Institut Polytechnique de Paris

Graph Learning with ℓ_0-norm and SLOPE Penalties

Abstract

Estimating sparse conditional dependence graphs from multivariate observations is a central problem in Gaussian graphical models. The graphical lasso(Least Absolute Selection and Shrinkage Operator) is a prominent method for recovering a sparse precision matrix from node signals through a penalized log-determinant optimization problem. The choice of penalty has a direct impact on the reconstructed topology. In this study, we compare three sparsity regularization: the classical ℓ_1 penalty used in graphical lasso, the ℓ_0 penalty, and the SLOPE penalty. Experiments are conducted on both synthetic datasets and on the S&P500 real dataset. On synthetic data, both methods show highly competitive performance. SLOPE based methods is particularly less sensitive to parameters. On the S&P500 dataset, Lasso and SLOPE based methods yield very strong ability to recover the topology structure, while ℓ_0 based method encourage a more sparse recovering. Interestingly, our results indicate that the SLOPE and ℓ_0 based methods yields the highest F-Score and the highest modularity receptively.

Keywords: statistical learning; optimization

Master's thesis
Student: Baozhuo Su
Advisor: Arnaud Breloy
Date final version: 24/03/2026

Contents

1 Introduction **2**

2 Background **3**
2.1 Gaussian graphical models and Glasso 3
2.2 Definition of different penalized problems 4

3 Methodology **5**
3.1 The alternating direction method of multipliers(ADMM) 5
3.2 Solving Glasso with ADMM 6
3.3 Solving Gslope with ADMM 7
3.4 Solving Gl0 with non-convex ADMM 7

4 Datasets **8**
4.1 Synthetic dataset 8
4.2 S&P500 index dataset 8

5 Experiments and Results **10**
5.1 Results on synthetic dataset 10
5.2 Results on real-world dataset 11

6 Conclusion **11**

7 Appendix **12**

References **17**

1 Introduction

Graphical models allow us to represent specific correlation structures between any two variables of observations. Inferring the topology of this structure directly from the data is referred to as graph learning [1]. The graph learning approach has been widely used in different applications, such as biology [2–4], finance [5], and signal processing [6, 7].

Within Gaussian graphical models (GGMs), graph learning boils down to the problem of estimating the precision matrix, which is the inverse of the covariance matrix of a Gaussian Markov random field [8, 9]. In practice, achieving an accurate covariance matrix estimation is often a difficult task. The estimator is typically unsatisfactory or ill-defined when the number of observations approaches the number of nodes [10], i.e., in the high-dimensional case. In particular, when the number of observations is smaller than or close to the number of nodes, the sample covariance matrix becomes ill-conditioned, and its inverse is unstable or may not exist.

The classical solution to the above issues relies on the application of penalized likelihood estimators [11]. Another possible route is to directly penalize the number of graph edges, as in the Akaike Information Criterion [12] or the Bayesian Information Criterion [13]. As discussed, e.g., in [14], when the number of variables becomes large, AIC and BIC can lead to many false recoveries and need to be replaced by criteria that account more carefully for model dimensionality, such as the modified Bayesian Information Criterion [15, 16] or the Extended Bayesian Information Criterion [17]. However, identifying the model that maximizes a given model selection criterion is NP-hard. Therefore, a penalty is often added directly to the objective function.

The ℓ_0 and ℓ_1 penalties are popular sparsity-inducing penalties in a wide range of statistical problems. The ℓ_0 norm was first proposed to solve minimum set cover problems in [18] and was extended to signal processing by [19]. The ℓ_1 norm is another popular penalization. It was originally employed by [20] in geophysics and by [21] in the context of signal processing. In [22], the ℓ_1 penalty was introduced into statistics through the well-known Least Absolute Shrinkage and Selection Operator (LASSO) for variable selection in regression models. The first application of LASSO to sparse inverse covariance estimation was proposed by [23] as the neighborhood selection method. Subsequently, in [8] the graphical lasso (Glasso) was proposed, where the ℓ_1 penalty is used directly in the penalized likelihood. Then several generalizations of LASSO were proposed to solve different problems. Elastic Net [24] relies on a linear combination of ℓ_1 and ℓ_2 penalties and encourages the inclusion of groups of correlated predictors. Another extension of LASSO is SLOPE [15], in which the absolute values of the parameters are sorted in descending order.

More recently, in [25], SLOPE has been applied to Gaussian graphical models. Similarly to the neighborhood selection version of graphical lasso, that method relies on the application of SLOPE to a system of multiple regression problems and has been given the name Neighborhood Selection Sorted L-One Penalized Estimator (nsSLOPE). In this report, we follow the path of Glasso and directly apply a SLOPE penalty to the penalized Gaussian log-likelihood. Moreover, we compare Glasso, the SLOPE penalty, and the ℓ_0 norm on both synthetic and real datasets. We observe that SLOPE and ℓ_0 demonstrate competitive ability to recover graph structure on both synthetic and real-world data.

2 Background

2.1 Gaussian graphical models and Glasso

We consider an undirected, weighted graph represented by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, where $\mathcal{V} = \{1, 2, \ldots, p\}$ denotes the set of vertices (nodes), $\mathcal{E} \subseteq \{\{u, v\} : u, v \in \mathcal{V},\ u \neq v\}$ denotes the edge set, and $\mathbf{A}$ is the adjacency matrix of the graph. Denote by $\mathbf{x}_i = [x_1, \ldots, x_p]^\top$ one observation vector over the p nodes. The data matrix is then denoted by

$$\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{p \times n},$$

with $\mathbf{x}_i \in \mathbb{R}^p$ being one observation of a graph signal on all the p nodes, as shown in Figure 1.

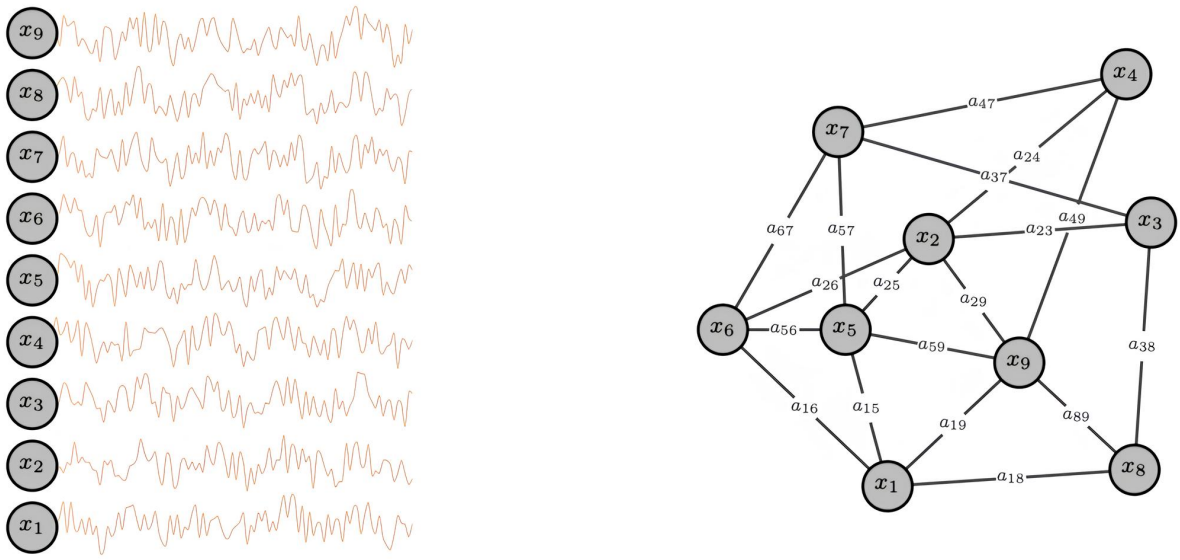


Figure 1: Signals at each node (data matrix $\mathbf{X}$) and conditional correlation graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$.

Gaussian graphical models assume that each observation is a centered multivariate Gaussian random vector $\mathbf{x} = [x_1, x_2, \ldots, x_p]^\top$ with covariance matrix $\mathbb{E}[\mathbf{x}\mathbf{x}^\top] = \mathbf{\Sigma}$, denoted by $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. For the corresponding Gaussian Markov random field, an undirected graph is matched to the variables as follows: each variable corresponds to a vertex, and an edge is present between two vertices if the corresponding random variables are conditionally dependent given all others [26, 27]. The support of the precision matrix $\mathbf{\Theta} = \mathbf{\Sigma}^{-1}$ directly accounts for this conditional dependency, since

$$\operatorname{corr}\left[x_q, x_l \mid \mathbf{x}_{[1,p]\setminus\{q,l\}}\right] = -\frac{\Theta_{ql}}{\sqrt{\Theta_{qq}\Theta_{ll}}}. \tag{1}$$

In this case, a non-zero entry Θ_{ql} implies a conditional dependency between the variables x_q and x_l, underlined by an edge between vertices q and l of the graph.

Within Gaussian graphical models, graph learning is therefore tied to the problem of estimating the precision matrix $\mathbf{\Theta}$ from the data matrix $\mathbf{X}$. The sample covariance matrix is defined as

$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top. \tag{2}$$

A standard approach is to resort to regularized maximum likelihood estimation with $\mathbf{\Theta}$ constrained to be positive defined, i.e., $\mathbf{\Theta} \in \mathcal{S}_p^{++}$, which is equivalent to solving the following optimization problem:

$$\max_{\mathbf{\Theta} \in \mathcal{S}_p^{++}} \log\det(\mathbf{\Theta}) - \mathrm{tr}(\mathbf{S\Theta}) - \lambda h(\mathbf{\Theta}), \tag{3}$$

where h denotes the penalty term and λ is the associated tuning parameter. If the ℓ_1 norm is applied, that is

$$h(\mathbf{\Theta}) = \|\mathbf{\Theta}\|_{1,\text{off}} \tag{4}$$

In this case, it yields the well known Glasso estimator. Many other penalties can also be used, such as the ℓ_0 and SLOPE norms, both of them promote sparse topology structures. In the next section, we will detail possible choices for $h(\mathbf{\Theta})$.

2.2 Definition of different penalized problems

The ℓ_0 and SLOPE penalties are widely used to find sparse approximate solutions for many problems, and they are well known in best subset selection problems in statistics [28]. Recently, they have also been applied to sparse inverse covariance estimation [29]. In this section, we define Gaussian graphical models penalized respectively by ℓ_1(denoted Glasso), by SLOPE (denoted Gslope) and ℓ_0 (denoted Gl0).

Glasso problem definition

As mentioned in the previous section, Glasso is nowadays widely used for estimating sparse inverse covariance, especially for recovering the topology structure. As defined in equation (4), the classical log-likelihood problem is penalized by $h(\mathbf{\Theta})$. If the penalized item is set to the ℓ_1 penalization, then it is called the Glasso problem.

$$\max_{\mathbf{\Theta} \in \mathcal{S}_p^{++}} \log\det(\mathbf{\Theta}) - \mathrm{tr}(\mathbf{S\Theta}) - \lambda\|\mathbf{\Theta}\|_{1,\text{off}} \tag{5}$$

Gslope problem definition

The graphical model based on the SLOPE penalty, denoted by Gslope, was introduced by [11]. The penalty is applied directly to the Gaussian log-likelihood. Similar to Glasso, Gslope replaces the ℓ_1 norm by the SLOPE penalty on the precision matrix $\mathbf{\Theta}$. In order to do this, the upper triangular part of $\mathbf{\Theta}$ is vectorized, which creates a new vector $\boldsymbol{\theta} \in \mathbb{R}^m$, where

$$\boldsymbol{\theta} = \mathrm{vec}_{\mathrm{up}}(\mathbf{\Theta}) = [\Theta_{1,2}, \ldots, \Theta_{1,p}, \Theta_{2,3}, \ldots, \Theta_{2,p}, \ldots, \Theta_{p-1,p}]^\top, \qquad m = \frac{p(p-1)}{2}.$$

We then define the sorted absolute value vector

$$\bar{\boldsymbol{\theta}} = \mathrm{sort}\Big(|\boldsymbol{\theta}|\Big) = [\bar{\theta}_1, \ldots, \bar{\theta}_m]^\top, \qquad \bar{\theta}_1 \geq \cdots \geq \bar{\theta}_m \geq 0.$$

$$h_\lambda(\mathbf{\Theta}) = h_\lambda(\boldsymbol{\theta}) = \sum_{i=1}^{m} \lambda_i \, \bar{\theta}_i. \tag{6}$$

each of the absolute values of $\boldsymbol{\theta}$ is assigned a specific tuning parameter such that $\lambda_1 \geq \cdots \geq \lambda_m \geq 0$.

Building on the penalty defined in (6), Gslope solves the following optimization problem:

$$\max_{\boldsymbol{\Theta} \in \mathcal{S}_p^{++}} \log\det(\boldsymbol{\Theta}) - \operatorname{tr}(\boldsymbol{\Theta}\mathbf{S}) - h_\lambda(\boldsymbol{\Theta}), \tag{7}$$

where $\mathbf{S}$ is the sample covariance matrix defined in (2).

Just like Glasso, Gslope is a convex optimization problem that can be solved efficiently. In Section 3.3, we provide a detailed formulation to solve Gslope using the Alternating Direction Method of Multipliers (ADMM) [30].

Gl0 problem definition

Another natural sparsity-promoting penalty is the so-called ℓ_0 penalty. In Gaussian graphical models, sparsity is imposed on the precision matrix rather than on the data matrix itself. For a precision matrix $\boldsymbol{\Theta} \in \mathbb{S}_{++}^p$, the off-diagonal ℓ_0 penalty is defined as

$$\|\boldsymbol{\Theta}\|_{0,\text{off}} = \sum_{i \neq j} \mathbb{I}(\Theta_{ij} \neq 0). \tag{8}$$

where the item $\sum_{i \neq j} \mathbb{I}(\Theta_{ij} \neq 0)$ indicates the number of all non-zero items. Then the ℓ_0-penalized log-likelihood problem is defined as

$$\max_{\boldsymbol{\Theta} \in S_p^{++}} \log\det(\boldsymbol{\Theta}) - \operatorname{tr}(\boldsymbol{\Theta}\mathbf{S}) - \lambda\|\boldsymbol{\Theta}\|_{0,\text{off}} \tag{9}$$

where $\lambda > 0$ is a tuning parameter.

3 Methodology

The Alternating Direction Method of Multipliers (ADMM) is a simple but powerful algorithm well suited to distributed convex optimization [30], in particular for problems arising in applied statistics and machine learning. It was originally proposed by [31] and later extended to sparse covariance estimation problems by [30]. It can be viewed as a decomposition–coordination procedure, in which the solutions to small local subproblems are coordinated to solve a larger global problem.

In this section, we describe how ADMM worked and then how this algorithm is applied to the three models considered in this report: Glasso, Gslope, and Gl0. We start from the generic penalized log-likelihood problem in (3), in which the penalty $h(\boldsymbol{\Theta})$ changes according to the estimator considered.

3.1 The alternating direction method of multipliers(ADMM)

We now consider the opposite problem defined in equation (3). The three graphical models considered can all be written under the unified form

$$\min_{\boldsymbol{\Theta} \in S_p^{++}} -\log\det(\boldsymbol{\Theta}) + \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + h(\boldsymbol{\Theta}) \tag{10}$$

where $h(\boldsymbol{\Theta})$ denotes a penalty term.

To apply the Alternating Direction Method of Multipliers (ADMM), we introduce an auxiliary variable $\boldsymbol{Y}$ and rewrite (10) as

$$\begin{aligned} \min_{\boldsymbol{\Theta},\boldsymbol{Y}} \quad & -\log\det(\boldsymbol{\Theta}) + \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \mathbb{I}[\boldsymbol{\Theta} \succ 0] + h(\boldsymbol{Y}) \\ \text{s.t.} \quad & \boldsymbol{\Theta} = \boldsymbol{Y}. \end{aligned} \tag{11}$$

The augmented Lagrangian associated with (11) is given by

$$\mathcal{L}_\rho(\boldsymbol{\Theta}, \boldsymbol{Y}, \boldsymbol{U}) = -\log\det(\boldsymbol{\Theta}) + \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \mathbb{I}[\boldsymbol{\Theta} \succ 0] + h(\boldsymbol{Y}) + \frac{\rho}{2}\|\boldsymbol{\Theta} - \boldsymbol{Y} + \boldsymbol{U}\|_F^2 - \frac{\rho}{2}\|\boldsymbol{U}\|_F^2, \tag{12}$$

where $\boldsymbol{U}$ is the scaled dual variable, $\rho > 0$ is the ADMM penalty parameter and $\|\cdot\|_F$ is the Frobenius norm.

Using the scaled dual variable $\boldsymbol{U}$, the ADMM iterations take the unified form

$$\boldsymbol{\Theta}^{(k+1)} = \arg\min_{\boldsymbol{\Theta} \in S_p^{++}} \left\{ -\log\det(\boldsymbol{\Theta}) + \operatorname{tr}(\boldsymbol{S}\boldsymbol{\Theta}) + \frac{\rho}{2}\|\boldsymbol{\Theta} - \boldsymbol{Y}^{(k)} + \boldsymbol{U}^{(k)}\|_F^2 \right\}, \tag{13}$$

$$\boldsymbol{Y}^{(k+1)} = \arg\min_{\boldsymbol{Y}} \left\{ h(\boldsymbol{Y}) + \frac{\rho}{2}\|\boldsymbol{\Theta}^{(k+1)} - \boldsymbol{Y} + \boldsymbol{U}^{(k)}\|_F^2 \right\}, \tag{14}$$

$$\boldsymbol{U}^{(k+1)} = \boldsymbol{U}^{(k)} + \boldsymbol{\Theta}^{(k+1)} - \boldsymbol{Y}^{(k+1)}. \tag{15}$$

The update of $\boldsymbol{\Theta}$ admits the same closed form for the three models. Define

$$\boldsymbol{A}^{(k)} = \rho\left(\boldsymbol{Y}^{(k)} - \boldsymbol{U}^{(k)}\right) - \boldsymbol{S} \tag{16}$$

we can apply the decomposition of the eigenvalue to $\mathbf{A}$

$$\boldsymbol{A}^{(k)} = \boldsymbol{Q}\operatorname{diag}(d_1, \ldots, d_p)\boldsymbol{Q}^\top. \tag{17}$$

Then

$$\boldsymbol{\Theta}^{(k+1)} = \boldsymbol{Q}\operatorname{diag}\left(\frac{d_i + \sqrt{d_i^2 + 4\rho}}{2\rho}\right)\boldsymbol{Q}^\top. \tag{18}$$

The concrete algorithm is shown in Algorithm 1.

The model dependent proximal step $\mathcal{P}_M$ is detailed in the following subsections.

3.2 Solving Glasso with ADMM

The ADMM algorithm has been successfully applied to the Glasso optimization problem, and its general form can be found in [30]. Specifically, the Glasso problem solves

$$\min_{\boldsymbol{\Theta} \in \mathcal{S}_p^{++}} \; -\log\det(\boldsymbol{\Theta}) + \operatorname{tr}(\mathbf{S}\boldsymbol{\Theta}) + \lambda\|\boldsymbol{\Theta}\|_1, \tag{19}$$

where

$$\|\boldsymbol{\Theta}\|_1 = \sum_{i \neq j} |\Theta_{ij}|$$

is the off-diagonal ℓ_1 penalty and $\lambda > 0$ is the tuning parameter.

Within the unified ADMM scheme above, the model dependent penalty is

$$h(\boldsymbol{Y}) = \lambda\|\boldsymbol{Y}\|_1 \tag{20}$$

This formulation is classical and described in detail in [30].

Algorithm 1 Unified ADMM solver for Glasso, Gslope and Gl0

Input: sample covariance matrix $\mathbf{S}$, model $M \in \{\text{GLASSO}, \text{GSLOPE}, \text{GL0}\}$, penalty parameter η, ADMM parameter $\rho > 0$, tolerances $\varepsilon_{\text{pri}}, \varepsilon_{\text{dual}}$

Output: estimated precision matrix $\widehat{\boldsymbol{\Theta}}$

Initialize $\boldsymbol{\Theta}^{(0)} = \mathbf{I}_p$, $\mathbf{Y}^{(0)} = \mathbf{I}_p$, $\mathbf{U}^{(0)} = \mathbf{0}$
for $k = 0, 1, 2, \ldots$ **do**
  $\mathbf{A}^{(k)} \leftarrow \rho\Big(\mathbf{Y}^{(k)} - \mathbf{U}^{(k)}\Big) - \mathbf{S}$
  Compute the eigendecomposition $\mathbf{A}^{(k)} = \mathbf{Q}\,\mathrm{diag}(d_1, \ldots, d_p)\mathbf{Q}^\top$
  $\boldsymbol{\Theta}^{(k+1)} \leftarrow \mathbf{Q}\,\mathrm{diag}\left(\dfrac{d_i + \sqrt{d_i^2 + 4\rho}}{2\rho}\right)\mathbf{Q}^\top$
  $\mathbf{W}^{(k)} \leftarrow \boldsymbol{\Theta}^{(k+1)} + \mathbf{U}^{(k)}$
  $Y_{ii}^{(k+1)} \leftarrow W_{ii}^{(k)}$ for all i
  $\mathbf{Y}_{\text{off}}^{(k+1)} \leftarrow \mathcal{P}_M\Big(\mathbf{W}_{\text{off}}^{(k)}; \eta, \rho\Big)$
  $\mathbf{U}^{(k+1)} \leftarrow \mathbf{U}^{(k)} + \boldsymbol{\Theta}^{(k+1)} - \mathbf{Y}^{(k+1)}$
  if $\|\boldsymbol{\Theta}^{(k+1)} - \mathbf{Y}^{(k+1)}\|_F \le \varepsilon_{\text{pri}}$ **and** $\rho\|\mathbf{Y}^{(k+1)} - \mathbf{Y}^{(k)}\|_F \le \varepsilon_{\text{dual}}$ **then**
    break
  end if
end for
return $\widehat{\boldsymbol{\Theta}} \leftarrow \boldsymbol{\Theta}^{(k+1)}$

3.3 Solving Gslope with ADMM

The ADMM formulation for Gslope is discussed in [11]. We also consider to start from its opposite problem, i.e., the minimization problem, the constrained reformulation is (7):

$$\begin{cases} \min\limits_{\boldsymbol{\Theta}, \boldsymbol{Y}} & -\log\det(\boldsymbol{\Theta}) + \mathrm{tr}(\boldsymbol{\Theta}\mathbf{S}) + \mathbb{I}[\boldsymbol{\Theta} \succ 0] + h_\lambda(\boldsymbol{Y}), \\ \text{s.t.} & \boldsymbol{\Theta} = \boldsymbol{Y}. \end{cases} \tag{21}$$

Within the unified ADMM scheme above, the model dependent penalty is

$$h(\boldsymbol{Y}) = h_\lambda(\boldsymbol{Y}) = \sum_{i=1}^{m} \lambda_i\, \bar{y}_i. \tag{22}$$

Accordingly, the update of $\boldsymbol{Y}$ is obtained from the generic step (14) by setting $h = h_\lambda$.

In our implementation, we also compare another strategies associated with the penalty parameter ρ: an adaptive-ρ version ADMM, which is called **Hybrid** in [32]. It adjusts ρ according to the usual primal and dual residual balancing rule,

$$\begin{cases} \text{increase } \rho, & \text{if } \|r^{(k)}\| \gg \|s^{(k)}\|, \\ \text{decrease } \rho, & \text{if } \|s^{(k)}\| \gg \|r^{(k)}\|, \end{cases} \tag{23}$$

where the primal and dual residuals are defined as

$$r^{(k)} = \boldsymbol{\Theta}^{(k)} - \boldsymbol{Y}^{(k)}, \qquad s^{(k)} = \rho\Big(\boldsymbol{Y}^{(k)} - \boldsymbol{Y}^{(k-1)}\Big). \tag{24}$$

3.4 Solving Gl0 with non-convex ADMM

Although the ℓ_0 norm is natural, easy to understand, and intuitive to implement, its lack of convexity and its computational cost have limited its use in sparse maximum likelihood

estimation [29]. Many algorithms have been proposed to solve ℓ_0-penalized problems. The package `El0ps` [33] is a recently proposed solver for ℓ_0 problems; however, due to its structural assumptions on the original objective, it cannot be directly applied to the Gaussian graphical model log-likelihood estimation problem, as discussed in the Appendix. In this study, we consider a non-convex ADMM formulation, which can efficiently handle the ℓ_0-penalized log-likelihood problem. A related approach is discussed in [34].

By introducing an auxiliary variable $\boldsymbol{Y}$, the log-likelihood estimation problem in (9) becomes

$$\min_{\boldsymbol{\Theta}, \boldsymbol{Y}} -\log\det(\boldsymbol{\Theta}) + \operatorname{tr}(\boldsymbol{\Theta}\mathbf{S}) + \lambda\|\boldsymbol{Y}\|_{0,\text{off}} \quad \text{s.t.} \quad \boldsymbol{\Theta} = \boldsymbol{Y}. \tag{25}$$

Within the unified ADMM scheme above, the model dependent penalty is

$$h(\boldsymbol{Y}) = \lambda\|\boldsymbol{Y}\|_{0,\text{off}}, \qquad Y_{ij}^{k+1} = \begin{cases} 0, & |W_{ij}^k| \le \sqrt{2\lambda/\rho}, \\ W_{ij}^k, & |W_{ij}^k| > \sqrt{2\lambda/\rho}, \end{cases} \qquad i \neq j, \tag{26}$$

while the diagonal entries remain unchanged, it facilitates computation.

4 Datasets

4.1 Synthetic dataset

For a fixed dimension p, we consider four standard random graph models described in Table 1. Given a sampled support of the graph, weights of the edges are sampled from a uniform distribution $\mathcal{U}(2, 5)$, which yields a symmetric weighted adjacency matrix $\mathbf{A}$ for each random graph. Figure 2 shows examples of sampled graph structures and their adjacency and Laplacian matrices. The Laplacian matrix of the graph is then used to construct the precision matrix $\boldsymbol{\Theta}$ through the linear transformation

$$\boldsymbol{\Theta} = \boldsymbol{L} + \kappa\mathbf{I}, \tag{27}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ is the Laplacian matrix and $\kappa = 10^{-1}$ is used so that $\boldsymbol{\Theta}$ is non-singular. A total of n samples are drawn for each graph model. In the final experiments reported here, the density generator is a zero-mean Gaussian distribution.

Graph Model	Edge Weights	Probability	Neighbors	Radius
Barabasi–Albert	$\mathcal{U}(2, 5)$	–	–	–
Erdos–Renyi	$\mathcal{U}(2, 5)$	0.1	–	–
Watts–Strogatz	$\mathcal{U}(2, 5)$	0.1	5	–
Random geometric	$\mathcal{U}(2, 5)$	–	–	0.2

Table 1: Four graph models used for numerical experiments.

4.2 S&P500 index dataset

After the experiments on synthetic datasets, we conduct experiments on real-world financial data to evaluate the performance of the different penalties. The dataset consists of $p = 30$ stocks selected from the S&P 500 index over the period ranging from January

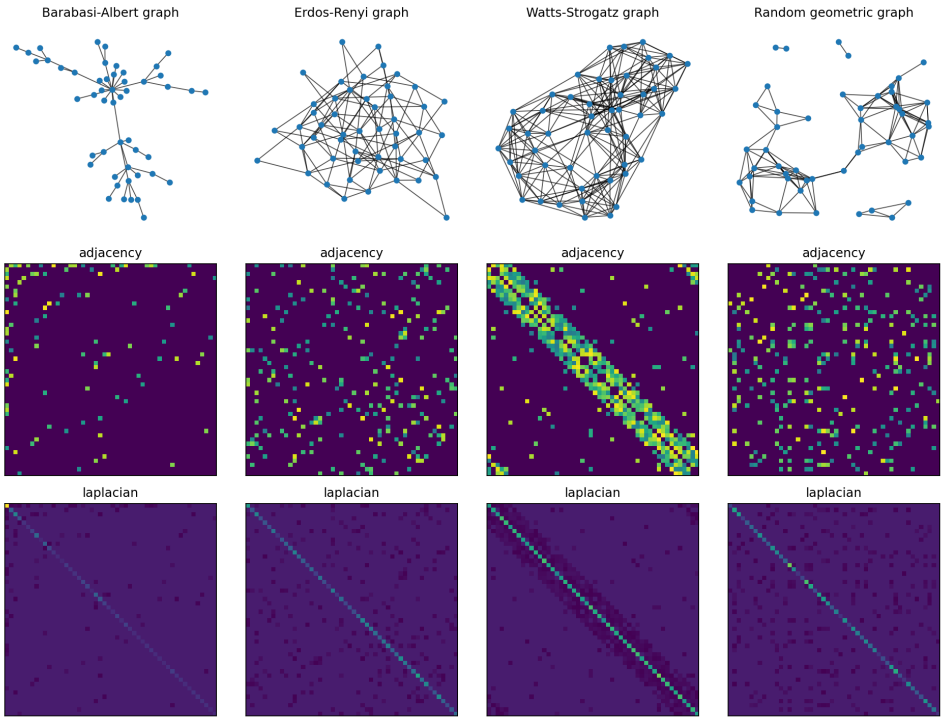


Figure 2: Four graph structures used with $p = 50$ and associated adjacency and Laplacian matrices.

2018 to July 2018. This time span yields $n = 200$ daily observations for each stock. The data are represented by a matrix of log-returns, defined entry-wise as

$$X_{i,j} = \log P_{i,j} - \log P_{i,j-1}, \tag{28}$$

where $P_{i,j}$ denotes the closing price of the i-th stock on day j. The $p = 30$ stocks are grouped into three sectors: Utilities, Materials, and Health Care. The ground-truth sector labels are obtained according to the Global Industry Classification Standard (GICS). The illustration of this dataset is in figure 3.

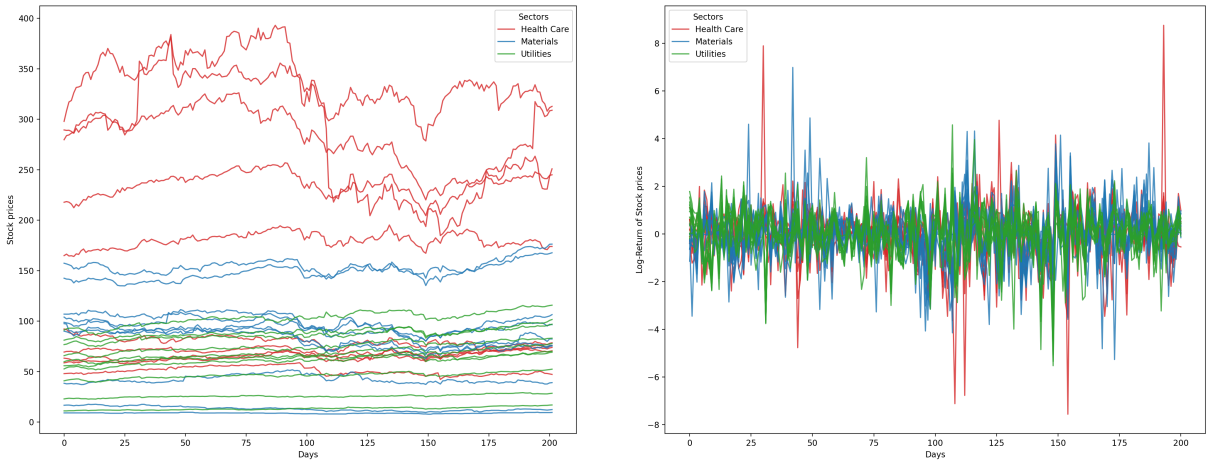


Figure 3: An illustration of S&P500 dataset. Left: Original stock price. Right: Log form of stock price.

5 Experiments and Results

For a given setup, different graph learning algorithms are applied to the input data $\mathbf{X}$. For a meaningful comparison on the synthetic datasets, we evaluate the receiver operating characteristic (ROC) curves obtained from edge scores derived from the estimated precision matrix. The ROC curve displays the true positive rate (TPR) as a function of the false positive rate (FPR). Here, the TPR denotes the ability of the algorithm to recover actual edges, whereas the FPR accounts for the false discovery of non-existing edges. The ROC curves are obtained as follows: we first compute the conditional correlation coefficients as shown in (1); then given these coefficient estimates, a ROC curve is obtained by varying the threshold tol, i.e., the edge (i, j) is considered active whenever $|\widetilde{\Theta}_{ij}| > \text{tol}$. The displayed main ROC curves are based on 50 Monte-Carlo experiments.

5.1 Results on synthetic dataset

For each synthetic graph, we compare the ROC curves for different numbers of observations n, while fixing $p = 50$ and varying $n \in \{2p, 5p, 8p, 10p\}$. The parameter λ for the different methods is selected by continuous cross-validation over a bounded interval. The table 2 shows how the parameter is tuned.

Model	λ Interval
Glasso	$[0.1, 10]$
Gslop(ADMM, Hybrid)	$[0.01, 10]$
Gl0	$[0.01, 10]$

Table 2: The Tuning interval for each model

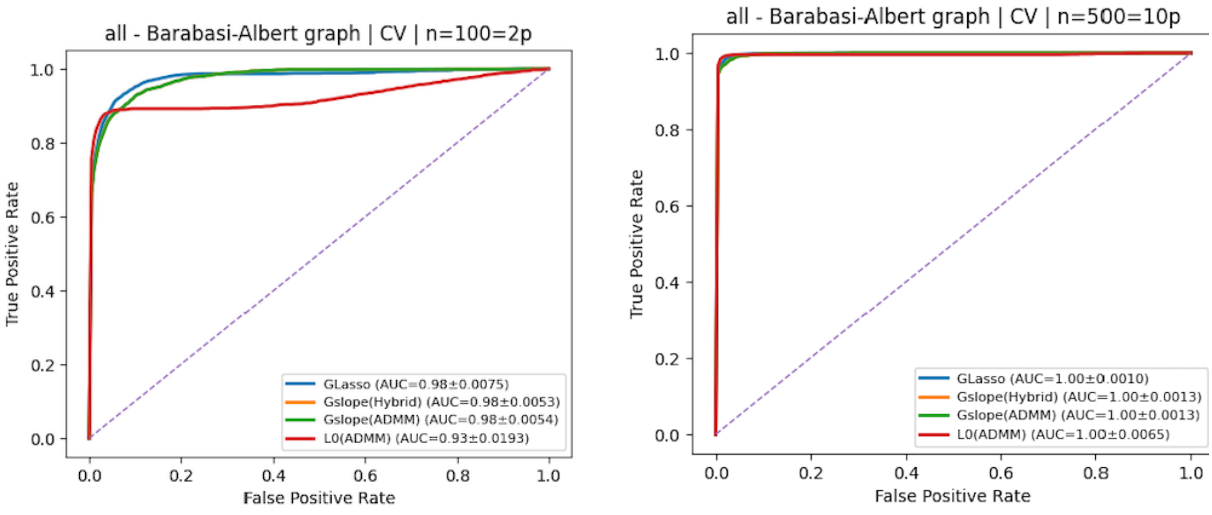


Figure 4: ROC curves for the Barabasi–Albert (BA) graph after parameter tuning with $n = 2p$(left) and $n = 10p$(right). The ℓ_0 norm performs better when FPR is low. Both Gslope and ℓ_0 demonstrate a competitive recovery ability.

From Figure 4, both proposed methods, Gslope and ℓ_0, demonstrate competitive ability to recover the topology structure. In particular, the ℓ_0 penalty performs better at small FPR when n is small ($2p$ and $5p$). Overall, when n is low, all methods show a

more moderate performance than in the larger-sample regimes. With n increasing, the performance gap between the proposed methods and Glasso becomes thinner.

Figure 5 shows the robustness analysis of the different methods with respect to the tuning parameter. Overall, ℓ_0 appears to be the most sensitive to parameter changes, while Glasso is also sensitive but to a lesser extent, Gslope is the less sensitive comapred with other two. As n increases, all methods become less sensitive, which indicates a more robust performance.

Similar results are observed for the three other synthetic graph families. The corresponding results are reported in the Appendix due to space limitations.

5.2 Results on real-world dataset

For the S&P500 dataset, we evaluate the performance of the proposed methods on a stock clustering task, with the objective of recovering the three underlying market sectors. Clustering performance is assessed using modularity (MOD). Additionally, edge detection performance is evaluated through the F-score (FS), defined as

$$\mathrm{FS} = \frac{2\ \mathrm{TP}}{2\,\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}, \tag{29}$$

where TP, FP, and FN denote the numbers of true positives, false positives, and false negatives, respectively. The F-score takes values in the interval $[0, 1]$, with $\mathrm{FS} = 1$ indicating perfect recovery of the underlying proxy graph structure. In this evaluation, the ground-truth graph is assumed to contain edges only between stocks belonging to the same sector. The parameter tuning on the real-world dataset follows the same practical optimization procedure as on the synthetic dataset.

From figure 6 and 7, Glasso presents the best overall clustering performance with the fewest false edges. The two Gslope implementations show almost the same results and are slightly worse than Glasso, with a larger number of false edges. The ℓ_0 estimator shows the most distinctive behavior: there are almost no false edges, but some correct edges are also suppressed. This may be due to the stronger sparsity induced by the ℓ_0 penalty. This indicates that the ℓ_0 penalty tends to reduce the false-positive ratio. Overall, the Utilities sector is more separable from the two other sectors. However, the Health Care and Materials sectors tend to be mixed in this case. This may be because the companies in the Materials and Health Care sectors are more highly correlated in this time period. Notably, AMCR does not appear correlated with any other company for any of the considered methods.

6 Conclusion

In this study, we investigate two alternative penalty terms for topology recovery in Gaussian graphical models, namely the SLOPE penalty and the ℓ_0 penalty, and compare them with Glasso. Gslope and Gl0 are designed for estimating Gaussian graphical models with SLOPE and ℓ_0 penalties respectively. We use ADMM to solve Gslope and a non-convex ADMM scheme to solve Gl0. We also implement continuous parameter tuning strategies for both methods. Experiments are performed on both synthetic datasets and a real-world dataset. For the synthetic datasets, our results show that both proposed methods, Gslope and Gl0, demonstrate competitive ability to recover the graph topology. The sensitivity analysis shows that Gl0 is particularly sensitive to the tuning parameter and

that this sensitivity can be mitigated by increasing the number of observations. For the real-world dataset, all three considered methods show the ability to separate the Utilities sector, while the Health Care and Materials sectors remain difficult to distinguish for all methods.

Concerning future work, Gl0 shows tendency to suppress the false-positive ratio, i.e., Gl0 tends to return a highly sparse structure with a reduced number of edges, which deserves further investigation. Although non-convex ADMM can solve Gl0 efficiently, it remains computationally costly. A more efficient algorithm may therefore be considered in the future. Overall, this study highlights the relevance of both SLOPE and ℓ_0 penalties for promoting sparse precision matrices in topology recovery.

7 Appendix

`El0ps` cannot be directly applied to GGMs log-likelihood estimation.
Although `El0ps` is an efficient solver for ℓ_0 problems, it cannot be applied directly to the Gaussian graphical model log-likelihood estimation problem. There are two main reasons.

First, `El0ps` assumes that the problem is formulated as

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(\mathbf{A}\mathbf{x}) + \lambda\|\mathbf{x}\|_0 + \sum_{i=1}^{d} h_i(x_i),$$

where $\mathbf{x} \in \mathbb{R}^d$ is an unconstrained variable, whereas in Gaussian graphical model log-likelihood estimation the optimized variable $\mathbf{\Theta} \in \mathcal{S}_{++}$ is required to be positive definite.

The second reason is the most important one: every entry in the gradient of the term $\log\det(\mathbf{\Theta})$, namely $-\mathbf{\Theta}^{-1}$, depends on the full matrix $\mathbf{\Theta}$. Hence the likelihood is not separable by coordinates, which is contrary to the structural assumptions required by `El0ps`.

Another idea would be to decompose $\mathbf{\Theta}$ into blocks so that `El0ps` could be applied block-wise, but such a strategy would make the computation substantially more complex than the non-convex ADMM approach considered here.

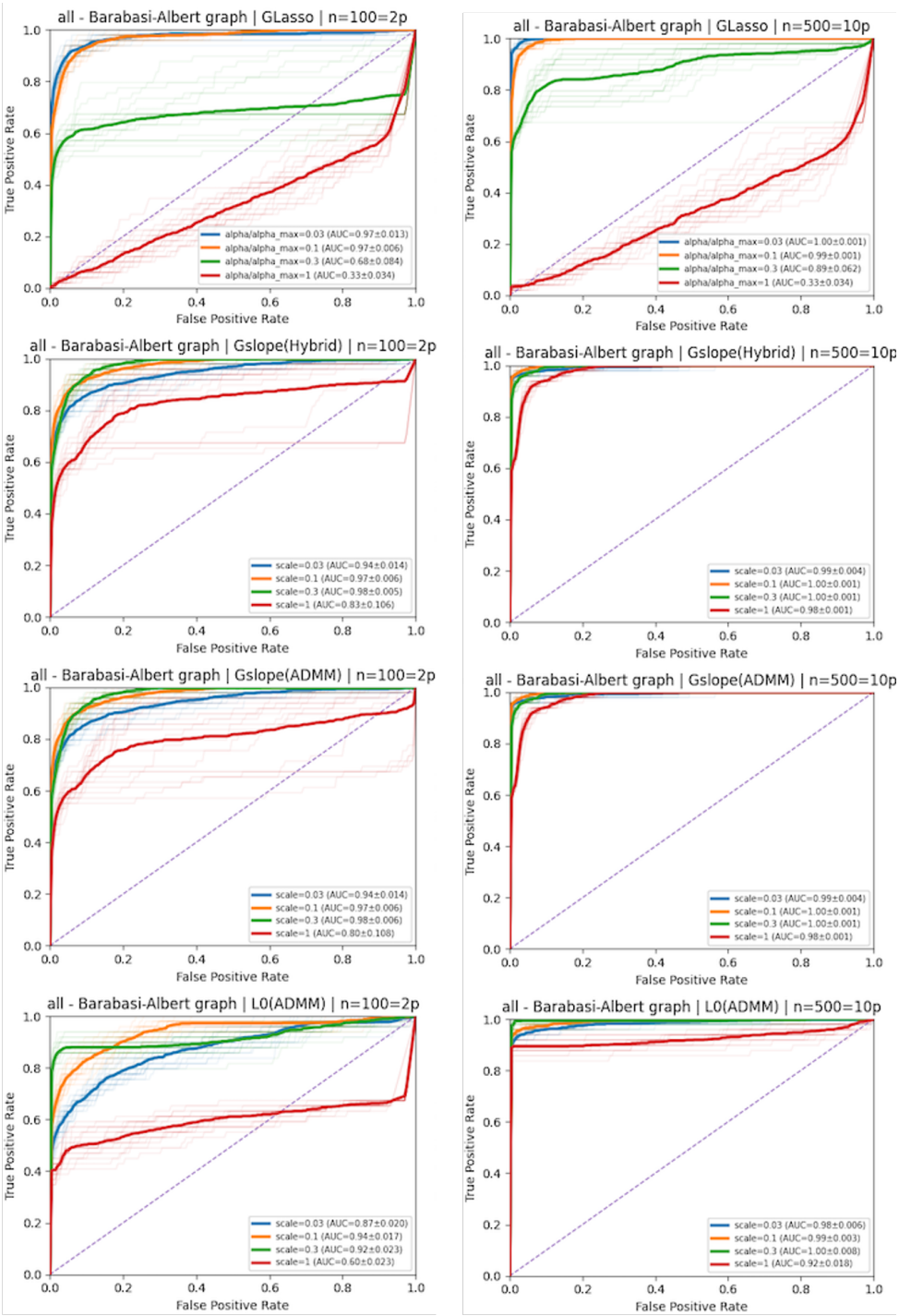
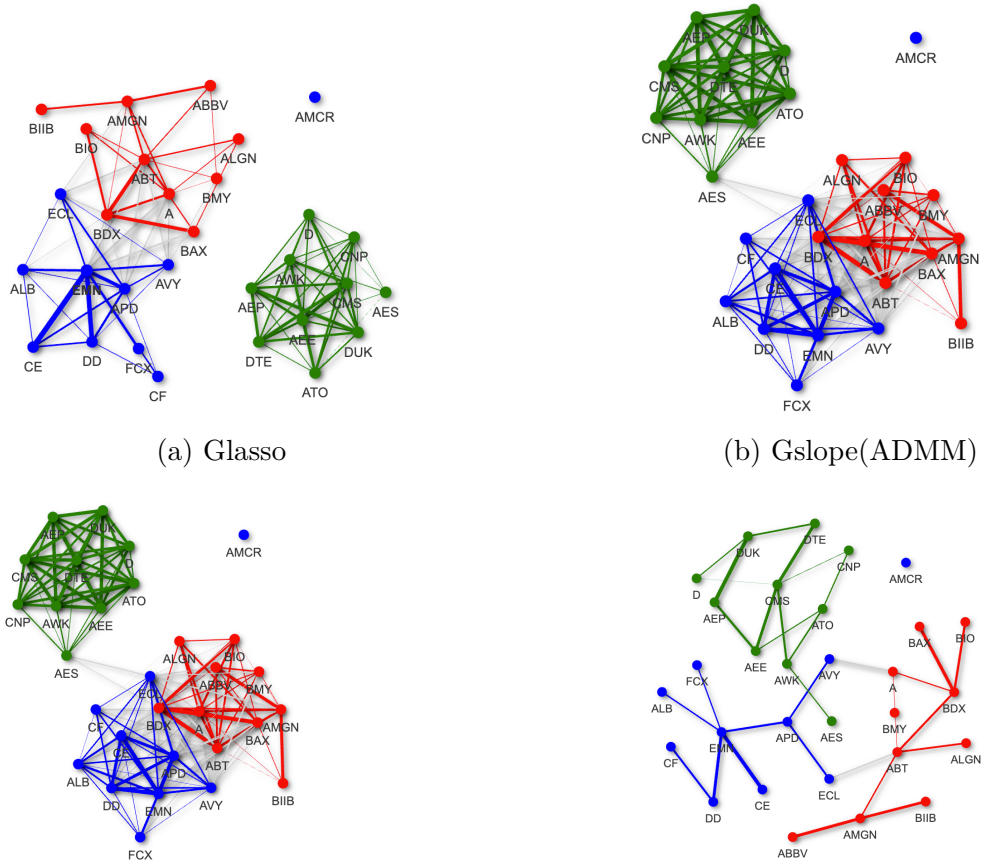


Figure 5: Sensitivity analysis of different methods on the BA graph with different $n = 2p$(left) and $n = 5p$(right) and different methods, Glasso,Gslope(Hybrid),Gslope(ADMM),Gl0(ADMM) from top to down. Not always but we oberserve that Glslops is less sensitive.



(a) Glasso

(b) Gslope(ADMM)

(c) Gslope(Hybrid)

(d) Gl0

Figure 6: Results of clustering for Glasso (upper left), Gslope with ADMM (upper right), Gslope with Hybrid(lower left), and ℓ_0 (lower right), each displayed with its best parameter. Green stands for Utilities, red for Health Care, and blue for Materials.

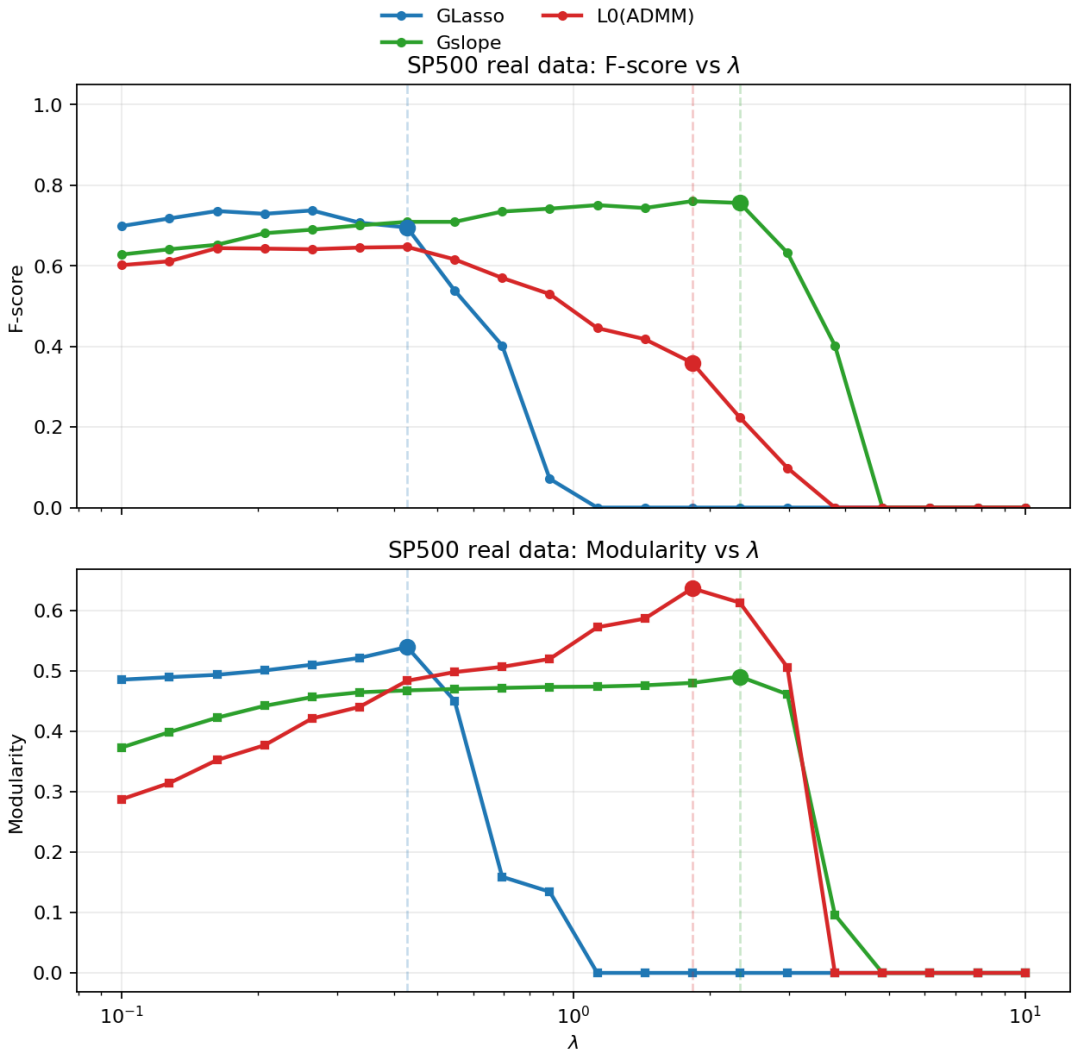


Figure 7: Computed modularity and FS score. The Gslope tends to given a higher F-score while ℓ_0 given the highest modularity, indicating that ℓ_0 divided the highest community structure. Noting that the curve of Gslope(Hybrid) is totally same with Gslope(ADMM).

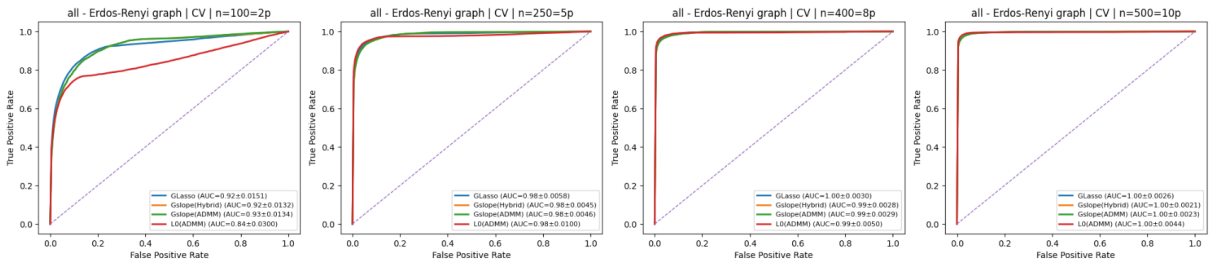


Figure 8: ROC curves for the Erdos–Renyi (ER) graph after parameter tuning.

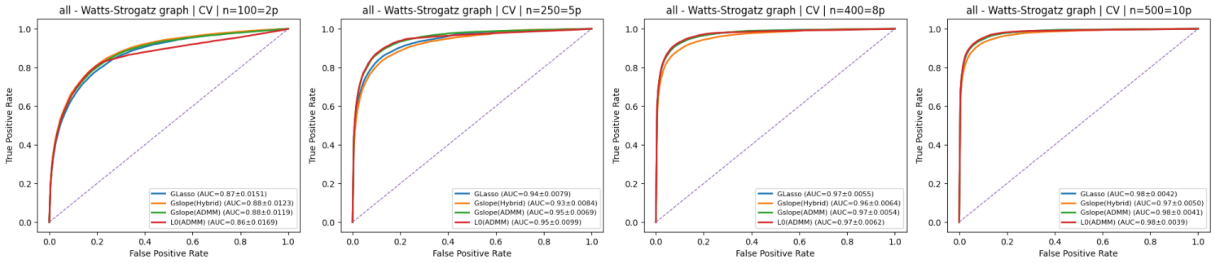


Figure 9: ROC curves for the Watts–Strogatz (WS) graph after parameter tuning.

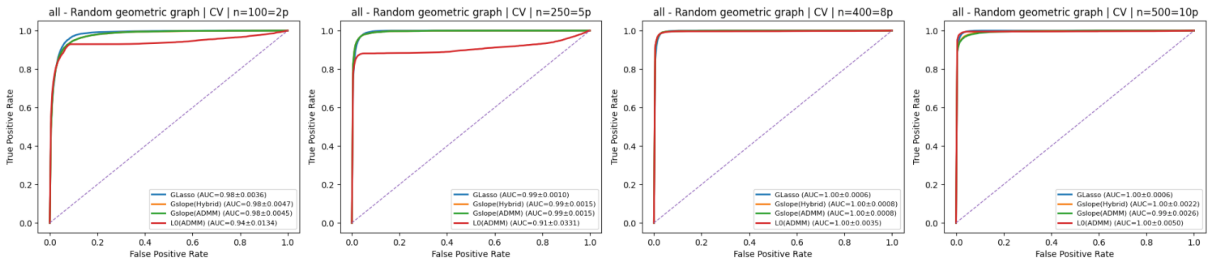


Figure 10: ROC curves for the Random Geometric (RG) graph after parameter tuning.

References

[1] Alexandre Hippert-Ferrer, Florent Bouchard, Ammar Mian, Titouan Vayer, and Arnaud Breloy. Learning graphical factor models with riemannian optimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 349–366. Springer, 2023.

[2] Hongzhe Li and Jiang Gui. Gradient directed regularization for sparse gaussian concentration graphs, with applications to inference of genetic networks. *Biostatistics*, 7(2):302–317, 2006.

[3] Stephen M Smith, Karla L Miller, Gholamreza Salimi-Khorshidi, Matthew Webster, Christian F Beckmann, Thomas E Nichols, Joseph D Ramsey, and Mark W Woolrich. Network modelling methods for fmri. *Neuroimage*, 54(2):875–891, 2011.

[4] Oliver Stegle, Sarah A Teichmann, and John C Marioni. Computational and analytical challenges in single-cell transcriptomics. *Nature Reviews Genetics*, 16(3):133–145, 2015.

[5] Gautier Marti, Frank Nielsen, Mikołaj Bińkowski, and Philippe Donnat. A review of two decades of correlations, hierarchies, networks and clustering in financial markets. *Progress in information geometry: Theory and applications*, pages 245–274, 2021.

[6] David I Shuman, Sunil K Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. *IEEE signal processing magazine*, 30(3):83–98, 2013.

[7] Vassilis Kalofolias. How to learn a graph from smooth signals. In *Artificial intelligence and statistics*, pages 920–929. PMLR, 2016.

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

[9] Brenden Lake and Joshua Tenenbaum. Discovering structure by learning sparse graphs. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 32, 2010.

[10] Eugen Pircalabelu and Gerda Claeskens. Community-based group graphical lasso. *Journal of Machine Learning Research*, 21(64):1–32, 2020.

[11] Riccardo Riccobello, Malgorzata Bogdan, Giovanni Bonaccolto, Philipp J Kremer, Sandra Paterlini, and Piotr Sobczyk. Sparse graphical modelling via the sorted l _1-norm. *arXiv preprint arXiv:2204.10403*, 2022.

[12] Hirotugu Akaike. A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723, 2003.

[13] Gideon Schwarz. Estimating the dimension of a model. *The annals of statistics*, pages 461–464, 1978.

[14] Rina Foygel and Mathias Drton. Extended bayesian information criteria for gaussian graphical models. *Advances in neural information processing systems*, 23, 2010.

[15] Małgorzata Bogdan, Jayanta K Ghosh, and RW Doerge. Modifying the schwarz bayesian information criterion to locate multiple interacting quantitative trait loci. *Genetics*, 167(2):989–999, 2004.

[16] Małgorzata Żak-Szatkowska and Małgorzata Bogdan. Modified versions of the bayesian information criterion for sparse generalized linear models. *Computational Statistics & Data Analysis*, 55(11):2908–2924, 2011.

[17] Jiahua Chen and Zehua Chen. Extended bayesian information criteria for model selection with large model spaces. *Biometrika*, 95(3):759–771, 2008.

[18] David S Johnson. Approximation algorithms for combinatorial problems. In *Proceedings of the fifth annual ACM symposium on Theory of computing*, pages 38–49, 1973.

[19] Yagyensh Chandra Pati, Ramin Rezaiifar, and Perinkulam Sambamurthy Krishnaprasad. Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar conference on signals, systems and computers*, pages 40–44. IEEE, 1993.

[20] Fadil Santosa and William W Symes. Linear inversion of band-limited reflection seismograms. *SIAM journal on scientific and statistical computing*, 7(4):1307–1330, 1986.

[21] Shaobing Scott Chen. *Basis pursuit*. Stanford University, 1996.

[22] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 58(1):267–288, 1996.

[23] Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. 2006.

[24] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 67(2): 301–320, 2005.

[25] Sangkyun Lee, Piotr Sobczyk, and Malgorzata Bogdan. Structure learning of gaussian markov random fields with false discovery rate control. *Symmetry*, 11(10):1311, 2019.

[26] Steffen L Lauritzen. *Graphical models*, volume 17. Clarendon press, 1996.

[27] Arthur P Dempster. Covariance selection. *Biometrics*, pages 157–175, 1972.

[28] Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The annals of statistics*, pages 813–852, 2016.

[29] Goran Marjanovic and Alfred O Hero. $l_\{0\}$ sparse inverse covariance estimation. *IEEE Transactions on Signal Processing*, 63(12):3218–3231, 2015.

[30] Parikh Neal, Chu Eric, Peleato Borja, and Eckstein Jonathan. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1):1–122, 2011.

[31] Roland Glowinski and Americo Marroco. Sur l'approximation, par éléments finis d'ordre un, et la résolution, par pénalisation-dualité d'une classe de problèmes de dirichlet non linéaires. *Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique*, 9(R2):41–76, 1975.

[32] Johan Larsson, Malgorzata Bogdan, Krystyna Grzesiak, Mathurin Massias, and Jonas Wallin. Efficient solvers for slope in r, python, julia, and c++. *arXiv preprint arXiv:2511.02430*, 2025.

[33] Théo Guyard, Cédric Herzet, and Clément Elvira. El0ps: An exact l0-regularized problems solver. *arXiv preprint arXiv:2506.06373*, 2025.

[34] Zhenqiu Liu, Shili Lin, Nan Deng, Dermot PB McGovern, and Steven Piantadosi. Sparse inverse covariance estimation with l 0 penalty for network construction with omics data. *Journal of Computational Biology*, 23(3):192–202, 2016.