Journal of Machine Learning Research 15 (2014) 2911-2947 Submitted 4/13; Revised 4/14; Published 10/14

# QUIC: Quadratic Approximation for Sparse Inverse Covariance Estimation

**Cho-Jui Hsieh** CJHSIEH@CS.UTEXAS.EDU
**Mátyás A. Sustik** MSUSTIK@GMAIL.COM
**Inderjit S. Dhillon** INDERJIT@CS.UTEXAS.EDU
**Pradeep Ravikumar** PRADEEPR@CS.UTEXAS.EDU
*Department of Computer Sciences*
*University of Texas at Austin*
*Austin, TX 78712, USA*

**Editor:** Xiaotong Shen

## Abstract

The $\ell_1$-regularized Gaussian maximum likelihood estimator (MLE) has been shown to have strong statistical guarantees in recovering a sparse inverse covariance matrix, or alternatively the underlying graph structure of a Gaussian Markov Random Field, from very limited samples. We propose a novel algorithm for solving the resulting optimization problem which is a regularized log-determinant program. In contrast to recent state-of-the-art methods that largely use first order gradient information, our algorithm is based on Newton's method and employs a quadratic approximation, but with some modifications that leverage the structure of the sparse Gaussian MLE problem. We show that our method is superlinearly convergent, and present experimental results using synthetic and real-world application data that demonstrate the considerable improvements in performance of our method when compared to previous methods.

**Keywords:** covariance, graphical model, regularization, optimization, Gaussian Markov random field

## 1. Introduction

Statistical problems under modern data settings are increasingly high-dimensional, so that the number of parameters is very large, potentially outnumbering even the number of observations. An important class of such problems involves estimating the graph structure of a Gaussian Markov random field (GMRF), with applications ranging from biological inference in gene networks, analysis of fMRI brain connectivity data and analysis of interactions in social networks. Specifically, given $n$ independently drawn samples $\{\mathbf{y_1}, \mathbf{y_2}, \ldots, \mathbf{y_n}\}$ from a $p$-variate Gaussian distribution, so that $\mathbf{y_i} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$, the task is to estimate its inverse covariance matrix $\Sigma^{-1}$, also referred to as the *precision* or *concentration* matrix. The non-zero pattern of this inverse covariance matrix $\Sigma^{-1}$ can be shown to correspond to the underlying graph structure of the GMRF. An active line of work in high-dimensional settings, where $p \gg n$, is based on imposing constraints on the model space; in the GMRF case a common structured constraint is that of sparsity of the inverse covariance matrix. Accordingly, recent papers by Banerjee et al. (2008); Friedman et al. (2008); Yuan and Lin (2007) have

proposed an estimator that minimizes the Gaussian negative log-likelihood regularized by the $\ell_1$ norm of the entries (typically restricted to those on the off-diagonal) of the inverse covariance matrix, which encourages sparsity in its entries. This estimator has been shown to have very strong statistical guarantees even under very high-dimensional settings, including convergence in Frobenius and spectral norms (Rothman et al., 2008; Lam and Fan, 2009; Ravikumar et al., 2011), as well as in recovering the non-zero pattern of the inverse covariance matrix, or alternatively the graph structure of the underlying GMRF (Ravikumar et al., 2011). Moreover, the resulting optimization problem is a log-determinant program, which is convex, and can be solved in polynomial time.

For such large-scale optimization problems arising from high-dimensional statistical estimation however, standard optimization methods typically suffer sub-linear rates of convergence (Agarwal et al., 2010). This would be too expensive for the Gaussian MLE problem, since the number of matrix entries scales quadratically with the number of nodes. Luckily, the log-determinant problem has special structure; the log-determinant function is strongly convex and one can thus obtain linear (i.e., geometric) rates of convergence via the state-of-the-art methods. However, even linear rates in turn become infeasible when the problem size is very large, with the number of nodes in the thousands and the number of matrix entries to be estimated in the millions. Here we ask the question: *can we obtain superlinear rates of convergence for the optimization problem underlying the $\ell_1$-regularized Gaussian MLE?*

For superlinear rates, one has to consider second-order methods which at least in part use the Hessian of the objective function. There are however some caveats to the use of such second-order methods in high-dimensional settings. First, a straightforward implementation of each second-order step would be very expensive for high-dimensional problems. Secondly, the log-determinant function in the Gaussian MLE objective acts as a barrier function for the positive definite cone. This barrier property would be lost under quadratic approximations so there is a danger that Newton-like updates will not yield positive-definite matrices, unless one explicitly enforces such a constraint in some manner.

In this paper, we present QUIC (QUadratic approximation of Inverse Covariance matrices), a second-order algorithm, that solves the $\ell_1$-regularized Gaussian MLE. We perform Newton steps that use iterative quadratic approximations of the Gaussian negative log-likelihood. The computation of the Newton direction is a *Lasso* problem (Meier et al., 2008; Friedman et al., 2010), which we then solve using coordinate descent. A key facet of our method is that we are able to reduce the computational cost of a coordinate descent update from the naive $O(p^2)$ to $O(p)$ complexity by exploiting the structure present in the problem, and by a careful arrangement and caching of the computations. Furthermore, an Armijo-rule based step size selection rule ensures sufficient descent and positive definiteness of the intermediate iterates. Finally, we use the form of the stationary condition characterizing the optimal solution to *focus* the Newton direction computation on a small subset of *free* variables, but in a manner that preserves the strong convergence guarantees of second-order descent. We note that when the solution has a block-diagonal structure as described in Mazumder and Hastie (2012); Witten et al. (2011), the *fixed/free* set selection in QUIC can automatically identify this block diagonal structure and avoid updates to the off-diagonal block elements. A preliminary version of this paper appeared in Hsieh et al. (2011). In this paper, we provide a more detailed analysis along with proofs of our algorithm, and cover a

more general weighted regularization case of the regularized inverse covariance estimation problem. We show that QUIC can automatically identify the sparsity structure under the block-diagonal case. We also conduct more experiments on both synthetic and real data sets to compare QUIC with other solvers. Our software package QUIC with MATLAB and R interface[1] is public available at `http://www.cs.utexas.edu/~sustik/QUIC/`.

The outline of the paper is as follows. We start with a review of related work and the problem setup in Section 2. In Section 3, we present our algorithm that combines quadratic approximation, Newton's method and coordinate descent. In Section 4, we show superlinear convergence of our method. We summarize the experimental results in Section 5, where we compare the algorithm using both real data and synthetic examples from Li and Toh (2010). We observe that our algorithm performs overwhelmingly better (quadratic instead of linear convergence) than existing solutions described in the literature.

Notation. In this paper, boldfaced lowercase letters denote vectors and uppercase letters denote $p \times p$ real matrices. $\mathcal{S}^p_{++}$ denotes the space of $p \times p$ symmetric positive definite matrices while $X \succ 0$ and $X \succeq 0$ means that $X$ is positive definite and positive semidefinite, respectively. The vectorized listing of the elements of a $p \times p$ matrix $X$ is denoted by $\mathrm{vec}(X) \in \mathbb{R}^{p^2}$ and the Kronecker product of the matrices $X$ and $Y$ is denoted by $X \otimes Y$. For a real-valued function $f(X)$, $\nabla f(X)$ is a $p \times p$ matrix with $(i,j)$ element equal to $\frac{\partial}{\partial X_{ij}} f(X)$ and denoted by $\nabla_{ij} f(X)$, while $\nabla^2 f(X)$ is the $p^2 \times p^2$ Hessian matrix. We will use the $\ell_1$ and $\ell_\infty$ norms defined on the vectorized form of matrix $X$: $\|X\|_1 := \sum_{i,j} |X_{ij}|$ and $\|X\|_\infty := \max_{i,j} |X_{ij}|$. We also employ elementwise $\ell_1$-regularization, $\|X\|_{1,\Lambda} := \sum_{i,j} \lambda_{ij} |X_{ij}|$, where $\Lambda = [\lambda_{ij}]$ with $\lambda_{ij} > 0$ for off-diagonal elements, and $\lambda_{ii} \geq 0$ for diagonal elements.

## 2. Background and Related Work

Let $\mathbf{y}$ be a $p$-variate Gaussian random vector, with distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$. Given $n$ independently drawn samples $\{\mathbf{y_1}, \ldots, \mathbf{y_n}\}$ of this random vector, the sample covariance matrix can be written as

$$S = \frac{1}{n-1} \sum_{k=1}^{n} (\mathbf{y_k} - \hat{\boldsymbol{\mu}})(\mathbf{y_k} - \hat{\boldsymbol{\mu}})^T, \text{ where } \hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{y_k}. \tag{1}$$

Given a regularization penalty $\lambda > 0$, the $\ell_1$-regularized Gaussian MLE for the inverse covariance matrix can be written as the solution of the following regularized *log-determinant* program:

$$\arg\min_{X \succ 0} \left\{ -\log\det X + \mathrm{tr}(SX) + \lambda \sum_{i,j=1}^{p} |X_{ij}| \right\}. \tag{2}$$

The $\ell_1$ regularization promotes sparsity in the inverse covariance matrix, and thus encourages a sparse graphical model structure. We consider a generalized weighted $\ell_1$ regularization, where given a symmetric nonnegative weight matrix $\Lambda = [\lambda_{ij}]$, we can assign different nonnegative weights to different entries, obtaining the regularization term $\|X\|_{1,\Lambda} = \sum_{i,j=1}^{p} \lambda_{ij} |X_{ij}|$. In this paper we will focus on solving the following generalized

1. The QUIC R package is also available from CRAN.

sparse inverse covariance estimation problem:

$$X^* = \arg\min_{X \succ 0} \left\{ -\log\det X + \mathrm{tr}(SX) + \|X\|_{1,\Lambda} \right\} = \arg\min_{X \succ 0} f(X), \tag{3}$$

where $X^* = (\Sigma^*)^{-1}$. In order to ensure that problem (3) has a unique minimizer, as we show later, it is sufficient to require that $\lambda_{ij} > 0$ for off-diagonal entries, and $\lambda_{ii} \geq 0$ for diagonal entries. The standard off-diagonal $\ell_1$ regularization variant $\lambda \sum_{i \neq j} |X_{ij}|$ is a special case of this weighted regularization function. For further details on the background and utility of $\ell_1$ regularization in the context of GMRFs, we refer the reader to Yuan and Lin (2007); Banerjee et al. (2008); Friedman et al. (2008); Duchi et al. (2008); Ravikumar et al. (2011).

Due in part to its importance, there has been an active line of work on efficient optimization methods for solving (2) and (3). Since the regularization term is non-smooth and hard to solve, many methods aim to solve the dual problem of (3):

$$\Sigma^* = \operatorname*{argmax}_{|W_{ij} - S_{ij}| \leq \lambda_{ij}} \log\det W, \tag{4}$$

which has a smooth objective function with bound constraints. Banerjee et al. (2008) propose a block-coordinate descent method to solve the dual problem (4), by updating one row and column of $W$ at a time. They show that the dual of the corresponding row subproblem can be written as a standard Lasso problem, which they then solve by Nesterov's first order method. Friedman et al. (2008) follow the same strategy, but propose to use a coordinate descent method to solve the row subproblems instead; their method is implemented in the widely used R package called GLASSO. In other approaches, the dual problem (4) is treated as a constrained optimization problem, for which Duchi et al. (2008) apply a projected subgradient method called PSM, while Lu (2009) proposes an accelerated gradient descent method called VSM.

Other first-order methods have been pursued to solve the primal optimization problem (2). d'Aspremont et al. (2008) apply Nesterov's first order method to (2) after smoothing the objective function; Scheinberg et al. (2010) apply an augmented Lagrangian method to handle the smooth and nonsmooth parts separately; the resulting algorithm is implemented in the ALM software package. In Scheinberg and Rish (2010), the authors propose to directly solve the primal problem by a greedy coordinate descent method called SINCO. However, each coordinate update of SINCO has a time complexity of $O(p^2)$, which becomes computationally prohibitive when handling large problems. We will show in this paper that after forming the quadratic approximation, each coordinate descent update can be performed in $O(p)$ operations. This trick is one of the key advantages of our proposed method, QUIC.

One common characteristic of the above methods is that they are first-order iterative methods that mainly use gradient information at each step. Such first-order methods have become increasingly popular in recent years for high-dimensional problems in part due to their ease of implementation, and because they require very little computation and memory at each step. The caveat is that they have at most linear rates of convergence (Bertsekas, 1995). To achieve superlinear convergence rates, one has to consider second-order methods, which have only recently attracted some attention for the sparse inverse covariance estimation problem. Li and Toh (2010) handle the non-smoothness of the $\ell_1$ regularization in the

objective function by doubling the number of variables, and solving the resulting constrained optimization problem by an inexact interior point method. Schmidt et al. (2009) propose a second order Projected Quasi-Newton method (PQN) that solves the dual problem (4), since the dual objective function is smooth. The key difference of our method when compared to these recent second order solvers is that we directly solve the $\ell_1$-regularized primal objective using a second-order method. As we show, this allows us to leverage structure in the problem, and efficiently approximate the generalized Newton direction using coordinate descent. Subsequent to the preliminary version of this paper (Hsieh et al., 2011), Olsen et al. (2012) have proposed generalizations to our framework to allow various inner solvers such as FISTA, conjugate gradient (CG), and LBFGS to be used, in addition to our proposed coordinate descent scheme. Also, Lee et al. (2012) have extended the quadratic approximation algorithm to solve general composite functions and analyze the convergence properties.

## 3. Quadratic Approximation Method

We first note that the objective $f(X)$ in the non-differentiable optimization problem (3), can be written as the sum of two parts, $f(X) = g(X) + h(X)$, where

$$g(X) = -\log\det X + \mathrm{tr}(SX) \text{ and } h(X) = \|X\|_{1,\Lambda}. \tag{5}$$

The first component $g(X)$ is twice differentiable, and strictly convex. The second part, $h(X)$, is convex but non-differentiable. Following the approach of Tseng and Yun (2007) and Yun and Toh (2011), we build a quadratic approximation around any iterate $X_t$ for this composite function by first considering the second-order Taylor expansion of the smooth component $g(X)$:

$$\bar{g}_{X_t}(\Delta) \equiv g(X_t) + \mathrm{vec}(\nabla g(X_t))^T \mathrm{vec}(\Delta) + \frac{1}{2}\mathrm{vec}(\Delta)^T \nabla^2 g(X_t)\,\mathrm{vec}(\Delta). \tag{6}$$

The Newton direction $D_t^*$ for the entire objective $f(X)$ can then be written as the solution of the regularized quadratic program:

$$D_t^* = \arg\min_{\Delta} \big\{\bar{g}_{X_t}(\Delta) + h(X_t + \Delta)\big\}. \tag{7}$$

We use this Newton direction to compute our iterative estimates $\{X_t\}$ for the solution of the optimization problem (3). This variant of Newton method for such composite objectives is also referred to as a "proximal Newton-type method," and was empirically studied in Schmidt (2010). Tseng and Yun (2007) considered the more general case where the Hessian $\nabla^2 g(X_t)$ is replaced by any positive definite matrix. See also the recent paper by Lee et al. (2012), where convergence properties of such general proximal Newton-type methods are discussed. We note that a key caveat to applying such second-order methods in high-dimensional settings is that the computation of the Newton direction appears to have a large time complexity, which is one reason why first-order methods have been so popular for solving the high-dimensional $\ell_1$-regularized Gaussian MLE.

Let us delve into the Newton direction computation in (7). Note that it can be rewritten as a standard Lasso regression problem (Tibshirani, 1996):

$$\arg\min_{\Delta} \frac{1}{2}\|H^{\frac{1}{2}}\,\mathrm{vec}(\Delta) + H^{-\frac{1}{2}}\boldsymbol{b}\|_2^2 + \|X_t + \Delta\|_{1,\Lambda}, \tag{8}$$

where $H = \nabla^2 g(X_t)$ and $\boldsymbol{b} = \text{vec}(\nabla g(X_t))$. Many efficient optimization methods exist that solve Lasso regression problems, such as the coordinate descent method (Friedman et al., 2007), the gradient projection method (Polyak, 1969), and iterative shrinking methods (Daubechies et al., 2004; Beck and Teboulle, 2009). When applied to the Lasso problem of (7), most of these optimization methods would require the computation of the gradient of $\bar{g}_{X_t}(\Delta)$:

$$\nabla \bar{g}_{X_t}(\Delta) = H \,\text{vec}(\Delta) + \boldsymbol{b}. \tag{9}$$

The straightforward approach for computing (9) for a general $p^2 \times p^2$ Hessian matrix $H$ would take $O(p^4)$ time, making it impractical for large problems. Fortunately, for the sparse inverse covariance problem (3), the Hessian matrix $H$ has the following special form (see for instance Boyd and Vandenberghe, 2009, Chapter A.4.3):

$$H = \nabla^2 g(X_t) = X_t^{-1} \otimes X_t^{-1},$$

where $\otimes$ denotes the Kronecker product. In Section 3.1, we show how to exploit this special form of the Hessian matrix to perform one coordinate descent step that updates one element of $\Delta$ in $O(p)$ time. Hence a full sweep of coordinate descent steps over all the variables requires $O(p^3)$ time. This key observation is one of the reasons that makes our Newton-like method viable for solving the inverse covariance estimation problem.

There exist other functions which allow efficient Hessian times vector multiplication. As an example, we consider the case of $\ell_1$-regularized logistic regression. Suppose we are given $n$ samples with feature vectors $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n \in \mathbb{R}^p$ and labels $y_1, \ldots, y_n$, and we solve the following $\ell_1$-regularized logistic regression problem to compute the model parameter $\boldsymbol{w}$:

$$\arg\min_{\boldsymbol{w} \in \mathbb{R}^p} \sum_{i=1}^{n} \log(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}) + \lambda \|\boldsymbol{w}\|_1.$$

Following our earlier approach, we can decompose this objective function into smooth and non-smooth parts, $g(\boldsymbol{w}) + h(\boldsymbol{w})$, where

$$g(\boldsymbol{w}) = \sum_{i=1}^{n} \log(1 + e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}) \text{ and } h(\boldsymbol{w}) = \lambda \|\boldsymbol{w}\|_1.$$

In order to apply coordinate descent to solve the quadratic approximation, we have to compute the gradient as in (9). The Hessian matrix $\nabla^2 g(\boldsymbol{w})$ is a $p \times p$ matrix, so direct computation of this gradient costs $O(p^2)$ flops. However, the Hessian matrix for logistic regression has the following simple form

$$H = \nabla^2 g(\boldsymbol{w}) = XDX^T,$$

where $D$ is a diagonal matrix with $D_{ii} = \frac{e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i}}{(1+e^{-y_i \boldsymbol{w}^T \boldsymbol{x}_i})^2}$ and $X = [\boldsymbol{x}_1,\ \boldsymbol{x}_2,\ \ldots,\ \boldsymbol{x}_n]$. Therefore we can write

$$\nabla g(\boldsymbol{w} + \Delta) = (\nabla^2 g(\boldsymbol{w}))\,\text{vec}(\Delta) + \boldsymbol{b} = XD(X^T \,\text{vec}(\Delta)) + \boldsymbol{b}. \tag{10}$$

The time complexity to compute (10) is only proportional to the number of nonzero elements in the data matrix $X$, which can be much smaller than $O(p^2)$ for high-dimensional sparse

data sets. Therefore similar quadratic approximation approaches are also efficient for solving the $\ell_1$-regularized logistic regression problem as shown by Friedman et al. (2010); Yuan et al. (2012).

In the following three subsections, we detail three innovations which make our quadratic approximation algorithm feasible for solving (3). In Section 3.1, we show how to compute the Newton direction using an efficient coordinate descent method that exploits the structure of Hessian matrix, so that we reduce the time complexity of each coordinate descent update step from $O(p^2)$ to $O(p)$. In Section 3.2, we employ an Armijo-rule based step size selection to ensure sufficient descent *and* positive-definiteness of the next iterate. Finally, in Section 3.3 we use the form of the stationary condition characterizing the optimal solution, to *focus* the Newton direction computation to a small subset of *free* variables, in a manner that preserves the strong convergence guarantees of second-order descent. A high level overview of our method is presented in Algorithm 1. Note that the initial point $X_0$ has to be a feasible solution, thus $X_0 \succ 0$, and the positive definiteness of all the following iterates $X_t$ will be guaranteed by the step size selection procedure (step 6 in Algorithm 1).

**Algorithm 1:** QUadratic approximation for sparse Inverse Covariance estimation (QUIC overview)

**Input** : Empirical covariance matrix $S$ (positive semi-definite, $p \times p$), regularization parameter matrix $\Lambda$, initial iterate $X_0 \succ 0$.

**Output**: Sequence $\{X_t\}$ that converges to $\arg\min_{X \succ 0} f(X)$, where $f(X) = g(X) + h(X)$, where $g(X) = -\log\det X + \mathrm{tr}(SX), h(X) = \|X\|_{1,\Lambda}$.

1 **for** $t = 0, 1, \ldots$ **do**
2 Compute $W_t = X_t^{-1}$.
3 Form the second order approximation $\bar{f}_{X_t}(\Delta) := \bar{g}_{X_t}(\Delta) + h(X_t + \Delta)$ to $f(X_t + \Delta)$.
4 Partition the variables into free and fixed sets based on the gradient, see Section 3.3.
5 Use coordinate descent to find the Newton direction $D_t^* = \arg\min_\Delta \bar{f}_{X_t}(X_t + \Delta)$ over the set of free variables, see (13) and (16) in Section 3.1. (A *Lasso* problem.)
6 Use an *Armijo*-rule based step-size selection to get $\alpha$ such that $X_{t+1} = X_t + \alpha D_t^*$ is positive definite and there is sufficient decrease in the objective function, see (21) in Section 3.2.
7 **end**

### 3.1 Computing the Newton Direction

In order to compute the Newton direction, we have to solve the Lasso problem (7). The gradient and Hessian for $g(X) = -\log\det X + \mathrm{tr}(SX)$ are (see, for instance, Boyd and Vandenberghe, 2009, Chapter A.4.3)

$$\nabla g(X) = S - X^{-1} \text{ and } \nabla^2 g(X) = X^{-1} \otimes X^{-1}. \tag{11}$$

In order to formulate our problem accordingly, we can verify that for a symmetric matrix $\Delta$ we have $\mathrm{tr}(X_t^{-1}\Delta X_t^{-1}\Delta) = \mathrm{vec}(\Delta)^T(X_t^{-1} \otimes X_t^{-1})\,\mathrm{vec}(\Delta)$, so that $\bar{g}_{X_t}(\Delta)$ in (7) can be

rewritten as

$$\bar{g}_{X_t}(\Delta) = -\log\det X_t + \mathrm{tr}(SX_t) + \mathrm{tr}((S - W_t)^T\Delta) + \frac{1}{2}\mathrm{tr}(W_t\Delta W_t\Delta), \tag{12}$$

where $W_t = X_t^{-1}$.

In Friedman et al. (2007), Wu and Lange (2008), the authors show that coordinate descent methods are very efficient for solving Lasso type problems. An obvious way to update each element of $\Delta$ in (7) requires $O(p^2)$ floating point operations since $W_t \otimes W_t$ is a $p^2 \times p^2$ matrix, thus yielding an $O(p^4)$ procedure for computing the Newton direction. As we show below, our implementation reduces the cost of updating one variable to $O(p)$ by exploiting the structure of the second order term $\mathrm{tr}(W_t\Delta W_t\Delta)$.

For notational simplicity, we will omit the iteration index $t$ in the derivations below where we only discuss a single Newton iteration; this applies to the rest of the this section and Section 3.2 as well. (Hence, the notation for $\bar{g}_{X_t}$ is also simplified to $\bar{g}$.) Furthermore, we omit the use of a separate index for the coordinate descent updates. Thus, we simply use $D$ to denote the current iterate approximating the Newton direction and use $D'$ for the updated direction. Consider the coordinate descent update for the variable $X_{ij}$, with $i < j$ that preserves symmetry: $D' = D + \mu(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T)$. The solution of the one-variable problem corresponding to (7) is:

$$\arg\min_{\mu} \quad \bar{g}(D + \mu(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T)) + 2\lambda_{ij}|X_{ij} + D_{ij} + \mu|. \tag{13}$$

We expand the terms appearing in the definition of $\bar{g}$ after substituting $D' = D + \mu(\mathbf{e}_i\mathbf{e}_j^T + \mathbf{e}_j\mathbf{e}_i^T)$ for $\Delta$ in (12) and omit the terms not dependent on $\mu$. The contribution of $\mathrm{tr}(SD') - \mathrm{tr}(WD')$ yields $2\mu(S_{ij} - W_{ij})$, while the regularization term contributes $2\lambda_{ij}|X_{ij} + D_{ij} + \mu|$, as seen from (13). The quadratic term can be rewritten (using the fact that $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ and the symmetry of $D$ and $W$) to yield:

$$\mathrm{tr}(WD'WD') = \mathrm{tr}(WDWD) \quad + \quad 4\mu\mathbf{w}_i^T D\mathbf{w}_j + 2\mu^2(W_{ij}^2 + W_{ii}W_{jj}), \tag{14}$$

where $\mathbf{w}_i$ refers to the $i$-th column of $W$. In order to compute the single variable update we seek the minimum of the following quadratic function of $\mu$:

$$\frac{1}{2}(W_{ij}^2 + W_{ii}W_{jj})\mu^2 + (S_{ij} - W_{ij} + \mathbf{w}_i^T D\mathbf{w}_j)\mu + \lambda_{ij}|X_{ij} + D_{ij} + \mu|. \tag{15}$$

Letting $a = W_{ij}^2 + W_{ii}W_{jj}$, $b = S_{ij} - W_{ij} + \mathbf{w}_i^T D\mathbf{w}_j$, and $c = X_{ij} + D_{ij}$ the minimum is achieved for:

$$\mu = -c + \mathcal{S}(c - b/a, \lambda_{ij}/a), \tag{16}$$

where

$$\mathcal{S}(z, r) = \mathrm{sign}(z)\max\{|z| - r, 0\} \tag{17}$$

is the soft-thresholding function. Similarly, when $i = j$, for $D' = D + \mu\mathbf{e}_i\mathbf{e}_i^T$, we get

$$\mathrm{tr}(WD'WD') = \mathrm{tr}(WDWD) \quad + \quad 2\mu\mathbf{w}_i^T D\mathbf{w}_i + \mu^2(W_{ii}^2). \tag{18}$$

Therefore the update rule for $D_{ii}$ can be computed by (16) with $a = W_{ii}^2, b = S_{ii} - W_{ii} + \mathbf{w}_i^T D \mathbf{w}_i$, and $c = X_{ii} + D_{ii}$.

Since $a$ and $c$ are easy to compute, the main computational cost arises while evaluating $\mathbf{w}_i^T D \mathbf{w}_j$, the third term contributing to coefficient $b$ above. Direct computation requires $O(p^2)$ time. Instead, we maintain a $p \times p$ matrix $U = DW$, and then compute $\mathbf{w}_i^T D \mathbf{w}_j$ by $\mathbf{w}_i^T \mathbf{u}_j$ using $O(p)$ flops, where $\mathbf{u}_j$ is the $j$-th column of matrix $U$. In order to maintain the matrix $U$, we also need to update $2p$ elements, namely two coordinates of each $\mathbf{u}_k$ when $D_{ij}$ is modified. We can compactly write the row updates of $U$ as follows: $\mathbf{u}_{i\cdot} \leftarrow \mathbf{u}_{i\cdot} + \mu \mathbf{w}_{j\cdot}$ and $\mathbf{u}_{j\cdot} \leftarrow \mathbf{u}_{j\cdot} + \mu \mathbf{w}_{i\cdot}$, where $\mathbf{u}_{i\cdot}$ refers to the $i$-th *row* vector of $U$.

#### 3.1.1 Update Rule when $X$ is Diagonal

The calculation of the Newton direction can be simplified if $X$ is also a diagonal matrix. For example, this occurs in the first Newton iteration when we initialize QUIC using the identity (or diagonal) matrix. When $X$ is diagonal, the Hessian $\nabla^2 g(X) = X^{-1} \otimes X^{-1}$ is also a diagonal matrix, which indicates that all one variable sub-problems are independent of each other. Therefore, we only need to update each variable once to reach the optimum of (7). In particular, by examining (16), the optimal solution $D_{ij}^*$ is

$$D_{ij}^* = \begin{cases} \mathcal{S}\left(-\frac{S_{ij}}{W_{ii}W_{jj}}, \frac{\lambda_{ij}}{W_{ii}W_{jj}}\right) & \text{if } i \neq j, \\ -X_{ii} + \mathcal{S}\left(X_{ii} - \frac{S_{ii}-W_{ii}}{W_{ii}^2}, \frac{\lambda_{ii}}{W_{ii}^2}\right) & \text{if } i = j, \end{cases} \tag{19}$$

where, as a reminder, $W_{ii} = 1/X_{ii}$. Thus, in this case, the closed form solution for each variable can be computed in $O(1)$ time, so the time complexity for the first Newton direction is further reduced from $O(p^3)$ to $O(p^2)$.

#### 3.1.2 Updating Only a Subset of Variables

In our QUIC algorithm we compute the Newton direction using only a subset of the variables we call the *free* set. We identify these variables in each Newton iteration based on the value of the gradient (we will discuss the details of the selection in Section 3.3). In the following, we define the Newton direction restricted to a subset $J$ of the variables.

**Definition 1** *Let $J$ denote a (symmetric) subset of variables. The Newton direction restricted to $J$ is defined as:*

$$D_J^*(X) \equiv \arg \min_{\substack{D: D_{ij}=0 \\ \forall (i,j) \notin J}} \operatorname{tr}(\nabla g(X)^T D) + \frac{1}{2} \operatorname{vec}(D)^T \nabla^2 g(X) \operatorname{vec}(D) + \|X + D\|_{1,\Lambda}. \tag{20}$$

The cost to compute the Newton direction is thus substantially reduced when the free set $J$ is small, which as we will show in Section 3.3, occurs when the optimal solution of the $\ell_1$-regularized Gaussian MLE is sparse.

### 3.2 Computing the Step Size

Following the computation of the Newton direction $D^* = D_J^*(X)$ (restricted to the subset of variables $J$), we need to find a step size $\alpha \in (0, 1]$ that ensures positive definiteness of the next iterate $X + \alpha D^*$ and leads to a sufficient decrease of the objective function.

We adopt Armijo's rule (Bertsekas, 1995; Tseng and Yun, 2007) and try step-sizes $\alpha \in \{\beta^0, \beta^1, \beta^2, \dots\}$ with a constant decrease rate $0 < \beta < 1$ (typically $\beta = 0.5$), until we find the smallest $k \in \mathbb{N}$ with $\alpha = \beta^k$ such that $X + \alpha D^*$ is (a) positive-definite, and (b) satisfies the following sufficient decrease condition:

$$f(X + \alpha D^*) \leq f(X) + \alpha\sigma\delta, \quad \delta = \mathrm{tr}(\nabla g(X)^T D^*) + \|X + D^*\|_{1,\Lambda} - \|X\|_{1,\Lambda}, \tag{21}$$

where $0 < \sigma < 0.5$. Notice that Condition (21) is a generalized version of Armijo line search rule for $\ell_1$-regularized problems (see (Tseng and Yun, 2007; Yun and Toh, 2011) for the detail). We can verify positive definiteness while we compute the Cholesky factorization (costs $O(p^3)$ flops) needed for the objective function evaluation that requires the computation of $\log\det(X + \alpha D^*)$. The Cholesky factorization dominates the computational cost in the step-size computations. We use the standard convention in convex analysis that $f(X) = +\infty$ when $X$ is not in the effective domain of $f$, i.e., $X$ is not positive definite. Using this convention, (21) enforces positive definiteness of $X + \alpha D^*$. Condition (21) has been proposed in Tseng and Yun (2007); Yun and Toh (2011) to ensure that the objective function value not only decreases but decreases by a certain amount $\alpha\sigma\delta$, where $\delta$ measures the closeness of the current solution to the global optimal. Our convergence proofs presented in Section 4 rely on this sufficient decrease condition.

In the rest of this section we present several lemmas about the step size computation. The reader mostly interested in the algorithm description may skip forward to Section 3.3 and revisit the details afterwards.

We start out by proving three important properties that we call (P1–P3) regarding the line search procedure governed by (21):

P1. The condition (21) is satisfied for some (sufficiently small) $\alpha$, establishing that the algorithm does not enter into an infinite line search step. We note that in Proposition 3 below we show that the line search condition (21) can be satisfied for any symmetric matrix $D$ (even one which is not the Newton direction).

P2. For the Newton direction $D^*$, the quantity $\delta$ in (21) is negative, which ensures that the objective function decreases. Moreover, to guarantee that $X_t$ converges to the global optimum, $|\delta|$ should be large enough when the current iterate $X_t$ is far from the optimal solution. In Proposition 4 we will prove the stronger condition that $\delta \leq -(1/M^2)\|D^*\|_F^2$ for some constant $M$. $\|D^*\|_F^2$ can be viewed as a measure of the distance from optimality of the current iterate $X_t$, and this bound ensures that the objective function decrease is proportional to $\|D^*\|_F^2$.

P3. When $X$ is close enough to the global optimum, the step size $\alpha = 1$ will satisfy the line search condition (21). We will show this property in Proposition 5. Moreover, combined with the global convergence of QUIC proved in Theorem 12, this property suggests that after a finite number of iterations $\alpha$ will always be 1; this also implies that eventually only one Cholesky factorization is needed per iteration (to evaluate $\log\det(X + \alpha D^*)$ for computing $f(X + \alpha D)$).

### 3.2.1 Detailed Proofs for P1-3

We first show the following useful property. For any matrices $X, D$, real number $0 \leq \alpha \leq 1$ and $\Lambda \geq 0$ that generates the norm $\|\cdot\|_{1,\Lambda}$, we have

$$\|X + \alpha D\|_{1,\Lambda} = \|\alpha(X + D) + (1 - \alpha)X\|_{1,\Lambda} \leq \alpha\|X + D\|_{1,\Lambda} + (1 - \alpha)\|X\|_{1,\Lambda}. \tag{22}$$

The above inequality can be proved by the convexity of $\|\cdot\|_{1,\Lambda}$, and will be used repeatedly in this paper. Next we show an important property that all the iterates $X_t$ will have eigenvalues bounded away from zero. Since the updates in our algorithm satisfy the line search condition (21), and $\delta$ is always a negative number (see Proposition 4), the function value is always decreasing. It also follows that all the iterates $\{X_t\}_{t=0,1,\ldots}$ belong to the level set $U$ defined by:

$$U = \{X \mid f(X) \leq f(X_0) \text{ and } X \in S^p_{++}\}. \tag{23}$$

**Lemma 2** *The level set $U$ defined in (23) is contained in the set $\{X \mid mI \preceq X \preceq MI\}$ for some constants $m, M > 0$, if we assume that the off-diagonal elements of $\Lambda$ and the diagonal elements of $S$ are positive.*

**Proof** We begin the proof by showing that the largest eigenvalue of any $X \in U$ is bounded by $M$, a constant that depends only on $\Lambda$, $f(X_0)$ and the matrix $S$. We note that $S \succeq 0$ and $X \succ 0$ implies $\operatorname{tr}(SX) \geq 0$ and therefore:

$$f(X_0) \geq f(X) \geq -\log\det X + \|X\|_{1,\Lambda}. \tag{24}$$

Since $\|X\|_2$ is the largest eigenvalue of the $p \times p$ matrix $X$, we have $\log\det X \leq p\log(\|X\|_2)$. Combine with (24) and the fact that the off-diagonal elements of $\Lambda$ are no smaller than some $\lambda > 0$:

$$\lambda \sum_{i \neq j} |X_{ij}| < \|X\|_{1,\Lambda} \leq f(X_0) + p\log(\|X\|_2). \tag{25}$$

Similarly, $\|X\|_{1,\Lambda} \geq 0$ implies that:

$$\operatorname{tr}(SX) < f(X_0) + p\log(\|X\|_2). \tag{26}$$

Next, we introduce $\gamma = \min_i S_{ii}$ and $\beta = \max_{i \neq j} |S_{ij}|$ and split $\operatorname{tr}(SX)$ into diagonal and off-diagonal terms in order to bound it:

$$\operatorname{tr}(SX) = \sum_i S_{ii}X_{ii} + \sum_{i \neq j} S_{ij}X_{ij} \geq \gamma \operatorname{tr}(X) - \beta \sum_{i \neq j} |X_{ij}|.$$

Since $\|X\|_2 \leq \operatorname{tr}(X)$,

$$\gamma\|X\|_2 \leq \gamma \operatorname{tr}(X) \leq \operatorname{tr}(SX) + \beta \sum_{i \neq j} |X_{ij}|.$$

Combine with (25) and (26) to get:

$$\gamma\|X\|_2 \leq (1 + \beta/\lambda)(f(X_0) + p\log(\|X\|_2)). \tag{27}$$

The left hand side of inequality (27), as a function of $\|X\|_2$, grows much faster than the right hand side (note $\gamma > 0$), and therefore $\|X\|_2$ can be upper bounded by $M$, where $M$ depends on the values of $f(X_0)$, $S$ and $\Lambda$.

In order to prove the lower bound, we consider the smallest eigenvalue of $X$ denoted by $a$ and use the upper bound on the other eigenvalues to get:

$$f(X_0) > f(X) > -\log\det X \geq -\log a - (p-1)\log M, \tag{28}$$

which shows that $m = e^{-f(X_0)}M^{-(p-1)}$ is a lower bound for $a$. ■

We note that the conclusion of the lemma also holds if the conditions on $\Lambda$ and $S$ are replaced by only the requirement that the diagonal elements of $\Lambda$ are positive, see Banerjee et al. (2008). We emphasize that Lemma 2 allows the extension of the convergence results to the practically important case when the regularization does not penalize the diagonal, i.e., $\Lambda_{ii} = 0\ \forall i$. In subsequent arguments we will continue to refer to the minimum and maximum eigenvalues $m$ and $M$ established in Lemma 2.

**Proposition 3 (corresponds to Property P1)** *For any $X \succ 0$ and symmetric $D$, there exists an $\bar{\alpha} > 0$ such that for all $\alpha < \bar{\alpha}$, the matrix $X + \alpha D$ satisfies the line search condition* (21)*.*

**Proof** When $\alpha < \sigma_n(X)/\|D\|_2$ (where $\sigma_n(X)$ stands for the smallest eigenvalue of $X$ and $\|D\|_2$ is the induced 2-norm of $D$, i.e., the largest eigenvalue in magnitude of $D$), we have $\|\alpha D\|_2 < \sigma_n(X)$, which implies that $X + \alpha D \succ 0$. So we can write:

$$\begin{aligned} f(X+\alpha D) - f(X) &= g(X+\alpha D) - g(X) + \|X+\alpha D\|_{1,\Lambda} - \|X\|_{1,\Lambda} \\ &\leq g(X+\alpha D) - g(X) + \alpha(\|X+D\|_{1,\Lambda} - \|X\|_{1,\Lambda}), \quad \text{by (22)} \\ &= \alpha\,\mathrm{tr}((\nabla g(X))^T D) + O(\alpha^2) + \alpha(\|X+D\|_{1,\Lambda} - \|X\|_{1,\Lambda}) \\ &= \alpha\delta + O(\alpha^2). \end{aligned}$$

Therefore for any fixed $0 < \sigma < 1$ and sufficiently small $\alpha$, the line search condition (21) must hold. ■

**Proposition 4 (corresponds to Property P2)** *$\delta = \delta_J(X)$ as defined in the line search condition* (21) *satisfies*

$$\delta \leq -(1/\|X\|_2^2)\|D^*\|_F^2 \leq -(1/M^2)\|D^*\|_F^2, \tag{29}$$

*where $M$ is as in Lemma 2.*

**Proof** We first show that $\delta = \delta_J(X)$ in the line search condition (21) satisfies

$$\delta = \mathrm{tr}((\nabla g(X))^T D^*) + \|X + D^*\|_{1,\Lambda} - \|X\|_{1,\Lambda} \leq -\,\mathrm{vec}(D^*)^T \nabla^2 g(X)\,\mathrm{vec}(D^*), \tag{30}$$

where $D^* = D_J^*(X)$ is the minimizer of the $\ell_1$-regularized quadratic approximation defined in (20).

According to the definition of $D^* \equiv D_J^*(X)$ in (20), for all $0 < \alpha < 1$ we have:

$$\begin{aligned}\operatorname{tr}(\nabla g(X)^T D^*) + \frac{1}{2}\operatorname{vec}(D^*)^T \nabla^2 g(X) \operatorname{vec}(D^*) + \|X + D^*\|_{1,\Lambda} \leq \\ \operatorname{tr}(\nabla g(X)^T \alpha D^*) + \frac{1}{2}\operatorname{vec}(\alpha D^*)^T \nabla^2 g(X) \operatorname{vec}(\alpha D^*) + \|X + \alpha D^*\|_{1,\Lambda}. \end{aligned} \tag{31}$$

We combine (31) and 22 to yield:

$$\begin{aligned}\operatorname{tr}(\nabla g(X)^T D^*) + \frac{1}{2}\operatorname{vec}(D^*)^T \nabla^2 g(X) \operatorname{vec}(D^*) + \|X + D^*\|_{1,\Lambda} \leq \\ \alpha \operatorname{tr}(\nabla g(X)^T D^*) + \frac{1}{2}\alpha^2 \operatorname{vec}(D^*)^T \nabla^2 g(X) \operatorname{vec}(D^*) + \alpha\|X + D^*\|_{1,\Lambda} + (1-\alpha)\|X\|_{1,\Lambda}.\end{aligned}$$

Therefore

$$(1-\alpha)[\operatorname{tr}(\nabla g(X)^T D^*) + \|X + D^*\|_{1,\Lambda} - \|X\|_{1,\Lambda}] + \frac{1}{2}(1-\alpha^2)\operatorname{vec}(D^*)^T \nabla^2 g(X) \operatorname{vec}(D^*) \leq 0.$$

Divide both sides by $1 - \alpha > 0$ to get:

$$\operatorname{tr}(\nabla g(X)^T D^*) + \|X + D^*\|_{1,\Lambda} - \|X\|_{1,\Lambda} + \frac{1}{2}(1+\alpha)\operatorname{vec}(D^*)^T \nabla^2 g(X) \operatorname{vec}(D^*) \leq 0.$$

By taking the limit as $\alpha \uparrow 1$, we get:

$$\operatorname{tr}(\nabla g(X)^T D^*) + \|X + D^*\|_{1,\Lambda} - \|X\|_{1,\Lambda} \leq -\operatorname{vec}(D^*)^T \nabla^2 g(X) \operatorname{vec}(D^*),$$

which proves (30).

Since $\nabla^2 g(X) = X^{-1} \otimes X^{-1}$ is positive definite, (30) ensures that $\delta < 0$ for all $X \succ 0$. Since the updates in our algorithm satisfy the line search condition (21), we have established that the function value is decreasing. It also follows that all the iterates $\{X_t\}_{t=0,1,\ldots}$ belong to the level set $U$ defined by (23). Since $\nabla^2 g(X) = X^{-1} \otimes X^{-1}$, the smallest eigenvalue of $\nabla^2 g(X)$ is $1/\|X\|_2^2$, and we combine with Lemma 2 to get (29). ■

The eigenvalues of any iterate $X$ are bounded by Lemma 2, and therefore $\nabla^2 g(X) = X^{-1} \otimes X^{-1}$ is Lipschitz continuous. Next, we prove that $\alpha = 1$ satisfies the line search condition in a neighborhood of the global optimum $X^*$.

**Proposition 5 (corresponds to Property P3)** *Assume that $\nabla^2 g$ is Lipschitz continuous, i.e., $\exists L > 0$ such that $\forall t > 0$ and any symmetric matrix $D$,*

$$\|\nabla^2 g(X + tD) - \nabla^2 g(X)\|_F \leq L\|tD\|_F = tL\|D\|_F. \tag{32}$$

*Then, if $X$ is close enough to $X^*$, the line search condition* (21) *will be satisfied with step size $\alpha = 1$.*

**Proof** We need to derive a bound for the decrease in the objective function value. We define $\tilde{g}(t) = g(X + tD)$, which yields $\tilde{g}''(t) = \operatorname{vec}(D)^T \nabla^2 g(X + tD) \operatorname{vec}(D)$. First, we bound

$|\tilde{g}''(t) - \tilde{g}''(0)|$:

$$\begin{aligned}
|\tilde{g}''(t) - \tilde{g}''(0)| &= |\operatorname{vec}(D)^T(\nabla^2 g(X+tD) - \nabla^2 g(X))\operatorname{vec}(D)| \\
&\leq \|\operatorname{vec}(D)^T(\nabla^2 g(X+tD) - \nabla^2 g(X))\|_2 \|\operatorname{vec}(D)\|_2 \text{ (by Cauchy-Schwartz)} \\
&\leq \|\operatorname{vec}(D)\|_2^2 \|\nabla^2 g(X+tD) - \nabla^2 g(X)\|_2 \qquad \text{(by definition of } \|\cdot\|_2 \text{ norm)} \\
&\leq \|D\|_F^2 \|\nabla^2 g(X+tD) - \nabla^2 g(X)\|_F \quad \text{(since } \|\cdot\|_2 \leq \|\cdot\|_F \text{ for any matrix)} \\
&\leq \|D\|_F^2 tL\|D\|_F \quad \text{by (32)} \\
&= tL\|D\|_F^3.
\end{aligned}$$

Therefore, an upper bound for $\tilde{g}''(t)$:

$$\tilde{g}''(t) \leq \tilde{g}''(0) + tL\|D\|_F^3 = \operatorname{vec}(D)^T \nabla^2 g(X) \operatorname{vec}(D) + tL\|D\|_F^3.$$

Integrate both sides to get

$$\begin{aligned}
\tilde{g}'(t) &\leq \tilde{g}'(0) + t\operatorname{vec}(D)^T \nabla^2 g(X) \operatorname{vec}(D) + \frac{1}{2}t^2 L\|D\|_F^3 \\
&= \operatorname{tr}((\nabla g(X))^T D) + t\operatorname{vec}(D)^T \nabla^2 g(X) \operatorname{vec}(D) + \frac{1}{2}t^2 L\|D\|_F^3.
\end{aligned}$$

Integrate both sides again:

$$\tilde{g}(t) \leq \tilde{g}(0) + t\operatorname{tr}((\nabla g(X))^T D) + \frac{1}{2}t^2 \operatorname{vec}(D)^T \nabla^2 g(X) \operatorname{vec}(D) + \frac{1}{6}t^3 L\|D\|_F^3.$$

Taking $t = 1$ we have

$$\begin{aligned}
g(X+D) &\leq g(X) + \operatorname{tr}(\nabla g(X)^T D) + \frac{1}{2}\operatorname{vec}(D)^T \nabla^2 g(X) \operatorname{vec}(D) + \frac{1}{6}L\|D\|_F^3 \\
f(X+D) &\leq g(X) + \|X\|_{1,\Lambda} + (\operatorname{tr}(\nabla g(X)^T D) + \|X+D\|_{1,\Lambda} - \|X\|_{1,\Lambda}) \\
&\quad + \frac{1}{2}\operatorname{vec}(D)^T \nabla^2 g(X) \operatorname{vec}(D) + \frac{1}{6}L\|D\|_F^3 \\
&\leq f(X) + \delta + \frac{1}{2}\operatorname{vec}(D)^T \nabla^2 g(X) \operatorname{vec}(D) + \frac{1}{6}L\|D\|_F^3 \\
&\leq f(X) + \frac{\delta}{2} + \frac{1}{6}L\|D\|_F^3 \text{ by (30)} \\
&\leq f(X) + (\frac{1}{2} - \frac{1}{6}LM^2\|D\|_F)\delta \text{ (by Proposition 4)} \\
&\leq f(X) + \sigma\delta \text{ (assuming } D \text{ is close to 0).}
\end{aligned}$$

The last inequality holds if $1/2 - LM^2\|D\|_F/6 > \sigma$ which is guaranteed if $X$ is close enough to $X^*$ and consequently $D$ is close to 0 and $\sigma < 0.5$. (Note $\delta < 0$ as well.) In this case the line search condition (21) holds with $\alpha = 1$. ■

### 3.3 Identifying Which Variables to Update

In this section, we use the stationary condition of the Gaussian MLE problem to select a subset of variables to update in any Newton direction computation. Specifically, we partition the variables into *free* and *fixed* sets based on the value of the gradient at the start of the outer loop that computes the Newton direction. We define the *free* set $S_{free}$ and *fixed* set $S_{fixed}$ as:

$$\begin{aligned} &X_{ij} \in S_{fixed} \text{ if } |\nabla_{ij} g(X)| \leq \lambda_{ij}, \text{ and } X_{ij} = 0, \\ &X_{ij} \in S_{free} \text{ otherwise.} \end{aligned} \tag{33}$$

We will now show that a Newton update restricted to the *fixed set* of variables would not change any of the coordinates in that set. In brief, the gradient condition $|\nabla_{ij} g(X)| \leq \lambda_{ij}$ entails that the inner coordinate descent steps, according to the update in (16), would set these coordinates to zero, so they would not change since they were zero to begin with.

To derive the optimality condition, we begin by introducing the minimum-norm subgradient of $f$ and relate it to the optimal solution $X^*$ of (3).

**Definition 6** *The minimum-norm subgradient* $\text{grad}^S_{ij} f(X)$ *is defined as follows:*

$$\text{grad}^S_{ij} f(X) = \begin{cases} \nabla_{ij} g(X) + \lambda_{ij} & \text{if } X_{ij} > 0, \\ \nabla_{ij} g(X) - \lambda_{ij} & \text{if } X_{ij} < 0, \\ \text{sign}(\nabla_{ij} g(X)) \max(|\nabla_{ij} g(X)| - \lambda_{ij}, 0) & \text{if } X_{ij} = 0. \end{cases}$$

**Lemma 7** *For any index set* $J$, $grad^S_{ij} f(X) = 0 \ \forall (i,j) \in J$ *if and only if* $\Delta^* = 0$ *is a solution of the following optimization problem:*

$$\arg\min_{\Delta} f(X + \Delta) \text{ such that } \Delta_{ij} = 0 \ \ \forall (i,j) \notin J. \tag{34}$$

**Proof** Any optimal solution $\Delta^*$ for (34) must satisfy the following, for all $(i,j) \in J$,

$$\nabla_{ij} g(X + \Delta^*) \begin{cases} = -\lambda_{ij} & \text{if } X_{ij} + \Delta^*_{ij} > 0, \\ = \lambda_{ij} & \text{if } X_{ij} + \Delta^*_{ij} < 0, \\ \in [-\lambda_{ij} \ \ \lambda_{ij}] & \text{if } X_{ij} + \Delta^*_{ij} = 0. \end{cases} \tag{35}$$

It can be seen immediately that $\Delta^* = 0$ satisfies (35) if and only if $\text{grad}^S_{ij} f(X) = 0$ for all $(i,j) \in J$. ■

In our case, $\nabla g(X) = S - X^{-1}$ and therefore

$$\text{grad}^S_{ij} f(X) = \begin{cases} (S - X^{-1})_{ij} + \lambda_{ij} & \text{if } X_{ij} > 0, \\ (S - X^{-1})_{ij} - \lambda_{ij} & \text{if } X_{ij} < 0, \\ \text{sign}((S - X^{-1})_{ij}) \max(|(S - X^{-1})_{ij}| - \lambda_{ij}, 0) & \text{if } X_{ij} = 0. \end{cases}$$

Our definition of the *fixed* and *free* sets is clearly motivated by the minimum norm subgradient. A variable $X_{ij}$ belongs to the *fixed* set if and only if $X_{ij} = 0$ and $\text{grad}^S_{ij} f(X) = 0$.

Therefore, taking $J = S_{fixed}$ in Lemma 7, we can show that for any $X_t$ and corresponding fixed and free sets $S_{fixed}$ and $S_{free}$ as defined by (33), $\Delta^* = 0$ is the solution of the following optimization problem:

$$\arg\min_{\Delta} f(X_t + \Delta) \text{ such that } \Delta_{ij} = 0 \ \ \forall (i,j) \in S_{free}.$$

Based on the above property, if we perform block coordinate descent restricted to the fixed set, then no updates would occur. We then perform the coordinate descent updates restricted to only the free set to find the Newton direction. With this modification, the number of variables over which we perform the coordinate descent update (16) can be potentially reduced from $p^2$ to the number of non-zeros in $X_t$. When the solution is sparse (depending on the value of $\Lambda$) the number of free variables can be much smaller than $p^2$ and we can obtain huge computational gains as a result. In essence, we very efficiently select a subset of the coordinates that need to be updated.

The attractive facet of this modification is that it leverages sparsity of the solution and intermediate iterates in a manner that falls within the block coordinate descent framework of Tseng and Yun (2007). The index sets $J_0, J_1, \ldots$ corresponding to the block coordinate descent steps in the general setting of Tseng and Yun (2007)[p. 392] need to satisfy a Gauss-Seidel type of condition:

$$\bigcup_{j=0,\ldots,T-1} J_{t+j} \supseteq \mathcal{N} \ \ \forall t = 1, 2, \ldots \tag{36}$$

for some fixed $T$, where $\mathcal{N}$ denotes the full index set. In our framework $J_0, J_2, \ldots$ denote the fixed sets at various iterations, and $J_1, J_3, \ldots$ denote the free sets. Since $J_{2i}$ and $J_{2i+1}$ is a partitioning of $\mathcal{N}$ the choice $T = 3$ will suffice. But will the size of the free set be small? We initialize $X_0$ to a diagonal matrix, which is sparse. The following lemma shows that after a *finite* number of iterations, the iterates $X_t$ will have a similar sparsity pattern as the limit $X^*$. Lemma 8 is actually an immediate consequence of Lemma 14 in Section 4.

**Lemma 8** *Assume that $\{X_t\}$ converges to $X^*$, the optimal solution of (3). If for some index pair $(i,j)$, $|\nabla_{ij} g(X^*)| < \lambda_{ij}$ (so that $X^*_{ij} = 0$), then there exists a constant $\bar{t} > 0$ such that for all $t > \bar{t}$, the iterates $X_t$ satisfy*

$$|\nabla_{ij} g(X_t)| < \lambda_{ij} \quad \textit{and} \ (X_t)_{ij} = 0. \tag{37}$$

Note that $|\nabla_{ij} g(X^*)| < \lambda_{ij}$ implies $X^*_{ij} = 0$ from the optimality condition of (3). This theorem shows that after $\bar{t}$-th iteration we can ignore all the indexes that satisfies (37), and in practice we can use (37) as a criterion for identifying the *fixed* set. A similar variable selection strategy is used in SVM (so called shrinking) and $\ell_1$-regularized logistic regression problems as mentioned in Yuan et al. (2010). In our experiments, we demonstrate that this strategy reduces the size of the free set very quickly.

Lemma 8 suggests that QUIC can identify the zero pattern in finite steps. As we will prove later, QUIC has an asymptotic quadratic convergence rate and therefore once the zero pattern is correctly recognized, the algorithm often converges in a few additional iterations. Hence, the time needed to converge to the global optimum is not much more than the time needed to arrive at the zero pattern of the inverse covariance matrix.

### 3.4 The Block-Diagonal Structure of $X^*$

It has been shown recently (Mazumder and Hastie, 2012; Witten et al., 2011) that when the thresholded covariance matrix $E$ defined by $E_{ij} = \mathcal{S}(S_{ij}, \lambda) = \text{sign}(S_{ij}) \max(|S_{ij}| - \lambda, 0)$ has the following block-diagonal structure:

$$E = \begin{bmatrix} E_1 & 0 & \dots & 0 \\ 0 & E_2 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & E_k \end{bmatrix}, \tag{38}$$

then the solution $X^*$ of the inverse covariance estimation problem (2) also has the same block-diagonal structure:

$$X^* = \begin{bmatrix} X_1^* & 0 & \dots & 0 \\ 0 & X_2^* & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & X_k^* \end{bmatrix}.$$

This result can be extended to the case when the elements are penalized differently, i.e., $\lambda_{ij}$'s are different. Then, if $E_{ij} = \mathcal{S}(S_{ij}, \lambda_{ij})$ is block diagonal, so is the solution $X^*$ of (3), see Hsieh et al. (2012). Thus each $X_i^*$ can be computed independently. Based on this observation one can decompose the problem into sub-problems of smaller sizes, which can be solved much faster. In the following, we show that our updating rule and *fixed/free* set selection technique can automatically detect this block-diagonal structure for free.

Recall that we have a closed form solution in the first iteration when the input is a diagonal matrix. Based on (19), since $X_{ij} = 0$ for all $i \neq j$ in this step, we have

$$D_{ij} = X_{ii}X_{jj}\mathcal{S}(-S_{ij}, \lambda_{ij}) = -X_{ii}X_{jj}\mathcal{S}(S_{ij}, \lambda_{ij}) \quad \text{for all } i \neq j.$$

We see that after the first iteration the nonzero pattern of $X$ will be exactly the same as the nonzero pattern of the thresholded covariance matrix $E$ as depicted in (38). In order to establish that the same is true at each subsequent step, we complete our argument using induction, by showing that the non-zero structure is preserved.

More precisely, we show that the off-diagonal blocks always belong to the *fixed* set if $|S_{ij}| \leq \lambda_{ij}$. Recall the definition of the *fixed* set in (33). We need to check whether $|\nabla_{ij} g(X)| \leq \lambda_{ij}$ for all $(i, j)$ in the off-diagonal blocks of $E$, whenever $X$ has the same block-diagonal structure as $E$. Taking the inverse preserves the diagonal structure, and therefore $\nabla_{ij} g(X) = S_{ij} - X_{ij}^{-1} = S_{ij}$ for all such $(i, j)$. We conclude noting that $E_{ij} = 0$ implies that $|\nabla_{ij} g(X)| \leq \lambda_{ij}$, meaning that $(i, j)$ will belong to the *fixed* set.

We decompose the matrix into smaller blocks prior to running Cholesky factorization to avoid the $O(p^3)$ time complexity on the whole problem. The connected components of $X$ can be detected in $O(\|X\|_0)$ time, which is very efficient when $X$ is sparse. A detailed description of QUIC is presented in Algorithm 2.

## 4. Convergence Analysis

In Section 3, we introduced the main ideas behind our QUIC algorithm. In this section, we first prove that QUIC converges to the global optimum, and then show that the convergence

**Algorithm 2:** QUadratic approximation for sparse Inverse Covariance estimation (*QUIC*)

**Input** : Empirical covariance matrix $S$ (positive semi-definite $p \times p$), regularization parameter matrix $\Lambda$, initial $X_0 \succ 0$, parameters $0 < \sigma < 0.5,\ 0 < \beta < 1$

**Output**: Sequence of $X_t$ converging to $\arg\min_{X \succ 0} f(X)$, where $f(X) = g(X) + h(X)$, where $g(X) = -\log\det X + \mathrm{tr}(SX), h(X) = \|X\|_{1,\Lambda}$.

Compute $W_0 = X_0^{-1}$.
**for** $t = 0, 1, \ldots$ **do**
  $D = 0$, $U = 0$
  **while** *not converged* **do**
    Partition the variables into fixed and free sets:
    $S_{fixed} := \{(i,j) \mid |\nabla_{ij} g(X_t)| \leq \lambda_{ij} \text{ and } (X_t)_{ij} = 0\}$,
    $S_{free} := \{(i,j) \mid |\nabla_{ij} g(X_t)| > \lambda_{ij} \text{ or } (X_t)_{ij} \neq 0\}$.
    **for** $(i,j) \in S_{free}$ **do**
      $a = w_{ij}^2 + w_{ii} w_{jj},\ b = s_{ij} - w_{ij} + \mathbf{w}_{\cdot i}^T \mathbf{u}_{\cdot j},\ c = x_{ij} + d_{ij}$
      $\mu = -c + \mathcal{S}(c - b/a, \lambda_{ij}/a)$
      $d_{ij} \leftarrow d_{ij} + \mu,\ \mathbf{u}_{i\cdot} \leftarrow \mathbf{u}_{i\cdot} + \mu \mathbf{w}_{j\cdot},\ \mathbf{u}_{j\cdot} \leftarrow \mathbf{u}_{j\cdot} + \mu \mathbf{w}_{i\cdot}$
    **end**
  **end**
  **for** $\alpha = 1, \beta, \beta^2, \ldots$ **do**
    Compute the Cholesky factorization $LL^T = X_t + \alpha D$.
    **if** $X_t + \alpha D \succ 0$ **then**
      Compute $f(X_t + \alpha D)$ from $L$ and $X_t + \alpha D$
      **if** $f(X_t + \alpha D) \leq f(X_t) + \alpha\sigma\left[\mathrm{tr}(\nabla g(X_t) D) + \|X_t + D\|_{1,\Lambda} - \|X\|_{1,\Lambda}\right]$ **then**
        break
      **end**
    **end**
  **end**
  $X_{t+1} = X_t + \alpha D$
  Compute $W_{t+1} = X_{t+1}^{-1}$ reusing the Cholesky factor.
**end**

rate is quadratic. Banerjee et al. (2008) showed that for the special case where $\Lambda_{ij} = \lambda$ the optimization problem (2) has a unique global optimum and that the eigenvalues of the primal optimal solution $X^*$ are bounded. In the following, we show this result for more general $\Lambda$ where only the off-diagonal elements need to be positive.

**Theorem 9** *There exists a unique minimizer $X^*$ for the optimization problem* (3)*, where $\lambda_{ij} > 0$ for $i \neq j$, and $\lambda_{ij} \geq 0$.*

**Proof** According to Lemma 2, the level set $U$ defined in (23) contains all the iterates, and it is in turn contained in the compact set $S \equiv \{X \mid mI \preceq X \preceq MI\}$. According to the Weierstrass extreme value theorem (Apostol, 1974), any continuous function in a compact set attains its minimum. Furthermore, $\nabla^2 g(X) = X^{-1} \otimes X^{-1}$ implies $\nabla^2 g(X) \succeq M^{-2} I$.

Since $\|X\|_{1,\Lambda}$ is convex and $-\log\det(X)$ is strongly convex, we have that $f(X)$ is strongly convex on the compact set $S$, and therefore the minimizer $X^*$ is unique (Apostol, 1974). ■

### 4.1 Convergence Guarantee

In order to show that QUIC converges to the optimal solution, we consider a more general setting of the quadratic approximation algorithm: at each iteration, the iterate $Y_t$ is updated by $Y_{t+1} = Y_t + \alpha_t D^*_{J_t}(Y_t)$ where $J_t$ is a subset of variables chosen to update at iteration $t$, $D^*_{J_t}(Y_t)$ is the Newton direction restricted to $J_t$ defined by (20), and $\alpha_t$ is the step size selected by the Armijo rule given in Section 3.2. The algorithm is summarized in Algorithm 3. Similar to the block coordinate descent framework of Tseng and Yun (2007), we assume the index set $J_t$ satisfies a Gauss-Seidel type of condition:

$$\bigcup_{j=0,\ldots,T-1} J_{t+j} \supseteq \mathcal{N} \quad \forall t = 1, 2, \ldots. \tag{39}$$

**Algorithm 3:** General Block Quadratic Approximation method for Sparse Inverse Covariance Estimation

**Input** : Empirical covariance matrix $S$ (positive semi-definite $p \times p$), regularization parameter matrix $\Lambda$, initial $Y_0$, inner stopping tolerance $\epsilon$

**Output**: Sequence of $Y_t$.

**for** $t = 0, 1, \ldots$ **do**
  Generate a variable subset $J_t$.
  Compute the Newton direction $D^*_t \equiv D^*_{J_t}(Y_t)$ by (20).
  Compute the step-size $\alpha_t$ using the *Armijo*-rule based step-size selection in (21).
  Update $Y_{t+1} = Y_t + \alpha_t D^*_t$.
**end**

In QUIC, $J_0, J_2, \ldots$ denote the fixed sets and $J_1, J_3, \ldots$ denote the free sets. If $\{X_t\}_{t=0,1,2,\ldots}$ denotes the sequence generated by QUIC, then

$$Y_0 = Y_1 = X_0, Y_2 = Y_3 = X_1, \ldots, Y_{2i} = Y_{2i+1} = X_i.$$

Moreover, since each $J_{2i}$ and $J_{2i+1}$ is a partitioning of $\mathcal{N}$, the choice $T = 3$ will satisfy (39). In the rest of this section, we show that $\{Y_t\}_{t=0,1,2,\ldots}$ converges to the global optimum, thus $\{X_t\}_{t=0,1,2,\ldots}$ generated by QUIC also converges to the global optimum.

Our first step towards the convergence proof is a lemma on convergent subsequences.

**Lemma 10** *For any convergent subsequence $Y_{s_t} \to \bar{Y}$ where $\bar{Y}$ is a limit point, we have $D^*_{s_t} \equiv D^*_{J_{s_t}}(Y_{s_t}) \to 0$.*

**Proof** The objective value decreases according to the line search condition (21) and Proposition 4. According to Lemma 2, $f(Y_{s_t})$ cannot converge to negative infinity, so $f(Y_{s_t}) - f(Y_{s_{t+1}}) \to 0$. The line search condition (21) implies that $\alpha_{s_t}\delta_{s_t} \to 0$.

We proceed to prove the statement by contradiction. If $D^*_{s_t}$ does not converge to 0, then there exists an infinite index set $\mathcal{T} \subseteq \{s_1, s_2, \ldots\}$ and $\eta > 0$ such that $\|D^*_t\|_F > \eta$ for all $t \in \mathcal{T}$. According to Proposition 4, $\delta_{s_t}$ is bounded away from 0, therefore $\delta_{s_t} \not\to 0$, while $\alpha_{s_t} \to 0$. We can assume without loss of generality that $\alpha_{s_t} < 1\ \forall t$, that is the line search condition is not satisfied in the first attempt. We will work in this index set $\mathcal{T}$ in the derivations that follow.

The line search step size $\alpha_t < 1$ ($t \in \mathcal{T}$) satisfies (21), but $\overline{\alpha}_t = \alpha_t/\beta$ does not satisfy (21) by the minimality of our line search procedure. So we have:

$$f(Y_t + \overline{\alpha}_t D^*_t) - f(Y_t) > \sigma \overline{\alpha}_t \delta_t. \tag{40}$$

If $Y_t + \overline{\alpha}_t D^*_t$ is not positive definite, then as is standard, $f(Y_t + \overline{\alpha}_t D^*_t) = \infty$, so (40) still holds. We expand (40) and apply 22 to get

$$\begin{aligned}\sigma\overline{\alpha}_t\delta_t &\le g(Y_t + \overline{\alpha}_t D^*_t) - g(Y_t) + \|Y_t + \overline{\alpha}_t D^*_t\|_{1,\Lambda} - \|Y_t\|_{1,\Lambda} \\ &\le g(Y_t + \overline{\alpha}_t D^*_t) - g(Y_t) + \overline{\alpha}_t(\|Y_t + D^*_t\|_{1,\Lambda} - \|Y_t\|_{1,\Lambda}), \forall t \in \mathcal{T}.\end{aligned}$$

By the definition of $\delta_t$, we have:

$$\begin{aligned}\sigma\delta_t &\le \frac{g(Y_t + \overline{\alpha}_t D^*_t) - g(Y_t)}{\overline{\alpha}_t} + \delta_t - \operatorname{tr}(\nabla g(Y_t)^T D^*_t), \\ (1-\sigma)(-\delta_t) &\le \frac{g(Y_t + \overline{\alpha}_t D^*_t) - g(Y_t)}{\overline{\alpha}_t} - \operatorname{tr}(\nabla g(Y_t)^T D^*_t).\end{aligned}$$

By Proposition 4 we have $\delta_t \le -(1/M^2)\|D^*_t\|_F^2$, so using $\|D^*_t\|$ to denote $\|D^*_t\|_F$ for the rest of the proof, we get

$$\begin{aligned}(1-\sigma)M^{-2}\|D^*_t\|^2 &\le \frac{g(Y_t + \overline{\alpha}_t D^*_t) - g(Y_t)}{\overline{\alpha}_t} - \operatorname{tr}(\nabla g(Y_t)^T D^*_t) \\ (1-\sigma)M^{-2}\|D^*_t\| &\le \frac{g\left(Y_t + \overline{\alpha}_t\|D^*_t\|\frac{D^*_t}{\|D^*_t\|}\right) - g(Y_t)}{\overline{\alpha}_t\|D^*_t\|} - \operatorname{tr}\left(\nabla g(Y_t)^T \frac{D^*_t}{\|D^*_t\|}\right).\end{aligned}$$

We set $\hat{\alpha}_t = \overline{\alpha}_t\|D^*_t\|$. Since $\|D^*_t\| > \eta$ for all $t \in \mathcal{T}$ we have:

$$\begin{aligned}(1-\sigma)M^{-2}\eta &< \frac{g\left(Y_t + \hat{\alpha}_t\frac{D^*_t}{\|D^*_t\|}\right) - g(Y_t)}{\hat{\alpha}_t} - \operatorname{tr}\left(\nabla g(Y_t)^T \frac{D^*_t}{\|D^*_t\|}\right) \\ &= \frac{\hat{\alpha}_t \operatorname{tr}\left(\nabla g(Y_t)\frac{D^*_t}{\|D^*_t\|}\right) + O(\hat{\alpha}_t^2)}{\hat{\alpha}_t} - \operatorname{tr}\left(\nabla g(Y_t)^T \frac{D^*_t}{\|D^*_t\|}\right) \\ &= O(\hat{\alpha}_t).\end{aligned} \tag{41}$$

Again, by Proposition 4,

$$-\alpha_t\delta_t \ge \alpha_t M^{-2}\|D^*_t\|^2 > M^{-2}\alpha_t\|D^*_t\|\eta.$$

Since $\{\alpha_t\delta_t\}_t \to 0$, it follows that $\{\alpha_t\|D^*_t\|\}_t \to 0$ and $\{\hat{\alpha}_t\}_t \to 0$. Taking limit of (41) as $t \in \mathcal{T}$ and $t \to \infty$, we have

$$(1-\sigma)M^{-2}\eta \le 0,$$

a contradiction, finishing the proof. ■

Now that we have proved that $D^*_{J_t}$ converges to zero for the converging subsequence, we next show that $D^*_J$ is closely related to the minimum-norm subgradient $\text{grad}^S f(Y)$ (see Definition 6), which in turn is an indicator of optimality as proved in Lemma 7.

**Lemma 11** *For any index set $J$ and positive definite $Y$, $D^*_J(Y) = 0$ if and only if $\text{grad}^S_{ij} f(Y) = 0$ for all $(i,j) \in J$.*

**Proof** The optimality condition of (20) can be written as

$$\nabla_{ij} g(X) + (\nabla^2 g(X)\,\text{vec}(D))_{ij} \begin{cases} = -\lambda & \text{if } X_{ij} + D_{ij} > 0 \\ = \lambda & \text{if } X_{ij} + D_{ij} < 0 \\ \in [-\lambda,\ \lambda] & \text{if } X_{ij} + D_{ij} = 0, \end{cases} \qquad \forall (i,j) \in J. \tag{42}$$

$D^*_J(Y) = 0$ if and only if $D^* = 0$ satisfies (42), and this condition is equivalent to (35) restricted to $(i,j) \in J$, which in turn is equivalent to the optimality condition of $f$. Therefore $D^*_J(Y) = 0$ iff $\text{grad}^S_{ij} f(Y) = 0$ for all $(i,j) \in J$. ■

Based on these lemmas, we are now able to prove our main convergence theorem.

**Theorem 12** *Algorithm 3 converges to the unique global optimum $Y^*$.*

**Proof** Since all the iterates $Y_t$ are in a compact set (as shown in Lemma 2), there exists a subsequence $\{Y_t\}_{\mathcal{T}}$ that converges to a limit point $\bar{Y}$. Since the cardinality of each index set $J_t$ selected is finite, we can further assume that $J_t = \bar{J}_0$ for all $t \in \bar{\mathcal{T}}$, where $\bar{\mathcal{T}}$ is a subsequence of $\mathcal{T}$. From Lemma 10, $D^*_{\bar{J}_0}(Y_t) \to 0$. By continuity of $\nabla g(Y)$ and $\nabla^2 g(Y)$, it is easy to show that $D^*_{\bar{J}_0}(Y_t) \to D^*_{\bar{J}_0}(\bar{Y})$. Therefore $D^*_{\bar{J}_0}(\bar{Y}) = 0$. Based on Lemma 11, we have

$$\text{grad}^S_{ij} f(Y) = 0 \quad \text{for all } (i,j) \in \bar{J}_0.$$

Furthermore, $\{D^*_{\bar{J}_0}(Y_t)\}_{\mathcal{T}} \to 0$ and $\|Y_t - Y_{t+1}\|_F \le \|D^*_{\bar{J}_0}(Y_t)\|_F$, so $\{Y_{t+1}\}_{t \in \mathcal{T}}$ also converges to $\bar{Y}$. By considering a subsequence of $\mathcal{T}$ if necessary, we can further assume that $J_{t+1} = \bar{J}_1$ for all $t \in \mathcal{T}$. By the same argument, we can show that $\{D^*_{J_{t+1}}(Y_t)\}_{\mathcal{T}} \to 0$, so $D^*_{\bar{J}_1}(\bar{Y}) = 0$. Similarly, we can show that $D^*_{\bar{J}_t}(\bar{Y}) = 0\ \forall t = 0, \ldots, T-1$ can be assumed for an appropriate subset of $\mathcal{T}$. With assumption (39) and Lemma 11 we have

$$\text{grad}^S_{ij} f(\bar{Y}) = 0\ \forall i, j. \tag{43}$$

Using Lemma 7 with $J$ as the set of all variables, we can show that (43) implies that $\bar{Y}$ is the global optimum. ■

It is straightforward to generalize Theorem 12 to prove the convergence of block coordinate descent when the Hessian $\nabla^2 g(X)$ is replaced by another positive definite matrix. The proof strategies are similar to Tseng and Yun (2007) and we omit the detailed derivation in this paper.

### 4.2 Asymptotic Convergence Rate

**Newton methods on constrained minimization problems:**
The convergence rate of the Newton method on bounded constrained minimization has been studied in Levitin and Polyak (1966) and Dunn (1980). Here, we briefly mention their results.

Assume we want to solve a constrained minimization problem

$$\min_{\mathbf{x}\in\Omega} F(\mathbf{x}),$$

where $\Omega$ is a nonempty subset of $R^n$ denoting the constraint set and $F : R^n \to R$ has a second derivative $\nabla^2 F(\mathbf{x})$. Then beginning from $\mathbf{x}_0$, the natural Newton updates entail computing the $(k+1)$-st iterate $\mathbf{x}_{k+1}$ as

$$\mathbf{x}_{k+1} = \arg\min_{\mathbf{x}\in\Omega} \nabla F(\mathbf{x}_k)^T(\mathbf{x}-\mathbf{x}_k) + \frac{1}{2}(\mathbf{x}-\mathbf{x}_k)^T\nabla^2 F(\mathbf{x}_k)(\mathbf{x}-\mathbf{x}_k). \tag{44}$$

For simplicity, we assume that $F$ is strictly convex, and has a unique minimizer $\mathbf{x}^*$ in $\Omega$. Then the following theorem holds.

**Theorem 13 (Theorem 3.1 in Dunn, 1980)** *Assume $F$ is strictly convex, has a unique minimizer $\mathbf{x}^*$ in $\Omega$, and that $\nabla^2 F(\mathbf{x})$ is Lipschitz continuous. Then for all $\mathbf{x}_0$ sufficiently close to $\mathbf{x}^*$, the sequence $\{\mathbf{x}_k\}$ generated by (44) converges quadratically to $\mathbf{x}^*$.*

This theorem is proved in Dunn (1980). In our case, the objective function $f(X)$ is non-smooth so Theorem 13 does not directly apply. Instead, we will first show that after a finite number of iterations the sign of the iterates $\{X_t\}$ generated by QUIC will not change, so that we can then use Theorem 13 to establish asymptotic quadratic convergence.

**Quadratic convergence rate for** QUIC**:**
Unlike as in (44), our Algorithm 3 does not perform an unrestricted Newton update: it iteratively selects subsets of variables $\{J_t\}_{t=1,\ldots}$ (*fixed* and *free* sets), and computes the Newton direction restricted to the free sets. In the following, we show that the sequence $\{X_t\}_{t=1,2,\ldots}$ generated by QUIC does ultimately converge quadratically to the global optimum.

Assume $X^*$ is the optimal solution, then we can divide the index set with $\lambda_{ij} \neq 0$ into three subsets:

$$P = \{(i,j) \mid X^*_{ij} > 0\}, \quad N = \{(i,j) \mid X^*_{ij} < 0\}, \quad Z = \{(i,j) \mid X^*_{ij} = 0\}. \tag{45}$$

From the optimality condition for $X^*$,

$$\nabla_{ij} g(X^*) \begin{cases} = -\lambda_{ij} & \text{if } (i,j) \in P, \\ = \lambda_{ij} & \text{if } (i,j) \in N, \\ \in [-\lambda_{ij}, \lambda_{ij}] & \text{if } (i,j) \in Z. \end{cases} \tag{46}$$

**Lemma 14** *Assume that the sequence $\{X_t\}$ converges to the global optimum $X^*$. Then there exists a $\bar{t}$ such that for all $t > \bar{t}$,*

$$(X_t)_{ij} \begin{cases} \geq 0 & \text{if } (i,j) \in P, \\ \leq 0 & \text{if } (i,j) \in N, \\ = 0 & \text{if } (i,j) \in Z. \end{cases} \tag{47}$$

**Proof** We prove the case for $(i,j) \in P$ by contradiction, the other two cases can be handled similarly. If we cannot find a $\bar{t}$ satisfying the first condition in (47), then there exists an infinite subsequence $\{X_{a_t}\}$ such that for each $a_t$ there exists a $(i,j) \in P$ such that $(X_{a_t})_{ij} < 0$. Since the cardinality of $P$ is finite, we can further find a specific pair $(i,j) \in P$ such that $(X_{s_t})_{ij} < 0$ for all $s_t$, where $s_t$ is a subsequence of $a_t$. We consider the update from $X_{s_t-1}$ to $X_{s_t}$. From Lemma 5, we can assume that $s_t$ is large enough so that the step size equals 1, therefore $X_{s_t} = X_{s_t-1} + D^*(X_{s_t-1})$ where $D^*(X_{s_t-1})$ is defined in (20). Since $(X_{s_t})_{ij} = (X_{s_t-1})_{ij} + (D^*(X_{s_t-1}))_{ij} < 0$, from the optimality condition of (20) we have

$$\left(\nabla g(X_{s_t-1}) + \nabla^2 g(X_{s_t-1}) \operatorname{vec}(D^*(X_{s_t-1}))\right)_{ij} = \lambda_{ij}. \tag{48}$$

Since $D^*(X_{s_t-1})$ converges to 0, (48) implies that $\{\nabla_{ij} g(X_{s_t-1})\}$ will converge to $\lambda_{ij}$. However, (46) implies $\nabla_{ij} g(X^*) = -\lambda_{ij}$, and by the continuity of $\nabla g$ we get that $\{\nabla_{ij} g(X_t)\}$ converges to $\nabla_{ij} g(X^*) = -\lambda_{ij}$, a contradiction, finishing the proof. ■

The following lemma shows that the coordinates from the fixed set remain zero after a finite number of iterations.

**Lemma 15** *Assume $X_t \to X^*$. There exists a $\bar{t} > 0$ such that variables in $P$ or $N$ will not be selected to be in the fixed set $S_{fixed}$, when $t > \bar{t}$. That is,*

$$S_{fixed} \subseteq Z.$$

**Proof** Since $X_t$ converges to $X^*$, $(X_t)_{ij}$ converges to $X^*_{ij} > 0$ if $(i,j) \in P$ and to $X^*_{ij} < 0$ if $(i,j) \in N$. Recall that $(i,j)$ belongs to the fixed set only if $(X_t)_{ij} = 0$. When $t$ is large enough, $(X_t)_{ij} \neq 0$ when $X_t \in P \cup N$, therefore $P$ and $N$ will be disjoint from the fixed set. Moreover, by the definition of the fixed set (33), indexes with $\lambda_{ij} = 0$ will never be selected. We proved that the fixed set will be a subset of $Z$ when $t$ is large enough. ■

**Theorem 16** *The sequence $\{X_t\}$ generated by the* QUIC *algorithm converges quadratically to $X^*$, that is for some constant $\kappa > 0$,*

$$\lim_{t \to \infty} \frac{\|X_{t+1} - X^*\|_F}{\|X_t - X^*\|_F^2} = \kappa.$$

**Proof** First, if the index sets $P, N$ and $Z$ (related to the optimal solution) are given, the optimum of (2) is the same as the optimum of the following constrained minimization problem:

$$\begin{aligned} \min_X \quad & -\log\det(X) + \operatorname{tr}(SX) + \sum_{(i,j)\in P} \lambda_{ij} X_{ij} - \sum_{(i,j)\in N} \lambda_{ij} X_{ij} \\ \text{s.t.} \quad & X_{ij} \geq 0 \ \ \forall (i,j) \in P, \quad X_{ij} \leq 0 \ \ \forall (i,j) \in N, \quad X_{ij} = 0 \ \ \forall (i,j) \in Z. \end{aligned} \tag{49}$$

In the following, we show that when $t$ is large enough, QUIC solves the minimization problem described by (49).

1. The constraints in (49) are satisfied by QUIC iterates after a finite number of steps, as shown in Lemma 14. Thus, the $\ell_1$-regularized Gaussian MLE (3) is equivalent to the smooth constrained objective (49), since the constraints in (49) are satisfied when solving (3).

2. Since the optimization problem in (49) is smooth, it can be solved using constrained Newton updates as in (44). The QUIC update direction $D_J^*(X_t)$ is restricted to a set of free variables in $J$. This is exactly equal to the unrestricted Newton update as in (44), after a finite number of steps, as established by Lemma 15. In particular, at each iteration the fixed set is contained in $Z$, which is the set which always satisfies $(D_t^*)_Z = 0$ for large enough $t$.

3. Moreover, by Lemma 5 the step size is $\alpha = 1$ when $t$ is large enough.

Therefore our algorithm is equivalent to the constrained Newton method in (44), which in turn converges quadratically to the optimal solution of (49). Since the revised problem (49) and our original problem (3) has the same minimum, we have shown that QUIC converges quadratically to the optimum of (3). ■

Note that the constant $\kappa$ is an increasing function of the Lipschitz constant of $\nabla^2 g(X)$ (as shown in Dunn, 1980), which is related to the quality of quadratic approximation. We have shown in Lemma 2 that $mI \preceq X \preceq MI$, therefore the Lipschitz constant of $\nabla^2 g(X) = X^{-1} \otimes X^{-1}$ is also upper bounded.

In the next section, we show that this asymptotic convergence behavior of QUIC is corroborated empirically as well.

## 5. Experimental Results

We begin this section by comparing QUIC to other methods on synthetic and real data sets. Then, we present some empirical analysis of QUIC regarding the use of approximate Newton directions and effects of parameterization.

### 5.1 Comparisons with Other Methods

We now compare the performance of QUIC on both synthetic and real data sets to other state-of-the-art methods. We have implemented QUIC in C++ with MATLAB interface, and all experiments were executed on 2.83GHz Xeon X5440 machines with 32G RAM and Linux OS.

We include the following algorithms in our comparisons:

- ALM: the Alternating Linearization Method proposed by Scheinberg et al. (2010). We use their MATLAB source code for the experiments.

- ADMM: another implementation of the alternating linearization method implemented by Boyd et al. (2012). The MATLAB code can be downloaded from `http://www.stanford.edu/~boyd/papers/admm/`. We found that the default parameters (which we note are independent of the regularization penalty) yielded slow convergence; we

set the augmented Lagrangian parameter to $\rho = 50$ and the over-relaxation parameter to $\alpha = 1.5$. These parameters achieved the best speed on the ER data set.

- GLASSO: the block coordinate descent method proposed by Friedman et al. (2008). We use the latest version GLASSO 1.7 downloaded from `http://www-stat.stanford.edu/~tibs/glasso/`. We directly call their Fortran procedure using a MATLAB interface.

- PSM: the Projected Subgradient Method proposed by Duchi et al. (2008). We use the MATLAB source code provided in the PQN package (available at `http://www.cs.ubc.ca/~schmidtm/Software/PQN.html`).

- SINCO: the greedy coordinate descent method proposed by Scheinberg and Rish (2010). The code can be downloaded from `https://projects.coin-or.org/OptiML/browser/trunk/sinco`.

- IPM: An inexact interior point method proposed by Li and Toh (2010). The source code can be downloaded from `http://www.math.nus.edu.sg/~mattohkc/Covsel-0.zip`.

- PQN: the projected quasi-Newton method proposed by Schmidt et al. (2009). The source code can be downloaded from `http://www.di.ens.fr/~mschmidt/Software/PQN.html`.

In the following, we compare QUIC and the above state-of-the-art methods on synthetic and real data sets with various settings of $\lambda$. Note that we use the identity matrix as the initial point for QUIC, ADMM, SINCO, and IPM. Since the identity matrix is not a dual feasible point for dual methods (including GLASSO, PSM and PQN), we use $S + \lambda I$ as the dual initial point, which is the default setting in their original package.

#### 5.1.1 Experiments on Synthetic Data Sets

We first compare the run times of the different methods on synthetic data. We generate the two following types of graph structures for the underlying Gaussian Markov Random Fields:

- Chain Graphs: The ground truth inverse covariance matrix $\Sigma^{-1}$ is set to be $\Sigma^{-1}_{i,i-1} = -0.5$ and $\Sigma^{-1}_{i,i} = 1.25$.

- Graphs with Random Sparsity Structures: We use the procedure given in Example 1 in Li and Toh (2010) to generate inverse covariance matrices with random non-zero patterns. Specifically, we first generate a sparse matrix $U$ with nonzero elements equal to $\pm 1$, set $\Sigma^{-1}$ to be $U^T U$ and then add a diagonal term to ensure $\Sigma^{-1}$ is positive definite. We control the number of nonzeros in $U$ so that the resulting $\Sigma^{-1}$ has approximately $10p$ nonzero elements.

Given the inverse covariance matrix $\Sigma^{-1}$, we draw a limited number, $n = p/2$ i.i.d. samples from the corresponding GMRF distribution, in order to simulate the high-dimensional setting.

| Data set | | | Parameter | Properties of the solution | | |
|---|---|---|---|---|---|---|
| pattern | $p$ | $\|\Sigma^{-1}\|_0$ | $\lambda$ | $\|X^*\|_0$ | TPR | FPR |
| chain | 1000 | 2998 | 0.4 | 3028 | 1 | $3 \times 10^{-5}$ |
| chain | 4000 | 11998 | 0.4 | 11998 | 1 | 0 |
| chain | 10000 | 29998 | 0.4 | 29998 | 1 | 0 |
| random | 1000 | 10758 | 0.12 | 10414 | 0.69 | $4 \times 10^{-3}$ |
| | | | 0.075 | 55830 | 0.86 | 0.05 |
| random | 4000 | 41112 | 0.08 | 41936 | 0.83 | $6 \times 10^{-3}$ |
| | | | 0.05 | 234888 | 0.97 | 0.05 |
| random | 10000 | 91410 | 0.08 | 89652 | 0.90 | $4 \times 10^{-6}$ |
| | | | 0.04 | 392786 | 1 | $3 \times 10^{-3}$ |

Table 1: The parameters and properties of the solution for the synthetic data sets. $p$ stands for dimension, $\|\Sigma^{-1}\|_0$ indicates the number of nonzeros in ground truth inverse covariance matrix, $\|X^*\|_0$ is the number of nonzeros in the solution. TPR and FPR denote the true and false recovery rates, respectively, defined in (50).

Table 1 shows the attributes of the synthetic data sets that we used in the timing comparisons. The dimensionality varies from $\{1000, 4000, 10000\}$. For chain graphs, we select $\lambda$ so that the solution has (approximately) the correct number of nonzero elements. In order to test the performance of the algorithms under different values of $\lambda$, for the case of random-structured graphs we considered two $\lambda$ values; one of which resulted in the discovery of the correct number of non-zeros and one which resulted in five-times thereof. We measured the accuracy of the graph structure recovered by the true positive rate (TPR) and false positive rate (FPR) defined as

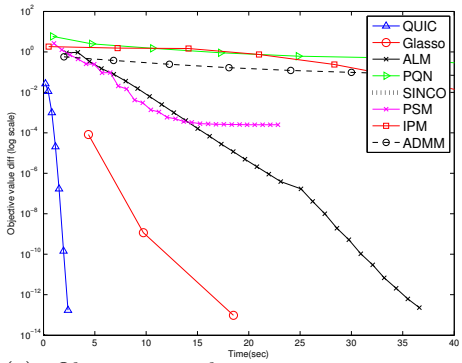
$$\text{TPR} = \frac{|\{(i,j) \mid (X^*)_{ij} > 0 \text{ and } Q_{ij} > 0\}|}{|\{(i,j) \mid Q_{ij} > 0\}|}, \text{FPR} = \frac{|\{(i,j) \mid (X^*)_{ij} > 0 \text{ and } Q_{ij} = 0\}|}{|\{(i,j) \mid Q_{ij} = 0\}|}, \tag{50}$$

where $Q$ is the ground truth sparse inverse covariance.

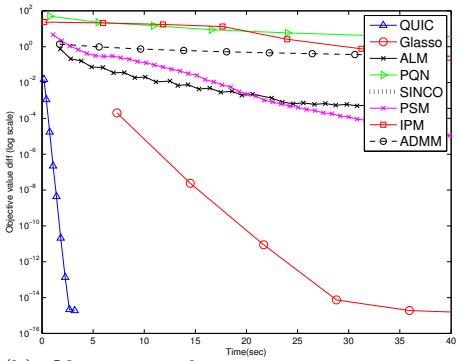
Since QUIC does not natively compute a dual solution, the duality gap cannot be used as a stopping condition.[2] In practice, we can use the minimum-norm sub-gradient (see Definition 6) as the stopping condition. There is no additional computational cost to this approach because $X^{-1}$ is computed as part of the QUIC algorithm. In the experiments, we report the time for each algorithm to achieve $\epsilon$-accurate solution defined by $f(X^k) - f(X^*) < \epsilon f(X^*)$. The global optimum $X^*$ is computed by running QUIC until it converges to a solution with $\|\operatorname{grad}^S f(X_t)\| < 10^{-13}$.

Table 2 shows the results for $\epsilon = 10^{-2}$ and $10^{-6}$, where $\epsilon = 10^{-2}$ tests the ability of the algorithm to get a good initial guess (the nonzero structure), and $\epsilon = 10^{-6}$ tests whether the algorithm can achieve an accurate solution. Table 2 shows that QUIC is consistently and

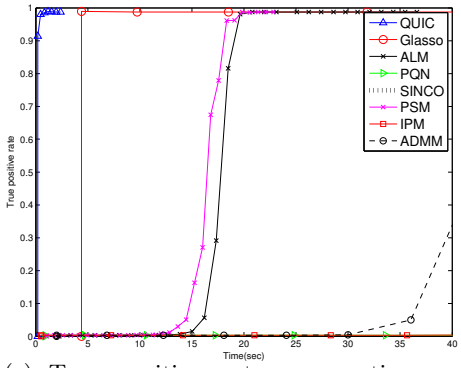
2. Note that $W = X^{-1}$ cannot be expected to satisfy the dual constraints $|W_{ij} - S_{ij}| \leq \lambda_{ij}$. One could project $X^{-1}$ in order to enforce the constraints and use the resulting matrix to compute the duality gap. Our implementation provides this computation only if the user requests it.



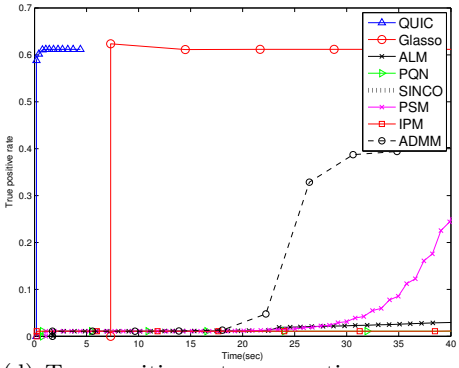

(a) Objective value versus time on chain1000



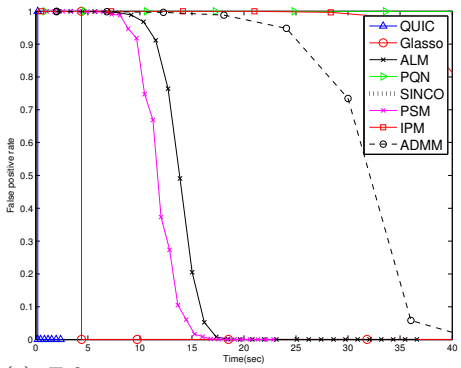

(b) Objective value versus time on random1000



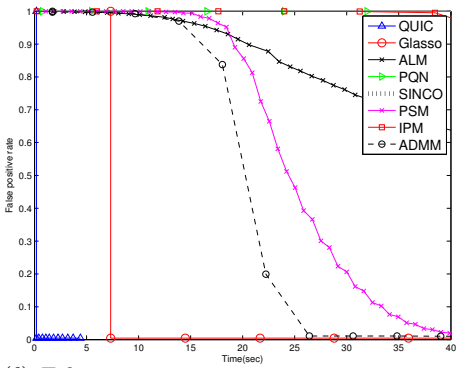

(c) True positive rate versus time on chain1000




(d) True positive rate versus time on random1000




(e) False positive rate versus time on chain1000




(f) False positive rate versus time on random1000

Figure 1: Comparison of algorithms on two synthetic data sets: chain1000 and random1000. The regularization parameter $\lambda$ is chosen to recover (approximately) correct number of nonzero elements (see Table 1). We can see that QUIC achieves a solution with better objective function value as well as better true positive and false positive rates in both data sets. Notice that each marker in the figures indicates one iteration. Note that all results are averaged over 5 replicated runs.

overwhelmingly faster than other methods, both initially with $\epsilon = 10^{-2}$, and at $\epsilon = 10^{-6}$. Moreover, for the $p = 10000$ random pattern, there are $p^2 = 100$ million variables and the

| Parameters | | | | Time (in seconds) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| pattern | $p$ | $\lambda$ | $\epsilon$ | QUIC | ALM | Glasso | PSM | IPM | SINCO | PQN | ADMM |
| chain | 1000 | 0.4 | $10^{-2}$ | **< 1** | 19 | 9 | 16 | 86 | 120 | 110 | 62 |
| | | | $10^{-6}$ | **2** | 42 | 20 | 35 | 151 | 521 | 210 | 281 |
| chain | 4000 | 0.4 | $10^{-2}$ | **11** | 922 | 460 | 568 | 3458 | 5246 | 672 | 1028 |
| | | | $10^{-6}$ | **54** | 1734 | 1371 | 1258 | 5754 | * | 10525 | 2584 |
| chain | 10000 | 0.4 | $10^{-2}$ | **217** | 13820 | 10250 | 8450 | * | * | * | * |
| | | | $10^{-6}$ | **987** | 28190 | * | 19251 | * | * | * | * |
| random | 1000 | 0.12 | $10^{-2}$ | **< 1** | 42 | 7 | 20 | 72 | 61 | 33 | 35 |
| | | | $10^{-6}$ | **1** | 28250 | 15 | 60 | 117 | 683 | 158 | 252 |
| | | 0.075 | $10^{-2}$ | **1** | 66 | 14 | 24 | 78 | 576 | 15 | 56 |
| | | | $10^{-6}$ | **7** | * | 43 | 92 | 146 | 4449 | 83 | * |
| random | 4000 | 0.08 | $10^{-2}$ | **23** | 1429 | 864 | 1479 | 4928 | 7375 | 2052 | 1025 |
| | | | $10^{-6}$ | **160** | * | 1743 | 4232 | 8097 | * | 4387 | * |
| | | 0.05 | $10^{-2}$ | **66** | * | 2514 | 2963 | 5621 | * | 2746 | * |
| | | | $10^{-6}$ | **479** | * | 5712 | 9541 | 13650 | * | 8718 | * |
| random | 10000 | 0.08 | $10^{-2}$ | **338** | 26270 | 14296 | * | * | * | * | * |
| | | | $10^{-6}$ | **1125** | * | * | * | * | * | * | * |
| | | 0.04 | $10^{-2}$ | **804** | * | * | * | * | * | * | * |
| | | | $10^{-6}$ | **2951** | * | * | * | * | * | * | * |

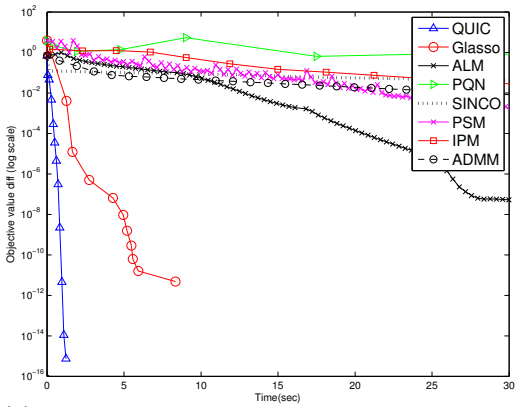
Table 2: Running time comparisons on synthetic data sets. See also Table 1 regarding the data set properties. We use $*$ to indicate that the run time exceeds 30,000 seconds (8.3 hours). The results show that QUIC is overwhelmingly faster than other methods, and is the only one which is able to scale up to solve problems with $p = 10000$.

selection of fixed/free sets helps QUIC to focus on a small subset of the variables. We converge to the solution in about 15 minutes, while other methods fail to obtain even an initial guess within 8 hours.

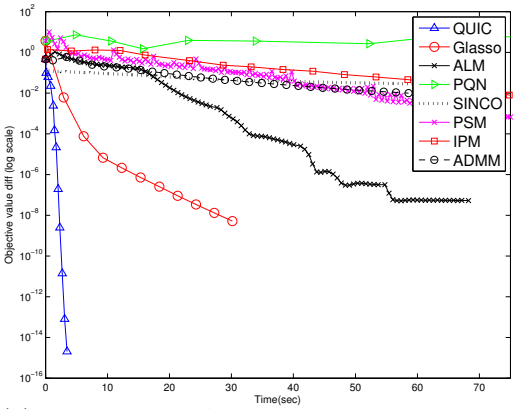
In some applications, researchers are primarily interested in just the graph structure represented by the solution. Therefore, in addition to the objective function value, we further compare the true positive and false positive rates of the nonzero pattern of the iterates $X_t$ obtained by each algorithm. In Figure 1, we use two synthetic data sets, chain1000 and random1000, as examples. For each algorithm, we plot the objective function value, true positive rate, and false positive rate of the iterates $X_t$ versus run time. For both ground truth pattern we generate 5 data sets and report the average results in Figure 1. For the methods that solve the dual problem, the sparse inverse covariance matrix $X_t = W_t^{-1}$ is usually dense, so we consider elements with absolute value larger than $10^{-6}$ as nonzero elements. We can see that QUIC not only obtains lower objective function value efficiently, but also recovers the ground truth structure of GMRF faster than other methods.

### 5.1.2 Experiments on Real Data sets

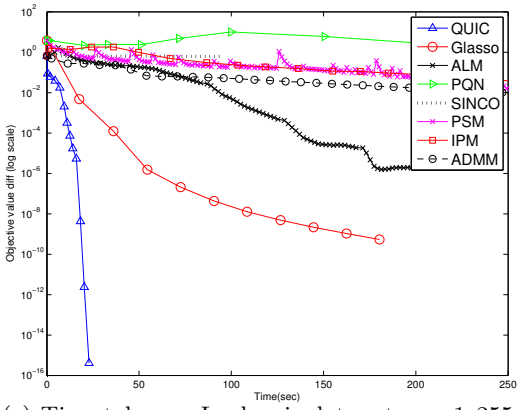
We use the real world biology data sets preprocessed by Li and Toh (2010) to compare the performance of our method with other state-of-the-art methods. In the first set of experiments, we set the regularization parameter $\lambda$ to be 0.5, which achieves reasonable sparsity for the following data sets: Estrogen ($p = 692$), Arabidopsis ($p = 834$), Leukemia ($p = 1,225$), Hereditary ($p = 1,869$). In Figure 2 we plot the relative error $(f(X_t) - f(X^*))/f(X^*)$ (on a log scale) against time in seconds. We can observe from Figure 2 that under the setting of large $\lambda$ and sparse solution, QUIC can be seen to achieve super-linear convergence while other methods exhibit at most a linear convergence. Overall, we see that QUIC can be five times faster than other methods, and can be expected to be even faster if a higher accuracy is desired.



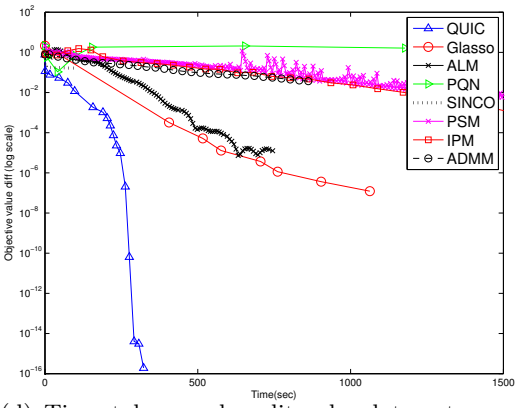

(a) Time taken on ER data set, $p = 692$, $\frac{\|X^*\|_0}{p^2} = 0.0222$




(b) Time taken on Arabidopsis data set, $p = 834$, $\frac{\|X^*\|_0}{p^2} = 0.0296$




(c) Time taken on Leukemia data set, $p = 1,255$, $\frac{\|X^*\|_0}{p^2} = 0.0221$




(d) Time taken on hereditarybc data set, $p = 1,869$, $\frac{\|X^*\|_0}{p^2} = 0.0198$

Figure 2: Comparison of algorithms on real data sets with $\lambda = 0.5$. The results show that QUIC converges faster than the other methods. Notice that each marker in the figures indicates one iteration. All the results are averaged over five runs.

In the second set of experiments, we compare the algorithms under different values of the regularization parameter $\lambda$ on the ER data set. In Figure 2(a) we show the results for $\lambda = 0.5$. We then decrease $\lambda$ to 0.1, 0.05, 0.01 using the same data sets and show the results in Figure 6. A smaller $\lambda$ yields a denser solution, and we list the density of the converged solution $X^*$ in Figure 6. From Figure 6 we can see that QUIC is the most efficient method when $\lambda$ is large (solution is sparse), but IPM and PSM outperform QUIC when $\lambda$ is small (solution is dense). However, such cases are usually not so useful in practice because when solving the $\ell_1$-regularized MLE problem one usually wants a sparse graph structure for the GMRF. The main reason that QUIC is so efficient for large $\lambda$ is that with *fixed/free* set selection, the coordinate descent method can focus on a small portion of variables, while in PSM and IPM the whole matrix is updated at each iteration.

### 5.2 Empirical Analysis of QUIC

Next we present some empirical analysis of QUIC regarding the effects of several parameters. We also demonstrate that the fixed/free set selection in QUIC significantly reduce the computational complexity.

#### 5.2.1 Effect of Approximate Newton Directions

In the convergence analysis of Section 4, we assumed that each Newton direction $D_t^*$ is computed exactly by solving the Lasso subproblem (20). In our implementation we use an iterative (coordinate descent) solver to compute $D_t^*$, which after a finite set of iterations only solves the problem approximately. In the first experiment we explore how varying the accuracy to which we compute the Newton direction affects overall performance. In Figure 3 we plot the total run times for the ER biology data set from Li and Toh (2010) corresponding to different numbers of inner iterations used in the coordinate descent solver.

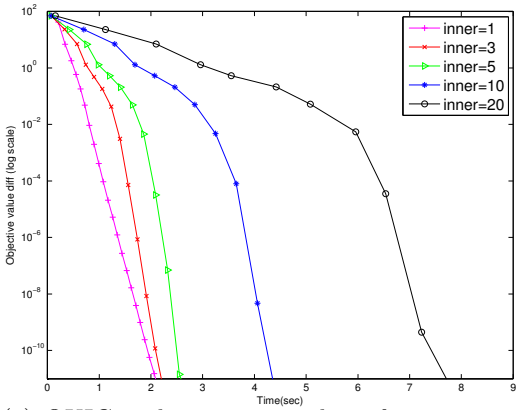
We can observe that QUIC with one inner iteration converges faster in the beginning, but eventually achieves just a linear convergence rate, while QUIC with 20 inner iterations converges more slowly in the beginning, but eventually achieves quadratic convergence. Based on this observation, we propose an adaptive stopping condition: we set the number of coordinate descent steps to be $\lceil \alpha t \rceil$ for the $t$-th outer iteration, where $\alpha$ is a constant; we use $\alpha = 1/3$ in our experiments. Figure 3(b) shows that by using this adaptive stopping condition, QUIC is not only efficient in the beginning, but also achieves quadratic convergence.

#### 5.2.2 Line Search Parameters

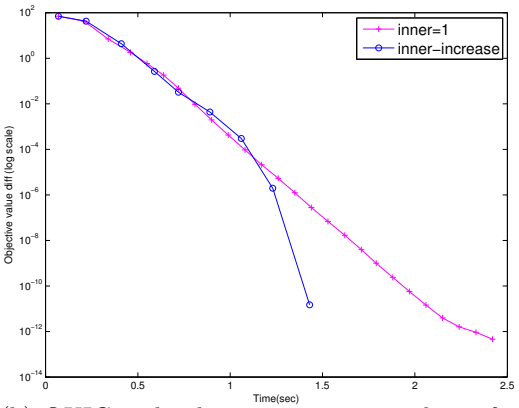
We demonstrate the robustness of QUIC to line search parameters $\sigma$ and $\beta$. The results are shown in Figure 4.

#### 5.2.3 Fixed/free Set Selection

To further demonstrate the power of *fixed/free* set selection, we use Hereditarybc data set as an example. In Figure 5, we plot the size of the free set versus the number of Newton iterations. Starting from a total of $1869^2 = 3,493,161$ variables, the size of the free set

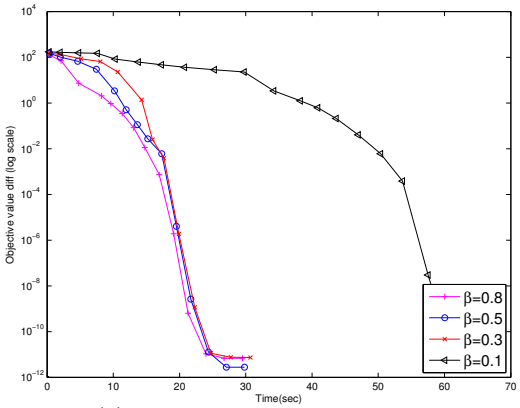



(a) QUIC with varying number of inner iterations (on ER data set).

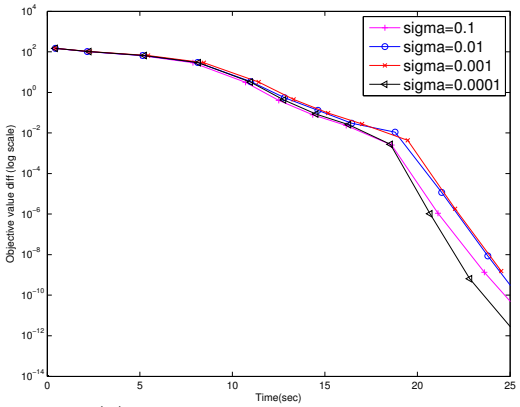



(b) QUIC with adaptive stopping condition for inner iterations.

Figure 3: The behavior of QUIC when varying the number of inner iterations. Figure 3(a) show that QUIC with one inner iteration converges faster in the beginning but eventually achieves just linear convergence, while QUIC with 20 inner iterations converges slower in the beginning, but has quadratic convergence. Figure 3(b) shows that by adaptively setting the number of iterations in QUIC, we get the advantages of both cases. Notice that each marker in the figures indicates one iteration.

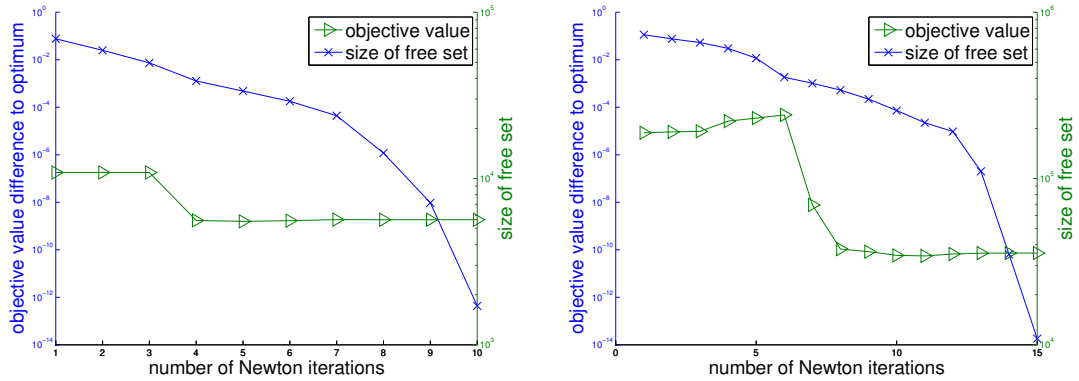



(a) Comparison on different $\beta$




(b) Comparison on different $\sigma$

Figure 4: Comparison of different line search parameters on the Leukemia data set. Figure 4(a) shows that QUIC is robust to a wide range of $\beta$ values, but becomes slower when $\beta$ is too small. Figure 4(b) demonstrates that QUIC is robust with respect to $\sigma$.




(a) ER data set (b) Hereditarybc data set

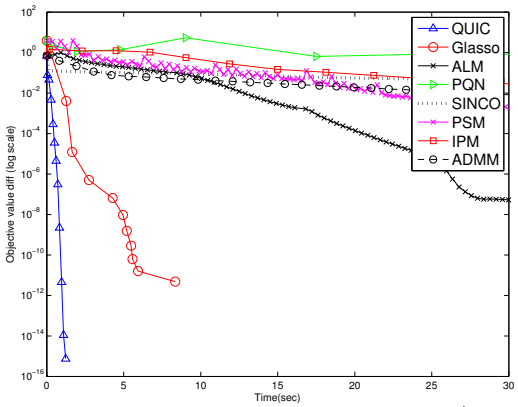
Figure 5: Size of free sets and objective value versus iterations. For both data sets, the sizes of free sets are always less than $6\|X^*\|_0$ when running QUIC algorithm.

progressively drops, in fact to less than $120,000$ in the very first iteration. We can see the super-linear convergence of QUIC even more clearly when we plot it against the number of iterations.

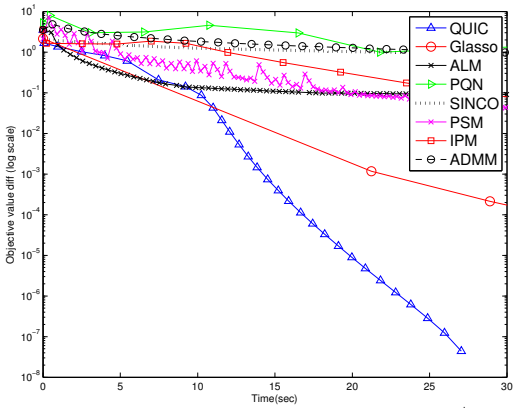
We further analyze the proportion of time taken by the two main steps of QUIC: coordinate descent updates and line search procedure (in line search, the most time intensive computation is the Cholesky factorization). We have looked at the ratio of time consumed by those two steps on different data sets. We found that when the size of the *free* set is large, QUIC will spend most of its time on coordinate updates. For example, in the Hereditarybc data set with $\lambda = 0.5$, where the size of free set is $0.11p^2$ in the beginning, coordinate descent takes 85.1% of the total run time, while line search only takes 14.9% of the total run time. In contrast, for data sets with small size of *free* set, coordinate descent updates will take noticeable less time. For example, when running on the ER data set with $\lambda = 0.5$, where the size of free set is $0.033p^2$ in the beginning, 59.6% of the total time is spent on coordinate descent.

### 5.2.4 Block-diagonal Structure

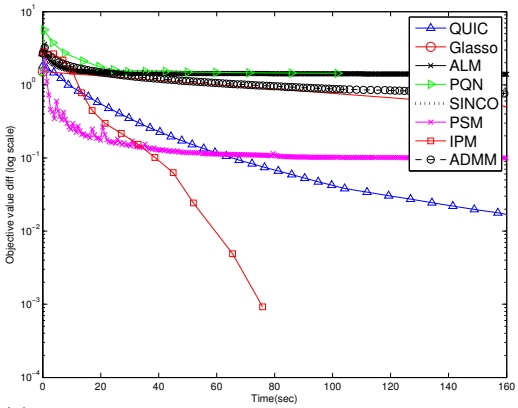
As discussed earlier, Mazumder and Hastie (2012); Witten et al. (2011) showed that when the thresholded covariance matrix $E = \max(|S| - \lambda, 0)$ is block-diagonal, then the problem can be naturally decomposed into sub-problems. This observation has been implemented in the latest version of glasso. In the end of Section 3, we showed that the *fixed/free* set selection can automatically identify the block-diagonal structure of the thresholded matrix, and thus QUIC can benefit from block-diagonal structure even when we do not explicitly decompose the matrix in the beginning. In the following experiment we show that with input sample covariance $S$ with block-diagonal structure represented by $E$ (see Section 3.4), QUIC still outperforms glasso. Moreover, we show that when some off-diagonal elements are added into the problem, while QUIC is still efficient because of its *fixed/free* set selection, glasso on the other hand suddenly becomes much slower.



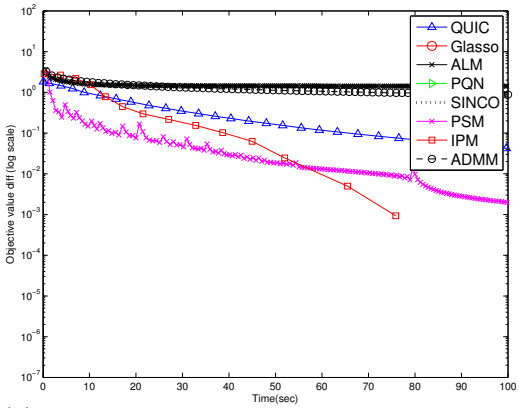

(a) Time taken on ER data set, $\lambda = 0.5$, $\frac{\|X^*\|_0}{p^2} = 0.0222$




(b) Time taken on ER data set, $\lambda = 0.1$, $\frac{\|X^*\|_0}{p^2} = 0.0724$




(c) Time taken on ER data set, $\lambda = 0.05$, $\frac{\|X^*\|_0}{p^2} = 0.125$




(d) Time taken on ER data set, $\lambda = 0.01$, $\frac{\|X^*\|_0}{p^2} = 0.322$

Figure 6: Comparison of algorithms on the ER data set ($p = 692$) under different $\lambda$. The results show that QUIC converges faster for larger $\lambda$ where solutions are sparse, while IPM and PSM are faster for smaller $\lambda$ which produces denser solutions. Note that each marker in the figures indicates one iteration, and that all the results are averaged over 5 replicated runs.

We generate synthetic data with block-diagonal structure as follows. We generate a sparse $150 \times 150$ inverse covariance matrix $\bar{\Theta}$ as discussed in Section 5.1.1, and then replicate $\bar{\Theta}$ eight times on the diagonal blocks to form a $1200 \times 1200$ block-diagonal matrix. Using this inverse covariance matrix to generate samples, we compare the following methods:

- QUIC: our proposed algorithm.
- GLASSO: In the latest version of GLASSO, the matrix is first decomposed into connected components based on the thresholded covariance matrix $\max(|S| - \lambda)$, and then each sub-problem is solved individually.

We then test the two algorithms for regularization parameter $\lambda$ taking values from the set $\{0.017, \ldots, 0.011\}$. When $\lambda = 0.017$, the thresholded covariance matrix $E$ has eight blocks, while when $\lambda = 0.011$ the block structure reduces to a single block. For each single $\lambda$ trial, we compare the time taken by QUIC and GLASSO to achieve $(f(X_t) - f(X^*))/f(X^*) < 10^{-5}$. Figure 7 shows the experimental results. We can see that both methods are very fast for the case where the problem can be decomposed into 8 sub-problems (large $\lambda$); however, when we slightly increase $\lambda$ so that there is only 1 connected component, QUIC is much faster than GLASSO. This is because even for the non-decomposable case, QUIC can still keep most of the elements of the very sparse off-diagonal blocks in the fixed set to speed up the process, while GLASSO cannot benefit from this sparse structure.

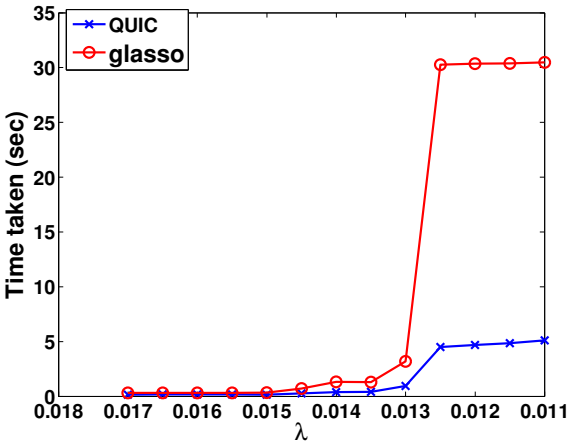



Figure 7: In this figure, we show the performance of QUIC and GLASSO for a sparse synthetic data with clustered structure. Using the same input covariance matrix $S$, we test the time for each algorithm to achieve $(f(X_t) - f(X^*))/f(X^*) < 10^{-5}$ under various values of $\lambda$. When $\lambda = 0.017$, the problem can be decomposed into 8 sub-problems, while when $\lambda = 0.011$ there is only one connected component. We can see that for the smaller values of $\lambda$, QUIC's approach of *free*/*fixed* set selection is able to exploit the sparsity structure of the solution, while GLASSO's training time increases dramatically.

## Acknowledgements

This research was supported by NSF grant IIS-1018426. ISD also acknowledges support from the Moncrief Grand Challenge Award. PR also acknowledges support from NSF IIS-1149803. We would like to thank Kim-Chuan Toh for providing data sets and the IPM code as well as Katya Scheinberg and Shiqian Ma for providing the ALM implementation.

## References

A. Agarwal, S. Negahban, and M. Wainwright. Convergence rates of gradient methods for high-dimensional statistical recovery. In *Advances in Neural Information Processing Systems*, 2010.

T. M. Apostol. *Mathematical Analysis*. Addison-Wesley, second edition, 1974.

O. Banerjee, L. E. Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *Journal of Machine Learning Research*, 9, 2008.

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

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 7th printing edition, 2009.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Matlab scripts for alternating direction method of multipliers. `http://www.stanford.edu/~boyd/papers/admm/`, 2012.

A. d'Aspremont, O. Banerjee, and L. El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and its Applications*, 30(1):56–66, 2008.

I. Daubechies, M. Defrise, and C. De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics*, 57(11):1413–1457, 2004.

J. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse Gaussians. In *Conference on Uncertainty in Artificial Intelligence*, 2008.

J.C. Dunn. Newton's method and the Goldstein step-length rule for constrained minimization problems. *SIAM J. Control and Optimization*, 18(6):659–674, 1980.

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, 2007.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, July 2008.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. In *Advances in Neural Information Processing Systems*. 2011.

C.-J. Hsieh, I. S. Dhillon, P. Ravikumar, and A. Banerjee. A divide-and-conquer method for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems*, 2012.

C. Lam and J. Fan. Sparsistency and rates of convergence in large covariance matrix estimation. *Annals of Statistics*, 37:4254–4278, 2009.

J. D. Lee, Y. Sun, and M. A. Saunders. Proximal Newton-type methods for convex optimization. In *Advances in Neural Information Processing Systems*, 2012.

E. S. Levitin and B. T. Polyak. Constrained minimization methods. *U.S.S.R. Computational Math. and Math. Phys.*, 6:1–50, 1966.

L. Li and K.-C. Toh. An inexact interior point method for L1-regularized sparse covariance selection. *Mathematical Programming Computation*, 2:291–315, 2010.

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

R. Mazumder and T. Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *Journal of Machine Learning Research*, 13:723–736, 2012.

L. Meier, S. Van de Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society, Series B*, 70:53–71, 2008.

P. Olsen, F. Oztoprak, J. Nocedal, and S. Rennie. Newton-like methods for sparse inverse covariance estimation. Technical report, Optimization Center, Northwestern University, 2012. URL `http://www.optimization-online.org/DB_HTML/2012/06/3506.html`.

B. T. Polyak. The conjugate gradient method in extremal problems. *U.S.S.R. Computational Mathematics and Mathematical Physics*, 9:94–112, 1969.

P. Ravikumar, M. J. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing $\ell_1$-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

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

K. Scheinberg and I. Rish. Learning sparse Gaussian Markov networks using a greedy coordinate ascent approach. In *Machine Learning and Knowledge Discovery in Databases*, volume 6323 of *Lecture Notes in Computer Science*, pages 196–212. Springer Berlin / Heidelberg, 2010.

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

M. Schmidt. *Graphical Model Structure Learning with l1-regularization*. PhD thesis, University of British Columbia, 2010.

M. Schmidt, E. Van Den Berg, M. P. Friedl, and K. Murphy. Optimizing costly functions with simple constraints: A limited-memory projected quasi-Newton algorithm. In *Proc. of Conf. on Artificial Intelligence and Statistics*, pages 456–463, 2009.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58:267–288, 1996.

P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423, 2007.

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

T. T. Wu and K. Lange. Coordinate descent algorithms for lasso penalized regression. *Annals of Applied Statistics*, 2(1):224–244, 2008.

G.-X. Yuan, K.-W. Chang, C.-J. Hsieh, and C.-J. Lin. A comparison of optimization methods and software for large-scale L1-regularized linear classification. *Journal of Machine Learning Research*, 11:3183–3234, 2010.

G.-X. Yuan, C.-H. Ho, and C.-J. Lin. An improved GLMNET for L1-regularized logistic regression. *Journal of Machine Learning Research*, 13:1999–2030, 2012.

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

S. Yun and K.-C. Toh. A coordinate gradient descent method for L1-regularized convex minimization. *Computational Optimizations and Applications*, 48(2):273–307, 2011.