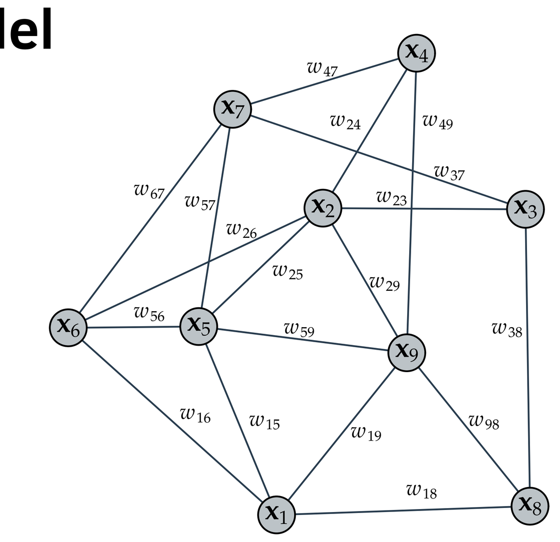
Nonconvex Sparse Graph Learning under Laplacian Constrained Graphical Model

Jiaxi Ying, José Vinícius de M. Cardoso, and Daniel P. Palomar

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

For more information visit: https://www.danielppalomar.com; R Package: https://github.com/mirca/sparseGraph

1 Graphical Model



- A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \boldsymbol{W})$ is an intuitive way to represent relationships between entities.
- Nodes: $\mathcal{V} = \{1, 2, \ldots, p\}$ correspond to the entities.
- Edges: $\mathcal{E} = \{(1,2), \ldots, (i,j), \ldots\}$ encodes conditional dependence between entities.
- Weights: $\boldsymbol{W}$ is weight matrix with W_{ij} the graph weight between node i and node j.

1.1 Laplacian Constrained Gaussian Graphical Model

- Graph Laplacian: $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$, where $\boldsymbol{D}$: the degree matrix with $D_{ii} = \sum_{j=1}^{p} W_{ij}$.
- $\mathcal{S}_L$: the set of all graph Laplacians for connected graphs,

$$\mathcal{S}_L := \{\boldsymbol{\Theta} \in \mathcal{S}_+^p | \, \Theta_{ij} = \Theta_{ji} \leq 0, \, \forall\, i \neq j, \, \boldsymbol{\Theta} \cdot \mathbf{1} = \mathbf{0}, \, \mathrm{rank}(\boldsymbol{\Theta}) = p - 1\}, \tag{1}$$

where $\mathbf{0}$ and $\mathbf{1}$ are the constant zero and one vectors.

Definition 1 (L-GMRF). A zero-mean random vector $\boldsymbol{x} = [x_1, \ldots, x_p]^\top \in V^{p-1}$ is called a Laplacian constrained Gaussian Markov Random Fields (L-GMRF) with parameters $(\mathbf{0}, \boldsymbol{\Theta})$ with $\boldsymbol{\Theta} \in \mathcal{S}_L$, if and only if its density function $q_L : V^{p-1} \to \mathbb{R}$ follows

$$q_L(\boldsymbol{x}) = (2\pi)^{-\frac{p-1}{2}} \det{}^\star(\boldsymbol{\Theta})^{\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \boldsymbol{\Theta} \boldsymbol{x}\right), \tag{2}$$

where $\det^\star$: the product of nonzero eigenvalues; $V^{p-1} := \{\boldsymbol{x} \in \mathbb{R}^p | \, \mathbf{1}^\top \boldsymbol{x} = 0\}$.

2 ℓ_1-norm Analysis and Proposed Method

2.1 ℓ_1-norm Does Not Work

The ℓ_1-norm regularized maximum likelihood estimation of Laplacian constrained precision matrices [1, 2] can be formulated as

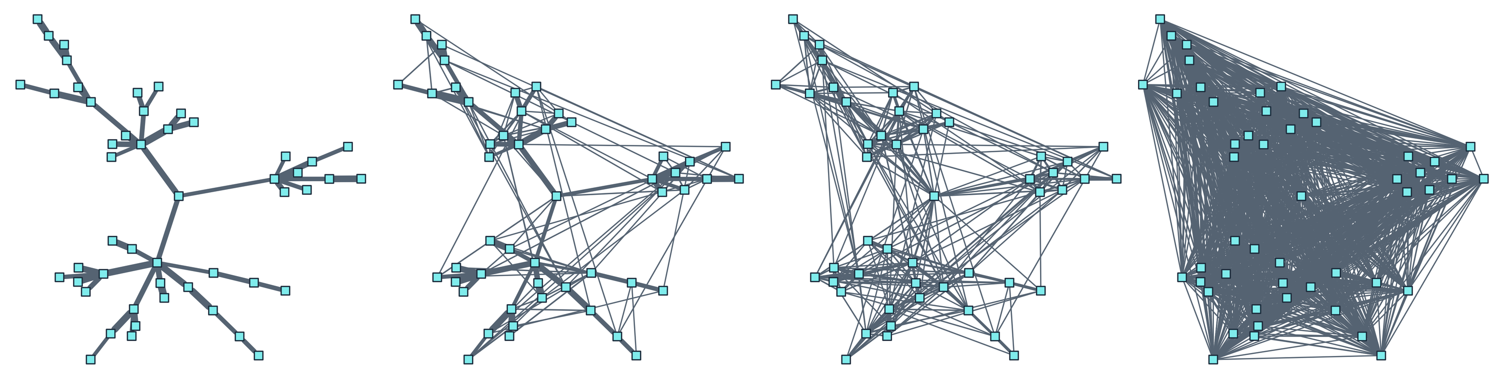
$$\min_{\boldsymbol{\Theta} \in \mathcal{S}_L} - \log\det(\boldsymbol{\Theta} + \boldsymbol{J}) + \mathrm{tr}\,(\boldsymbol{\Theta}\boldsymbol{S}) + \lambda \sum_{i>j} |\Theta_{ij}|, \tag{3}$$

where $\boldsymbol{S}$: sample covariance matrix; λ: regularization parameter; $\boldsymbol{J}$: constant matrix with each element equal to $\frac{1}{p}$.

Theorem 2. *Let $\hat{\Theta} \in \mathbb{R}^{p \times p}$ be the global minimum of* (3) *with $p > 3$. Define $s_1 = \max_k S_{kk}$ and $s_2 = \min_{ij} S_{ij}$. If the regularization parameter λ in* (3) *satisfies $\lambda \in [(2 + 2\sqrt{2})(p+1)(s_1 - s_2), +\infty)$, then the estimated graph weight $\hat{W}_{ij} = -\hat{\Theta}_{ij}$ obeys*

$$\hat{W}_{ij} \geq \frac{1}{(s_1 - (p+1)s_2 + \lambda)p} > 0, \quad \forall\, i \neq j.$$

Theorem 2 shows that a large regularization parameter of the ℓ_1-norm will make every graph weight strictly positive and the estimated graph is fully connected.



(a) Ground-truth (b) $\lambda = 0$ (c) $\lambda = 0.1$ (d) $\lambda = 10$

Figure 1: Graph learning using ℓ_1-norm with different regularization parameters. The number of nonzero edges in (a), (b), (c) and (d) are 49, 135, 286 and 1225.

2.2 Proposed Method

The penalized maximum likelihood of the precision matrix with Laplacian structural constraints can be formulated as

$$\min_{\boldsymbol{\Theta} \in \mathcal{S}_L} - \log\det(\boldsymbol{\Theta} + \boldsymbol{J}) + \mathrm{tr}\,(\boldsymbol{\Theta}\boldsymbol{S}) + \sum_{i>j} h_\lambda(\Theta_{ij}), \tag{4}$$

where h_λ: nonconvex sparsity-promoting function such as **SCAD** and **MCP**.

To handle the constraint $\boldsymbol{\Theta} \in \mathcal{S}_L$, we introduce a linear operator $\mathcal{L}$ [3] that maps a vector $\boldsymbol{w}$ to a Laplacian matrix $\mathcal{L}\boldsymbol{w}$. For example $\boldsymbol{w} = [w_1, w_2, w_3, w_4, w_5, w_6]^\top$,

$$\mathcal{L}\boldsymbol{w} = \begin{bmatrix} \sum_{i=1,2,3} w_i & -w_1 & -w_2 & -w_3 \\ -w_1 & \sum_{i=1,4,5} w_i & -w_4 & -w_5 \\ -w_2 & -w_4 & \sum_{i=2,4,6} w_i & -w_6 \\ -w_3 & -w_5 & -w_6 & \sum_{i=3,5,6} w_i \end{bmatrix}.$$

With the usage of the linear operator $\mathcal{L}$, the optimization (4) can be reformulated as

$$\min_{\boldsymbol{w} \geq \mathbf{0}} - \log\det(\mathcal{L}\boldsymbol{w} + \boldsymbol{J}) + \mathrm{tr}\,(\boldsymbol{S}\mathcal{L}\boldsymbol{w}) + \sum_i h_\lambda(w_i). \tag{5}$$

We establish a sequence $\{\hat{\boldsymbol{w}}^{(k)}\}_{k \geq 1}$ by solving a sequence of sub-problems

$$\hat{\boldsymbol{w}}^{(k)} = \arg\min_{\boldsymbol{w} \geq \mathbf{0}} - \log\det(\mathcal{L}\boldsymbol{w} + \boldsymbol{J}) + \mathrm{tr}\,(\boldsymbol{S}\mathcal{L}\boldsymbol{w}) + \sum_i h'_\lambda\left(\hat{w}_i^{(k-1)}\right) w_i. \tag{6}$$

The optimization (6) can be solved by a projected gradient descent algorithm with backtracking line search.

Algorithm 1 Nonconvex Graph Learning (NGL)

Input: Sample covariance $\boldsymbol{S}$, λ, $\hat{\boldsymbol{w}}^{(0)}$;
$k \leftarrow 1$;
1: **while** Stopping criteria not met **do**
2: Update $z_i^{(k-1)} = h'_\lambda(\hat{w}_i^{(k-1)})$, for $i = 1, \ldots, p(p-1)/2$;
3: Update $\hat{\boldsymbol{w}}^{(k)} = \arg\min_{\boldsymbol{w} \geq \mathbf{0}} - \log\det(\mathcal{L}\boldsymbol{w} + \boldsymbol{J}) + \mathrm{tr}\,(\boldsymbol{S}\mathcal{L}\boldsymbol{w}) + \sum_i z_i^{(k-1)} w_i$;
4: $k \leftarrow k + 1$;
5: **end while**
Output: $\hat{\boldsymbol{w}}^{(k)}$.

2.3 Theoretical Results

Assumption 3. The function $h_\lambda : \mathbb{R} \to \mathbb{R}$ satisfies the following conditions:
1. $h_\lambda(0) = 0$, and $h'_\lambda(x)$ is monotone and Lipschitz continuous for $x \in [0, +\infty)$;
2. There exists a $\gamma > 0$ such that $h'_\lambda(x) = 0$ for $x \geq \gamma\lambda$;
3. $h'_\lambda(x) = \lambda$ for $x \leq 0$ and $h'_\lambda(c\lambda) \geq \lambda/2$, where $c = (2 + \sqrt{2})\lambda_{\max}^2(\mathcal{L}\boldsymbol{w}^\star)$ is a constant.

Assumption 4. The minimal nonzero graph weight satisfies $\min_{i \in \mathcal{S}^\star} w_i^\star \geq (c + \gamma)\lambda \gtrsim \lambda$, where c and γ are defined in Assumption 3.

Theorem 5. *Under Assumptions 3 and 4, take the regularization parameter $\lambda = \sqrt{4\alpha c_0^{-1} \log p / n}$ for some $\alpha > 2$. If the sample size n is lower bounded by*

$$n \geq \max(94\alpha c_0^{-1} \lambda_{\max}^2(\mathcal{L}\boldsymbol{w}^\star) s \log p, 8\alpha \log p),$$

then with probability at least $1 - 1/p^{\alpha - 2}$, the sequence $\hat{\boldsymbol{w}}^{(k)}$ returned by Algorithm 1 satisfies

$$\left\|\hat{\boldsymbol{w}}^{(k)} - \boldsymbol{w}^\star\right\| \leq \underbrace{2(3\sqrt{2} + 4)\lambda_{\max}^2(\mathcal{L}\boldsymbol{w}^\star)\sqrt{\alpha c_0^{-1} s \log p / n}}_{\text{Statistical error}} + \underbrace{\left(\frac{3}{2 + \sqrt{2}}\right)^k \left\|\hat{\boldsymbol{w}}^{(0)} - \boldsymbol{w}^\star\right\|}_{\text{Optimization error}},$$

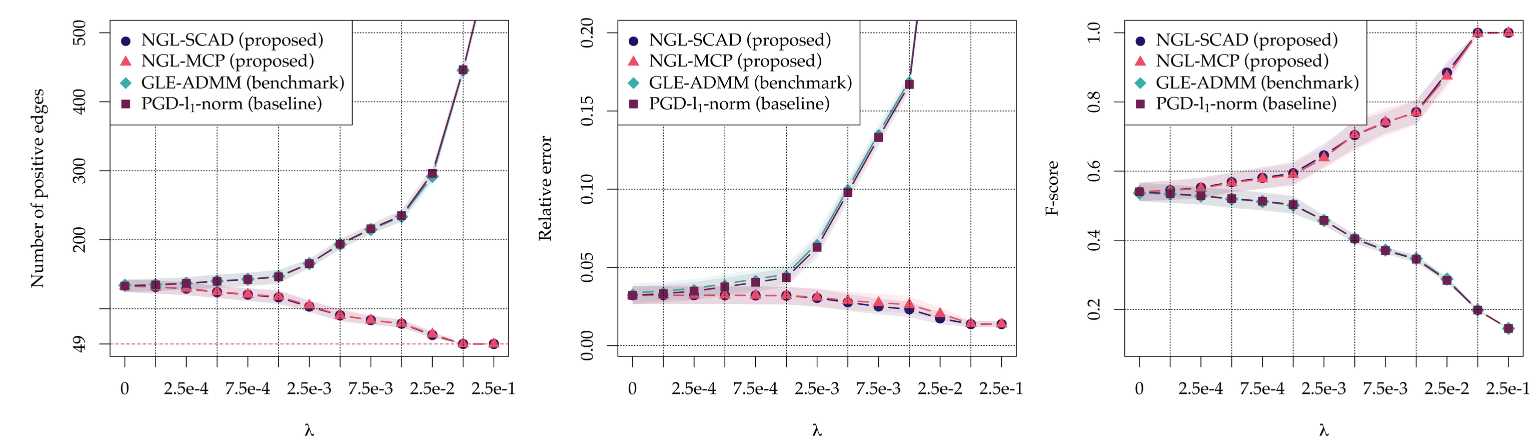
where $c_0 = 1/\left(8\left\|\mathcal{L}^*(\mathcal{L}\boldsymbol{w}^\star + \boldsymbol{J})^{-1}\right\|_{\max}^2\right)$ is a constant.

Theorem 5 shows that the estimation error is bounded by the optimization error and statistical error. The optimization error decays to zero at a linear rate with respect to the iteration number k. The statistical error is with the order of $\sqrt{s \log p / n}$, and a large sample size n will lead to a small statistical error.

3 Experimental Results

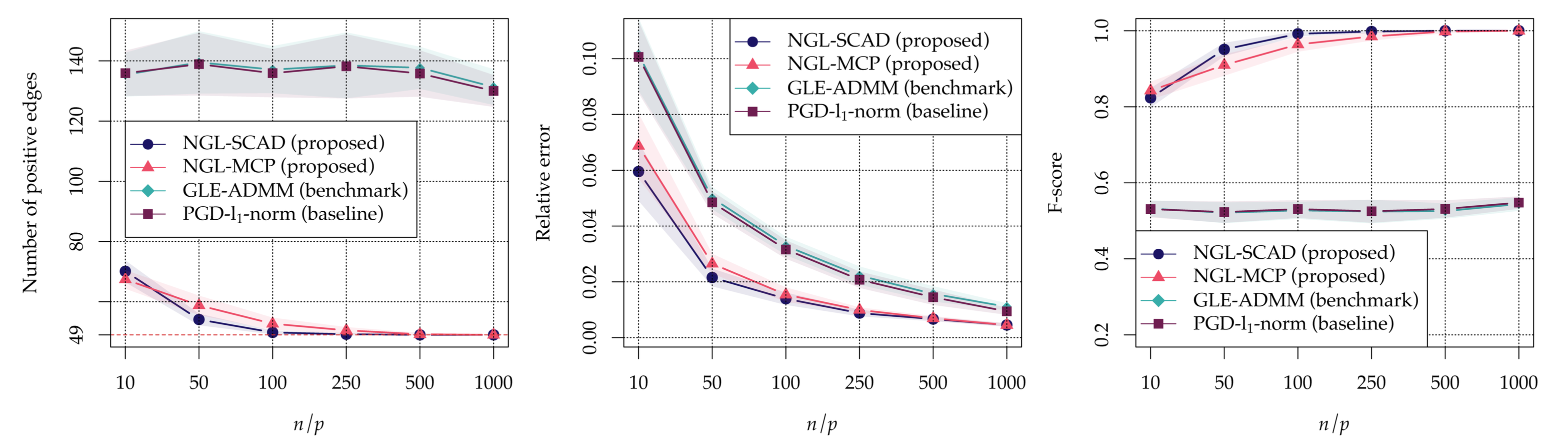
3.1 Synthetic Data

The data matrix $\boldsymbol{X} \in \mathbb{R}^{p \times n}$ with each column independently sampled from L-GMRF. The ground-truth graph is a random Barabasi-Albert graph with 50 nodes, and the weights are randomly sampled from $U(2, 5)$. The compared methods include the state-of-the-art GLE-ADMM algorithm [2] and the baseline projected gradient descent with ℓ_1-norm.



(a) (b) (c)

Figure 2: Performance measures (a) Number of positive edges, (b) Relative error and (c) F-score as a function of regularization parameter λ.

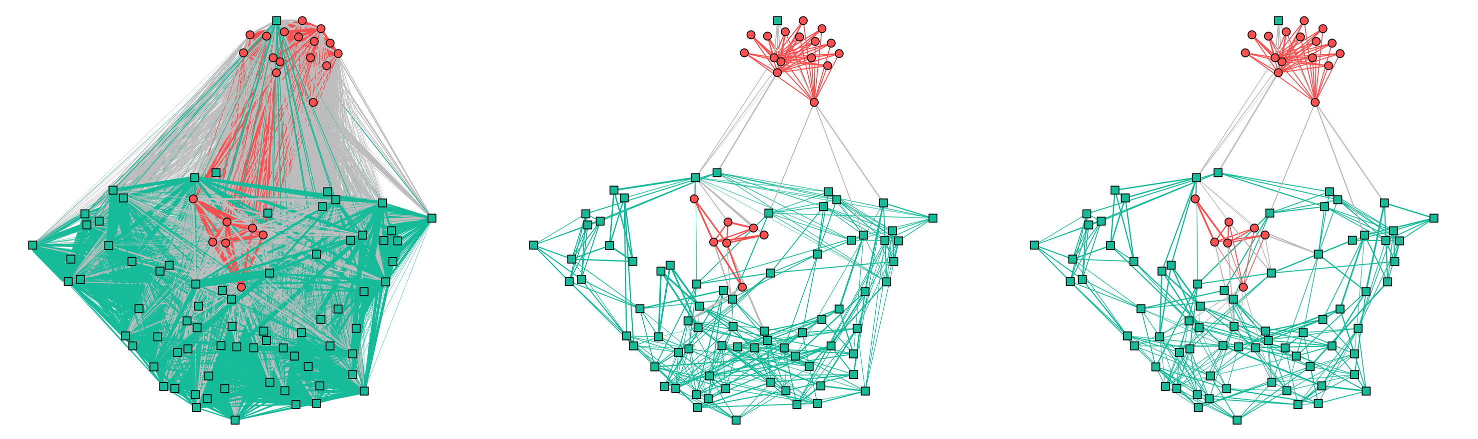


(a) (b) (c)

Figure 3: Performance measures (a) Number of positive edges, (b) Relative error and (c) F-score as a function of the sample size ratio n/p.

3.2 Real-world Data

The data set is 2019-nCoV [4] from 98 Chinese patients affected by the outbreak of 2019-nCoV on early February, 2020. The features include age, gender, and location. The labels represent the life status of patients, alive (green) or no longer alive (red).



(a) (b) (c)

Figure 4: The learned graphs using the 2019-nCoV data set by (a) GLE-ADMM, (b) NGL-SCAD (proposed method), and (c) NGL-MCP (proposed method).

References

[1] H. E. Egilmez, E. Pavez, and A. Ortega, "Graph learning from data under laplacian and structural constraints," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 6, pp. 825–841, 2017.

[2] L. Zhao, Y. Wang, S. Kumar, and D. P. Palomar, "Optimization algorithms for graph Laplacian estimation via ADMM and MM," *IEEE Transactions on Signal Processing*, vol. 67, no. 16, pp. 4231–4244, 2019.

[3] S. Kumar, J. Ying, J. V. d. M. Cardoso, and D. P. Palomar, "A unified framework for structured graph learning via spectral constraints," *Journal of Machine Learning Research*, vol. 21, no. 22, pp. 1–60, 2020.

[4] T. Wu, X. Ge, G. Yu, and E. Hu, "Open-source analytics tools for studying the covid-19 coronavirus outbreak," *medRxiv*, 2020.

[5] J. Ying, J. V. d. M. Cardoso, and D. P. Palomar, "Does the ℓ_1-norm learn a sparse graph under laplacian constrained graphical models?" *arXiv preprint arXiv:2006.14925*, 2020.