Inverse Covariance Estimation with Structured Groups

Shaozhe Tao
University of Minnesota
taoxx120@umn.edu

Yifan Sun
Technicolor Research
yifan.sun@technicolor.com

Daniel Boley
University of Minnesota
boley@cs.umn.edu

Abstract

Estimating the inverse covariance matrix of p variables from n observations is challenging when $n \ll p$, since the sample covariance matrix is singular and cannot be inverted. A popular solution is to optimize for the ℓ_1 penalized estimator; however, this does not incorporate structure domain knowledge and can be expensive to optimize. We consider finding inverse covariance matrices with group structure, defined as potentially overlapping principal submatrices, determined from domain knowledge (*e.g.* categories or graph cliques). We propose a new estimator for this problem setting that can be derived efficiently via the Frank-Wolfe method, leveraging chordal decomposition theory for scalability. Simulation results show significant improvement in sample complexity when the correct group structure is known. We also apply these estimators to 14,910 stock closing prices, with noticeable improvement when group sparsity is exploited.

1 Introduction

The inverse covariance matrix is of interest to statisticians in biology, finance, machine learning, *etc.* In finance, it is a key ingredient for computing value-at-risk, a factor in portfolio optimization. In graphical models, for p random variables with true covariance matrix C, the sparsity pattern of C^{-1} gives the conditional independence between each pair of variables. However, if $n \ll p$, then the sample covariance matrix $\hat{C}$ is not invertible, and the pseudoinverse $\hat{C}^\dagger$ is inaccurately dense. The most popular alternative is the graphical LASSO (G-LASSO) estimator [Yuan and Lin, 2007; Banerjee *et al.*, 2008], the solution to

$$\begin{array}{ll} \underset{X}{\text{minimize}} & -\text{logdet}(X) + \mathbf{tr}(\hat{C}X) + \rho\|X\|_1 \\ \text{subject to} & X \succeq 0 \end{array} \tag{1}$$

for some regularization parameter $\rho > 0$. By adding a sparsity-inducing regularizer, the effective degrees of freedom are reduced, and it has been shown that the resulting estimator has a much lower sample complexity than inverting $\hat{C}$. However, this estimator does not incorporate any *prior structural knowledge* from the problem domain. Additionally, in general solving (1) is computationally challenging if p is large.

Most existing methods for solving (1) require a sequence of eigenvalue decompositions (EDs) [Banerjee *et al.*, 2006; Friedman *et al.*, 2008; d'Aspremont *et al.*, 2008; Yuan, 2012; Rolfs *et al.*, 2012]. This is expensive if p is large; a dense ED requires $O(p^3)$ computations, and sparse EDs (like Lanczos type methods) are not suitable when the full eigenvalue spectrum is needed. (Used this way, they may be far slower than dense methods.) There are some exceptions; for example [Scheinberg and Rish, 2009] updates each row in a block coordinate descent fashion, and maintains inverses using only rank-2 updates; [Dahl *et al.*, 2008] uses chordal decomposition to compute Newton steps efficiently in an interior point solver; and [Meinshausen and Bühlmann, 2006] uses neighborhood selection, which enforces the conditional independence condition one variable at a time. These methods are more or less intuitive, relying on general convex optimization principles; however, their scalability is limited. On the other end of the spectrum is BIG-QUIC [Hsieh *et al.*, 2013] which can solve up to 1 million variables. This breakthrough method simultaneously makes estimates of the matrix sparsity while also optimizing for it, and updating via block coordinate descent with carefully chosen (non-principal) submatrices. However, it demonstrates the tradeoff between simplicity and scalability; there are many intricate details for a successful implementation.

At the same time, there has been growing interest in the statistics community to exploit *group structure* in the estimators [Negahban *et al.*, 2009; Chandrasekaran *et al.*, 2012; Obozinski *et al.*, 2011]. For example, [Mazumder and Hastie, 2012] proposes thresholding the sample covariance matrix in order to identify fully-connected components of the graphical model, effectively decomposing (1). More recently, [Hosseini and Lee, 2016] learns overlapping submatrix groups probablistically and penalizes accordingly. To our knowledge, this is the only work that addresses overlapping group sparsity in matrices; however, repeated full EDs are still needed to find the inverse covariance estimate.

We propose an estimator that exploits group structure, where a matrix group is described as either a principal submatrix or the matrix diagonal in Section 2. The solution X is then described as a sum of these possibly overlapping components. We then apply the Frank-Wolfe method to derive

the estimator in Section 3. The algorithm at each iteration decomposes into parallelizable eigenvalue computations on the submatrices. In this way, unlike [Mazumder and Hastie, 2012; Hosseini and Lee, 2016], this estimator *explicitly* uses the predetermined groups as components for decomposition, thus using group structure to improve both performance and computation cost. In Section 4 we give simulation results, which demonstrate that knowing and exploiting group structure significantly improves sample complexity. Finally, in Section 5, we show the performance of our model on the stock datasets.

2 Group Norm Constrained Estimator

For an index set $\gamma \subset \{1, \ldots, p\}$ and a vector $x \in \mathbb{R}^p$, define x_γ as the subvector of x indexed by γ; for the reverse, define the *augmenting linear map* $\gamma : \mathbb{R}^{|\gamma|} \to \mathbb{R}^p$ such that

$$(A_\gamma u)_\gamma = u, \quad (A_\gamma u)_i = 0 \text{ if } i \notin \gamma.$$

In [Obozinski *et al.*, 2011], the *overlapping group norm* is defined as the solution to

$$\|x\|_G \;=\; \min_{u_1,\ldots,u_l} \left\{ \sum_{k=1}^{l} w_k \|u_k\| \;:\; x = \sum_{k=1}^{l} A_{\gamma_k} u_k \right\} \quad (2)$$

for some proper norm $\|\cdot\|$ and nonnegative weights $w_1, \ldots, w_l$. (A common choice is $w_k = |\gamma_k|^{-1}$.) Used as a penalty term or in a constraint, this norm is shown to promote *group structure*; a small subset of index sets γ_k are "active", and $x_i = 0$ whenever i is not in an active set.

We extend this concept to matrices, by defining groups implicitly through index sets $\beta \subset \{1, \ldots, p\}$, where $X_{\beta,\beta}$ is the submatrix of X selected by the rows and columns indicated by β. Let $\mathbb{S}^p$ denote the set of $p \times p$ matrices. We define $\mathcal{A}_\beta : \mathbb{S}^{|\beta|} \to \mathbb{S}^p$ such that

$$(\mathcal{A}_\beta(U))_{\beta,\beta} = U, \quad \mathcal{A}_\beta(U)_{i,j} = 0 \text{ if } i \notin \beta \text{ or } j \notin \beta$$

and extend the overlapping group norm as the solution

$$\|X\|_G := \begin{cases} \min\limits_{v,U_1,\ldots,U_l} & w_0\|v\|_2 + \sum\limits_{k=1}^{l} w_k \|U_k\|_F \\ \text{subj. to} & X = \mathbf{diag}(v) + \sum\limits_{k=1}^{l} \mathcal{A}_{\beta_k}(U_k). \end{cases} \quad (3)$$

Here, X can be considered as the sum of smaller principal submatrices U_k and a diagonal term v, and w_k are nonnegative scalar weights. Note that the affine constraint imposes a sparsity pattern on X; if X does not adhere to this pattern (*e.g.* $X_{ij} \neq 0$ for $i, j \notin \beta_k, \forall k$) then we define $\|X\|_G = \infty$.

2.1 Our estimator

Given p random variables, define $C \in \mathbb{S}^p$ and $\hat{C} \in S^p$ as the true and sample covariance matrices. The *group norm regularized graphical LASSO estimator (NG-LASSO)* is the solution to

$$\begin{aligned} \min_X \quad & -\text{logdet}(X) + \mathbf{tr}(\hat{C}X) \\ \text{s.t.} \quad & X = \sum_{k=1}^{l} \mathcal{A}_{\beta_k}(U_k) + \mathbf{diag}(v) \\ & w_0\|v\|_2 + \sum_{k=1}^{l} w_k \|U_k\|_F \leq \alpha \\ & U_k \succeq 0, \; k = 1, \ldots, l \\ & v_i \geq 0, \; i = 1, \ldots, p. \end{aligned} \quad (4)$$

We note that the first two constraints in (4) can be equivalently written as $\|X\|_G \leq \alpha$ with G-norm defined in (3). As defined, this constraint restricts X to be implicitly within the sparsity pattern defined by the groups β_k.

Problem (4) is a computationally tractable approximation of

$$\begin{aligned} \underset{X}{\text{minimize}} \quad & -\text{logdet}(X) + \mathbf{tr}(\hat{C}X) \\ \text{subject to} \quad & X \succeq 0, \quad \|X\|_G \leq \alpha \end{aligned} \quad (5)$$

a natural group norm penalized version of the G-LASSO problem. Specifically, in (5), $\|X\|_G$ can be written in terms of smaller matrices $W_k \in \mathbb{S}^{|\beta_k|}$ and $z \in \mathbb{R}^p$. If additionally the sparsity pattern is *chordal* (*i.e.* if the intersection graph of the groups β_k is a tree) then the positive semidefinite (PSD) matrix constraint $X \succeq 0$ can be decomposed to several smaller matrix constraints, via the equivalence in the following theorem.

Theorem 2.1 *[Agler* et al.*, 1988; Griewank and Toint, 1984] ([Grone* et al.*, 1984] dual) If $X \in \mathbb{S}^p$ has chordal sparsity, corresponding to groups $\beta_1, \ldots, \beta_l$, then*

$$X \succeq 0 \iff X = \sum_{k=1}^{l} \mathcal{A}_{\beta_k}(U_k), \quad U_k \succeq 0, \; k = 1, \ldots l.$$

In this case $X \succeq 0$ can be decomposed into smaller matrices $U_k \in \mathbb{S}_+^{|\beta_k|}$ and a positive diagonal $v \in \mathbb{R}_+^p$ (where $\mathbb{S}_+^p$ and $\mathbb{R}_+^p$ are the PSD cone and nonnegative orthant, both of order p). Then (4) is equivalent to (5) if and only if at optimality, $W_k = U_k$ for all k, and $v = z$.

3 Optimization

The Frank-Wolfe algorithm has regained much attention in minimizing sparse problems [Jaggi, 2013], mimicking greedy approaches yet having guaranteed optimality for convex problems. The Frank-Wolfe algorithm for solving $\min_X \{f(X) : X \in \mathcal{D}\}$ is described in Alg. 1. Using step sizes $\eta^{[t]} = 2/(t+2)$, it is known that the iterates of algorithm 1 converge at the sublinear rate of $O(1/t)$ [Frank and Wolfe, 1956; Dunn and Harshbarger, 1978]. We apply this algorithm for problem (4).

Algorithm 1 One step of Frank-Wolfe algorithm

Input: $X^{[t]} \in \mathcal{D}$, t-th iteration, step size η
1: Compute gradient $\nabla f(X^{[t]})$
2: Compute forward step : $S = \arg\min_{S \in \mathcal{D}} \langle U, \nabla f(X^{[t]}) \rangle$
3: Update primal variable : $X^{[t+1]} = (1 - \eta^{[t]})X^{[t]} + \eta^{[t]} S$
Output: optimal $X^{[t+1]}$

Forward step At each iteration, the forward step consists of l parallelizable projections on cones $\mathcal{C}_1, \ldots, \mathcal{C}_l$. Specifically, at each forward step, we compute

$$U_j^* = \frac{\alpha}{w_j \|Z_j\|_F} Z_j, \quad U_k = 0, \ \forall k \neq j.$$

where index $j = \arg\max_k w_k^{-1} \|Z_k\|_F$ and $Z_j = \mathbf{proj}_{\mathcal{C}_j}(-\nabla f(X)_{\beta_j,\beta_j})$. Then $S = \sum_k w_k \mathcal{A}_{\beta_k,\beta_k}(U_k)$. The derivations are given in appendix A.

Gradient computation In general, to compute the gradient $\nabla(\log\det(X)) = X^{-1}$ requires matrix inversion, which negates the computational complexity gain by decomposing the PSD cone. However, if the groups β_k form a chordal pattern, fast inversion methods exist [Liu, 1992; Andersen *et al.*, 2013] which require at each step l inversions of matrices at most of order $|\beta_k|$.

Applying both techniques, Alg. 2 describes the procedure for one iteration to find the NG-LASSO estimator (4). The main computational bottleneck at each step is a sequence of EDs of the submatrices $\nabla f(X)_{\beta_k,\beta_k}$, both for inverting X and for projecting on the PSD cone. For both operations, the complexity is $O(|\beta_k|^3)$ per group. If $|\beta_k| < p/l$ excluding the diagonal group, then the total per-iteration complexity of the proposed optimization procedure has a per-iteration complexity of $O(p^3/l^2 + p)$, and much smaller than $O(p^3)$ for G-LASSO.

Algorithm 2 One step of Frank-Wolfe algorithm for (4)

Input: $X^{[t]} \in \mathcal{D}$, t-th iteration, step size $\eta := \frac{2}{t+2}$

1: Find $\nabla f(X) = X^{-1} + C$

2: Find the forward direction U^+:

$$Z_0 = \mathbf{proj}_{\mathbb{R}^p_+}(-\mathbf{diag}(\nabla f(X)))$$
$$Z_k = \mathbf{proj}_{\mathbb{S}^{|\beta_k|}_+}(-\nabla f(X)_{\beta_k,\beta_k})$$
$$j = \arg\max_k w_k^{-1} \|Z_k\|_F$$
$$U_j^+ = \tfrac{\alpha}{w_j \|Z_j\|_F} Z_j, \quad U_k^+ = 0, \ \forall k \neq j$$

3: Update $X^{[t+1]} = X^{[t]} + \frac{2}{t+2} U^+$

Output: $X^{[t+1]}$.

4 Numerical Simulations

Here we show two simulation results: one on a general group sparsity pattern, and the other on a specific chordal pattern (banded).

To pick α and ρ, we swept powers of two in $\{2^{-5}, \ldots, 2^5\}$ and picked the best performing ρ or α for each test. In cases where the best performing ρ or α was on the boundary, additional parameters were tested until this was no longer the case.

4.1 Baselines

As one baseline, we solve (1). However, since group structure also reveals matrix sparsity, for fair comparison we also solve (1) restricted to the sparsity pattern induced by the groups:

$$\begin{array}{ll} \min\limits_X & -\text{logdet}(X) + \mathbf{tr}(\hat{C}X) + \rho\|X\|_1 \\ \text{s.t.} & X \succeq 0 \\ & X \in \mathbf{B} := \{X \mid X_{ij} = 0 \text{ if } i,j \notin \beta_k \ \forall k\}, \end{array} \tag{6}$$

which we call restricted group LASSO (RG-LASSO). We solve these baselines using the Douglas-Rachford method [Lions and Mercier, 1979; Combettes and Pesquet, 2011] for minimizing the sum of m convex functions, [1] detailed exactly as in [Combettes and Pesquet, 2011], Alg 10.27, with

$$f_1(X) = -\text{logdet}(X) + \mathbf{tr}(\hat{C}X), \quad f_2(X) = \rho\|X\|_1$$

and f_3, f_4 as indicator functions for constraints

$$f_3(X) = \begin{cases} 0 & X \succeq 0 \\ \infty & \text{else} \end{cases}, \quad f_4(X) = \begin{cases} 0 & X \in \mathbf{B} \\ \infty & \text{else} \end{cases}.$$

The proximal operator [Moreau, 1962] for a convex function $f(X)$ is defined as

$$\mathbf{prox}_f(Z) = \arg\min_X f(X) + (1/2)\|X - Z\|_F^2$$

and is defined for all Z, even if Z is not in the domain of f. (This is especially useful for $f = \log\det$ and $Z \not\succeq 0$.) From optimality conditions, it can be shown that

$$\mathbf{prox}_{tf_1}(Z) := V\mathbf{diag}(q)V^T, \quad 2q_i = -d_i + \sqrt{d_i^2 + 4t}$$

where $V\mathbf{diag}(d)V^T$ is the eigenvalue decomposition of $t\hat{C} - Z$. Similarly, $\mathbf{prox}_{tf_2}$ is the well-known *shrinkage operator*, and $\mathbf{prox}_{tf_3}$, $\mathbf{prox}_{tf_4}$ are projections on their respective constraint spaces.

4.2 Random sparsity

For $X \in \mathbb{S}^p$, we randomly select l groups $\beta_k \subset \{1, \ldots p\}$ of size b, and assume that this is the known group structure. Additionally, we select $\lceil \sigma_G \cdot l \rceil$ "active" groups (for $0 < \sigma_G < 1$); the identity of these groups are not known in training. In this simulation, we investigate the sample size required to recover the active groups, comparing G-LASSO, RG-LASSO, and NG-LASSO. In general, the sparsity patterns here are not chordal.

To determine the sparsity pattern of X, *i.e.*the estimate of C^{-1}, we threshold so that

$$i,j \text{ is a nonzero index if } \frac{|X_{ij}|}{\max_{kl}|X_{kl}|} > \theta,$$

for $\theta \in [0,1]$. Figure 1 shows the receiver operating characteristic (ROC) curve for $p = 100$ variables and $n = 25$ samples, where θ is implicitly swept. It is clear that NG-LASSO outperforms both other estimators; however, it is also evident that restricting the sparsity already offers much improvement. In order to remain agnostic to the right choice of θ, we use the area under the ROC curve (AUC) as the primary metric of estimator quality. Figure 2 plots the AUC for varying sample sizes n. Again, it is clear that for most cases, NG-LASSO outperforms RG-LASSO and G-LASSO. The exception is at very small values of n, where the number of observations is too low.

[1] This ADMM-type of method has the advantage over the projected gradient method because it has no step size restrictions based on the objective function's Lipschitz constant, which for $\log\det$ is unbounded.

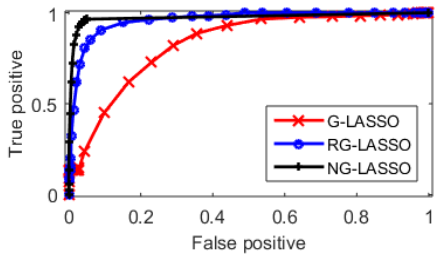

Figure 1: ROC curve for random block patterns where $p = 100$, $n = 25$. For each estimator, α, ρ picked to maximize AUC (area under this curve).

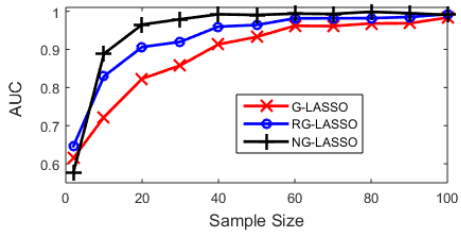

Figure 2: AUC for growing sample sizes (n) for random block patterns, averaged over 20 trials. $p = 100$, $l = 100$, $b = 5$ and $\sigma_G = 0.1$.

4.3 Banded sparsity

Next, we present a more extensive experiment with a chordal sparsity pattern; specifically, a banded pattern. For $X \in \mathbb{S}^p$, we assume that the true sparsity pattern consists of a nonzero diagonal and some active diagonal blocks of size b, where b is known but the true sparsity pattern is not. This gives in total $l = p - b + 1$ candidate groups $\beta_k = \{k, \cdots, k + b\}$ for $k = 1, \cdots, l$. Among l groups, we assume $\sigma_G l$ groups are active (where $0 < \sigma_G < 1$). Moreover, we simulate in-group sparsity; that is, for $0 < \sigma_I \leq 1$, we fix $\Pr(X_{ij} \neq 0 | i, j \in \beta_k) = \sigma_I$. Note that using only known information, we must assume the sparsity pattern is banded with bandwidth b.

We construct an invertible C^{-1} with the true sparsity pattern, and form a sample covariance matrix $\hat{C}$ by sampling from a multivariate Gaussian with 0 mean and covariance C. The goal is to use the estimators to correctly recover the sparsity pattern of C using $\hat{C}$ where the number of observations n is as small as possible.

Figure 3 shows a small example when $p = 100, n = 50$ and $\sigma_I = 0.25$. There are in total 90 groups in the banded sparsity pattern, where the 9 active groups (true sparsity) are in blue. We pick the estimator nonzeros by thresholding on the absolute value, choosing the threshold to, in each case, maximize $\min\{\#\text{ true positives}, \#\text{ true negatives}\}$.

It is clear that, for this small example, G-LASSO (left) yields many spurious nonzeros. By simply restricting the sparsity pattern to $\mathbf{B}$, the performance of RG-LASSO (center) already improves significantly, but NG-LASSO is still the best, since it accounts for sparsity in group selection as well.

Table 1 gives the result of a more extensive experiment, where the threshold, α, and ρ are picked to maximize the AUC, which is given for several p, n and σ_I. Here, we see that NG-LASSO is comparable with RG-LASSO when $p \approx n$, but is consistently better for $p \gg n$; both, however, are considerably improved over G-LASSO.

p	n	σ_I	$\hat{C}$	$\hat{C}^\dagger$	G	RG	NG
100	10	0.1	0.43	0.44	0.50	0.57	**0.65**
100	10	0.25	0.39	0.40	0.48	0.58	**0.64**
100	100	0.1	0.59	0.60	**0.89**	0.88	0.88
100	100	0.25	0.52	0.69	0.83	0.80	**0.85**
1000	10	0.1	0.41	0.41	0.49	0.52	**0.63**
1000	10	0.25	0.34	0.34	0.40	0.38	**0.68**
1000	100	0.1	0.55	0.56	0.56	0.58	**0.89**
1000	100	0.25	0.48	0.49	0.48	0.52	**0.90**

Table 1: Best AUC scores for $p \times p$ matrices with n samples, bandwidth $= 10$, and block sparsity σ_I. G = G-LASSO. RG = RG-LASSO. NG = NG-LASSO. Bolded are the best estimators. $\hat{C}, \hat{C}^\dagger$ = AUC score using sampled $\hat{C}, \hat{C}^\dagger$ directly.

		Per Iteration			Overall		
p	n	G	RG	NG	G	RG	NG
100	10	<0.1	<0.1	<0.1	**0.94**	3.93	7.3
100	100	<0.1	<0.1	<0.1	0.19	**0.17**	0.39
1000	10	1.6	1.6	**0.35**	**93.6**	108.69	351.1
1000	100	1.4	1.4	**0.35**	**13.8**	9.97	349.5
2500	100	21.3	19.6	**1.1**	852.9	556.7	**333.9**
5000	100	150.9	125.8	**1.7**	-	-	-

Table 2: Runtimes (in seconds) for $p \times p$ matrices with n samples, bandwidth $p/10$, and block sparsity $\sigma_I = 0.1$. G = G-LASSO. RG = RG-LASSO. NG = NG-LASSO. For $p \leq 100$, ρ and α are the same as those used in Table 1. For $p = 2500$, we just run $\rho = 0.125$ and $\alpha = 32$, which was observed to work well for smaller p. For $p = 5000$, we only give runtimes for 1 iteration, to illustrate the growing gap in per-iteration runtime.

Table 2 gives the per-iteration and total runtime of the three methods. In all cases, the per-iteration runtime depends only on p and b, and for larger p, NG-LASSO enjoys a much smaller per-iteration runtime. Of course, the total runtime (influenced also by the number of iterations) is also important; however, our experiments suggest this value is much less predictable. Several trends do emerge, though; for all methods, the runtime grows significantly when n is very small or σ_I is very low, suggesting that problem conditioning influences convergence rate as well. However, the key takeaway is that for very large p it is impossible to maintain full EDs for each iteration, and some decomposition must be used.

5 Financial application

In this section, we examine the performance of G-LASSO, RG-LASSO, and NG-LASSO in estimating the inverse covariance matrix for 14910 stocks, over a period of 100 or fewer days. This problem arises in finance in *Markowitz portfolio optimization* [Markowitz, 1952], which discusses optimal portfolio allocations using only mean and covariance information. Specifically, when the objective is to minimize return volatility, it is advised to invest in each stock a weight w_i of one's assets, where w minimizes the following quadratic optimization

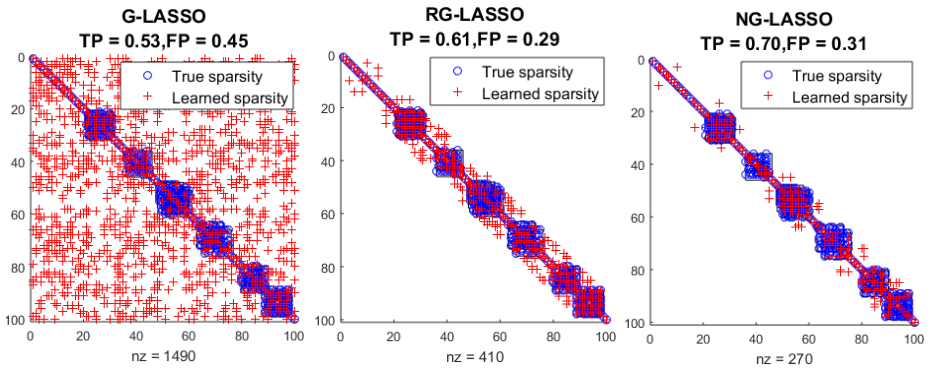

Figure 3: Banded pattern sparse inverse covariance estimation for $p = 100, n = 50$. From left to right are G-LASSO (1), RG-LASSO (6) and NG-LASSO (5). TP = true positive, FP = false positive.

problem

$$\min_w \quad w^T C w \qquad \text{s.t.} \quad \textstyle\sum_i w_i = 1$$

with optimal solution $w = (\mathbf{1}^T C^{-1}\mathbf{1})^{-1} C^{-1}\mathbf{1}$. However, in practice it is very difficult to apply this principle to large markets, especially when the amount of available historical data is limited, as the exact problem that C^{-1} is hard to estimate when p (thousands of stocks) is much greater than n (at most a few hundred days' opening or closing prices); using Markowitz' method in this regime can consistently underperform much simpler methods [DeMiguel *et al.*, 2009]. Thus, there is great interest in applying sparse estimators for a more generalizable calculation.

We attempt to estimate the inverse covariance matrix for stocks obtained from Yahoo! Finance, using daily closing prices as features. Details on the data scraping are given in appendix B. Define u_i as the full length observation vector for stock i, and S_i as the set of indices of u_i where that stock price observation is available. Define $V = \{1, 2, \ldots, 100\}$, $T = \{101, 102, \ldots, 200\}$, and $R = \{201, 202, \ldots, 200 + n\}$ as the indices of a validation, test, and train set respectively, and for all stocks i, $V_i = V \cap S_i, T_i = T \cap S_i, R_i = R \cap S_i$. [2] The sample covariance is then calculated as

$$\hat{C}_{ij} = \frac{1}{|R_i \cap R_j|} \sum_{k \in R_i \cap R_j} u_i[k] u_j[k].$$

We solve (1), (6), and (4) sweeping ρ, α for powers of 2 from 2^{-5} to 2^5, using cross validation to pick the best ρ and γ. [3]

Unlike the controlled simulation, we have no real understanding of the ground truth, so we test the estimator's generalizability, by measuring the maximum likelihood over the test samples. Specifically, we compute the negative log likelihood (NLL) for precision matrix X and samples $\{u_i\}_{i \in T}$ in the test set:

$$\text{NLL} = -\log\det(X) + \frac{X_{ij}}{|T_i \cap T_j|} \sum_{k \in T_i \cap T_j} u_i[k] u_j[k].$$

[2] Not every day's value is provided for every stock.

[3] As before, if the best ρ or α is on the boundary, additional powers of 2 are computed until this is no longer the case.

			sectors		industry	
p	n	G	RG	NG	RG	NG
100	10	**2.2e2**	2.5e2	7.0e2	2.7e2	5.5e2
500	10	2.0e3	1.8e3	3.7e3	**1.3e3**	3.5e3
500	100	1.3e3	**9.1e2**	4.1e3	9.3e2	3.1e3
1000	10	**2.5e3**	2.6e3	7.9e3	2.8e3	6.1e3
1000	100	1.1e13	**4.4e3**	7.1e3	**4.4e3**	7.4e3
2500	100	-	**1.1e4**	3.9e4	**1.1e4**	1.7e4
5000	100	-	1.5e12	1.1e7	5.9e10	**7.9e4**
14910	100	-	-	8.6e12	-	**5.2e5**

Table 3: Best test negative log likelihood for different methods, varying the number of stocks (p) and observations (n). G = G-LASSO. RG = RG-LASSO. NG = NG-LASSO. '-' = runtime was too long.

In this case, the smaller the value, the better.

Table 3 gives the test NLL for various p and n. As p grows (and especially as p/n grows) it is clear that G-LASSO has poorer performance, though the behavior is not as consistent as in table 1. Additionally, it seems that restricting the sparsity pattern gives the regularization needed to achieve very good performance, though we note that for very large p, NG-LASSO still does better.

Table 4 gives the runtime of each experiment for the best set of parameters, showing 1) the time to do one set of EDs ($p \times p$ for LASSO and sum of $|\beta_k| \times |\beta_k|, k = 1, \ldots l$ for group methods), 2) the average per-iteration runtime (≈ 1 ED) and 3) the total runtime. Since in this application all groups are nonoverlapping (all stocks are assigned a single sector and industry) computing RG-LASSO can be done in a fully decomposable manner, and can run at the same per-iteration speed as NG-LASSO. However, because the Douglas-Rachford method requires keeping 9 copies of the variables as separate iterates, it runs out of memory at $p = 14910$. Therefore our method still maintains the scalability advantage.

6 Conclusion

We present an inverse covariance estimator that regularizes for overlapping group sparsity, and provide better estimates when $p \gg n$. This is a challenging problem, common in finance, cli-

p	500	1000	2500	5000	14910
$p \times p$ ED	< 0.1	3.4e-1	5.2	4.3e1	1.1e3
S-ED	< 0.1	< 0.1	1.4e-1	6.4e-1	1.5e1
I-ED	< 0.1	< 0.1	< 0.1	< 0.1	1.0
G 1 it.	0.381	1.33	20.5	150.5	-
RG (S) 1 it.	< 0.1	0.20	1.6	6.0	-
RG (I) 1 it.	< 0.1	**0.17**	0.74	2.3	-
NG (S) 1 it.	< 0.1	0.20	1.4	5.5	79.6
NG (I) 1 it.	< 0.1	0.18	**0.72**	**2.2**	**20.2**
G all	20.0	1324.2	-	-	-
RG (S) all	3.0	88.9	713.6	**34.7**	-
RG (I) all	**2.5**	74.0	341.5	2351.1	-
NG (S) all	**2.5**	29.8	169.1	560.7	80377.2
NG (I) all	4.6	**26.9**	**79.7**	677.2	**2149.5**

Table 4: Runtimes (in seconds) of various algorithms for different matrix sizes ($\hat{C}$ is $p \times p$). S = sectors. I = industries. S-ED (I-ED) = time to compute $p_i \times p_i$ ED where $p_1, \ldots, p_l$ are the sizes of the l sector (industry) groups. G = G-LASSO. RG = RG-LASSO. NG = NG-LASSO. '-' = runtime was too long.

mate research, bioinformatics *etc.*. The techniques we present here incorporate known group structure in a principled way, and practically such that the group structure itself can be used to incorporate decomposition. A natural extension, and an avenue for future work, is to apply it to applications where group structure is not known, but can be ascertained through unsupervised methods, like K-means, spectral clustering, *etc.*.

Acknowledgements

This research was supported in part by NSF grant IIS-1319749.

Appendix

A Forward step derivation

The following is the derivation for Frank-Wolfe forward step in solving (4), which can be applied to more general problems. We first reformulate (4) into a generalized vector optimization problem:

$$\min_x \{f(x) : x \in \mathcal{D}\} \tag{7}$$

where

$$\mathcal{D} := \{x = \sum_{k=1}^{l} w_k A_{\gamma_k} u_k, \sum_{k=1}^{l} w_k \|u_k\|_2 \leq \alpha, u_k \in \mathcal{C}_k\}. \tag{8}$$

Here $f(x)$ can be any differentiable convex function, and $\mathcal{C}_1, \ldots, \mathcal{C}_l$ are proper convex cones. The index $\gamma_1, \ldots, \gamma_l$ define the groups. The vector variables are $u_k \in \mathbb{R}^{|\gamma|}$, As before, the parameters $w_1, \ldots, w_l > 0$ are weights. Problem (4) is a special case of (7). To match β_k with the sets γ_k, the equivalence is such that $\mathbf{vec}(A_{\beta_k,\beta_k}) = \mathbf{vec}(A)_{\gamma_k}$. The forward step to (7) is then

$$\begin{array}{ll} \underset{u_k}{\text{minimize}} & \langle \nabla f(x), \sum_{k=1}^{l} A_{\gamma_k} u_k \rangle \\ \text{subject to} & \sum_{k=1}^{l} w_k \|u_k\|_2 \leq \alpha \\ & u_k \in \mathcal{C}_k. \end{array} \tag{9}$$

For notational convenience, define $c_k = \mathbf{vec}(\nabla f(X)_{\gamma_k})$ for $k = 1, \ldots, l$. Then we can rewrite the objective function in (9) as

$$\underset{u_k}{\text{maximize}} \sum_k (-c_k)^T u_k$$

with constraints unchanged. From Moreau's decomposition, any vector a can be written as the sum of its projection on a closed convex cone $\mathcal{C}$ and its polar cone $\mathcal{C}^\circ$, of which are orthogonal. If we then expand

$$c_k^T u_k = \mathbf{proj}_{\mathcal{C}_k}(c_k)^T u_k + \mathbf{proj}_{\mathcal{C}_k^\circ}(c_k)^T u_k$$

then since feasible $u_k \in \mathcal{C}_k$, by definition of polar cone $\mathbf{proj}_{\mathcal{C}_k^\circ}(c_k)^T u_k \leq 0$, and $= 0$ only if $u_k = s_k \mathbf{proj}_{\mathcal{C}_k}(c_k)$. This is the optimal choice of direction for u_k, since it also maximizes the first term $\mathbf{proj}_{\mathcal{C}_k}(c_k)^T u_k$, and does not affect the norm constraint. If we additionally define scalars

$$a_k = \|P_{\mathcal{C}_k}(-c_k)\|_2^2, \quad b_k = w_k \|P_{\mathcal{C}_k}(-c_k)\|_2$$

then an even simpler equivalent formulation is

$$\begin{array}{ll} \underset{s_k}{\text{maximize}} & a^T s \\ \text{subject to} & b^T s \leq \alpha, \quad s \geq 0 \end{array}$$

which is a linear program with a known optimal solution of

$$s_i = \begin{cases} \alpha / b_i & \text{if } i = \underset{i}{\text{argmax}}\ a_i / b_i \\ 0 & \text{else.} \end{cases}$$

Substituting gives the closed form solution in the text.

B Yahoo! Finance data scraping details

Using the Yahoo! ticker downloader [4] we downloaded 27684 tickers for different stocks. We then used the Yahoo! finance API [5] to gather daily open, high, low, close, volume, and adjusted closing prices. We chose to monitor daily closing prices. We define groups as industries or sectors, as described in `https://biz.yahoo.com/p/`.

Any stock that we could not identify with an industry and sector was removed. We then prune the data to make sure it is as dense as possible. This resulted in 14,910 stocks and 1,005 daily closing prices. In total there are 9 sectors and 214 industries. with an average sector size 1656.7 and industry size 69.3. All stock vectors were then demeaned.

References

[Agler *et al.*, 1988] Jim Agler, William Helton, Scott McCullough, and Leiba Rodman. Positive semidefinite matrices with a given sparsity pattern. *Linear algebra and its applications*, 107:101–149, 1988.

[4] `https://pypi.python.org/pypi/Yahoo-ticker-downloader`

[5] `http://chart.finance.yahoo.com/table.csv?s=TICKERNAMEHERE&a=400&b=23&c=2016&d=0&e=23&f=2017&g=d&ignore=.csv`

[Andersen *et al.*, 2013] Martin S. Andersen, Joachim Dahl, and Lieven Vandenberghe. Logarithmic barriers for sparse matrix cones. *Optimization Methods and Software*, 28(3):396–423, 2013.

[Banerjee *et al.*, 2006] Onureena Banerjee, Laurent El Ghaoui, Alexandre d'Aspremont, and Georges Natsoulis. Convex optimization techniques for fitting sparse gaussian graphical models. In *Proc. of the 23rd ICML*, pages 89–96. ACM, 2006.

[Banerjee *et al.*, 2008] Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *JMLR*, 9(Mar):485–516, 2008.

[Chandrasekaran *et al.*, 2012] Venkat Chandrasekaran, Benjamin Recht, Pablo A. Parrilo, and Alan S. Willsky. The convex geometry of linear inverse problems. *Foundations of Computational mathematics*, 12(6):805–849, 2012.

[Combettes and Pesquet, 2011] Patrick L. Combettes and Jean-Christophe Pesquet. Proximal splitting methods in signal processing. In *Fixed-point algorithms for inverse problems in science and engineering*, pages 185–212. Springer, 2011.

[Dahl *et al.*, 2008] Joachim Dahl, Lieven Vandenberghe, and Vwani Roychowdhury. Covariance selection for nonchordal graphs via chordal embedding. *Optimization Methods & Software*, 23(4):501–520, 2008.

[d'Aspremont *et al.*, 2008] Alexandre d'Aspremont, Onureena Banerjee, and Laurent El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66, 2008.

[DeMiguel *et al.*, 2009] Victor DeMiguel, Lorenzo Garlappi, and Raman Uppal. Optimal versus naive diversification: How inefficient is the 1/n portfolio strategy? *Review of Financial Studies*, 22(5):1915–1953, 2009.

[Dunn and Harshbarger, 1978] Joseph C Dunn and S Harshbarger. Conditional gradient algorithms with open loop step size rules. *Journal of Mathematical Analysis and Applications*, 62(2):432–444, 1978.

[Frank and Wolfe, 1956] Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

[Friedman *et al.*, 2008] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics*, 9(3):432–441, 2008.

[Griewank and Toint, 1984] Andreas Griewank and Philippe L Toint. On the existence of convex decompositions of partially separable functions. *Mathematical Programming*, 28(1):25–49, 1984.

[Grone *et al.*, 1984] Robert Grone, Charles R. Johnson, Eduardo M. Sá, and Henry Wolkowicz. Positive definite completions of partial hermitian matrices. *Linear algebra and its applications*, 58:109–124, 1984.

[Hosseini and Lee, 2016] Mohammad Javad Hosseini and Su-In Lee. Learning sparse gaussian graphical models with overlapping blocks. In *Advances in Neural Information Processing Systems*, pages 3801–3809, 2016.

[Hsieh *et al.*, 2013] Cho-Jui Hsieh, Mátyás A Sustik, Inderjit S Dhillon, Pradeep K Ravikumar, and Russell Poldrack. BIG & QUIC: Sparse inverse covariance estimation for a million variables. In *Advances in Neural Information Processing Systems*, pages 3165–3173, 2013.

[Jaggi, 2013] Martin Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. In *ICML (1)*, pages 427–435, 2013.

[Lions and Mercier, 1979] Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[Liu, 1992] Joseph W.H. Liu. The multifrontal method for sparse matrix solution: Theory and practice. *SIAM review*, 34(1):82–109, 1992.

[Markowitz, 1952] Harry Markowitz. Portfolio selection. *The journal of finance*, 7(1):77–91, 1952.

[Mazumder and Hastie, 2012] Rahul Mazumder and Trevor Hastie. Exact covariance thresholding into connected components for large-scale graphical lasso. *JMLR*, 13(Mar):781–794, 2012.

[Meinshausen and Bühlmann, 2006] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the LASSO. *The annals of statistics*, pages 1436–1462, 2006.

[Moreau, 1962] Jean-Jacques Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *CR Acad. Sci. Paris Sér. A Math*, 255:2897–2899, 1962.

[Negahban *et al.*, 2009] Sahand Negahban, Bin Yu, Martin J. Wainwright, and Pradeep K. Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

[Obozinski *et al.*, 2011] Guillaume Obozinski, Laurent Jacob, and Jean-Philippe Vert. Group LASSO with overlaps: the latent group LASSO approach. *arXiv preprint arXiv:1110.0413*, 2011.

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

[Scheinberg and Rish, 2009] Katya Scheinberg and Irina Rish. Sinco-a greedy coordinate ascent method for sparse inverse covariance selection problem. *preprint*, 2009.

[Yuan and Lin, 2007] Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

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