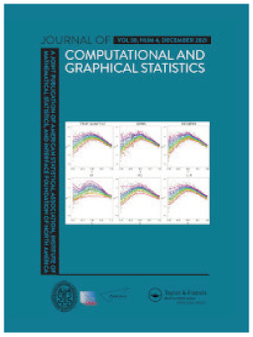

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/ucgs20

Estimating Multiple Precision Matrices With Cluster Fusion Regularization

Bradley S. Price, Aaron J. Molstad & Ben Sherwood

To cite this article: Bradley S. Price, Aaron J. Molstad & Ben Sherwood (2021) Estimating Multiple Precision Matrices With Cluster Fusion Regularization, Journal of Computational and Graphical Statistics, 30:4, 823-834, DOI: 10.1080/10618600.2021.1874963

To link to this article: https://doi.org/10.1080/10618600.2021.1874963

JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2021, VOL. 30, NO. 4, 823–834
https://doi.org/10.1080/10618600.2021.1874963

Estimating Multiple Precision Matrices With Cluster Fusion Regularization

Bradley S. Price[a], Aaron J. Molstad[b], and Ben Sherwood[c]

[a]Management Information Systems Department, West Virginia University, Morgantown, WV; [b]Department of Statistics and Genetics Institute, University of Florida, Gainesville, FL; [c]School of Business, University of Kansas, Lawrence, KS

ABSTRACT

We propose a penalized likelihood framework for estimating multiple precision matrices from different classes. Most existing methods either incorporate no information on relationships between the precision matrices or require this information be known a priori. The framework proposed in this article allows for simultaneous estimation of the precision matrices and relationships between the precision matrices. Sparse and nonsparse estimators are proposed, both of which require solving a nonconvex optimization problem. To compute our proposed estimators, we use an iterative algorithm which alternates between a convex optimization problem solved by blockwise coordinate descent and a k-means clustering problem. Blockwise updates for the sparse estimator require computing an elastic net penalized precision matrix estimation problem, which we solve using a proximal gradient descent algorithm. We prove that this subalgorithm has a linear rate of convergence. In simulation studies and two real data applications, we show that our method can outperform competitors that ignore relevant relationships between precision matrices and performs similarly to methods which use prior information often unknown in practice. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received January 2020
Revised October 2020

KEYWORDS
Discriminant analysis; Fusion penalties; Gaussian graphical models; Precision matrix estimation

1. Introduction

Many applications in statistics and machine learning require the estimation of multiple, possibly related, precision matrices. For example, to perform classification using quadratic discriminant analysis (QDA), a practitioner must estimate two or more precision matrices, see, for example, Chapter 4 of Friedman, Hastie, and Tibshirani (2001). Similarly, it is often of scientific interest to estimate multiple Gaussian graphical models when the same variables are measured on subjects from multiple classes, see, for example, Guo et al. (2011).

In this work, the data $(x_1, y_1), \ldots, (x_n, y_n)$ are assumed to be a realization of n independent copies of the random pair (X, Y) such that Y has support $\mathcal{C} = \{1, \ldots, C\}$ and

$$(X|Y = c) \sim \mathrm{N}_p\left(\mu_{*c}, \Omega_{*c}^{-1}\right), \quad c \in \mathcal{C},$$

where $\mu_{*c} \in \mathbb{R}^p$ and $\Omega_{*c} \in \mathbb{S}_+^p$ are unknown, and $\mathbb{S}_+^p$ denotes the set of $p \times p$ symmetric, positive-definite matrices. Let $n_c = \sum_{i=1}^n 1(y_i = c)$ be the sample size for the cth class, let $\bar{x}_c = n_c^{-1} \sum_{i=1}^n x_i 1(y_i = c)$ be the observed sample mean for the cth class, and let

$$S_c = \frac{1}{n_c} \sum_{i=1}^n (x_i - \bar{x}_c)(x_i - \bar{x}_c)^T 1(y_i = c), \qquad c \in \mathcal{C},$$

be the sample covariance matrix for the cth class, where $1(\cdot)$ is the indicator function. Define $\mathbf{\Omega} = \{\Omega_1, \ldots, \Omega_C\}$. After profiling over the means and class probabilities, negative two times the log-likelihood (ignoring constants) is

$$g(\mathbf{\Omega}) = \sum_{c \in \mathcal{C}} n_c \{\mathrm{tr}(S_c \Omega_c) - \log \det(\Omega_c)\}. \tag{1}$$

A natural estimator of Ω_{*c}, when it exists, is the maximum likelihood estimator S_c^{-1}. In settings where the maximum likelihood estimator does not exist, for example, when $p > n_c$, a practitioner could instead estimate the Ω_{*c}'s separately using penalized normal maximum likelihood (Pourahmadi 2011; Fan, Liao, and Liu 2016). Sparsity inducing penalties are especially popular in penalized normal maximum likelihood (Yuan and Lin 2007; d'Aspermont, Banerjee, and Ghaoui 2008; Friedman, Hastie, and Tibshirani 2007; Rothman et al. 2008; Witten, Friedman, and Simon 2011) because a zero in the (j, k)th entry of Ω_{*c} implies the conditional independence of the jth and kth variables (given all other variables) in the cth class. However when the precision matrices are similar across classes, for example, when the Ω_{*c}'s share sparsity patterns, jointly estimating the Ω_{*c}'s can be more efficient than methods that estimate each precision matrix separately.

Many methods exist for estimating multiple precision matrices under the assumption of shared structures (e.g., sparsity patterns) across classes. For example, Guo et al. (2011), Honorio and Samaras (2010), and Chiquet, Grandvalet, and Ambroise (2011) proposed to use a hierarchical penalty, $L_{1,\infty}$-norm penalty, and group-lasso penalty, respectively, to encourage zeroes in the same entries of estimates of the Ω_{*c}'s. Chiquet, Grandvalet, and Ambroise (2011) also proposed an estimator which replaces the class-wise sample covariance with a linear combination of the class-wise sample covariance and pooled sample covariance in the L_1-penalized normal maximum likelihood criterion. Hara and Washio (2011) assumed each precision matrix can be decomposed into the sum of a class-specific matrix and sparse matrix which is shared across classes.

CONTACT Bradley S. Price brad.price@mail.wvu.edu Management Information Systems Department, West Virginia University, Morgantown, WV 26506.
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

The problem of estimating multiple precision matrices is sometimes characterized as "multi-task structure learning" for networks (Niculescu-Mizil and Caruana 2007; Argyriou et al. 2008; Honorio and Samaras 2010; Gonçalves et al. 2014).

Most relevant to the method proposed here, Danaher, Wang, and Witten (2014) proposed the *fused graphical lasso* estimator (FGL)

$$\underset{\Omega_c \in \mathbb{S}^p_+, c \in \mathcal{C}}{\arg\min} \left\{ g(\boldsymbol{\Omega}) + \lambda_1 \sum_{c \in \mathcal{C}} \|\Omega_c\|_1 + \lambda_2 \sum_{(j,k) \in \mathcal{C} \times \mathcal{C}} \|\Omega_j - \Omega_k\|_1 \right\},$$

where $\|A\|_1 = \sum_{j=1}^p \sum_{k=1}^p |A_{j,k}|$. The first FGL penalty, controlled by positive tuning parameter λ_1, promotes elementwise sparsity separately *within* classes. The second penalty, controlled by λ_2, promotes elementwise equality jointly *across* classes. For sufficiently large values of the tuning parameter λ_2, the FGL estimates of the Ω_{*c}'s will have exactly equivalent sparsity patterns. Price, Geyer, and Rothman (2015) proposed a computationally efficient alternative to FGL, called *ridge fusion* (RF), which used squared Frobenius norm penalties in place of the L_1-norm penalties in FGL. Price, Geyer, and Rothman (2015) also investigated FGL and RF as methods for fitting the QDA model. These approaches for fitting the QDA model are related to Friedman (1989), who proposed regularized discriminant analysis (RDA). The RDA approach estimates multiple precision matrices for QDA using a linear combination of the sample covariance matrices for each class and the pooled sample covariance matrix across all the classes. Bilgrau et al. (2020) generalized the work of Price, Geyer, and Rothman (2015) to estimate multiple precision matrices sharing a common target matrix.

Joint estimation procedures such as those proposed by Danaher, Wang, and Witten (2014) and Price, Geyer, and Rothman (2015) can perform well when all C precision matrices are similar. However, there are settings when FGL and RF may perform unnecessary or inappropriate shrinkage. Notice, the second part of the ridge fusion penalty proposed by Price, Geyer, and Rothman (2015) can be rewritten as

$$\frac{1}{2C} \sum_{(j,k) \in \mathcal{C} \times \mathcal{C}} \|\Omega_j - \Omega_k\|_F^2 = \sum_{l \in \mathcal{C}} \|\Omega_l - \bar{\Omega}\|_F^2,$$
$$\text{where} \quad \bar{\Omega} = \frac{1}{C} \sum_{c \in \mathcal{C}} \Omega_c$$

and $\|A\|_F^2 = \text{tr}(A^T A)$ is the squared Frobenius norm. This formulation suggests that FGL and RF can be viewed as shrinking all precision matrices toward a common precision matrix. Regularization of this type may be problematic if there are substantial differences in the population precision matrices across classes. For instance, consider the case that there are two *groupings* (i.e., clusters) of the C classes denoted D_1 and D_2, where $D_1 \cap D_2$ is empty and $D_1 \cup D_2 = \mathcal{C}$. Suppose the (j,k)th entry of Ω_{*c}, $[\Omega_{*c}]_{j,k} = 0$ for $c \in D_1$, but $[\Omega_c]_{j,k} \neq 0$ for $c \in D_2$ for many $j \neq k$. This type of scenario may occur when the variables are the expression of p genes belonging to some pathway and the classes represent certain disease subtypes. Two subtypes may have similar gene-gene dependence, which are distinct from the another subtype (e.g., controls). In these settings, FGL may perform poorly since sparsity patterns are only shared within a subset of classes. If such clusters were known a priori, it may be preferable to apply FGL or RF to the groups separately, but when groupings are unknown, they must be estimated from the data.

Methods proposed by Zhu, Shen, and Pan (2014) and Saegusa and Shojaie (2016) addressed this issue of FGL and RF. The structural pursuit method proposed by Zhu, Shen, and Pan (2014) allows for heterogeneity in precision matrices using the truncated lasso penalty of Shen, Pan, and Zhu (2012) to promote elementwise equality and shared sparsity patterns across predefined groups of precision matrices. The method proposed by Saegusa and Shojaie (2016), known as LASICH, allows for heterogeneity of precision matrices through the use of a graph Laplacian penalty which incorporates prior information about how different classes' sparsity patterns are related. Since such prior information is often not available in practice, the authors propose the HC-LASICH method: a two-step procedure which first uses hierarchical clustering to estimate relationship between precision matrices, then uses this estimate to apply the LASICH procedure. Peterson, Stingo, and Vannucci (2015) addressed this problem from a Bayesian perspective by employing a spike and slab prior on parameters characterizing precision matrix relatedness. In somewhat related work, Ma and Michailidis (2016) proposed the joint structural estimation method to use prior information on shared sparsity patterns in a two-step procedure that first estimates the shared sparsity pattern and then estimates the precision matrices based on the shared sparsity constraints. More recently, Jalali, Khare, and Michailidis (2019) extended the work of Ma and Michailidis (2016) to the case where prior information on edge relationships need not be known. Jalali, Khare, and Michailidis (2019) used a Bayesian approach that incorporates a multivariate Gaussian mixture distribution on all possible sparsity patterns.

The joint estimation of multiple precision matrices has also been of interest in the model-based clustering and semi-supervised model-based clustering literature (Banfield and Raftery 1993; Fraley and Raftery 2002; Ruan, Yuan, and Zou 2011; Zhou, Pan, and Shen 2009; Xie, Pan, and Shen 2008; McNicholas and Murphy 2008). In these applications, some or all of the class labels $(y_1, \ldots, y_n)$ are unobserved. Price, Geyer, and Rothman (2015) extended the RF and FGL methods to the semi-supervised model-based clustering problem. Similarly, Gao et al. (2016) applied the non-convex structural pursuit penalty of Zhu, Shen, and Pan (2014) in the context of model-based clustering. More recently, Hao et al. (2018) proposed to estimate multiple graphical models using the SCAN method, which simultaneously estimates the parameters associated with multiple Gaussian graphical models using a group-lasso penalty.

In this article, we propose a penalized likelihood framework for simultaneously estimating the C precision matrices and how the precision matrices relate to one another. Like FGL and RF, our method can exploit the similarity of precision matrices belonging to a group, but avoids the unnecessary shrinkage of FGL or RF when groups differ. Unlike some existing methods, the proposed method does not require any prior information about the relationships between the classes, nor does it require clustering to take place before estimation of the precision matrices. We study the use of our estimator for QDA and Gaussian graphical modeling in settings where there are groupings of classes which share common dependence

structures. Computing our estimator is nontrivial since the objective function we minimize is discontinuous. To overcome this challenge, we propose an iterative algorithm, in which we alternate between updating groupings and updating precision matrix estimates. As part of our algorithm for the sparse estimator we propose (see Section 2), we must solve an elastic net penalized precision matrix estimation problem. To do so, we propose a graphical elastic net iterative shrinkage thresholding algorithm (GEN-ISTA). We prove this GEN-ISTA has a linear convergence rate and characterize the set to which the solution belongs. We provide R implementations of our proposed methods, as well as scripts to reproduce all simulations and data examples, at *https://github.com/bprice2652/cluster_fusion_precision*.

2. Joint Estimation With Cluster Fusion Penalties

2.1. Methods

Define $(D_1, \ldots, D_Q)$ to be an unknown Q element partition of the set $\mathcal{C}$. For convenience, we will refer to D_q as the qth cluster. Let $\lambda_1 > 0$, $\lambda_2 \geq 0$, and the positive integer Q be user defined tuning parameters.

For any set B define $\text{card}(B)$ as the cardinality of B. The first estimator we will investigate is the *cluster ridge fusion estimator* (CRF), which is defined as

$$(\hat{\boldsymbol{\Omega}}_{\mathbf{CRF}}, \hat{D}_{\text{CRF}}) = \underset{\Omega_c \in \mathbb{S}_+^p, c \in \mathcal{C}, D_1, \ldots, D_Q}{\arg\min} \left\{ g(\boldsymbol{\Omega}) + \frac{\lambda_1}{2} \sum_{c \in \mathcal{C}} \|\Omega_c\|_F^2 + \frac{\lambda_2}{2} \sum_{q=1}^{Q} \frac{1}{\text{card}(D_q)} \sum_{c,m \in D_q} \|\Omega_c - \Omega_m\|_F^2 \right\}. \tag{2}$$

We refer to the penalty associated with λ_2 as the *cluster fusion penalty*, which promotes similarities in precision matrices that are in the same cluster. The ridge fusion method (RF) proposed by Price, Geyer, and Rothman (2015) can be viewed as a special case of (2) when $Q = 1$.

We also propose a sparsity inducing version of the estimator, the *precision cluster elastic net* (PCEN), which is defined as

$$(\hat{\boldsymbol{\Omega}}_{\mathbf{PCEN}}, \hat{D}_{\text{PCEN}}) = \underset{\Omega_c \in \mathbb{S}_+^p, c \in \mathcal{C}, D_1, \ldots, D_Q}{\arg\min} \left\{ g(\boldsymbol{\Omega}) + \lambda_1 \sum_{c \in \mathcal{C}} \|\Omega_c\|_1 + \frac{\lambda_2}{2} \sum_{q=1}^{Q} \frac{1}{\text{card}(D_q)} \sum_{c,m \in D_q} \|\Omega_c - \Omega_m\|_F^2 \right\}. \tag{3}$$

When $\lambda_2 = 0$, $\hat{\boldsymbol{\Omega}}_{\mathbf{PCEN}}$ is equivalent to estimating the C precision matrices separately with L_1-penalized normal maximum likelihood.

In our proposed estimators, the cluster fusion penalty is used to promote similarity in precision matrices that are in the same cluster, while estimating precision matrices in different clusters separately from one another. When estimating Gaussian graphical models, PCEN promotes elementwise similarity between precision matrices in the same cluster, in turn promoting similar sparsity patterns within the same cluster. This differs from other methods, for example, FGL proposed by Danaher, Wang, and Witten (2014), which penalize the absolute value of entrywise differences across all precision matrices.

Unlike the FGL fusion penalty, the squared Frobenius norm fusion penalty will not lead to exact entrywise equality between estimated precision matrices—even those belonging to the same cluster. However, the squared Frobenius norm penalty facilitates fast computation and more importantly, an efficient search for clusters using existing algorithms for k-means clustering.

The cluster fusion penalty used in this work was first proposed in the context in univariate response linear regression by Witten, Shojaie, and Zhang (2014) to detect and promote similarity in effect sizes. More recently, Price and Sherwood (2018) used this type of cluster fusion penalty in multivariate response linear regression to detect and promote similarity in fitted values.

If $D_1, \ldots, D_Q$ were known, then Equations (2) and (3) could be rewritten as

$$\tilde{\boldsymbol{\Omega}}_{\mathbf{CRF}} = \underset{\Omega_c \in \mathbb{S}_+^p, c \in \mathcal{C}}{\arg\min} \left\{ g(\boldsymbol{\Omega}) + \frac{\lambda_1}{2} \sum_{c \in \mathcal{C}} \|\Omega_c\|_F^2 + \frac{\lambda_2}{2} \sum_{q=1}^{Q} \frac{1}{\text{card}(D_q)} \sum_{c,m \in D_q} \|\Omega_c - \Omega_m\|_F^2 \right\}, \tag{4}$$

$$\tilde{\boldsymbol{\Omega}}_{\mathbf{PCEN}} = \underset{\Omega_c \in \mathbb{S}_+^p, c \in \mathcal{C}}{\arg\min} \left\{ g(\boldsymbol{\Omega}) + \lambda_1 \sum_{c \in \mathcal{C}} \|\Omega_c\|_1 + \frac{\lambda_2}{2} \sum_{q=1}^{Q} \frac{1}{\text{card}(D_q)} \sum_{c,m \in D_q} \|\Omega_c - \Omega_m\|_F^2 \right\}. \tag{5}$$

The optimization in Equation (4) can be identified as Q separate ridge fusion estimation problems (Price, Geyer, and Rothman 2015). The optimization in Equation (5) is also separable over the Q clusters, and in Section 3.3 we propose a block coordinate descent algorithm to solve Equation (5).

A reviewer pointed out if $C = n$, the problem we are considering is related to clustering (of subjects). However, we assume each class has its own distinct mean vector, so to apply our method to the clustering problem, one may also have to perform some type of fusion or regularization of class means. We leave extensions of our method to model-based clustering as a direction for future research.

2.2. Tuning Parameter Selection for CRF and PCEN

We propose selecting tuning parameters, including the number of clusters, for both CRF and PCEN using validation likelihood with V-fold cross-validation. A similar approach was proposed by Price, Geyer, and Rothman (2015) for RF, which is a generalization of its use in the single precision matrix estimation problem (Huang et al. 2006). The data are randomly split into V subsets, dividing each of the C classes as evenly as possible. Let (v) index subjects belonging to the vth subset and let $\widehat{\Omega}_{c(-v)}^{(\lambda_1,\lambda_2,Q)}$ be an estimator of Ω_{*c} with the vth subset removed using tuning parameters $(\lambda_1, \lambda_2, Q)$. The validation likelihood score is then

calculated as

$$\mathrm{VL}(\lambda_1,\lambda_2,Q)=\sum_{v=1}^{V}\sum_{c\in\mathcal{C}} n_{c(v)}\left\{\mathrm{tr}\left(S_{c(v)}\widehat{\Omega}^{(\lambda_1,\lambda_2,Q)}_{c(-v)}\right)-\log\det\left(\widehat{\Omega}^{(\lambda_1,\lambda_2,Q)}_{c(-v)}\right)\right\}. \quad (6)$$

The tuning parameters are selected as the (λ_1,λ_2,Q) combination minimizing $\mathrm{VL}(\lambda_1,\lambda_2,Q)$ over a grid of candidate values. In the case of PCEN, if not enough data is available for cross-validation or computational time is a concern, then an information criterion, such as AIC from Danaher, Wang, and Witten (2014), could be used.

3. Computation

3.1. Overview

The objective functions in Equations (2) and (3) are discontinuous (and nonconvex) with respect to $D_1,\ldots,D_Q$ because changing cluster membership results in discrete changes in the objective function. However, in Equations (4) and (5), $D_1,\ldots,D_Q$ are fixed so that the objective functions are both convex with respect to $\mathbf{\Omega}$. To compute both CRF and PCEN, we propose an algorithm that iterates between solving for $D_1,\ldots,D_Q$ with $\mathbf{\Omega}$ fixed, and then solving for $\mathbf{\Omega}$ with $D_1,\ldots,D_Q$ fixed. This procedure is similar to those of Witten, Shojaie, and Zhang (2014) and Price and Sherwood (2018) which iterate between k-means clustering and solving an optimization with fixed clusters. Similar to Price and Sherwood (2018), we are able to exploit the embarrassingly parallel structure of the optimization when solving for $\mathbf{\Omega}$ with $D_1,\ldots,D_Q$ fixed. We describe both algorithms in the following subsections.

3.2. Cluster Ridge Fusion Algorithm

Assume $\lambda_1>0$, $\lambda_2\geq 0$, and $Q\in\{1,2,\ldots,C\}$. We propose the following iterative algorithm to solve Equation (2):

1. Initialize $\tilde{\mathbf{\Omega}}^1_{\mathbf{CRF}}=\{\tilde{\Omega}^1_1,\ldots,\tilde{\Omega}^1_C\}$ as a set of diagonal matrices where the jth diagonal element of $\tilde{\Omega}^1_k$ is $([S_k]_{j,j})^{-1}$.
2. For $w=2,3,4,\ldots$, repeat the steps below until the iterates $\tilde{D}^{w-1}_1,\ldots,\tilde{D}^{w-1}_Q$ are equivalent to $\tilde{D}^{w}_1,\ldots,\tilde{D}^{w}_Q$.
   (a) Holding $\tilde{\mathbf{\Omega}}^{w-1}_{\mathbf{CRF}}$ fixed, obtain the wth iterate of $(\tilde{D}_1,\ldots,\tilde{D}_Q)$ with

$$(\tilde{D}^w_1,\ldots,\tilde{D}^w_Q)=\underset{D_1,\ldots,D_Q}{\arg\min}\times\left\{\sum_{q=1}^{Q}\frac{1}{\mathrm{card}(D_q)}\sum_{c,m\in D_q}\|\tilde{\Omega}^{w-1}_c-\tilde{\Omega}^{w-1}_m\|_F^2\right\}. \quad (7)$$

   This is equivalent to solving the well-studied k-means clustering optimization problem on C vectors of dimension p^2 (Witten, Shojaie, and Zhang 2014).
   (b) Holding $(\tilde{D}^w_1,\ldots,\tilde{D}^w_Q)$ fixed, obtain the wth iterate of the precision matrices with

$$\tilde{\mathbf{\Omega}}^w_{\mathbf{CRF}}=\underset{\Omega_c\in\mathbb{S}^p_+,c\in\mathcal{C}}{\arg\min}\left\{g(\mathbf{\Omega})+\frac{\lambda_1}{2}\sum_{c\in\mathcal{C}}\|\Omega_c\|_F^2+\frac{\lambda_2}{2}\sum_{q=1}^{Q}\frac{1}{\mathrm{card}(\tilde{D}^w_q)}\sum_{c,m\in\tilde{D}^w_q}\|\Omega_c-\Omega_m\|_F^2\right\}. \quad (8)$$

   This is identical to the optimization in Equation (4) and can be solved with Q parallel RF estimation problems, with the qth objective function taking the form

$$\sum_{c\in D_q}\left[n_c\{\mathrm{tr}(S_c\Omega_c)-\log\det(\Omega_c)\}+\frac{\lambda_1}{2}\|\Omega_c\|_F^2\right]+\frac{\lambda_2}{2\mathrm{card}(D_q)}\sum_{c,m\in D_q}\|\Omega_c-\Omega_m\|_F^2.$$

To protect against the k-means clustering update in 2(a) from selecting a local optima, our implementation uses 100 random starts, and selects the clustering which gives the lowest objective function value (Hartigan and Wong 1979; Krishna and Murty 1999). Note that k-means clustering is just one approach to solving the optimization in (7). When C and Q are small, one could use an exhaustive search to solve (7). The complete CRF algorithm, including details for (b), can be found in the Supplementary Material.

3.3. Precision Cluster Elastic Net Algorithm

For the PCEN estimator, we propose to use the same iterative procedure as in Section 3.2. The algorithm iterates between a k-means clustering algorithm and a blockwise coordinate descent algorithm which uses the *graphical elastic net iterative soft-thresholding algorithm* (GEN-ISTA) to obtain new iterates of the precision matrices at each iteration.

Again, let $\lambda_1>0$, $\lambda_2\geq 0$, and $Q\in\{1,2,\ldots,C\}$ be fixed. Formally, the iterative algorithm is as follows:

1. Initialize $\tilde{\mathbf{\Omega}}^1_{\mathbf{PCEN}}=\{\tilde{\Omega}^1_1,\ldots,\tilde{\Omega}^1_C\}$ as a set of diagonal matrices where the jth diagonal element of $\tilde{\Omega}^1_k$ is $([S_k]_{j,j})^{-1}$.
2. For $w=2,3,4,\ldots$, repeat the steps below until the iterates $\tilde{D}^{w-1}_1,\ldots,\tilde{D}^{w-1}_Q$ are equivalent to $\tilde{D}^{w}_1,\ldots,\tilde{D}^{w}_Q$.
   (a) Holding $\tilde{\mathbf{\Omega}}^{w-1}_{\mathbf{PCEN}}$ fixed obtain the wth iterate of $(\tilde{D}_1,\ldots,\tilde{D}_Q)$ with

$$(\tilde{D}^w_1,\ldots,\tilde{D}^w_Q)=\underset{D_1,\ldots,D_Q}{\arg\min}\sum_{q=1}^{Q}\frac{1}{\mathrm{card}(D_q)}\times\sum_{c,m\in D_q}\|\tilde{\Omega}^{w-1}_c-\tilde{\Omega}^{w-1}_m\|_F^2. \quad (9)$$

   (b) Holding $(\tilde{D}^w_1,\ldots,\tilde{D}^w_Q)$, obtain the wth iterate of the precision matrix estimates with

$$\tilde{\boldsymbol{\Omega}}^{w}_{\mathbf{PCEN}} = \underset{\Omega_c \in \mathbb{S}^p_+, c \in \mathcal{C}}{\arg\min} \left\{ g(\boldsymbol{\Omega}) + \lambda_1 \sum_{c \in \mathcal{C}} \|\Omega_c\|_1 + \frac{\lambda_2}{2} \sum_{q=1}^{Q} \frac{1}{\text{card}(\tilde{D}^w_q)} \sum_{c,m \in \tilde{D}^w_q} \|\Omega_c - \Omega_m\|_F^2 \right\}. \tag{10}$$

Just as in the CRF Algorithm, to protect against selecting a local optima in the k-means clustering update in 2(a), our implementation uses 100 random starts, and selects the clustering which gives the lowest objective function value.

The update in Equation (10) is a nontrivial convex optimization problem. As noted previously, Equation (10) can be separated into Q separate optimization problems, where the qth optimization is

$$\underset{\Omega_c \in \mathbb{S}^p_+, c \in D_q}{\arg\min} \left(\left[\sum_{c \in D_q} n_c \{ \text{tr}(S_c \Omega_c) - \log\det(\Omega_c) \} \right] + \lambda_1 \sum_{c \in D_q} \|\Omega_c\|_1 + \frac{\lambda_2}{2\text{card}(\tilde{D}^w_q)} \sum_{c,m \in D_q} \|\Omega_c - \Omega_m\|_F^2 \right). \tag{11}$$

Since the D_q's are fixed, we propose to solve Equation (11) using blockwise coordinate descent where each Ω_c is treated as a block. That is, for each D_q, we update one Ω_c for $c \in D_q$ with all other $\Omega_{c'}$ for $c' \in D_q$ held fixed. The objective function for the Ω_c blockwise update, treating all other $\Omega_{c'}$, $c' \in D_q \setminus \{c\}$ as fixed is

$$n_c \left(\text{tr}\left[\left\{ S_c - \frac{\lambda_2}{n_c \text{card}(D^w_q)} \left(\sum_{c' \in D^w_q \setminus \{c\}} \Omega_{c'} \right) \right\} \Omega_c \right] - \log\det(\Omega_c) \right) + \lambda_1 \|\Omega_c\|_1 + \frac{\lambda_2 (\text{card}(D^w_q) - 1)}{2\text{card}(D^w_q)} \|\Omega_c\|_F^2. \tag{12}$$

Hence, by defining

$$\tilde{S}_c = \left\{ S_c - \frac{\lambda_2}{n_c \text{card}(D^w_q)} \left(\sum_{c' \in D^w_q \setminus \{c\}} \Omega_{c'} \right) \right\}, \quad \gamma_{c1} = \frac{\lambda_1}{n_c}, \quad \gamma_{c2} = \frac{\lambda_2 (\text{card}(D^w_q) - 1)}{2 n_c \text{card}(D^w_q)},$$

the argument minimizing (12) can be expressed

$$\underset{\Omega_c \in \mathbb{S}^p_+}{\arg\min} \left\{ \text{tr}(\tilde{S}_c \Omega_c) - \log\det(\Omega_c) + \gamma_{c1} \|\Omega_c\|_1 + \gamma_{c2} \|\Omega_c\|_F^2 \right\}, \tag{13}$$

which is the elastic net penalized normal maximum likelihood estimation criterion. To compute Equation (13), we propose the GEN-ISTA, an elastic net variation of the algorithm proposed by Rolfs et al. (2012), called G-ISTA, which was used to solve problems like (13) with $\gamma_{c2} = 0$. Iterative shrinkage thresholding algorithms (ISTAs) are a special case of the proximal gradient method, which are commonly used to solve penalized likelihood optimization problems. We refer the reader to Beck and Teboulle (2009) and Polson, Scott, and Willard (2015) for more on iterative shrinkage thresholding algorithms and proximal algorithms, respectively.

This approach uses a first-order Taylor expansion to derive a majorizing function of the objective in Equation (13). Let $f(\Omega_c) \equiv \text{tr}(\tilde{S}_c \Omega_c) - \log\det(\Omega_c) + \gamma_{c2} \|\Omega_c\|_F^2$ and let $\tilde{\Omega}_c$ be the previous iterate of Ω_c. Because ∇f is Lipschitz over compact sets of $\mathbb{S}^p_+$ (see Lemma 2), we have that

$$f(\Omega_c) \leq f(\tilde{\Omega}_c) + \text{tr}[(\Omega_c - \tilde{\Omega}_c)' \nabla f(\tilde{\Omega}_c)] + \frac{1}{2t} \|\Omega_c - \tilde{\Omega}_c\|_F^2, \tag{14}$$

for sufficiently small step size t, with equality when $\Omega_c = \tilde{\Omega}_c$. Thus, we can majorize f with the right-hand side of Equation (14): using this inequality and that $\nabla f(\Omega_c) = \tilde{S}_c - \Omega_c^{-1} + 2\gamma_{2c}\Omega_c$, we have

$$\begin{aligned} f(\Omega_c) \leq & -\log\det(\tilde{\Omega}_c) + \text{tr}[\tilde{\Omega}_c(\tilde{S}_c + \gamma_{c2}\tilde{\Omega}_c)] \\ & + \text{tr}[(\Omega_c - \tilde{\Omega}_c)'(\tilde{S}_c - \tilde{\Omega}_c^{-1} + 2\gamma_{c2}\tilde{\Omega}_c)] \\ & + \frac{1}{2t} \|\Omega_c - \tilde{\Omega}_c\|_F^2. \end{aligned} \tag{15}$$

Letting $g_t(\Omega_c; \tilde{\Omega}_c)$ denote the right-hand side of Equation (15), for all Ω_c with t sufficiently small,

$$f(\Omega_c) + \gamma_{c1} \|\Omega_c\|_1 \leq g_t(\Omega_c; \tilde{\Omega}_c) + \gamma_{c1} \|\Omega_c\|_1, \tag{16}$$

so that at $\tilde{\Omega}_c$, the right-hand side of Equation (16) is a majorizer of Equation (13). Thus, to solve Equation (13), we use an iterative procedure: given the previous iterate $\tilde{\Omega}_c$, we construct $g_t(\Omega_c; \tilde{\Omega}_c)$, then we minimize $g_t(\Omega_c; \tilde{\Omega}_c) + \gamma_{c1} \|\Omega_c\|_1$ to obtain the new iterate. This choice of majorizer is convenient since the new optimization problem simplifies to the proximal operator for the L_1-norm because

$$\begin{aligned} & \underset{\Omega_c \in \mathbb{S}^p}{\arg\min} \left\{ g_t(\Omega_c; \tilde{\Omega}_c) + \gamma_{c1} \|\Omega\|_1 \right\} \\ & \quad = \underset{\Omega_c \in \mathbb{S}^p}{\arg\min} \left\{ \frac{1}{2} \|\Omega_c - Z_{c,t}\|_F^2 + t\gamma_{c1} \|\Omega_c\|_1 \right\}, \end{aligned}$$

where $Z_{c,t} = \tilde{\Omega}_c - t(\tilde{S}_c - \tilde{\Omega}_c^{-1} + 2\gamma_{c2}\tilde{\Omega}_c)$ and $\mathbb{S}^p$ denotes the set of $p \times p$ symmetric matrices. In the following subsection, we will show that there always exists a step size such that the solution to the proximal operator above is positive definite, and hence, the iterates remain feasible for Equation (13).

To summarize, we propose the GEN-ISTA, which updates from iterate k to iterate $k+1$ with

$$\begin{aligned} \Omega_c^{(k+1)} &= \underset{\Omega \in \mathbb{S}^p}{\arg\min} \left\{ \frac{1}{2} \|\Omega_c - \Omega_c^{(k)} + t\{\tilde{S}_c - (\Omega_c^{(k)})^{-1} + 2\gamma_{c2}\Omega_c^{(k)}\}\|_F^2 + t\gamma_{c1}\|\Omega_c\|_1 \right\} \\ &= \mathcal{S}\left(\Omega_c^{(k)} - t\{\tilde{S}_c - \Omega_c^{-1(k)} + 2\gamma_{c2}\Omega_c^{(k)}\}, t\gamma_{c1} \right), \end{aligned} \tag{17}$$

where for a $p \times p$ matrix A and $\eta > 0$, $\mathcal{S}(A, \eta)$ is the elementwise soft-thresholding operator such that $[\mathcal{S}(A, \tau)]_{j,k} = \text{sign}(A_{j,k}) \max(|A_{j,k}| - \tau, 0)$. To select t for use in Equation (17), we use a backtracking line search. For the step to be accepted,

we check a descent condition and check that $\Omega_c^{(k+1)} \in \mathbb{S}_+^p$. If both conditions are not met, a smaller step size t must be used. The steps for implementing GEN-ISTA with backtracking line search can be found in the Supplementary Material. In Section 3.4 we show that for a prespecified t, which is a function of $\tilde{S}_c, \gamma_{c2}$, and p, that this update will always be contained in $\mathbb{S}_+^p$. The complete algorithm for PCEN can be found in the Supplementary Material.

As previously mentioned, the G-ISTA algorithm proposed by Rolfs et al. (2012) is a special case of the GEN-ISTA algorithm when $\gamma_{c2} = 0$, but there are substantial differences. In particular, Rolfs et al. (2012) only considered the case where $\tilde{S}_c$ is a symmetric, nonnegative-definite matrix, but in our application there is no guarantee that $\tilde{S}_c$ is nonnegative definite. In Section 3.4 we demonstrate the role of γ_{c2} in the rate of convergence and the choice of appropriate step size, t. The elastic net penalized normal likelihood precision matrix estimation problem was also studied by Atchadé, Mazumder, and Chen (2019), who proposed a stochastic gradient descent algorithm for solving Equation (13) with p very large and $\tilde{S}_c$ being a sample covariance matrix.

3.4. Convergence Analysis of GEN-ISTA Algorithm

In this section we will discuss the convergence of the GEN-ISTA subroutine proposed in the previous section. Our approach to convergence analysis is based on that of Rolfs et al. (2012), but in our application, we must address that the input matrix $\tilde{S}_c$ may be indefinite. We show that despite the generality of the input matrix, our proximal gradient descent scheme is guaranteed to converge at a linear rate and that the maximum step size is a function known quantities. Specifically, we show that there exists a worst case contraction constant, $\delta \in (0, 1)$, such that

$$\|\Omega_c^{(k+1)} - \Omega_c^*\|_F \leq \delta \|\Omega_c^{(k)} - \Omega_c^*\|_F,$$

where Ω_c^* is the solution to Equation (13). In our case δ is a function of $\tilde{S}_c, \gamma_{c2}$, and p. We will show that as γ_{c2} increases, δ approaches 0. Throughout this section, for a $p \times p$ matrix A, let $\rho_1(A) \geq \rho_2(A) \geq \cdots \geq \rho_p(A)$ denote the ordered eigenvalues of A. All proofs can be found in the Supplementary Material.

We first will show that Ω_c^* is contained in a compact subset of $\mathbb{S}_+^p$.

Lemma 1. If $\gamma_{c1} > 0, \gamma_{c2} > 0$, and Ω_c^* is the solution to Equation (13), then $\alpha I \preceq \Omega_c^* \preceq \beta I$, where

$$\alpha^{-1} = .5\left(\rho_1(\tilde{S}_c) + \gamma_{c1}p + \sqrt{(\rho_1(\tilde{S}_c) + \gamma_{c1}p)^2 + 8\gamma_{c2}}\right)$$

and

$$\beta^{-1} = .5\left(\rho_p(\tilde{S}_c) - \gamma_{c1}p + \sqrt{(\rho_p(\tilde{S}_c) - \gamma_{c1}p)^2 + 8\gamma_{c2}}\right).$$

Our bounds are distinct from those in Rolfs et al. (2012) as theirs do not allow for $\tilde{S}_c$ which is indefinite. Notably, the α we obtain is the same as that in Atchadé, Mazumder, and Chen (2019), although the β we obtain is distinct, again owing to the indefiniteness of $\tilde{S}_c$. Next, we establish that the Lipschitz continuity of the gradient of Equation (13), which we used to construct the majorizing function (14).

Lemma 2. If $\alpha I \preceq \Omega_A, \Omega_B \preceq \beta I$ such that $0 < \alpha < \beta < \infty$, then $\|\nabla f(\Omega_A) - \nabla f(\Omega_B)\|_F \leq \sqrt{p}\left(\frac{1}{\alpha^2} + 2\gamma_{c2}\right)\|\Omega_A - \Omega_B\|_F$. Hence, $\nabla f(\Omega) = \tilde{S}_c - \Omega^{-1} + 2\gamma_{c2}\Omega$ is Lipschitz on any compact subset of $\mathbb{S}_+^p$.

The combination of Lemmas 1 and 2 give us necessary and sufficient conditions to apply Theorem 3.1 of Beck and Teboulle (2009) to Equation (13) to obtain a sublinear convergence rate between iterates of the objective function.

Next, we present a lemma that ensures that there always exists a step size parameter t such that the iterates of the algorithm are contained in a compact subset of $\mathbb{S}_+^p$. The result of Lemma 3 is similar to those in Rolfs et al. (2012) and Atchadé, Mazumder, and Chen (2019).

Lemma 3. Let $\gamma_{c1} > 0$, $\gamma_{c2} > 0$, and define α and β as in Lemma 1. If $t \leq \frac{\alpha^2}{2\alpha^2\gamma_{c2}+1}$, then the iterates of the proposed algorithm satisfy $\alpha I \preceq \Omega_c^{(k)} \preceq b'I$ for all k where $b' = \|\Omega_c^*\|_2 + \|\Omega_c^{(0)} - \Omega_c^*\|_F \leq \beta + \sqrt{p}(\beta - \alpha)$.

Finally, we present a result establishing the linear convergence rate for our algorithm.

Theorem 1. Let α and β be as defined in Lemma 1. Then for constants $\gamma_{c1} > 0$, $\gamma_{c2} > 0$, and $t \leq \frac{\alpha^2}{2\alpha^2\gamma_{c2}+1}$ the iterates of our algorithm converge linearly with a rate of

$$\delta = 1 - 2\left[1 + \frac{2\gamma_{c2} + \alpha^{-2}}{2\gamma_{c2} + \left\{\beta + \sqrt{p}(\beta - \alpha)\right\}^{-2}}\right]^{-1} < 1.$$

Theorem 1 establishes the linear convergence of our proposed ISTA algorithm. Furthermore, these results show how γ_{c2} influences the convergence of the algorithm, and the optimal solution bounds. In particular, for a fixed γ_{c1}, as γ_{c2} gets larger, the rate approaches 0. From a practical perspective, these results suggest that we could fix the step size parameter t and avoid the backtracking line search when p is large because α and γ_{c2} can be calculated directly at each iteration.

4. Gaussian Graphical Modeling Simulation Studies

4.1. Overview

In our first set of simulations, we focus on both estimation accuracy and sparsity detection in Gaussian graphical modeling using PCEN. We generate data from $C = 4$ classes or $C = 6$ classes, where the cth class is generated from $\mathrm{N}_p(0, \Omega_{*c}^{-1})$ and $p \in \{20, 50, 100\}$. By construction, the sparsity patterns of Ω_{*1} and Ω_{*2} will be nearly equivalent; as will the sparisty patterns of Ω_{*3} and Ω_{*4}. However, the sparsity patterns of Ω_{*1} and Ω_{*2} will be distinct from the sparsity patterns of Ω_{*3} and Ω_{*4}. When six classes are present we set $\Omega_{*5} = \Omega_{*6}$.

We compare two versions of PCEN, PCEN-2 and PCEN-3 (i.e., (3) with $Q = 2$ and $Q = 3$, respectively) to the fused graphical lasso (FGL, Danaher, Wang, and Witten 2014), graphical lasso with the same tuning parameter for all classes (Glasso), the cooperative lasso (Coop-Lasso; Chiquet, Grandvalet, and Ambroise 2011), and two versions of the method proposed by

Saegusa and Shojaie (2016) which we call LASICH-OR and LASICH-PR (denoting "oracle" and "practical", respectively). The method proposed by Saegusa and Shojaie (2016) requires the network information between the classes to be known before fitting the precision matrices (i.e., "oracle" information), though it may be estimated using hierarchical clustering. In this simulation, a network where the edges are $\{(1,2),(1,3),(2,4),(3,4)\}$ is used. The difference between LASICH-OR and LASICH-PR is that LASICH-OR applies weights of 10^{-3} to the edges in the set $\{(1,3),(2,4)\}$ while LASICH-PR weights all edges equally. Thus, this can be considered a "best-case" version of the HC-LASICH method. In the case when $C = 6$ we add the edges in the set $\{(1,5),(3,5),(2,4),(4,6),(5,6)\}$, with the edge weight of 10^{-3} applied in the case of LASICH-OR to all of the edges added with the exception of the edge $(5,6)$. Tuning parameters for each of the methods are investigated based a subset of $(\lambda_1,\lambda_2) \in \{10^{-10},10^{-9.9},\ldots,10^{9.9},10^{10}\} \times \{10^{-3},10^{-1},10^{1}\}$ unless otherwise specified. The R implementation of the cooperative lasso estimator (`simone` on CRAN) could only be used to obtain relatively sparse estimates of the Ω_{*c}. We report those in our simulation results.

To evaluate performance of each estimator, we measure the true positive rate across all C classes, which we define as $\sum_{c=1}^{C}\sum_{(j,k)} 1([\Omega_{*c}]_{j,k} \neq 0 \cap [\hat{\Omega}_c]_{j,k} \neq 0)/\sum_{c=1}^{C}\sum_{(j,k)} 1([\Omega_{*c}]_{j,k} \neq 0)$, where $\hat{\Omega}_c$ is an estimate of Ω_{*c}.

In addition, we also report the sum of the Frobenius norm squared error which is defined as $\sum_{c=1}^{C}\|\Omega_{*c} - \hat{\Omega}_c\|_F^2$. We compare these metrics to the number of nonzero elements across all $\hat{\Omega}_c$ $(c = 1,\ldots,C)$ as a way to measure the level of the total sparsity of the estimates. A full analysis of the ability of PCEN to detect the correct clustering for a given set of tuning parameters and timing analysis are contained in the Supplementary Material of this manuscript.

In each replication, the training data consist of n independent draws from each of the class distributions. We investigate three different settings each based on Erdős-Rényi graphs. Throughout the settings we consider, we define $E(A,p)$ to be a $p \times p$ matrix where A is an adjacency matrix associated with an Erdős-Rényi graph. To generate the elements of $E(A,p)$, we randomly assign each of the nonzero elements of A a value from the set $(-0.7,-0.5) \cup (0.5,0.7)$. Each off diagonal element is normalized by 1.5 times the row sum of the matrix, and each diagonal element is set to 1. The matrix is then scaled such that the associated variance of each of the p variables is 1. Furthermore, we define $R(A,\Omega_*,V)$ to be a $p \times p$ matrix that is generated using the adjacency matrix A, such that nonzero elements are equal to the corresponding value in Ω_* plus a randomly selected value from the set V. The off-diagonal elements are normalized by 1.5 times the row sum of the matrix, the diagonal elements are set to 1. Finally, the entire matrix is normalized such that the variance of each variable is 1. Similar data-generating mechanisms have been used in Danaher, Wang, and Witten (2014) and Saegusa and Shojaie (2016).

4.2. Two Clusters, Block Erdős-Rényi Graphs

We first compare PCEN-2 and PCEN-3 to competing methods under block Erdős-Rényi graphs. Each (p,λ_1,λ_2) described in Section 4.1 is replicated 50 times with $n = 200$ and $C = 4$. In this setting, we generate Ω_{*1} to be block diagonal with each block of size $p/2 \times p/2$. The first block is generated using $U = E(A_1,p/2)$, and the second is generated using $L = E(A_2,p/2)$ where A_1 and A_2 are adjacency matrices associated with independent Erdős-Rényi graphs with $p/2$ edges. Using Ω_{*1}, we generate Ω_{*2} such that it is block diagonal with block size $p/2 \times p/2$. We define the upper block of Ω_{*2} as $R(A_3,L,(-.01,.01))$, and the lower block to be $R(A_4,U,(-.01,.01))$ where A_3 is the adjacency matrix A_1 with four edges removed. Similarly A_4 is the adjacency matrix A_2 with four edges removed. Hence, Ω_{*1} and Ω_{*2} have nearly equivalent sparsity patterns minus eight nonzero entries in Ω_{*1} which are zero in Ω_{*2}.

To generate Ω_{*3} we randomly select $p/2$ variables and define this set of variables as s_1 and define $s_2 = \{1,\ldots,p\} \setminus s_1$. The submatrix of Ω_{*3} corresponding to the indices in s_1 are generated such that $G = E(A_5,p/2)$ and submatrix of Ω_* corresponding to the indices in s_2 is generated such that $H = E(A_6,p/2)$, where A_5 and A_6 are independent Erdős-Rényi graphs with $p/2$ edges. The submatrices of Ω_{*4} corresponding to the indices in s_1 and s_2 are generated using $R(A_7,G,(-.01,.01))$ and $R(A_8,H,(-.01,.01))$, respectively. The adjacency matrices A_7 and A_8 are the same as A_5 and A_6, respectively, with 4 randomly selected edges removed in each.

The results in panels (a) and (b) of Figure 1 are average log sum of squared Frobenius norm error and the average true positive rate as the number of nonzero elements in the estimated precision matrices varies with $p = 100$. The results for the case of $p = 20$ and $p = 50$ can be found in the Supplementary Material. The results in panels (a) and (b) of Figure 1 suggest that PCEN-2 can preform as well or better than competitors in terms of Frobenius norm error and graph recovery. Notably, for some tuning parameters, PCEN-2 outperforms LASICH-OR both in terms of log sum of squared Frobenius norm and TPR, even though LASICH-OR knows the relations between precision matrices *a priori*. Investigating further in the $p = 100$ case, we find that this corresponds to situations where the groupings are identified correctly in every replication. In cases where $p = 20$ and $p = 50$ PCEN-2 performs the best compared to other methods when it is able to detect the correct groupings of precision matrices. For every value of p when the proportion of nonzero elements in $\hat{\boldsymbol{\Omega}}$ is small, PCEN-2 and PCEN-3 perform similarly regardless of the true number of groups. Nonetheless, the minimum Frobenius norm achieved by either method is always achieved by that with the correct number of clusters. Further analysis of the cluster detection can be found in the Supplementary Material.

In the Supplementary Material, we present additional simulation results examining the effect of sample size and λ_2 on the performance of PCEN-2. Briefly, as one would expect, as the sample size increases, the performance of PCEN-2 improves. In general, as λ_2 increases, the performance also improves.

4.3. Two Clusters, Block Diagonal Erdős-Rényi Graphs

In contrast to the data-generating models in Section 4.2, in these simulations we consider settings where all four precision matrices have a high degree of shared sparsity with high probability. We generate Ω_{*1} such that it is block diagonal with each block size of $p/2 \times p/2$. The first block is generated

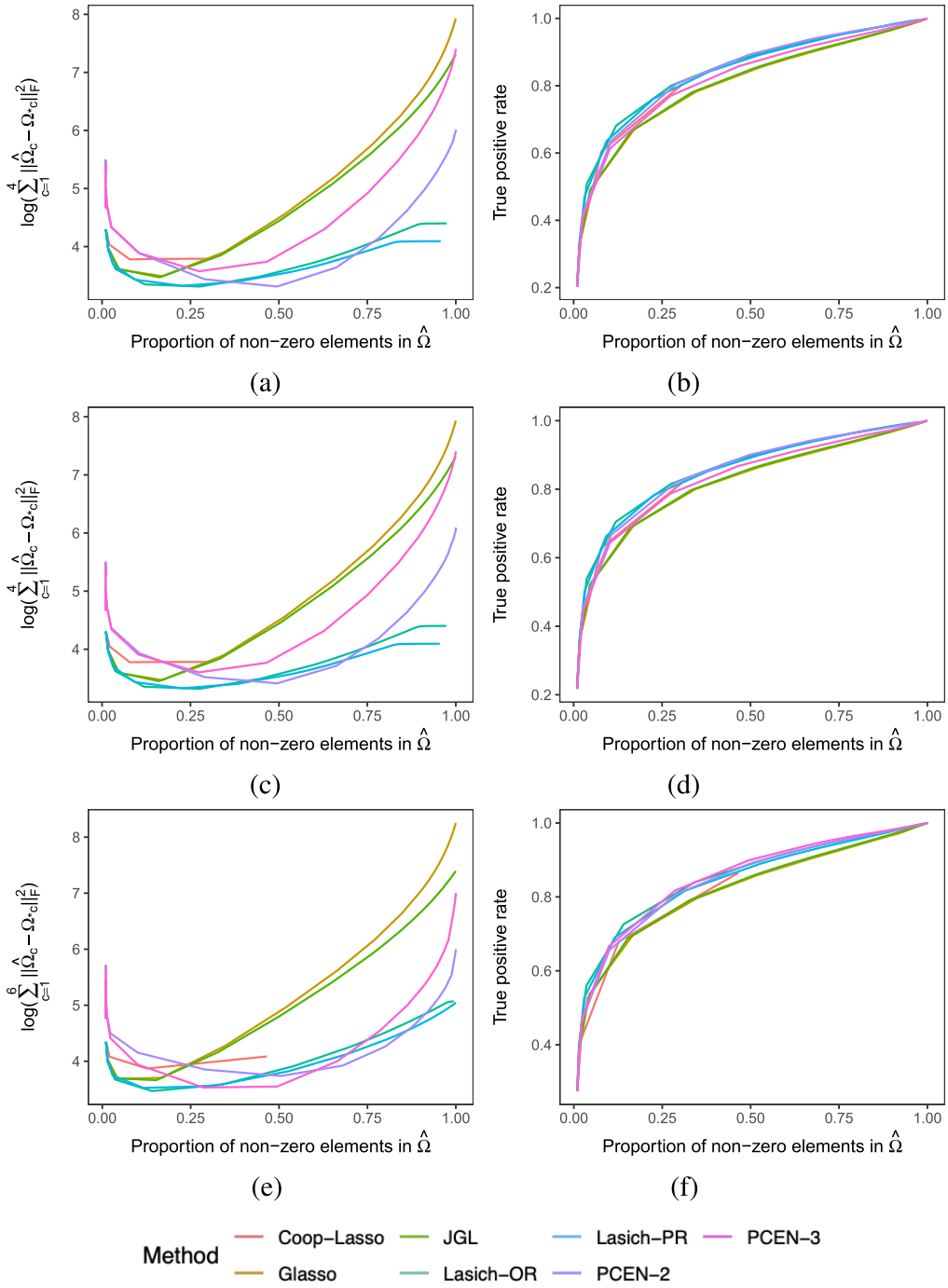

Figure 1. Results for the simulation setting described in (a)–(b) Section 4.2, (c)–(d) Section 4.3, and (e)–(f) Section 4.4 when $p = 100$. In (a)–(d), there are $Q = 2$ clusters, and in (e)–(f), there are $Q = 3$ clusters. Each line represents the average of 50 replications of the denoted method when λ_2 is fixed, and λ_1 varies. Note that LASICH-OR and LASICH-PR use exact and approximate information about the true clusters in estimation, respectively.

using $U = E(A_1, p/2)$, and the second block is generated from $L = E(A_2, p/2)$ where A_1 and A_2 are adjacency matrices associated with independent Erdős-Rényi graphs, with $p/2$ edges. Using Ω_{*1} we generate Ω_{*2} such that it is block diagonal with block size $p/2 \times p/2$. We define the upper block of Ω_{*2} as $R(A_3, L, (-.01, .01))$, and the lower block to be $R(A_4, U, (-.01, .01))$ where A_3 is the adjacency matrix A_1 with four edges removed. Similarly A_4 is the adjacency matrix A_2 with $p/10$ edges removed. Next, Ω_{*3} is generated in a similar way to Ω_{*1}, and Ω_{*4} is generated from Ω_{*3} in the same fashion Ω_{*2} is generated from Ω_{*1}. By generating precision matrices in this way, entries not in the upper or lower block submatrix are zero in all four precision matrices.

The results in panels (c) and (d) of Figure 1 are average log sum of squared Frobenius norm error and the average true positive rate as the number of nonzero elements in the precision matrices varying with $p = 100$ and $n = 200$. The results for the case of $p = 20$ and $p = 50$ can be found in the Supplementary Material. These results show a similar pattern to the results from the simulation studies in Section 4.2. For certain values of λ_1, which control the number of nonzero entries in the matrices $\hat{\mathbf{\Omega}}$, PCEN-2 is competitive in Frobenius norm error

and graph recovery with the all other methods, most notably LASICH-OR. In the case of $p = 100$, this corresponds to values of the tuning parameters where PCEN-2 is able to correctly identify the groupings of precision matrices. When $p = 20$ and $p = 50$, this is when groupings of precision matrices are correctly identified and estimates are sparse. Further analysis on cluster detection can be found in the Supplementary Material. As mentioned, LASICH-OR has oracle knowledge of the true relationships between precision matrices, while PCEN is estimating the relationships as well as the precision matrices.

4.4. Three Clusters, Block Diagonal Structures

In the final setting, we assume a data-generating model where six precision matrices are divided into three groups. We generate Ω_{*1} such that it is block diagonal with each block size of $p/2 \times p/2$. The first block is generated using $U = E(A_1, p/2)$, and the second block is the identity matrix, where A_1 is an adjacency matrix from an Erdős-Rényi, with $p/2$ connections. Using Ω_{*1} we generate Ω_{*2} such that it is block diagonal with block size $p/2 \times p/2$. We define the upper block of Ω_{*2} as $R(A_3, L, (-.01, .01))$, and the lower block to be the identity where A_3 is the adjacency matrix A_1 with four edges removed. Next, Ω_{*3} is generated in a similar way to Ω_{*1} and Ω_{*4} is generated from Ω_{*3} in the same fashion Ω_{*2} is generated from Ω_{*1}. We generate $\Omega_{*5} = \Omega_{*6}$ such that they are equivalent to the precision matrix from class 1 described in Section 4.3.

The results in panels (e) and (f) Figure 1 are average log sum of squared Frobenius norm error and the average true positive rate as the number of nonzero elements in the precision matrices varying with $p = 100$ and $n = 200$. The results for the case of $p = 20$ and $p = 50$ can be found in the Supplementary Material. Results exhibit a similar pattern to the results displayed in Sections 4.2 and 4.3. For certain values of the tuning parameters, PCEN-3 is competitive in estimation and graph recovery with the other methods, specifically LASICH-OR. As p increases, we see the estimation and graph recovery of PCEN decreases relative to LASICH-OR, but is still competitive with other competitors. Again, this can be attributed to LASICH-OR having oracle information and its use of the group penalty which exploits similar sparsity patterns across all precision matrices. As in Sections 4.2 and 4.3, PCEN-3 performs the best with respect to Frobenius norm error and graph recovery when it is able to identify the true relationships between the precision matrices for all p. A full analysis of cluster recovery can be found in the Supplementary Material.

5. QDA Simulations Studies

Since CRF produces nonsparse estimates of multiple precision matrices, it is not appropriate for Gaussian graphical modeling, but is a natural estimator for QDA. Hence, in this section, we study CRF as a method for fitting the QDA model. We generate data from $C = 4$ classes, where predictors for the cth class are generated from $\mathrm{N}_p(\mu_{*c}, \Sigma_{*c})$ with $p \in \{20, 50\}$. The training data consists of 25 independent realizations from each class. Tuning parameters are selected using 5-fold cross-validation maximizing the validation likelihood. We measure classification accuracy to compare methods. To quantify classification accuracy, we generate an independent testing set consisting of 500 observations from each of the $C = 4$ classes.

In addition to CRF, RF, and RDA (Friedman 1989), we include two methods which have oracle knowledge of the population parameters: Oracle, which uses Ω_{*c} and μ_{*c} in the classification rule; and TC (for "true covariance"), which uses Ω_{*c} and the sample means in the classification rule. These oracle methods provide a benchmark for classification accuracy in these data. We omit the sparse methods discussed in Section 4 as we study a class of dense precision matrices in this particular simulation study. For further discussion on the differences between L_1 and ridge-penalized precision matrix estimators in QDA, we refer the reader to Price, Geyer, and Rothman (2015) and references therein.

We consider a situation where each of the two clusters has a distinct structure and precision matrices in both clusters are dense. For 100 independent replications, we generate $Z_3 \in \mathbb{R}^{100 \times p}$ where each row is an independent realization of $\mathrm{N}_p(0, I)$ and let V_3 be the right singular vectors of Z_3. We then let $\Sigma_{*1} = V_3^T H_3 V_3$ and $\Sigma_{*2} = V_3^T H_4 V_3$ where H_3 and H_4 are diagonal matrices with the jth element equal to $D(1000, 100, j)$ and $D(999, 99, j)$, respectively. Define the (j,k)th element of $(\Sigma_{*3})_{j,k} = 1(j = k) + 0.45 \cdot 1(|j - k| = 1)$ and $(\Sigma_{*4})_{j,k} = 1(j = k) + \rho \cdot 1(|j - k| = 1)$ where $1(\cdot)$ is the indicator function. We consider $(p, \rho) \in \{20, 50\} \times \{0.40, 0.47, 0.50\}$. Finally, we set all elements of $\mu_{*1} = 20\log(p)/p$, $\mu_{*2} = -10\log(p)/p$, $\mu_{*3} = 10\log(p)/p$, and $\mu_{*4} = -20\log(p)/p$. A similar data generating model was used in Price, Geyer, and Rothman (2015). We expect CRF to perform well in this setting as it should be able to identify the distinct clusters, while RDA and RF implicitly assume similar structures across all precision matrices.

Table 1 presents a comparison of the classification error rate, and demonstrates that CRF out performs RDA and RF for every (p, ρ) combination. Interestingly, in the case that $p = 20$, CRF

Table 1. Results of simulation described in Section 5 comparing classification error rates and standard errors of CRF, RDA, RF and the two oracle methods for $(p, \epsilon) \in \{20, 50\} \times \{1.0\}$.

	$p = 20$					$p = 50$				
	RF	CRF	RDA	Oracle	TC	RF	CRF	RDA	Oracle	TC
$\rho = 0.40$	0.237 (0.001)	0.106 (0.003)	0.237 (0.001)	0.015 (0.002)	0.108 (0.013)	0.238 (0.001)	0.130 (0.002)	0.238 (0.001)	0.005 (0.000)	0.075 (0.010)
$\rho = 0.47$	0.238 (0.002)	0.113 (0.004)	0.237 (0.001)	0.015 (0.002)	0.090 (0.012)	0.238 (0.001)	0.130 (0.002)	0.238 (0.001)	0.005 (0.002)	0.075 (0.010)
$\rho = 0.50$	0.238 (0.002)	0.111 (0.004)	0.236 (0.001)	0.103 (0.002)	0.108 (0.012)	0.238 (0.001)	0.130 (0.002)	0.238 (0.001)	0.005 (0.002)	0.075 (0.010)

performs nearly as well as TC, which uses the true covariance matrices. Moreover, when $p = 20$ CRF is able to recover the true grouping of precisions matrices in 28% of replications for $\rho = 0.40$, 54% of replications for $\rho = 0.47$, and 62% of replications for $\rho = 0.50$ respectively. In all cases where the correct grouping was not identified, $Q = 3$ was selected, and the precision matrices for class 1 and 2 were placed in the same group. In the case of $p = 50$, CRF is able to recover the true grouping of precision matrices in all replications for each (ρ, p) combination.

In the Supplementary Material, we provide additional simulation study settings and results under clustered, dense, and ill-conditioned precision matrices.

6. Data Examples

6.1. Gene Expression from Pulmonary Hypertension Patients

Cheadle et al. (2012) collected gene expression profiles of 30 idiopathic pulmonary arterial hypertension patients (IPAH), 19 systemic sclerosis patients without pulmonary hypertension (SS w/o PH), 42 scleroderma-associated pulmonary arterial hypertension patients (SPAH), 8 systemic sclerosis patients with interstitial lung disease and pulmonary hypertension, and 41 healthy individuals, for a total of 140 individuals from five distinct groups. The collected gene expression profiles consist of data from 49,576 probes. We scaled each probe to have a median of 256 and then performed a $\log_2$ transformation. Next, we scaled and centered the log transformed data to have mean zero and a standard deviation of one. Our analysis was focused on 132 individuals ($C = 4$), excluding the 8 systemic sclerosis patients with interstitial lung disease and pulmonary hypertension, and 132 gene expression probes. The 132 probes we used were selected by running a one-factor ANOVA for each probe, using disease type as the factor, and then selecting the 132 probes with the smallest p-values.

After this processing, we fit the PCEN model to the normalized data. The PCEN shrinkage tuning parameters were selected to promote sparsity in the graph and similarity between the graphs based on AIC and interpretability, similar to the procedure of Danaher, Wang, and Witten (2014). We investigated the use of $Q = 2$ and $Q = 3$ clusters for these data. In both settings, PCEN was able to differentiate between the controls and patients with hypertension. In the case of two clusters, IPAH, SPAH and SS w/o PH are placed into a cluster while the control group is isolated in the second cluster. In the case of three clusters IPAH and SS w/o PH are placed into a cluster, while SPAH and the control group are both their own cluster of size one.

Figure 2 displays the corresponding network structures found using PCEN with $Q = 3$, representing the graph with the lowest AIC. A similar plot for $Q = 2$ is displayed in the Supplementary Material. In Figure 2, the blue edges represent probes that are related and were only found in patients diagnosed with IPAH, while light blue edges correspond to related probes found only in patients diagnosed with SPAH. Red edges denote relationships between probes that could be found in patients who were diagnosed with SPAH and those patients who were diagnosed with IPAH. Purple edges denote relationships between probes that could be found in patients who were diagnosed with SPAH and those patients who were diagnosed with IPAH and those who were diagnosed with SS w/o PH. Table 2 presents the number of edges that appear in only IPAH and SPAH, and then the edges that are present in both graphs.

At first inspection, the results between the cases of $Q = 3$, shown in Figure 2, and $Q = 2$, presented in the Supplementary Material, appear similar, but there are very notable differences.

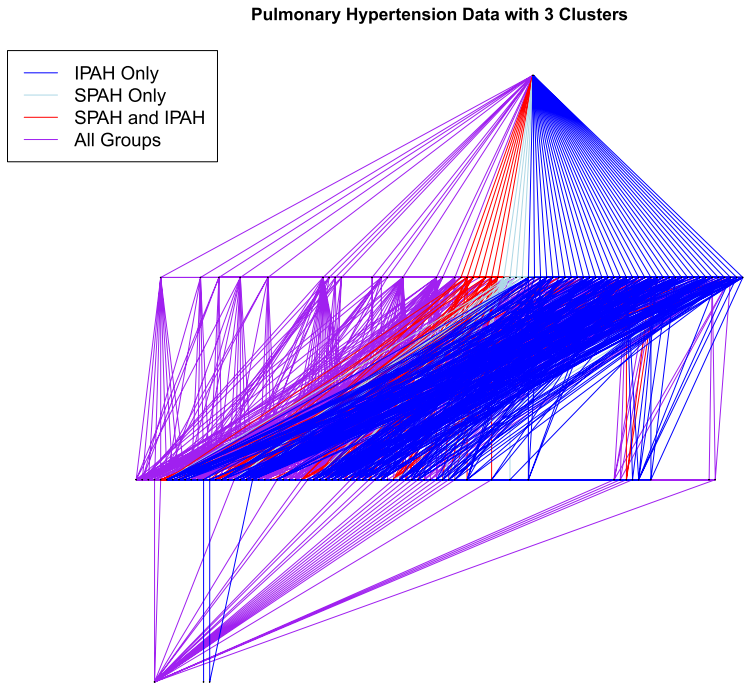

Figure 2. Resulting network comparison from PCEN applied to the pulmonary hypertension patients data using $Q = 3$ clusters.

Table 2. A comparison of network differences produced by PCEN using 2 and 3 clusters for the pulmonary hypertension patients data.

	IPAH	SPAH	IPAH and SPAH	All Groups	Total
PCEN 2 Clusters	594	343	412	688	2037
PCEN 3 Clusters	867	102	290	942	2201

NOTE: The values in the table are the number of edges that are present only in IPAH, SPAH, or are present in both.

Table 3. Classification results from the Libras Data example.

Method	CRF	RDA	Ridge	Ridge Fusion
Error rate	13/60	20/60	51/60	49/60

When $Q = 3$ and SPAH belongs to its own cluster, we see that the number of shared edges between all groups is larger when compared to the same metric using $Q = 2$. The other differences, which are quantified in Table 2, can be attributed to the changing cluster structure and may have important biological implications. In the Supplementary Material, we provide the graphs associated with the graphical lasso and cooperative lasso, where tuning parameters were chosen using AIC. This shows that cooperative lasso produces a much more sparse graph than the other methods. The graphical lasso produces a similar graph structure to PCEN, but with more edges in the graph and PCEN is able to detect different similarities in the graph.

6.2. Libras Data Example

To further demonstrate the useful of our proposed method, we apply CRF to a classification problem based on the Libras data set from the UCI Machine Learning repository (Dheeru and Taniskidou 2017). These data contain 15 classes, each of which corresponds to a videoed hand movement of Brazilian sign language. Each hand movement was recorded at 45 distinct time frames and the coordinates on an $x-y$ plane were documented, which results in 90 predictor variables for the hand movement. Each of the 15 classes has 24 observations for a total of 360 observations. Training was done using 20 randomly selected observations from each class, and testing was done on the four remaining observations. Our test and training sets are available in the supplementary material. We compare four methods: CRF, RF, ridge penalized normal likelihood precision matrix estimation, and RDA. The ridge penalized normal likelihood precision matrix estimator is equivalent to CRF with $\lambda_2 = 0$. Tuning parameters were selected by five-fold cross-validation maximizing a validation likelihood for all likelihood based methods. In the case of CRF, the number of clusters was chosen from the set of integers ranging from 2 to 10. For the non-likelihood method, RDA, we selected tuning parameters by five-fold cross-validation minimizing the misclassification rate.

Table 3 contains the classification error rate for each of the five methods on the testing data. The CRF method outperforms the other methods in terms of classification rate and detects two clusters, with one cluster containing 14 of the classes and the other containing the horizontal zig–zag class. Further investigation shows that for CRF 9 out of 15 of the classes had a CER of 0. For comparison, results presented by Li and Liu (2018) show that using modern machine learning methods with a training sample size of 240 observations produced CER that varied between 0.20 and 0.53, with the best method on average being 0.31. This work also showed that for these data, as the number of training samples increased the average CER decreased. Our results are consistent with these findings.

Acknowledgments

We thank to the AE and two anonymous reviewers for their valuable feedback which helped improve this article.

Funding

This work has been supported in part by NSF MRI Award # 11726534 and Big XII Faculty Fellowships by the University of Kansas and West Virginia University.

Supplementary Materials

The supplementary material provides details on proofs for the theoretical results provided as well as deeper insights on simulations and algorithms described in this article.

ORCID

Bradley S. Price http://orcid.org/0000-0002-0619-3347

References

Argyriou, A., Pontil, M., Ying, Y., and Micchelli, C. A. (2008), "A Spectral Regularization Framework for Multi-task Structure Learning," in *Advances in Neural Information Processing Systems* (Vol. 20), eds. J. Platt, D. Koller, Y. Singer, and S. Rowes, Curran Associates, Inc., pp. 25–32. [824]

Atchadé, Y. F., Mazumder, R., and Chen, J. (2019), "Scalable Algorithms for Regularized Precision Matrices via Stochasitc Optimization." Available at *http://dept.stat.lsa.umich.edu/~yvesa/glasso_v1.pdf*. [828]

Banfield, J., and Raftery, A. (1993), "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics*, 49, 803–821. [824]

Beck, A., and Teboulle, M. (2009), "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM Journal on Imaging Sciences*, 2, 183–202. [827,828]

Bilgrau, A. E., Peeters, C. F., Eriksen, P. S., Bogsted, M., and Wieringen, W. N. V. (2020), "Targeted Fused Ridge Estimation of Inverse Covariance Matrices from Multiple High-Dimensional Data Classes," *Journal of Machine Learning Research*, 21, 1–52. [824]

Cheadle, C., Berger, A. E., Mathai, S. C., Grigoryev, D. N., Watkins, T. N., Sugawara, Y., Barkataki, S., Fan, J., Boorgula, M., Hummers, L., Zaiman, A. L., Girgis, R., McDevitt, M. A., Johns, R. A., Wigley, F., Barnes, K. C., and Hassoun, P. M. (2012), "Erythroid-Specific Transcriptional Changes in PBMCs from Pulmonary Hypertension Patients," *PLOS One*, 7, 1–14. [832]

Chiquet, J., Grandvalet, Y., and Ambroise, C. (2011), "Inferring Multiple Graphical Structures," *Statistics and Computing*, 21, 537–553. [823,828]

Danaher, P., Wang, P., and Witten, D. M. (2014), "The Joint Graphical Lasso for Inverse Covariance Estimation Across Multiple Classes," *Journal of the Royal Statistical Society*, 76, 373–397. [824,825,826,828,829,832]

d'Aspermont, A., Banerjee, O., and Ghaoui, L. E. (2008), "First-Order Methods for Sparse Covariance Selection," *SIAM Journal of Matrix Analysis and Applications*, 30, 56–66. [823]

Dheeru, D., and Taniskidou, E. K. (2017), "UCI Machine Learning Repository." [online]. Available at *http://archive.ics.uci.edu/ml*. [833]

Fan, J., Liao, Y., and Liu, H. (2016), "An Overview of the Estimation of Large Covariance and Precision Matrices," *The Econometrics Journal*, 19, C1–C32. [823]

Fraley, C., and Raftery, A. (2002), "Model Based Clustering, Discriminant Analysis, and Density Estimation," *Journal of American Statistical Association*, 97, 611–632. [824]

Friedman, J. (1989), "Regularized Discriminant Analysis," *Journal of the American Statistical Association*, 84, 249–266. [824,831]

Friedman, J., Hastie, T., and Tibshirani, R. (2001), *The Elements of Statistical Learning*, Springer Series in Statistics (Vol. 1). New York: Springer. [823]

——— (2007), "Sparse Inverse Covariance Estimation With the Graphical Lasso," *Biostatistics*, 9, 432–441. [823]

Gao, C., Zhu, Y., Shen, X., and Pan, W. (2016), "Estimation of Multiple Networks in Gaussian Mixture Models," *Electronic Journal of Statistics*, 10, 1133–1154, [824]

Gonçalves, A. R., Das, P., Chatterjee, S., Sivakumar, V., Von Zuben, F. J., and Banerjee, A. (2014), "Multi-task Sparse Structure Learning," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 451–460. [824]

Guo, J., Levina, E., Michailidis, G., and Zhu, J. (2011), "Joint Estimation of Multiple Graphical Models," *Biometrika*, 98, 1–15. [823]

Hao, B., Sun, W. W., Liu, Y., and Cheng, G. (2018), "Simultaneous Clustering and Estimation of Heterogeneous Graphical Models," *Journal of Machine Learning Research*, 18, 1–58. [824]

Hara, S., and Washio, T. (2011), "Common Substructure Learning of Multiple Graphical Gaussian Models," in *Machine Learning and Knowledge Discovery in Databases*, eds. D. Gunopulos, T. Hofmann, D. Malerba, and M. Vazirgiannis, Berlin: Springer Berlin Heidelberg. [823]

Hartigan, J. A. and Wong, M. A. (1979), "Algorithm AS 136: A K-Means Clustering Algorithm," *Journal of the Royal Statistical Society*, Series C, 28, 100–108 [online]. Available at *http://www.jstor.org/stable/2346830*. [826]

Honorio, J., and Samaras, D. (2010), "Multi-Task Learning of Gaussian Graphical Models," in *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML–10)*, eds. Johannes Fürnkranz and Thorsten Joachims, Madison, WI: Omnipress. [823,824]

Huang, J. Z., Liu, N., Pourahmadi, M., and Liu, L. (2006), "Covariance Matrix Selection and Estimation Via Penalized Normal Likelihood," *Biometrika*, 93, 85–98. [825]

Jalali, P., Khare, K., and Michailidis, G. (2019), "A Bayesian Approach to Joint Estimation of Multiple Graphical Models." [online]. Available at *https://arxiv.org/abs/1902.03651*. [824]

Krishna, K. and Murty, M. N. (1999), "Genetic K-Means Algorithm," *IEEE Transactions on Systems, Man, and Cybernetics–PART B: Cybernetics*, 20, 433–439. [826]

Li, Y. and Liu, F. (2018), "Whiteout: Guassian Adaptive Noise Regularization in Deep Neural Networks." [online]. Available at *https://arxiv.org/abs/1612.01490*. [833]

Ma, J. and Michailidis, G. (2016), "Joint Structural Estimation of Multiple Graphical Models," *Journal of Machine Learning Research*, 17, 5777–5824, *http://dl.acm.org/citation.cfm?id=2946645.3053448*. [824]

McNicholas, P. and Murphy, T. (2008), "Parsimonious Gaussian Mixture Model," *Statistics and Computing*, 18, 285–296. [824]

Niculescu-Mizil, A. and Caruana, R. (2007), "Inductive Transfer for Bayesian Network Structure Learning," in *Artificial intelligence and statistics* (Vol. 27), eds. I. Guyon, G. Dror, V. Lemaire, G. Taylor, and D. Silver, pp. 167–181. [824]

Peterson, C., Stingo, F. C., and Vannucci, M. (2015), "Bayesian Inference of Multiple Gaussian Graphical Models," *Journal of the American Statistical Association*, 110, 159–174. [824]

Polson, N. G., Scott, J. G., and Willard, B. T. (2015), "Proximal Algorithms in Statistics and Machine Learning," *Statistical Science*, 30, 559–581. [827]

Pourahmadi, M. (2011), "Covariance Estimation: The GLM and Regularization Perspective," *Statistical Science*, 26, 369–387. [823]

Price, B. S., Geyer, C. J., and Rothman, A. J. (2015), "Ridge Fusion in Statistical Learning," *Journal of Computational and Graphical Statistics*, 24, 439–454, [824,825,831]

Price, B. S., and Sherwood, B. (2018), "A Cluster Elastic Net for Multivariate Regression," *Journal of Machine Learning Research*, 19, 1–37. [825,826]

Rolfs, B., Rajaratnam, B., Guillot, D., Wong, I., and Maleki, A. (2012), "Iterative Thresholding Algorithm for Sparse Inverse Covariance Estimation," in *Advances in Neural Information Processing Systems* (vol. 25), eds. F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Curran Associates, Inc., 1574–1582, *http://papers.nips.cc/paper/4574-iterative-thresholding-algorithm-for-sparse-inverse-covariance-estimation.pdf*. [827,828]

Rothman, A., Bickel, P., Levina, E., and Zhu, J. (2008), "Sparse Permutation Invariant Covariance Estimation," *Electronic Journal of Statistics*, 2, 494–515. [823]

Ruan, L., Yuan, M., and Zou, H. (2011), "Regularized Parameter Estimation in High-Dimensional Gaussian Mixture Models," *Neural Computation*, 23, 1605–1622. [824]

Saegusa, T. and Shojaie, A. (2016), "Joint Estimation of Precision Matrices in Heterogeneous Populations," *Electronic Journal of Statistics*, 10, 1341–1392, [824,829]

Shen, X., Pan, W., and Zhu, Y. (2012), "Likelihood-Based Selection and Sharp Parameter Estimation," *Journal of the American Statistical Association*, 107, 223–232, [824]

Witten, D., Friedman, J., and Simon, N. (2011), "New Insights and Faster Computations for the Graphical Lasso," *Journal of Computational and Graphical Statistics*, 20, 892–900. [823]

Witten, D. M., Shojaie, A., and Zhang, F. (2014), "The Cluster Elastic Net for High-Dimensional Regression With Unknown Variable Grouping," *Technometrics*, 56, 112–122. [825,826]

Xie, B., Pan, W., and Shen, X. (2008), "Penalized Model Based Clustering with Cluster Specific Diagonal Covariance Matrices and Grouped Variables," *Electronic Journal of Statistics*, 2, 168–212. [824]

Yuan, M., and Lin, Y. (2007), "Model Selection and Estimation in the Guassian Graphical Model," *Biometrika*, 94, 19–35. [823]

Zhou, H., Pan, W., and Shen, X. (2009), "Penalized Model-Based Clustering with Unconstrained Covariance Matrices," *Electronic Journal of Statistics*, 3, 1473–1496. [824]

Zhu, Y., Shen, X., and Pan, W. (2014), "Structural Pursuit Over Multiple Undirected Graphs," *Journal of the American Statistical Association*, 109, 1683–1696. [824]