Mathematical Programming manuscript No.
(will be inserted by the editor)

Certifiably Optimal Sparse Inverse Covariance Estimation

Dimitris Bertsimas · Jourdain Lamperski

Received: date / Accepted: date

Abstract We consider the maximum likelihood estimation of sparse inverse covariance matrices. We demonstrate that current heuristic approaches primarily encourage robustness, instead of the desired sparsity. We give a novel approach that solves the cardinality constrained likelihood problem to certifiable optimality. The approach uses techniques from mixed integer optimization and convex optimization, and provides a high-quality solution with a guarantee on its suboptimality, even if the algorithm is terminated early. Using a variety of synthetic and real datasets, we demonstrate that our approach can solve problems where the dimension of the inverse covariance matrix is in the 1000s. We also demonstrate that our approach produces significantly sparser solutions than Glasso and other popular learning procedures, while still maintaining desirable statistical properties.

Keywords Sparse Inverse Covariance Estimation · Cardinality Constraint Optimization · Mixed Integer Optimization · Glasso · Maximum Likelihood Estimation

1 Introduction

Estimating inverse covariance (precision) matrices is a fundamental task in modern multivariate analysis. Applications include undirected Gaussian graphical models [37], high dimensional discriminant analysis [19], portfolio allocation [25,26], complex data visualization [43], amongst many others, see [27]. For example, in the context of undirected Gaussian graphical models, estimating the precision matrix

Dimitris Bertsimas
Massachusetts Institute of Technology,
MIT Sloan School of Management and Operations Research Center,
Cambridge, MA.
E-mail: dbertsim@mit.edu

Jourdain Lamperski
Massachusetts Institute of Technology,
Operations Research Center,
Cambridge, MA.
E-mail: jourdain@mit.edu

corresponds to inferring the conditional independence structure on the related graphical model; zero entries in the precision matrix indicate that variables are conditionally independent.

It is often assumed that the underlying precision matrix is sparse [49,16,36, 23,41]. The reason for this is twofold.

1. There are often less samples than the dimension of the empirical covariance matrix. When this is the case, it is known that the empirical covariance matrix corresponding to the observations is singular, and thus the empirical covariance matrix does not accurately model the true covariance matrix. Moreover, the empirical covariance matrix can not be inverted to obtain an estimate of the true precision matrix. It is often assumed that the true precision matrix is sparse in this setting in order to recover the precision matrix.
2. In many applications (undirected graphical models, for example) it is desirable to obtain a simple model that has strong predictive power; that is, a model that is parsimonious. A natural assumption used to achieve this end is sparsity.

The most common method for encouraging sparsity in precision matrix estimation involves solving a ℓ_1-regularized maximum likelihood problem. The problem is convex and can be solved in high dimensions. Though this approach is tractable, solutions suffer from similar drawbacks as Lasso solutions in linear regression [14]. For example, one drawback is the ℓ_1-penalty introduces extra bias when estimating nonzero entries in the precision matrix with large absolute values [36].

In this paper, we seek to confront these drawbacks by solving the cardinality constrained optimization problem for which the ℓ_1-regularized problem is a convex surrogate. The cardinality constrained problem parallels the relation the best subset selection (or feature selection) problem plays in linear regression with Lasso. The main goal of this work is to solve the cardinality constrained problem for problem sizes of interest, and compare the solutions with current approaches. A summary of the contributions in this paper is given below.

1. Recent results establish that Lasso can be solved as a robust optimization problem for an appropriately chosen uncertainty set [6,50,10]. We generalize this result and show that the ℓ_1-regularized problem for estimating precision matrices can also be formulated as a robust optimization problem. This suggests that the ℓ_1-regularization approach is primarily encouraging robustness, not sparsity.
2. We formulate the cardinality constrained maximum likelihood problem for the inverse covariance matrix as a mixed-integer nonlinear optimization problem with a semidefinite constraint. An advantage of the approach over existing approaches is that it provides near optimal solutions fast, and a guarantee on the solutions suboptimality if the method is terminated early.
3. We propose a combination of generalized Benders decomposition (GBD), dynamic constraint generation, and first order methods to solve the mixed integer nonlinear problem. To our knowledge, this is the first time in which GBD is used to solve a mixed integer nonlinear optimization problem with semidefinite constraints. It is well-known that problems of this type are notoriously hard to solve, and we observe that our approach significantly outperforms available mixed integer nonlinear solvers.
4. We report computational results with both synthetic and real-world datasets that show that our proposed approach can deliver near optimal solutions in a

matter of seconds, and provably optimal solutions in a matter of minutes. In this manner, we solve problems of dimension in the 1000s.
5. We investigate statistical properties of solutions for the cardinality constrained problem. We compare solutions with ℓ_1-regularized estimates and other popular learning procedures, and observe that the cardinality constrained estimates produce sparser solutions. We also show the cardinality constrained estimates recover the sparsity pattern of the true underlying precision matrix with more accuracy.

The structure of the paper is as follows. In Section 2, we describe the problem of interest, provide a more detailed overview of relevant results, and give background on mixed integer optimization. We also discuss known drawbacks of current methods, and show that ℓ_1-regularization can be interpreted as primarily encouraging robustness, not sparsity. In Section 3, we provide a mixed integer formulation for the cardinality constrained problem and discuss how to find parameters the formulation requires. We also present an additional mixed integer formulation that we will use as a starting point for an algorithm for the cardinality constrained problem. In Section 4, we propose the algorithm and discuss additional considerations to enhance the overall method's performance. We perform a variety of computational tests in Section 5, and use synthetic and real datasets to asses the algorithmic and statistical performance of our approach. In Section 6, we provide concluding remarks.

2 Overview and Preliminaries

In this section, we provide a description of the problem formulation, an overview of current approaches for inducing sparsity in inverse covariance estimation, and background on mixed integer optimization. We conclude the section by showing that the ℓ_1-regularization approach is equivalent to solving a robust optimization problem with an appropriately chosen uncertainty set. This suggests that current approaches are primarily encouraging robustness, not sparsity.

2.1 Problem Description

Let $X \sim N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ with unknown mean $\boldsymbol{\mu} \in \mathbb{R}^p$ and covariance $\boldsymbol{\Sigma} \in S_{++}$, where S_{++} denotes the set of symmetric positive definite matrices in $\mathbb{R}^{p\times p}$. Given a random sample $X^{(1)}, ..., X^{(n)}$ of X, we seek to estimate the precision matrix $\boldsymbol{\Sigma}^{-1}$. Let $\overline{\boldsymbol{\Sigma}} \in \mathbb{R}^{p\times p}$ be the empirical covariance matrix corresponding to the n observations. The problem of interest is the following cardinality constrained negative log-likelihood problem

$$\min_{\boldsymbol{\Theta}\succ\mathbf{0}} \quad \langle \overline{\boldsymbol{\Sigma}}, \boldsymbol{\Theta}\rangle - \log\det\boldsymbol{\Theta} \quad \text{s.t.} \quad ||\boldsymbol{\Theta}||_0 \leq k, \tag{1}$$

where $k \in \mathbb{R}$, and $||\boldsymbol{\Theta}||_0 := \sum_{i>j} 1_{\Theta_{ij}\neq 0}$ counts the number of nonzero entries in the strictly lower triangular part of $\boldsymbol{\Theta}$. The expression $\langle\cdot,\cdot\rangle$ is the usual trace inner product $\langle \overline{\boldsymbol{\Sigma}}, \boldsymbol{\Theta}\rangle = \text{tr}(\overline{\boldsymbol{\Sigma}}^\top \boldsymbol{\Theta})$. The objective function in (1) is the Gaussian log-likelihood of the data [49].

Problem (1) parallels the role best subset selection plays in the context of linear regression. Like best subset selection, the cardinality constraint makes it computationally challenging. There is also the extra difficulty that the problem is a minimization over S_{++}. To our knowledge, the problem has yet to be considered in the literature as a discrete optimization problem over positive definite matrices. Thus, this paper provides the first exact optimization approach for solving problem (1).

At the core of our methodology is the exploitation of advances in discrete optimization. Recently best subset selection and other cardinality constrained problems have been solved with great success using discrete optimization [14,13,12]. These approaches exploit the significant advances in mixed integer optimization and motivate our approach. We provide a brief overview of these methods in Section 2.3.

2.2 Current Approaches

A variety of convex and nonlinear based optimization methods have been proposed to induce sparsity using the maximum likelihood problem [28]. Many of these methods can be interpreted as heuristics for problem (1). As mentioned, one of the most common heuristics involves solving the ℓ_1-regularized negative log-likelihood minimization

$$\min_{\mathbf{\Theta} \succ \mathbf{0}} \quad \langle \overline{\mathbf{\Sigma}}, \mathbf{\Theta} \rangle - \log\det \mathbf{\Theta} + \lambda ||\mathbf{\Theta}||_1, \tag{2}$$

where $||\mathbf{\Theta}||_1 := \sum_{ij} |\Theta_{ij}|$ is an ℓ_1-penalty term. Note that the diagonal elements of $\mathbf{\Theta}$ are usually excluded from the penalty term. In practice, it has been observed that the penalty term shrinks the coefficients of $\mathbf{\Theta}$ towards zero, and produces a sparse solution by setting many coefficients equal to zero. Problem (2) was originally motivated by the development and use of Lasso as a convex surrogate for the best subset selection problem [49]. The problem is well-studied in the literature [49,5,29,42]. A popular algorithm for solving the problem is a block coordinate descent procedure, which was originally proposed in [5]. A modified version is given in [29,39], where the subproblems in the block descent procedure are solved via Lasso. The modified algorithm is referred to as Graphical Lasso (Glasso), and can solve high dimensional problems of (2) in a short amount of time.

Though the problem is tractable, it shares in the statistical shortcomings of its motivator, Lasso. Problem (2) leads to biased estimates because the ℓ_1-norm penalty term penalizes large entries more than the smaller entries [36]. Accordingly, upon increasing the degree of regularization, (2) sets more entries of $\mathbf{\Theta}$ to zero, but leaves true predictors out of the active set. Thus, as soon as certain regularity conditions on the data are violated, Problem (2) becomes suboptimal as a variable selector, and in terms of delivering a model with good predictive performance. In contrast, problem (1) chooses variables to enter the active set without shrinking the entries in $\mathbf{\Theta}$. [36] discuss other statistical shortcomings of (2).

In the linear regression literature, the shortfalls of the Lasso heuristic have recently been explained by showing Lasso actually imposes robustness, not sparsity. In the next section, we show this result also holds for (2). We also present

empirical evidence in Section 5, which shows what can be achieved by (1), but cannot be achieved as well by (2).

To address these shortcomings, non-convex penalized methods are often used instead. Two popular approaches are smoothly clipped absolute deviation (SCAD) and minimax concave penalty (MCP), which are folded concave penalties that do not introduce extra bias for estimating nonzero entries with large absolute values. Theoretical properties of these methods are well studied [42,36]. One drawback of these methods is they do not provide a guarantee on how close their optimal solution is to the optimal solution of problem (1).

Estimators and approaches other than using maximum likelihood have also been proposed for inducing sparsity. Two such estimators are the constrained ℓ_1-minimization for inverse matrix estimation (CLIME) estimator and the graphical Dantzig selector [48,19]. These estimators differ from (1) in that they usually require a tuning parameter to control the bias-variance tradeoff. Rank and factor based methods have also been proposed; for a more complete survey of the different methods methods, see [28].

2.3 Mixed Integer Optimization Background

In this section, we provide a brief overview of relevant results and developments in mixed integer optimization (MIO). We first provide an overview of MIO problems that have attracted a significant amount of interest, and we then discuss relevant developments.

MIO Problems. The general form of a MIO problem is as follows:

$$\begin{array}{lll} \min & f(\alpha) & \\ \text{s.t.} & g_j(\alpha) \leq 0, & \forall j \in J, \\ & \alpha_i \in \{0,1\}, & \forall i \in I, \\ & \alpha_i \in \mathbb{R}, & \forall i \notin I, \end{array} \tag{3}$$

where I denotes the index set for the discrete variables, and J denotes the index set for the constraints.

Classes of MIO problems that have attracted significant amounts of interest include mixed integer linear optimization (MILO) problems (f and g_j are linear functions), mixed integer quadratic optimization (MIQO) problems (f is quadratic, g_j are linear functions), and mixed integer nonlinear optimization (MINLO) problems (f and g_j are continuously differentiable nonlinear functions). Note that when $I = \emptyset$, MILO problems reduce to linear optimization (LO) problems, MIQO problems reduce to quadratic optimization (QO) problems, and MINLO problems reduce to nonlinear optimization (NLO) problems. More recently, researchers have also considered a subcategory of MINLO problems that have conic constraints [4]. Problems of this form are often referred to in the literature as conic mixed integer optimization problems.

Developments. Among the types of MIO problems, MILO has seen the most advances. Over the past 25 years, the overall speedup of MILO solvers has been a factor of $780,000$, and of hardware has been a factor of $570,000$, leading to an

overall speedup of approximately 450 billion [14,13,11]. Most MILO solvers utilize branch and bound or branch and cut algorithms together with other techniques [15].

Convex MINLO is also well-studied and has been an active area of research since the 1970s. Algorithms for this type of problem combine techniques from nonlinear, discrete, and linear optimization [17]. Among the most popular algorithms are branch and bound and outer approximation. See [17] for a survey of these and other techniques.

However, despite the progress in the aforementioned fields, many types of conic MILO problems are still in their infancy [4]. On one hand, problems with second order cone constraints can be solved relatively efficiently using a branch and bound scheme, but on the other hand, problems with semidefinite constraints still remain computationally very challenging. Conic MILO problems were first solved using a basic branch and bound method that solves conic relaxations of the problem at each node of a branch and bound tree. More recently, branch and cut methods have been proposed in light of their success in MILO. [4] reports that these methods improve algorithm runtime for problems with semidefinite and second order cone constraints [4]. Usually the cuts are added in a branch and bound or outer approximation scheme [8]. Even though these methods yield speedups, they do not alleviate the challenge of solving the underlying conic problem, which generally cannot be solved as fast as a linear optimization problem, especially when the problem has semidefinite constraints.

In Section 3, we consider a MINLO formulation, with a semidefinite constraint and nonlinear objective, for the cardinality constrained problem. Given the objective is nonlinear, and even MILO with semidefinite constraints is in its infancy, a branch and bound method is currently not a viable approach for problems of even modest size. Instead, in Section 4, we present an alternative approach that exploits the advances in MILO and first order methods. The method avoids the challenges presented by current branch and bound approaches, which would require solving an underlying semidefinite optimization problem.

2.4 Robust Interpretation

In this section, we show the ℓ_1-regularized problem can be interpreted as a robust optimization problem with an appropriately chosen uncertainty set. For a survey of robust optimization, see [9]. The relationship between robustness and regularization is well-studied in a variety of statistical estimation problems. See [10] for an extensive survey. A relevant result is that Lasso can be solved as a robust optimization problem, which suggests Lasso primarily encourages robustness, not sparsity [6,50,10]. In this section, we generalize this result for problem (2).

We first recall some definitions. For $\mathbf{\Theta} \in \mathbb{R}^{p\times p}$, let

$$||\mathbf{\Theta}||_F := \left(\sum_{i,j} \Theta_{ij}^2\right)^{\frac{1}{2}}$$

be the Frobenius norm on $\mathbb{R}^{p\times p}$ and let

$$(\mathrm{sgn}(\boldsymbol{\Theta}))_{ij} := \begin{cases} -1, & \Theta_{ij} < 0, \\ 0, & \Theta_{ij} = 0, \\ 1, & \Theta_{ij} > 0. \end{cases}$$

The following theorem shows that the ℓ_1-regularized problem can be solved as an equivalent robust optimization problem with uncertainty set $\mathcal{U} = \{\boldsymbol{\Delta} \in \mathbb{R}^{p\times p} : ||\boldsymbol{\Delta}||_F \leq \lambda\}$.

Theorem 1 *Let $\overline{\boldsymbol{\Sigma}} \in \mathbb{R}^{p\times p}$. Then,*

$$\min_{\boldsymbol{\Theta}\succ\mathbf{0}} \max_{\boldsymbol{\Delta}\in\mathcal{U}} \left\{\langle \boldsymbol{\Theta}, \overline{\boldsymbol{\Sigma}} + \boldsymbol{\Delta}\rangle - \log\det\boldsymbol{\Theta}\right\} = \min_{\boldsymbol{\Theta}\succ\mathbf{0}} \left\{\langle \boldsymbol{\Theta}, \overline{\boldsymbol{\Sigma}}\rangle - \log\det\boldsymbol{\Theta} + \lambda||\boldsymbol{\Theta}||_1\right\},$$

where $\mathcal{U} = \{\boldsymbol{\Delta} \in \mathbb{R}^{p\times p} : ||\boldsymbol{\Delta}||_F \leq \lambda\}$.

Proof Let $\hat{\boldsymbol{\Theta}}$ be a symmetric positive definite matrix. We first show

$$\max_{\boldsymbol{\Delta}\in\mathcal{U}} \left\{\langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}} + \boldsymbol{\Delta}\rangle - \log\det\hat{\boldsymbol{\Theta}}\right\} = \langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}}\rangle - \log\det\hat{\boldsymbol{\Theta}} + \lambda||\hat{\boldsymbol{\Theta}}||_1. \tag{4}$$

Let $\boldsymbol{\Delta}^* \in \mathcal{U}$ be optimal for the left hand side of (4). Then,

$$\begin{aligned} &\max_{\boldsymbol{\Delta}\in\mathcal{U}} \left\{\langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}} + \boldsymbol{\Delta}\rangle - \log\det\hat{\boldsymbol{\Theta}}\right\} \\ &= \langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}}\rangle - \log\det\hat{\boldsymbol{\Theta}} + \langle \boldsymbol{\Delta}^*, \hat{\boldsymbol{\Theta}}\rangle \\ &\leq \langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}}\rangle - \log\det\hat{\boldsymbol{\Theta}} + ||\boldsymbol{\Delta}^*||_F||\hat{\boldsymbol{\Theta}}||_F \\ &\leq \langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}}\rangle - \log\det\hat{\boldsymbol{\Theta}} + \lambda||\hat{\boldsymbol{\Theta}}||_1, \end{aligned} \tag{5}$$

where the first inequality follows from the Cauchy-Schwarz inequality, and the second follows from the fact that $\boldsymbol{\Delta}^* \in \mathcal{U}$ and $||\hat{\boldsymbol{\Theta}}||_F \leq ||\hat{\boldsymbol{\Theta}}||_1$.

Define $\hat{\boldsymbol{\Delta}} := \lambda\,\mathrm{sgn}(\hat{\boldsymbol{\Theta}})$. Note that $\hat{\boldsymbol{\Delta}} \in \mathcal{U}$. Thus,

$$\begin{aligned} &\max_{\boldsymbol{\Delta}\in\mathcal{U}} \left\{\langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}} + \boldsymbol{\Delta}\rangle - \log\det\hat{\boldsymbol{\Theta}}\right\} \\ &\geq \langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}}\rangle - \log\det\hat{\boldsymbol{\Theta}} + \langle \hat{\boldsymbol{\Delta}}, \hat{\boldsymbol{\Theta}}\rangle \\ &= \langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}}\rangle - \log\det\hat{\boldsymbol{\Theta}} + \lambda\langle \mathrm{sgn}(\hat{\boldsymbol{\Theta}}), \hat{\boldsymbol{\Theta}}\rangle \\ &= \langle \hat{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Sigma}}\rangle - \log\det\hat{\boldsymbol{\Theta}} + \lambda||\boldsymbol{\Theta}||_1 \end{aligned} \tag{6}$$

where the inequality follows from the fact that $\hat{\boldsymbol{\Delta}} \in \mathcal{U}$. Thus, inequalities (5) and (6) establish equation (4). Minimizing over both sides of (4) yields the desired result.

Thus, problem (2) primarily encourages robustness, not sparsity. Theorem 1 provides motivation for solving the cardinality constrained problem. In the next section, we present MIO formulations of the cardinality constrained problem, which we will use for solving the problem.

3 Mixed Integer Optimization Formulations

We formulate problem (1) as a MIO problem. In Section 3.1, we provide a simple mixed integer optimization formulation. In Section 3.2, we discuss how to find parameters that the formulation requires. Lastly, in Section 3.3, we present an alternative mixed integer formulation that will serve as the foundation for the algorithm developed in Section 4.

3.1 Problem Formulation

We formulate the cardinality constrained problem (1) as the following MINLO problem with a semidefinite constraint,

$$\begin{array}{ll} \min\limits_{\boldsymbol{\Theta}\succ\mathbf{0}} & \langle\overline{\boldsymbol{\Sigma}},\boldsymbol{\Theta}\rangle-\log\det\boldsymbol{\Theta} \\ \text{s.t.} & -M_{ij}z_{ij}\leq\Theta_{ij}\leq M_{ij}z_{ij},\quad\forall i>j \\ & \sum\limits_{i>j}z_{ij}\leq k, \\ & z_{ij}\in\{0,1\},\quad\forall i>j. \end{array} \tag{7}$$

where $\mathbf{z}$ is a vector of binary variables, and for all $i>j$, $M_{ij}\in\mathbb{R}_+$ are constants chosen such that if $\boldsymbol{\Theta}^*$ is a minimizer for problem (1), then $|\Theta^*_{ij}|\leq M_{ij}z_{ij}$. Define $N:=\binom{p}{2}$ to be the number of off-diagonal elements in the lower triangular portion of $\boldsymbol{\Theta}$, which is also equal to the number of binary decision variables. We write the bounds together as a vector $\mathbf{M}\in\mathbb{R}^N$. We now include some remarks about important properties of the formulation.

1. If $z_{ij}=1$, then $|\Theta_{ij}|\leq M_{ij}$, and if $z_{ij}=0$, then $\Theta_{ij}=0$. So, $\sum_{i>j}z_{ij}$ counts the number of non-zeros in the off-diagonal of $\boldsymbol{\Theta}$. Thus, provided $M_{ij}\in\mathbb{R}$ is chosen sufficiently large (that is, $M_{ij}\geq|\Theta^*_{ij}|$) for all $i>j$, the optimal solutions for (1) and (7) are the same.
2. Constants $\mathbf{M}$ are not known a priori, and a small value of M_{ij} may lead to a different solution than problem (1). The choice of bounds also affects the strength of the formulation, and thus it is critical to find appropriate values for $\mathbf{M}$. In Section 3.2, we discuss in detail how we choose these constants.

3.2 Optimal Bounds on $\mathbf{M}$

In this section, we present a method for obtaining constants $\mathbf{M}$ for the MIO formulation (7). The approach involves solving two optimization problems for each off-diagonal entry of the matrix being estimated. The problems provide lower and upper bounds for each entry of the optimal solution. First we present the problems, then we discuss how they are solved.

Bound Optimization Problems. Let $\hat{\boldsymbol{\Theta}}$ be a feasible solution for (1) and define,

$$u:=\langle\hat{\boldsymbol{\Theta}},\overline{\boldsymbol{\Sigma}}\rangle-\log\det\hat{\boldsymbol{\Theta}}.$$

A simple way to obtain lower bounds for the ijth entry of the optimal solution is to solve,

$$\begin{array}{ll} \min\limits_{\boldsymbol{\Theta} \succ \mathbf{0}} & \Theta_{ij} \\ \text{s.t.} & \langle \overline{\boldsymbol{\Sigma}}, \boldsymbol{\Theta} \rangle - \log\det \boldsymbol{\Theta} \leq u \end{array} \tag{8}$$

Likewise, to obtain upper bounds we solve,

$$\begin{array}{ll} \max\limits_{\boldsymbol{\Theta} \succ \mathbf{0}} & \Theta_{ij} \\ \text{s.t.} & \langle \overline{\boldsymbol{\Sigma}}, \boldsymbol{\Theta} \rangle - \log\det \boldsymbol{\Theta} \leq u \end{array} \tag{9}$$

Note that it is sufficient to find a feasible solution $\hat{\boldsymbol{\Theta}}$ to formulate (8) and (9), and a feasible solution with a smaller value leads to better bounds. We provide a method for obtaining such an initial feasible solution in Section 4.3.

Solution Approach. We first derive the dual of problem (8). The dual for problem (9) is similar and we therefore omit its derivation and presentation. Note an advantage for considering the dual is we do not need to solve the problem to optimality to obtain a valid bound.

We derive the dual using basic arguments from convex duality theory [18]. For $\lambda > 0$, the Lagrangian dual function for (8) is given by,

$$g(\lambda) = \min_{\boldsymbol{\Theta} \succ 0} \left\{ \frac{1}{2} \langle \boldsymbol{\Theta}, \mathbf{E}_{ij} \rangle + \lambda \left(\langle \boldsymbol{\Theta}, \overline{\boldsymbol{\Sigma}} \rangle - \log\det \boldsymbol{\Theta} - u \right) \right\}, \tag{10}$$

where $\mathbf{E}_{ij} \in \mathbb{R}^{p \times p}$ is a matrix of zeros, except for the ijth and jith entries, which are set equal to one. Define function $F : S_{++} \to \mathbb{R}$ as

$$F(\boldsymbol{\Theta}) := \frac{1}{2} \langle \boldsymbol{\Theta}, \mathbf{E}_{ij} \rangle + \lambda \left(\langle \boldsymbol{\Theta}, \overline{\boldsymbol{\Sigma}} \rangle - \log\det \boldsymbol{\Theta} - u \right),$$

which is the term inside the minimization in (10). Then, the gradient of F evaluated at $\boldsymbol{\Theta}$ is

$$\nabla F(\boldsymbol{\Theta}) = \frac{1}{2} \mathbf{E}_{ij} + \lambda \overline{\boldsymbol{\Sigma}} - \lambda \boldsymbol{\Theta}^{-1} \tag{11}$$

Setting (11) equal to zero yields $\boldsymbol{\Theta} = (\frac{1}{2\lambda} \mathbf{E}_{ij} + \overline{\boldsymbol{\Sigma}})^{-1}$. Substituting into (10) gives,

$$g(\lambda) = \lambda \left(p - u + \log\det \left(\frac{1}{2\lambda} \mathbf{E}_{ij} + \overline{\boldsymbol{\Sigma}} \right) \right), \tag{12}$$

which leads to the dual problem for problem (8),

$$\max_{\lambda > 0} \left\{ \lambda \left(p - u + \log\det \left(\frac{1}{2\lambda} \mathbf{E}_{ij} + \overline{\boldsymbol{\Sigma}} \right) \right) \right\} \tag{13}$$

Computationally, problem (13) is very easy to solve because it is a convex optimization problem, where the objective function is a real function. We solve it by applying the Newton's method, and accordingly we next derive the gradient of the objective of (13).

Define function $g : \mathbb{R} \to \mathbb{R}$ as

$$g(\lambda) = \lambda \left(p - u + \log\det\left(\frac{1}{2\lambda}\mathbf{E}_{ij} + \overline{\mathbf{\Sigma}} \right) \right),$$

which is the objective function in (13). The derivative of g is given by

$$\begin{aligned} g'(\lambda) &= p - u + \log\det\left(\frac{1}{2\lambda}\mathbf{E}_{ij} + \overline{\mathbf{\Sigma}} \right) + \lambda \left[\log\det\left(\frac{1}{2\lambda}\mathbf{E}_{ij} + \overline{\mathbf{\Sigma}} \right) \right]' \\ &= p - u + \log\det\left(\frac{1}{2\lambda}\mathbf{E}_{ij} + \overline{\mathbf{\Sigma}} \right) - \frac{1}{\lambda}\left(\frac{1}{2\lambda}\mathbf{E}_{ij} + \overline{\mathbf{\Sigma}} \right)^{-1}_{ij}, \end{aligned} \tag{14}$$

where the first equality follows from the chain rule, and the second equality follows from the well-known fact that $(\det \mathbf{A}(\lambda))' = (\det \mathbf{A}(\lambda))\langle \mathbf{A}^{-1}(\lambda), \mathbf{A}'(\lambda)\rangle$, where $\mathbf{A}$ is a matrix function of λ.

Thus, we use equation (14) in a Newton's method to compute constants $\mathbf{M}$ for the mixed integer formulation. This method is clearly significantly faster than solving for bounds with (8), which would require solving a semidefinite optimization problem. This is critical, as when p is large, it is necessary that bounds for each entry are found fast because there are $N = \binom{p}{2}$ entries to bound.

3.3 Alternative Formulation

In this section, we provide an alternative formulation for the mixed integer formulation (7) presented in Section 3.1. The formulation plays a foundational role in the algorithm developed in Section 4.

Alternative Formulation. Let $P = \{\mathbf{z} \in \{0,1\}^N : \sum_{i>j} z_{ij} \leq k\}$ be the set of feasible binary vectors for the MIO formulation (7). Define function $h : P \to \mathbb{R}$ for $\mathbf{z} \in P$ by,

$$\begin{array}{rcll} h(\mathbf{z}) & := & \min\limits_{\mathbf{\Theta} \succ \mathbf{0}} & \langle \overline{\mathbf{\Sigma}}, \mathbf{\Theta} \rangle - \log\det \mathbf{\Theta} \\ & & \text{s.t.} & -M_{ij} z_{ij} \leq \Theta_{ij} \leq M_{ij} z_{ij}, \quad \forall i > j. \end{array} \tag{15}$$

The value $h(\hat{\mathbf{z}})$ for $\hat{\mathbf{z}} \in P$ is the optimal value of (7) when the binary decision variable $\mathbf{z}$ is set equal to $\hat{\mathbf{z}}$, and the resulting optimization problem is solved for $\mathbf{\Theta}$. Thus, we can write (7) as,

$$\min_{\mathbf{z} \in P} \; h(\mathbf{z}). \tag{16}$$

Note that the inner minimization problem in (16) (namely, problem (15)) is convex because the objective is convex and the constraints are linear. The inner minimization also is feasible and satisfies Slater's conditions as the identity matrix $\mathbf{I} \succ \mathbf{0}$ is feasible. Thus, strong duality holds [18] and we can reformulate (15) using Lagrange multipliers $\mathbf{r}_u, \mathbf{r}_l \in \mathbb{R}^N$ as,

$$h(\mathbf{z}) = \max_{\mathbf{r}_u, \mathbf{r}_l \geq \mathbf{0}} \min_{\mathbf{\Theta} \succ \mathbf{0}} \; v(\mathbf{\Theta}, \mathbf{r}_u, \mathbf{r}_l, \mathbf{z}), \tag{17}$$

where the function v is the Lagrangian of (15) defined by,

$$\begin{aligned} v(\mathbf{\Theta}, \mathbf{r}_u, \mathbf{r}_l, \mathbf{z}) = \langle \overline{\mathbf{\Sigma}}, \mathbf{\Theta} \rangle - \log\det \mathbf{\Theta} \\ + 2\mathbf{r}_u^\top (\mathbf{\Theta}_L - \mathbf{M} \circ \mathbf{z}) - 2\mathbf{r}_l^\top (\mathbf{\Theta}_L + \mathbf{M} \circ \mathbf{z}), \end{aligned} \tag{18}$$

where $\mathbf{\Theta}_L$ denotes the strictly lower triangular portion of $\mathbf{\Theta}$ as a vector in $\mathbb{R}^N$, and the operator $\circ$ in (18) denotes the Hadamard product $(\mathbf{M} \circ \mathbf{z})_{ij} = M_{ij} z_{ij}$. Thus, from (16) and (17), we express the optimization problem (7) as,

$$\min_{\mathbf{z} \in P} \max_{\mathbf{r}_u, \mathbf{r}_l \geq \mathbf{0}} \min_{\mathbf{\Theta} \succ \mathbf{0}} v(\mathbf{\Theta}, \mathbf{r}_u, \mathbf{r}_l, \mathbf{z}),$$

which can further be expressed as the following MIO problem,

$$\begin{aligned} \min_{\mathbf{z}, \eta} \quad & \eta \\ s.t. \quad & \eta \geq \min_{\mathbf{\Theta} \succ \mathbf{0}} v(\mathbf{\Theta}, \mathbf{r}_u, \mathbf{r}_l, \mathbf{z}) \text{ for all } \mathbf{r}_u, \mathbf{r}_l \geq 0 \\ & \mathbf{z} \in P. \end{aligned} \tag{19}$$

Problem (19) is a MIO problem with infinitely many constraints. We use the formulation in the next section to develop an algorithm for solving the cardinality constrained problem.

4 Optimization Approach

In this section, we present an optimization approach for solving the MIO problem (7). In Section 4.1, we provide motivation for the approach by presenting a method that efficiently solves the separation problem of (19); the method we propose forms the foundation of the optimization approach and therein provides key insights. We then discuss how we use the separation problem in a cutting-planes (constraint generation) algorithm that solves MIO formulation (19) exactly. The cutting-planes method is known in the optimization literature as generalized Benders decomposition (GBD) [30]. In Section 4.2, we state the algorithm in the context of problem (19) and discuss some of its properties. In Section 4.3 and 4.4, we discuss additional considerations for improving the algorithm's overall performance.

4.1 Seperation Problem

We consider the separation problem for (19), which involves the following two steps.

1. Given $(\hat{\mathbf{z}}, \hat{\eta}) \in P \times \mathbb{R}$, decide if $(\hat{\mathbf{z}}, \hat{\eta})$ is feasible for (19).
2. If it is not feasible, provide a vector $(\pi, \pi_0) \in \mathbb{R}^{N+1} \times \mathbb{R}$ such that $\pi^\top (\mathbf{z}, \eta) \leq \pi_0$ is satisfied by all feasible points $(\mathbf{z}, \eta) \in P \times \mathbb{R}$ for (1), but violated by $(\hat{\mathbf{z}}, \hat{\eta})$, i.e., $\pi^\top (\hat{\mathbf{z}}, \hat{\eta}) > \pi_0$.

Thus, the first part of the separation problem corresponds to deciding if $(\hat{\mathbf{z}}, \hat{\eta}) \in P \times \mathbb{R}$ is feasible, and the second part corresponds to finding a hyperplane that separates $(\hat{\mathbf{z}}, \hat{\eta})$ from the feasible region if it is not. We will first provide an efficient approach for solving the separation problem, and then we will discuss how we use our approach in a cutting-planes method that solves the MIO formulation (19).

Solution Approach: Part 1. We first consider Part 1 of the separation problem. Let $(\hat{\mathbf{z}}, \hat{\eta}) \in P \times \mathbb{R}$. From MIO formulation (19), $(\hat{\mathbf{z}}, \hat{\eta})$ is feasible if and only if

$$\hat{\eta} \geq \max_{\mathbf{r}_u, \mathbf{r}_l \geq \mathbf{0}} \min_{\boldsymbol{\Theta} \succ \mathbf{0}} v(\boldsymbol{\Theta}, \mathbf{r}_u, \mathbf{r}_l, \hat{\mathbf{z}}). \tag{20}$$

Thus, we can decide feasibility by solving the optimization problem on the right hand side of (20). From (15) and (17), we can alternatively formulate the optimization problem in (20) as,

$$\begin{array}{ll} \min\limits_{\boldsymbol{\Theta} \succ \mathbf{0}} & \langle \overline{\boldsymbol{\Sigma}}, \boldsymbol{\Theta} \rangle - \log\det \boldsymbol{\Theta} \\ \text{s.t.} & -M_{ij}\hat{z}_{ij} \leq \Theta_{ij} \leq M_{ij}\hat{z}_{ij}, \quad \forall i > j. \end{array} \tag{21}$$

Note that in addition to providing a certificate of feasibility (or infeasibility) for $(\hat{\mathbf{z}}, \hat{\eta})$ for (19), the optimal solution of problem (21) also serves as a feasible positive definite matrix for (7), and thus an upper bound on the optimal value of the problem.

We now write problem (21) in a more common form. Define $\hat{S} := \{(i,j) : \hat{z}_{ij} \neq 0 \text{ or } i = j\}$, which we will refer to as the sparsity pattern for problem (21). The sparsity pattern is the set of indices that correspond to the non-zero entries of $\boldsymbol{\Theta}$ in (21). Assuming constants $\mathbf{M}$ are appropriately chosen (see Section 3.2), we can formulate (21) as

$$\begin{array}{ll} \min\limits_{\boldsymbol{\Theta} \succ 0} & \langle \overline{\boldsymbol{\Sigma}}, \boldsymbol{\Theta} \rangle - \log\det \boldsymbol{\Theta} \\ \text{s.t.} & \Theta_{ij} = 0, \quad (i,j) \notin \hat{S}. \end{array} \tag{22}$$

Note that problem (22) does not require knowing constants $\mathbf{M}$. Problem (22) is a covariance selection problem [22], which is a well-studied problem in the literature, and can be efficiently solved [21]. Thus, using (22), we can efficiently decide if $(\hat{\mathbf{z}}, \eta)$ is feasible for (19). In Section 4.3, we discuss more details of how the problem can be solved using a first order method and techniques introduced in [21].

Solution Approach: Part 2. We now discuss Part 2 of the separation problem. Suppose that $(\hat{\mathbf{z}}, \hat{\eta})$ is not feasible for problem (19). We seek to provide a hyperplane that separates $(\hat{\mathbf{z}}, \eta)$ from the feasible region of problem (19). We construct the hyperplane using optimal solutions for (21) and its dual; accordingly, we begin by deriving the dual of (21) and discussing how to find the dual optimal solutions given a primal optimal solution. We then construct the desired hyperplane from these solutions.

We derive the dual of (21) using basic convex optimization techniques. Recall the Lagrangian of (21) is given by equation (18). The gradient of the Lagrangian with respect to $\boldsymbol{\Theta}$ is given by,

$$\nabla v(\boldsymbol{\Theta}, \mathbf{r}_u, \mathbf{r}_l, \hat{\mathbf{z}}) = \overline{\boldsymbol{\Sigma}} - \boldsymbol{\Theta}^{-1} + \mathbf{R}_u - \mathbf{R}_l,$$

where $\mathbf{R}_u, \mathbf{R}_l \in \mathbb{R}^{p\times p}$ have off-diagonal entries equal to $\mathbf{r}_u, \mathbf{r}_l \geq \mathbf{0}$ and diagonal equal to the zero vector. Setting the gradient to zero yields

$$\mathbf{\Theta} = (\overline{\mathbf{\Sigma}} + \mathbf{R}_u - \mathbf{R}_l)^{-1}. \tag{23}$$

Substituting (23) in (18) yields the following dual problem,

$$\max_{\mathbf{r}_u, \mathbf{r}_l \geq 0} \quad p + \log\det(\overline{\mathbf{\Sigma}} + \mathbf{R}_u - \mathbf{R}_l) - 2(\mathbf{r}_u + \mathbf{r}_l)^\top (\mathbf{M} \circ \hat{\mathbf{z}}). \tag{24}$$

We are now ready to construct the hyperplane. We begin by solving (21) using formulation (22). Let $\hat{\mathbf{\Theta}}$ be an optimal solution. We next seek to compute the corresponding optimal solutions of the dual (24); we provide analytic formulas for the optimal dual solutions in terms of $\hat{\mathbf{\Theta}}$ and $\overline{\mathbf{\Sigma}}$ in Proposition 1 below. For a vector $\mathbf{x} \in \mathbb{R}^N$, we introduce the following notation,

$$[\mathbf{x}]_i^+ := \begin{cases} x_i, & x_i > 0, \\ 0, & x_i \leq 0. \end{cases}$$

Proposition 1 *Let $\mathbf{\Theta} \succ \mathbf{0}$ be an optimal solution for (22). Then,*

$$\mathbf{r}_u := [(\mathbf{\Theta}^{-1} - \overline{\mathbf{\Sigma}})_L]^+ \quad \textit{and} \quad \mathbf{r}_l := [-(\mathbf{\Theta}^{-1} - \overline{\mathbf{\Sigma}})_L]^+, \tag{25}$$

are optimal for the dual problem (24), where $(\mathbf{\Theta}^{-1} - \overline{\mathbf{\Sigma}})_L$ denotes the strictly lower triangular portion of the matrix $\mathbf{\Theta}^{-1} - \overline{\mathbf{\Sigma}}$ as a vector in $\mathbb{R}^N$.

Proof Let $\mathbf{\Theta}$ be optimal for (22). Then, the stationary Karush-Kuhn-Tucker (KKT) condition ensures that,

$$\overline{\mathbf{\Sigma}} - \mathbf{\Theta}^{-1} + \mathbf{R}_u - \mathbf{R}_l = \mathbf{0}, \tag{26}$$

where $\mathbf{r}_u, \mathbf{r}_l \geq 0$ are optimal for problem (24). Because the dual is a maximization problem and $\mathbf{r}_u, \mathbf{r}_l \geq 0$, at most one of $(r_l)_{ij}$ and $(r_u)_{ij}$ are nonzero for $i > j$. This remark together with equation (26) implies equations (25).

Thus, $\hat{\mathbf{r}}_u = [(\hat{\mathbf{\Theta}}^{-1} - \overline{\mathbf{\Sigma}})_L]^+$ and $\hat{\mathbf{r}}_l = [(\overline{\mathbf{\Sigma}} - \hat{\mathbf{\Theta}}^{-1})_L]^+$. We now construct the following hyperplane,

$$\eta - v(\hat{\mathbf{\Theta}}, \hat{\mathbf{r}}_u, \hat{\mathbf{r}}_l, \mathbf{z}) = 0, \tag{27}$$

where $v(\cdot)$ is defined in Eq. (18). From (18), we see that (27) is indeed a hyperplane as $\hat{\mathbf{\Theta}}$, $\hat{\mathbf{r}}_u$, and $\hat{\mathbf{r}}_l$ are known. The next theorem establishes that (27) is a separating hyperplane for $(\hat{\mathbf{z}}, \hat{\eta})$ with respect to (19).

Theorem 2 *Suppose $(\hat{\mathbf{z}}, \hat{\eta}) \in P \times \mathbb{R}$ is not feasible for (19). Then,*

$$\eta - v(\hat{\mathbf{\Theta}}, \hat{\mathbf{r}}_u, \hat{\mathbf{r}}_l, \mathbf{z}) = 0,$$

as defined in (27), is a separating hyperplane for $(\hat{\mathbf{z}}, \hat{\eta})$.

Proof First we establish that $\hat{\eta} - v(\hat{\mathbf{\Theta}}, \hat{\mathbf{r}}_u, \hat{\mathbf{r}}_l, \hat{\mathbf{z}}) < 0$. Because $(\hat{\mathbf{z}}, \hat{\eta})$ is not feasible,

$$\eta < \max_{\mathbf{r}_u, \mathbf{r}_l \geq 0} \min_{\mathbf{\Theta} \succ \mathbf{0}} v(\mathbf{\Theta}, \mathbf{r}_u, \mathbf{r}_l, \hat{\mathbf{z}}) = v(\hat{\mathbf{\Theta}}, \hat{\mathbf{r}}_u, \hat{\mathbf{r}}_l, \hat{\mathbf{z}}),$$

where the equality follows from the definition of $\hat{\mathbf{\Theta}}$, $\hat{\mathbf{r}}_u$, and $\hat{\mathbf{r}}_l$. Thus,

$$\hat{\eta} - v(\hat{\mathbf{\Theta}}, \hat{\mathbf{r}}_u, \hat{\mathbf{r}}_l, \hat{\mathbf{z}}) < 0.$$

Let $(\mathbf{z}, \eta) \in P \times \mathbb{R}$ be feasible for (19). We show that $\eta - v(\hat{\mathbf{\Theta}}, \hat{\mathbf{r}}_u, \hat{\mathbf{r}}_l, \mathbf{z}) \geq 0$. We observe that,

$$\begin{aligned}
\eta &\geq \max_{\mathbf{r}_u, \mathbf{r}_l \geq 0} \min_{\mathbf{\Theta} \succ \mathbf{0}} v(\mathbf{\Theta}, \mathbf{r}_u, \mathbf{r}_l, \mathbf{z}) \\
&\geq \min_{\mathbf{\Theta} \succ \mathbf{0}} v(\mathbf{\Theta}, \hat{\mathbf{r}}_u, \hat{\mathbf{r}}_l, \hat{\mathbf{z}}_0) \\
&= v(\hat{\mathbf{\Theta}}, \hat{\mathbf{r}}_u, \hat{\mathbf{r}}_l, \hat{\mathbf{z}}_0),
\end{aligned}$$

where the second inequality follows from the fact that $\hat{\mathbf{r}}_u$ and $\hat{\mathbf{r}}_l$ are feasible for the outer maximization problem, and the equality follows by taking the gradient of the objective in the minimization problem. Thus, $\hat{\eta} - v(\hat{\mathbf{\Theta}}, \hat{\mathbf{r}}_u, \hat{\mathbf{r}}_l, \hat{\mathbf{z}}) \geq 0$.

Thus, (27) is a separating hyperplane for $(\hat{\mathbf{z}}, \hat{\eta})$ and we have established a method for solving the separation problem. Namely, given $(\hat{\mathbf{z}}, \hat{\eta}) \in P \times \mathbb{R}$, we can solve (22) for $\hat{\mathbf{\Theta}}$, $\hat{\mathbf{r}}_u$, and $\hat{\mathbf{r}}_l$, and check if the optimal value of the problem is less than $\hat{\eta}$. If so, the solution is feasible. If not, the solution is not feasible and we can construct a separating hyperplane from the primal and dual solutions. We next discuss how we incorporate this result in a cutting-planes method that solves (19).

Cutting-planes Method. We now provide a brief overview of how we use the separation problem to solve (19) via a cutting-planes method. As noted above, the method is referred to as generalized Benders decomposition (GBD) in the literature [30]. We discuss further connections to the literature in the next section.

We start by solving a relaxed version of the MIO formulation (19) that has one constraint. We obtain an optimal solution $(\mathbf{z}, \eta)$, which gives us a lower bound η for (19). We then solve the corresponding covariance selection problem (22) for a primal optimal solution $\mathbf{\Theta}$ and dual optimal solutions $\mathbf{r}_u$ and $\mathbf{r}_l$. We then test if $(\mathbf{z}, \eta)$ is feasible for (19). If so, we have found an optimal solution. If not, we have found a feasible solution $\mathbf{\Theta}$ for (1) and a corresponding upper bound. We then construct a separating hyperplane from $\mathbf{\Theta}$, $\mathbf{r}_u$, and $\mathbf{r}_l$ and include it in the relaxed formulation of (19). The solution $(\mathbf{z}, \eta)$ is then excluded from the feasible region of the new relaxed problem. We then solve the new relaxed formulation of (19), and iterate the above steps.

Note that iteratively resolving the relaxed version of (7) gives a sequence of nondecreasing lower bounds. Iteratively solving covariance selection problems (22) gives a sequence of upper bounds, not necessarily decreasing or non-increasing, and so we keep track of the smallest such upper bound. We continue this procedure until we find an optimal solution, or the gap between the smallest found upper bound and the current lower bound satisfies a given tolerance threshold.

In the next section, we provide a more formal description of the method and discuss convergence and other properties.

4.2 Algorithm

In this section, we state the generalized Benders decomposition algorithm for problem (19). First we present the method, and then we discuss its properties. We also provide further connections to the literature.

GBD for Cardinality Constrained Problem
Input: $\mathbf{z}^{(1)} \in P$, covariance matrix $\overline{\mathbf{\Sigma}}$, sparsity parameter k for (1), and tolerance ϵ.
Output: $\mathbf{\Theta}_{GBD} \succ 0$ feasible solution, $\eta^{(t)}$ lower bound, and μ upper bound for (1).

Set $t = 1$, $\eta^{(t)} = -\infty$, and $\mu = \infty$.

1. Let $S^{(t)} = \{(i,j) : z_{ij}^{(t)} \neq 0 \text{ or } i = j\}$. Solve the covariance selection problem,

$$\begin{array}{ll} \min\limits_{\mathbf{\Theta} \succ 0} & \langle \overline{\mathbf{\Sigma}}, \mathbf{\Theta} \rangle - \log\det \mathbf{\Theta} \\ \text{s.t.} & \Theta_{ij} = 0, \quad (i,j) \notin S^{(t)}. \end{array} \tag{28}$$

   Let $\mathbf{\Theta}^{(t)}$ be the optimal solution and f^* be its value.
   (a) If $f^* < \mu$, set $\mathbf{\Theta_{GBD}} = \mathbf{\Theta}^{(t)}$ and $\mu = f^*$. If $\mu - \eta^{(t)} < \epsilon$, stop. Otherwise, go to Step 1b.
   (b) Let $\mathbf{r}_u^{(t)} = [(\hat{\mathbf{\Theta}}^{-1} - \overline{\mathbf{\Sigma}})_L]^+$ and $\mathbf{r}_l^{(t)} = [(\overline{\mathbf{\Sigma}} - \hat{\mathbf{\Theta}}^{-1})_L]^+$. Go to Step 2.
2. Solve the mixed integer optimization problem,

$$\begin{array}{ll} \min\limits_{\mathbf{z},\eta} & \eta \\ s.t. & \eta \geq v(\mathbf{\Theta}^{(i)}, \mathbf{r}_u^{(i)}, \mathbf{r}_l^{(i)}, \mathbf{z}) \quad i = 1, ..., t \\ & \mathbf{z} \in P, \end{array} \tag{29}$$

   where v is as defined in equation (18). Set $t = t + 1$. Let $(z^{(t)}, \eta^{(t)})$ be the optimal solution. Go to Step 1.

Properties and Observations. We summarize some important observations, properties, and connections to the literature for the above GBD algorithm.

1. Generalized Benders decomposition is a method that can be used to solve convex MINLO problems. In this context, problem (29) is often referred to as the master problem, and problem (28) is referred to as the subproblem. In general, the method involves fixing the integer variables for a convex MINLO and then solving the resulting convex optimization problem. The solutions to the convex optimization problem are then used to construct a constraint to include in the master problem. The GBD algorithm converges in this context in a finite number of steps because subproblems (28) are convex and satisfy Slater's condition, and the set P is finite (see Theorem 2.4 in [30]). Thus, the above algorithm converges to an optimal solution for the cardinality constrained problem in a finite number of steps.
2. As briefly mentioned, the subproblems (28) are known in the literature as covariance selection problems, which were originally introduced in the 1970s [22]. We discuss how we solve these problems in Section 4.3. In general, we

found these problems can be solved two orders of magnitude quicker (and to higher accuracy) than the semidefinite relaxation of formulation (7).
3. Problem (29) is a mixed integer optimization problem that accumulates constraints as the algorithm progresses. Accordingly, the problem becomes harder to solve at each iteration. To temper this setback, we use lazy constraint callbacks, which we discuss in Section 4.4. Due to the significant advances in discrete optimization (see Section 2.3), we found that the successive versions of the master problem (29) can be solved quickly at each iteration of the algorithm.
4. The algorithm supplies a feasible solution, an upper bound, and a lower bound on the optimal solution even if terminated early. Current heuristic approaches do not offer such a certificate of suboptimality.

4.3 Subproblems

In this section, we discuss how we use a first order method to solve the subproblem (28),

$$\begin{array}{ll} \min\limits_{\boldsymbol{\Theta} \succ 0} & \langle \overline{\boldsymbol{\Sigma}}, \boldsymbol{\Theta} \rangle - \log\det \boldsymbol{\Theta} \\ \text{s.t.} & \Theta_{ij} = 0, \quad (i,j) \notin S. \end{array} \tag{30}$$

We first present some preliminary definitions and results. We then reformulate the problem as an unconstrained optimization problem and present the gradient for the unconstrained problem as derived in [21]. Lastly, we discuss how we efficiently compute the gradient and use it in a first order method.

Preliminaries. For the sparsity pattern S in (30), we define an undirected graph $G_S = (V, S_0)$ with vertices $V = \mathbb{Z}_n$ and edges $S_0 = \{(i,j) \in S : i \neq j\}$. The edges define the non-zero off-diagonal entries of $\boldsymbol{\Theta}$ in (30).

An undirected graph $G = (V, E)$ is called chordal if every cycle of length greater than three has a chord, i.e., an edge joining nonconsecutive nodes of a cycle. In the graphical model literature, the terms triangulated graph or decomposable graph are used as synonyms for a chordal graph.

If the graph G_S for sparsity pattern S is chordal, we will say S is chordal. If $S \subset S'$ and S' is chordal, then we say S' is a chordal embedding for S. [47] establishes that if the graph G_S defined by S is chordal, then there are analytic formulas in terms of the principle minors of $\overline{\boldsymbol{\Sigma}}$ for the solution of (30). We next consider a more general method that solves (30) for any sparsity pattern S.

Reformulation and Gradient. We first reformulate (30) as an unconstrained optimization problem. Recall that $S = \{(i,j) : z_{ij} \neq 0 \text{ or } i = j\}$. Write

$$S = \{(i_1, j_1), ..., (i_q, j_q)\},$$

and define two $p \times q$ matrices,

$$\mathbf{E}_1 := [\mathbf{e_{i_1}}, \mathbf{e_{i_2}} ..., \mathbf{e_{i_q}}] \quad \mathbf{E}_2 := [\mathbf{e_{j_1}}, \mathbf{e_{j_2}}, ..., \mathbf{e_{j_q}}],$$

where the elements of $\mathbf{E}_1$ and $\mathbf{E}_2$ are zero, except $(\mathbf{E}_1)_{i_k,k} = 1$ and $(\mathbf{E}_2)_{j_k,k} = 1$ for $k \in \mathbb{Z}_q$, where $\mathbb{Z}_q := \{1, ..., q\}$. Then, $\mathbf{\Theta}$ can be expressed as,

$$\mathbf{\Theta}(\mathbf{x}) = \mathbf{E}_1 \operatorname{diag}(x)\mathbf{E}_2^\top + \mathbf{E}_2 \operatorname{diag}(x)\mathbf{E}_1^\top,$$

where $\mathbf{x} \in \mathbb{R}^q$ is defined for $k \in \mathbb{Z}_q$ as,

$$x_k := \begin{cases} \Theta_{i_k,j_k}, & i_k \neq j_k, \\ \frac{1}{2}\Theta_{i_k,i_K}, & i_k = j_k. \end{cases}$$

With this notation, problem (30) can be reformulated as the following unconstrained optimization problem,

$$\min_{\mathbf{x}} \quad f(\mathbf{x}) = \langle \overline{\mathbf{\Sigma}}, \mathbf{\Theta}(\mathbf{x}) \rangle - \log\det \mathbf{\Theta}(\mathbf{x}). \tag{31}$$

The gradient of the objective function f in (31) evaluated at $\mathbf{x}$ for $k \in \mathbb{Z}_q$ is derived in [21] and is as follows

$$\nabla_k f(\mathbf{x}) = 2\overline{\Sigma}_{i_k,j_k} - 2(\Theta(x)^{-1})_{i_k,j_k}. \tag{32}$$

Thus, to compute the gradient at $\mathbf{x}$, the $i_k j_k$th entry of the inverse of $\mathbf{\Theta}(\mathbf{x})$ must be computed for each $k \in \mathbb{Z}_q$. This could be done by simply computing the entire inverse $\mathbf{\Theta}(\mathbf{x})^{-1}$, but when p is large this becomes computationally expensive.

As an alternative, [21] shows that if the graph G_S defined by sparsity pattern S is sparse and chordal, then the $i_k j_k$th entries of the inverse of $\mathbf{\Theta}(\mathbf{x})$ can be efficiently computed without computing the entire inverse. Of course, it is not always the case that G_S is chordal; to circumvent this, we compute a chordal embedding S' for S, and then apply the method of [21] to compute the entries of $(\mathbf{\Theta}(\mathbf{x})^{-1})_{ij}$ for all $(i, j) \in S'$. If S' is not much larger than S, this method can be significantly faster than computing the entire inverse. For large and sparse matrices, [21] reports speedups of two to three orders of magnitude for the runtime for computing the inverse, and hence the gradient (32). To compute S', we use the following suggested heuristic in [21]. We first generate an approximate minimum degree ordering (see [1]), followed by a symbolic Cholesky factorization. The sparsity pattern S' of the Cholesky factor defines a chordal embedding for S [21].

Lastly, we now discuss how we use (32) in a first order method. [21] reports that a limited-memory Broyden-Fletcher-Goldfarb-Reeves (BFGS) method [40] significantly outperforms other first order methods (coordinate steepest descent and conjugate gradient, for example). Thus, for all computations involving covariance selection problem (30), we use a limited-memory BFGS method.

4.4 Lazy Constraint Callbacks

In this section, we describe a method called lazy constraint callbacks, which we use in combination with the GBD algorithm to enhance overall algorithmic performance. The GBD algorithm (as stated in Section 4.2) requires that problem (29) be resolved at each iteration when a new constraint is added. Since the master problem is a MIO problem, a branch and bound tree is constructed by the solver to solve the problem (see Section 3.2). Thus, a branch and bound tree is built at each iteration of the algorithm to solve the master problem.

Lazy constraint callbacks provide an alternative to building a new branch and bound tree at each iteration of the algorithm. When a constraint is added to the master problem, instead of resolving the problem, the constraint is added to all active nodes in master problem's current branch and bound tree. This enables the same tree to be used for all iterations. This saves the rework of building a new tree every time a mixed integer feasible solution is found.

Lazy constraint callbacks are a relatively new type of callback. CPLEX 12.3 introduced lazy constraint callbacks in 2010 and Gurobi 5.0 introduced lazy constraint callbacks in 2012. To date, the only mixed integer solvers which provide lazy callback functionality are CPLEX [35], Gurobi [31], and GLPK (see http://gnu.org/software/glpk/).

5 Computational Results

In this section, we present numerical results on both synthetic and real data. In Section 5.1, we describe the synthetic data used for experiments. In Section 5.2, we provide results for GBD runtimes on the synthetic data. In Section 5.3, we test the algorithm's ability to recover the underlying precision matrix and compare its performance with Glasso. Lastly, in Section 5.4, we consider the statistical performance on a gene expression dataset and compare our results with known results.

5.1 Synthetic Experiments

Recall the data of interest is a covariance matrix $\mathbf{\Sigma}_0$. Our approach for generating such a matrix is motivated by existing approaches [5].

We first generate a sparse underlying precision matrix $\mathbf{\Theta}_0 \in \mathbb{R}^{p\times p}$. We let $\mathbf{\Theta}_0 = \mathbf{B} + \gamma\mathbf{I}$, where $\mathbf{B}$ is a symmetric matrix with $t \in \mathbb{Z}_+$ non-zero strictly lower triangular entries equal to 0.5. The scalar γ is chosen such that the condition number of $\mathbf{\Theta}_0$ is equal to p, the dimension of the matrix. Then, the matrix is standardized to have unit diagonal entries.

Next we generate a training sample of size n from a multivariate normal distribution with mean $\mathbf{0}$ and covariance $\mathbf{\Theta}_0^{-1}$. Lastly, we construct the synthetic covariance $\mathbf{\Sigma}_0$ from these samples.

5.2 Computational Performance

In this section, we consider the runtime of GBD with additional considerations as given in Section 4. In particular, we study the average time over ten instances it takes the algorithm to find an optimal solution, and the average time over ten instances it takes the algorithm to verify the solution's optimality. First we describe computer specifications and then discuss results.

Computer Specifications and Software. All computation is done on a 2015 Macbook with a 1.1 GHz Intel Core and 8 GB of RAM. In general all computation used all available cores. All synthetic data is generated in Julia. Master

problems in the GBD algorithm are solved using Julia with JuMP package [38] and Gurobi solver. The problems (8) and (9) (used to estimate bounds on M_{ij}) are also solved in Julia. A limited memory BFGS method implemented in Python is used for solving the nonlinear subproblems; package CVXOPT [2] is used for sparse matrix computation, and package CHOMPACK [3] is used for computing the partial inverse via chordal embeddings in gradient calculations. Python code is interfaced with Julia code using Julia package PyCall.

Runtime and Optimality Detection. We consider the average time it takes GBD to find an optimal solution and verify its optimality for the cardinality constrained problem. We carry out all experiments by generating 10 instances of synthetic data for $(p, t) \in \{30, 50, 80, 120, 200\} \times \{5, 10\}$ and $n \in \{20k, 20k + 50, 20k + 100\}$. We solve each instance with GBD with $k = t$ and report average performance. For a summary of results, see Table 1.

p	t	n	ver-time	opt-time	laz-cons
30	5	200	1.27	0.10	21
		150	2.88	0.10	53
		100	3.49	0.13	60
30	10	300	39.95	0.10	189
		250	49.11	4.31	379
		200	68.25	6.71	471
50	5	200	5.85	0.12	76
		150	8.86	0.15	93
		100	15.41	0.55	154
50	10	300	75.89	0.15	231
		250	96.48	7.89	452
		200	124.21	14.23	541
80	5	200	8.22	0.15	89
		150	11.20	0.35	112
		100	17.42	0.73	212
80	10	300	90.30	2.21	275
		250	117.83	10.25	547
		200	153.67	17.54	625
120	5	200	21.72	0.45	188
		150	45.61	0.63	223
		100	59.92	1.02	321
120	10	300	110.32	6.32	362
		250	160.56	22.21	621
		200	219.32	28.21	673
200	5	200	45.23	1.24	274
		150	98.23	2.32	501
		100	127.32	2.65	752
200	10	300	621.43	32.33	963
		250	763.21	43.22	1022
		200	855.65	48.21	1332

Table 1 Average performance on instances of synthetic data with $k = t$ and $n = 20k,\ 20k + 50,\ 20k + 100$. All problems are solved to a tolerance gap of 10^{-4}, where the tolerance gap is the percentage difference between the final lower and upper bounds. Title ver-time refers to the time in seconds it takes to verify optimality, opt-time refers to the time at which the optimal solution is found, and laz-cons refers to the number of lazy constraints generated for the GBD master problem.

We include the following observations.

1. In general the algorithm provides an optimal solution in a matter of seconds, and a certificate of optimality in seconds or minutes even for p in the 100s. Optimal verification occurs significantly quicker when the sample size n is larger because the sparsity pattern of the underlying matrix is easier to recover. However, we note that finding the optimal solution is not as drastically effected by the sample size n.
2. As p or k increase, optimal detection also does not significantly change, but optimal verification generally becomes significantly harder. Similar observations have been made for mixed integer formulations of the best subset selection problem in linear and logistic regression [14]. We also observe that changes in k have a more substantial impact on the runtime than changes in n or p, especially when p is large.

Thus, the GBD algorithm in general provides an optimal or near-optimal solution fast, but optimal verification strongly depends on p, k, and n. Nonetheless, we observe that optimality of solutions can be verified for p in the 100s and k in the 10s in a matter of minutes.

5.3 Recovering Structure

In this section, we study the ability of the algorithm to recover the underlying precision matrix. We carry out all experiments by generating 10 instances of synthetic data for $p \in \{100, 200, 500, 1000\}$, $n \in \{0.75p, p, 2p, 3p\}$, and t is chosen such that the off-diagonal non-zero density of the matrix is 0.25 or 0.50 percent. We solve each instance with GBD with $k = t$ and report average performance. Because optimal verification can take awhile, but optimal detection occurs quickly, we allow the algorithm to run for 300 seconds for each instance tested and use the current iterate at this time. Note that by assuming $k = t$, we are assuming we know the amount of nonzero entries in the true matrix; we postpone how to choose k via model selection until the next section.

To assess how well the solutions recover the underlying precision matrix, we employ two techniques. First, we measure the distance under the Frobenius and operator norm between our solution and the true underlying precision matrix $\mathbf{\Theta}_0$. For $\mathbf{A} \in \mathbb{R}^{p\times p}$, the operator norm is defined by $\max_i \sqrt{\lambda_i}$, where λ_i are the eigenvalues of $\mathbf{A}^\top \mathbf{A}$. Second, we study how well the solution recovers the support and sparsity pattern of the underlying precision matrix $\mathbf{\Theta}_0$; this is especially of interest in the context of estimating the structure of sparse graphical models. We measure support recovery by computing the proportion of non-zero and zero entries that are identified correctly. We will refer to these metrics as true positive (TP) and true negative (TN), respectively.

For a summary of recovery results, see Table 2. We include observations below.

1. The algorithm accurately recovers the sparsity structure in all cases, even for p in the 1000s and k in the 1000s to high accuracy for both $n \leq p$ and $n > p$. We observe that for all cases of $n \geq p$, greater than 90 percent of the non-zero elements are recovered.

2. We observe that as p gets larger, the underlying precision matrix is recovered with more accuracy. One reason for this is that the sample-size substantially increases for each value of p.

p	t	n	TP	TN	Operator	Frobenious
100	12	300	100.00	100.00	0.35	1.21
		200	100.00	100.00	0.35	1.40
		100	95.00	99.99	0.71	2.37
		75	75.00	99.93	0.83	2.82
100	25	300	100.00	100.00	0.37	1.22
		200	96.00	99.98	0.43	1.55
		100	90.00	99.92	0.78	1.60
		75	72.00	99.79	1.62	4.12
200	50	600	100.00	100.00	0.40	1.25
		400	100.00	100.00	0.42	1.67
		200	96.00	99.98	0.73	2.37
		150	86.00	99.96	0.75	3.30
200	100	600	100.00	100.00	0.54	1.75
		400	97.00	99.98	0.77	2.44
		200	90.00	99.94	0.84	4.11
		150	75.00	99.87	0.99	5.42
500	300	1500	100.00	100.00	0.60	2.37
		1000	100.00	100.00	0.63	2.89
		500	96.33	99.99	0.75	4.61
		375	91.00	99.97	0.85	6.20
500	600	1500	100.00	100.00	0.65	3.88
		1000	97.00	99.99	0.97	5.36
		500	91.00	99.98	1.21	11.06
		375	78.16	99.92	1.68	13.99
1000	1250	3000	100.00	100.00	0.67	3.24
		2000	100.00	100.00	0.72	4.27
		1000	97.20	99.99	0.86	7.06
		750	93.44	99.98	1.05	10.01
1000	2500	3000	100.00	100.00	0.75	5.79
		2000	98.92	99.99	0.90	7.71
		1000	93.92	99.98	1.46	17.46
		750	82.44	99.83	2.08	24.51

Table 2 Average recovery performance of the algorithm's solutions on instances of synthetic data with off-diagonal density approximately 0.50 and 0.25 percent. Labels operator and Frobenius refer to norm difference between true underlying precision matrix and estimate under consideration.

Thus, our approach recovers precision matrices to high accuracy for all instances tested in Table 2, even when p and k are in the 1000s and the number of samples is less than the dimension.

5.4 Performance Comparison

In this section, we compare solutions on instances of the synthetic data with Glasso solutions. We consider each estimator's ability to accurately recover the true underlying precision matrix using the four metrics Frobenius (FR), Operator (OP), TP and TN introduced in the previous section.

Recall we must perform model selection on λ and k for the ℓ_1-regularized and the cardinality constrained problem, respectively. Accordingly, we generate instances of the models as in the previous sections, but now additionally create two independent samples of size $n = 100$, one for training (λ and k) and one for validation. We report the average performance on 10 instances of each problem of interest.

We train Glasso by successively computing estimates for 50 different values of λ on the training set. We choose the estimate with the largest likelihood on the validation set. All computation is done in R using the Glasso package found at http://cran.r-project.org/web/packages/glasso. We train the MIO model as follows.

MIO Training. We train the MIO model (7) via solving the problem for a variety of values of k on the training set. In particular, we solve all models in this section for $k = 50$ to $k = 5$. Rather than resolving the entire model for each value of k, instead we use lazy constraint callbacks (see Section 4.4) to iteratively solve the models from $k = 50$ down to $k = 5$. That is, once a solution is found for k, a constraint of form $\sum_{i>j} z_{ij} \leq k-1$ is added to the model. This process is iterated until a solution is found for $k = 5$. For each value of k, we solved either to the time limit of 300 seconds or the optimality gap of 10^{-4}. We choose the estimate with the largest likelihood on the validation set.

Results. We provide a comparison for each model for $p \in \{30, 60, 90, 120, 200\}$, $t = 0.2p$, and $n = 100$. For a summary of results, see Figure 1 and Table 3. We include the following observations.

1. We observe that for all cases tested, the cardinality constrained estimates provide a significantly sparser estimate, and a more accurate estimate. Similar observations have been observed for mixed integer model solutions for the best subset selection in linear regression [14] .
2. As p increases, we observe that the difference between the cardinality constrained solution's TP rate and Glasso's TP rate increases. That is, the rate at which Glasso misidentifies elements as p increases is larger than the rate at which the cardinality constrained solutions misidentify nonzero elements. We also observe that the cardinality constrained solutions are consistently very sparse, while the sparsity of Glasso's solutions depend on the dimension p.

p	GBD				Glasso			
	TP	TN	OP	FR	TP	TN	OP	FR
30	100.00	100.00	0.29	0.79	99.98	67.30	0.67	1.65
50	97.00	99.98	0.45	1.35	96.78	73.22	0.98	2.83
80	93.13	99.97	0.57	1.62	92.62	75.51	1.27	4.21
120	90.00	99.97	0.63	1.92	84.45	79.98	1.75	5.12
200	83.75	99.97	0.68	3.34	75.23	71.64	2.21	7.35

Table 3 Average performance of algorithm and ℓ_1-regularized estimates with model selection on 10 instances of synthetic data. Parameter t is chosen such that $t = 0.2p$.

Thus, we observe that the cardinality constrained estimates provide a significantly sparser estimate, and a significantly more accurate estimate than Glasso. In

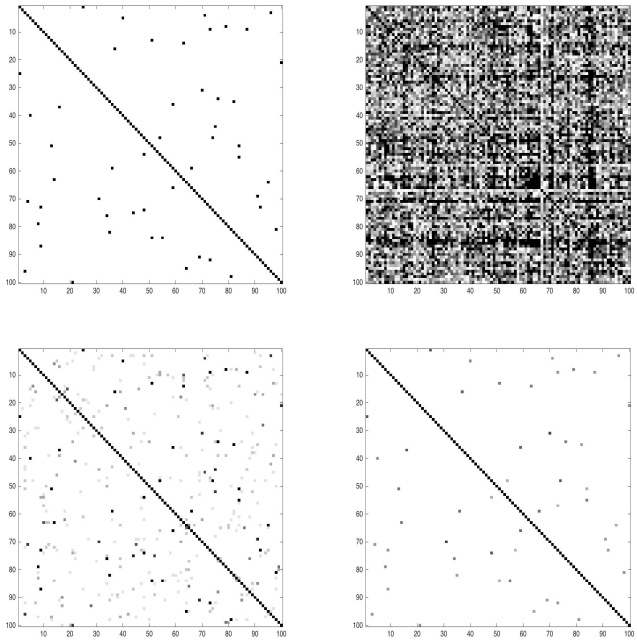

Fig. 1 Typical recovery plots for synthetic data with $p = 50$, $t = 25$, and $n = 125$. The upper-left plot is the true precision matrix, the lower-left plot is the ℓ_1-regularized estimate with $\lambda = 0.12$, the upper-right plot is the sampled matrix, and the lower-right plot is the estimate of the GBD algorithm with $k = 25$. TP and TN for our estimate are 96.00 and 99.98, respectively.

the next section, we compare the cardinality constrained estimates with additional statistical estimators on a real dataset.

5.5 Analysis of a Breast Cancer Dataset

We apply our method on a real breast cancer dataset analyzed in [33]. The dataset can be found at http:// bioinformatics.mdanderson.org/. The dataset consists of 22,283 gene expression levels for 133 patients, including 34 with pathological complete response (pCR) and 99 with residual disease (RD). The pCR subjects are considered to have a high chance of cancer-free survival in the long term, and thus it is of interest to study the response states of the patients (pCR or RD) to preoperative chemotherapy. The main objective of this analysis is to estimate the inverse covariance matrix of the gene expression levels and then apply linear discriminant analysis (LDA) to predict whether or not a subject can achieve the pCR state.

The dataset has been studied in [24] using Glasso, revised Glasso, and SCAD. Later the same analysis was performed with the CLIME estimator [19]. For the sake of consistency, we perform the same analysis, but use our method to estimate inverse covariance matrices when needed. We first briefly describe how the data

is prepared and analyzed. We then present our results and compare with known results in [24, 19].

The data is first randomly divided into testing and training sets using stratified sampling. 5 pCR subjects and 16 RD subjects are randomly chosen to constitute the testing data. The remaining 112 subjects are chosen to constitute the training data. This process is repeated 100 times and the following data preparation techniques are used on each of the 100 instances of the training and testing data. A two sample t-test is performed between the two groups in the training dataset to determine the most significant genes; we retain the 113 genes with the smallest p-values as the variables for prediction and the rest are discarded. The data for each variable (gene) is then standardized by dividing the data with the corresponding standard deviation, estimated from the training dataset.

We next perform the linear discriminant analysis. We assume the normalized gene expression data are normally distributed as $N(\mu_k, \mathbf{\Sigma})$, where the two groups have the same covariance $\mathbf{\Sigma}$, but different means, μ_k ($k = 1$ for pCR and $k = 2$ for RD). The linear discriminant scores are as follows:

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \hat{\mathbf{\Sigma}}^{-1} \hat{\mu}_k - \frac{1}{2} \hat{\mu}_k^\top \hat{\mathbf{\Sigma}}^{-1} \hat{\mu}_k + \log \pi_k,$$

where $\pi_k = n_k/n$ is the proportion of the number of observations in the training data belonging to class k, and the classification rule is given by $\text{argmax}_k \, \delta_k(\mathbf{x})$. Based on each training dataset, we estimate the mean $\hat{\mu}_k$ as,

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i \in class-k} \mathbf{x}_i \quad \text{for } k = 1, 2,$$

and the precision matrix $\hat{\mathbf{\Sigma}}^{-1}$ using the cardinality constrained problem. Since the sample size is less than the dimension of the matrix, the empirical covariance is not invertible and can not be used in LDA.

Comparison Metrics	Description
Specificity	$\frac{TN}{TN+FP}$
Sensitivity	$\frac{TP}{TP+FN}$
MCC	$\frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

Table 4 Metrics used for prediction performance comparison for the breast cancer dataset. TP, TN, FP, and FN are the number of true positives, true negatives, false positives and false negatives, respectively. Positives correspond to pCR subjects and negatives correspond to RD subjects.

The classification performance of δ_k is clearly associated with the estimation performance of $\hat{\mathbf{\Sigma}}^{-1}$. Let true positive (TP) be the number of pCR subjects δ_k identifies as pCR subjects and let true negative (TN) be the number of RD subjects δ_k identifies as RD Subjects. To compare prediction performance, we use comparison metrics: specificity, sensitivity, and also Matthews Correlation Coefficient (MCC). They are each defined in Table 4. MCC is widely used in machine

learning for assessing the quality of a binary classifier; it takes true and false, positives and negatives, into account and is generally regarded as a balanced measure. A larger MCC value indicates a better classifier [24].

Method	Specificity	Sensitivity	MCC	NNZ
Glasso	0.768 (0.009)	0.630 (0.021)	0.366 (0.018)	3923 (2)
Adaptive Lasso	0.787 (0.009)	0.622 (0.022)	0.381 (0.018)	1233 (1)
SCAD	0.794 (0.009)	0.634 (0.022)	0.402 (0.020)	674 (1)
CLIME	0.749 (0.009)	0.806 (0.017)	0.506 (0.020)	492 (7)
GBD	0.906 (0.009)	0.651 (0.013)	0.543 (0.017)	317 (11)

Table 5 Comparison of estimators on the breast cancer dataset. Data for Glasso, revised Glasso and SCAD is from [24] and data for CLIME is from [19]. Average performance is reported on 100 instances of training and testing data; standard deviations are included in parentheses. NNZ refers to the number of non-zero entries in the estimate.

We perform the LDA for each of the 100 instances and report a summary of average performance in Table 5. For each experiment, we use 2-fold cross-validation on the training dataset to determine the parameter k. We train models using a method similar to the one used in Section 5.4, but instead of decreasing k by increments of 1 and starting at 50, we decrease k by increments of 10 and start at 150. We observe that our proposed method outperforms all other methods in specificity and the majority of other methods in sensitivity. We also observe our method obtains the best MCC. However, the biggest advantage of the method over the others is it produces a significantly sparser estimate. This is especially desirable in the context of graphical models, when it is desirable to induce sparsity for explanatory and predictive power.

6 Summary

We use a variety of modern optimization methods to provide the first exact algorithm for solving the cardinality constrained negative log-likelihood problem (1). The method provides provably optimal solutions, and delivers near optimal solutions fast for p and k in the 1000s. Computational experiments show that the our method delivers significantly sparser and more accurate (in terms of sparsity pattern recovery) solutions than existing methods.

References

1. Amestoy, P., Davis, T., and Duff, I. An approximate minimum degree ordering. *SIAM Journal on Matrix Analysis and Applications*, 17(4):886-905, 1996.
2. Andersen, M. S., Dahl, J., and Vandenberghe, L. CVXOPT: A Python package for convex optimization, Version 1.1.7. *Available at cvxopt.org*, 2014.
3. Andersen, M. S. and Vandenberghe, L. A Python Package for Chordal Matrix Computations, Version 2.1. *cvxopt.github.io/chompack*, 2014.
4. Atamtürk, A., Narayanan, V. Cuts for Conic Mixed-Integer Programming. *Integer Programming and Combinatorial Optimization*, 4513:16-29.
5. Banjerjee, O., El Ghaoui, L., and D'Aspremont, A. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *The Journal of Machine Learning Research*, 9:485-516, 2008.

6. Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. Robust Optimization. *Princeton University Press*, 2009.
7. Benders, J. F. Partitioning Procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4: 238-252, 1962.
8. Benson, Y. H., and Saglam U. Mixed-Integer Second-Order Cone Programming: A survey. *Operations Research*, Tutorials Chapter 2, 13-36, 2014.
9. Bertsimas, D., Brown B. D., and Caramanis, C. Theory and Applications of Robust Optimization. *SIAM Review*, 53(3):464-501, 2011.
10. Bertsimas, D. and Copenhaver, M. Characterization of the equivalence of robustification and regularization in linear, median, and matrix regression. *Submitted for publication*, 2014.
11. Bertsimas, D. and Dunn, J. Optimal Trees. *Submitted for publication*, 2015.
12. Bertsimas, D. and King, A. An Algorithmic approach to Linear Regression. Appeared online in *Operations Research*, 2015.
13. Bertsimas, D. and Mazumder, R. Least Quantile of Squares Regression via Modern Optimization. *Annals of Statistics*, 6:2394-2525, 2014.
14. Bertsimas, D., King, A., and Mazumder, R. Best Subset Selection via a Modern Optimization Lens. *Annals of Statistics to appear*, 2016.
15. Bertsimas, D. and Weismantel, R. Optimization over integers. *Dynamic Ideas*, Belmont, Massachusetts, 2005.
16. Bickel, P. and Levina, E. Covariance regularization by thresholding. *Annals of Statistics*, 36:2577-2604, 2008.
17. Bonami, P., Kilinc, M., and Linderoth, J. Algorithms and software for convex mixed integer nonlinear programs. *Mixed Integer Nonlinear Programming*, 1-39, 2012.
18. Boyd, V. and Vandenberghe, L. Convex Optimization. *Cambridge University Press*, New York, USA. 2004.
19. Cai, T., Liu, W., and Luo X. A constrained l_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Assocation*, 106, 594-607. 2011.
20. Costa, A. M. A survey on Benders decomposition applied to fixed-charge network design problems. *Computers and Operations Research*, 32: 1429-1450, 2005.
21. Dahl, J., Vandenberghe, L., Roychowdhury, V. Covariance selection for non-chordal graphs via chordal embedding. *Optimization Methods and Software*, 23(4):501-520, 2008.
22. Dempster, A. P. Covariance Selection. *Biometrics*, 28:157-175, 1972.
23. El Karoui, N. High-dimensionality effects in the markowitz problem and other quadratic programs with linear constraints: risk underestimation. *Annals of Statistics*, 38:3487-3566, 2010.
24. Fan, J., Feng, Y., and Wu, Y. Network Exploration via the Adaptive Lasso and SCAD Penalties. *The Annals of Applied Statistics*, 2:521-541, 2009.
25. Fan, J., Fan, Y., Lv, J. High dimensional covariance estimation using a factor model. *Journal of Econometrics*, 147:186-197, 2008.
26. Fan, J., Zhang, J., and Yu, K. Vast portfolio selection with gross-exposure constraints. *Journal of American Statistical Association*, 107:592-606, 2012.
27. Fan, J., Han, F., and Liu, H. Challenges of big data analysis. *National Science review*, 1:293-314, 2014.
28. Fan J., Liao Y., and Liu H. An Overview on the Estimation of Large Covariance and Precision Matrices. *arXiv preprint arXiv:1504.02995v2*, 2015.
29. Friedman, J., Hastie, T., and Tibshirani, R. Sparse inverse covariance estimation with graphical lasso. *Biostatistics*, 9:432-441, 2008.
30. Geoffrion, A. M. Generalized Benders Decomposition. *Journal of Optimization Theory and Applications*, 10, 4: 237-260. 1972.
31. Gurobi Inc. Gurobi Optimizer Reference Manual. *Accessed:* 2014.
32. Hoang Hai Hoc. Topological optimization of networks: A nonlinear mixed integer model employing generalized Benders decomposition. *Automatic Control, IEEE Transactions on Automatic Control*, 27, 1: 164-169, 2003.
33. Hess, K. R., Anderson, K., Symmans, W. F., Valero, V., Ibrahim, N., Mejia, J. A., Booser, D., Theriault, R. L., Buzdar, A. U., Dempsey, P. J., Rouzier, R., Sneige, N., Ross, J. S., Vidaurre, T., Gmez, H. L., Hortobagyi, G. N., and Pusztai, L. (2006), Pharmacogenomic Predictor of Sensitivity to Preoperative Chemotherapy With Paclitaxel and Fluorouracil, Doxorubicin, and Cyclophosphamide in Breast Cancer. *Journal of Clinical Oncology*, 24:4236-4244, 2006.

34. Hooker, J.N., Ottosson, G. Logic-Based Benders Decomposition. *Mathematical Programming*, 96, 1: 33-60., 2003.
35. IBM ILOG CPLEX. Optimization Studio. Cplex optimizer. *Accesed:* 2015.
36. Lam, C. and Fan, J. Sparsistency and rates of convergence in large covariance matrix estimation. *Annals of Statistics*, 37:4254-4278, 2009.
37. Lauritzen, S. Graphical Models (Oxford Statistical Science Series). *Clarendon Press*, New York, New York, 1996.
38. Lubin, M. and Dunning, I. Computing in Operations Research Using Julia. *INFORMS Journal on Computing*, 27(2):238-248, 2015.
39. Mazumder, R. and Hastie, T. Exact covariance thresholding into connected components for large-scale graphical Lasso. *The Journal of Machine Learning Research*, 13:781-794, 2012.
40. Nocedal, J. and Wright, S.J. Numerical Optimization. *Springer. 2nd edition*, New York, New York, 2011.
41. Rigollet, P. and Tsybakov, A. Estimation of covariance matrices under sparsity constraints. *arXiv preprint arXiv:1205.1210*, 2012.
42. Rothman, A. J., Bickel, P. J., Levina, E., and Zhu, J. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494-515, 2008.
43. Tokuda, T., Goodrich, B., Van Mechelen, I., Gelman, A. and Tuerlinckx, F. Visualizing distributions of covariance matrices. *Technical report*, Columbia University.
44. Türkay, M. and Grossmann, I. Logic-based MINLP algorithms for the optimal synthesis of process networks. *Computers and Chemical Engineering*, 20, 8: 959-978. 1996.
45. Vandenberghe, L., Boyd, S., and Wu, S. Determinant maximization with linear matrix inequality constraints. *SIAM Journal on Matrix Analysis and Applications*, 19(2):499-533, 1998.
46. Wei, Z. and Ali, M. Generalized Benders Decomposition for one Class of MINLPs with Vector Conic Constraint. *SIAM Journal on Optimization*, 25(3):1809-1825, 2015.
47. Wermuth, N. Linear recursive equations, covariance selection, and path analysis. *SIAM Journal on Computing*, 13(3):556-579,1984.
48. Yuan, M. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11, 2261-2286.
49. Yuan, M. and Lin, Y. Model selection and estimation in the Gaussian graphical model. *Biometrica*, 94,1:19-35, 2007.
50. Xu, H., Caramanis, C., and Mannor, S. Robust Regression and Lasso. *IEEE Transactions on Information Theory*, 56:3561-3574, 2010.