Biometrika (2017), **104**, 2, *pp.* 379–395
Printed in Great Britain
doi: 10.1093/biomet/asx003
Advance Access publication 17 April 2017

Sparse Gaussian graphical model estimation via alternating minimization

BY ONKAR DALAL

Institute for Computational & Mathematical Engineering, Stanford University, 475 Via Ortega, Stanford, California 94305, U.S.A.
onkar.dalal@gmail.com

AND BALA RAJARATNAM

Department of Statistics, Math Sciences Bldg 4118, University of California, Davis, One Shields Avenue, Davis, California 95616, U.S.A.
brajaratnam01@gmail.com

SUMMARY

Several methods have recently been proposed for estimating sparse Gaussian graphical models using ℓ_1-regularization on the inverse covariance or precision matrix. Despite recent advances, contemporary applications require even faster methods to handle ill-conditioned high-dimensional datasets. In this paper, we propose a new method for solving the sparse inverse covariance estimation problem using the alternating minimization algorithm, which effectively works as a proximal gradient algorithm on the dual problem. Our approach has several advantages: it is faster than state-of-the-art algorithms by many orders of magnitude; its global linear convergence has been rigorously demonstrated, underscoring its good theoretical properties; it facilitates additional constraints on pairwise or marginal relationships between feature pairs based on domain-specific knowledge; and it is better at handling extremely ill-conditioned problems. Our algorithm is shown to be more accurate and faster on simulated and real datasets.

Some key words: Graphical model; Projected gradient method; Sparse inverse covariance matrix.

1. INTRODUCTION

1·1. *Gaussian graphical models*

We consider sparse inverse covariance estimation for undirected Gaussian graphical models using ℓ_1-regularized maximum likelihood estimation. Given n independent realizations of a p-dimensional Gaussian random vector, with population and sample covariance matrices denoted by Σ and S, respectively, the goal is to estimate $\hat{\Sigma} \in \mathcal{S}_+^p$ such that $\hat{\Sigma}^{-1}$ is sparse, where $\mathcal{S}_+^p$ denotes the cone of p-dimensional positive-definite matrices. For the multivariate Gaussian distribution, sparsity in the inverse covariance is related to conditional independence among the random variables. Two variables X_i and X_j are conditionally independent given all other variables if and only if the corresponding entry in the inverse covariance matrix is zero, i.e., $(\Sigma^{-1})_{ij} = 0$. Since maximum likelihood estimation is formulated in terms of the inverse covariance $\Omega = \Sigma^{-1}$, adding an ℓ_1-regularization with a penalty parameter λ induces sparsity in the estimated inverse

covariance matrix. The regularized maximum likelihood estimation problem in Ω is

$$\underset{\Omega \in \mathcal{S}_+^p}{\text{minimize}} \; -\log\det\Omega + \langle S, \Omega\rangle + \lambda \sum_{i,j=1}^{p} |\Omega_{ij}|, \tag{1}$$

where $\langle S, \Omega\rangle = \sum_{i,j=1}^{p} S_{ij}\Omega_{ij}$. The minimization problem (1) is convex in Ω. For $\lambda = 0$, the solution of (1), $\hat{\Omega}^{(0)} = (\hat{\Sigma}^{(0)})^{-1} = S^{-1}$, is an unbiased estimator of the covariance matrix, but it is not well-defined for $n < p$. However, for any $\lambda > 0$ the estimate is well-defined for any n, with the sparsity in $\hat{\Omega}^{(\lambda)}$ increasing with λ. The dual of problem (1) can be formulated as

$$\begin{aligned} &\underset{\Gamma \in \mathcal{S}_+^p}{\text{minimize}} \quad -\log\det\Gamma - p \\ &\text{subject to} \quad |\Gamma_{i,j} - S_{i,j}| \leqslant \lambda \quad (i = 1, \ldots, p; j = 1, \ldots, p). \end{aligned} \tag{2}$$

In this paper, we propose a new graphical alternating minimization algorithm for graphical model selection that uses the alternating minimization algorithm of Tseng (1991), which has been shown to solve the dual problem with the forward-backward splitting method from Rockafellar (1976) and also proven to converge linearly for strongly monotone operators. The iterates of our algorithm always maintain a feasible dual estimate for the covariance matrix $\hat{\Sigma}$. This property is useful in solving large problems when there are time constraints and early termination is necessary. In addition, our algorithm is very fast for very small values of λ, a situation where other state-of-the-art methods converge slowly because the inverse covariance estimates have large condition numbers. Such problems are common in high dimensions when the sample size is small.

In many modern applications, information on the covariance structure is known but not always leveraged. This knowledge is often available for the covariance matrix rather than the inverse covariance matrix, since the sample covariance matrix S is still computable even in the sample-starved setting, although it is only positive semidefinite. In genomics it is well known that the majority of correlations between pairs of genes are close to zero, with a few hub genes being correlated with many others (Barabasi & Oltvai, 2004). In the environmental sciences, correlations between distant points of spatial fields are often less pronounced (Guillot et al., 2015). As a by-product of our analysis, we demonstrate how to modify our algorithm to incorporate any available domain-specific information about covariances.

1·2. *Background*

In this subsection, we briefly describe algorithms that have been proposed to solve the primal problem (1) and its dual (2). These algorithms can be classified into block coordinate descent methods and proximal methods. Our algorithm belongs to the second class.

A dual block coordinate descent which solves a box-constrained quadratic program for each coordinate was proposed by Banerjee et al. (2008). Friedman et al. (2008) introduced the graphical lasso algorithm, which solves the dual of the box-constrained quadratic program, equivalent to solving a lasso problem. Mazumder & Hastie (2012) developed the primal graphical lasso algorithm, which improves upon the graphical lasso by applying similar techniques to the primal problem. These algorithms take a coordinate descent step for each row and corresponding column, and iterate over the row-column pairs until convergence. They have been shown to converge to the optimal primal or dual solution. However, theoretical guarantees for their convergence rates have not yet been established.

Lu (2009, 2010), d'Aspremont et al. (2008) and Scheinberg et al. (2010) have used Nesterov's smooth approximation method and its variations to develop algorithms that achieve ϵ-convergence

in $O(\epsilon^{-1})$ and $O(\epsilon^{-1/2})$ iterations. Hsieh et al. (2011) proposed a proximal Newton algorithm and proved its local quadratic convergence. This algorithm was later generalized by Lee et al. (2012). Guillot et al. (2012) proposed a proximal gradient algorithm and proved its global linear convergence. Both of these proximal algorithms use the inverse covariance matrix as their operating variable and outperform the graphical lasso methods in numerical experiments; they have respective advantages in various sparsity and condition number regimes.

2. Graphical alternating minimization algorithm

In this section, we describe our approach to solving (1) by using an alternating minimization algorithm. We rewrite (1) in composite form as

$$\begin{aligned} &\underset{\Omega\in\mathcal{S}^p_+,\Phi\in\mathcal{S}^p}{\text{minimize}} \quad -\log\det\Omega + \langle S,\Phi\rangle + \lambda\sum_{i,j=1}^{p}|\Phi_{ij}| \\ &\text{subject to} \quad \Phi-\Omega=0, \end{aligned} \tag{3}$$

where Φ is a dummy variable and $\mathcal{S}^p$ is the set of p-dimensional symmetric matrices. Each alternating minimization iteration updates the primal variables Ω and Φ and the dual variable Γ sequentially to Ω_+, Φ_+ and Γ_+ according to

$$\Omega_+ = \underset{\Omega\in\mathcal{S}^p_+}{\arg\min}\ -\log\det\Omega - \langle\Gamma,-\Omega\rangle, \tag{4}$$

$$\Phi_+ = \underset{\Phi\in\mathcal{S}^p}{\arg\min}\ \langle S,\Phi\rangle + \lambda\sum_{i,j=1}^{p}|\Phi_{ij}| - \langle\Gamma,\Phi\rangle + \frac{\tau}{2}\|\Omega_+-\Phi\|_{\mathrm{F}}^2, \tag{5}$$

$$\Gamma_+ = \Gamma + \tau(\Omega_+-\Phi_+), \tag{6}$$

where τ is the step size and $\|\Theta\|_{\mathrm{F}} = \langle\Theta,\Theta\rangle^{1/2}$ is the Frobenius norm of Θ. The optimization problems (4) and (5) can be solved analytically. The respective optimality conditions are

$$0 = -\Omega_+^{-1} + \Gamma, \tag{7}$$

$$0 = S + \lambda\,\mathrm{sign}(\Phi_+) - \Gamma + \tau(\Phi_+-\Omega_+). \tag{8}$$

The optimality condition (8) can be rewritten using $\mathcal{S}_\lambda(x) = \mathrm{sign}(x)[\max\{(|x|-\lambda),0\}]$ as $\tau\Phi_+ = \mathcal{S}_\lambda(\tau\Omega_+ + \Gamma - S)$ where $\mathcal{S}_\lambda(x)$, the soft-thresholding operator, is applied entrywise. Substituting Ω_+ and Φ_+ from (7) and (8) into (6) yields a one-step update for Γ_+ in terms of Γ,

$$\Gamma_+ = \mathcal{C}_\lambda(\Gamma - S + \tau\Gamma^{-1}) + S, \tag{9}$$

where $\mathcal{C}_\lambda(x) = \min\{\max(x,-\lambda),\lambda\}$, the clip function, is related to the soft-thresholding function via the identity $x = \mathcal{S}_\lambda(x) + \mathcal{C}_\lambda(x)$. The details are given in Algorithm 1, which is terminated when the duality gap

$$\Delta_{\mathrm{opt}} = -\log\det\Gamma_k - p - \log\det\Phi_k + \langle S,\Phi_k\rangle + \lambda\sum_{i,j=1}^{p}\left|(\Phi_k)_{ij}\right|$$

is reduced below a certain tolerance, ϵ_{opt}. An alternative tolerance criterion, ϵ_{prim}, can be imposed on the progress of Γ_k iterates, $\Delta_{\text{prim}} = (\|\Gamma_{k+1} - \Gamma_k\|_{\text{F}})\|\Gamma_k\|_{\text{F}}^{-1}$. This quantity relates to the primal constraint violation $\|\Omega_k - \Phi_k\|_{\text{F}} = \tau^{-1}\|\Gamma_{k+1} - \Gamma_k\|_{\text{F}}$ and indirectly imposes primal feasibility. The step size τ_k for each iteration is chosen by backtracking line search such that the next iterate Γ_{k+1} is positive definite and satisfies the sufficient descent condition

$$-\log\det\Gamma_{k+1} \leqslant -\log\det\Gamma_k + \langle\Gamma_{k+1} - \Gamma_k, \Gamma_k^{-1}\rangle + (2\tau)^{-1}\|\Gamma_{k+1} - \Gamma_k\|_{\text{F}}^2,$$

where the right-hand side is a local quadratic approximation of the dual objective around Γ_k. Further details about the choice of step size are given in the Supplementary Material.

Algorithm 1. Graphical alternating minimization algorithm.

Input S, λ, Γ_0
Set $k = 0$, $\Delta_{\text{opt}} = 2\epsilon_{\text{opt}}$
While $\Delta_{\text{opt}} > \epsilon_{\text{opt}}$
  $\Omega_{k+1} \leftarrow \Gamma_k^{-1}$
  Compute τ_k using line search
  $\Phi_{k+1} \leftarrow \tau_k^{-1}\mathcal{S}_\lambda(\tau_k\Omega_{k+1} + \Gamma_k - S)$
  $\Gamma_{k+1} \leftarrow \Gamma_k + \tau_k(\Omega_{k+1} - \Phi_{k+1})$
  $\Delta_{\text{opt}} \leftarrow -\log\det\Gamma_{k+1} - p - \log\det\Omega_{k+1} + \langle S, \Omega_{k+1}\rangle + \lambda\sum_{i,j=1}^p |\Omega_{ij}|$
Output Γ_{k+1}

3. Convergence analysis

3·1. *Strong convexity and linear convergence*

In this subsection, we prove theoretical results on the global convergence of our algorithm. We first show strong convexity and Lipschitz continuity of the gradient of the objective function over any compact domain. Next we show that the iterates of our algorithm belong to a compact domain bounded away from the boundary of the positive-definite cone S_+^p. Finally we prove linear convergence of our algorithm. All proofs are given in the Appendix.

The optimal solution Ω_* of (1) satisfies the optimality condition

$$0 = -\Omega_*^{-1} + S + \lambda\,\text{sign}(\Omega_*). \tag{10}$$

This implies that for an optimal solution Ω_*, the element $(\Omega_*)_{ij}$ is zero if and only if $|S_{ij} - (\Omega_*^{-1})_{ij}| \leqslant \lambda_{ij}$, and $(\Omega_*)_{ij} \neq 0$ if and only if $S_{ij} - (\Omega_*^{-1})_{ij} = -\lambda_{ij}\,\text{sign}\{(\Omega_*)_{ij}\}$. Next, we highlight the relation between the optimal solution Ω_* and the fixed point Γ_* of our iterations.

LEMMA 1. *The optimal solution* Ω_* *of* (1) *satisfies* (10) *if and only if the inverse* $\Gamma_* = \Omega_*^{-1}$ *is a fixed point of the algorithm in* (9)*, i.e.,*

$$\Gamma_* = \mathcal{C}_\lambda(\tau\Gamma_*^{-1} + \Gamma_* - S) + S. \tag{11}$$

The existence of a fixed point satisfying (11) will allow us to exploit arguments similar to those from Guillot et al. (2012) to prove global linear convergence.

The gradient of the log det function $\nabla \log\det\Gamma = \Gamma^{-1}$ can be shown to be Lipschitz-continuous over any compact domain

$$\mathcal{D} = \{\Gamma : 0 \prec \alpha I \preceq \Gamma \preceq \beta I\} \quad (0 < \alpha < \beta), \tag{12}$$

where $\Psi \preceq \Theta$ means that $\Theta - \Psi$ is positive semidefinite.

LEMMA 2 (Guillot et al., 2012, Lemma 2). *For $\Gamma_1, \Gamma_2 \in \mathcal{S}_+^p$, the gradient of the* log det *function $\nabla \log\det\Gamma = \Gamma^{-1}$ satisfies*

$$\beta^{-2}\|\Gamma_1 - \Gamma_2\|_2 \leqslant \|\Gamma_1^{-1} - \Gamma_2^{-1}\|_2 \leqslant \alpha^{-2}\|\Gamma_1 - \Gamma_2\|_2,$$

where $\alpha = \min\{\lambda_{\min}(\Gamma_1), \lambda_{\min}(\Gamma_2)\}$ and $\beta = \max\{\lambda_{\max}(\Gamma_1), \lambda_{\max}(\Gamma_2)\}$.

The Hessian $\nabla^2 \log\det\Gamma = -\Gamma^{-1} \otimes \Gamma^{-1}$ is strictly positive definite over $\mathcal{D}$ defined in (12), so the log det function is strongly convex over $\mathcal{D}$. Using Lemma 2, we prove the following result, which is key to establishing an upper bound on iterates of our algorithm as well as its global linear convergence.

LEMMA 3. *Let Γ_+ and Γ be iterates from (9), and let Γ_* be a fixed point as in (11). Then*

$$\|\Gamma_+ - \Gamma_*\|_{\mathrm{F}} \leqslant \max\big(|1 - \tau\alpha^{-2}|, |1 - \tau\beta^{-2}|\big)\|\Gamma - \Gamma_*\|_{\mathrm{F}},$$

where $\alpha = \min\{\lambda_{\min}(\Gamma), \lambda_{\min}(\Gamma_*)\}$ and $\beta = \max\{\lambda_{\max}(\Gamma), \lambda_{\max}(\Gamma_*)\}$.

Next, we show that the iterates of our algorithm belong to the bounded compact set $\mathcal{D}$ defined in (12) and give explicit values for α and β.

LEMMA 4 (Guillot et al., 2012, Lemma 8). *Let $0 \prec \alpha_l I \preceq \Gamma_l$ for $l = 0, \ldots, k$, and suppose the step size is such that $\tau_k < \alpha_k^2 < \beta^2$ for $\beta = \|\Gamma_0 - \Gamma_*\|_{\mathrm{F}} + \|\Gamma_*\|_2$. Then the next iterate Γ_{k+1} satisfies*

$$\Gamma_{k+1} \preceq \beta I, \tag{13}$$

and hence, by induction on k, $\Gamma_k \preceq \beta I$ for all k.

Lemma 4 establishes an upper bound on Γ_{k+1}, assuming a strictly positive lower bound α_l for $l = 0, \ldots, k$. Based on Lemma 3 in Hsieh et al. (2011), we show that the iterates of our algorithm are bounded away from the boundary of the positive-definite cone and satisfy $\alpha I \preceq \Gamma_k$ with some universal $\alpha > 0$ for all k. To prove the lower bound α, we define a level set of $-\log\det$,

$$\mathcal{U} = \big\{\Gamma : 0 \preceq \Gamma \prec \beta I,\ -\log\det\Gamma < -\log\det\Gamma_0\big\}. \tag{14}$$

LEMMA 5 (Hsieh et al., 2011, Lemma 3). *The set $\mathcal{U}$ in (14) satisfies $\alpha I \preceq \Gamma$ with $\alpha = \lambda^p \beta^{-(p-1)}$, where β comes from the upper bound in (13) of Lemma 4.*

The α and β bounds on the eigenvalues of Γ_k are theoretical bounds which facilitate the global convergence analysis. They ensure that the iterates remain in a compact domain. In practical problems, the iterates remain well within the interior of this compact domain. We now show global linear convergence of our algorithm using Lemmas 4 and 5.

THEOREM 1. *The iterates Γ_k of Algorithm 1 satisfy $\alpha I \preceq \Gamma_k \preceq \beta I$ and*

$$\|\Gamma_{k+1} - \Gamma_*\|_{\mathrm{F}} \leqslant \max\big(|1 - \tau_k \alpha^{-2}|, |1 - \tau_k \beta^{-2}|\big)\|\Gamma_k - \Gamma_*\|_{\mathrm{F}} \quad (k = 0, 1, \ldots).$$

More specifically, for step size $\tau_k < \alpha^2$,

$$\|\Gamma_{k+1} - \Gamma_*\|_{\mathrm{F}} \leqslant \gamma \|\Gamma_k - \Gamma_*\|_{\mathrm{F}},$$

where $\gamma < 1$ and the iterates converge to an ϵ-optimal solution Γ_* in $O(\log \epsilon)$ iterations.

The maximum step size α^2 provided by α from Lemma 5 is very conservative. Significantly better performance can be achieved by using other heuristics; see the Supplementary Material.

3·2. *Connections to the proximal algorithms*

In Guillot et al. (2012), (1) is solved using the forward-backward splitting method, and global linear convergence of the algorithm is established using strong convexity of the log det function and Lipschitz continuity of the gradient over a compact domain. Given an initial point, the subsequent iterates are shown to remain in a compact domain and hence the proximal gradient method converges linearly, as shown in Rockafellar (1976). Our algorithm is closely related to the proximal gradient algorithm, and as expected there are some parallels in their convergence analyses and choices of step size.

Hsieh et al. (2012) proposed a divide-and-conquer implementation which partitions a large problem (1) into multiple smaller subproblems of the same form. The subproblems are solved using the proximal Newton algorithm (Hsieh et al., 2011), and the solutions are combined to get an approximate solution of the original problem. Our algorithm is comparable to the proximal Newton algorithm and can replace it in the divide-and-conquer implementation to solve large problems.

The proximal methods of d'Aspremont et al. (2008), Lu (2009, 2010) and Scheinberg et al. (2010) attain an ϵ-optimal solution in $O(\epsilon^{-1})$ or $O(\epsilon^{-1/2})$ iterations, as compared to $O(\log \epsilon)$ iterations required by our algorithm. These methods are substantially slower than the proximal gradient (Guillot et al., 2012) and proximal Newton (Hsieh et al., 2011) algorithms and are therefore not considered state-of-the-art. In §5, our algorithm is shown to outperform the state-of-the-art proximal algorithms.

3·3. *Choice of penalty parameter and statistical consistency*

Various methods have been proposed for choosing the regularization or the penalty parameter λ. Some of the popular approaches to selecting λ are based on: crossvalidation; achieving some desired level of sparsity, such as 3% edge density; Bayesian information-type criteria (Khare et al., 2015); or controlling the probability of false detection of edges, i.e., a certain predetermined level of error control (Banerjee et al., 2008). An important question concerns asymptotic recovery of the underlying graph when the data are generated from a sparse Gaussian graphical model, i.e., model selection consistency. Since our algorithm yields an ℓ_1-regularized maximum likelihood estimate, we can use established statistical theory to assert consistency of the estimates generated by our algorithm. Banerjee et al. (2008) showed that provided the dimension p is $O(n^\gamma)$ for some $\gamma > 0$, ℓ_1-regularized maximum likelihood estimation recovers the underlying graph with probability tending to 1. More specifically, for a value of $\alpha \in [0, 1]$, if the penalty parameter λ is of the form

$$\lambda(\alpha) \equiv \max_{i>j} (\hat{\sigma}_i \hat{\sigma}_j) \big[t_{n-2}(\bar{\alpha}) \{ n - 2 + t^2_{n-2}(\bar{\alpha}) \}^{-1/2} \big]$$

where $\bar{\alpha} = \alpha(2p^2)^{-1}$, $\hat{\sigma}_i$ denotes the sample variance of variable i, and t_{n-2} denotes the 100 $(1-\alpha)$th percentile of the Student t-distribution with $n-2$ degrees of freedom, then the probability that the estimated connectivity component of one or more nodes is not contained in the true connectivity component is less than α. Moreover, the penalty parameter λ must decay to zero and be at least as large as $C(\log p)^{1/2}n^{-1/2}$.

4. Generalized graphical alternating minimization algorithm

4·1. *Generalization of the dual constraint*

The first generalization modifies the dual problem (2) to include explicit constraints on the bivariate covariances in Σ. As discussed in § 1, scenarios where domain-specific information regarding correlation structure is available arise in numerous applications. Such knowledge of the correlation structure is more easily found in bivariate pairwise or marginal relationships, since such quantities can be calculated straightforwardly from the sample covariance matrix, in contrast to the sample inverse covariance matrix, which is undefined when $n < p$. Since the operating variable for our algorithm is the estimate of the covariance matrix $\hat{\Sigma} = \Gamma$, it can easily incorporate domain-specific knowledge by adding constraints on the covariance matrix. Specifically, the constraint set on Σ in (2) can be generalized to any convex constraint $\mathcal{D}$, extending the formulation to a more general form

$$\begin{aligned} &\underset{\Gamma \in \mathcal{S}_+^p}{\text{minimize}} \quad -\log\det\Gamma, \\ &\text{subject to} \quad \Gamma \in \mathcal{D}. \end{aligned} \tag{15}$$

In this subsection, we illustrate the extension of our algorithm to $\mathcal{D}$ defined by arbitrary bound constraints on the bivariate covariances:

$$\begin{aligned} &\underset{\Gamma \in \mathcal{S}_+^p}{\text{minimize}} \quad -\log\det\Gamma, \\ &\text{subject to} \quad -l_{ij} \leqslant \Gamma_{ij} \leqslant u_{ij} \quad (i = 1, \ldots, p; j = 1, \ldots, p). \end{aligned} \tag{16}$$

Here, each bivariate covariance σ_{ij} is allowed to vary in the interval $[l_{ij}, u_{ij}]$, which is a generalization of the interval $[s_{ij}-\lambda, s_{ij}+\lambda]$ for the dual problem (2). These modified constraints break away from the maximum likelihood framework and allow regularization based on domain-specific knowledge. The corresponding primal problem formulated in terms of the inverse covariance matrix is

$$\underset{\Omega \in \mathcal{S}_+^p}{\text{minimize}} \quad -\log\det\Omega + \langle \bar{S}, \Omega \rangle + \sum_{i,j=1}^{p} \bar{\lambda}_{ij}|\Omega_{ij}|, \tag{17}$$

where $\bar{S}$ is a modified sample covariance and the ℓ_1-regularization uses a more flexible penalty $\bar{\lambda}_{ij}$ for Ω_{ij}. These are related to the lower and upper bounds of the dual formulation via

$$\bar{S}_{ij} = (l_{ij} + u_{ij})/2, \quad \bar{\lambda}_{ij} = (u_{ij} - l_{ij})/2.$$

Unlike the maximum likelihood estimation formulation, where $S \succeq 0$ provided a feasible point $S + \lambda I$ for any $\lambda > 0$, the generalized problem is no longer guaranteed to be feasible for arbitrary

choices of the parameters l_{ij} and u_{ij}. For problem (15), the algorithm update for Γ is

$$\Gamma_+ = \Pi_{\mathcal{D}}(\Gamma + \tau\Gamma^{-1}), \tag{18}$$

where $\Pi_{\mathcal{D}}$ is the projection onto the convex constraint set $\mathcal{D}$. The simplified projection for the bound constraints defined in (16) is

$$(\Gamma_+)_{ij} = \min\Big[\max\Big\{\big(\Gamma + \tau\Gamma^{-1}\big)_{ij}, l_{ij}\Big\}, u_{ij}\Big].$$

The method of choosing the step size remains the same as for the standard graphical alternating minimization algorithm, and the linear convergence results hold.

In the case of simple bound constraints, the constraints on the covariance matrix can be translated to a corresponding primal problem (17), so primal methods can also be useful in this context. However, a more complicated constraint might not be easily translated to the primal problem, and so the advantage of using a fast coordinatewise descent algorithm for the lasso subproblems may be lost. In contrast, our algorithm and framework allow additional constraints to be included while maintaining the sparsity in the inverse covariance matrix, thus enriching the types of graphical modelling that can be achieved.

4·2. *Generalization of the primal penalty*

The second generalization modifies the primal problem (1) by generalizing the ℓ_1-regularization on the inverse covariance matrix Ω. In many applications the underlying graph structure is known to be symmetric, i.e., the edge weights or the partial correlations assume equal values either naturally or by design (Gehrmann & Lauritzen, 2012). In such settings, it is often necessary and useful to estimate a partial correlation graph that respects these symmetries, also known as equisparsity. Such additional constraints on graphical models are of both theoretical and applied interest due to the added layer of regularization and the structure it provides. This type of information about the structure of the underlying graph can be incorporated by using a regularization term in the form of a symmetric linear transform. Consider a linear transform T with conjugate transpose T^*. A generalized primal problem formulated by applying an ℓ_1-regularization on $T(\Omega)$ instead of Ω is

$$\underset{\Omega \in \mathcal{S}_+^p}{\text{minimize}} \; -\log\det\Omega + \langle S, \Omega\rangle + \lambda \sum_{i,j=1}^{p} |T(\Omega)_{ij}|. \tag{19}$$

Choosing $T(\Omega) = \Omega - P\Omega P^{\mathrm{T}}$ for a permutation matrix P, the above formulation enforces equisparsity or linear constraints on the elements of Ω, the inverse covariance matrix. As in (3), problem (19) can be written in composite form using a constraint $T(\Omega) - \Phi = 0$, and is solved using alternating minimization updates

$$\Omega_+ = \underset{\Omega \in \mathcal{S}_+^p}{\arg\min} \; -\log\det\Omega + \langle S, \Omega\rangle - \langle \Gamma, -T(\Omega)\rangle, \tag{20}$$

$$\Phi_+ = \underset{\Phi \in \mathcal{S}^p}{\arg\min} \; \lambda \sum_{i,j=1}^{p} |\Phi_{ij}| - \langle \Gamma, \Phi\rangle + \frac{\tau}{2}\|\Phi - T(\Omega)\|_{\mathrm{F}}^2, \tag{21}$$

$$\Gamma_+ = \Gamma + \tau\{T(\Omega_+) - \Phi_+\}. \tag{22}$$

The optimality conditions for (20) and (21) are

$$0 = -\Omega_+^{-1} + S + T^*(\Gamma), \tag{23}$$

$$0 = \lambda\,\mathrm{sign}(\Phi_+) - \Gamma + \tau\{\Phi_+ - T(\Omega_+)\}. \tag{24}$$

Substituting the optimality conditions (23) and (24) for Ω_+ and Φ_+ into (22) gives a direct update for Γ_+ in terms of Γ:

$$\Gamma_+ = \mathcal{C}_\lambda\big(\Gamma + \tau T\big[\{S + T^*(\Gamma)\}^{-1}\big]\big). \tag{25}$$

4·3. *Convergence analysis*

The proofs from § 3 can be modified to apply to the generalizations described in §§ 4·1 and 4·2. Lemma 5 holds for the generalized iterates which belong to a subset of $\mathcal{U}$. Lemma 2 can be applied directly, and Lemmas 3 and 4 can be modified for the generalized iterates by using nonexpansiveness of the proximal operator for any convex set $\mathcal{D}$.

LEMMA 6. *The proximal operator* $\mathrm{prox}_{\mathcal{D}}$ *for an indicator function of a convex set* $\mathcal{D}$*, defined by*

$$\mathrm{prox}_{\mathcal{D}}(x) = \underset{z\in\mathcal{D}}{\arg\min}\ \|z - x\|^2,$$

satisfies $\|\mathrm{prox}_{\mathcal{D}}(x) - \mathrm{prox}_{\mathcal{D}}(y)\|_2 \leqslant \|x - y\|_2$.

Lemma 6 is used to show global linear convergence of the iterates of our generalized algorithm.

THEOREM 2. *The new iterate* Γ_+ *of our generalized algorithm from* (18) *and* (25) *satisfies*

$$\|\Gamma_+ - \Gamma_*\|_{\mathrm{F}} \leqslant \gamma\|\Gamma - \Gamma_*\|_{\mathrm{F}}$$

for some $\gamma < 1$*, and the iterates converge to an* ϵ*-optimal solution* Γ_* *in* $O(\log\epsilon)$ *iterations.*

5. NUMERICAL EXPERIMENTS

5·1. *Linear convergence*

We begin by demonstrating linear convergence of our algorithm and how it varies with λ and $\kappa = \lambda_{\max}(\hat{\Sigma}^*)/\lambda_{\min}(\hat{\Sigma}^*)$, the condition number of the solution $\hat{\Sigma}^*$, computed after the algorithm has converged and not as a function of the iteration number. Figure 1 shows the convergence of the duality gap as the number of iterations increases. The algorithm takes longer to converge as the regularization parameter λ is reduced, explained by the increasing values of κ. The simulated data for Fig. 1 were generated using the method described in § 5·3.

5·2. *Timing comparisons with the proximal algorithms*

We compare the convergence times of our algorithm, the proximal Newton algorithm (Hsieh et al., 2011) and the proximal gradient algorithm (Guillot et al., 2012) in solving (1) for synthetic and real datasets. Similar to the proximal algorithms, our algorithm is implemented using the programming language C with a Matlab software wrapper. The three algorithms use a linear algebra package for computing Cholesky factors and matrix inversion, which leverages multiple processors when available. We use a 32-bit version of Ubuntu 12.10 with an Intel Core i7 870 processor having 4 cores and 8 GB memory. The times reported in this section are wall-clock timings,

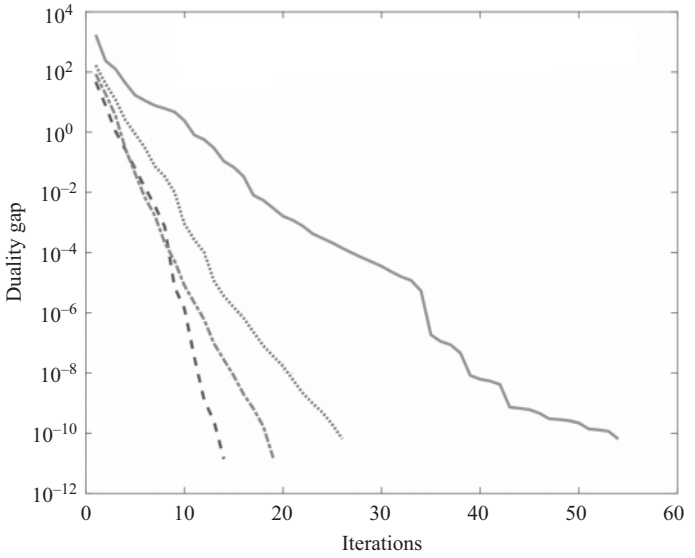

Fig. 1. Convergence of our algorithm for: $\lambda = 0{\cdot}1$ and $\kappa = 4{\cdot}2$ (solid); $\lambda = 0{\cdot}3$ and $\kappa = 2{\cdot}6$ (dotted); $\lambda = 0{\cdot}5$ and $\kappa = 2{\cdot}0$ (dot-dash); and $\lambda = 0{\cdot}75$ and $\kappa = 1{\cdot}7$ (dashed).

not processor timings. Our algorithm and the proximal gradient algorithm are terminated by a tolerance condition on the duality gap. However, termination of the proximal Newton algorithm, as implemented by Hsieh et al. (2011), is controlled by the tolerance on a subgradient condition only. The problems are solved using the proximal Newton algorithm by setting $\epsilon_{\text{tol}} = 10^{-10}$, and the corresponding duality gap achieved by the proximal Newton algorithm is then used for the other two algorithms.

5·3. *Simulated datasets*

First we conducted comparisons on synthetic data generated using the procedure in Lu (2010). For a given problem size p, the underlying inverse covariance matrix $\Omega = \Sigma^{-1}$ is generated by independently choosing the off-diagonal entries from a uniform distribution over the interval $[-1, 1]$. To obtain a desired sparsity level s, the fraction of nonzero entries in $\hat{\Omega}$, we use the method of Guillot et al. (2012), setting the off-diagonal entries to zero with probability s. A multiple of the identity matrix is added to this matrix to adjust the smallest eigenvalue of $\Omega = \Sigma^{-1}$ to 1. This procedure ensures that Ω is positive definite, sparse and well-conditioned. Finally, datasets of n independent samples are generated from the normal distribution $N_p(0, \Omega^{-1})$. For each Ω, sparsity levels of 0·03 and 0·15 are considered, and sample sizes of $n = 0{\cdot}2p$ and $n = 1{\cdot}2p$ are chosen to illustrate cases where $n \ll p$ and where $n \geqslant p$. Table 1 displays the results for problems with $p = 5000$ for varying n and s. The choice of penalty parameter λ was made in line with Guillot et al. (2012). The times reported for our algorithm and the proximal gradient algorithm in Table 1 are the times required to achieve a better duality gap than the proximal Newton algorithm. This explains the different duality gaps reported for each of the three methods.

The convergence times of our algorithm are much lower than for the two proximal algorithms, in all instances except one. The most computationally expensive step in our algorithm is the inversion step. In contrast, the most computationally expensive step of the proximal Newton algorithm is the Newton step, which solves a very large lasso problem using sequential coordinatewise

Table 1. *Comparisons of convergence time, number of iterations and duality gap for simulated datasets*

			Proximal gradient			Proximal Newton			GAMA		
λ	nnz(Φ_*)	$\kappa(\Gamma_*)$	Time (s)	Iter	Gap	Time (s)	Iter	Gap	Time (s)	Iter	Gap
Dataset: $p = 5000$, $n = 1000$, $s = 0{\cdot}03$											
0·10	0·5%	1·4	59	11	2e-7	102	7	3e-7	41	7	7e-8
0·08	2·0%	2·3	215	28	1e-11	434	9	1e-11	92	16	1e-11
0·06	6·4%	8·3	478	54	9e-9	1607	10	9e-9	284	47	7e-9
0·04	13·7%	23·1	935	127	6e-7	6315	14	1e-6	322	53	1e-6
Dataset: $p = 5000$, $n = 6000$, $s = 0{\cdot}03$											
0·08	0·2%	1·2	76	11	2e-8	40	5	9e-8	31	5	1e-9
0·06	1·1%	1·6	59	11	2e-7	145	6	1e-6	52	9	9e-9
0·04	3·2%	3·6	213	27	2e-9	563	8	3e-9	167	27	8e-10
0·02	12·9%	15·0	391	61	4e-6	4682	12	4e-6	247	41	1e-6
Dataset: $p = 5000$; $n = 1000$, $s = 0{\cdot}15$											
0·08	2·5%	3·3	84	12	6e-7	365	7	9e-7	135	20	7e-7
0·06	6·9%	18·2	933	109	1e-8	2099	11	1e-8	507	82	7e-9
0·04	13·9%	39·4	1599	229	4e-5	5270	12	4e-5	270	45	4e-5
0·02	26·0%	84·1	6865	1000	4e-3	31841	24	3e-6	390	62	2e-7
Dataset: $p = 5000$, $n = 6000$, $s = 0{\cdot}15$											
0·08	0·0%	1·1	66	11	1e-10	31	5	3e-9	24	4	4e-11
0·06	0·7%	1·4	53	10	8e-7	79	5	1e-5	35	6	8e-8
0·04	5·5%	7·1	236	31	1e-7	1386	10	8e-7	218	37	9e-7
0·02	18·7%	26·9	1771	218	4e-7	7132	12	4e-6	225	39	3e-7

GAMA, graphical alternating minimization algorithm; Iter, number of iterations; nnz(Φ^*), the percentage of nonzero entries or the sparsity level of the matrix Φ.

descent. This step is nonparallelizable and therefore the speed-ups in sparse matrix inversion enjoyed by our algorithm are not leveraged by the proximal Newton algorithm in solving large lasso problems. Thus the latter algorithm requires many fewer iterations than the former, but each iteration is much more computationally intensive. The advantage of our algorithm is even more prominent when λ is small or when the solution is ill-conditioned. As discussed in § 5·4, this advantage is magnified when working with real data, where the optimal solutions are often highly ill-conditioned.

5·4. *Real datasets*

We compare the three methods using two sets of real data, on estrogen levels (Pittman et al., 2004) and temperature (Brohan et al., 2006). The estrogen dataset consists of expression data for $p = 682$ genes from $n = 158$ patients with breast cancer. The temperature dataset consists of average annual temperature measurements from $p = 1732$ locations over $n = 157$ years between 1850 and 2006. The values of λ chosen to test the algorithms are varied to obtain sparsity levels between 2% and 12%. Both datasets yield extremely ill-conditioned problems for these values of λ. We know that the ability of our algorithm to control the duality gap is very useful for obtaining accurate solutions to highly ill-conditioned problems. We used a maximum of 5000 iterations for our algorithm and the proximal gradient algorithm. The duality gap achieved by the proximal Newton algorithm decreases with tolerance on the subgradient but then stops reducing despite lowering of the tolerance provided to the solver. The duality gaps reported in Table 2 for the proximal Newton algorithm are the best that could be achieved by any tolerance on the subgradient condition; these can be as high as 10^{-5} when the problem is highly ill-conditioned.

Table 2. *Comparisons of convergence time, number of iterations and duality gap for real datasets*

			Proximal gradient			Proximal Newton			GAMA			
λ	nnz(Φ_*)	$\kappa(\Gamma_*)$	Time (s)	Iter	Gap	Time (s)	Iter	Gap	Time (s)	Iter	Gap	Time[1]
Dataset: estrogen; $p = 682$, $n = 158$												
0·40	2·6%	42	21	532	1e-6	3	13	2e-6	5	180	2e-6	10
0·30	3·4%	88	35	911	7e-7	6	19	2e-7	17	548	4e-8	21
0·20	4·4%	193	80	2182	7e-7	23	27	8e-7	16	488	8e-7	27
0·10	7·1%	475	—	5000	—	64	49	2e-6	11	339	2e-6	22
0·05	12·4%	986	—	5000	—	335	93	2e-6	11	321	1e-6	16
Dataset: temperature; $p = 1732$, $n = 157$												
0·30	1·7%	589	1242	2940	5e-7	279	21	1e-6	438	1161	9e-7	639
0·20	2·0%	1076	—	5000	—	589	28	4e-6	356	931	3e-6	642
0·10	3·3%	2302	—	5000	—	953	47	2e-5	216	581	1e-5	301
0·05	5·6%	4505	—	5000	—	3402	90	8e-6	249	652	6e-6	374
0·03	7·8%	7201	—	5000	—	9433	136	2e-5	209	580	9e-6	346

GAMA, graphical alternating minimization algorithm; Iter, number of iterations; Time[1], time for achieving an optimality gap of $\epsilon = 10^{-10}$; nnz(Φ^*), the percentage of nonzero entries or the sparsity level of the matrix Φ.

In contrast, our algorithm can achieve duality gaps of less than 10^{-10}. Similar to synthetic data experiments, we report the time required to achieve a duality gap no greater than that achieved by the proximal Newton algorithm. Further, in the last column we report the time needed to achieve a fixed duality gap of $\epsilon = 10^{-10}$. The iterates of the proximal Newton algorithm are unable to attain this duality gap for any subgradient tolerance, whereas the proximal gradient algorithm is very slow for such problems and fails to converge in 5000 iterations. Our algorithm suffers from neither of these two drawbacks.

As seen in Table 2, for both examples the condition number of the optimal solution is extremely high for smaller values of λ. The proximal Newton algorithm performs better than our algorithm when λ is larger, but becomes extremely slow in comparison as λ decreases. Our algorithm is up to 30 times faster for the estrogen dataset and up to 45 times faster for the temperature dataset than the proximal Newton algorithm when the condition numbers of the solution are very high. Such speed-ups are necessary in applications where the inverse covariance needs to be estimated several times. For example, in climate field reconstruction (Schneider, 2001), crossvalidation requires solving (1) for a grid of values of the penalty parameter to determine the best penalty parameter. Similarly, for uncertainty quantification, both parametric and nonparametric bootstrap require covariance estimation numerous times.

5·5. *Heuristics and insights*

Across all the numerical experiments we have observed that as p increases, our algorithm and the proximal gradient algorithm scale more favourably than the proximal Newton algorithm. In general, as n increases, the condition number of the sample covariance S improves, and all the methods perform better because the iterates and the optimal solution are better conditioned. We also observed that as λ increases, the number of iterations required for convergence reduces. For large values of λ, the performance of each algorithm depends on the initial point that is chosen.

While the proximal methods use $\Omega_0 = \{\text{diag}(S) + \lambda I\}^{-1}$ as the starting point, the inverse $\Gamma_0 = \text{diag}(S) + \lambda I$ works very well in our algorithm for large values of λ. However, as the value of λ decreases, we have found that the next iterate Γ_1 is not always positive definite. This can be explained by violation of dual feasibility of Γ_0 for small values of λ. A better, consistent starting

point which works for all values of λ is $\Gamma_0 = S + \lambda I$. The timing comparisons reported in this section are times starting with the best initial point. In the majority of cases, the best initial point for our algorithm was $S + \lambda I$, which can be used in general; $\mathrm{diag}(S) + \lambda I$ can be used to obtain additional speed-ups for larger λ.

6. Portfolio optimization

6·1. *Minimum variance portfolio with rebalancing*

Portfolio optimization refers to the problem of determining weights or proportions, in monetary terms, in order to invest in a set of securities that minimize the risk for a given level of return. For a portfolio with p risky assets, let r_i denote the return of asset i over a given period, i.e., the change in price over one time period divided by the price at the beginning of the period. Let Σ denote the covariance matrix of $r = (r_1, \ldots, r_p)$ and w_i the weight of asset i in the portfolio during a given period; w_i could be positive or negative depending on long or short positions. The minimum variance portfolio selection problem solves

$$\underset{w \in \mathbb{R}^p}{\text{minimize}}\ w^{\mathrm{T}} \Sigma w \quad \text{subject to} \quad \mathbb{1}^{\mathrm{T}} w = 1, \tag{26}$$

where $\mathbb{1}$ is a p-dimensional vector of ones and $\mathbb{R}^p$ is the space of p-dimensional vectors. The objective represents the variance of the return, also defined as the risk associated with the particular portfolio, and the linear constraint represents the budget constraint. For a given Σ, $w^* = (\mathbb{1}^{\mathrm{T}} \Sigma^{-1} \mathbb{1})^{-1} \Sigma^{-1} \mathbb{1}$ is the analytic solution of (26). The standard problem defined above assumes stationarity of the returns. To account for nonstationarity, we employ the minimum variance portfolio rebalancing strategy. This approach updates the portfolio weights every L units of time, dividing the trading horizon into blocks each consisting of L time units. At the start of each block, (26) is solved based on the past estimation horizon size of N_{est} observations of returns. The weights w are then held constant for L time units during the holding periods. We assume that the total number of time units in the entire period, N_{tot}, is given by $N_{\mathrm{tot}} = N_{\mathrm{est}} + KL$ for some positive integer K. Therefore we have K updates of portfolio weights $w^{(j)}$ over the holding period $\{N_{\mathrm{est}} + (j-1)L + 1, N_{\mathrm{est}} + jL\}$ $(j = 1, \ldots, K)$. A regularized covariance estimate is a critical ingredient in determining the minimum variance portfolio. Moreover, extremely fast methods like our algorithm are required, as the covariance needs to be estimated repeatedly over moving time blocks.

6·2. *Application to the Dow Jones industrial average*

In the numerical study, we use data for 29 stocks that constituted the Dow Jones Industrial Average in July 2008 and which had a start date of 1 January 1995 and an end date of 26 October 2012. We chose this basket of securities deliberately, as covariance estimation methods have already been illustrated on these data. The first rebalancing interval begins on 1 January 1995, and the last interval begins on 2 July 2012. The entire trading horizon consists of $K = 54$ holding periods of $L = 80$ days each, and we take $N_{\mathrm{est}} = (75, 150, 300)$. We estimate the sample covariance S using N_{est} daily returns from the Dow Jones Industrial Average. We compare the minimum variance portfolio rebalancing strategy that employs covariance matrices generated using our algorithm with constant penalty parameter $\lambda = 0{\cdot}1\|S\|_2$ and the other covariance estimation methods considered in Won et al. (2013), as well as the Dow Jones Industrial Average itself. The covariance estimation methods considered include the condition number regularized

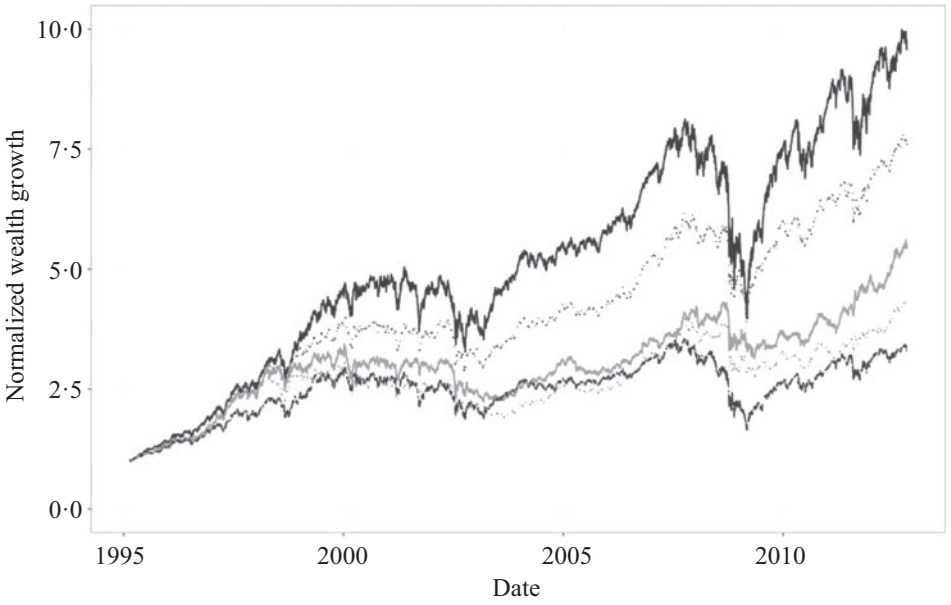

Fig. 2. Normalized wealth growth obtained from our algorithm (solid black), the condition number regularized covariance approach (dotted black), the linear shrinkage method (solid grey), sample covariance (dotted grey), and the Dow Jones Industrial Average (dashed black).

Table 3. *Realized return, risk, Sharpe ratio, turnover and size of short side*

Method	Return (%)	Risk (%)	Sharpe ratio	Turnover	Size of short side
			$N_{\text{est}} = 75$		
GAMA	14·4 (4·1)	19·0 (0·6)	0·49 (0·2)	0·19 (0·04)	0·00 (0·00)
CR	13·3 (3·8)	17·7 (0·5)	0·47 (0·2)	0·59 (0·44)	0·04 (0·07)
LS	9·7 (3·6)	15·4 (0·4)	0·31 (0·2)	1·79 (0·54)	0·20 (0·07)
Sample	9·6 (4·1)	17·5 (0·3)	0·26 (0·2)	2·67 (0·76)	0·30 (0·06)
			$N_{\text{est}} = 150$		
GAMA	14·4 (4·1)	19·0 (0·6)	0·49 (0·2)	0·18 (0·04)	0·00 (0·00)
CR	12·9 (3·6)	16·5 (0·5)	0·48 (0·2)	0·92 (0·37)	0·09 (0·06)
LS	9·8 (3·4)	14·9 (0·4)	0·33 (0·2)	1·75 (0·53)	0·19 (0·07)
Sample	9·5 (3·6)	15·7 (0·4)	0·29 (0·2)	2·14 (0·68)	0·25 (0·06)
			$N_{\text{est}} = 300$		
GAMA	14·4 (4·1)	19·1 (0·6)	0·49 (0·2)	0·17 (0·03)	0·00 (0·00)
CR	13·1 (3·5)	16·2 (0·5)	0·50 (0·2)	1·04 (0·16)	0·10 (0·03)
LS	10·7 (3·3)	14·7 (0·4)	0·39 (0·2)	1·70 (0·54)	0·19 (0·07)
Sample	10·8 (3·4)	15·1 (0·4)	0·39 (0·2)	1·89 (0·62)	0·22 (0·07)

GAMA, our algorithm; CR, condition number regularization; LS, linear shrinkage.

covariance approach (Oh et al., 2013), the linear shrinkage scheme (Ledoit & Wolf, 2004), and the sample covariance matrix. The performance of these methods is compared in terms of the realized return, the realized risk, the Sharpe ratio, the turnover, the size of the short side, and normalized wealth growth; see the Supplementary Material.

6·3. *Performance metrics*

Figure 2 shows sample time series of normalized wealth growth over the trading horizon for our algorithm, the three methods being compared with it, and the Dow Jones Industrial Average

for $N_{\text{est}} = 150$. The normalized wealth growth obtained using our algorithm is better than for the other methods. Table 3 reports the realized return, realized risk, Sharpe ratio, turnover, and size of the short side. Each entry is the mean of the metric over the trading period, with the corresponding standard deviation in parentheses. The standard deviations are computed in a heteroskedasticity- and autocorrelation-consistent manner as discussed in Ledoit & Wolf (2008, § 3.1). Our algorithm gives high values of the Sharpe ratio across estimation horizons. An additional advantage of our algorithm is that the turnover is lower and the size of the short side is negligible. Our algorithm yields a stable portfolio which avoids excessive costs of continuous rebalancing, transaction and borrowing. Hence the net effective wealth growth is even higher than that indicated in Fig. 2.

Acknowledgement

The authors were partially supported by the U.S. National Science Foundation, the U.S. Air Force Office of Scientific Research, the Defense Advanced Research Projects Agency, the UPS Foundation and Stanford Management Company-DBNKY. The authors thank Sang-Yun Oh for making R code available, Dominique Guillot for reading the proofs, and Mukta Gore for her help with formatting. Rajaratnam is also affiliated with Stanford University and was visiting the University of Sydney when a part of this work was undertaken. The authors would also like to acknowledge the authors of the QUIC and G-ISTA algorithms for use of their code.

Supplementary material

Supplementary material available at *Biometrika* online includes additional results of the numerical experiments, details of the step size heuristics, and definitions of the portfolio optimization metrics.

Appendix

Proof of Lemma 1

For Ω_* in (10), $(\Omega_*^{-1})_{ij} - S_{ij} = \lambda\,\text{sign}\{(\Omega_*)_{ij}\}$. Therefore $\mathcal{C}_\lambda\{\tau(\Omega_*)_{ij} + (\Omega_*^{-1})_{ij} - S_{ij}\} = \mathcal{C}_\lambda[\tau(\Omega_*)_{ij} + \lambda\,\text{sign}\{(\Omega_*)_{ij}\}] = \lambda\,\text{sign}\{(\Omega_*)_{ij}\}$, which proves that $\Gamma_* = \Omega_*^{-1}$ satisfies (11).

For Γ_* in (11), its inverse Ω_* satisfies $\Omega_*^{-1} - S = \mathcal{C}_\lambda\{\tau\Omega_* + (\Omega_*^{-1} - S)\}$. If $(\Omega_*)_{ij} = 0$, then $|(\Omega_*^{-1})_{ij} - S_{ij}| \leqslant \lambda$; and if $(\Omega_*)_{ij} \neq 0$, then $(\Omega_*^{-1})_{ij} - S_{ij} = \lambda\,\text{sign}\{(\Omega_*)_{ij}\}$. Combining the two cases, we have $(\Omega_*^{-1}) - S = \lambda\,\text{sign}\{(\Omega_*)\}$. Therefore Ω_* satisfies (10).

Proof of Lemma 3

Using (9) and (11) together with Lemma 6, the nonexpansive property of $\mathcal{C}_\lambda$, we get

$$\begin{aligned}\|\Gamma_+ - \Gamma_*\|_{\text{F}} &= \big\|\mathcal{C}_\lambda(\Gamma + \tau\Gamma^{-1} - S) - \mathcal{C}_\lambda(\Gamma_* + \tau\Gamma_*^{-1} - S)\big\|_{\text{F}},\\ &\leqslant \big\|(\Gamma + \tau\Gamma^{-1}) - (\Gamma_* + \tau\Gamma_*^{-1})\big\|_{\text{F}}. \end{aligned} \tag{A1}$$

Following the arguments for Lemma 3 in Guillot et al. (2012), define a function $h_\tau(\Gamma) = \text{vec}(\Gamma) + \text{vec}(\tau\Gamma^{-1})$. The Jacobian J_{h_τ} of h_τ is $J_{h_\tau}(\Gamma) = I_{p^2} - \tau\Gamma^{-1} \otimes \Gamma^{-1}$, where $\otimes$ is the Kronecker product. The function $h_\tau(\Gamma)$ is differentiable on $\mathcal{D}$. Applying the mean value theorem to (A1) gives

$$\|\Gamma_+ - \Gamma_*\|_{\text{F}} \leqslant \sup_{\delta\in[0,1]} \big(\|I_{p^2} - \tau\Gamma_\delta^{-1} \otimes \Gamma_\delta^{-1}\|_2\big)\|\Gamma - \Gamma_*\|_{\text{F}}, \tag{A2}$$

where $\Gamma_\delta = \delta\Gamma + (1-\delta)\Gamma_*$ is a convex combination of Γ and Γ_*. The eigenvalues of Γ_δ are bounded using Weyl's inequality, $\alpha I \preceq \Gamma_\delta \preceq \beta I$, for α and β from Lemma 2. The eigenvalues of $I_{p^2} - \tau\Gamma_\delta^{-1} \otimes \Gamma_\delta^{-1}$, $1 - \tau\{\text{eig}(\Gamma_\delta)^{-2}\}$, are bounded by $\max(|1-\tau\alpha^{-2}|, |1-\tau\beta^{-2}|)$. Substituting into (A2) yields

$$\|\Gamma_+ - \Gamma_*\|_{\text{F}} \leqslant \max\big(|1-\tau\alpha^{-2}|, |1-\tau\beta^{-2}|\big)\|\Gamma - \Gamma_*\|_{\text{F}}.$$

Proof of Lemma 4

The maximum eigenvalue of Γ_{k+1} satisfies $\|\Gamma_{k+1}\|_2 \leqslant \|\Gamma_{k+1} - \Gamma_*\|_2 + \|\Gamma_*\|_2 \leqslant \|\Gamma_{k+1} - \Gamma_*\|_{\text{F}} + \|\Gamma_*\|_2$. Since $\tau_k < \alpha_k^2 \leqslant \beta^2$, we have $\max(|1-\tau_k\alpha_k^{-2}|, |1-\tau_k\beta^{-2}|) \leqslant 1$. Therefore $\|\Gamma_{k+1}\|_2 \leqslant \|\Gamma_k - \Gamma_*\|_{\text{F}} + \|\Gamma_*\|_2$; repeatedly applying Lemma 3 then gives $\|\Gamma_{k+1}\|_2 \leqslant \|\Gamma_0 - \Gamma_*\|_{\text{F}} + \|\Gamma_*\|_2$, which completes the proof upon using an inductive argument on k.

Proof of Lemma 5

Let $a = \lambda_{\min}(\Gamma)$ be the smallest eigenvalue of Γ. By Lemma 4 we have

$$\log\det\Gamma_0 < \log\det\Gamma \leqslant \log(a) + (p-1)\log(\beta).$$

The initial point $\Gamma_0 = S + \lambda I$ satisfies $\lambda I \preceq \Gamma_0$. Therefore, $p\lambda \leqslant \log\det\Gamma_0 < \log(a) + (p-1)\log(\beta)$ or $\lambda^p\beta^{-(p-1)} < a = \lambda_{\min}(\Gamma)$.

Proof of Theorem 1

Substituting Ω_{k+1} and Φ_{k+1} in terms of Γ_k as in (9), we get

$$\Gamma_{k+1} = \mathcal{C}_\lambda(\Gamma_k + \tau\Gamma_k^{-1} - S) + S.$$

The initial point Γ_0 is $S + \lambda I$, and the subsequent iterates satisfy $\alpha I \preceq \Gamma_k \preceq \beta I$ by Lemmas 4 and 5 with α and β as defined in the lemmas. Using Lemma 3,

$$\|\Gamma_{k+1} - \Gamma_*\|_{\text{F}} \leqslant \max\big(|1-\tau_k\alpha_k^{-2}|, |1-\tau_k\beta^{-2}|\big)\|\Gamma_k - \Gamma_*\|_{\text{F}} \quad (k = 0, 1, \ldots).$$

For a constant step size $\tau_k < \alpha_k^2 < \beta$, the term $\max(|1-\tau_k\alpha_k^{-2}|, |1-\tau_k\beta^{-2}|)$ is less than 1. So $\|\Gamma_{k+1} - \Gamma_*\|_{\text{F}} \leqslant \gamma\|\Gamma_k - \Gamma_*\|_{\text{F}}$, thereby proving the linear convergence of Algorithm 1.

Proof of Theorem 2

By Lemma 6, the modified Γ_+ update (18) satisfies

$$\|\Gamma_+ - \Gamma_*\|_{\text{F}} = \big\|\text{prox}_{\tau\mathcal{D}}(\Gamma + \tau\Gamma^{-1}) - \text{prox}_{\tau\mathcal{D}}(\Gamma_* + \tau\Gamma_*^{-1})\big\|_{\text{F}} \leqslant \|(\Gamma + \tau\Gamma^{-1}) - (\Gamma_* + \tau\Gamma_*^{-1})\|_{\text{F}}.$$

The rest of the proof of Lemma 3 follows accordingly. Similarly, the proof of Theorem 1 can be adapted to prove the linear convergence of the modified algorithm.

To prove the convergence of (25), we use a similar mean value theorem argument on the function $h_\tau(\Gamma) = \text{vec}(\Gamma) + \tau\,\text{vec}(T[\{S + T^*(\Gamma)\}^{-1}])$ to get $\|\Gamma_+ - \Gamma_*\|_{\text{F}} \leqslant \max\{|1-\tau v|, |1-\tau w|\}\|\Gamma - \Gamma_*\|_{\text{F}}$, where v and w are constants which depend on S, T and λ.

References

Banerjee, O., El Ghaoui, L. & d'Aspremont, A. (2008). Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.* **9**, 485–516.

Barabasi, A.-L. & Oltvai, Z. N. (2004). Network biology: Understanding the cell's functional organization. *Nature Rev. Genet.* **5**, 101–13.

Brohan, P., Kennedy, J. J., Harris, I., Tett, S. F. B. & Jones, P. D. (2006). Uncertainty estimates in regional and global observed temperature changes: A new data set from 1850. *J. Geophys. Res.* **111**, D12106.

d'Aspremont, A., Banerjee, O. & El Ghaoui, L. (2008). First-order methods for sparse covariance selection. *SIAM J. Matrix Anal. Appl.* **30**, 56–66.

Friedman, J. H., Hastie, T. J. & Tibshirani, R. J. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9**, 432–41.

Gehrmann, H. & Lauritzen, S. L. (2012). Estimation of means in graphical Gaussian models with symmetries. *Ann. Statist.* **40**, 1061–73.

Guillot, D., Rajaratnam, B. & Emile-Geay, J. (2015). Statistical paleoclimate reconstructions via Markov random fields. *Ann. Appl. Statist.* **9**, 324–52.

Guillot, D., Rajaratnam, B., Rolfs, B., Wong, I. & Maleki, A. (2012). Iterative thresholding algorithm for sparse inverse covariance estimation. *Adv. Neural Info. Proces. Syst.* **25**, 1574–82.

Hsieh, C. J., Dhillon, I., Ravikumar, P. & Banerjee, A. (2012). A divide-and-conquer method for sparse inverse covariance estimation. *Adv. Neural Info. Proces. Syst.* **25**, 2339–47.

Hsieh, C. J., Dhillon, I. S., Ravikumar, P. K. & Sustik, M. A. (2011). Sparse inverse covariance matrix estimation using quadratic approximation. *Adv. Neural Info. Proces. Syst.* **24**, 2330–8.

Khare, K., Oh, S.-Y. & Rajaratnam, B. (2015). A convex pseudolikelihood framework for high dimensional partial correlation estimation with convergence guarantees. *J. R. Statist. Soc.* B **77**, 803–25.

Ledoit, O. & Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *J. Mult. Anal.* **88**, 365–411.

Ledoit, O. & Wolf, M. (2008). Robust performance hypothesis testing with the Sharpe ratio. *J. Empirical Finan.* **15**, 850–9.

Lee, J. D., Sun, Y. & Saunders, M. A. (2012). Proximal Newton-type methods for convex optimization. *Adv. Neural Info. Proces. Syst.* **25**, 836–44.

Lu, Z. (2009). Smooth optimization approach for sparse covariance selection. *SIAM J. Optimiz.* **19**, 1807–27.

Lu, Z. (2010). Adaptive first-order methods for general sparse inverse covariance selection. *SIAM J. Matrix Anal. Appl.* **31**, 2000–16.

Mazumder, R. & Hastie, T. J. (2012). The graphical lasso: New insights and alternatives. *Electron. J. Statist.* **6**, 2125–49.

Oh, S.-Y., Rajaratnam, B. & Won, J. H. (2013). *CondReg: Condition number regularized covariance estimation*. R package version 0.16.

Pittman, J., Huang, E., Dressman, H., Horng, C. F., Cheng, S. H., Tsou, M. H., Chen, C. M., Bild, A., Iversen, E. S., Huang, A. T. et al. (2004). Integrated modeling of clinical and gene expression information for personalized prediction of disease outcomes. *Proc. Nat. Acad. Sci.* **101**, 8431–6.

Rockafellar, R. T. (1976). Monotone operators and the proximal point algorithm. *SIAM J. Contr. Optimiz.* **14**, 877–98.

Scheinberg, K., Ma, S. & Goldfarb, D. (2010). Sparse inverse covariance selection via alternating linearization methods. *Adv. Neural Info. Proces. Syst.* **23**, 2101–9.

Schneider, T. (2001). Analysis of incomplete climate data: Estimation of mean values and covariance matrices and imputation of missing values. *J. Climate* **14**, 853–71.

Tseng, P. (1991). Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM J. Contr. Optimiz.* **29**, 119–38.

Won, J., Kim, S. J., Lim, J. & Rajaratnam, B. (2013). Condition number-regularized covariance estimation. *J. R. Statist. Soc.* B **75**, 427–50.

[*Received on* 25 *April* 2014. *Editorial decision on* 29 *August* 2016]