Structured Graph Learning Via Laplacian Spectral Constraints

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

The Hong Kong University of Science and Technology (HKUST)

NeurIPS 2019, Vancouver, Canada

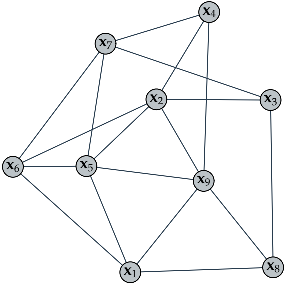
11 December 2019

Outline

Outline

Graphical models

Representing knowledge through graphical models



- Nodes correspond to the entities (variables).
- Edges encode the relationships between entities (dependencies between the variables)

Why do we need graphical models?

- Graphs are intuitive way of representing and visualising the **relationships** between entities.
- Graphs allow us to abstract out the **conditional independence** relationships between the variables from the details of their **parametric** forms. Thus we can answer questions like: "Is x_1 dependent on x_6 given that we know the value of x_8?" just by looking at the graph.
- Graphs are widely used in a variety of applications in machine learning, graph CNN, graph signal processing, etc.
- Graphs offer a **language** through which different **disciplines** can seamlessly **interact** with each other.
- Graph-based approaches with big data and machine learning are driving the current research frontiers.

Graphical Models = Statistics $\times$ Graph Theory $\times$ Optimization $\times$ Engineering

Why do we need graph learning?

Graphical models are about having a graph representation that can encode **relationships** between entities.

In many cases, the relationships between entities are **straightforward**:

- Are two people **friends** in a social network?
- Are two researchers **co-authors** in a published paper?

In many other cases, relationships are **not known** and must be learned:

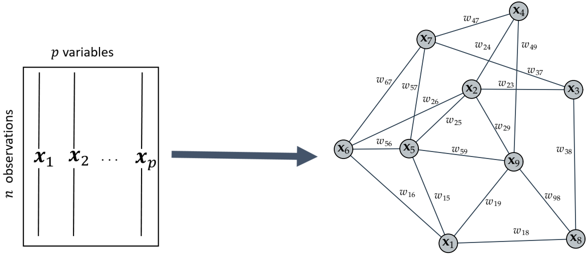
- Does one gene **regulate** the expression of others?
- Which drug **alters** the pharmacologic effect of another drug?

The **choice** of graph representation **affects** the subsequent analysis and eventually the **performance** of any graph-based algorithm.

The goal is to learn a graph representation of data with specific properties (e.g., structures).

Schematic of graph learning

- Given a data matrix $\mathbf{X} \in \mathbb{R}^{n \times p} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p]$, each column $\mathbf{x}_i \in \mathbb{R}^n$ is assumed to reside on one of the p nodes and each of the n rows of $\mathbf{X}$ is a signal (or feature) on the same graph.
- The goal is to obtain a graph representation of the data.



Graph is a simple **mathematical structure** of form $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where

- $\mathcal{V}$ contains the set of nodes $\mathcal{V} = \{1, 2, 3, \ldots, p\}$, and
- $\mathcal{E} = \{(1,2), (1,3), \ldots, (i,j), \ldots, (p, p-1)\}$ contains the set of edges between any pair of nodes (i, j).
- Weights $\{w_{12}, w_{13}, \ldots, w_{ij}, \ldots\}$ encode the relationships strength.

Examples

Learning relational dependencies among entities benefits numerous application domains.

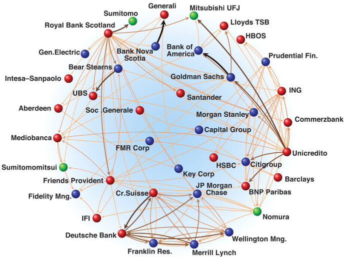


Figure 1: Financial Graph

Figure 2: Social Graph

Objective: To infer inter-dependencies of financial companies.

Input $\mathbf{x}_i$ is the economic indices (stock price, volume, etc.) of each entity.

Objective: To model behavioral similarity/ influence between people.

Input: Input $\mathbf{x}_i$ is the individual online activities (tagging, liking, purchase).

Types of graphical models

- Models encoding direct dependencies: simple and intuitive.
  - **Sample correlation** based graph.
  - **Similarity function** (e.g., Gaussian RBF) based graph.

- Models based on some assumption on the data: $\mathbf{X} \sim \mathcal{F}(\mathcal{G})$
  - **Statistical models**: $\mathcal{F}$ represents a distribution by $\mathcal{G}$ (e.g., Markov model and Bayesian model).
  - **Physically-inspired models**: $\mathcal{F}$ represents generative model on $\mathcal{G}$ (e.g., diffusion process on graphs).

Outline

Gaussian Markov random field (GMRF)

A random vector $\mathbf{x} = (x_1, x_2, \ldots, x_p)^\top$ is called a GMRF with parameters $(\mathbf{0},\ \mathbf{\Theta})$, if its density follows:

$$p(\mathbf{x}) = (2\pi)^{(-p/2)}(\det(\mathbf{\Theta}))^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{x}^\top \mathbf{\Theta}\mathbf{x})\right).$$

The nonzero pattern of $\mathbf{\Theta}$ determines a conditional graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$:

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

- For a Gaussian distributed data $\mathbf{x} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Sigma} = \mathbf{\Theta}^\dagger)$ the graph learning is simply an inverse covariance (precision) matrix estimation problem [Lauritzen, 1996].
- If the rank$(\mathbf{\Theta}) < p$ then $\mathbf{x}$ is called an **improper** GMRF (IGMRF) [Rue and Held, 2005].
- If $\mathbf{\Theta}_{ij} \leq 0\ \forall\ i \neq j$ then $\mathbf{x}$ is called an **attractive** improper GMRF [Slawski and Hein, 2015].

Historical timeline of Markov graphical models

Data $\mathbf{X} = \{\mathbf{x}^{(i)} \sim \mathcal{N}(0, \mathbf{\Sigma} = \mathbf{\Theta}^{\dagger})\}_{i=1}^{n}$, $\quad \mathbf{S} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^{\top}$

- Covariance selection [Dempster, 1972]: graph from the elements of $\mathbf{S}^{-1}$ inverse sample covariance matrix. Not applicable when sample covariance is not invertible!
- Neighborhood regression [Meinshausen and Bühlmann, 2006]:

$$\arg\min_{\beta_1} |\mathbf{x}^{(1)} - \beta_1 \mathbf{X}_{/\mathbf{x}^{(1)}}|^2 + \alpha\|\beta_1\|_1$$

- ℓ_1-regularized MLE [Friedman et al., 2008, Banerjee et al., 2008]:

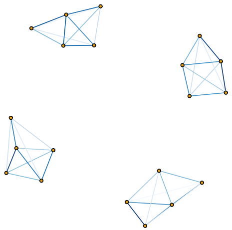
$$\underset{\mathbf{\Theta}\succ\mathbf{0}}{\text{maximize}} \quad \log\det(\mathbf{\Theta}) - \text{tr}\big(\mathbf{\Theta}\mathbf{S}\big) - \alpha\|\mathbf{\Theta}\|_1.$$

- Ising model: ℓ_1-regularized logistic regression [Ravikumar et al., 2010].
- Attractive IGMRF [Slawski and Hein, 2015].
- Laplacian structure in $\mathbf{\Theta}$ [Lake and Tenenbaum, 2010].
- ℓ_1-regularized MLE with Laplacian structure [Egilmez et al., 2017, Zhao et al., 2019]

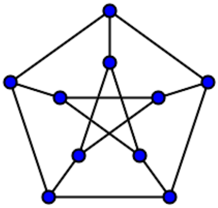
Limitation: Existing methods are not suitable for learning graphs with specific structures.

Outline

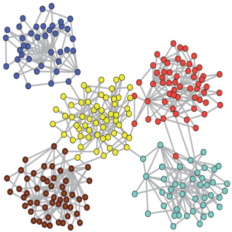
Structured graphs



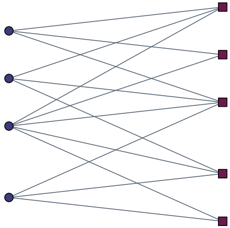
(i) Multi-component graph



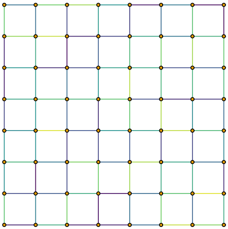
(ii) Regular graph



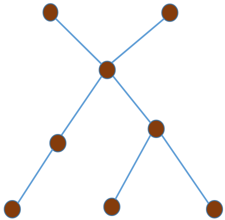
(iii) Modular graph



(iv) Bipartite graph



(v) Grid graph



(vi) Tree graph

Structured graphs: importance

Useful structures:

- ***Multi-component***: graph for clustering, classification.
- ***Bipartite***: graph for matching and constructing two-channel filter banks.
- ***Multi-component bipartite***: graph for co-clustering.
- ***Tree***: graphs for sampling algorithms.
- ***Modular***: graph for social network analysis.
- ***Connected sparse***: graph for graph signal processing applications.

Structured graph learning: challenges

Structured graph learning from data

- involves both the estimation of structure (**graph connectivity**) and parameters (**graph weights**),
- parameter estimation is well explored (e.g., maximum likelihood),
- but structure is a combinatorial property which makes structure estimation very challenging.

Structure learning is NP-hard for a general class of graphical models [Bogdanov et al., 2008].

Structured graph learning: direction

State-of-the-art direction:

- The effort has been on **characterizing** the families of structures for which learning can be made feasible e.g., **maximum weight spanning tree** for tree structure [Chow and Liu, 1968] and **local-separation** and **walk summability** for Erdos-Renyi graphs, power-law graphs, and small-world graphs [Anandkumar et al., 2012].
- Existing methods are restricted to some particular structures and it is difficult to extend them to learn other useful structures, e.g., multi-component, bipartite, etc.
- A recent method in [Hao et al., 2018], for learning multi-component structure follows a two-stage approach: non-optimal and not scalable to large-scale problems.

Proposed direction: Graph (**structure**) $\Longleftrightarrow$ Graph matrix (**spectrum**)

- **Spectral properties** of a graph matrix is one such characterization [Chung, 1997] which is considered in the present work.
- Under this framework, **structure learning** of a large class of graph structures can be expressed as an **eigenvalue problem** of the graph Laplacian matrix.

Outline

Problem statement

To learn structured graphs via Laplacian spectral constraints

Laplacian matrix

A set of $p \times p$ symmetric graph Laplacian matrices $\boldsymbol{\Theta}$:

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

Properties of $\boldsymbol{\Theta}$: Symmetric, diagonally dominant, positive semi-definite, and eigenvalues of $\boldsymbol{\Theta}$ encodes the structural properties of many important structures.

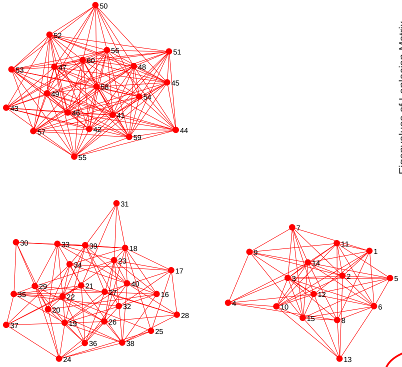
Laplacian quadratic energy function:

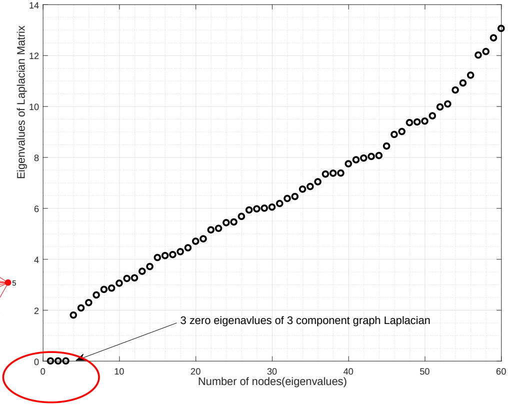
$$\text{tr}(\mathbf{S}\boldsymbol{\Theta}) = \sum_{i,j} -\boldsymbol{\Theta}_{ij}(x_i - x_j)^2$$

- The above trace term is used to quantify smoothness of graph signals: a smaller $\text{tr}(\mathbf{S}\boldsymbol{\Theta})$ indicating a smoother signal $\mathbf{x}$.
- A graph learned by minimizing the trace term tends to put more weight on the relationship of x_i, x_j if they are similar, and vice versa.
- If the signals x_i, x_j are similar then the learned Laplacian weights $|\boldsymbol{\Theta}_{ij}|$ will be large, and vice versa.

Motivating example: structure via Laplacian eigenvalues

Spectral graph theory: Graph (**structure**) $\Longleftrightarrow$ Graph Matrix (**spectrum**)





A graph and its Laplacian matrix eigenvalues: $k = 3$ zero eigenvalues corresponding to $k = 3$ connected components.

Proposed framework for structured graph learning

$$\begin{array}{ll} \underset{\boldsymbol{\Theta}}{\text{maximize}} & \log \, \text{gdet}(\boldsymbol{\Theta}) - \text{tr}\big(\boldsymbol{\Theta}\mathbf{S}\big) - \alpha h(\boldsymbol{\Theta}), \\ \text{subject to} & \boldsymbol{\Theta} \in \mathcal{S}_{\boldsymbol{\Theta}}, \; \boldsymbol{\lambda}(\mathcal{T}(\boldsymbol{\Theta})) \in \mathcal{S}_{\lambda}, \end{array}$$

- gdet is the **generalized determinant** defined as the non-zero eigenvalues product,
- $\mathcal{S}_{\boldsymbol{\Theta}}$ encodes the typical constraints of a **Laplacian matrix**,
- $\boldsymbol{\lambda}(\mathcal{T}(\boldsymbol{\Theta}))$ is the vector containing the eigenvalues of matrix $\mathcal{T}(\boldsymbol{\Theta})$,
- $\mathcal{T}(\cdot)$ is the **transformation matrix** to consider the eigenvalues of different graph matrices, and
- $\mathcal{S}_{\lambda}$ allows to include **spectral constraints** in the eigenvalues.
- Precisely $\mathcal{S}_{\lambda}$ will facilitate the process of incorporating the spectral properties required for enforcing structure.

The proposed formulation has converted the **combinatorial** structural constraints into **analytical** spectral constraints.

Structures via Laplacian spectral constraints

$$\mathcal{T}(\boldsymbol{\Theta}) = \boldsymbol{\Theta}$$

- **Connected**: $\mathcal{S}_\lambda = \{\lambda_1 = 0, c_1 \leq \lambda_2 \leq \cdots \leq \lambda_p \leq c_2\}$
- $k-$**component**: $\mathcal{S}_\lambda = \{\{\lambda_i = 0\}_{i=1}^{k}, c_1 \leq \lambda_{k+1} \leq \cdots \leq \lambda_p \leq c_2\}$
- $d-$**regular**: $\mathcal{S}_\lambda = \{\{\lambda_i = 0\}_{i=1}^{k}, c_1 \leq \lambda_{k+1} \leq \cdots \leq \lambda_p \leq c_2\}$ and Diag$(\boldsymbol{\Theta}) = d\mathbf{I}$
- Popular connected structures, e.g., **Grid**, **Modular**, and **Erdos-Renyi** can also be learned under the connected spectral constraint.

Note: By properly specifying the transformation matrix $\mathcal{T}(\cdot)$ in the proposed formulation, the spectral properties of other than graph Laplacian, e.g., **adjacency**, **normalized Laplacian**, and **signless Laplacian** can also be utilized to learn more non-trivial structures (e.g., **bipartite** and **multi-component bipartite** graph structures)
[Van Mieghem, 2010, Kumar et al., 2019, Chung, 1997].

Outline

Problem formulation for Laplacian spectral constraints

$$\begin{array}{ll} \underset{\boldsymbol{\Theta},\boldsymbol{\lambda},\mathbf{U}}{\text{maximize}} & \log\ \text{gdet}(\boldsymbol{\Theta}) - \text{tr}\big(\boldsymbol{\Theta}\mathbf{S}\big) - \alpha \left\|\boldsymbol{\Theta}\right\|_1, \\ \text{subject to} & \boldsymbol{\Theta} \in \mathcal{S}_{\boldsymbol{\Theta}},\ \boldsymbol{\Theta} = \mathbf{U}\text{Diag}(\boldsymbol{\lambda})\mathbf{U}^T,\ \boldsymbol{\lambda} \in \mathcal{S}_{\lambda},\ \mathbf{U}^T\mathbf{U} = \mathbf{I}, \end{array}$$

where $\boldsymbol{\lambda} = [\lambda_1, \lambda_2, \ldots, \lambda_p]$ is the vector of eigenvalues and $\mathbf{U}$ is the matrix of eigenvectors.

The resulting formulation is still **complicated** and **intractable**:

- Laplacian structural constraints,
- non-convex constraints coupling $\boldsymbol{\Theta}, \mathbf{U}, \boldsymbol{\lambda}$, and
- non-convex constraints on $\mathbf{U}$.

In order to derive a feasible formulation:

- we first introduce a **linear operator** $\mathcal{L}$ that transforms the Laplacian structural constraints to simple algebraic constraints and
- then **relax** the eigen-decomposition expression into the objective function.

Linear operator for $\mathbf{\Theta} \in \mathcal{S}_{\mathbf{\Theta}}$

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

$\Theta_{ij} = \Theta_{ji} \leq 0$ and $\mathbf{\Theta 1} = \mathbf{0}$ implying the target matrix is **symmetric** with **degrees of freedom** of $\mathbf{\Theta}$ equal to $p(p-1)/2$.
We define a linear operator $\mathcal{L} : \mathbf{w} \in \mathbb{R}_{+}^{p(p-1)/2} \rightarrow \mathcal{L}\mathbf{w} \in \mathbb{R}^{p \times p}$, which maps a weight vector $\mathbf{w}$ to the Laplacian matrix:

$$[\mathcal{L}\mathbf{w}]_{ij} = [\mathcal{L}\mathbf{w}]_{ji} \leq 0; \; i \neq j$$

$$[\mathcal{L}\mathbf{w}]_{ii} = -\sum_{j \neq i} [\mathcal{L}\mathbf{w}]_{ij}$$

Example of $\mathcal{L}\mathbf{w}$ on $\mathbf{w} = [w_1, w_2, w_3, w_4, w_5, w_6]^\top$:

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

Problem reformulation

$$\begin{array}{ll} \underset{\boldsymbol{\Theta},\boldsymbol{\lambda},\mathbf{U}}{\text{maximize}} & \log \text{gdet}(\boldsymbol{\Theta}) - \text{tr}\big(\boldsymbol{\Theta}\mathbf{S}\big) - \alpha \left\|\boldsymbol{\Theta}\right\|_1, \\ \text{subject to} & \boldsymbol{\Theta} \in \mathcal{S}_{\boldsymbol{\Theta}},\ \boldsymbol{\Theta} = \mathbf{U}\text{Diag}(\boldsymbol{\lambda})\mathbf{U}^T,\ \boldsymbol{\lambda} \in \mathcal{S}_\lambda,\ \mathbf{U}^T\mathbf{U} = \mathbf{I}, \end{array}$$

Using: i) $\boldsymbol{\Theta} = \mathcal{L}\mathbf{w}$ and ii) $\text{tr}\big(\boldsymbol{\Theta}\mathbf{S}\big) + \alpha h(\boldsymbol{\Theta}) = \text{tr}\big(\boldsymbol{\Theta}\mathbf{K}\big)$, $\mathbf{K} = \mathbf{S} + \mathbf{H}$ and $\mathbf{H} = \alpha(2\mathbf{I} - \mathbf{1}\mathbf{1}^T)$ the proposed problem formulation becomes:

$$\Downarrow$$

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

SGL algorithm for $k-$component graph learning

- Variables: $\mathcal{X} = (\mathbf{w},\ \boldsymbol{\lambda},\ \mathbf{U})$
- Spectral constraint: $\mathcal{S}_\lambda = \{\{\lambda_j = 0\}_{j=1}^k,\ c_1 \leq \lambda_{k+1} \leq \cdots \leq \lambda_p \leq c_2\}$.
- Positivity constraint: $\mathbf{w} \geq 0$
- Orthogonality constraint: $\mathbf{U}^T\mathbf{U} = \mathbf{I}_{p-k}$

We develop a block majorization-minimization (block-MM) type method which updates **each block sequentially** while keeping the other **blocks fixed** [Sun et al., 2016, Razaviyayn et al., 2013].

Update for $\mathbf{w}$

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

$$\underset{\mathbf{w}\geq 0}{\text{minimize}} \qquad \text{tr}\left(\mathbf{K}\mathcal{L}\mathbf{w}\right) + \frac{\beta}{2}\|\mathcal{L}\mathbf{w} - \mathbf{U}\text{Diag}(\boldsymbol{\lambda})\mathbf{U}^T\|_F^2.$$

$$\underset{\mathbf{w}\geq 0}{\text{minimize}} \qquad f(\mathbf{w}) = \frac{1}{2}\left\|\mathcal{L}\mathbf{w}\right\|_F^2 - \mathbf{c}^T\mathbf{w},$$

This problem is a convex quadratic program, but does not have a closed-form solution due to the non-negativity constraint $\mathbf{w} \geq 0$.

We obtain a closed-form update by using the MM technique [Sun et al., 2016]

$$\mathbf{w}^{t+1} = \left(\mathbf{w}^t - \frac{1}{2p}\nabla f(\mathbf{w}^t)\right)^+,$$

where $(a)^+ = \max(a, 0)$.

Update for $\mathbf{U}$

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

$$\begin{array}{ll} \underset{\mathbf{U}}{\text{maximize}} & \text{tr}\left(\mathbf{U}^T \mathcal{L}\mathbf{w}\mathbf{U}\text{Diag}(\boldsymbol{\lambda})\right) \\ \text{subject to} & \mathbf{U}^T\mathbf{U} = \mathbf{I}_{p-k}. \end{array}$$

This sub- problem is an optimization on the orthogonal Stiefel manifold [Absil et al., 2009, Benidis et al., 2016]. From the KKT optimality conditions the solution is given by

$$\mathbf{U}^{t+1} = \text{eigenvectors}(\mathcal{L}\mathbf{w}^{t+1})[k+1:p],$$

that is, the $p-k$ principal eigenvectors of the matrix $\mathcal{L}\mathbf{w}^{t+1}$ in increasing order of eigenvalue magnitude.

Update for $\boldsymbol{\lambda}$

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

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

$$\underset{c_1 \leq \lambda_{k+1} \leq \cdots \leq \lambda_p \leq c_2}{\text{minimize}} \quad -\sum_{i=1}^{p-k} \log \lambda_{k+i} + \frac{\beta}{2}\|\boldsymbol{\lambda} - \mathbf{d}\|^2,$$

The sub-problem is popularly known as a regularized isotonic regression problem. This is a convex optimization problem and the solution can be obtained from the KKT optimality conditions. We develop an efficient algorithm with a fast convergence to the global optimum in a maximum of $p - k$ iterations [Kumar et al., 2019].

Sandeep Kumar, Jiaxi Ying, José Vinícius de M. Cardoso, and Daniel P. Palomar," A Unified Framework for Structured Graph Learning via Spectral Constraints." arXiv preprint arXiv:1904.09792 (2019).

Proposed SGL algorithm summary

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

Proposed algorithm:

Input: SCM $\mathbf{S}$, k, c_1, c_2, β
Output: $\mathcal{L}\mathbf{w}$
$t \leftarrow 0$
while stopping criterion is not met **do**
    $\mathbf{w}^{t+1} = \left(\mathbf{w}^t - \frac{1}{2p}\nabla f(\mathbf{w}^t)\right)^+$
    $\mathbf{U}^{t+1} \leftarrow$ eigenvectors$(\mathcal{L}\mathbf{w}^{t+1})$, suitably ordered.
    Update $\boldsymbol{\lambda}^{t+1}$ (via isotonic regression method with maxm iter $p-k$).
    $t \leftarrow t+1$
end while
return $\mathbf{w}^{(t+1)}$

Convergence and the computational complexity

The **worst-case** computational complexity of the proposed algorithm is $O(p^3)$.

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

Sandeep Kumar, Jiaxi Ying, José Vinícius de M. Cardoso, and Daniel P. Palomar, "Structured graph learning via Laplacian spectral constraints," *in Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

Outline

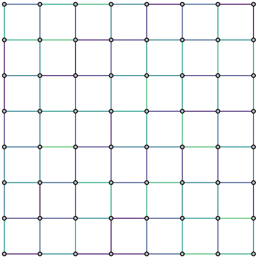
Synthetic experiment setup

- Generate a graph with desired structure.
- Sample weights for the graph edges.
- Obtain true Laplacian $\boldsymbol{\Theta}_{\text{true}}$.
- Sample data $\mathbf{X} = \{\mathbf{x}^{(i)} \in \mathbb{R}^p \sim \mathcal{N}(0, \Sigma = \boldsymbol{\Theta}_{\text{true}}^{\dagger})\}_{i=1}^{n}$.
- $\mathbf{S} = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}^{(i)})(\mathbf{x}^{(i)})^{\top}$.
- Use $\mathbf{S}$ and some prior spectral information, if available.
- **Performance metric**

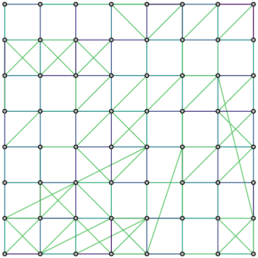
$$\text{Relative Error} = \frac{\left\| \hat{\boldsymbol{\Theta}}^{\star} - \boldsymbol{\Theta}_{\text{true}} \right\|_F}{\left\| \boldsymbol{\Theta}_{\text{true}} \right\|_F}, \quad \text{F-Score} = \frac{2\text{tp}}{2\text{tp} + \text{fp} + \text{fn}}$$

- Where $\hat{\boldsymbol{\Theta}}^{\star}$ is the final estimation result the algorithm and $\boldsymbol{\Theta}_{\text{true}}$ is the true reference graph Laplacian matrix, and tp, fp, fn correspond to true positives, false positives, and false negatives, respectively.

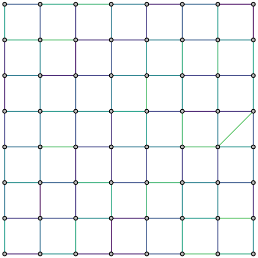
Grid graph



(i) True

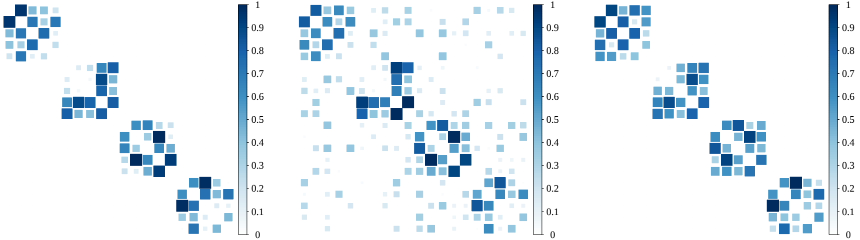


(ii) [Egilmez et al., 2017]

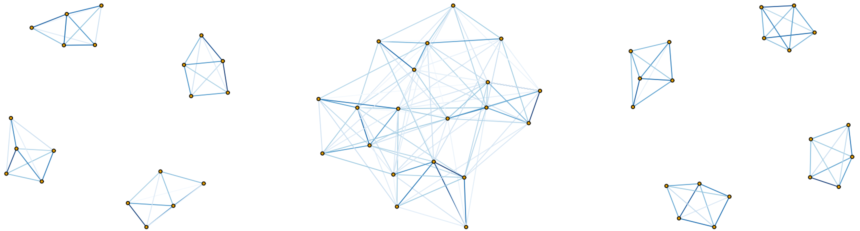


(iii) SGL with $k = 1$

Noisy multi-component graph

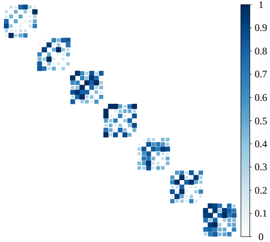


(iv) True (v) Noisy (vi) Learned

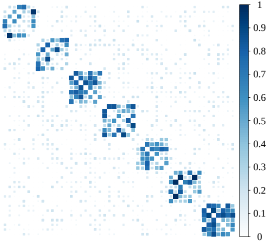


(vii) True graph (viii) Noisy graph (ix) Learned

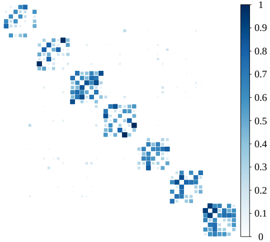
Model mismatch



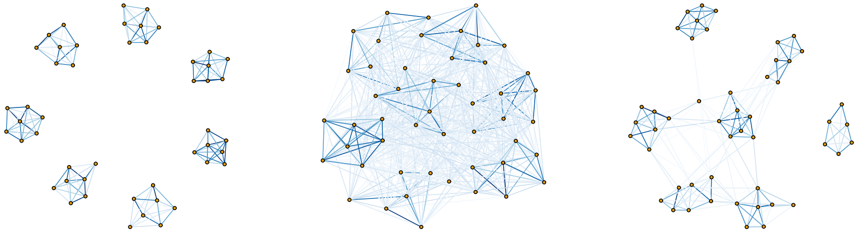
(x) True $k = 7$



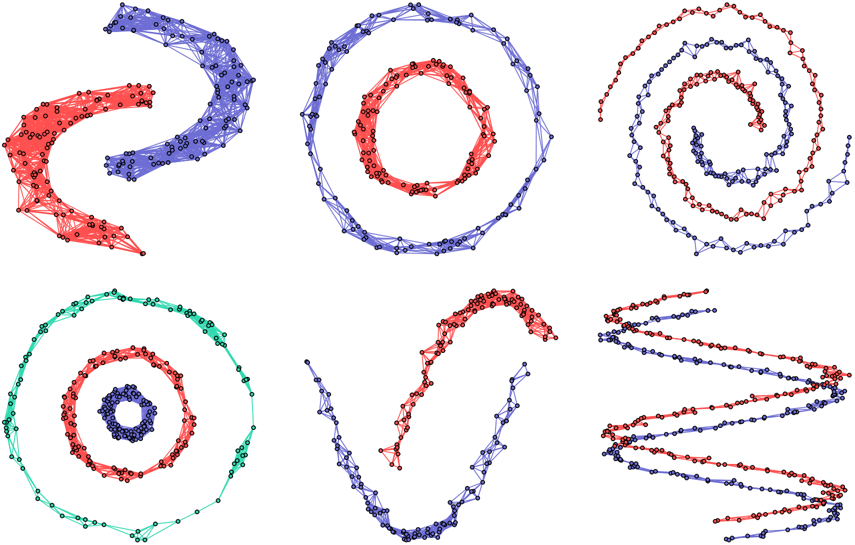
(xi) Noisy



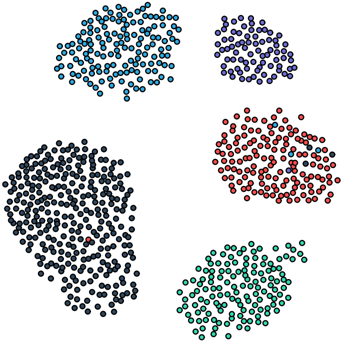
(xii) Learned with $k = 2$



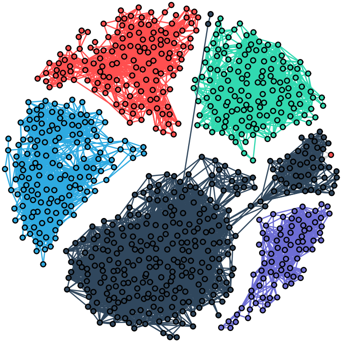
Popular multi-component structures



Real data: cancer dataset [Weinstein et al., 2013]



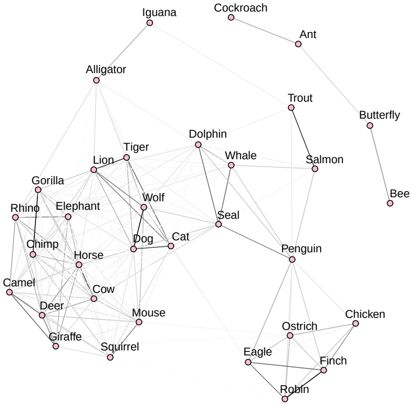
(xxii) CLR (Nie et al., 2016)



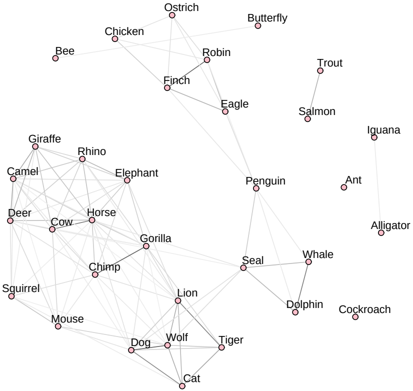
(xxiii) SGL with $k = 5$

Clustering accuracy (ACC): CLR $= 0.9862$ and SGL $= 0.99875$.

Animal dataset [Osherson et al., 1991]

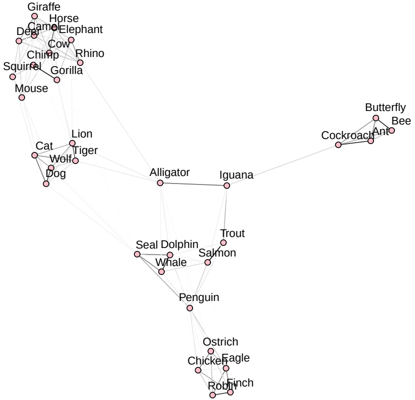


(xxiv) GGL [Egilmez et al., 2017]

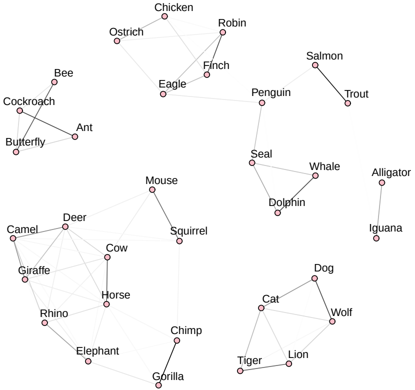


(xxv) GLasso [Friedman et al., 2008]

Animal dataset contd...



(xxvi) SGL: proposed ($k = 1$)



(xxvii) SGL: proposed ($k = 4$)

Resources

An R package "**spectralGraphTopology**" containing code for all the experimental results is available at
https://cran.r-project.org/package=spectralGraphTopology

NeurIPS paper: Sandeep Kumar, Jiaxi Ying, José Vinícius de M. Cardoso, and Daniel P. Palomar, "Structured graph learning via Laplacian spectral constraints," *in Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
https://arxiv.org/pdf/1909.11594.pdf

Extended version paper: Sandeep Kumar, Jiaxi Ying, José Vinícius de M. Cardoso, and Daniel P. Palomar, "A Unified Framework for Structured Graph Learning via Spectral Constraints, (2019)." https://arxiv.org/pdf/1904.09792.pdf

Authors:

Thanks

For more information visit:

https://www.danielppalomar.com

References

Absil, P.-A., Mahony, R., and Sepulchre, R. (2009).
Optimization algorithms on matrix manifolds.
Princeton University Press.

Anandkumar, A., Tan, V. Y., Huang, F., and Willsky, A. S. (2012).
High-dimensional gaussian graphical model selection: Walk summability and local separation criterion.
Journal of Machine Learning Research, 13(Aug):2293–2337.

Banerjee, O., Ghaoui, L. E., and d'Aspremont, A. (2008).
Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data.
Journal of Machine Learning Research, 9(Mar):485–516.

Benidis, K., Sun, Y., Babu, P., and Palomar, D. P. (2016).
Orthogonal sparse pca and covariance estimation via procrustes reformulation.
IEEE Transactions on Signal Processing, 64(23):6211–6226.

Bogdanov, A., Mossel, E., and Vadhan, S. (2008).
The complexity of distinguishing markov random fields.
In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 331–342. Springer.

References

Chow, C. and Liu, C. (1968).
Approximating discrete probability distributions with dependence trees.
IEEE transactions on Information Theory, 14(3):462–467.

Chung, F. R. (1997).
Spectral graph theory.
Number 92. American Mathematical Soc.

Dempster, A. P. (1972).
Covariance selection.
Biometrics, pages 157–175.

Egilmez, H. E., Pavez, E., and Ortega, A. (2017).
Graph learning from data under laplacian and structural constraints.
IEEE Journal of Selected Topics in Signal Processing, 11(6):825–841.

Friedman, J., Hastie, T., and Tibshirani, R. (2008).
Sparse inverse covariance estimation with the graphical lasso.
Biostatistics, 9(3):432–441.

References

Hao, B., Sun, W. W., Liu, Y., and Cheng, G. (2018).
Simultaneous clustering and estimation of heterogeneous graphical models.
Journal of Machine Learning Research, 18(217):1–58.

Kumar, S., Ying, J., Cardoso, J. V. d. M., and Palomar, D. (2019).
A unified framework for structured graph learning via spectral constraints.
arXiv preprint arXiv:1904.09792.

Lake, B. and Tenenbaum, J. (2010).
Discovering structure by learning sparse graphs.
In Proceedings of the 33rd Annual Cognitive Science Conference.

Lauritzen, S. L. (1996).
Graphical models, volume 17.
Clarendon Press.

Meinshausen, N. and Bühlmann, P. (2006).
High-dimensional graphs and variable selection with the lasso.
The annals of statistics, 34(3):1436–1462.

References

Osherson, D. N., Stern, J., Wilkie, O., Stob, M., and Smith, E. E. (1991).
Default probability.
Cognitive Science, 15(2):251–269.

Ravikumar, P., Wainwright, M. J., Lafferty, J. D., et al. (2010).
High-dimensional ising model selection using ℓ_1-regularized logistic regression.
The Annals of Statistics, 38(3):1287–1319.

Razaviyayn, M., Hong, M., and Luo, Z.-Q. (2013).
A unified convergence analysis of block successive minimization methods for nonsmooth optimization.
SIAM Journal on Optimization, 23(2):1126–1153.

Rue, H. and Held, L. (2005).
Gaussian Markov random fields: theory and applications.
CRC press.

Slawski, M. and Hein, M. (2015).
Estimation of positive definite m-matrices and structure learning for attractive gaussian markov random fields.
Linear Algebra and its Applications, 473:145–179.

References

Sun, Y., Babu, P., and Palomar, D. P. (2016).
Majorization-minimization algorithms in signal processing, communications, and machine learning.
IEEE Transactions on Signal Processing, 65(3):794–816.

Van Mieghem, P. (2010).
Graph spectra for complex networks.
Cambridge University Press.

Weinstein, J. N., Collisson, E. A., Mills, G. B., Shaw, K. R. M., Ozenberger, B. A., Ellrott, K., Shmulevich, I., Sander, C., Stuart, J. M., Network, C. G. A. R., et al. (2013).
The cancer genome atlas pan-cancer analysis project.
Nature Genetics, 45(10):1113.

Zhao, L., Wang, Y., Kumar, S., and Palomar, D. P. (2019).
Optimization algorithms for graph laplacian estimation via admm and mm.
IEEE Transactions on Signal Processing, 67(16):4231–4244.