Computational Statistics and Data Analysis 75 (2014) 15–27

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

A joint convex penalty for inverse covariance matrix estimation

Ashwini Maurya *

Department of Statistics and Probability, Michigan State University, United States

ARTICLE INFO

Article history:
Received 29 July 2013
Received in revised form 8 January 2014
Accepted 22 January 2014
Available online 31 January 2014

Keywords:
Proximal gradient
Joint penalty
Convex optimization
Sparsity

ABSTRACT

The paper proposes a joint convex penalty for estimating the Gaussian inverse covariance matrix. A proximal gradient method is developed to solve the resulting optimization problem with more than one penalty constraints. The analysis shows that imposing a single constraint is not enough and the estimator can be improved by a trade-off between two convex penalties. The developed framework can be extended to solve wide arrays of constrained convex optimization problems. A simulation study is carried out to compare the performance of the proposed method to graphical lasso and the SPICE estimate of the inverse covariance matrix.

Published by Elsevier B.V.

1. Introduction

Recent surge in the use of electronic and digital technology has created vast amount of high dimensional data whose analysis demands advanced statistical tools and computational techniques. Examples are biological data of gene expression measurement, FMRI scanned images of human brain and Netflix data. In these datasets, one often have very few observations as compared to the number of variables and therefore the choice of standard statistical methods becomes inappropriate for making valid inference. Thus the concept of parsimony becomes very crucial. In many applications the problem of interest is often estimating the dependence structure of the data where the underlying probability distribution of observations is either fixed (static) or evolving over time (i.e. time varying or dynamic). In the static framework, a common assumption is that the data are independently and identically distributed (i.i.d.), whereas in dynamic setting, the distribution of data evolves over time and hence the i.i.d. assumption no longer remains valid. Here we focus on the static framework.

1.1. Background

The covariance selection was introduced by Dempster (1972) where the basic idea was to (i) introduce parsimony in parameter model fitting and (ii) exploit the powerful yet elegant theory of exponential family as a tool of practical data analysis. The computational ease along with attractive statistical features of Gaussian distribution makes it a popular choice for most of the application problems. Estimation of the inverse covariance matrix is important in a number of statistical analyses including:

* Correspondence to: 619 Red Cedar Road, Department of Statistics and Probability, Michigan State University, C508 Wells Hall, East Lansing, MI, 48824-1027, United States. Tel.: +1 517 802 7757.
E-mail addresses: mauryaas@stt.msu.edu, mauryaas@msu.edu, akmaurya07@gmail.com.

0167-9473/$ – see front matter. Published by Elsevier B.V.
http://dx.doi.org/10.1016/j.csda.2014.01.015

- Gaussian Graphical Modeling: In Gaussian graphical modeling, a zero entry of an element in the inverse of a covariance matrix corresponds to conditional independence between the variables.
- Linear or Quadratic Discriminant Analysis: When the features are assumed to have multivariate Gaussian density, the resulting discriminant rule requires an estimate of the inverse covariance matrix.
- Principal Component Analysis (PCA): In multivariate high dimensional data it is often desirable to transform the high-dimensional feature space to a lower dimension without loosing much information. The covariance matrix method is a popular method for PCA estimates.

Several approaches have been suggested to address the estimation problem of the inverse of a covariance matrix. These approaches are either based on regularized estimation of the inverse of the covariance matrix (Banerjee et al., 2008; Friedman et al., 2007; Bickel and Levina, 2008; Rothman et al., 2008) or regularized high dimensional regression (Meinshausen and Bühlmann, 2006; Zhou et al., in press). Among earlier developments, an exact maximization of ℓ_1 penalized log likelihood using interior point methods was suggested for estimating the inverse covariance matrix (Dahl et al., 2008; Yuan and Lin, 2007; Banerjee et al., 2008). Let $X = (X_1, X_2, \ldots, X_p)^T$ be a p-variate random vector from multivariate Gaussian distribution $N_p(\mu, \Sigma)$. μ is the mean vector and Σ is the positive definite covariance matrix. Let $X^1, X^2, \ldots, X^n$ be n independent copies of X. The sample covariance matrix is given by

$$S = (1/n)\sum_{i=1}^{n}(X^i - \bar{X})(X^i - \bar{X})^T \tag{1.1}$$

where X^{iT} is the transpose of X^i and $\bar{X}$ is the mean vector of sample observations.

Let $\hat{W}$ be the estimate of inverse of the covariance matrix Σ. Banerjee et al. (2008) have considered the following determinant maximization (MAXDET) (Vandenberghe and Boyd, 2004) problem:

$$\hat{W} = \arg\max_{X \succ 0}\{\log(\det(X)) - \mathrm{tr}(SX) - \lambda\|X\|_1\} \tag{1.2}$$

where $\mathrm{tr}(S)$ is the trace of the matrix S, $\|X\|_1$ is the ℓ_1 norm and defined as the sum of absolute values of elements of matrix X and λ is the regularization parameter which controls the sparsity structure of the estimated inverse covariance matrix. Another approach to solve the above optimization problem is based on high dimensional regression (Yuan, 2009; Meinshausen and Bühlmann, 2006; Wainwright et al., 2006). For Gaussian graphical models, the main motivation behind the regression approach is the explicit relation between the elements of the inverse covariance matrix and coefficients of the predictor variables (Yuan, 2009). Meinshausen and Bühlmann (2006) follow this approach and estimate the neighborhood structure of the variables by fitting ℓ_1 regularized regression to each of the variables on the remaining set of variables as predictors. They also have established the consistency of their estimates under certain assumptions of sparsity and stability.

Friedman et al. (2007) introduced graphical lasso which has better computational power compared to earlier methods. In graphical lasso, the algorithm obtains the lasso estimate of each row/column of the covariance matrix given the sample covariance matrix. It uses the block co-ordinate descent algorithm for optimization of the objective function.

Rothman et al. (2008) has proposed the sparse permutation invariance covariance estimate (SPICE). This method uses Cholesky decomposition of the inverse covariance matrix and a quadratic approximation of the likelihood function to simplify the problem as finding the minimum of each univariate parameter in the objective function in closed form. The objective function is invariant and consequently the estimator remains permutation invariant. The method uses the cyclical co-ordinate descent algorithm to do the optimization.

In another approach, Sheena and Gupta (2003) have proposed a constrained maximum likelihood estimator with restrictions on the lower or upper bound of the eigenvalues. This method focuses on only one of the two ends of the eigenspectrum and thus the resulting estimator does not correct for the overestimation of the large eigenvalues and underestimation of the small eigenvalues simultaneously. Consequently their approach does not address the distortion of the entire eigenspectrum especially in small sample sizes. Won et al. (2012) consider a maximum likelihood estimation of the covariance matrix with condition number constraint. The condition number of a matrix is defined as the ratio of largest to smallest eigenvalue of the matrix. However this approach itself requires an estimation of condition number.

To control the distortion of eigenspectrum of the covariance matrix, we consider a joint penalty of sum of singular values (trace norm) in addition to the ℓ_1 norm. By minimizing the joint penalty function of ℓ_1 and the trace norm, the resulting estimated inverse covariance matrix is sparse as well as singular values of the corresponding covariance matrix are more centered than the observed sample covariance matrix. Rolfs et al. (2012) consider the estimation of the inverse covariance matrix which can be seen as a particular case of the proposed approach by setting-off the trace norm penalty. A single penalty of the ℓ_1 norm is appropriate when the underlying true inverse covariance matrix is sparse. However it does not control the distortion in eigenspectrum of the inverse covariance matrix. Controlling the eigenspectrum is an intuitive way to get a well conditioned estimate of the inverse covariance matrix.

1.2. Contribution

We propose a joint convex penalty of ℓ_1 and the *trace norm* to the inverse covariance matrix estimation. The estimator thus obtained is simultaneously sparse and gives better performance than graphical lasso for small sample size in terms of

mean squared error loss, average relative error and Kullback–Leibler loss. We implement a *proximal gradient* method for optimization of the objective function. The proposed algorithm is shown to converge in function value as $O(1/k)$ under mild conditions.

2. Main result

2.1. Problem formulation and notations

Let $X \sim N_p(0, \Sigma)$, $\Sigma \succ 0$. For simplicity of notation, we denote $W = \Sigma^{-1}$. Let $\|W\|_1$ be the ℓ_1 norm and is defined as the sum of absolute values of its entries and $\|W\|_*$ be the *trace* norm and defined as the sum of singular values of the matrix W.

We consider the following optimization problem with joint convex penalty:

$$\arg\min_{W \succ 0} F(W) := f(W) + g_1(W) + g_2(W) \tag{2.1}$$

where

$$f(W) = -\log(\det(W)) + \operatorname{tr}(SW); \qquad g_1(W) = \lambda\|W\|_1; \qquad g_2(W) = \tau\|W\|_* \tag{2.2}$$

where λ and τ are non negative constants. In this paper, we particularly focus on the inverse covariance matrix for a multivariate Gaussian distribution. The algorithm developed in the paper addresses a wide array of problems in statistics and machine learning. Some of the other important applications include Matrix Classification Problems (Tomioka and Aihara, 2007; Bach, 2008), Matrix Completion Problems (Candès and Recht, 2009), and Multi-Task Learning (Argyriou et al., 2008).

Note that $f(W)$ in (2.2) is a strictly convex function, the ℓ_1 norm is a smooth convex function except at the origin and the trace norm is convex surrogate of rank over the unit ball of spectral norm (Fazel, 2002). The above problem is the convex optimization problem with non-smooth constraints. A natural choice to solve the above optimization problem is the subgradient method which generates a sequence of estimates $\{W_k,\ k = 1, 2, 3, \ldots\}$ as

$$W_k = W_{k-1} - \alpha \bigtriangledown F(W_{k-1})$$

where α is some positive step size and $\bigtriangledown F(W_{k-1})$ is the subgradient indicating the direction of greatest value increase of the function $F(W)$ at W_{k-1}. This method has a well known convergence rate of $O(k^{-\frac{1}{2}})$ for non-smooth convex functions (Nesterov, 2005). We employ the proximal gradient method to obtain a better rate of convergence of order $O(k^{-1})$. This method can be generalized to solve an arbitrary combination of convex functions (Bertsekas, 2010).

2.2. Proximal gradient algorithm

Much like Newton's method is a standard tool for solving unconstrained smooth optimization problems of modest size, proximal algorithms can be viewed as an analogous tool for non-smooth, constrained, large-scale, or distributed versions of these problems. They are very generally applicable, but are especially well-suited to problems of substantial recent interest involving large or high-dimensional dataset. The main motivation behind the success of the proximal gradient algorithm is the availability of inexpensive operators of a function, which itself involves solving a small convex optimization problem. These sub-problems, which generalize the problem of projecting a point onto a convex set, often admit closed form solutions or can be solved very quickly with standard or simple specialized methods. In our setup of the problem, we require computation of a proximal operator for ℓ_1 and *trace norm*.

Let $h(W)$ be a lower semi-continuous convex function of W, which is not identically equal to $+\infty$. Then the proximal point algorithm (Rockafellar, 1976) generates a sequence of solutions $\{W_k,\ k = 1, 2, 3, \ldots\}$ as

$$W_k = \operatorname{Prox}_h(W_{k-1}) = \arg\min_{W \succ 0}\left(h(W) + \frac{1}{2}\|W - W_{k-1}\|_2^2\right).$$

The above sequence $\{W_k,\ k = 1, 2, 3, \ldots\}$ weekly converges to the optimal solution of $\min_{W \succ 0} h(W)$ (Rockafellar, 1976). To use the structure of the above optimization algorithm, we use quadratic approximation of $f(W)$, which is justified since f is strictly convex.

2.2.1. Basic approximation model

For any $L > 0$, consider the following quadratic approximation model of $f(W)$ at W':

$$Q_L(W, W') := f(W') + \langle W - W', \bigtriangledown f(W')\rangle + \frac{L}{2}\|W - W'\|_2^2 \tag{2.3}$$

where $\langle A, B\rangle$ is the inner product of A and B.

The optimization problem in (2.1) has two convex penalties. Some of the methods to solve multiple constraints optimization problems are *proximal Gradient Method* (Bertsekas, 2010) and *Generalized Forward–Backward Algorithm* (Raguet et al.,

2011). The proximal gradient method consists of sequential optimization of (2.1) by taking one constraint at a time in either cyclic or random order. Rewriting the optimization problem (2.1) with single constraint gives

$$\begin{aligned}\text{Prox}_{\frac{1}{L}g_i}(W') &= \arg\min_{W\succ 0}\left(Q_L(W,W') + g_i(W)\right)\\ &= \arg\min_{W\succ 0}\left(f(W') + \langle W - W', \bigtriangledown f(W')\rangle + \frac{L}{2}\|W - W'\|_2^2 + g_i(W)\right)\\ &= \arg\min_{W\succ 0}\left(\frac{L}{2}\left\|W - \left\{W' - \frac{1}{L}\bigtriangledown f(W')\right\}\right\|_2^2 + g_i(W)\right). \end{aligned} \tag{2.4}$$

The projected gradient method is a special case of the proximal gradient method for the convex function where the constraint function $g_i(\cdot)$ is an indicator function. In general, value of L is unknown and even if known, a local estimate is preferred (Bach et al., 2011). For a convex differential function $f(W)$, it turns out to be an upper bound of the Lipschitz parameter of $\bigtriangledown f(W)$. i.e. L satisfies:

$$\|\bigtriangledown f(W) - \bigtriangledown f(W')\|_2 \leq L\|W - W'\|_2 \quad \forall\, W, W' \in \mathbb{D}\text{om}(f). \tag{2.5}$$

A common method of generating value of L is to do a line search. In the above optimization problem, we sequentially generate new estimates and increase the value of L by a factor $\gamma > 1$ until the following condition is met:

$$f_L(W_k) \leq f(W_{k-1}) + \langle W_k - W_{k-1}, \bigtriangledown f(W_{k-1})\rangle + \frac{L}{2}\|W_k - W_{k-1}\|_2^2, \tag{2.6}$$

where W_k is a solution at the kth iteration. However the above procedure could be time consuming as we need to optimize (2.1) arbitrary number of times in order to get the suitable value of L. For a current setup of $f(W)$ as in (2.2) which is strictly convex and differentiable, at iteration k, Lipschitz parameter $L > 0$ satisfies:

$$\bigtriangledown f(W) = S - W^{-1}$$

because

$$L \geq \frac{\|(W_k)^{-1} - (W_{k-1})^{-1}\|_2}{\|W_k - W_{k-1}\|_2} \geq \|(W_k)(W_{k-1})\|_2.$$

Therefore a suitable value of L can be taken as $\|(W_{k-1} - \bigtriangledown f(W_{k-1}))W_{k-1}\|_2$.
In Lemmas 2.1 and 2.2, we give the proximal operator for ℓ_1 and the trace norm.

Lemma 2.1. *Let $M \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$. The proximal operator of $\|\cdot\|_1$ with constant λ is given by*

$$\text{Prox}_{\lambda\|\cdot\|_1}(M) = \arg\min_{C\succ 0}\left(\lambda\|C\|_1 + \frac{1}{2}\|C - M\|_2^2\right), \tag{2.7}$$

where

$$\text{Prox}_{\lambda\|\cdot\|_1}(M) = \text{sign}(M)(0, M - \lambda)_+, \quad \lambda > 0.$$

Proof. Proof of the lemma is given in the Appendix. □

Lemma 2.2. *Let $M \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$ and $M = U\Sigma V^T$ be a singular value decomposition of M where $U \in \mathbb{R}^{\mathbf{m}\times\mathbf{r}}$ and $V \in \mathbb{R}^{\mathbf{r}\times\mathbf{n}}$ have orthogonal columns, Σ is the diagonal matrix of singular values of M and r is the rank of the matrix M. Then the proximal operator of $\|\cdot\|_*$ with constant τ is given by*

$$\text{Prox}_{\tau\|\cdot\|_*}(M) = \arg\min_{C\succ 0}\left(\tau\|C\|_* + \frac{1}{2}\|C - M\|_2^2\right), \tag{2.8}$$

where $\text{Prox}_{\tau\|\cdot\|_*}(M) = U\Sigma_\tau V^T$, Σ_τ is the diagonal matrix with $((\Sigma_\tau))_{ii} = \max(0, \Sigma_{ii} - \tau)$, $\tau > 0$.

Proof. Proof of the lemma is given in the Appendix. □

For $\ell_\mathbf{1}$ and *trace norm*, the proximal operators are inexpensive to calculate. This results in efficient optimization of the objective function. The proximal operator for $\ell_\mathbf{1}$ is elementwise soft-thresholding operator. The proximal operator for the trace norm is obtained by shrinking the singular values of the inverse covariance matrix by a regularization parameter. A larger value of the regularization parameter would make the resulting inverse covariance matrix singular, therefore in the optimization algorithm we do not shrink the singular values below threshold of minimum singular value. Sheena and Gupta (2003) do similar where they assume that the eigenvalues have either a lower or an upper bound but this allows minimization of the singular values in only one direction. Instead we consider minimization of the sum of singular values which leads to minimization of the distortion of the entire eigenspectrum.

2.2.2. Algorithm for optimization

Below we summarize the optimization algorithm for (2.1).

Initialize $L_0 = 1,\ \gamma > 1, W_0 = \text{diag}(1/\text{diag}(S))$.

Iterate

Step 1: Set $\bar{L} = L_{k-1}$.

Step 2: While $F(W^*) > Q_{\bar{L}}((W^*), W_{k-1}) + g_1(W^*) + g_2(W^*)$

(where $W^* = \arg\min_{W \succ 0} Q_{\bar{L}}(W, W_{k-1}) + g_1(W) + g_2(W)$)

Set $\bar{L} = \gamma \bar{L}$.

Step 3: Set $L_k = \bar{L}$,

Set $Z_k = W_k - \frac{1}{L_k} \bigtriangledown f(W_k)$,

Set $Z_{k+1} = \text{Prox}_{(\tau/L_k)g_2}(Z_k)$,

Set $W_{k+1} = \text{Prox}_{(\lambda/L_k)g_1}(Z_{k+1})$.

Repeat until convergence.

2.2.3. Choosing the regularization parameter

Choice of the regularization parameter is a challenging problem in high dimensional data analysis. Regularization has clear benefit in producing a sparse solution as well reduces false discovery rate. A smaller value of λ accounts for a sparser structure of the inverse covariance matrix. Some of the methods for choosing the regularization parameter include K-fold cross validation (KCV), stability approach to regularization selection (StARS) (Liu et al., 2010), Akaike-Information Criteria (AIC) and Bayesian Information criteria (BIC). Experiments (Liu et al., 2010) have shown that AIC and BIC methods tend to give poor performance for smaller sample size. Also K-fold cross validation tends to select smaller values of the regularization parameter and results in higher false discovery rate. We follow StARS approach for estimating the regularization parameters λ and τ.

3. Convergence analysis

We use the following result from Bertsekas (2010).

Lemma 3.1. *Let $\{W_n,\ L_n, n = 1, 2, \ldots\}$ be the sequence generated by algorithm Section 2.2.2. Let $c > 0$ be a constant satisfying:*

$$\max\{\bigtriangledown\|f(W)\|, \bigtriangledown\|g_j(W)\|\} \leq c \quad \text{and}$$
$$\max\{f(W_n) - f(Z_{n+j-1}), g_j(W_n) - g_j(Z_{n+j-1})\} \leq c\|W_n - Z_{n+j-1}\|, \quad j = 1, 2.$$

Then for a cyclic order optimization of components $g_1(\cdot)$ and $g_2(\cdot)$, the following holds:

$$\|W_{n1} - W^*\|^2 \leq \|W_n - W^*\|^2 - \frac{2}{L_n}(F(W_n) - F(W^*)) + 18c^2/L_n^2 \tag{3.1}$$

where $W_{n1} = \text{Prox}_{(\tau/L_n)g_2}(W_n)$ and W^* is a solution of (2.1) and (2.2).

Proof. Refer Bertsekas (2010) for a detailed proof. □

The following proposition will be used in proving the convergence result of the proposed algorithm.

Proposition 1. *Let $\{W_n,\ L_n, n = 1, 2, \ldots\}$ be the sequence generated by algorithm Section 2.2.2. Let* **c** *be a constant as defined in lemma (3.1). Then,*

$$F(W_n) - F(W^*) \leq \frac{L_n}{4}\left(\|W_{n-1} - W^*\|^2 - \|W_n - W^*\|^2\right) + \frac{9c^2}{2L_n} - \frac{\lambda}{2}\langle W^* - W_n, \bigtriangledown\|W_n\|_1\rangle. \tag{3.2}$$

Proof. Proof of Proposition 1 is given in the Appendix. □

We give below the convergence of the proposed algorithm (Section 2.2.2).

Theorem 3.2. *Let $\{W_k,\ k = 1, 2, \ldots\}$ be the sequence generated by the algorithm in Section 2.2.2. Let* **c** *be a constant as defined in Lemma 3.1. In addition, we assume that there exists a constant $M < \infty$ such that $\sum_{n=1}^{\infty} |\langle W^* - W_n, \bigtriangledown\|W_n\|_1\rangle| < M$.*

We have,

$$F(W_k) - F(W^*) \leq \left(\frac{\gamma L \|W^0 - W^*\|_F^2 + 18c^2 + M}{4k} \right)$$

where $L > 0$ is the least upper Lipschitz constant of the gradient of $f(W)$ as defined in (2.5) and $\gamma > 1$ is a constant as defined in the algorithm in Section 2.2.2.

Proof. Note that $\frac{L_n}{\gamma} \leq L \leq Ł_n$, for all $n = 1, 2, \ldots$ Using Proposition 1, by adding $F(W_n) - F(W^*)$ over $n = 1, 2, \ldots,$ we get the desired result. □

Rolfs et al. (2012) consider the estimation of the inverse covariance matrix with single lasso penalty. They also used the proximal gradient algorithm for optimization. The rate of convergence in case of joint penalty is different from that of Rolfs et al. as they are derived under different settings. Due to non-smoothness of the trace norm, the optimal first order black box method for solving such problems has the convergence rate of $O(k^{-\frac{1}{2}})$. The proximal gradient algorithm uses the special structure of the trace and the ℓ_1 norm which improve the convergence rate for joint penalty of order $O(k^{-1})$.

4. Simulation and results

To implement the proposed method, we perform the simulation study for various choices of the inverse covariance matrix. We generate random sample of observations from multivariate Gaussian distribution for varying n and p. We consider different types of inverse covariance matrix as below:

(i) *Hub graph*: The rows/columns are partitioned into J equally-sized disjoint groups: $\{V_1 \cup V_2 \cup, \ldots, \cup V_J\} = \{1, 2, \ldots, p\}$, each group is associated with a *pivotal* row k. Let $|V_1| = s$. We set $w_{i,j} = w_{j,i} = \rho$ for $i \in V_k$ and $w_{i,j} = w_{j,i} = 0$ otherwise. In our experiment, $J = [p/s]$, $k = 1, s + 1, 2s + 1, \ldots,$ and we always set $\rho = 1/(s + 1)$ with $J = 20$.

(ii) *Neighborhood graph*: We first uniformly sample $(y_1, y_2, \ldots, y_n)$ from a unit square. We then set $w_{i,j} = w_{j,i} = \rho$ with probability $(\sqrt{2\pi})^{-1} \exp(-4\|y_i - y_j\|^2)$. The remaining entries of W are set to be zero. The number of nonzero off-diagonal elements of each row or column is restricted to be smaller than $[1/\rho]$. In this paper, ρ is set to be 0.245.

(iii) *Block diagonal matrix*: In this setting W is a block diagonal matrix with varying block size. For $p = 50$ and $p = 100$ number of blocks is fixed to be 5 and for $p = 200$, number of blocks is fixed to be 10. Each block of the inverse covariance matrix is taken to be a Toeplitz type matrix as in case (iv).

(iv) *Toeplitz matrix*: We set $w_{i,j} = 2$ for $i = j$; $w_{i,j} = |0.75|^{|i-j|}$ for $|i - j| = 1, 2, 3, 4$; and $w_{i,j} = 0$ otherwise.

For all these choices of inverse covariance matrices, we generate random numbers from multivariate normal distribution with varying n and p. We set $n = 50, 100, 200$ and $p = 50, 100, 200$. The performance of the proposed method is compared to graphical lasso and SPICE estimates of the inverse covariance matrix. The joint penalty estimate of the inverse covariance matrix was computed using R software version 3.0.2 based on the algorithm in Section 2.2.2. The graphical lasso estimate of the inverse covariance matrix was computed using R package "glasso" (http://statweb.stanford.edu/tibs/glasso/). In "glasso" there is option of not penalizing the diagonal elements by setting the option "penalize.diagonal = FALSE" which gives SPICE estimate of the inverse covariance matrix. For each of the inverse covariance matrix estimate, we calculate Kullback–Leibler (KL) Loss, Average Relative Error (ARE) and Mean Squared Error (MSE) as below:

$$\text{KL Loss } (W, \hat{W}) = -\log(\det(\hat{W})) + \text{tr}(W^{-1}\hat{W}) + \log(\det(W)) - p.$$

$$\text{MSE } (W, \hat{W}) = \|W - \hat{W}\|_2^2.$$

$$\text{ARE } (W, \hat{W}) = |\log(f(S, \hat{W})) - \log(f(S, W))| / \log(f(S, W))$$

where $f(\cdot, \cdot)$ is the density of multivariate Gaussian distribution and S is the sample covariance matrix. Estimation of λ and τ is implemented using StARS Liu et al. (2010) which is described below.

Given a sample of size n, the method generates N samples of size b, where $b < n$. In our setting for $n < 200$, we choose $b = 0.8n$ and for $n \geq 200$, $b = 10\sqrt{n}$. For each of these N samples, an estimate of the inverse covariance matrix is obtained. For each entry of the inverse covariance matrix, a measure of instability is calculated based on all N estimates. Finally a regularization parameter is selected which minimizes the average instability over all possible entries of the estimated inverse covariance matrix. In practice this method tends to select least amount of the regularization parameter that simultaneously makes estimates of the inverse covariance matrix sparse and replicable under random sampling. StARS is used for estimating the penalization parameter for all the competing methods as given in simulation *viz.* Joint Penalty, Graphical Lasso and SPICE.

The simulation results are given in the Appendix. The numbers in bracket are standard error of the estimate based on 20 simulations. For a hub graph type inverse covariance matrix, the proposed method consistently outperforms other two in terms of KL Loss, ARE and MSE except for $n = 50,\ p = 200$ and $n = 100,\ p = 200$ where "glasso" performs better than other two methods.

For a neighborhood type inverse covariance matrix, the proposed method dominates the graphical lasso and SPICE in terms of KL-loss. Note that the KL-loss estimate is obtained by plugging in the estimated inverse covariance matrix in the density, which in practice is a better criteria of evaluating the performance of different methods if the underlying Gaussian

assumption is true. Therefore in practice if there is some evidence of observations to be Gaussian, the proposed method will outperform other two methods for a neighborhood and hub type inverse covariance matrix structure. We get mixed performance for all the three methods in terms of MSE and ARE.

For the block type inverse covariance matrix, the proposed method outperforms other methods for smaller sample size in terms of KL-loss. For $n = 200$ we get mixed performance of the methods. This suggests that a joint penalty yields a better estimate for small sample settings. In terms of ARE, we get mixed performance of all the methods. In terms of MSE, SPICE and glasso perform better than the Joint Penalty approach. This shows that performance of a method also depends upon the type of inverse covariance matrix as well as the criteria of performance evaluation. For Toeplitz type inverse covariance matrix, all the methods give mixed performance in terms of KL-loss. However graphical lasso outperforms Joint Penalty and SPICE method in terms of ARE whereas SPICE estimate has better performance than glasso and Joint Penalty estimate in terms of the mean squared error.

Overall Joint Penalty method has a better performance than other two methods for hub, neighborhood and block type of inverse covariance matrix. For Toeplitz type inverse covariance matrix, the graphical lasso and SPICE have better performances than Joint Penalty method. In terms of KL-loss, the proposed Joint Penalty approach gives better performance. This suggests that if the Gaussian assumption holds well in practice for the observations, the Joint Penalty would be a better alternative to estimate the covariance structure and hence the underlying graphical model structure. The initial simulation results also show that the proposed method performs well for small n and large p which is encouraging as in practice; we often have very small sample size as compared to the number of unknown parameters (e.g. micro-array data). We also see that the performance of methods varies for three choices of loss functions as they have different formulas. Note that the graphical lasso algorithm estimates the covariance matrix rather than the inverse covariance matrix. Whereas Joint Penalty and SPICE methods, both estimate the inverse covariance matrix. The values of MSE, KL-loss and average relative are substantially different for different choice of inverse covariance matrix. The hub and the neighborhood type inverse covariance matrix have lower MSE as compared to other type of inverse covariance matrix. This shows that the loss function value of the estimate depends upon the structure of the underlying true inverse covariance matrix. For fixed p the estimates tend to improve for increasing sample size. However for fixed n, as expected, the estimate's performance goes down for increasing p.

5. Summary

The proposed method imposes joint penalty which is more flexible for penalizing different entries of the inverse covariance matrix than Graphical Lasso and SPICE. The simulation shows that the performance of the method also depends upon the evaluation criteria and the underlying covariance structure. The proposed methods have better performance than other two methods for at least three of the four underlying covariance matrix settings. The method can be extended to problems where one has an arbitrary number of convex penalty constraints. Under mild conditions the algorithm achieves sublinear rate of convergence which makes it an attractive choice for many optimization problems. On the other hand these estimates of the inverse covariance matrix are consistent as MSE, ARE and KL-Loss decrease rapidly (as well as the corresponding standard errors) for increasing sample size.

Acknowledgments

I would like to express my deep gratitude to Professor Hira Koul for his valuable and constructive suggestions during the planning and development of this research work. Also I would like to thank the Co-Editor of the journal and both referees for their feedback and helpful comments. I specially thank the referee for pointing out a mistake in the simulation which led to improvement in the manuscript.

Appendix

Proof of Lemma 2.1. Let M^* be a solution for (2.7) which exists because (2.7) is the convex optimization problem. Then the following subgradient optimality condition holds (Boyd et al., 2004)

$$\mathbf{0} \in M^* - M + \lambda \partial \|M^*\|_1 \tag{A.1}$$

where $((\partial \|M\|_1))_{ij} = \partial |m_{ij}|$ and is given by

$$\partial |m_{ij}| = \begin{cases} +1 & \text{if } m_{ij} > 0 \\ -1 & \text{if } m_{ij} < 0 \\ \in [-1, 1] & \text{if } m_{i,j} = 0 \end{cases} \quad i = 1, 2, \ldots, m;\ j = 1, 2, \ldots, n.$$

Note that (A.1) is satisfied if and only if $|m_{ij}| \leq \lambda$ and therefore the optimal solution is given by

$$m^*_{ij} = \text{sign}(m_{ij})(m_{ij} - \lambda)_+.$$

This completes the proof. □

Proof of Lemma 2.2. Let L^* be a solution to (2.8). Then the following subgradient optimality condition holds:

$$0 \in L^* - M + \tau \partial \|L^*\|_*. \tag{A.2}$$

Let $W = U\Sigma_\tau V^T$, we shall show that this choice of W satisfies the above optimality condition. The subdifferential $\partial\|W\|_*$ of $\|W\|_*$ is given by Bach (2008)

$$\partial \|W\|_* = \left\{ UV^T + H \text{ such that } H \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}},\ \|H\|_2 \le 1,\ U^T H = 0 \text{ and } HV = 0 \right\}.$$

Therefore

$$W - M + \tau\partial\|W\|_* = U\Sigma_\tau V^T - U\Sigma V^T + \tau(UV^T + H).$$

Multiplying both sides by UU^T, and noting that $UU^T = I$ we obtain

$$\begin{aligned} W - M + \partial\|W\|_* &= UU^T(U\Sigma_\tau V^T - U\Sigma V^T + \tau(UV^T + H)) \\ &= U\Sigma_\tau V^T - U(\Sigma - \tau I)V^T + UU^T H = 0. \end{aligned}$$

Therefore, $W = U\Sigma_\tau V^T$ is a solution to (2.8), this completes the proof. □

Proof of Proposition 1. For such choice of L_n we have

$$\begin{aligned} F(W_n) &= f(W_n) + \lambda\|W_n\|_1 + \gamma\|W_n\|_* = Q_{L_n}(W_n, W_{n1}) + \lambda\|W_n\|_1 + \tau\|W_n\|_* \\ &= f(W_{n1}) + \frac{L_n}{2}\|W_n - W_{n1}\|^2 + \langle W_n - W_{n1}, \bigtriangledown f(W_{n1})\rangle + \lambda\|W_n\|_1 + \tau\|W_n\|_*. \end{aligned}$$

Also we have,

$$\begin{aligned} F(W^*) &= f(W^*) + \lambda\|W^*\|_1 + \gamma\|W^*\|_* \\ f(W^*) &\ge f(W_{n1}) + \langle W^* - W_{n1}, \bigtriangledown f(W_{n1})\rangle \\ \|W^*\|_1 &\ge \|W_n\|_1 + \langle W^* - W_n, \bigtriangledown\|W_n\|_1\rangle \\ \|W^*\|_* &\ge \|W_n\|_* + \langle W^* - W_n, \bigtriangledown\|W_n\|_*\rangle. \end{aligned}$$

We get,

$$F(W^*) - F(W_n) \ge -\frac{L_n}{2}\|W_n - W_{n1}\|^2 + \langle W^* - W_n, \bigtriangledown f(W_{n1}) + \lambda \bigtriangledown \|W_n\|_1 + \tau \bigtriangledown \|W_n\|_*\rangle. \tag{A.3}$$

Note that W_n is a solution of

$$\bigtriangledown f(W_{n-1}) + L_n(W_n - W_{n1}) + \tau \bigtriangledown \|W_n\|_* = 0 \quad \text{(using (2.4))}.$$

Therefore (A.3) becomes

$$F(W^*) - F(W_n) \ge -\frac{L_n}{2}\left(\|W_{n1} - W_n\|^2 + 2\langle W_{n1} - W_n, W_n - W^*\rangle - \frac{2\lambda}{L_n}\langle W^* - W_n, \bigtriangledown\|W_n\|_1\rangle\right).$$

We know that for any three matrices A, B, C

$$\|B - A\|^2 + 2\langle B - A, A - C\rangle = \|B - C\|^2 - \|A - C\|^2.$$

Using this, we get

$$F(W_n) - F(W^*) \le \frac{L_n}{2}\left(\|W_{n1} - W^*\|^2 - \|W_n - W^*\|^2 - \frac{2\lambda}{L_n}\langle W^* - W_n, \bigtriangledown\|W_n\|_1\rangle\right).$$

Using Lemma 3.1, we get

$$F(W_n) - F(W^*) \le \frac{L_n}{4}\left(\|W_{n-1} - W^*\|^2 - \|W_n - W^*\|^2\right) + \frac{9c^2}{2L_n} - \frac{\lambda}{2}\langle W^* - W_n, \bigtriangledown\|W_n\|_1\rangle,$$

which completes the proof. □

Appendix A. Hub type inverse covariance matrix

See Tables A.1–A.3.

Table A.1
Average KL-loss and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	**2.852** (**0.1941**)	**8.032** (**0.0734**)	21.89 (1.5192)
Glasso	3.237 (0.1725)	8.449 (0.0847)	**20.15** (**1.4906**)
SPICE	3.172 (0.1352)	10.86 (0.1126)	32.33 (3.2141)
$n = 100$			
Mixed penalty	**1.652** (**0.0429**)	**5.111** (**0.0573**)	12.56 (1.1628)
Glasso	1.9 (0.0149)	5.546 (0.0716)	**11.65** (**0.2497**)
SPICE	1.884 (0.121)	6.429 (0.079)	15.37 (0.266)
$n = 200$			
Mixed penalty	**0.8774** (**0.0542**)	**3.131** (**0.0831**)	**6.54** (**0.06**)
Glasso	1.047 (0.0563)	3.454 (0.0984)	7.578 (0.0777)
SPICE	0.9612 (0.0556)	3.68 (0.0763)	8.997 (0.0645)

Table A.2
Average relative error and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	0.1041 (0.0077)	**0.1985** (**0.001**)	0.4284 (0.0221)
Glasso	**0.08216** (**0.0147**)	0.1996 (0.001)	**0.4143** (**0.0259**)
SPICE	0.1267 (0.0073)	0.2695 (0.0012)	0.5179 (0.0328)
$n = 100$			
Mixed penalty	**0.05224** (**0.0052**)	**0.1103** (**0.0023**)	0.2274 (0.0169)
Glasso	0.05226 (0.0052)	0.116 (0.0021)	**0.2119** (**0.0032**)
SPICE	0.07454 (0.0051)	0.1489 (0.0024)	0.2617 (0.0033)
$n = 200$			
Mixed penalty	**0.02989** (**0.0014**)	**0.05962** (**0.0014**)	**0.1081** (**0.0011**)
Glasso	0.03077 (0.0013)	0.06483 (0.0013)	0.1213 (0.0011)
SPICE	0.04077 (0.0013)	0.07971 (0.0013)	0.1426 (0.0012)

Table A.3
MSE and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	**2.176** (**0.0562**)	**3.58** (**0.0211**)	6.346 (0.3268)
Glasso	2.346 (0.0373)	3.582 (0.0237)	**5.617** (**0.301**)
SPICE	2.313 (0.0382)	4.37 (0.0686)	8.512 (0.5666)
$n = 100$			
Mixed penalty	**1.771** (**0.0211**)	**2.951** (**0.0107**)	4.77 (0.2974)
Glasso	1.891 (0.0237)	3.04 (0.0164)	**4.321** (**0.0341**)
SPICE	1.868 (0.0686)	3.358 (0.0244)	5.531 (0.052)
$n = 200$			
Mixed penalty	**1.322** (**0.0372**)	**2.366** (**0.0239**)	**3.437** (**0.0122**)
Glasso	1.442 (0.0341)	2.477 (0.0271)	3.609 (0.0128)
SPICE	1.352 (0.0402)	2.559 (0.0197)	4.183 (0.0172)

Appendix B. Neighborhood graph type inverse covariance matrix

See Tables B.1–B.3.

Table B.1
Average KL-loss and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	**3.389** (**0.1122**)	**7.753** (**0.17**)	26.94 (0.3287)
Glasso	3.615 (0.0706)	8.035 (0.1373)	**18.25** (**0.2979**)
SPICE	4.077 (0.1574)	10.82 (0.2606)	27.21 (0.3258)
$n = 100$			
Mixed penalty	**2.403** (**0.051**)	**4.601** (**0.114**)	**11.28** (**0.3824**)
Glasso	2.642 (0.0465)	5.105 (0.1174)	12.4 (0.2531)
SPICE	2.539 (0.0978)	5.822 (0.114)	16 (0.2897)
$n = 200$			
Mixed penalty	**1.36** (**0.0273**)	**2.55** (**0.0768**)	**6.86** (**0.1132**)
Glasso	1.521 (0.0266)	2.957 (0.0848)	7.923 (0.1185)
SPICE	1.403 (0.0407)	3.1 (0.076)	9.316 (0.1156)

Table B.2
Average relative error and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	0.1065 (0.004)	**0.1912** (**0.0048**)	0.4579 (0.0048)
Glasso	**0.09335** (**0.0064**)	0.1917 (0.0047)	**0.3597** (**0.0069**)
SPICE	0.1483 (0.0034)	0.2634 (0.0059)	0.4459 (0.005)
$n = 100$			
Mixed penalty	0.05349 (0.0044)	**0.1101** (**0.0026**)	**0.2021** (**0.0046**)
Glasso	**0.05296** (**0.0047**)	0.1152 (0.0025)	0.2145 (0.0029)
SPICE	0.075 (0.0048)	0.1475 (0.0027)	0.2636 (0.0029)
$n = 200$			
Mixed penalty	0.03255 (0.0011)	**0.06201** (**0.0015**)	**0.1131** (**0.001**)
Glasso	**0.03041** (**0.0031**)	0.06676 (0.0015)	0.1251 (0.001)
SPICE	0.04151 (0.0029)	0.08115 (0.0015)	0.1468 (0.0011)

Table B.3
MSE and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	**2.425** (**0.0251**)	3.536 (0.0591)	7.086 (0.0357)
Glasso	2.535 (0.0235)	**3.509** (**0.0437**)	**5.063** (**0.0363**)
SPICE	2.645 (0.0488)	4.411 (0.0805)	7.274 (0.0347)
$n = 100$			
E2	2.086 (0.0234)	**2.769** (**0.0342**)	**4.32** (**0.0783**)
Glasso	2.216 (0.0334)	2.884 (0.0811)	4.363 (0.0362)
SPICE	**2.066** (**0.0311**)	3.159 (0.0775)	5.463 (0.0349)
$n = 200$			
E2	1.605 (0.0197)	**2.129** (**0.0244**)	**3.439** (**0.0283**)
Glasso	1.737 (0.0275)	2.271 (0.0265)	3.619 (0.0275)
SPICE	**1.578** (**0.0238**)	2.347 (0.0147)	4.125 (0.0236)

Appendix C. Block type inverse covariance matrix

See Tables C.1–C.3.

Table C.1
Average KL-loss and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	**6.112** (**0.0426**)	14.16 (0.3076)	**34.52** (**0.3266**)
Glasso	6.459 (0.0784)	**13.72** (**0.1317**)	34.66 (0.2849)
SPICE	6.272 (0.0299)	14.73 (0.4188)	45.47 (0.487)
$n = 100$			
Mixed penalty	**4.8** (**0.0955**)	**10.94** (**0.1326**)	**25.95** (**0.2334**)
Glasso	4.993 (0.1186)	11.05 (0.0987)	26.71 (0.2455)
SPICE	4.809 (0.0909)	11.03 (0.0765)	31.23 (0.2415)
$n = 200$			
Mixed penalty	3.349 (0.0992)	**7.743** (**0.1447**)	18.65 (0.0642)
Glasso	3.595 (0.0636)	8.069 (0.1492)	**18.47** (**0.1883**)
SPICE	**3.249** (**0.0766**)	7.753 (0.1382)	19.83 (0.5733)

Table C.2
Average relative error and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	0.1716 (0.0278)	0.359 (0.009)	**0.6643** (**0.0065**)
Glasso	**0.1474** (**0.0189**)	**0.2371** (**0.0266**)	0.676 (0.0055)
SPICE	0.2146 (0.0124)	0.3747 (0.029)	0.8935 (0.0062)
$n = 100$			
Mixed penalty	0.04998 (0.0167)	0.1924 (0.0179)	**0.382** (**0.0048**)
Glasso	**0.03783** (**0.0143**)	**0.1469** (**0.0225**)	0.4109 (0.0041)
SPICE	0.08744 (0.0131)	0.1966 (0.004)	0.5208 (0.0033)
$n = 200$			
Mixed penalty	0.01631 (0.0053)	0.054 (0.0115)	0.2026 (0.0008)
Glasso	0.02754 (0.0073)	**0.02849** (**0.006**)	**0.1538** (**0.0192**)
SPICE	**0.01282** (**0.0036**)	0.07825 (0.0082)	0.2463 (0.0239)

Table C.3
MSE and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	10.5 (0.0982)	16.3 (0.0433)	**22.18** (**0.0435**)
Glasso	10.84 (0.0796)	16.07 (0.1187)	22.36 (0.0378)
SPICE	**10.23** (**0.0385**)	**15.3** (**0.0656**)	22.54 (0.0758)
$n = 100$			
Mixed penalty	9.952 (0.1008)	14.87 (0.1281)	21.09 (0.0509)
Glasso	10.22 (0.0784)	15.41 (0.0926)	21.28 (0.0513)
SPICE	**9.625** (**0.0557**)	**14.71** (**0.0606**)	**20.82** (**0.0589**)
$n = 200$			
Mixed penalty	9.072 (0.1217)	13.91 (0.1462)	19.68 (0.0305)
Glasso	9.438 (0.0389)	14.46 (0.0652)	20.44 (0.1438)
SPICE	**8.813** (**0.0504**)	**13.6** (**0.1365**)	**19.36** (**0.1194**)

Appendix D. Toeplitz type inverse covariance matrix

See Tables D.1–D.3.

Table D.1
Average KL-loss and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	**6.114** (**0.1082**)	14.72 (0.3109)	36.36 (0.4745)
Glasso	6.481 (0.0788)	**14.21** (**0.291**)	**35.3** (**0.7003**)
SPICE	6.525 (0.1855)	15.81 (0.779)	47.34 (0.2559)
$n = 100$			
Mixed penalty	4.909 (0.0882)	11.34 (0.1637)	**27.36** (**0.1706**)
Glasso	5.169 (0.0989)	**11.28** (**0.0695**)	27.5 (0.1706)
SPICE	**4.889** (**0.1004**)	11.42 (0.1086)	31.78 (0.1706)
$n = 200$			
Mixed penalty	3.89 (0.047)	8.246 (0.0444)	19.61 (0.1693)
Glasso	4 (0.0336)	8.53 (0.0441)	**19.09** (**0.1432**)
SPICE	**3.74** (**0.0481**)	**8.199** (**0.0331**)	19.86 (0.5867)

Table D.2
Average relative error and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	0.1369 (0.026)	0.3641 (0.0185)	0.6727 (0.0051)
Glasso	**0.1289** (**0.026**)	**0.2688** (**0.0292**)	**0.6376** (**0.0398**)
SPICE	0.2181 (0.0247)	0.4039 (0.0309)	0.8861 (0.002)
$n = 100$			
Mixed penalty	0.0393 (0.0093)	0.19 (0.024)	0.3965 (0.0054)
Glasso	**0.03887** (**0.0093**)	**0.1278** (**0.006**)	**0.3948** (**0.0304**)
SPICE	0.08752 (0.0093)	0.1944 (0.0059)	0.4985 (0.0318)
$n = 200$			
Mixed penalty	0.01466 (0.0029)	0.04256 (0.0093)	0.1971 (0.0024)
Glasso	**0.01453** (**0.0021**)	**0.03121** (**0.0064**)	**0.1316** (**0.0022**)
SPICE	0.01348 (0.0021)	0.06633 (0.0064)	0.2013 (0.0208)

Table D.3
MSE and standard error over 20 replications.

	$p = 50$	$p = 100$	$p = 200$
$n = 50$			
Mixed penalty	11.64 (0.0281)	16.3 (0.072)	**23.46** (**0.0045**)
Glasso	11.91 (0.0233)	16.88 (0.0635)	23.72 (0.0961)
SPICE	**11.29** (**0.0464**)	**16.23** (**0.0571**)	23.78 (0.0258)
$n = 100$			
Mixed penalty	11.16 (0.0724)	15.68 (0.1016)	22.38 (0.0333)
Glasso	11.36 (0.0712)	16.3 (0.0145)	22.68 (0.0333)
SPICE	**10.75** (**0.0853**)	**15.55** (**0.0187**)	**22.16** (**0.0333**)
$n = 200$			
Mixed penalty	10.55 (0.0295)	15.01 (0.1505)	21.02 (0.0385)
Glasso	10.72 (0.0184)	15.39 (0.013)	21.91 (0.0343)
SPICE	**10.17** (**0.0245**)	**14.66** (**0.0172**)	**20.9** (**0.0822**)

References

Argyriou, A., Evgeniou, T., Pontil, M., 2008. Convex multi-task feature learning. Mach. Learn. 73 (3), 243–272.
Bach, F., 2008. Consistency of trace norm minimization. J. Mach. Learn. Res. 9, 1019–1048.
Bach, F., Jenatton, R., Mairal, J., Obozinski, G., 2011. Convex optimization with sparsity-inducing norms. In: Optimization for Machine Learning. MIT Press.
Banerjee, O., El Ghaoui, L., d'Aspremont, A., 2008. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. J. Mach. Learn. Res. 9, 485–516.
Bertsekas, D.P., 2010. Incremental gradient, subgradient, and proximal methods for convex optimization, a survey. In: Labaratory for Information and Decision Systems Report LIDS-P-2848. MIT.
Bickel, P., Levina, E., 2008. Regulatized estimation of large covariance matrices. Ann. Statist. 36, 199–227.
Candès, E., Recht, B., 2009. Foundations of Computational Mathematics. Springer.
Dahl, J., Vandenberghe, L., Roychowdhury, V., 2008. Covariance selection for non-chordal graphs via chordal embedding. Optim. Methods Softw. 23 (4), 501–520. Methods and Software.
Dempster, A., 1972. Covariance selection. Biometrika 32, 95–108.
Fazel, M., 2002. Matrix Rank Minimization with Applications. Elec. Eng. Dept, Stanford University (March).
Friedman, J., Hastie, T., Tibshirani, R., 2007. Sparse inverse covariance estimation with the graphical Lasso. Biostatistics 9 (3), 432–441. 2008 Jul.
Liu, H., Roeder, K., Wasserman, L., 2010. Stability approach to regularization selection (StARS) for high dimensional graphical models. In: Proceedings of the Twenty-Third Annual Conference on Neural Information Processing Systems (NIPS).
Meinshausen, , Bühlmann, P., 2006. High dimensional graphs and variable selection with the Lasso. Ann. Statist. 34, 1436–1462.
Nesterov, Yu., 2005. Smooth minimization of non-smooth functions. Math. Program. 127–152.
Raguet, H., Fadili, J., Peyre, G., 2011. Generalized forward-backward splitting. Arxiv preprint arXiv:1108.4404.
Rockafellar, R.T., 1976. Monotone operators and proximal point algorithm. SIAM J. Control Optim. 14 (5).
Rolfs, B., Rajaratnam, B., Guillot, D., Wong, I., Maleki, A., 2012. Iterative thresholding algorithm for sparse inverse covariance estimation. In: Adv. Neural Inf. Process. Syst. (NIPS), pp. 1583–1591.
Rothman, A.J., Bickel, P.J., Levina, E., Zhu, J., 2008. Sparse permutation invariant covariance estimation. Electron. J. Stat. 2, 494–515.
Sheena, Y., Gupta, A., 2003. Estimation of the multivariate normal covariance matrix under some restrictions. Statist. Decisions 21, 327–342.
Tomioka, R., Aihara, K., 2007. Classifying matrices with a spectral regularization. In: Proc. 24th Int. Conf. Machine Learning, pp. 895–902.
Vandenberghe, L., Boyd, S., 2004. Convex Optimization. Cambridge University Press.
Wainwright, M., Ravikumar, P., Lafferty, J.D., 2006. High-dimensional graphical model selection using L_1-regularized logistic regression. In: Proceedings of Advances in Neural in formation Processing Systems.
Won, J., Lim, J., Kim, S., Rajaratnam, B., 2012. Condition number regularized covariance estimation. J. R. Stat. Soc. Ser. B.
Yuan, M., 2009. Sparse inverse covariance matrix estimation via linear programming. J. Mach. Learn. Res. 11, 2261–2286.
Yuan, M., Lin, Y., 2007. Model selection and estimation in the Gaussian graphical model. Biometrika 94 (1), 19–35.
Zhou, S., Rutimann, P., Xu, M., Buḧlmann, P., 2011. High-dimensional covariance estimation based on Gaussian graphical models. J. Mach. Learn. Res. (in press).