieven Vandenberghe, Stephen Boyd, and Shao-Po Wu. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499–533, 1998.

David I Warton. Penalized normal likelihood and ridge regularization of correlation and covariance matrices. *Journal of the American Statistical Association*, 103(481), 2008.

Daniela M Witten, Jerome H Friedman, and Noah Simon. New insights and faster computations for the graphical lasso. *Journal of Computational and Graphical Statistics*, 20(4): 892–900, 2011.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

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

Xiaoming Yuan. Alternating direction method for covariance selection models. *Journal of Scientific Computing*, 51(2):261–273, 2012.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B.*, 67(2):301–320, 2005.

Appendix

Appendix A. Related Work and Algorithms

In this section we review some of the state-of-the art methods and approaches for the Glasso problem (Problem (1)). Problem (1) is a nonlinear convex semidefinite optimization problem (Vandenberghe and Boyd, 1996) and off-the-shelf interior point solvers typically have a per-iteration complexity of $O(p^6)$ that stems from solving a typically dense system with $O(p^2)$ variables (Vandenberghe et al., 1998). This makes generic interior point solvers inapplicable for solving problems with p of the order of a few hundred.

A popular approach to optimize problem (1) is to focus on its dual optimization problem, given by:

$$\underset{w \in \mathcal{M}_+}{\text{maximize}} \quad \log\det(w) \quad \text{subject to} \quad \|S - w\|_\infty \le \lambda, \tag{54}$$

with primal dual relationship given by $w = \theta^{-1}$. The dual problem has a smooth function appearing in its objective. Many efficient solvers for Problem (1) optimize the dual Problem (54) — see for example Banerjee et al. (2008); Friedman et al. (2007a); Lu (2009); Mazumder and Hastie (2012b) and references therein.

In one of the earlier works, Banerjee et al. (2008) consider solving the dual Problem (54). They propose a smooth accelerated gradient based method (Nesterov, 2005) with complexity $O(\frac{p^{4.5}}{\delta})$ to obtain a δ-accurate solution — the per iteration cost being $O(p^3)$. They also proposed a block coordinate method which requires solving at every iteration, a box constrained quadratic program (QP) which they solve using Interior point methods—leading to an overall complexity of $O(p^4)$.

The *graphical lasso* algorithm (Friedman et al., 2007b) is widely regarded as one of the most efficient and practical algorithms for Problem (1). The algorithm uses a row-by-row block coordinate method that requires to solve a ℓ_1 regularized quadratic program for every row/column—the authors use one-at-a-time cyclical coordinate descent to solve the QPs to high accuracy. While it is difficult to provide a precise complexity result for this method, the cost is roughly $O(p^3)$ for (reasonably) sparse-problems with p nodes. For dense problems the cost can be as large as $O(p^4)$, or even more. Mazumder and Hastie (2012b) further investigate the properties of the *graphical lasso* algorithm, its operational characteristics and propose another block coordinate method for Problem (1) that often enjoys better numerical behavior than *graphical lasso*.

The algorithm proposed in Lu (2009) employs accelerated gradient based algorithms (Nesterov, 2005, 2013). The algorithm smacs proposed in the paper has a per iteration complexity of $O(p^3)$ and an overall complexity of $O(\frac{p^4}{\sqrt{\delta}})$ to reach a δ-accurate optimal solution.

Li and Toh (2010) propose a specialized interior point algorithm for problem (1). By rewriting the objective as a smooth convex optimization problem by doubling the number of variables, the paper proposes a scheme to scale interior point like methods up to $p = 2000$.

Scheinberg et al. (2010) propose alternating direction based methods for the problem, the main complexity per iteration being $O(p^3)$ associated with a full spectral decomposition of a $p \times p$ symmetric matrix and a matrix inversion. Yuan (2012) propose an alternating direction method for problem (1), with per iteration complexity of $O(p^3)$. Computational scalability of a similar type can also be achieved by using the alternating direction method of multipliers ADMM Boyd et al. (2011) which perform spectral decompositions and/or matrix inversions with per iteration complexity $O(p^3)$.

Fairly recently, Hsieh et al. (2014) propose a Newton-like method for Problem (1), the algorithm is known as Quic. The main idea is to reduce the problem to iteratively solving large scale ℓ_1 regularized quadratic programs, which are solved using one-at-a-time coordinate descent update rules. The authors develop asymptotic convergence guarantees of the algorithm. It appears that several computational tricks and fairly advanced implementations in C++ are used to make the approach scalable to large problems. At the time of writing this paper, QUIC seems to be one of the most advanced algorithms for Glasso. Oztoprak et al. (2012) propose a related approach based on a Newton-like quadratic approximation of the log-determinant function.

Appendix B. Additional Computational Details

We initialize all the solvers using the diagonal matrix obtained by taking the inverse sample variances. For all the simulated-data experiments, the step-size and the Monte Carlo batch-size are taken as follows. The step-size is set to $\gamma = 10$, the Monte Carlo batch-size is set to $N_k = \lceil 30 + k^{1.8} \rceil$ at iteration k. Additionally, for Algorithm 3 we use $N = 400$, and $\zeta_k = k^{-0.7}$.

For $p = 1000$, the values of the regularization parameters were taken as $(\alpha, \lambda) = (0.89, 0.01)$. We computed $\hat{\theta}$ (the target solution to the optimization problem) by running the deterministic algorithm for 1000 iterations.

The size of the largest component is 967, one component had size two with all other components having size one. In this case, the splitting offered marginal improvements since the size of the maximal component was quite close to p.

For $p = 5000$ the values of the regularization parameters were taken as $(\alpha, \lambda) = (0.93, 0.0085)$ and we computed $\hat{\theta}$ (the target solution to the optimization problem) by running the deterministic algorithm for 1000 iterations.

For the case, $p = 5,000$ splitting leads to 76 connected components, The size of the largest component is 4924 with all other components having size one.

For $p = 10,000$, the values of the regularization parameters were taken as $(\alpha, \lambda) = (0.96, 0.01)$. We computed $\hat{\theta}$ (the target solution to the optimization problem) by running the deterministic algorithm for 500 iterations.

For the case, $p = 10,000$ splitting leads to 1330 connected components, The size of the largest component is 8670, one component has size two with all other components having size one.

We present the results for the cases $p = 5,000$ and $p = 10,000$ in Table 1.

For the real-data example, the stochastic algorithms are set up as follows. The step-size is set to $\gamma = 5 \times 10^{-5}$, the Monte Carlo batch-size is set to $N_k = \lceil 100 + k^{1.8} \rceil$ at iteration k. Additionally, for Algorithm 3 we use $N = 200$, and $\zeta_k = k^{-0.52}$.

Appendix C. Some Technical Lemmas and Proofs

Lemma 12 *Consider the function $f(\theta) = -\log\det\theta + \mathsf{Tr}(\theta S)$, $\theta \in \mathcal{M}_+$. Take $0 < \ell < \psi \leq \infty$. If $\theta \in \mathcal{M}_+(\ell, \psi)$, and $H \in \mathcal{M}$ are such that $\theta + H \in \mathcal{M}_+(\ell, \psi)$, then*

$$f(\theta) + \left\langle S - \theta^{-1}, H\right\rangle + \frac{1}{2\psi^2}\|H\|^2 \leq f\left(\theta + H\right) \leq f(\theta) + \left\langle S - \theta^{-1}, H\right\rangle + \frac{1}{2\ell^2}\|H\|^2.$$

Proof First notice that $\mathcal{M}_+(\ell, \psi)$ is a convex set. Hence for all $t \in [0, 1]$, $\theta + tH = (1-t)\theta + t(\theta + H) \in \mathcal{M}_+(\ell, \psi)$. Then by Taylor expansion we have,

$$\log\det(\theta + H) = \log\det\theta + \left\langle \theta^{-1}, H\right\rangle + \int_0^1 \left\langle (\theta + tH)^{-1} - \theta^{-1}, H\right\rangle \mathrm{d}t.$$

This gives

$$f(\theta + H) - f(\theta) - \left\langle S - \theta^{-1}, H\right\rangle = -\int_0^1 \left\langle (\theta + tH)^{-1} - \theta^{-1}, H\right\rangle \mathrm{d}t.$$

However $(\theta + tH)^{-1} - \theta^{-1} = -t\theta^{-1}H(\theta + tH)^{-1}$. Therefore, if $\theta = \sum_{j=1}^p \lambda_j u_j u_j'$ denotes the eigen-decomposition of θ, we have

$$\begin{aligned} -\left\langle (\theta + tH)^{-1} - \theta^{-1}, H\right\rangle &= t\mathsf{Tr}\left(\theta^{-1}H(\theta + tH)^{-1}H\right) \\ &= t\sum_{j=1}^p \lambda_j^{-1} u_j' H(\theta + tH)^{-1} H u_j \\ &\leq \frac{t}{\ell^2}\sum_{j=1}^p \|Hu_j\|^2 = \frac{t}{\ell^2}\|H\|^2. \end{aligned}$$

Similarly calculations gives

$$-\left\langle (\theta + tH)^{-1} - \theta^{-1}, H\right\rangle \geq \frac{t}{\psi^2}\|H\|^2.$$

The lemma is proved. ■

We also use the following well known property of the proximal map.

Lemma 13 *For all $\theta, \vartheta \in \mathcal{M}$, and for all $\alpha \in [0, 1]$, $\gamma > 0$,*

$$g_\alpha(\mathrm{Prox}_\gamma(\theta; \alpha)) \leq g_\alpha(\vartheta) + \frac{1}{\gamma}\left\langle \vartheta - \mathrm{Prox}_\gamma(\theta; \alpha), \mathrm{Prox}_\gamma(\theta; \alpha) - \theta\right\rangle.$$

Proof See (Bauschke and Combettes, 2011, Propositions 12.26 and 12.27). ■

Lemma 12 amd Lemma 13 together give the following key result.

Lemma 14 *Fix $0 < \ell < \psi \leq \infty$, and $\gamma \in (0, \ell^2]$. Suppose that $\hat{\theta}, \theta \in \mathcal{M}_+(\ell, \psi)$, and $H \in \mathcal{M}$ are such that $\bar{\theta} \overset{\text{def}}{=} \text{Prox}_\gamma(\theta - \gamma(S - H); \alpha) \in \mathcal{M}_+(\ell, \psi)$. Then*

$$\left\|\bar{\theta} - \hat{\theta}\right\|_F^2 \leq 2\gamma\left(\phi_\alpha(\bar{\theta}) - \phi_\alpha(\hat{\theta})\right) + \left\|\bar{\theta} - \hat{\theta}\right\|_F^2$$
$$\leq \left(1 - \frac{\gamma}{\psi^2}\right)\left\|\theta - \hat{\theta}\right\|_F^2 + 2\gamma\left\langle \bar{\theta} - \hat{\theta}, H - \theta^{-1}\right\rangle,$$

where we recall that $\phi_\alpha(\theta) = f(\theta) + g_\alpha(\theta)$.

Proof Set $f(\theta) = -\log\det\theta + \mathsf{Tr}(\theta S)$, $\theta \in \mathcal{M}_+$. By Lemma 12,

$$f(\bar{\theta}) \leq f(\theta) + \left\langle S - \theta^{-1}, \bar{\theta} - \theta\right\rangle + \frac{1}{2\gamma}\left\|\bar{\theta} - \theta\right\|_{\mathsf{F}}^2. \tag{55}$$

Subtracting $f(\hat{\theta})$ from both sides of the above inequality and re-arranging gives

$$\begin{aligned} f(\bar{\theta}) - f(\hat{\theta}) \quad &\leq \quad \left[f(\theta) + \left\langle S - \theta^{-1}, \hat{\theta} - \theta\right\rangle - f(\hat{\theta})\right] \\ &\quad + \left\langle S - \theta_i^{-1}, \bar{\theta} - \hat{\theta}\right\rangle + \frac{1}{2\gamma}\left\|\bar{\theta} - \theta\right\|_{\mathsf{F}}^2. \end{aligned} \tag{56}$$

Since $\theta, \hat{\theta} \in \mathcal{M}_+(\ell, \psi)$, the strong convexity of $\theta \mapsto -\log\det\theta + \mathsf{Tr}(\theta S)$ established in Lemma 12 implies that $f(\theta) + \left\langle S - \theta^{-1}, \hat{\theta} - \theta\right\rangle - f(\hat{\theta}) \leq -\frac{1}{2\psi^2}\left\|\theta - \hat{\theta}\right\|_{\mathsf{F}}^2$. Using this in (56) gives

$$f(\bar{\theta}) - f(\hat{\theta}) \leq -\frac{1}{2\psi^2}\left\|\theta - \hat{\theta}\right\|_{\mathsf{F}}^2 + \left\langle S - \theta^{-1}, \bar{\theta} - \hat{\theta}\right\rangle + \frac{1}{2\gamma}\left\|\bar{\theta} - \theta\right\|_{\mathsf{F}}^2. \tag{57}$$

By Lemma 13,

$$\begin{aligned} g_\alpha(\bar{\theta}) - g_\alpha(\hat{\theta}) \quad &\leq \quad \frac{1}{\gamma}\left\langle \hat{\theta} - \bar{\theta}, \bar{\theta} - (\theta - \gamma(S - H))\right\rangle, \\ &= \quad \frac{1}{\gamma}\left\langle \hat{\theta} - \bar{\theta}, \bar{\theta} - \theta\right\rangle + \left\langle \hat{\theta} - \bar{\theta}, S - H\right\rangle. \end{aligned} \tag{58}$$

We combine (57) and (58) and re-arrange to deduce that

$$\begin{aligned} \phi_\alpha(\bar{\theta}) - \phi_\alpha(\hat{\theta}) &\leq -\frac{1}{2\psi^2}\left\|\theta - \hat{\theta}\right\|_{\mathsf{F}}^2 + \frac{1}{2\gamma}\left\langle \bar{\theta} - \theta, 2\hat{\theta} - \bar{\theta} - \theta\right\rangle + \left\langle \bar{\theta} - \hat{\theta}, H - \theta^{-1}\right\rangle \\ &= \frac{1}{2}\left(\frac{1}{\gamma} - \frac{1}{\psi^2}\right)\left\|\theta - \hat{\theta}\right\|_{\mathsf{F}}^2 - \frac{1}{2\gamma}\left\|\bar{\theta} - \hat{\theta}\right\|_{\mathsf{F}}^2 + \left\langle \bar{\theta} - \hat{\theta}, H - \theta^{-1}\right\rangle. \end{aligned} \tag{59}$$

Since $\phi_\alpha(\bar{\theta}) \geq \phi_\alpha(\hat{\theta})$, we conclude that

$$\left\|\bar{\theta} - \hat{\theta}\right\|_{\mathsf{F}}^2 \leq 2\gamma\left(\phi_\alpha(\bar{\theta}) - \phi_\alpha(\hat{\theta})\right) + \left\|\bar{\theta} - \hat{\theta}\right\|_{\mathsf{F}}^2 \leq \left(1 - \frac{\gamma}{\psi^2}\right)\left\|\theta - \hat{\theta}\right\|_{\mathsf{F}}^2 + 2\gamma\left\langle \bar{\theta} - \hat{\theta}, H - \theta^{-1}\right\rangle,$$

as claimed. ■

Lemma 15 *Take $\ell > 0$, and $\theta \in \mathcal{M}_+(\ell)$. Let $z_{1:N} \overset{i.i.d.}{\sim} \boldsymbol{N}(0, \theta^{-1})$, and set $G_N \overset{\text{def}}{=} N^{-1} \sum_{i=1}^N z_i z_i'$. Then*

$$\mathbb{E}\left[\|G_N - \theta^{-1}\|_{\mathsf{F}}^2\right] \leq \frac{p + p^2}{N\ell^2},$$

and for any $\delta > 0$ such that $\ell\delta \leq 4$,

$$\mathbb{P}\left(\|G_N - \theta^{-1}\|_\infty > \delta\right) \leq 4p^2 \exp\left(-\min(1, \ell^2\delta^2/16)N\right).$$

Proof

$$\begin{aligned}\mathbb{E}\left[\|G_N - \theta^{-1}\|_{\mathsf{F}}^2\right] &= \sum_{j,k} \mathbb{E}\left[\left(\frac{1}{N}\sum_{i=1}^N (z_i z_i')_{j,k} - \theta^{-1}_{j,k}\right)^2\right] = \frac{1}{N}\sum_{j,k}\mathbb{E}\left[\left((z_1 z_1')_{j,k} - \theta^{-1}_{j,k}\right)^2\right] \\ &= \frac{1}{N}\sum_{l,k}\left(\theta^{-1}_{j,j}\theta^{-1}_{k,k} + (\theta^{-1}_{j,k})^2\right) = \frac{1}{N}\left(\mathsf{Tr}(\theta^{-1})^2 + \|\theta^{-1}\|_{\mathsf{F}}^2\right) \leq \frac{1}{N}\left(\left(\frac{p}{\ell}\right)^2 + \frac{p}{\ell^2}\right).\end{aligned}$$

For the exponential bound, we reduce the problem to an exponential bound for chi-squared distributions, and apply the following corollary of Lemma 1 of Laurent and Massart (2000). Let $W_{1:N} \overset{i.i.d.}{\sim} \chi^2_1$, the chi-square distribution with one degree of freedom. For any $x \in [0, 1]$,

$$\mathbb{P}\left[\left|\sum_{k=1}^N (W_k - 1)\right| > 4\sqrt{x}N\right] \leq 2e^{-Nx}. \tag{60}$$

For $1 \leq i, j \leq p$, arbitrary, set $Z_{ij}^{(k)} = z_{k,i} z_{k,j}$, and $\sigma_{ij} = \theta^{-1}_{ij}$. Suppose that $i \neq j$. It is easy to check that

$$\begin{aligned}\sum_{k=1}^N \left[Z_{ij}^{(k)} - \sigma_{ij}\right] = \frac{1}{4}\sum_{k=1}^N &\left[(z_{k,i} + z_{k,j})^2 - \sigma_{ii} - \sigma_{jj} - 2\sigma_{ij}\right] \\ &- \frac{1}{4}\sum_{k=1}^N \left[(z_{k,i} - z_{k,j})^2 - \sigma_{ii} - \sigma_{jj} + 2\sigma_{ij}\right].\end{aligned}$$

Notice that $z_{k,i} + z_{k,j} \sim \mathbf{N}(0, \sigma_{ii} + \sigma_{jj} + 2\sigma_{ij})$, and $z_{k,i} - z_{k,j} \sim \mathbf{N}(0, \sigma_{ii} + \sigma_{jj} - 2\sigma_{ij})$. It follows that for all $x \geq 0$,

$$\begin{aligned}\mathbb{P}\left[\left|\sum_{k=1}^N \left[Z_{ij}^{(k)} - \sigma_{ij}\right]\right| > x\right] &\leq \mathbb{P}\left[\left|\sum_{k=1}^N (W_k - 1)\right| > \frac{2x}{\sigma_{ii} + \sigma_{jj} + 2\sigma_{ij}}\right] \\ &\quad + 2\mathbb{P}\left[\left|\sum_{k=1}^N (W_k - 1)\right| > \frac{2x}{\sigma_{ii} + \sigma_{jj} - \sigma_{ij}}\right], \\ &\leq 2\mathbb{P}\left[\left|\sum_{k=1}^N (W_k - 1)\right| > \ell x\right].\end{aligned}$$

where $W_{1:N} \overset{i.i.d.}{\sim} \chi^2_1$, the chi-square distribution with one degree of freedom. The last inequality uses the fact that $\sigma_{ii} + \sigma_{jj} + 2\sigma_{ij} = u'\theta^{-1}u \leq \frac{1}{\ell}\|u\|^2 \leq \frac{2}{\ell}$, where u is the vector with

1 on components i and j and zero everywhere else (similarly for $\sigma_{ii} + \sigma_{jj} - 2\sigma_{ij}$ by putting -1 on the j-th entry). Then we apply (60) to obtain

$$\mathbb{P}\left[\left|\sum_{k=1}^{N}\left[Z_{ij}^{(k)} - \sigma_{ij}\right]\right| > N\delta\right] \leq 4e^{-\min(1,\ell^2\delta^2/16)N}.$$

When $i = j$, the bound $\mathbb{P}\left[\left|\sum_{k=1}^{N}\left[Z_{ij}^{(k)} - \sigma_{ij}\right]\right| > x\right] \leq \mathbb{P}\left[\left|\sum_{k=1}^{N}(W_k - 1)\right| > \ell x\right]$ is straightforward. The lemma follows from a standard union-sum argument. ■