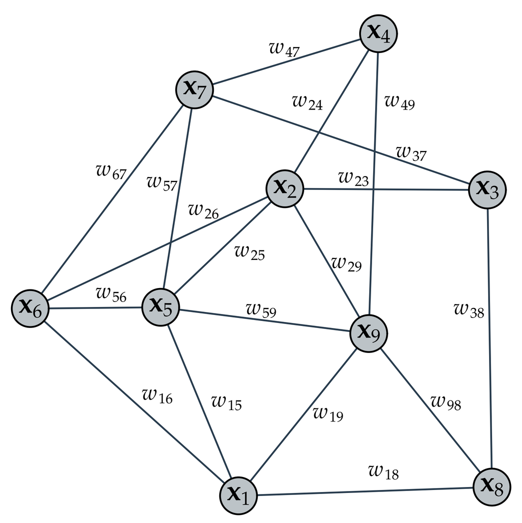
Structured Graph Learning via Laplacian Spectral Constraints

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

An R package for this paper is available at: https://cran.r-project.org/package=spectralGraphTopology.
For more information visit: https://www.danielppalomar.com

THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

Graphical modeling $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$

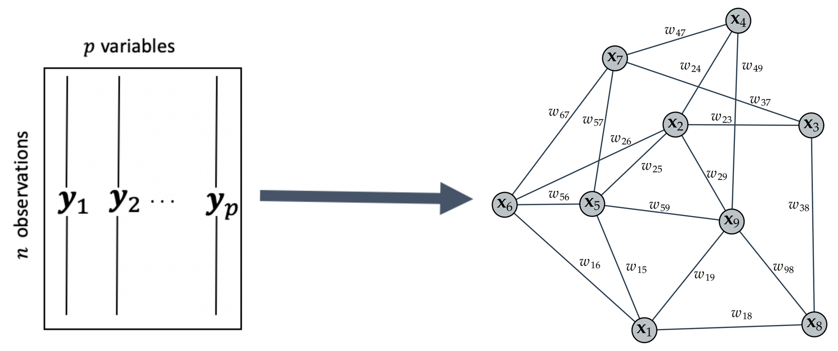


Representing knowledge through graphical model

- A graph is an intuitive way of **representing** and **visualizing** the relationships between entities.
- Nodes: $\mathcal{V} = \{1, 2, \ldots, p\}$ correspond to the **entities**.
- Edges: $\mathcal{E} = \{(1,2), (1,3), \ldots, (3,p), \ldots\}$ encode the **relationships** between entities.
- Elements of the weight matrix $\mathbf{W}$ encode the **strength** of the relationships.

Graph learning from data

Let $Y \in \mathbb{R}^{n \times p} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_p]$ be data matrix, each column $\mathbf{y}_i \in \mathbb{R}^n$ resides on one of p nodes of a graph $\mathcal{G}$, while each of the n rows of Y is an observation corresponding to one of the nodes of the graph.



Gaussian graphical modeling (GGM)

Sample covariance matrix $S = \frac{1}{n} Y^\top Y$.

$$\underset{\Theta \in \mathcal{S}^p_{++}}{\text{maximize}} \quad \log\det(\Theta) - \text{tr}(S\Theta) - \alpha h(\Theta),$$

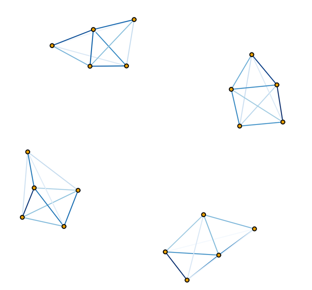
- When the data follows a Gaussian distribution $\mathbf{x} \sim \mathcal{N}(0, \Theta^{-1})$, it is the penalized **MLE** of the **precision matrix**.
- The entries of Θ determines a **conditional** graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

$$x_i \perp x_j | \mathbf{x}/(x_i, x_j) \iff \Theta = 0$$
$$\Theta_{ij} \neq 0 \iff \{i, j\} \in \mathcal{E} \; \forall \, i \neq j.$$

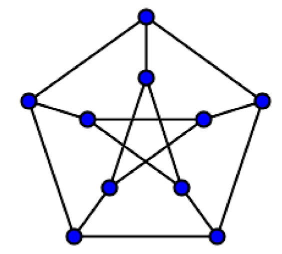
- $\mathbf{x} = [x_1, x_2, \ldots, x_p]$ is a Gaussian Markov random field (GMRF).
- Also a $\log\det(\cdot)$ regularized learning.
- If $\Theta_{ij} \leq 0 \; \forall \, i \neq j$ then X is an **attractive** GMRF.

Limitations: Existing GGM based methods cannot learn a graph Θ with specific structure [1, 2]. Enforcing structure is in general **NP-hard**.

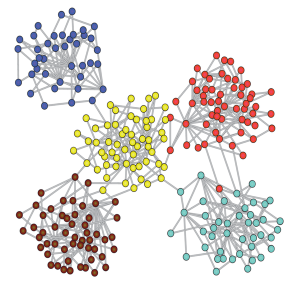
Structured graphs



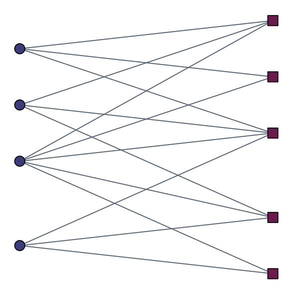
Multi-component graph



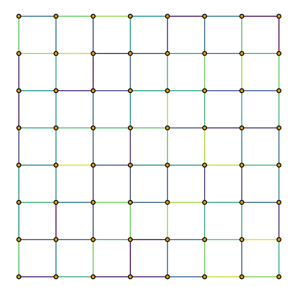
Regular graph



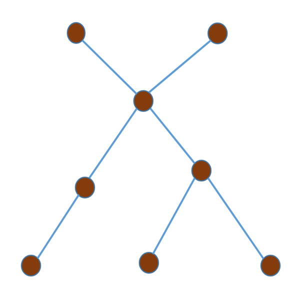
Modular graph



Bipartite graph



Grid graph



Tree graph

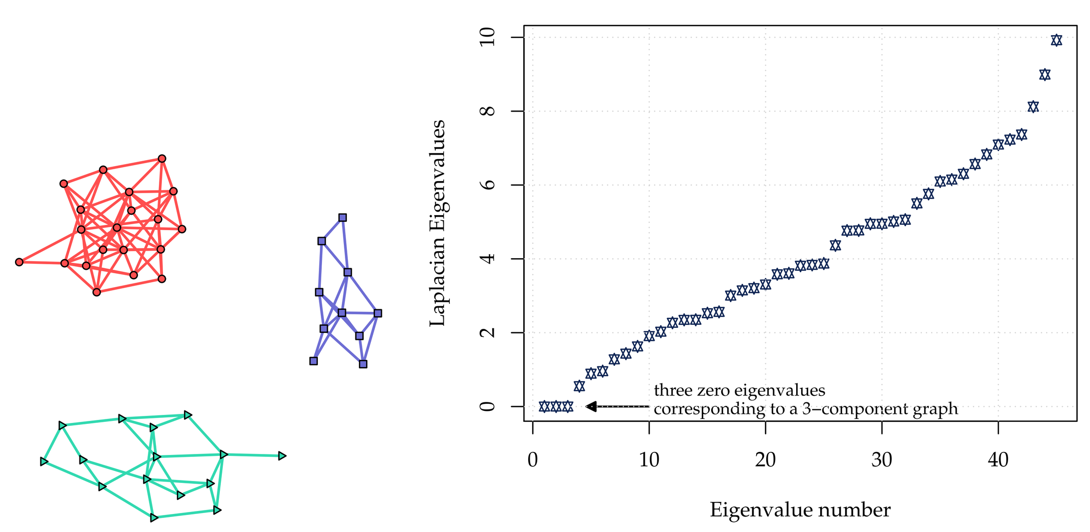
Graph Laplacian matrix

- **Laplacian** matrix Θ belongs to the following set:

$$\mathcal{S}_\Theta = \Big\{ \Theta | \Theta_{ij} = \Theta_{ji} \leq 0 \text{ for } i \neq j, \Theta_{ii} = -\sum_{i \neq j} \Theta_{ij} \Big\}.$$

- **Smoothness**: $\text{tr}(S\Theta)$ is used to quantify smoothness of signals: a smaller $\text{tr}(S\Theta)$ indicating a smoother signal.
- The **structural** properties of many important graph structures can be inferred from **eiegnvalues** of the **Laplacian** matrix $\boldsymbol{\lambda}(\Theta)$.

$$\mathcal{S}_\lambda = \{ \boldsymbol{\lambda} \in \mathbb{R}^p | \{\lambda_i = 0\}_{i=1}^k, c_1 \leq \lambda_{k+1} \leq \cdots \leq \lambda_p \leq c_2 \}.$$



The number of zero eigenvalues denote the number of components in the graph.

Proposed formulation for structured graph learning

$$\underset{\Theta}{\text{maximize}} \quad \log \text{gdet}(\Theta) - \text{tr}(S\Theta) - \alpha \|\Theta\|_{1,\text{off}}$$
$$\text{subject to } \Theta \in \mathcal{S}_\Theta, \quad \boldsymbol{\lambda}(\Theta) \in \mathcal{S}_\lambda$$

- ***Connected graph***: $\mathcal{S}_\lambda = \{\lambda_1 = 0, c_1 \leq \lambda_2 \leq \cdots \leq \lambda_p \leq c_2\}$ with a proper choice of $c_1 > 0, c_2 > 0$.
- **Grid**, **Modular**, **Erdos-Renyi** and other connected structures can also be learned under the connected spectral constraint.
- k-***component graph***: $\mathcal{S}_\lambda = \{\{\lambda_i\}_{i=1}^k, c_1 \leq \lambda_{k+1} \leq \cdots \leq \lambda_p \leq c_2\}$.
- $d-$**regular graph**: $\mathcal{S}_\lambda = \{\{\lambda_i = 0\}_{i=1}^k, c_1 \leq \lambda_{k+1} \leq \cdots \leq \lambda_p \leq c_2\}$ and $\text{Diag}(\Theta) = d\mathbf{I}$
- ***Cospectral graphs***

Laplacian operator: $\mathcal{L}$

$\mathcal{L}$ maps a weight vector $\mathbf{w} = [w_1, w_2, w_3, w_4, w_5, w_6]^\top$ to the Laplacian matrix.

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

Problem reformulation

$$\underset{\mathbf{w}, \boldsymbol{\lambda}, U}{\text{maximize}} \quad \log \text{gdet}(\text{Diag}(\boldsymbol{\lambda})) - \text{tr}(K\mathcal{L}\mathbf{w}) - \frac{\beta}{2}\|\mathcal{L}\mathbf{w} - U\text{Diag}(\boldsymbol{\lambda})U^\top\|_F^2,$$
$$\text{subject to } \mathbf{w} \geq 0, \; \boldsymbol{\lambda} \in \mathcal{S}_\lambda, \; U^\top U = I.$$

where $K = S + \alpha(I - \mathbf{1}\mathbf{1}^\top)$.

Proposed algorithm

Variables $\mathcal{X} = (\mathbf{w}, \boldsymbol{\lambda}, U)$: we develop a block majorization-minimization (**block-MM**) based algorithm.

Sub-problem for $\mathbf{w}$:

$$\underset{\mathbf{w} \geq 0}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{L}\mathbf{w}\|_F^2 - \mathbf{c}^\top \mathbf{w}.$$

where $\mathbf{c} = \mathcal{L}^*\left(U\text{Diag}(\boldsymbol{\lambda})U^\top - \beta^{-1}K\right)$. This is a constrained convex quadratic program, we solve it using the MM approach.

Sub-problem for U :

$$\underset{U}{\text{maximize}} \quad \text{tr}\left(U^\top(\mathcal{L}\mathbf{w})U\text{Diag}(\boldsymbol{\lambda})\right) \quad \text{subject to} \quad U^\top U = I.$$

Sub-problem for $\boldsymbol{\lambda}$:

$$\underset{\boldsymbol{\lambda} \in \mathcal{S}_\lambda}{\text{minimize}} \quad -\log \text{gdet}(\text{Diag}(\boldsymbol{\lambda})) + \frac{\beta}{2}\|U^\top(\mathcal{L}\mathbf{w})\mathbf{U} - \text{Diag}(\boldsymbol{\lambda})\|_F^2.$$

A convex **isotonic** regression problem, we develop a **fast** iterative algorithm that converges within $(p-k)$ iterations.

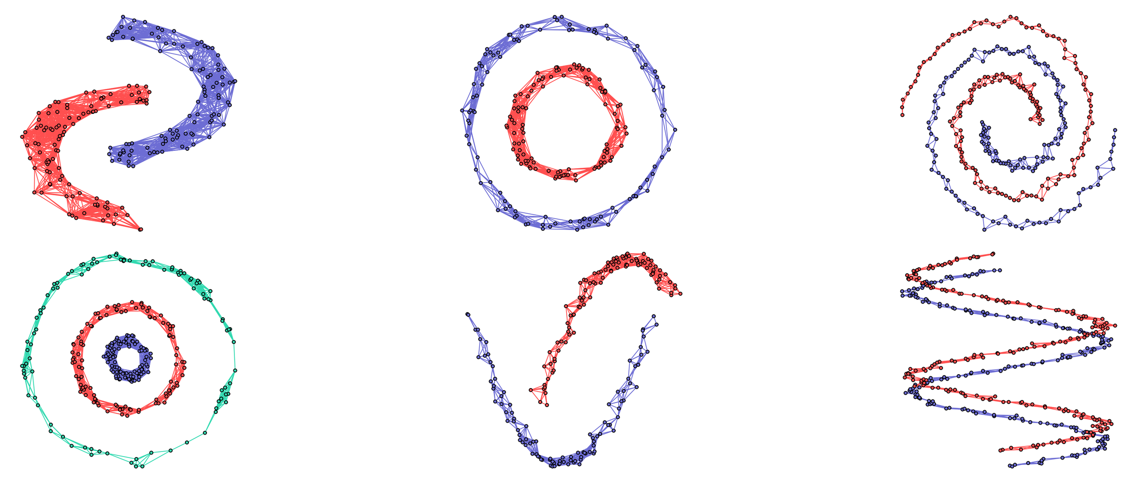
Algorithm summary

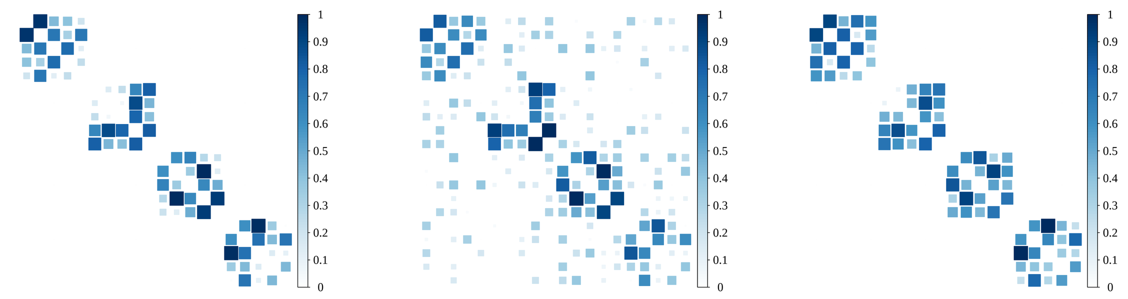
Input: SCM S and $\mathcal{S}_\lambda$.
while Stopping criteria not met **do**
    $\mathbf{w}^{t+1} = \left(\mathbf{w}^t - \frac{1}{2p}\left(\mathcal{L}^*(\mathcal{L}\mathbf{w}^t) - \mathbf{c}\right)\right)^+$
    $U^{t+1} = \text{eigenvectors}(\mathcal{L}\mathbf{w}^{t+1})$
    $\boldsymbol{\lambda}^{t+1}$: Update via isotonic regression maxm iter $(p-k)$.
end while
Return $\mathbf{w}^{t+1}$

The **worst-case** computational complexity $O(p^3)$.

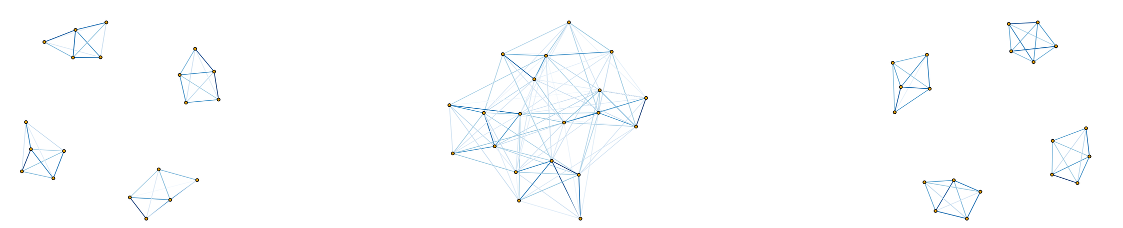
Theorem: *The sequence $(\mathbf{w}^t, U^t, \boldsymbol{\lambda}^t)$ generated by this algorithm converges to the set of KKT points of the optimization problem.*

Learning k-component structures

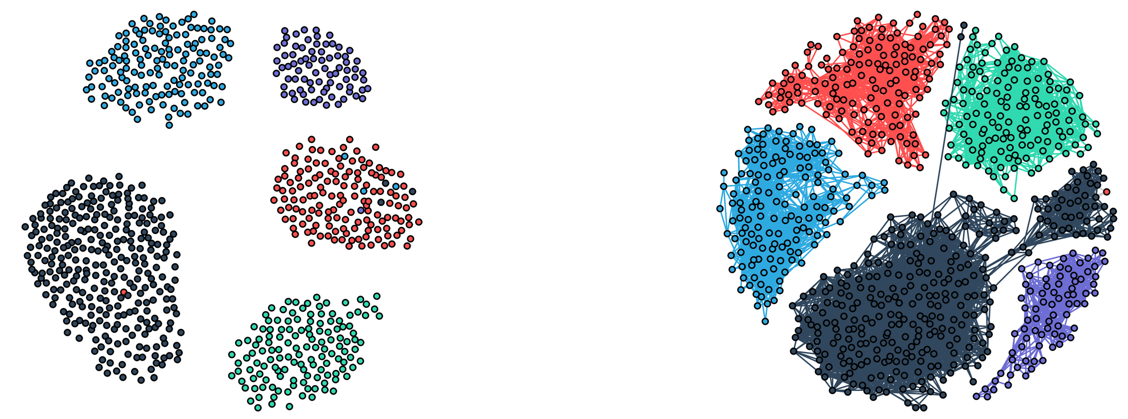




True matrix | Noisy matrix | Learned matrix



True graph | Noisy graph | Learned graph



CLR [3] | SGL with $k = 5$

Clustering with cancer data:
Clustering accuracy (ACC): CLR $= 0.98$ and SGL $= 0.99$.

References

1. J. Friedman, T. Hastie, and R. Tibshirani, “Sparse inverse covariance estimation with the graphical lasso,” *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.
2. 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.
3. F. Nie, X. Wang, M. I. Jordan, and H. Huang, “The constrained laplacian rank algorithm for graph-based clustering,” in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
4. S. Kumar, J. Ying, J. V. d. M. Cardoso, and D. Palomar, “Structured graph learning via Laplacian spectral constraints,” in *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
5. ——, “A unified framework for structured graph learning via spectral constraints,” *arXiv preprint arXiv:1904.09792*, 2019.