
Learning Gaussian Graphical Models with Ordered Weighted ℓ_1 Regularization

Cody Mazza-Anthony

McGill University

cody.mazza-anthony@mail.mcgill.ca

Bogdan Mazouze

McGill University, MILA

bogdan.mazouze@mail.mcgill.ca

Mark Coates

McGill University

mark.coates@mcgill.ca

Abstract

We address the task of identifying densely connected subsets of multivariate Gaussian random variables within a graphical model framework. We propose two novel estimators based on the Ordered Weighted ℓ_1 (OWL) norm: 1) The Graphical OWL (GOWL) is a penalized likelihood method that applies the OWL norm to the lower triangle components of the precision matrix. 2) The column-by-column Graphical OWL (ccGOWL) estimates the precision matrix by performing OWL regularized linear regressions. Both methods can simultaneously identify highly correlated groups of variables and control the sparsity in the resulting precision matrix. We formulate GOWL such that it solves a composite optimization problem and establish that the estimator has a unique global solution. In addition, we prove sufficient grouping conditions for each column of the ccGOWL precision matrix estimate. We propose proximal descent algorithms to find the optimum for both estimators. For synthetic data where group structure is present, the ccGOWL estimator requires significantly reduced computation and achieves similar or greater accuracy than state-of-the-art estimators. Timing comparisons are presented and demonstrates the superior computational efficiency of the ccGOWL. We illustrate the grouping performance of the ccGOWL method on a cancer gene expression data set and an equities data set.

1 Introduction

The task of estimating highly modular structures based on relationships found in the data frequently arises in computational biology and finance [22]. Due to the large volume and high dimensionality of data in these disciplines, inference procedures can be considerably sped up by leveraging the structured relations between inputs [24]. For example, it is often desirable to induce sparsity in an effort to reduce cross-pathway connections between genes when considering gene expression data.

Gaussian graphical models (GGMs) are well-suited for modeling a sparse graph through the non-zero pattern of the inverse covariance matrix. From a probabilistic point of view, the precision (inverse covariance) matrix directly encodes the conditional independence relations between its elements. The most well-known algorithm for learning GGMs is the *graphical lasso* [11], which imposes graph sparsity through maximization of an ℓ_1 -penalized Gaussian log-likelihood with block coordinate descent. Current state-of-the-art methods [13, 26, 7, 9] also rely on ℓ_1 penalized structural learning. However, estimating the precision matrix column-by-column has received much attention since these methods can be numerically simpler and more accommodating to theoretical analysis [3, 19].

Related work can be organized into three classes: 1) *Methods which learn a graph given groups a priori*: This class includes [9] which applies a group ℓ_1 penalty to encourage structured sparsity but requires a set of pre-defined hyperparameters to control the topology of the network. In [25, 17, 29, 4], prior knowledge is used to assign edge weights in order to predict underlying group structure. 2) *Methods which find similar groups and then learn the structure of each group*: This class includes [26] which proposes a two-step approach to the problem: first applying hierarchical clustering to identify groups and then using the graphical lasso within each group. Devijver et al. (2018) detect groups in the covariance matrix using thresholding and then apply graphical lasso to each group [8]. 3) *Methods which learn both group and graph structures simultaneously*: Defazio et al. (2012) propose to use a non-decreasing, concave function in the penalty to enforce the submodularity of the objective [7]. Hosseini et al. (2016) propose the GRAB estimator that solves a joint optimization problem which alternates between estimating overlapping groups encoded into a Laplacian prior and learning the precision matrix [13]. Kumar et al. (2019) propose a framework for structured graph learning that imposes Laplacian spectral constraints [16]. Tarzanagh et al. (2018) apply a structured norm to precision matrix estimation to identify overlapping groups [27].

This paper presents two novel estimators for identifying the precision matrices associated with Gaussian graphical models and makes four major contributions. First, both estimators require a small amount of *a priori* information and do not require any information about group structure. More specifically, they only require two hyperparameters (one for sparsity and one for grouping) when using the OSCAR-like [2] weight generation procedure. This differs from [9] which requires p^2 hyperparameters and GRAB which requires a more constrained objective accompanied by an additional hyperparameter for each constraint. Second, both estimators can learn overlapping groups and network structure in a single step proximal descent procedure. This is an advantage over the *cluster graphical lasso* [26], which solves the task with a two-step procedure by alternating between clustering and gradient steps. Likewise, the GRAB algorithm also alternates between learning the overlapping group prior matrix and learning the inverse covariance. Our focus is on estimating the graphical model, rather than identifying groups, but we can apply a Gaussian Mixture Model (GMM) to identify overlapping groups from our precision matrix estimate. Third, we establish the uniqueness of the GOWL estimator by deriving its dual formulation. Fourth, the ccGOWL framework provides new theoretical guarantees for grouping related entries in the precision matrix and offers a more computationally efficient algorithm. When comparing ccGOWL to the previously mentioned penalized likelihood methods, it is clear that the ccGOWL is more computationally attractive in the high dimensional setting as it can be obtained one column at a time by solving a simple linear regression that can be easily parallelized/distributed.

Notation and Preliminaries. Throughout the paper, we highlight vectors and matrices by lowercase and uppercase boldfaced letters, respectively. For a vector $\mathbf{a} \in \mathbb{R}^p$, let \mathbf{a}_{-i} denote a vector with its i^{th} component removed. For a symmetric matrix $\mathbf{A} \in \mathbb{R}^{p \times p}$, \mathbf{A}_{i*} represents the i^{th} row of \mathbf{A} and \mathbf{A}_{*j} denotes the j^{th} column of \mathbf{A} , \mathbf{A}_{i-j} denotes the i^{th} row of \mathbf{A} with its j^{th} entry removed, \mathbf{A}_{-ij} denotes the j^{th} column of \mathbf{A} with its i^{th} entry removed, the matrix \mathbf{A}_{-i-j} denotes a $(p-1) \times (p-1)$ matrix obtained by removing the i^{th} row and j^{th} column. Moreover, we denote as $\text{vechs}(\mathbf{A})$ the strict column-wise vectorization of the lower triangular component of \mathbf{A} :

$$\text{vechs}(\mathbf{A}) = [\mathbf{A}_{2,1}, \dots, \mathbf{A}_{n,1}, \mathbf{A}_{3,2}, \dots, \mathbf{A}_{n,2}, \mathbf{A}_{n,n-1}].$$

For a vector $\mathbf{a} = (a_1, \dots, a_p)^T \in \mathbb{R}^p$, we use the classical definition of the ℓ_q norm, that is $\|\mathbf{a}\|_q = (\sum_{i=1}^p |a_i|^q)^{1/q}$ for $1 \leq q \leq \infty$. The i^{th} largest component in magnitude of $\{|a_1|, |a_2|, \dots, |a_p|\}$ is denoted $a_{[i]}$. The vector obtained by sorting (in non-increasing order) the components of \mathbf{a} is denoted $\mathbf{a}_\downarrow = \{a_{[1]}, a_{[2]}, \dots, a_{[p]}\}$. For matrices and vectors, we define $|\cdot|$ to be the element-wise absolute value function.

2 Background

2.1 Gaussian Graphical Models (GGM)

A GGM aims to determine the conditional independence relations of a set of random jointly Gaussian variables. Suppose $\mathbf{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_n\} \sim N(\mathbf{0}, \Sigma)$ is a collection of i.i.d. p -dimensional random samples. Assume that the columns of the design matrix \mathbf{X} have been standardized to have zero mean and unit variance. Let S denote the (biased, maximum likelihood) sample covariance matrix,

defined as $S = n^{-1} \sum_{k=1}^n \mathbf{X}_k \mathbf{X}_k^T$ and let Θ denote the precision matrix, defined as $\Theta = \Sigma^{-1}$. It is well understood that the sparsity pattern of Θ encodes the pairwise partial correlation between variables. More specifically, the ij^{th} entry in Θ is zero if and only if variables i and j are conditionally independent given the remaining components [15, 30]. The GLASSO algorithm has been proposed to estimate the conditional independence graph embedded into Θ through ℓ_1 regularization of Gaussian maximum likelihood estimation:

$$\min_{\Theta \succ 0} -\log \det \Theta + \text{tr}(S\Theta) + \lambda \|\Theta\|_1, \quad (1)$$

where λ is a nonnegative tuning parameter that controls the level of sparsity in Θ . Under the same assumptions [19] showed that Θ can be estimated through column-by-column linear regressions. Letting $\mathbf{x}_i \sim N(\mathbf{0}, \Sigma)$ for $i = 1, \dots, n$, the conditional distribution of x_i given \mathbf{x}_{-i} satisfies:

$$x_j | \mathbf{x}_{-j} \sim N(\beta_j^T \mathbf{x}_{-j}, \sigma_j^2),$$

where $\beta = (\Sigma_{-j,-j})^{-1} \Sigma_{-j,j} \in \mathbb{R}^{p-1}$ and $\sigma_j^2 = \Sigma_{j,j} - \Sigma_{j,-j}(\Sigma_{-j,-j})\Sigma_{-j,j}$. Thus, we can write the following

$$x_j = \beta_j^T \mathbf{x}_{-j} + \varepsilon_j,$$

where $\varepsilon_j \sim N(0, \sigma_j^2)$. By the block matrix inversion formula we define:

$$\Theta_{jj} = \sigma_j^{-2}, \quad \text{and} \quad \Theta_{-j,j} = -\sigma_j^{-2} \beta_j.$$

Thus, we can estimate the precision matrix column-by-column by regressing x_j on \mathbf{x}_{-j} , and a LASSO procedure can be adopted by solving:

$$\hat{\beta}_j = \arg \min_{\beta_j \in \mathbb{R}^{p-1}} \|\mathbf{X}_{*,j} - \mathbf{X}_{*,-j} \beta_j\|_2^2 + \lambda \|\beta_j\|_1. \quad (2)$$

2.2 OWL in the Linear Model Setting

In the linear regression setting, much interest has been given to regularizers that can identify groups of highly correlated predictors. This is often referred to as *structured/group sparsity*. Regularization penalties targeting such structure include the *elastic net* [33], the *fused lasso* [28] and the recently proposed *order weighted ℓ_1 lasso* [1, 10]. Suppose that the response vector $\mathbf{y} \in \mathbb{R}^n$ is generated from a linear model of the form: $\mathbf{y} = \mathbf{X}\beta + \varepsilon$ where $\mathbf{X} \in \mathbb{R}^{n \times p}$ is the design matrix, $\beta \in \mathbb{R}^p$ is the vector of regression coefficients, $\mathbf{y} \in \mathbb{R}^n$ is the response vector, and $\varepsilon \in \mathbb{R}^n$ is a vector of zero-centered homoscedastic random errors distributed according to $\mathcal{N}(0, \sigma^2)$. The OWL regularizer is then defined to be:

$$\Omega_{\text{OWL}}(\beta) = \sum_{i=1}^p \lambda_i |\beta|_{[i]}, \quad (3)$$

where $|\beta|_{[i]}$ is the i^{th} largest component in magnitude of $\beta \in \mathbb{R}^p$, and $\lambda \in \mathbb{R}_+^p$. OSCAR [2] is a particular case of the OWL that consists of a combination of a ℓ_1 and a pairwise ℓ_∞ terms and in its original formulation is defined as

$$\lambda_1 \sum_{i=1}^p |\beta_i| + \lambda_2 \sum_{i < j} \max\{|\beta_i|, |\beta_j|\}. \quad (4)$$

The OSCAR weights can be recovered within OWL by setting $\lambda_i = \lambda_1 + \lambda_2(p - i)$ where $\lambda_1 > 0$ controls the level of sparsity and $\lambda_2 > 0$ controls the degree of grouping. In [1] the OWL regularizer is referred to as *sorted ℓ_1 penalized estimation* (SLOPE).

2.3 Proximal Methods

Proximal algorithms form a class of robust methods which can help solving composite optimization problems of the form

$$\min_{x \in \mathcal{X}} g(x) + h(x), \quad (5)$$

where \mathcal{X} is a Hilbert space with associated inner product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|$, $g : \mathcal{X} \mapsto \mathbb{R}$ is a closed, continuously differentiable, convex function, with an L -Lipschitz continuous gradient ∇g for $L > 0$. The function $h : \mathcal{X} \mapsto \mathbb{R}$ is a convex, not necessarily differentiable function. The *proximal mapping* of a convex function h , denoted by $\text{prox}_h : \mathcal{X} \rightarrow \mathcal{X}$, is given by

$$\text{prox}_h(v) = \arg \min_x \left(f(x) + \frac{1}{2} \|x - v\|_2^2 \right). \quad (6)$$

In general, this family of algorithms uses the proximal mapping to handle the non-smooth component of (5) and then performs a quadratic approximation of the smooth component g every iteration:

$$x_{k+1} = \text{prox}_h(x_k - t_k \nabla g(x_k)),$$

where t_k denotes the step size at iteration k . [23] proposed a proximal gradient method for precision matrix estimation which possesses attractive theoretical properties as well as a linear convergence rate under a suitable step size. The authors define g, h as two continuous convex functions mapping the domain of covariance matrices (the positive definite cone \mathbb{S}_{++}^p) onto \mathbb{R} and jointly optimize them using Graphical ISTA [23]. G-ISTA conducts a backtracking line search for determining an optimal step size or specifies a constant step size of $\lambda_{\min}(\Theta_k)$ for problems with a small p . This algorithm uses the duality gap as a stopping criteria and achieves a linear rate of convergence in $O(\log \varepsilon)$ iterations to reach a tolerance of ε .

3 Methodology

3.1 Overview of OWL Estimators

We define the Graphical Order Weighted ℓ_1 (GOWL) estimator to be the solution to the following constrained optimization problem:

$$\min_{\Theta \succ 0} -\log \det \Theta + \text{tr}(\mathbf{S}\Theta) + \Omega_{\text{OWL}}(\Theta), \quad (7)$$

where

$$\Omega_{\text{OWL}}(\Theta) = \lambda^T |\text{vechs}(\Theta)|_{\downarrow} = \sum_{i=1}^K \lambda_i |\text{vechs}(\Theta)|_{[i]}. \quad (8)$$

Here $\Theta \in \mathbb{R}^{p \times p}$, $\text{vechs}(\Theta)_{[i]}$ is the i^{th} largest off-diagonal component in magnitude of Θ , $K = (p^2 - p)/2$, and $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_p \geq 0$. The proximal mapping of Ω_{OWL} denoted by $\text{prox}_{\Omega_{\text{OWL}}} : \mathbb{R}^{p \times p} \mapsto \mathbb{R}^{p \times p}$ can be efficiently computed with $O(n \log n)$ complexity using the *pool adjacent violators* (PAV) algorithm for isotonic regression outlined in [31].

In a similar way, we define the column-by-column Graphical Order Weighted ℓ_1 (ccGOWL) estimator to be the solution to the following unconstrained optimization problem:

$$\hat{\beta}_j = \arg \min_{\beta_j \in \mathbb{R}^{p-1}} \|\mathbf{X}_{*,j} - \mathbf{X}_{*,-j} \beta_j\|_2^2 + \Omega_{\text{OWL}}(\beta_j), \quad (9)$$

where $\beta_j \in \mathbb{R}^{p-1}$ with hyperparameters defined the same way as above.

3.2 Uniqueness of GOWL

In this section, we derive a dual formulation of GOWL. These play an important role for implementing efficient algorithms and allow us to establish uniqueness of the solution of GOWL.

Definition 1. The duality gap Δ given a point in the feasible set defined by $B_\lambda = \{\mathbf{W} : |\mathbf{W}_i| \leq \lambda_i, \forall i, \mathbf{W} \in \mathbb{S}_{++}^p\}$ is

$$\Delta = \text{tr}(\mathbf{S}\Theta) + \sum_{i=1}^K \lambda_i |\Theta|_{[i]} - p. \quad (10)$$

where tr denotes the trace operator. Please see Supplementary Material for details of the derivation of the duality gap for the GOWL estimator. In practice, Δ is estimated using the difference between the primal and the dual and acts as a stopping criterion for the iterative procedure. As opposed to other sparse graph estimation techniques, our proposed objective function has a unique solution.

Theorem 1. If $\lambda_i > 0$ for all i , then problem (7) has a unique optimal point Θ^* .

The proof is provided in the Supplementary Material and involves selecting \mathbf{W} within $\text{interior}(B_\lambda) = \{\mathbf{W} : |\mathbf{W}_i| < \lambda_i, \forall i, \mathbf{W} \in \mathbb{R}^p\}$ such that Slater's condition (see Supplementary Material) is satisfied.

3.3 Algorithms

Our work borrows from G-ISTA discussed in Section 2.3 for solving the non-smooth problem defined in (7); the details are outlined in Algorithm 1. In a similar way, Algorithm 2 is based on the proximal method for solving the OWL regularized linear regression proposed in [31] and has a convergence rate of $O(1/k)$. Algorithm 2 applies a proximal method p times and combines each column to arrive at the final precision matrix estimate.

Algorithm 1 Algorithm for GOWL

Require: \mathbf{S} , tolerance ε , λ , $t_0 > 0$, Θ_0 , $c \in (0, 1)$

while $\Delta > \varepsilon$ **do**

(1) Let t_k be the largest element of $\{c^j t_{k,0}\}$ so that for $\Theta_{k+1} = \text{prox}_{\Omega_{\text{OWL}}}(\Theta_k - t_k(\mathbf{S} - (\Theta_k^{-1})))$, the following are satisfied:

$$\Theta_{k+1} \succ 0 \quad \text{and} \quad -\log \det \Theta_{k+1} + \text{tr}(\mathbf{S}\Theta_{k+1}) \leq Q_{t_k}(\Theta_{k+1}, \Theta_k),$$

where

$$Q_{t_k}(\Theta_{k+1}, \Theta_k) = -\log \det \Theta + \text{tr}(\mathbf{S}\Theta) + \text{tr}(\Theta_{k+1} - \Theta_k, \mathbf{S} - \Theta_k^{-1}) + \frac{1}{2t_k} \|\Theta_{k+1} - \Theta_k\|_F^2.$$

(2) Compute $\nabla g(\Theta_k) := \Theta_k - t_k(\mathbf{S} - \Theta_k^{-1})$.

(3) Set $\Theta_{k+1} := \text{prox}_{\text{OWL}}(\nabla g(\Theta_k))$.

(4) Compute the duality gap Δ .

end while

Ensure: ε -optimal solution $\hat{\Theta}$.

Algorithm 2 Algorithm for ccGOWL

Require: \mathbf{X} , λ , $t_0 > 0$

for $j \rightarrow p$ **do**

(1) Determine $t_k > 0$ such that the following is satisfied:

$$\|\mathbf{X}_{*,j} - \mathbf{X}_{*,j} \beta_j^{k+1}\|_2^2 \leq Q_{t_k}(\beta_j^k, \beta_j^{k+1}),$$

where

$$Q_{t_k}(\beta_j^k, \beta_j^{k+1}) = \|\mathbf{X}_{*,j} - \mathbf{X}_{*,j} \beta_j^{k+1}\|_2^2 + 2(\beta_j^k - \beta_j^{k+1})^T \mathbf{X}_{*,j}^T (\mathbf{X}_{*,j} - \mathbf{X}_{*,j} \beta_j^{k+1}) + \frac{t_k}{2} \|\beta_j^k - \beta_j^{k+1}\|_2^2.$$

(2) Compute $\nabla g(\beta_j^k) := n^{-1} \mathbf{X}_{*,j}^T (\mathbf{X}_{*,j} - \mathbf{X}_{*,j} \beta_j^k)$.

(3) Set $\beta_j^{k+1} \leftarrow \text{prox}_{\text{OWL}}(\beta_j^k - t_k \nabla g(\beta_j^k))$.

end for

Combine all $\hat{\beta}_j^{k+1}$ to form $\hat{\Theta}$.

3.4 Sufficient Grouping Conditions for ccGOWL

We can establish sufficient grouping conditions when estimating each column of Θ by drawing on previous work for the OSCAR and OWL regularizers. The term ‘‘grouping’’ refers to components of each column estimate being equal.

Theorem 2. Let $\hat{\beta}_k, \hat{\beta}_l > 0$ be elements in the column estimate $\hat{\beta}_j$ of $\hat{\Theta}_j$ generated with hyperparameter λ_2 , and let them be unique from other entries in $\hat{\beta}_j$. Then there exists a λ'_2 such that

$$0 < |\lambda'_2(p-1)| \leq 2\|\mathbf{X}_{*,j}\|_2^2 \sqrt{2(1-\rho_{kl})},$$

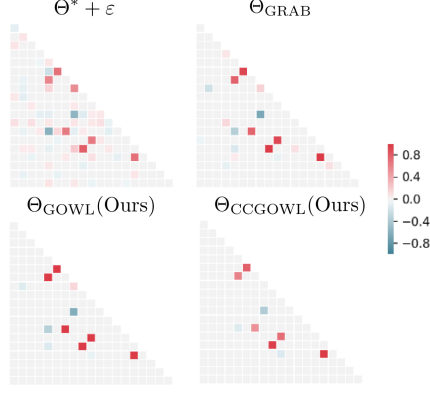


Figure 1: Synthetic data results example for methods ccGOWL/GOWL with $p = 20$ and $\kappa = 0.1$.

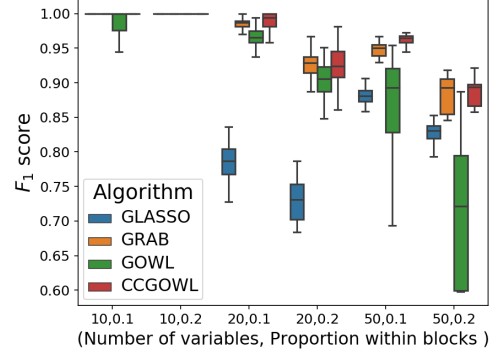


Figure 2: Quantiles of weighted F_1 score metric for GLASSO, GRAB, GOWL, CCGOWL.

so that for all

$$|\lambda_2(p-1)| > |\lambda'_2(p-1)|$$

we have

$$\hat{\beta}_k = \hat{\beta}_l$$

for $j = 1, \dots, (p-1)$ where $\rho_{kl} = \mathbf{x}_k^T \mathbf{x}_l$ is the sample correlation between columns of $\mathbf{X}_{*, -j}$.

Depending on the sample correlation between covariates in $\mathbf{X}_{*, -j} \in \mathbb{R}^{n \times (p-1)}$ the sufficient grouping property can be quantified according to Theorem 2.

4 Experimental Results

To assess the performance of our algorithm, we conducted a series of experiments: we first compared the estimates found by GOWL and ccGOWL to those from GRAB[13] and GLASSO[11] on a synthetic dataset with controlled sparsity and grouping. Then, we compared the average weighted F_1 scores obtained by all four algorithms on random positive definite grouped matrices of varying size and group density. All algorithms were implemented in Python and executed on an Intel i7-8700k 3.20 GHz and 32 GB of RAM.

4.1 Synthetic Data

Synthetic data was generated by first creating a proportion of groups $\kappa \in \{0.1, 0.2\}$ for a p -dimensional matrix. We randomly chose each group size to be between a minimum size of 0.1 of p and a maximum size of 0.4 of p . Furthermore, group values were determined by uniformly sampling their mean between $(0.9, 1.0)$ and $(-0.9, -1.0)$ respectively. After setting all values of a given group to its mean, we collect the groups into the matrix Θ^* . In order to add noise to the true group values, we randomly generated a positive semi-definite matrix with entries set to zero with a fixed probability of 0.5 and remaining values sampled between $(-0.1, 0.1)$. We then added the grouped matrix to the aforementioned noise matrix to create a $\Theta^* + \epsilon$ matrix. Each grouped matrix was generated 5 times and random noise matrices were added to each of the 5 grouped matrices 20 times. A dataset was then generated by drawing i.i.d from $\mathcal{N}_p(0, (\Theta^* + \epsilon)^{-1})$ distribution, from which the empirical covariance matrix can be estimated after standardization of covariates. Hyperparameters λ_1 and λ_2 using the OSCAR weight generation defined in Section 3.3 were selected using a 2-fold standard cross-validation procedure. Similar to [13], the regularization parameter for the GRAB estimator was also selected using a 2-fold cross-validation procedure (Supplementary Material Table 1, 2, 3).

Figure 1 shows an example of a synthetic precision matrix with 2 groups with Θ^* as the ground truth. This ground truth was then used to sample a dataset of $n = 100$, from which we estimated the

Table 1: Timing Comparisons in Seconds

Method	$p = 10$	$p = 20$	$p = 50$	$p = 100$	$p = 500$	$p = 1000$
GLASSO	0.006	0.012	0.017	0.023	0.118	0.456
GRAB	0.071	0.096	0.466	1.243	51.225	499.225
ccGOWL	0.003	0.012	0.034	0.095	0.305	4.126

empirical covariance matrix and provided it as input to the algorithms. The GOWL and ccGOWL precision matrix estimates almost fully recover the 6 red entries from group 1 and the two blue entries from group two. The GLASSO was not included as a result of its poor performance in recovering the true group values. We assessed the performance of all three methods using the weighted F_1 classification score (harmonic mean of precision and recall) since we are interested in multi-group classification. The F_1 score is defined as $F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$; the weighted F_1 score is obtained by calculating the F_1 score for each label and then evaluating the weighted average, where the weights are proportional to the number of true instances of each group. For each value of $(p, \kappa) \in \{10, 20, 50\} \times \{0.1, 0.2\}$, we generated 5 randomly grouped matrices using the procedure outlined in the previous section and fit GLASSO, GRAB, GOWL, ccGOWL to each of them. The estimates were then clustered using a Gaussian Mixture Model (GMM) with the same number of groups as originally set. The goodness-of-fit of the clusters was compared to the original group labeling using the weighted F_1 metric, from which we report the permutation of labels giving the highest weighted F_1 score. Figure 2 shows the distribution of the scores for each algorithm, for each class of matrices. Overall, the ccGOWL outperforms GOWL and GLASSO in terms of variance and mean of F_1 scores. However, ccGOWL achieves similar or greater performance when compared to GRAB. Mean squared error and absolute error were also measured for each estimate and are provided in the supplementary material (Figure 1 and 2).

4.2 Timing Comparisons

The GLASSO, GRAB and ccGOWL algorithms were run on synthetic datasets generated in the same way as in Section 4.1 with varying p , $n = 100$, $\kappa = 0.2$ and different levels of regularization. The GRAB algorithm had a fixed duality gap of 10^{-4} and the ccGOWL had a fixed precision of 10^{-5} . The GRAB algorithm was implemented in python and utilizes the R package QUIC implemented in C++ [14] to find the optimum of the log-likelihood function. The ccGOWL algorithm is implemented completely in python and requires solving p linear regressions with most of the computational complexity attributed to evaluating the proximal mapping. The GLASSO algorithm utilized the python package sklearn [21]. Running times are presented in Table 1 and show a significant advantage of ccGOWL over GRAB. This difference for large p is due to GRAB requiring matrix inversions within QUIC ($O(p^2)$) and applying the k-means clustering algorithm ($O(pK)$) on the rows of the block matrix \mathbf{Z} .

4.3 Cancer Gene Expression Data

We consider a dataset that uses expression monitoring of genes using DNA microarrays in 38 patients that have been diagnosed with either acute myeloid leukemia (AML) or acute lymphoblastic leukemia (ALL). This dataset was initially investigated in [12]. In order to allow for a model that is easier to interpret we selected the 50 most highly correlated genes associated with ALL-AML diseases, as identified in [12]. Expression levels were standardized to have zero and unit variance. Of the 50 genes selected, the first 25 genes are known to be highly expressed in ALL and the remaining 25 genes are known to be highly expressed in AML. It is important to note that no single gene is consistently expressed across both AML and ALL patients. This fact illustrates the need for an estimation method that takes into account multiple genes when diagnosing patients. Genes that are highly expressed in AML should appear in the same group and likewise for the ALL disease. In addition, we utilize the Reactome [5] to identify each gene’s main biological pathway.

Figure 3 illustrates that ccGOWL groups genes according to their disease. In fact, ccGOWL correctly identifies all 25 genes associated with the AML disease in one group and 24 of 25 genes associated with the ALL disease in the other group (Supplementary Material Table 4). We compare the network identified by ccGOWL with the commonly-used baseline GLASSO method, to illustrate

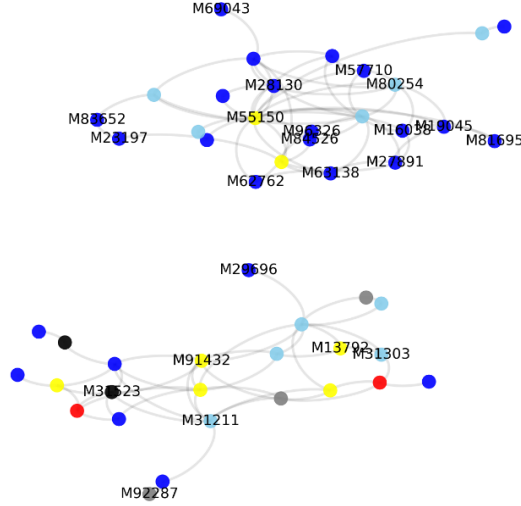


Figure 3: Network constructed by ccGOWL on gene expression data. The ccGOWL network estimate clearly groups genes associated with AML (top) and ALL (bottom). Each color represents a biological pathway: Signal Transduction (red), Immune System (blue), Cell Cycle (gray), Metabolism (yellow), Gene Expression (black), Uncategorized (skyblue).

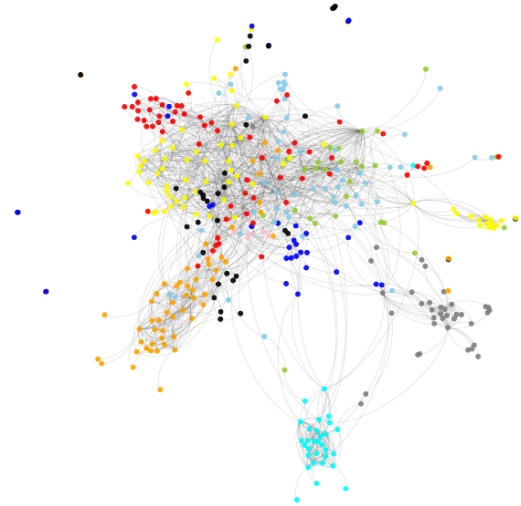


Figure 4: Network constructed by ccGOWL on equities expression data. Each colour represents a GICS sector: Consumer Discretionary (red), Consumer Staples (blue), Energy (gray), Financials (yellow), Health Care (black), Industrials (sky-blue), Information Technology (orange), Materials (yellow-green), Telecommunications Services (pink), Utilities (cyan).

the importance of employing an estimator that uses grouping (Supplementary Material Figure 3). In addition, examining the connections within each group can also lead to useful insights. The AML group (top) contains highly connected genes beginning with “M” (blue nodes) which are associated with the neutrophil granulation process in the cell. AML is a disorder in the production of neutrophils [6]. Neutrophils are normal white blood cells with granules inside the cell that fight infections. AML leads to the production of immature neutrophils (referred to as blasts), leading to large infections.

4.4 Equities Data

We consider the stock price dataset described in [18], which is available in the huge package on CRAN [32]. The dataset consists of the closing prices of stocks in the S&P 500 between January 1, 2003 and January 1, 2008. The collection of stocks can be categorized into 10 Global Industry Classification Standard (GICS) sectors [20]. Stocks that were not consistently in S&P 500 index or were missing too many closing prices were removed from the dataset.

The design matrix $\mathbf{X} \in \mathbb{R}^{1257 \times 452}$ contains the log-ratio of the price at time t to the price at time $t - 1$ for each of the 452 stocks for 1257 trading days. More formally we write that the (i, j) -th entry of \mathbf{X} is defined as $x_{ij} = \log(S_{(i+1)j}/S_{ij})$ where S_{ij} is the closing price of the j th stock on the i th day. The matrix \mathbf{X} was then standardized so each stock has a mean of zero and unit variance. The GICS sector for each stock is known, however, this information was not used when estimating the precision matrix based on the return matrix \mathbf{X} . Figure 4 illustrates the ability of ccGOWL to group stocks that belong to the same sector and is more interpretable than the network constructed by GLASSO (Supplementary Material Figure 4). Information Technology and Utilities largely exhibit conditional independence of stocks in other GICS sectors. On the other hand, there appears to be a conditional dependence between stocks in the Materials and Industrials sectors, probably as a result of multiple customer-supplier relationships between the companies in these sectors. Both sectors are sensitive to economic cycles and the equities offer exposure to global infrastructure replacement.

5 Conclusion

We proposed the GOWL and ccGOWL estimators, two novel precision matrix estimation procedures based on the Order Weighted ℓ_1 norm. We proved the uniqueness of the GOWL estimator and identified sufficient grouping conditions for the ccGOWL estimator. Based on empirical results on synthetic and real datasets, the ccGOWL estimator has the ability to accurately identify structure in the precision matrix in a much more computationally efficient manner than state-of-the-art estimators that achieve comparable accuracy. Although penalized likelihood methods for learning structure in precision matrix estimation are widely used and perform well, it is also important to consider column-by-column estimators that accomplish similar feats while reducing computational complexity.

References

- [1] M. Bogdan, E. Van Den Berg, C. Sabatti, W. Su, and E. J. Candès. SLOPE—adaptive variable selection via convex optimization. *The Annals of Applied Statistics*, 9(3):1103, 2015.
- [2] H. D. Bondell and B. J. Reich. Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics*, 64(1):115–123, 2008.
- [3] T. Cai, W. Liu, and X. Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.
- [4] T. T. Cai, H. Li, W. Liu, and J. Xie. Joint estimation of multiple high-dimensional precision matrices. *Statistica Sinica*, 26(2):445, 2016.
- [5] D. Croft, A. F. Mundo, R. Haw, M. Milacic, J. Weiser, G. Wu, M. Caudy, P. Garapati, M. Gillespie, M. R. Kamdar, et al. The reactome pathway knowledgebase. *Nucleic Acids Research*, 42(D1):D472–D477, 2013.
- [6] F. R. Davey, W. N. Erber, K. C. Gatter, and D. Y. Mason. Abnormal neutrophils in acute myeloid leukemia and myelodysplastic syndrome. *Human Pathology*, 19(4):454–459, 1988.
- [7] A. Defazio and T. S. Caetano. A convex formulation for learning scale-free networks via submodular relaxation. In *Advances in Neural Information Processing Systems*, pages 1250–1258, 2012.
- [8] E. Devijver and M. Gallopin. Block-diagonal covariance selection for high-dimensional Gaussian graphical models. *Journal of the American Statistical Association*, 113(521):306–314, 2018.
- [9] J. Duchi, S. Gould, and D. Koller. Projected subgradient methods for learning sparse Gaussians. In *In Proc. Int. Conf. Artificial Intelligence and Statistics*, pages 153–160, 2008.
- [10] M. Figueiredo and R. Nowak. Ordered weighted ℓ_1 regularized regression with strongly correlated covariates: Theoretical aspects. In *Proc. Int. Conf. Artificial Intelligence and Statistics*, pages 930–938, 2016.
- [11] J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.
- [12] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.
- [13] M. J. Hosseini and S.-I. Lee. Learning sparse Gaussian graphical models with overlapping blocks. In *Advances in Neural Information Processing Systems*, pages 3808–3816, 2016.
- [14] C.-J. Hsieh, M. A. Sustik, I. S. Dhillon, and P. K. Ravikumar. Sparse inverse covariance matrix estimation using quadratic approximation. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 2330–2338. 2011.
- [15] D. Koller, N. Friedman, and F. Bach. *Probabilistic graphical models: principles and techniques*. MIT press, 2009.
- [16] S. Kumar, J. Ying, J. V. d. M. Cardoso, and D. Palomar. A unified framework for structured graph learning via spectral constraints. *arXiv preprint arXiv:1904.09792*, 2019.

- [17] W. Lee and Y. Liu. Joint estimation of multiple precision matrices with common structures. *The Journal of Machine Learning Research*, 16(1):1035–1062, 2015.
- [18] H. Liu, F. Han, M. Yuan, J. Lafferty, L. Wasserman, et al. High-dimensional semiparametric Gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326, 2012.
- [19] N. Meinshausen, P. Bühlmann, et al. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.
- [20] MSCI. The Global Industry Classification Standard (GICS), 2019.
- [21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [22] W. E. Peterman, B. H. Ousterhout, T. L. Anderson, D. L. Drake, R. D. Semlitsch, and L. S. Eggert. Assessing modularity in genetic networks to manage spatially structured metapopulations. *Ecosphere*, 7(2), 2016.
- [23] B. Rolfs, B. Rajaratnam, D. Guillot, I. Wong, and A. Maleki. Iterative thresholding algorithm for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems*, pages 1574–1582, 2012.
- [24] E. A. Stone and J. F. Ayroles. Modulated modularity clustering as an exploratory tool for functional genomic inference. *PLoS Genetics*, 5(5):e1000479, 2009.
- [25] S. Sun, H. Wang, and J. Xu. Inferring block structure of graphical models in exponential families. In *In Proc. Int. Conf. Artificial Intelligence and Statistics*, pages 939–947, 2015.
- [26] K. M. Tan, D. Witten, and A. Shojaie. The cluster graphical lasso for improved estimation of Gaussian graphical models. *Computational Statistics & Data Analysis*, 85:23–36, 2015.
- [27] D. A. Tarzanagh and G. Michailidis. Estimation of graphical models through structured norm minimization. *Journal of Machine Learning Research*, 18(1), 2018.
- [28] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B*, 67(1):91–108, 2005.
- [29] J. Wang. Joint estimation of sparse multivariate regression and conditional graphical models. *Statistica Sinica*, pages 831–851, 2015.
- [30] M. Wytock and Z. Kolter. Sparse gaussian conditional random fields: Algorithms, theory, and application to energy forecasting. In *Proc. Int. Conf. of Machine Learning Research*, pages 1265–1273, 2013.
- [31] X. Zeng and M. A. Figueiredo. The ordered weighted ℓ_1 norm: Atomic formulation, projections, and algorithms. *arXiv preprint arXiv:1409.4271*, 2014.
- [32] T. Zhao, H. Liu, K. Roeder, J. Lafferty, and L. Wasserman. The huge package for high-dimensional undirected graph estimation in R. *Journal of Machine Learning Research*, 13(Apr):1059–1062, 2012.
- [33] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, 67(2):301–320, 2005.