Inferring Networks and Network Properties from Graph Dynamic Processes

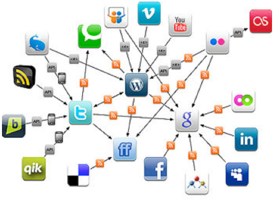
Santiago Segarra

Electrical and Computer Engineering
Rice University
segarra@rice.edu
http://segarra.rice.edu

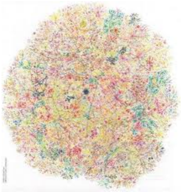
One World SP Seminar Series, July 16-17, 2020

Network Science

Online social media



Internet

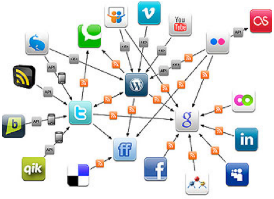


Clean energy and grid analytics

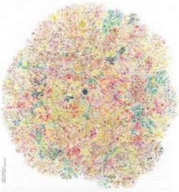
- Desiderata: Process, analyze and learn from network data [Kolaczyk09]

Network Science

Online social media



Internet



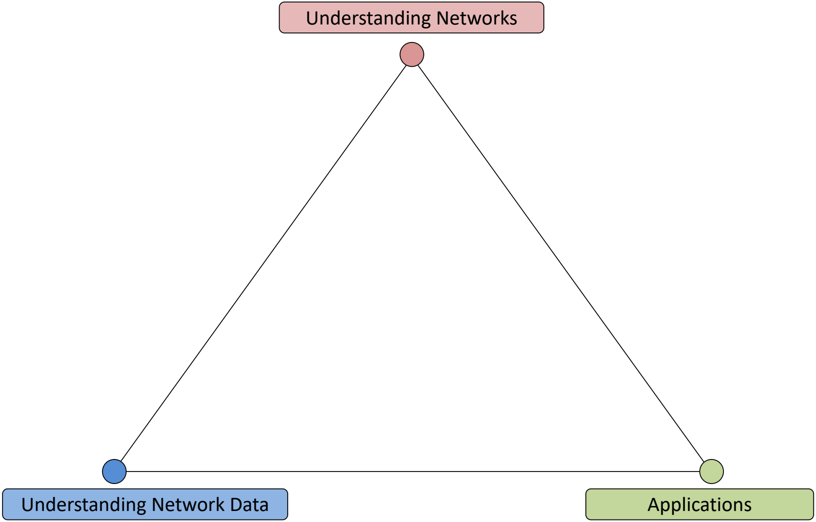
Clean energy and grid analytics

- Desiderata: Process, analyze and learn from network data [Kolaczyk09]

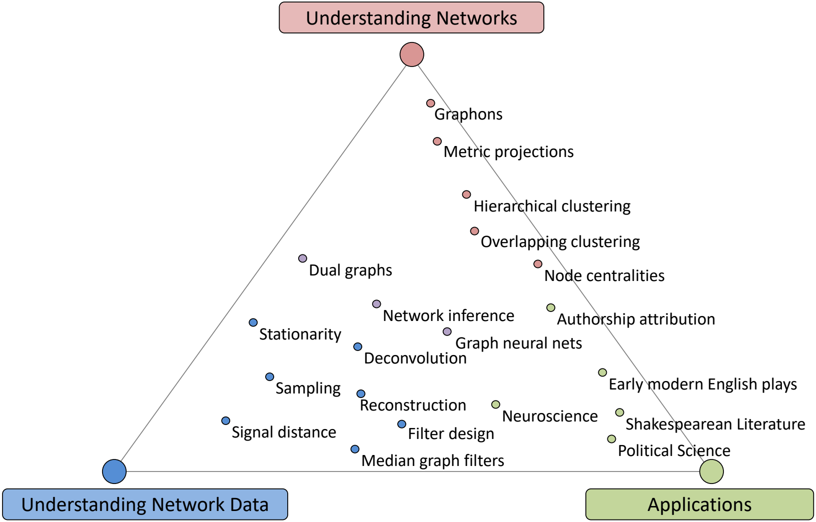
- Network as graph G: encodes pairwise relationships between agents
- Interest not only in G, also in network data associated with the nodes

Combine Network Science and Signal Processing and Machine Learning to leverage the structure of networks for the better understanding of data defined on them

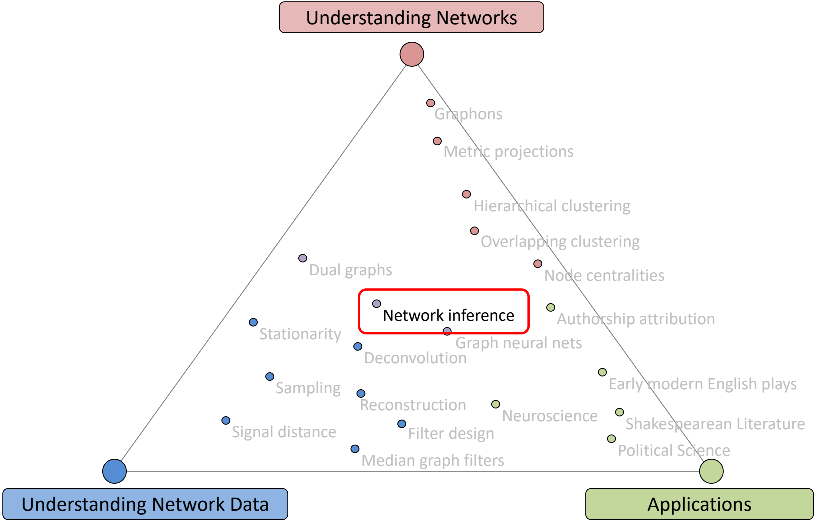
Data Science for Networks



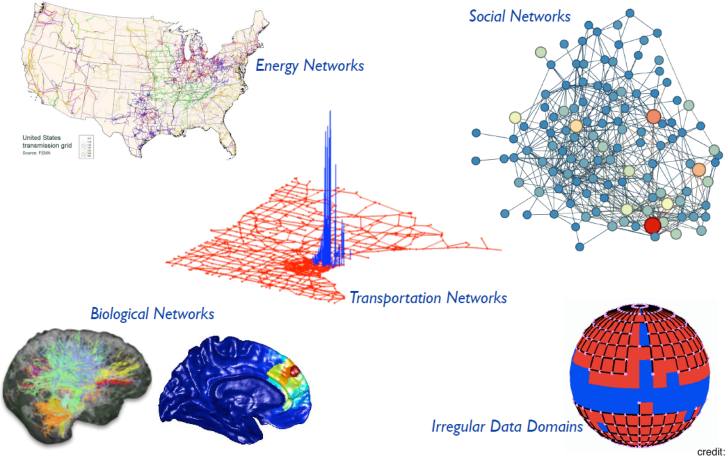
Data Science for Networks



Data Science for Networks



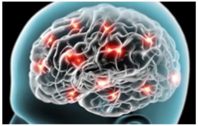
Motivating examples – Graph signals



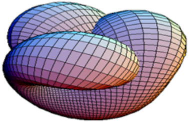
credit: LTS2

Motivating examples – Processing signals

Interpolate a brain signal from local observations



Compress a signal in an irregular domain



Localize the source of a rumor

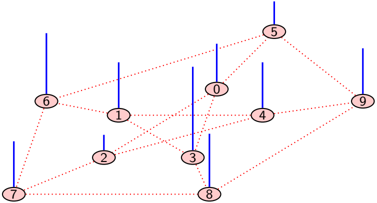
Smooth an observed network profile

Predict the evolution of a network process

Infer the topology where the signals reside

- Consider graph $G = (\mathcal{V}, \mathcal{E}, W)$. Graph signals are mappings $x : \mathcal{V} \to \mathbb{R}$
  - $\Rightarrow$ Defined on the nodes of the graph

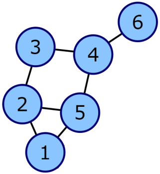
- May be represented as a vector $\mathbf{x} \in \mathbb{R}^N$
  - $\Rightarrow$ x_i denotes the signal value at the i-th vertex in $\mathcal{V}$



$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{10} \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

Graph-shift operator

- Associated with G is the graph-shift operator $\mathbf{S} \in \mathbb{R}^{N \times N}$
- $\mathbf{S}$ can take nonzero values in the edges of G or in its diagonal
  - $\Rightarrow$ $\mathbf{S}$ transformation that can be computed locally at the nodes



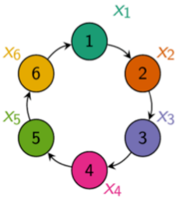
$$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{23} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix}$$

- Ex: Adjacency $\mathbf{A}$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices

Graph-shift operator

- Associated with G is the graph-shift operator $\mathbf{S} \in \mathbb{R}^{N \times N}$
- $\mathbf{S}$ can take nonzero values in the edges of G or in its diagonal

  $\Rightarrow$ $\mathbf{S}$ transformation that can be computed locally at the nodes

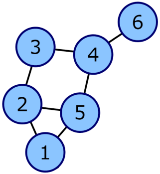


$$\mathbf{S} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$$

- Ex: Adjacency $\mathbf{A}$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices

Graph-shift operator

- Associated with G is the graph-shift operator $\mathbf{S} \in \mathbb{R}^{N \times N}$
- $\mathbf{S}$ can take nonzero values in the edges of G or in its diagonal
  - $\Rightarrow$ $\mathbf{S}$ transformation that can be computed locally at the nodes



$$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{23} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix}$$

- Ex: Adjacency $\mathbf{A}$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices

Graph Fourier Transform (GFT)

- Let $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ be the shift associated with G
- The Graph Fourier Transform (GFT) of $\mathbf{x}$ is defined as

$$\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$$

- While the inverse GFT (iGFT) of $\tilde{\mathbf{x}}$ is defined as $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$

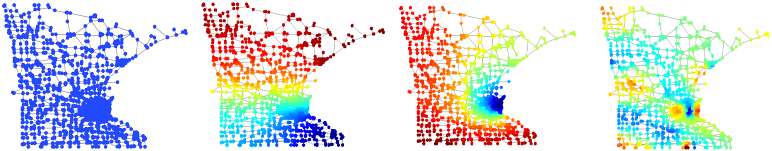
Graph Fourier Transform (GFT)

- Let $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ be the shift associated with G
- The Graph Fourier Transform (GFT) of $\mathbf{x}$ is defined as

$$\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$$

- While the inverse GFT (iGFT) of $\tilde{\mathbf{x}}$ is defined as $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$

$\mathbf{v}_1$ $\quad$ $\mathbf{v}_2$ $\quad$ $\mathbf{v}_3$ $\quad$ $\mathbf{v}_{20}$



$\lambda_1 = 0 \quad \leq \quad \lambda_2 \quad \leq \quad \lambda_3 \quad \leq \quad \lambda_{20}$

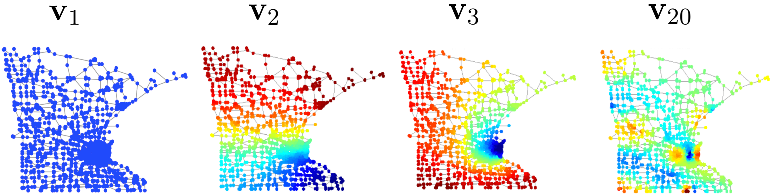
Graph Fourier Transform (GFT)

- Let $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ be the shift associated with G
- The Graph Fourier Transform (GFT) of $\mathbf{x}$ is defined as

$$\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$$

- While the inverse GFT (iGFT) of $\tilde{\mathbf{x}}$ is defined as $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$

$$\mathbf{v}_k^T \mathbf{L} \mathbf{v}_k = \sum_{(i,j)\in\mathcal{E}} A_{ij}([\mathbf{v}_k]_i - [\mathbf{v}_k]_j)^2 = \mathrm{TV}(\mathbf{v}_k)$$



$\lambda_1 = 0 \quad \leq \quad \lambda_2 \quad \leq \quad \lambda_3 \quad \leq \quad \lambda_{20}$

Graph Fourier Transform (GFT)

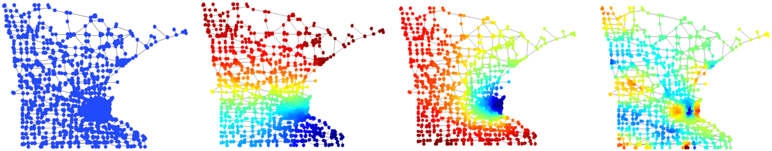
- Let $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ be the shift associated with G
- The Graph Fourier Transform (GFT) of $\mathbf{x}$ is defined as

$$\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$$

- While the inverse GFT (iGFT) of $\tilde{\mathbf{x}}$ is defined as $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$

$$\lambda_k = \lambda_k \mathbf{v}_k^T \mathbf{v}_k = \mathbf{v}_k^T \mathbf{L} \mathbf{v}_k = \sum_{(i,j)\in\mathcal{E}} A_{ij}([\mathbf{v}_k]_i - [\mathbf{v}_k]_j)^2 = \mathrm{TV}(\mathbf{v}_k)$$

$\mathbf{v}_1$ $\mathbf{v}_2$ $\mathbf{v}_3$ $\mathbf{v}_{20}$



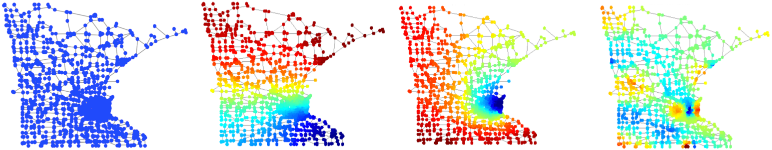
$\lambda_1 = 0 \quad \leq \quad \lambda_2 \quad \leq \quad \lambda_3 \quad \leq \quad \lambda_{20}$

- Let $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ be the shift associated with G
- The Graph Fourier Transform (GFT) of $\mathbf{x}$ is defined as

$$\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$$

- While the inverse GFT (iGFT) of $\tilde{\mathbf{x}}$ is defined as $\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$

$$\lambda_k = \lambda_k \mathbf{v}_k^T \mathbf{v}_k = \mathbf{v}_k^T \mathbf{L} \mathbf{v}_k = \sum_{(i,j)\in\mathcal{E}} A_{ij}([\mathbf{v}_k]_i - [\mathbf{v}_k]_j)^2 = \mathrm{TV}(\mathbf{v}_k)$$



Linear (shift-invariant) graph filter

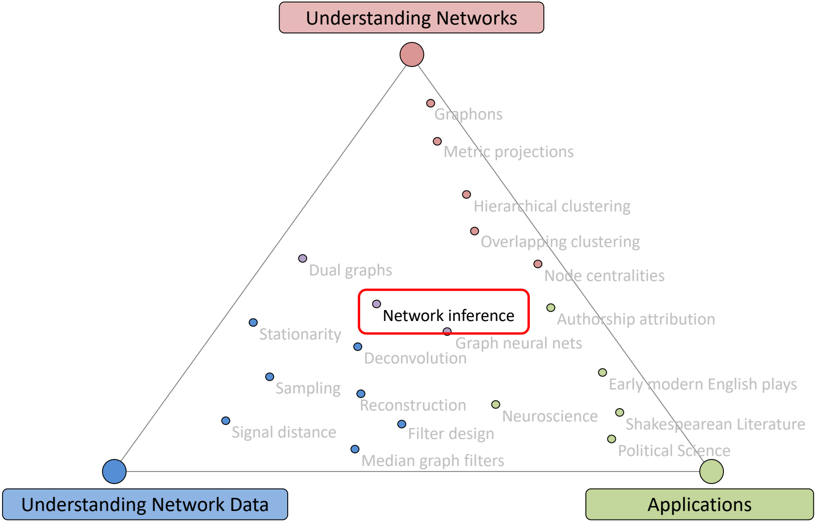
- A graph filter $H : \mathbb{R}^N \to \mathbb{R}^N$ is a map between graph signals of the form
- Polynomial in $\mathbf{S}$ of degree L, with coeff. $\mathbf{h} = [h_0, \ldots, h_L]^T$ [Sandryhaila-Moura13]

$$\mathbf{H} := h_0\mathbf{S}^0 + h_1\mathbf{S}^1 + \ldots + h_L\mathbf{S}^L = \sum_{l=0}^{L} h_l\mathbf{S}^l$$

- If $\mathbf{y} := \mathbf{H}\mathbf{x}$, Def 1 says $\mathbf{y} = \sum_{l=0}^{L} h_l\mathbf{x}^{(l)}$ (shifted versions of $\mathbf{x}$)

  $\Rightarrow \mathbf{x}^{(l)} := \mathbf{S}^l\mathbf{x} = \mathbf{S}\mathbf{x}^{(l-1)}$ can be found locally and sequentially

Network topology inference



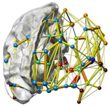
Network topology inference and GSP

- Network topology inference from nodal observations [Kolaczyk09]
  - ⇒ Approaches use Pearson correlations to construct graphs [Brovelli04]
  - ⇒ Partial correlations and conditional dependence [Friedman08]
  - ⇒ [Banerjee08], [Lake10], [Slawski15], [Meinshausen06], [Karanikolas16]

Network topology inference and GSP

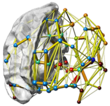
- Network topology inference from nodal observations [Kolaczyk09]
  - ⇒ Approaches use Pearson correlations to construct graphs [Brovelli04]
  - ⇒ Partial correlations and conditional dependence [Friedman08]
  - ⇒ [Banerjee08], [Lake10], [Slawski15], [Meinshausen06], [Karanikolas16]

- Key in neuroscience [Sporns10]
  - ⇒ Functional net inferred from activity



Network topology inference and GSP

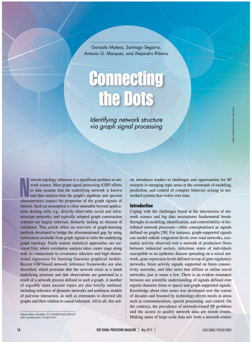
- Network topology inference from nodal observations [Kolaczyk09]
  - ⇒ Approaches use Pearson correlations to construct graphs [Brovelli04]
  - ⇒ Partial correlations and conditional dependence [Friedman08]
  - ⇒ [Banerjee08], [Lake10], [Slawski15], [Meinshausen06], [Karanikolas16]
- Key in neuroscience [Sporns10]
  - ⇒ Functional net inferred from activity

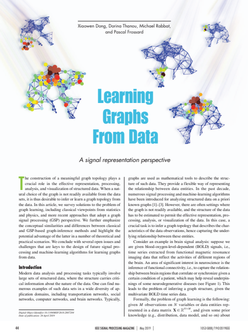


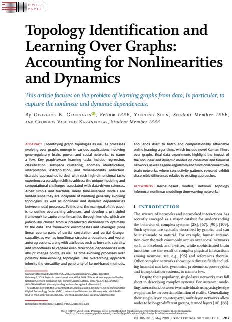
- Most GSP works assume that $\mathbf{S}$ (hence the graph) is known
  - ⇒ Analyze how the characteristics of $\mathbf{S}$ affect the signals and filters
- We take the reverse path
  - ⇒ How to use GSP to infer the graph topology? [TSIPN17] [SPMag19]
  - ⇒ [Dong16], [Kalofolias16], [Mei17], [Shen17], [Pasdeloup17], [Egilmez17]

Connecting the dots

- Recent tutorials on learning graphs from data
  - IEEE Signal Processing Magazine and Proceedings of the IEEE





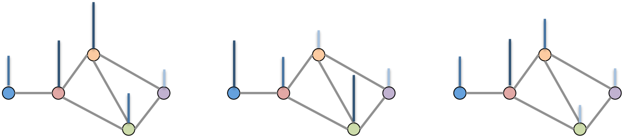


- IEEE Trans. on Signal and Information Processing over Networks
  - Issue on Network Topology Inference earlier this year

Problem formulation

Setup

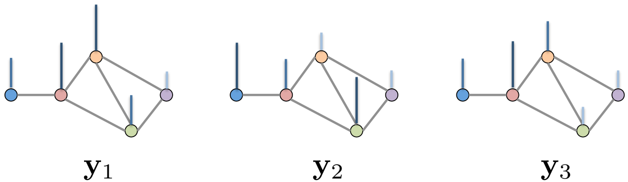
- Undirected network G with unknown graph shift $\mathbf{S}$
- Observe signals $\{\mathbf{y}_i\}_{i=1}^{P}$ defined on the unknown graph



$\mathbf{y}_1$ $\mathbf{y}_2$ $\mathbf{y}_3$

Setup

- Undirected network G with unknown graph shift $\mathbf{S}$
- Observe signals $\{\mathbf{y}_i\}_{i=1}^P$ defined on the unknown graph



Problem statement

> Given observations $\{\mathbf{y}_i\}_{i=1}^P$, determine the network $\mathbf{S}$ knowing that $\{\mathbf{y}_i\}_{i=1}^P$ are outputs of a diffusion process on $\mathbf{S}$.

Generating structure of a diffusion process

- Signal $\mathbf{y}_i$ is the response of a linear diffusion process to input $\mathbf{x}_i$

$$\mathbf{y}_i \;=\; \alpha_0 \prod_{l=1}^{\infty} (\mathbf{I} - \alpha_l \mathbf{S}) \mathbf{x}_i \;=\; \sum_{l=0}^{\infty} \beta_l \mathbf{S}^l \mathbf{x}_i, \quad i = 1, \ldots, P$$

$\Rightarrow$ Common generative model, e.g., heat diffusion, consensus

Generating structure of a diffusion process

- Signal $\mathbf{y}_i$ is the response of a linear diffusion process to input $\mathbf{x}_i$

$$\mathbf{y}_i \;=\; \alpha_0 \prod_{l=1}^{\infty}(\mathbf{I} - \alpha_l \mathbf{S})\mathbf{x}_i \;=\; \sum_{l=0}^{\infty} \beta_l \mathbf{S}^l \mathbf{x}_i, \quad i = 1, \ldots, P$$

$\Rightarrow$ Common generative model, e.g., heat diffusion, consensus

- Cayley-Hamilton asserts we can write diffusion as ($L \leq N$)

$$\mathbf{y}_i = \left(\sum_{l=0}^{L-1} h_l \mathbf{S}^l \right) \mathbf{x}_i := \mathbf{H}\mathbf{x}_i, \quad i = 1, \ldots, P$$

$\Rightarrow$ Graph filter $\mathbf{H}$ is shift invariant [Sandryhaila-Moura'13]

$\Rightarrow$ $\mathbf{H}$ diagonalized by the eigenvectors $\mathbf{V}$ of the shift operator

Generating structure of a diffusion process

- Signal $\mathbf{y}_i$ is the response of a linear diffusion process to input $\mathbf{x}_i$

$$\mathbf{y}_i \;=\; \alpha_0 \prod_{l=1}^{\infty} (\mathbf{I} - \alpha_l \mathbf{S}) \mathbf{x}_i \;=\; \sum_{l=0}^{\infty} \beta_l \mathbf{S}^l \mathbf{x}_i, \quad i = 1, \ldots, P$$

$\Rightarrow$ Common generative model, e.g., heat diffusion, consensus

- Cayley-Hamilton asserts we can write diffusion as ($L \leq N$)

$$\mathbf{y}_i = \left(\sum_{l=0}^{L-1} h_l \mathbf{S}^l \right) \mathbf{x}_i := \mathbf{H} \mathbf{x}_i, \quad i = 1, \ldots, P$$

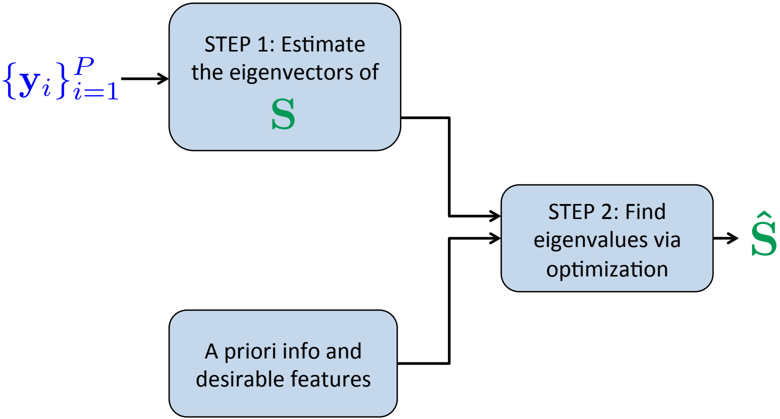
$\Rightarrow$ Graph filter $\mathbf{H}$ is shift invariant [Sandryhaila-Moura'13]

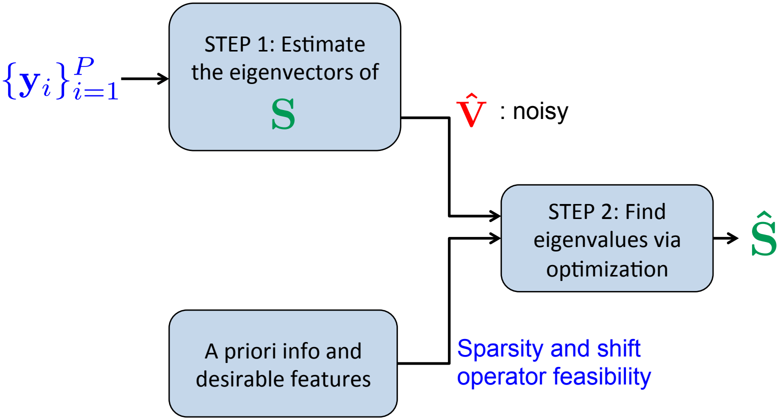
$\Rightarrow$ $\mathbf{H}$ diagonalized by the eigenvectors $\mathbf{V}$ of the shift operator

- **Goal**: estimate undirected network $\mathbf{S}$ from signal realizations $\{\mathbf{y}_i\}_{i=1}^P$

$\Rightarrow$ Unknowns: filter order L, coefficients $\{h_l\}_{l=1}^{L-1}$, inputs $\{\mathbf{x}_i\}_{i=1}^P$

Blueprint of our solution




$\{\mathbf{y}_i\}_{i=1}^P$
STEP 1: Estimate the eigenvectors of $\mathbf{S}$
$\hat{\mathbf{V}}$: noisy
STEP 2: Find eigenvalues via optimization
$\hat{\mathbf{S}}$
A priori info and desirable features
Sparsity and shift operator feasibility

Step 1: Obtaining the eigenvectors of $\mathbf{S}$

- $\mathbf{y}$ is the output of a local diffusion of a white input

$$\mathbf{y} = \alpha_0 \prod_{l=1}^{\infty} (\mathbf{I} - \alpha_l \mathbf{S})\mathbf{x} = \left(\sum_{l=0}^{N-1} h_l \, \mathbf{S}^l \right) \mathbf{x} := \mathbf{H}\mathbf{x}$$

Step 1: Obtaining the eigenvectors of $\mathbf{S}$

- $\mathbf{y}$ is the output of a local diffusion of a white input

$$\mathbf{y} = \alpha_0 \prod_{l=1}^{\infty} (\mathbf{I} - \alpha_l \mathbf{S})\mathbf{x} = \left(\sum_{l=0}^{N-1} h_l \, \mathbf{S}^l \right) \mathbf{x} := \mathbf{H}\mathbf{x}$$

- The covariance $\mathbf{C}_y$ of $\mathbf{y}$ shares $\mathbf{V}$ with $\mathbf{S}$

$$\mathbf{C}_y = \mathbf{H}^2 = h_0^2 \mathbf{I} + 2h_0 h_1 \mathbf{S} + h_1^2 \mathbf{S}^2 + \ldots$$

Step 1: Obtaining the eigenvectors of $\mathbf{S}$

- $\mathbf{y}$ is the output of a local diffusion of a white input

$$\mathbf{y} = \alpha_0 \prod_{l=1}^{\infty} (\mathbf{I} - \alpha_l \mathbf{S})\mathbf{x} = \left(\sum_{l=0}^{N-1} h_l \, \mathbf{S}^l \right) \mathbf{x} := \mathbf{H}\mathbf{x}$$

- The covariance $\mathbf{C}_y$ of $\mathbf{y}$ shares $\mathbf{V}$ with $\mathbf{S}$

$$\mathbf{C}_y = \mathbf{H}^2 = h_0^2 \mathbf{I} + 2 h_0 h_1 \mathbf{S} + h_1^2 \mathbf{S}^2 + \ldots$$

- Mapping $\mathbf{S} \to \mathbf{C}_y$ is polynomial

  $\Rightarrow$ Correlation methods $\Rightarrow \mathbf{C}_y = \mathbf{S}$

  $\Rightarrow$ Precision methods (graphical Lasso) $\rightarrow \mathbf{C}_y = \mathbf{S}^{-1}$

  $\Rightarrow$ Structural EM methods $\Rightarrow \mathbf{C}_y = (\mathbf{I} - \mathbf{S})^{-2}$

Step 2: Convex recovery of the eigenvalues

- Use extra knowledge/assumptions to find the eigenvalues
  - $\Rightarrow$ Of all graphs, select one that is optimal in some sense

$$\mathbf{S}_0^* := \operatorname*{argmin}_{\mathbf{S}, \boldsymbol{\lambda}} \|\mathbf{S}\|_0 \quad \text{s. to} \quad \mathbf{S} = \sum_{k=1}^{N} \lambda_k \mathbf{v}_k \mathbf{v}_k^T, \;\; \mathbf{S} \in \mathcal{S}$$

- Set $\mathcal{S}$ contains all admissible scaled adjacency matrices

$$\mathcal{S} := \{\mathbf{S} \,|\, S_{ij} \geq 0, \;\; \mathbf{S} \in \mathcal{M}^N, \; S_{ii} = 0, \;\; \textstyle\sum_j S_{1j} = 1\}$$

Step 2: Convex recovery of the eigenvalues

- Use extra knowledge/assumptions to find the eigenvalues
  - $\Rightarrow$ Of all graphs, select one that is optimal in some sense

$$\mathbf{S}_0^* := \underset{\mathbf{S},\boldsymbol{\lambda}}{\operatorname{argmin}} \ \|\mathbf{S}\|_0 \quad \text{s. to} \quad \mathbf{S} = \sum_{k=1}^{N} \lambda_k \mathbf{v}_k \mathbf{v}_k^T, \ \ \mathbf{S} \in \mathcal{S}$$

- Set $\mathcal{S}$ contains all admissible scaled adjacency matrices

$$\mathcal{S} := \{\mathbf{S} \,|\, S_{ij} \geq 0, \ \mathbf{S} \in \mathcal{M}^N, \ S_{ii} = 0, \ \textstyle\sum_j S_{1j} = 1\}$$

- Non-convex problem, relax to ℓ_1-norm minimization, e.g., [Tropp06]

$$\mathbf{S}_1^* := \underset{\mathbf{S},\boldsymbol{\lambda}}{\operatorname{argmin}} \ \|\mathbf{S}\|_1 \quad \text{s. to} \quad \mathbf{S} = \sum_{k=1}^{N} \lambda_k \mathbf{v}_k \mathbf{v}_k^T, \ \ \mathbf{S} \in \mathcal{S}$$

- Does the solution $\mathbf{S}_1^*$ coincide with the ℓ_0 solution $\mathbf{S}_0^*$?

- Define $\mathbf{W} := \mathbf{V} \odot \mathbf{V}$
- Build $\mathbf{M} := (\mathbf{I} - \mathbf{W}\mathbf{W}^{\dagger})_{\mathcal{D}^c}$ the orthogonal projector onto $\mathrm{range}(\mathbf{W})$
  - $\Rightarrow$ Construct $\mathbf{R} := [\mathbf{M}, \;\; \mathbf{e}_1 \otimes \mathbf{1}_{N-1}]$
  - $\Rightarrow$ Denote by $\mathcal{K}$ the indices of the support of $\mathbf{s}_0^* = \mathrm{vec}(\mathbf{S}_0^*)$

$\mathbf{S}_1^*$ and $\mathbf{S}_0^*$ coincide if the two following conditions are satisfied:
1) $\mathrm{rank}(\mathbf{R}_{\mathcal{K}}) = |\mathcal{K}|$; and
2) There exists a constant $\delta > 0$ such that

$$\psi_{\mathbf{R}} := \|\mathbf{I}_{\mathcal{K}^c}(\delta^{-2}\mathbf{R}\mathbf{R}^T + \mathbf{I}_{\mathcal{K}^c}^T\mathbf{I}_{\mathcal{K}^c})^{-1}\mathbf{I}_{\mathcal{K}}^T\|_{\infty} < 1.$$

- Cond. 1) ensures uniqueness of solution $\mathbf{S}_1^*$
- Cond. 2) guarantees existence of a dual certificate for ℓ_0 optimality

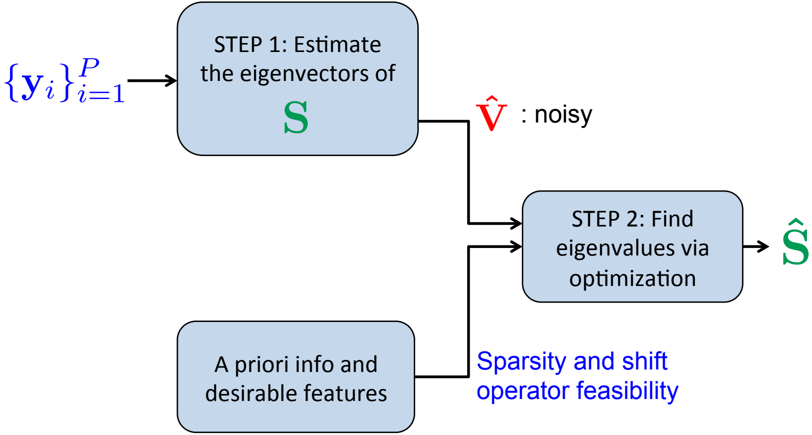
Recovery guarantee for ℓ_1 relaxation

- Define $\mathbf{W} := \mathbf{V} \odot \mathbf{V}$
- Build $\mathbf{M} := (\mathbf{I} - \mathbf{W}\mathbf{W}^\dagger)_{\mathcal{D}^c}$ the orthogonal projector onto $\mathrm{range}(\mathbf{W})$
  - $\Rightarrow$ Construct $\mathbf{R} := [\mathbf{M}, \;\; \mathbf{e}_1 \otimes \mathbf{1}_{N-1}]$
  - $\Rightarrow$ Denote by $\mathcal{K}$ the indices of the support of $\mathbf{s}_0^* = \mathrm{vec}(\mathbf{S}_0^*)$

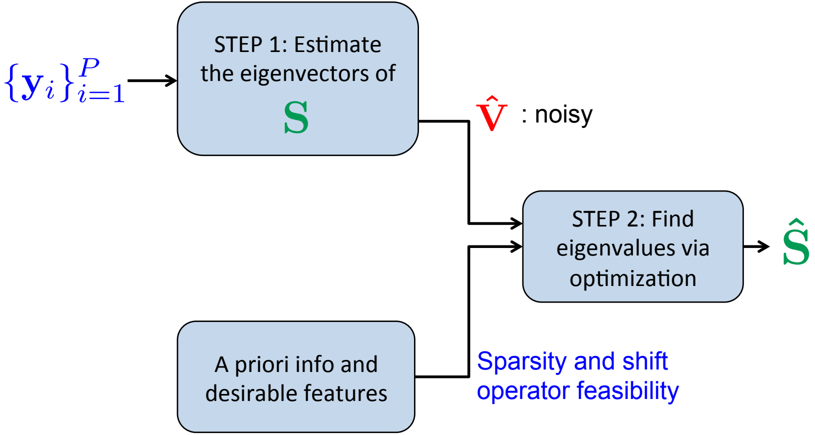
$\mathbf{S}_1^*$ and $\mathbf{S}_0^*$ coincide if the two following conditions are satisfied:
1) $\mathrm{rank}(\mathbf{R}_{\mathcal{K}}) = |\mathcal{K}|$; and
2) There exists a constant $\delta > 0$ such that

$$\psi_{\mathbf{R}} := \|\mathbf{I}_{\mathcal{K}^c}(\delta^{-2}\mathbf{R}\mathbf{R}^T + \mathbf{I}_{\mathcal{K}^c}^T\mathbf{I}_{\mathcal{K}^c})^{-1}\mathbf{I}_{\mathcal{K}}^T\|_\infty < 1.$$

- Cond. 1) ensures uniqueness of solution $\mathbf{S}_1^*$
- Cond. 2) guarantees existence of a dual certificate for ℓ_0 optimality


$\{\mathbf{y}_i\}_{i=1}^P$
STEP 1: Estimate the eigenvectors of $\mathbf{S}$
$\hat{\mathbf{V}}$: noisy
STEP 2: Find eigenvalues via optimization
$\hat{\mathbf{S}}$
A priori info and desirable features
Sparsity and shift operator feasibility

Robust shift identification



- Q1: How to modify the optimization problem to make it robust?
- Q2: Recovery guarantees in this robust setting?

Noisy eigenvectors

- We might have access to $\hat{\mathbf{V}}$, a noisy version of the eigenvectors
- With $d(\cdot,\cdot)$ denoting a (convex) distance between matrices

$$\min_{\{\mathbf{S},\boldsymbol{\lambda},\hat{\mathbf{S}}\}} \|\mathbf{S}\|_1 \quad \text{s. to} \;\; \hat{\mathbf{S}} = \textstyle\sum_{k=1}^{N} \lambda_k \hat{\mathbf{v}}_k \hat{\mathbf{v}}_k{}^T, \;\; \mathbf{S} \in \mathcal{S}, \;\; d(\mathbf{S}, \hat{\mathbf{S}}) \leq \epsilon$$

- How does the noise in $\hat{\mathbf{V}}$ affect the recovery?

Noisy eigenvectors

- We might have access to $\hat{\mathbf{V}}$, a noisy version of the eigenvectors
- With $d(\cdot,\cdot)$ denoting a (convex) distance between matrices

$$\min_{\{\mathbf{S},\boldsymbol{\lambda},\hat{\mathbf{S}}\}} \|\mathbf{S}\|_1 \quad \text{s. to} \quad \hat{\mathbf{S}} = \sum_{k=1}^{N} \lambda_k \hat{\mathbf{v}}_k \hat{\mathbf{v}}_k{}^T, \quad \mathbf{S} \in \mathcal{S}, \quad d(\mathbf{S}, \hat{\mathbf{S}}) \leq \epsilon$$

- How does the noise in $\hat{\mathbf{V}}$ affect the recovery?
- Conditions 1) and 2) but based on $\hat{\mathbf{R}}$, guaranteed

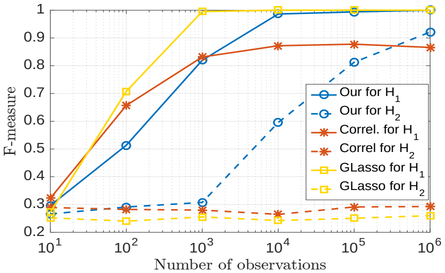
$$d(\mathbf{S}^*, \mathbf{S}_0^*) \leq C\epsilon$$

$\Rightarrow$ ϵ large enough to guarantee feasibility of $\mathbf{S}_0^*$

$\Rightarrow$ Constant C depends on $\hat{\mathbf{V}}$ and the support $\mathcal{K}$

Performance comparisons

- Comparison with graphical lasso and sparse correlation methods
  - Evaluated on 100 realizations of ER graphs with $N = 20$ and $p = 0.2$



- Graphical lasso implicitly assumes a filter $\mathbf{H}_1 = (\rho\mathbf{I} + \mathbf{S})^{-1/2}$

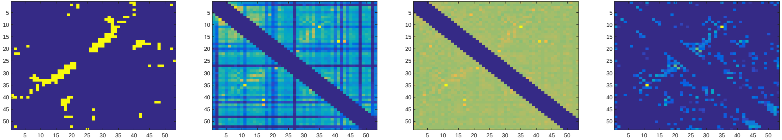
  $\Rightarrow$ For this filter our method works, but not as well

- For general diffusion filters $\mathbf{H}_2$ our method still works fine

- Our method can be used to sparsify a given network
  - $\Rightarrow$ Keep direct and important edges or relations
  - $\Rightarrow$ Discard indirect relations that can be explained by direct ones
- Use eigenvectors $\hat{\mathbf{V}}$ of given network as noisy eigenvectors of $\mathbf{S}$

Ex: Infer contact between amino-acid residues in BPT1 BOVIN

$\Rightarrow$ Use mutual information of amino-acid covariation as input

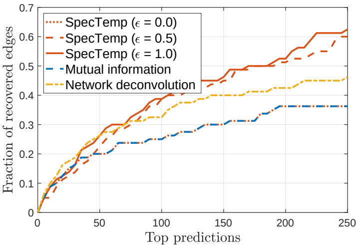


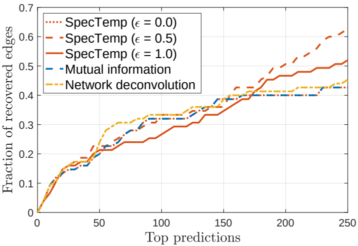
Ground truth | Mutual info. | Network deconv. | Our approach

- Network deconvolution assumes a specific filter model [Feizi13]
  - $\Rightarrow$ We achieve better performance by being agnostic to this

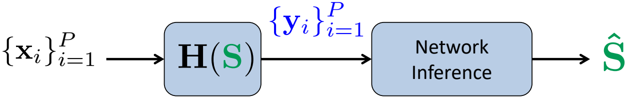
Sensitivity of recovered edges

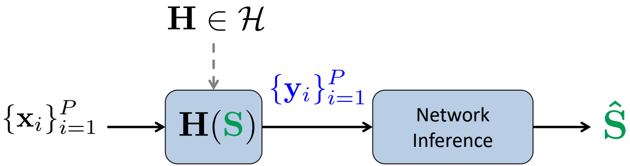
- Sensitivity of the top edge predictions
  - $\Rightarrow$ Fraction of the real contact edges recovered
- For $\epsilon = 0$ we force $\mathbf{S}$ to be mutual information matrix $\mathbf{S}'$
- For larger values of ϵ, we get a better recovery





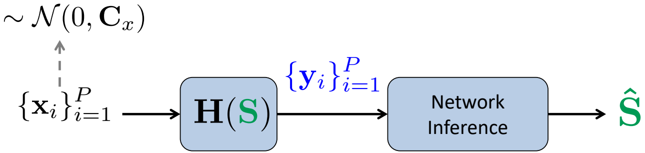
A rich framework for network inference





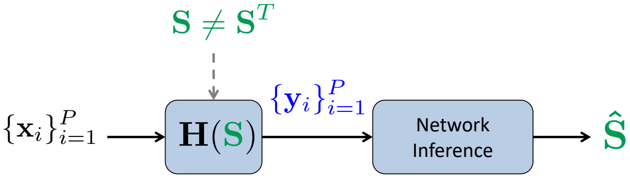
- Prior knowledge on the filter class [Segarra et al'17] [Zhu et al'20]

A rich framework for network inference



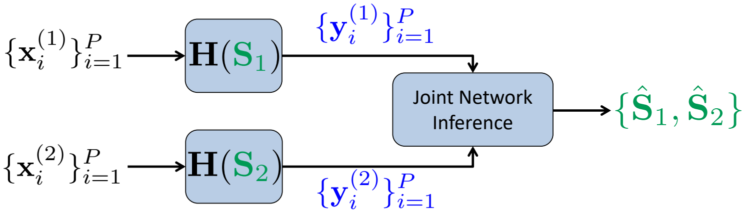
- Prior knowledge on the filter class [Segarra et al'17] [Zhu et al'20]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]

A rich framework for network inference



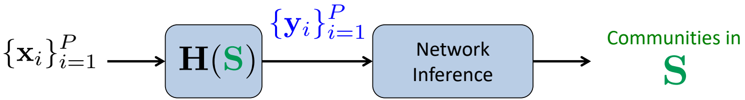
- Prior knowledge on the filter class [Segarra et al'17] [Zhu et al'20]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]

A rich framework for network inference

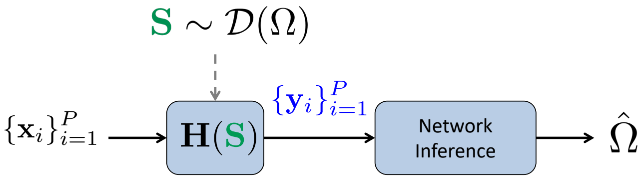


- Prior knowledge on the filter class [Segarra et al'17] [Zhu et al'20]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]
- Joint inference of multiple networks [Segarra et al'17]

A rich framework for network inference

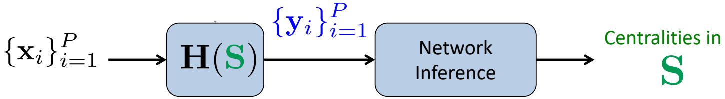


- Prior knowledge on the filter class [Segarra et al'17] [Zhu et al'20]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]
- Joint inference of multiple networks [Segarra et al'17]
- Recovering the community structure [Wai'18,'19] [Roddenberry'20]



- Prior knowledge on the filter class [Segarra et al'17] [Zhu et al'20]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]
- Joint inference of multiple networks [Segarra et al'17]
- Recovering the community structure [Wai'18,'19] [Roddenberry'20]
- Estimating the graph model coefficients [Schaub et al'20]

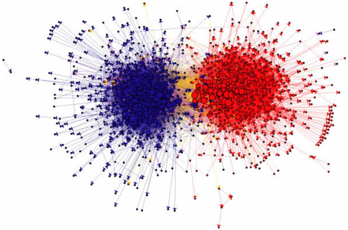
A rich framework for network inference



- Prior knowledge on the filter class [Segarra et al'17] [Zhu et al'20]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]
- Joint inference of multiple networks [Segarra et al'17]
- Recovering the community structure [Wai'18,'19] [Roddenberry'20]
- Estimating the graph model coefficients [Schaub et al'20]
- Recovering the node centralities [Roddenberry et al'20] [He at al'20]

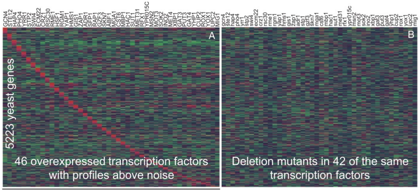
- **Low-rankness** is a prevalent feature in graph signals / network data

Social network data / opinion:



- **Polarization** is common in social networks' opinions

Gene network data:

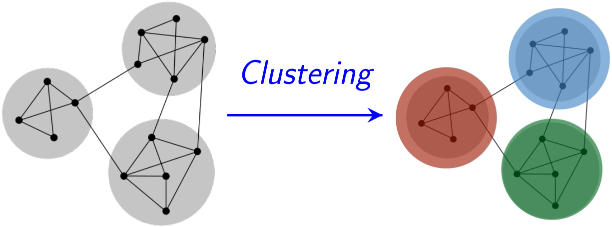


[Chua et al., PNAS, 2006]

- Number of experiments available is limited due to time and labor cost.

- **Bad news**: oftentimes we do not have **full-rank** data

Remedy: **relax** the goal of learning the whole graph



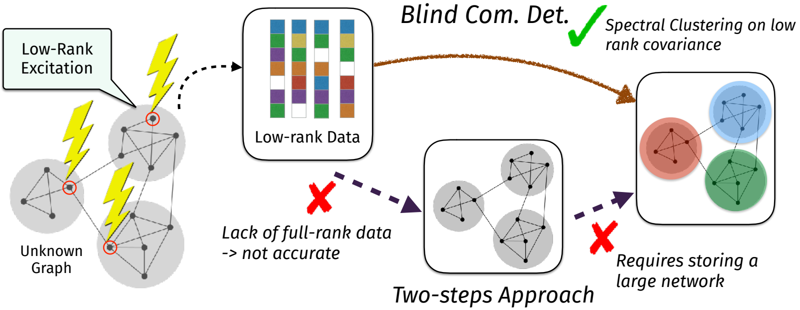
- Understanding the **community structure** gives a **macroscopic** (or reduced resolution) view of the graph

  $\Rightarrow$ useful for network analysis and influence maximization

- Typical approach requires **perfect knowledge** of the graph

Blind community detection

- A **Blind Community Detection** method to detect communities from graph signals without learning/storing the graph itself



Problem statement

Given observations $\{\mathbf{y}_i\}_{i=1}^{P}$, determine the communities in G when:
(AS1) $\{\mathbf{y}_i\}_{i=1}^{P}$ are the outputs of a low-rank diffusion on G

Given observations $\{\mathbf{y}_i\}_{i=1}^{P}$, determine the communities in G when:
(AS1) $\{\mathbf{y}_i\}_{i=1}^{P}$ are the outputs of a low-rank diffusion on G

$$\mathbf{y} = \sum_{\ell=0}^{L} h_\ell \mathbf{S}^\ell \mathbf{w} = \mathbf{H}(\mathbf{S})\mathbf{w}$$

Problem statement

Given observations $\{\mathbf{y}_i\}_{i=1}^{P}$, determine the communities in G when:
(AS1) $\{\mathbf{y}_i\}_{i=1}^{P}$ are the outputs of a low-rank diffusion on G

$$\mathbf{y} = \sum_{\ell=0}^{L} h_\ell \mathbf{S}^\ell \mathbf{w} = \mathbf{H}(\mathbf{S})\mathbf{w}$$

$$\mathbf{w} = \mathbf{B}\mathbf{z}, \quad \mathbf{B} \in \mathbb{R}^{N \times R}, \quad R \ll N.$$

Problem statement

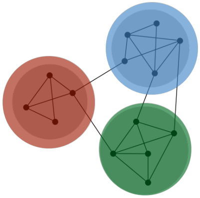
Given observations $\{\mathbf{y}_i\}_{i=1}^{P}$, determine the communities in G when:
(AS1) $\{\mathbf{y}_i\}_{i=1}^{P}$ are the outputs of a low-rank diffusion on G

$$\mathbf{y} = \sum_{\ell=0}^{L} h_\ell \mathbf{S}^\ell \mathbf{w} = \mathbf{H}(\mathbf{S})\mathbf{w}$$

$$\mathbf{w} = \mathbf{B}\mathbf{z}, \quad \mathbf{B} \in \mathbb{R}^{N \times R}, \quad R \ll N.$$

- Additional unknowns:
- The filter $\mathbf{H}$ $\Rightarrow$ Unknown L and $\{h_\ell\}_{\ell=0}^{L}$
- Tall matrix $\mathbf{B}$
- Input $\mathbf{z}$ $\Rightarrow$ Assume statistical knowledge $\mathbb{E}[\mathbf{z}\mathbf{z}^\top] = \mathbf{I}$

- Undirected graph $G = (V, E, \mathbf{A})$
- Let $V = \mathcal{C}_1 \cup \cdots \cup \mathcal{C}_K$ with $\mathcal{C}_k \cap \mathcal{C}_{k'} = \emptyset, k \neq k'$.
- "Best" clustering is achieved by **minimizing**:



$$\mathrm{RatioCut}(\mathcal{C}_1, ..., \mathcal{C}_K) := \sum_{k=1}^{K} \frac{1}{|\mathcal{C}_k|} \sum_{i \in \mathcal{C}_k} \sum_{j \in \overline{\mathcal{C}}_k} A_{ij}$$

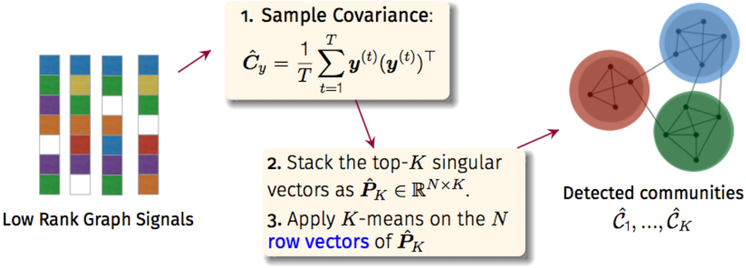
- **Spectral clustering** tackles it via **low rank approximation** —

Let $\mathbf{S} := \mathrm{Diag}(\mathbf{A1}) - \mathbf{A}$ be the *graph Laplacian*, and $\mathbf{V}_K \in \mathbb{R}^{N \times K}$ as the collection of its **smallest K singular vectors**. Perform **K-means** to:

$$\min_{\mathcal{C}_1, ..., \mathcal{C}_K} F(\mathcal{C}_1, ..., \mathcal{C}_K) := \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \left\| \mathbf{v}_i^{\mathrm{row}} - \frac{1}{|\mathcal{C}_k|} \sum_{j \in \mathcal{C}_k} \mathbf{v}_j^{\mathrm{row}} \right\|_2^2,$$

where $\mathbf{v}_i^{\mathrm{row}}$ is the ith row vector of $\mathbf{V}_K$.

Blind Community Detection (BlindCD):

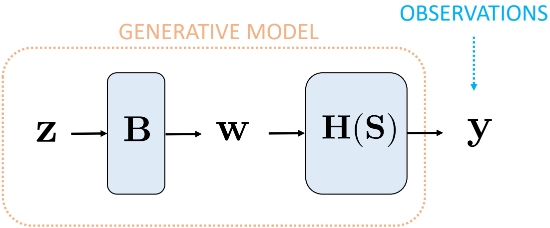


- A **blind** method as the graph and graph filter are unknown.
- If $\hat{\mathbf{P}}_K$ spans the **same subspace** as $\mathbf{V}_K$, then BlindCD is accurate.

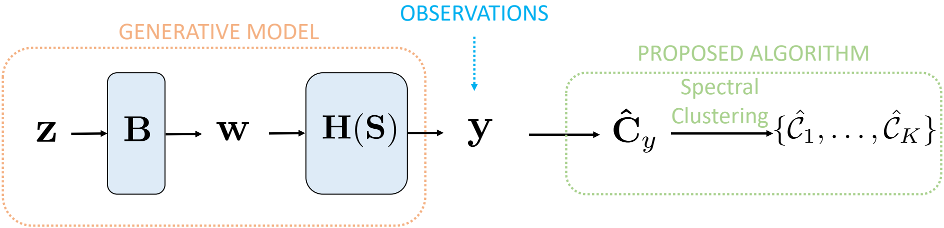
High-level view of the problem

$$\mathbf{z} \longrightarrow \boxed{\mathbf{B}} \longrightarrow \mathbf{w} \longrightarrow \boxed{\mathbf{H}(\mathbf{S})} \longrightarrow \mathbf{y}$$

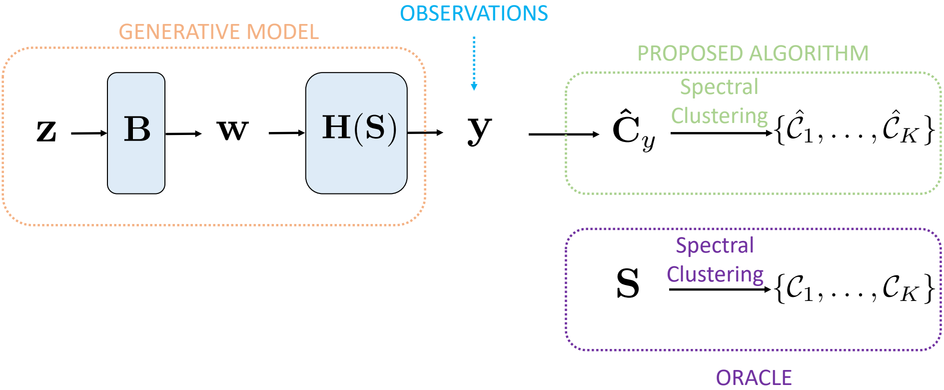
High-level view of the problem



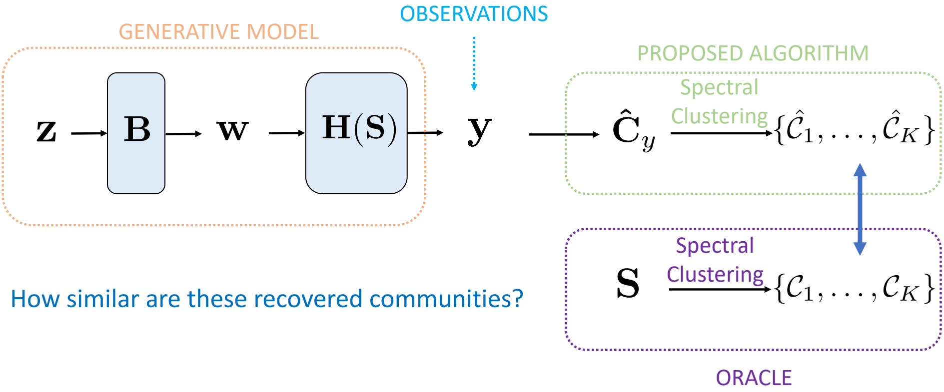
High-level view of the problem



High-level view of the problem



High-level view of the problem



How similar are these recovered communities?

Building up some intuition

- **Recall**: Spectral clustering require **the smallest K eigenvectors** of $\mathbf{S}$
- Specifically, we want $\mathbf{V}_K$ from $\mathbf{V} := (\mathbf{V}_K \;\; \mathbf{V}_{N-K})$ [recall $\mathbf{S} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^\top$].
- Under the low-rank model, $\mathbf{w} = \mathbf{B}\mathbf{z}$, we have

$$\mathbf{C}_y = \mathbf{H}\mathbf{B}\mathbf{B}^\top\mathbf{H}^\top = \mathbf{V}\big(\mathrm{Diag}(\tilde{\mathbf{h}})\big)(\mathbf{V}^\top\mathbf{B})(\mathbf{B}^\top\mathbf{V})\big(\mathrm{Diag}(\tilde{\mathbf{h}})\big)\mathbf{V}^\top$$

- Intuitively, the singular vectors $\mathbf{V}_K$ are in $\mathbf{C}_y$ if

$$\mathrm{rank}(\mathbf{V}_K^\top\mathbf{B}) = K, \;\; \mathbf{V}_{N-K}^\top\mathbf{B} \approx \mathbf{0}$$

$$\tilde{h}_i \begin{cases} \neq 0 & , \; i = 1, ..., K, \\ \approx 0 & , \; i \geq K+1. \end{cases}$$

Low-pass graph filters

Definition: a graph filter is said to be (K, η)-**low pass** if

$$\frac{\tilde{h}_{K+1}}{\tilde{h}_K} \leq \eta < 1, \quad \tilde{h}_1 \geq \tilde{h}_2 \geq \cdots \geq \tilde{h}_N \geq 0,$$

where $\tilde{h}_i := [\sum_{\ell=0}^{L-1} h_\ell \mathbf{\Lambda}^\ell]_{ii}$, $i = 1, ..., N$.

- An **ideal low-pass graph filter** have $\eta = 0$.
- **Example 1** — *consensus dynamics*

$$\mathbf{H}_1(\mathbf{S}) = (\mathbf{I} - \alpha\mathbf{S})^{L-1} \Longrightarrow \eta = \left(\frac{1 - \alpha\lambda_{K+1}}{1 - \alpha\lambda_K}\right)^{L-1}.$$

- **Example 2** — *steady-state of DeGroot dynamics*

$$\mathbf{H}_2(\mathbf{S}) = (\mathbf{I} + c^{-1}\mathbf{S})^{-1} \Longrightarrow \eta = \frac{1 + c^{-1}\lambda_K}{1 + c^{-1}\lambda_{K+1}}.$$

Theoretical guarantee of BlindCD

- Recall $F(\mathcal{C}_1, ..., \mathcal{C}_K) := \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} \left\| \mathbf{v}_i^{\text{row}} - \frac{1}{|\mathcal{C}_k|} \sum_{j \in \mathcal{C}_k} \mathbf{v}_j^{\text{row}} \right\|_2^2$.
- Let $F^\star := \min_{\mathcal{C}_1, ..., \mathcal{C}_K} F(\mathcal{C}_1, ..., \mathcal{C}_K)$ and $\hat{\mathcal{C}}_1, ..., \hat{\mathcal{C}}_K$ be the communities found by BlindCD, we have:

Under some conditions (e.g., $\|\mathbf{C}_y - \hat{\mathbf{C}}_y\|$ is small, $R \geq K$), we have

$$\sqrt{F(\hat{\mathcal{C}}_1, ..., \hat{\mathcal{C}}_K)} - \sqrt{F^\star} \leq \sqrt{8K} \left(\sqrt{\frac{\gamma^2}{1+\gamma^2}} + \frac{\|\mathbf{C}_y - \hat{\mathbf{C}}_y\|_2}{\delta} \right),$$

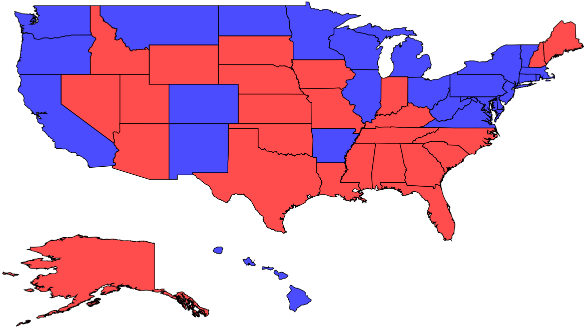
$$\text{where} \qquad \gamma \leq \eta \cdot \|\mathbf{V}_{N-K}^\top \mathbf{B} \mathbf{Q}_K\|_2 \cdot \|(\mathbf{V}_K^\top \mathbf{B} \mathbf{Q}_K)^{-1}\|_2,$$

and $\mathbf{Q}_K$ is a set of K orthogonal vectors.

- The error of BlindCD is decomposed into two parts —
  - **Non-ideal low-pass graph filter** and **mismatch between $\mathbf{V}_K$, $\mathbf{B}$**.
  - Due to **insufficient samples** in covariance estimation.

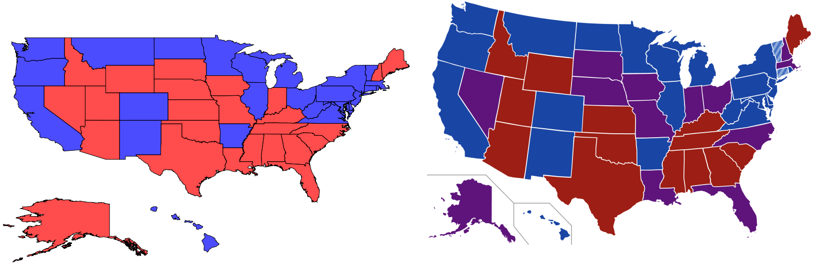
United States Senate data

- Rollcall data from the 110-114th congress of the US Senate (2007-2017)
- Infer partition of the network of $n = 50$ states of USA
- i-th rollcall data is mapped into a graph signal $\mathbf{y}_i \in \mathbb{R}^{50}$



United States Senate data

- Rollcall data from the 110-114th congress of the US Senate (2007-2017)
- Infer partition of the network of $n = 50$ states of USA
- i-th rollcall data is mapped into a graph signal $\mathbf{y}_i \in \mathbb{R}^{50}$



Conclusions

- GSP approach to network inference in the graph spectral domain
  - $\Rightarrow$ Two step approach: i) Obtain $\mathbf{V}$; ii) Estimate $\mathbf{S}$ given $\mathbf{V}$
- How to obtain the spectral templates $\mathbf{V}$
  - $\Rightarrow$ Based on covariance of diffused signals
  - $\Rightarrow$ Other sources: network operators, network deconvolution

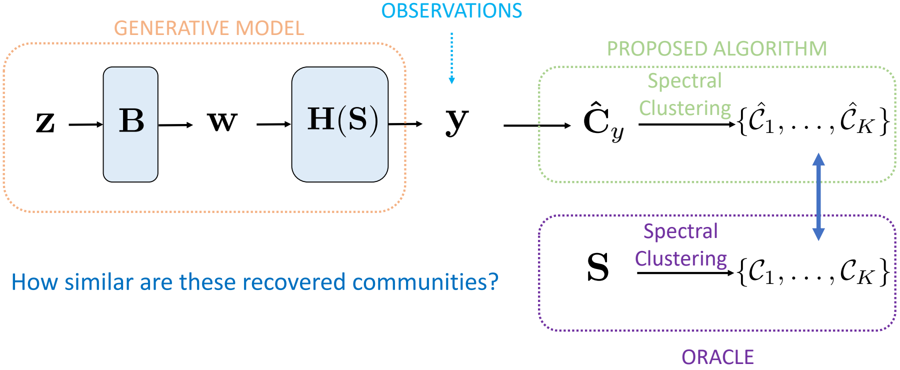
Conclusions

- GSP approach to network inference in the graph spectral domain
  - $\Rightarrow$ Two step approach: i) Obtain $\mathbf{V}$; ii) Estimate $\mathbf{S}$ given $\mathbf{V}$
- How to obtain the spectral templates $\mathbf{V}$
  - $\Rightarrow$ Based on covariance of diffused signals
  - $\Rightarrow$ Other sources: network operators, network deconvolution
- Infer $\mathbf{S}$ via convex optimization
  - $\Rightarrow$ Objectives promote desirable physical properties
  - $\Rightarrow$ Constraints encode a priori information on structure
  - $\Rightarrow$ Robust formulations for noisy (and incomplete) templates

Conclusions

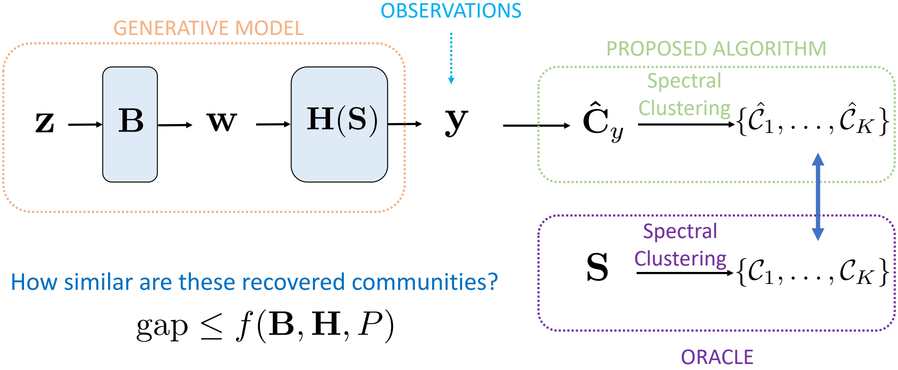
- Whenever recovering the whole graph is not feasible
  - $\Rightarrow$ What can data tell us about the graph?
  - $\Rightarrow$ Try to recover coarser features $\Rightarrow$ Communities

- Whenever recovering the whole graph is not feasible
  - $\Rightarrow$ What can data tell us about the graph?
  - $\Rightarrow$ Try to recover coarser features $\Rightarrow$ Communities



How similar are these recovered communities?

- Whenever recovering the whole graph is not feasible
  - $\Rightarrow$ What can data tell us about the graph?
  - $\Rightarrow$ Try to recover coarser features $\Rightarrow$ Communities



How similar are these recovered communities?

$$\text{gap} \leq f(\mathbf{B}, \mathbf{H}, P)$$

Thank you

Antonio Marques (K. Juan Carlos)
Gonzalo Mateos (U. Rochester)
Hoi-To Wai (CUHK)
Asuman Ozdaglar (MIT)
Michael Schaub (RWTH Aachen)
Yuhao Wang (MIT)
Yu Zhu (Rice)

Alejandro Ribeiro (UPenn)
Rasoul Shafipour (U. Rochester)
Anna Scaglione (ASU)
Ali Jadbabaie (MIT)
John Tsitsiklis (MIT)
Caroline Uhler (MIT)
T. Mitchell Roddenberry (Rice)

References: Journal papers (1)

- H. T. Wai, S. Segarra, A. E. Ozdaglar, A. Scaglione and A. Jadbabaie, "Blind Community Detection from Low-rank Excitations of a Graph Filter," *IEEE Trans. Signal Process.*, vol. 68, pp. 436 – 451, 2020.
- Y. Zhu, M. T. Schaub, A. Jadbabaie and S. Segarra, "Network Inference from Consensus Dynamics with Unknown Parameters," *IEEE Trans. Signal and Info. Process. over Networks*, vol. 6, pp. 300-315, 2020.
- M. T. Schaub, S. Segarra and J. N. Tsitsiklis, "Blind Identification of Stochastic Block Models from Dynamical Observations," *SIAM J. Math. Data Science (SIMODS)*, vol. 2, no. 2, pp. 335–367, 2020.
- G. Mateos, S. Segarra, A. G. Marques and A. Ribeiro, "Connecting the Dots: Identifying Network Structure via Graph Signal Processing," *IEEE Signal Process. Magazine*, vol. 36, no. 3, pp. 16-43, May 2019.

References: Journal papers (2)

- S. Segarra, A. G. Marques, G. Mateos and A. Ribeiro, “Network Topology Inference from Spectral Templates,” *IEEE Trans. Signal and Info. Process. over Networks*, vol. 3, no. 3, pp. 467-483, Sept. 2017.
- R. Shafipour, S. Segarra, A. G. Marques and G. Mateos, “Identifying the Topology of Undirected Networks from Diffused Non-stationary Graph Signals,” *arXiv:1801.03862*, 2020. (Under review OJSP)
- T. M. Roddenberry, M. T. Schaub, H. T. Wai and S. Segarra, “Exact Blind Community Detection from Signals on Multiple Graphs”, *arXiv:2001.10944*, 2020. (Under review TSP)

References: Conference papers (1)

- T. M. Roddenberry and S. Segarra, "Blind Inference of Centrality Rankings from Graph Signals", *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, pp. 5335-5339, Barcelona, Spain, May 4-8, 2020.
- M. T. Schaub, S. Segarra and H.T. Wai, "Spectral Partitioning of Time-Varying Networks with Unobserved Edges," *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, pp. 4938-4942, Brighton, UK, May 12-17, 2019.
- R. Shafipour, S. Segarra, A. G. Marques and G. Mateos, "Directed Network Topology Inference via Graph Filter Identification," *IEEE Data Science Workshop*, pp. 210-214, Lausanne, Switzerland, June 4-6, 2018.
- H. T. Wai, S. Segarra, A. E. Ozdaglar, A. Scaglione and A. Jadbabaie, "Community Detection from Low-rank Excitations of Graph Filters," *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, pp. 4044-4048, Calgary, Canada, April 15-20, 2018.

References: Conference papers (2)

- R. Shafipour, S. Segarra, A. G. Marques and G. Mateos, "Identifying Undirected Network Structure via Semidefinite Relaxation," *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, pp. 4049-4053, Calgary, Canada, April 15-20, 2018.
- S. Segarra, Y. Wang, C. Uhler and A. G. Marques, "Joint Inference of Networks from Stationary Graph Signals," *Asilomar Conference on Signals, Systems, and Computers*, pp. 975-979, Pacific Grove, CA, October 29 – November 1, 2017.
- R. Shafipour, S. Segarra, A. G. Marques and G. Mateos, "Network Topology Inference from Non-Stationary Graph Signals," *IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP)*, pp. 5870-5874, New Orleans, USA, March 5-9, 2017.
- T. M. Roddenberry and S. Segarra, "Blind Estimation of Eigenvector Centrality from Graph Signals: Beyond Low-pass Filtering", *Asilomar Conference on Signals, Systems, and Computers*, 2020. (submitted)