Connecting the Dots: Identifying Network Structure via Graph Signal Processing

Gonzalo Mateos and Santiago Segarra

University of Rochester — Rice University
gmateosb@ece.rochester.edu — segarra@rice.edu
http://www.ece.rochester.edu/~gmateosb/
http://segarra.rice.edu/

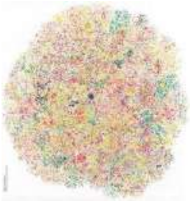
Collaborators: R. Shafipour, A. G. Marques, and A. Ribeiro
Acknowledgment: NSF Awards CCF-1750428 and ECCS-1809356

A Coruña, Spain, September 2, 2019

Network Science analytics

Online social media

Internet



Clean energy and grid analytics

- Network as graph $G = (\mathcal{V}, \mathcal{E})$: encode pairwise relationships
- Desiderata: Process, analyze and learn from network data [Kolaczyk'09]
  - $\Rightarrow$ Use G to study graph signals, data associated with nodes in $\mathcal{V}$
- Ex: Opinion profile, buffer congestion levels, neural activity, epidemic

Roadmap

Graph signal processing: Motivation and fundamentals

Statistical methods for network topology inference

Learning graphs from observations of smooth signals

Identifying the structure of network diffusion processes

Discussion

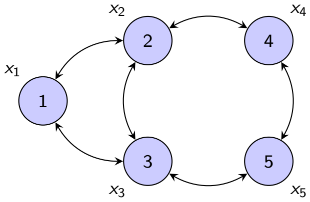
Graph signal processing (GSP)

- Graph G with adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$

  $\Rightarrow A_{ij}$ = proximity between i and j

- Define a signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph

  $\Rightarrow x_i$ = signal value at node i

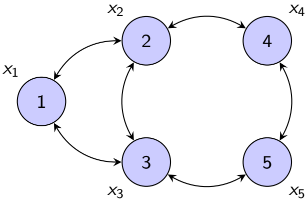


- Graph Signal Processing $\rightarrow$ Exploit structure encoded in $\mathbf{A}$ to process $\mathbf{x}$

  $\Rightarrow$ Our view: GSP well suited to study (network) diffusion processes

Graph signal processing (GSP)

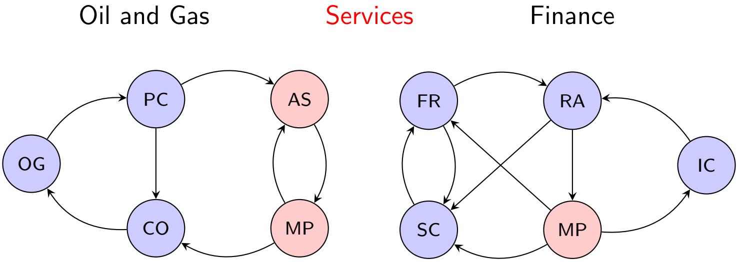
- Graph G with adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$
  - $\Rightarrow A_{ij}$ = proximity between i and j
- Define a signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph
  - $\Rightarrow x_i$ = signal value at node i



- Graph Signal Processing $\rightarrow$ Exploit structure encoded in $\mathbf{A}$ to process $\mathbf{x}$
  - $\Rightarrow$ Our view: GSP well suited to study (network) diffusion processes
- **Q:** Graph signals common and interesting as networks are?
- **Q:** Why do we expect the graph structure to be useful in processing $\mathbf{x}$?

Network of economic sectors of the United States

- Bureau of Economic Analysis of the U.S. Department of Commerce
  - A_{ij} = Output of sector i that becomes input to sector j (62 sectors)

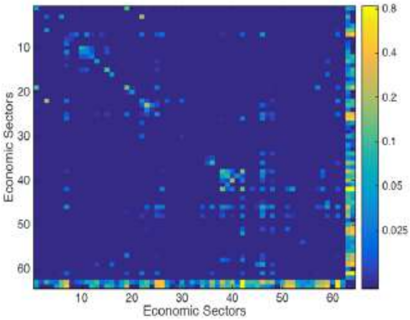


- Oil extraction (OG), Petroleum and coal products (PC), Construction (CO)
- Administrative services (AS), Professional services (MP)
- Credit intermediation (FR), Securities (SC), Real state (RA), Insurance (IC)

- Only interactions stronger than a threshold are shown

Network of economic sectors of the United States

- Bureau of Economic Analysis of the U.S. Department of Commerce
  - A_{ij} = Output of sector i that becomes input to sector j (62 sectors)

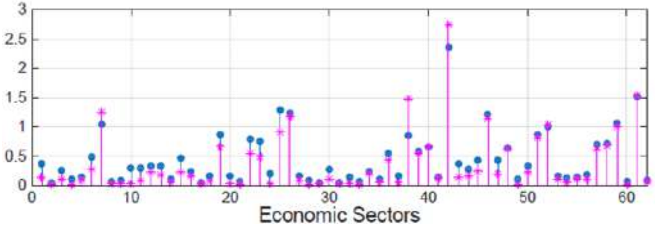


- A few sectors have widespread strong influence (services, finance, energy)
- Some sectors have strong indirect influences (oil)
- The heavy last row is final consumption

- This is an interesting network $\Rightarrow$ Signals on this graph are as well

Disaggregated GDP of the United States

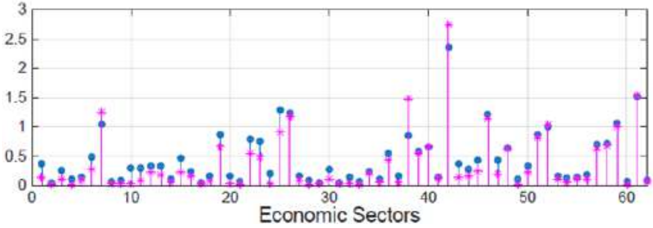
- Signal $\mathbf{x}$ = output per sector = disaggregated GDP
  - $\Rightarrow$ Network structure used to, e.g., reduce GDP estimation noise



- Signal is as interesting as the network itself. Arguably more

Disaggregated GDP of the United States

- Signal $\mathbf{x}$ = output per sector = disaggregated GDP
  - $\Rightarrow$ Network structure used to, e.g., reduce GDP estimation noise

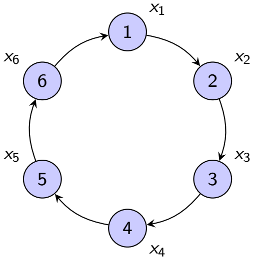


- Signal is as interesting as the network itself. Arguably more
  - Same is true for brain connectivity and fMRI brain signals, ...
  - Gene regulatory networks and gene expression levels, ...
  - Online social networks and information cascades, ...

Importance of signal structure in time

- Signal and Information Processing is about exploiting signal structure

- Discrete time described by cyclic graph
  - $\Rightarrow$ Time n follows time $n-1$
  - $\Rightarrow$ Signal value x_n similar to x_{n-1}
- Formalized with the notion of frequency



- Cyclic structure $\Rightarrow$ Fourier transform $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}^H \mathbf{x}$ $\left(F_{kn} = \dfrac{e^{j2\pi kn/N}}{\sqrt{N}} \right)$
- **Fourier transform** $\Rightarrow$ Projection on eigenvector space of cycle

Covariances and principal components

- Random signal with mean $\mathbb{E}[\mathbf{x}] = 0$ and covariance $\mathbf{C}_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right]$

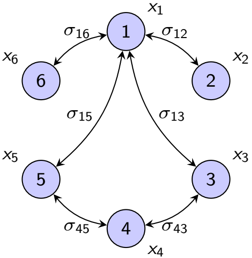
  $\Rightarrow$ Eigenvector decomposition $\mathbf{C}_x = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^H$

- Covariance matrix $\mathbf{A} = \mathbf{C}_x$ is a graph

  $\Rightarrow$ Not a very good graph, but still

- Precision matrix $\mathbf{C}_x^{-1}$ a common graph too

  $\Rightarrow$ Conditional dependencies of Gaussian $\mathbf{x}$

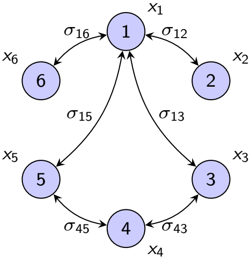


- Covariance matrix structure $\Rightarrow$ Principal components (PCA) $\Rightarrow \tilde{\mathbf{x}} = \mathbf{V}^H\mathbf{x}$

- **PCA transform** $\Rightarrow$ Projection on eigenvector space of (inverse) covariance

Covariances and principal components

- Random signal with mean $\mathbb{E}[\mathbf{x}] = 0$ and covariance $\mathbf{C}_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right]$
  - $\Rightarrow$ Eigenvector decomposition $\mathbf{C}_x = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$
- Covariance matrix $\mathbf{A} = \mathbf{C}_x$ is a graph
  - $\Rightarrow$ Not a very good graph, but still
- Precision matrix $\mathbf{C}_x^{-1}$ a common graph too
  - $\Rightarrow$ Conditional dependencies of Gaussian $\mathbf{x}$



- Covariance matrix structure $\Rightarrow$ Principal components (PCA) $\Rightarrow \tilde{\mathbf{x}} = \mathbf{V}^H\mathbf{x}$
- **PCA transform** $\Rightarrow$ Projection on eigenvector space of (inverse) covariance
- **Q:** Can we extend these principles to general graphs and signals?

Graph Fourier Transform

- Adjacency $\mathbf{A}$, Laplacian $\mathbf{L}$, or, generically graph shift $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$

  $\Rightarrow S_{ij} = 0$ for $i \neq j$ and $(i,j) \notin \mathcal{E}$ (captures local structure in 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

$$\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$$

  $\Rightarrow$ Eigenvectors $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_N]$ are the frequency basis (atoms)

Graph Fourier Transform

- Adjacency $\mathbf{A}$, Laplacian $\mathbf{L}$, or, generically graph shift $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$

  $\Rightarrow$ $S_{ij} = 0$ for $i \neq j$ and $(i, j) \notin \mathcal{E}$ (captures local structure in 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

$$\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$$

  $\Rightarrow$ Eigenvectors $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_N]$ are the frequency basis (atoms)

- Additional structure

  $\Rightarrow$ If $\mathbf{S}$ is normal, then $\mathbf{V}^{-1} = \mathbf{V}^H$ and $\tilde{x}_k = \mathbf{v}_k^H\mathbf{x} = <\mathbf{v}_k, \mathbf{x}>$

  $\Rightarrow$ Parseval holds, $\|\mathbf{x}\|^2 = \|\tilde{\mathbf{x}}\|^2$

- **GFT** $\Rightarrow$ Projection on eigenvector space of graph shift operator $\mathbf{S}$

Frequency modes of the Laplacian

- Total variation of signal $\mathbf{x}$ with respect to $\mathbf{L}$

$$\text{TV}(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{i,j=1, j>i}^{N} A_{ij}(x_i - x_j)^2$$

$\Rightarrow$ Smoothness measure on the graph G (Dirichlet energy)

- For Laplacian eigenvectors $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_N]$ $\Rightarrow$ $\text{TV}(\mathbf{v}_k) = \lambda_k$

$\Rightarrow$ Can view $0 = \lambda_1 < \cdots \leq \lambda_N$ as frequencies

Frequency modes of the Laplacian

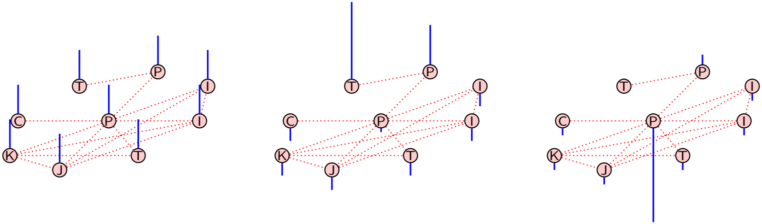
- Total variation of signal $\mathbf{x}$ with respect to $\mathbf{L}$

$$\text{TV}(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{i,j=1,j>i}^{N} A_{ij}(x_i - x_j)^2$$

$\Rightarrow$ Smoothness measure on the graph G (Dirichlet energy)

- For Laplacian eigenvectors $\mathbf{V} = [\mathbf{v}_1, \cdots, \mathbf{v}_N]$ $\Rightarrow$ $\text{TV}(\mathbf{v}_k) = \lambda_k$
  $\Rightarrow$ Can view $0 = \lambda_1 < \cdots \leq \lambda_N$ as frequencies

- Ex: gene network, $N = 10$, $k = 1$, $k = 2$, $k = 9$

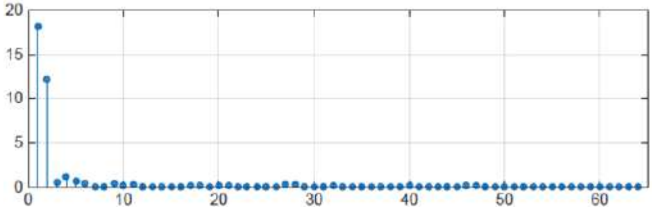


Is this a reasonable transform?

- Particularized to cyclic graphs $\Rightarrow$ GFT $\equiv$ Fourier transform
- Also for covariance graphs $\Rightarrow$ GFT $\equiv$ PCA transform
- But really, this is an empirical question.

Is this a reasonable transform?

- Particularized to cyclic graphs $\Rightarrow$ GFT $\equiv$ Fourier transform
- Also for covariance graphs $\Rightarrow$ GFT $\equiv$ PCA transform
- But really, this is an empirical question. GFT of disaggregated GDP



- Spectral domain representation characterized by a few coefficients
  - $\Rightarrow$ Notion of bandlimitedness: $\mathbf{x} = \sum_{k=1}^{K} \tilde{x}_k \mathbf{v}_k$
  - $\Rightarrow$ Sampling, compression, filtering, pattern recognition

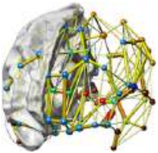
Graph frequency analysis of brain signals

- GFT of brain signals during a visual-motor learning task [Huang et al'16]
  - ⇒ Decomposed into low, medium and high frequency components

- Brain: Complex system where regularity coexists with disorder [Sporns'11]
  - ⇒ Signal energy mostly in the low and high frequencies
  - ⇒ In brain regions akin to the visual and sensorimotor cortices

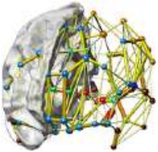
What is this tutorial about?

- Learning graphs from nodal observations
- Key in neuroscience
  - ⇒ Functional network from fMRI signals



What is this tutorial about?

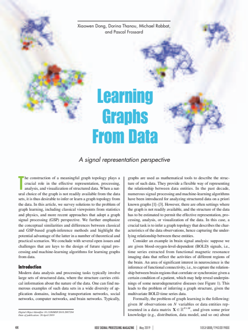
- Learning graphs from nodal observations
- Key in neuroscience
  - ⇒ Functional network from fMRI signals

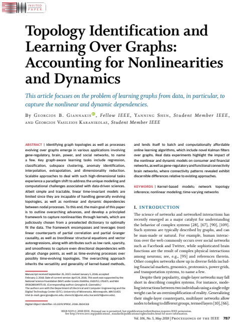


- Most GSP works: how known graph **S** affects signals and filters
- Here, reverse path: how to use GSP to infer the graph topology?
  - Gaussian graphical models [Egilmez et al'16], [Rabbat'17], ...
  - Smooth signals [Dong et al'15], [Kalofolias'16], [Sardellitti et al'17], ...
  - Graph filtering models [Shafipour et al'17], [Thanou et al'17], ...
  - Stationary signals [Pasdeloup et al'15], [Segarra et al'16], ...
  - Directed graphs [Mei-Moura'15], [Shen et al'16], ...

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
  - Forthcoming issue on Network Topology Inference (Jan. 2020)

Network topology inference

Graph signal processing: Motivation and fundamentals

Statistical methods for network topology inference

Learning graphs from observations of smooth signals

Identifying the structure of network diffusion processes

Discussion

Network topology inference problems

- **Q:** If G (or a portion thereof) is unobserved, can we infer it from data?
- Formulate as a statistical inference task, i.e. given
  - Signal measurements x_i at some or all vertices $i \in \mathcal{V}$
  - Indicators y_{ij} of edge status for some vertex pairs $\{i, j\} \in \mathcal{V}_{obs}^{(2)}$
  - A collection $\mathcal{G}$ of candidate graphs G

  Goal: infer the topology of the network graph $G(\mathcal{V}, \mathcal{E})$
- Bring to bear existing statistical concepts and tools

  $\Rightarrow$ Study identifiability, consistency, robustness, complexity

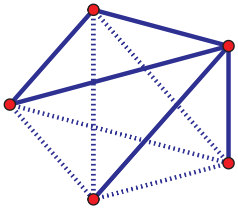
Network topology inference problems

- **Q:** If G (or a portion thereof) is unobserved, can we infer it from data?
- Formulate as a statistical inference task, i.e. given
  - Signal measurements x_i at some or all vertices $i \in \mathcal{V}$
  - Indicators y_{ij} of edge status for some vertex pairs $\{i, j\} \in \mathcal{V}^{(2)}_{obs}$
  - A collection $\mathcal{G}$ of candidate graphs G

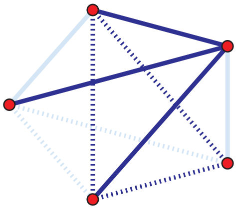
  Goal: infer the topology of the network graph $G(\mathcal{V}, \mathcal{E})$
- Bring to bear existing statistical concepts and tools

  $\Rightarrow$ Study identifiability, consistency, robustness, complexity
- Three canonical network topology inference problems [Kolaczyk'09]
  (i) Link prediction
  (ii) Association network inference $\leftarrow$ Focus of this tutorial
  (iii) Tomographic network topology inference

Link prediction



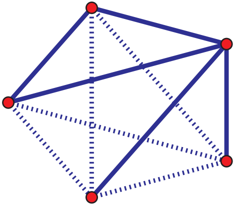
Original graph



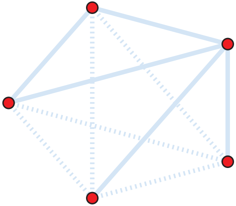
Link prediction

- Suppose we observe the graph signal $\mathbf{x} = [x_1, \ldots, x_N]^\top$; and
- Edge status is only observed for some subset of pairs $\mathcal{V}^{(2)}_{obs} \subset \mathcal{V}^{(2)}$
- **Goal:** predict edge status for all other pairs, i.e., $\mathcal{V}^{(2)}_{miss} = \mathcal{V}^{(2)} \setminus \mathcal{V}^{(2)}_{obs}$

Association network inference



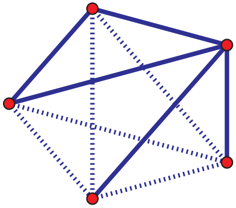
Original graph



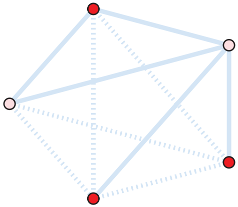
Association network inference

- Suppose we only observe the graph signal $\mathbf{x} = [x_1, \ldots, x_N]^\top$; and
- Assume (i, j) defined by nontrivial 'level of association' among x_i, x_j
- **Goal:** predict edge status for all vertex pairs $\mathcal{V}^{(2)}$

Tomographic network topology inference



Original graph



Tomographic inference

- Suppose we only observe x_i for vertices $i \subset \mathcal{V}$ in the 'perimeter' of G
- **Goal:** predict edge and vertex status in the 'interior' of G

Association network inference

- Given a collection of N elements represented as vertices $v \in \mathcal{V}$
  - Graph signal $\mathbf{x} = [x_1, \ldots, x_N]^\top \in \mathbb{R}^N$ of observed vertex attributes
- User-defined similarity $\texttt{sim}(i, j) = f(x_i, x_j)$ specifies edges $(i, j) \in \mathcal{E}$
  - **Q:** What if `sim` values themselves (i.e., edge status) not observable?

Association network inference

- Given a collection of N elements represented as vertices $v \in \mathcal{V}$
  - Graph signal $\mathbf{x} = [x_1, \ldots, x_N]^\top \in \mathbb{R}^N$ of observed vertex attributes
- User-defined similarity $\texttt{sim}(i,j) = f(x_i, x_j)$ specifies edges $(i,j) \in \mathcal{E}$
  - **Q:** What if `sim` values themselves (i.e., edge status) not observable?

Association network inference

Infer non-trivial `sim` values from i.i.d. observations $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^P$

- Various choices to be made, hence multiple possible approaches
  - Choice of `sim`: correlation, partial correlation, mutual information
  - Choice of inference: hypothesis testing, regression, ad hoc
  - Choice of parameters: testing thresholds, tuning regularization

Correlation networks

- Pearson product-moment correlation as $\texttt{sim}$ between vertex pairs

$$\texttt{sim}(i,j) := \rho_{ij} = \frac{\text{cov}[x_i, x_j]}{\sqrt{\text{var}\,[x_i]\,\text{var}\,[x_j]}}, \quad i,j \in \mathcal{V}$$

Correlation networks

- Pearson product-moment correlation as $\mathtt{sim}$ between vertex pairs

$$\mathtt{sim}(i,j) := \rho_{ij} = \frac{\text{cov}[x_i, x_j]}{\sqrt{\text{var}[x_i]\,\text{var}[x_j]}}, \;\; i,j \in \mathcal{V}$$

- **Def:** the correlation network graph $G(\mathcal{V}, \mathcal{E})$ has edge set

$$\mathcal{E} = \left\{(i,j) \in \mathcal{V}^{(2)} : \rho_{ij} \neq 0\right\}$$

  - Association network inference $\Leftrightarrow$ Inference of non-zero correlations

- Inference of $\mathcal{E}$ typically approached as a testing problem

$$H_0 : \rho_{ij} = 0 \quad \text{versus} \quad H_1 : \rho_{ij} \neq 0$$

Test statistics

- Common choice of test statistic are empirical correlations

$$\hat{\rho}_{ij} = \frac{\hat{\sigma}_{ij}}{\sqrt{\hat{\sigma}_{ii}\hat{\sigma}_{jj}}}, \quad \text{where} \quad \hat{\mathbf{\Sigma}} = [\hat{\sigma}_{ij}] = \frac{1}{P-1}\sum_{p=1}^{P} \mathbf{x}_p \mathbf{x}_p^T$$

- Convenient alternative statistic is Fisher's transformation

$$\hat{z}_{ij} = \frac{1}{2}\log\left(\frac{1+\hat{\rho}_{ij}}{1-\hat{\rho}_{ij}}\right), \quad i,j \in \mathcal{V}$$

$\Rightarrow$ Under H_0, $\hat{z}_{ij} \sim \mathcal{N}(0, \frac{1}{P-3})$ $\Rightarrow$ Simple to assess significance

Test statistics

- Common choice of test statistic are empirical correlations

$$\hat{\rho}_{ij} = \frac{\hat{\sigma}_{ij}}{\sqrt{\hat{\sigma}_{ii}\hat{\sigma}_{jj}}}, \quad \text{where} \;\; \hat{\mathbf{\Sigma}} = [\hat{\sigma}_{ij}] = \frac{1}{P-1}\sum_{p=1}^{P} \mathbf{x}_p \mathbf{x}_p^T$$

- Convenient alternative statistic is Fisher's transformation

$$\hat{z}_{ij} = \frac{1}{2}\log\left(\frac{1+\hat{\rho}_{ij}}{1-\hat{\rho}_{ij}}\right), \;\; i,j \in \mathcal{V}$$

$\Rightarrow$ Under H_0, $\hat{z}_{ij} \sim \mathcal{N}(0, \frac{1}{P-3})$ $\Rightarrow$ Simple to assess significance

- Reject H_0 at significance level α, i.e., assign edge (i,j) if $|\hat{z}_{ij}| > \frac{z_{\alpha/2}}{\sqrt{P-3}}$

Error rate control: $\mathrm{P}_{H_0}\,(\text{false edge}) = \mathrm{P}_{H_0}\left(|\hat{z}_{ij}| > \frac{z_{\alpha/2}}{\sqrt{P-3}}\right) = \alpha$

Networks and multiple testing

- Interesting testing challenges emerge with large-scale networks
  - $\Rightarrow$ Suppose we test all $\binom{N}{2}$ vertex pairs, each at level α
- Even if the true G is the empty graph, i.e., $\mathcal{E} = \emptyset$
  - $\Rightarrow$ We expect to declare $\binom{N}{2}\alpha$ spurious edges just by chance!
  - $\Rightarrow$ For a large graph, this number can be considerable

Networks and multiple testing

- Interesting testing challenges emerge with large-scale networks
  - $\Rightarrow$ Suppose we test all $\binom{N}{2}$ vertex pairs, each at level α
- Even if the true G is the empty graph, i.e., $\mathcal{E} = \emptyset$
  - $\Rightarrow$ We expect to declare $\binom{N}{2}\alpha$ spurious edges just by chance!
  - $\Rightarrow$ For a large graph, this number can be considerable
- Ex: For G of order $N = 100$ and individual tests at level $\alpha = 0.05$
  - $\Rightarrow$ Expected number of spurious edges is $4950 \times 0.05 \approx 250$
- This predicament known as the multiple testing problem in statistics

Correction for multiple testing

- **Idea:** Control errors at the level of collection of tests, not individually
- False discovery rate (FDR) control, i.e., for given level γ ensure

$$\text{FDR} = \mathbb{E}\left[\frac{R_{false}}{R} \,\middle|\, R > 0\right] \text{P}\left[R > 0\right] \leq \gamma$$

  - R is the total number of edges detected; and
  - R_{false} is the number of false edges detected

Correction for multiple testing

- **Idea:** Control errors at the level of collection of tests, not individually
- False discovery rate (FDR) control, i.e., for given level γ ensure

$$\text{FDR} = \mathbb{E}\left[\frac{R_{false}}{R} \,\middle|\, R > 0\right] \text{P}\left[R > 0\right] \leq \gamma$$

  - R is the total number of edges detected; and
  - R_{false} is the number of false edges detected

- Method of FDR control at level γ [Benjamini-Hochberg'94]

  Step 1: Sort p-values for all $\bar{N} := \binom{N}{2}$ tests, yields $p_{(1)} \leq \ldots \leq p_{(\bar{N})}$

  Step 2: Reject H_0, i.e., declare all those edges for which

$$p_{(k)} \leq \left(\frac{k}{\bar{N}}\right)\gamma$$

Partial correlations

- Use correlations carefully: 'correlation does not imply causation'
  - Vertices $i, j \in \mathcal{V}$ may have high ρ_{ij} because they influence each other
- But ρ_{ij} could be high if both i, j influenced by a third vertex $k \in \mathcal{V}$

  $\Rightarrow$ Correlation networks may declare edges due to confounders

Partial correlations

- Use correlations carefully: 'correlation does not imply causation'
  - Vertices $i, j \in \mathcal{V}$ may have high ρ_{ij} because they influence each other
- But ρ_{ij} could be high if both i, j influenced by a third vertex $k \in \mathcal{V}$

  $\Rightarrow$ Correlation networks may declare edges due to confounders
- Partial correlations better capture direct influence among vertices
  - For $i, j \in \mathcal{V}$ consider latent vertices $S_m = \{k_1, \ldots, k_m\} \subset \mathcal{V} \setminus \{i, j\}$
- Partial correlation of x_i and x_j, adjusting for $\mathbf{x}_{S_m} = [x_{k_1}, \ldots, x_{k_m}]^T$ is

$$\rho_{ij|S_m} = \frac{\text{cov}[x_i, x_j \,|\, \mathbf{x}_{S_m}]}{\sqrt{\text{var}\left[x_i \,|\, \mathbf{x}_{S_m}\right] \text{var}\left[x_j \,|\, \mathbf{x}_{S_m}\right]}}, \quad i, j \in \mathcal{V}$$

- **Q:** How do we obtain these partial correlations?

Computing partial correlations

- Given $\mathbf{x}_{S_m} = [x_{k_1}, \ldots, x_{k_m}]^T$, the partial correlation of x_i and x_j is

$$\rho_{ij|S_m} = \frac{\mathrm{cov}[x_i, x_j \mid \mathbf{x}_{S_m}]}{\sqrt{\mathrm{var}\left[x_i \mid \mathbf{x}_{S_m}\right] \mathrm{var}\left[x_j \mid \mathbf{x}_{S_m}\right]}} = \frac{\sigma_{ij|S_m}}{\sqrt{\sigma_{ii|S_m}\sigma_{jj|S_m}}}$$

Computing partial correlations

- Given $\mathbf{x}_{S_m} = [x_{k_1}, \ldots, x_{k_m}]^T$, the partial correlation of x_i and x_j is

$$\rho_{ij|S_m} = \frac{\text{cov}[x_i, x_j \mid \mathbf{x}_{S_m}]}{\sqrt{\text{var}\left[x_i \mid \mathbf{x}_{S_m}\right] \text{var}\left[x_j \mid \mathbf{x}_{S_m}\right]}} = \frac{\sigma_{ij|S_m}}{\sqrt{\sigma_{ii|S_m}\sigma_{jj|S_m}}}$$

- Here $\sigma_{ii|S_m}, \sigma_{jj|S_m}$ and $\sigma_{ij|S_m}$ are diagonal and off-diagonal elements of

$$\boldsymbol{\Sigma}_{11|2} := \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21} \in \mathbb{R}^{2\times 2}$$

- Matrices $\boldsymbol{\Sigma}_{11}$, $\boldsymbol{\Sigma}_{22}$ and $\boldsymbol{\Sigma}_{21} = \boldsymbol{\Sigma}_{12}^{\top}$ are blocks of the covariance matrix

$$\text{cov}\begin{bmatrix} \mathbf{w}_1 \\ \mathbf{w}_2 \end{bmatrix} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}, \quad \text{where } \mathbf{w}_1 := [x_i, x_j]^T \text{ and } \mathbf{w}_2 := \mathbf{x}_{S_m}$$

Partial correlation networks

- Various ways to use partial correlations to define edges in G

  Ex: x_i, x_j correlated regardless of what m vertices we condition upon

$$\mathcal{E} = \left\{ (i,j) \in \mathcal{V}^{(2)} : \rho_{ij|S_m} \neq 0, \text{ for all } S_m \in \mathcal{V}^{(m)}_{\setminus\{i,j\}} \right\}$$

Partial correlation networks

- Various ways to use partial correlations to define edges in G

  Ex: x_i, x_j correlated regardless of what m vertices we condition upon

$$\mathcal{E} = \left\{ (i,j) \in \mathcal{V}^{(2)} : \rho_{ij|S_m} \neq 0, \text{ for all } S_m \in \mathcal{V}^{(m)}_{\setminus\{i,j\}} \right\}$$

- Inference of potential edge (i,j) as a testing problem

$$H_0 : \rho_{ij|S_m} = 0 \text{ for some } S_m \in \mathcal{V}^{(m)}_{\setminus\{i,j\}}$$

$$H_1 : \rho_{ij|S_m} \neq 0 \text{ for all } S_m \in \mathcal{V}^{(m)}_{\setminus\{i,j\}}$$

- Again, given measurements $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^{P}$ need to:
  - Select a test statistic
  - Construct an appropriate null distribution
  - Adjust for multiple testing

Case study: Inferring gene-regulatory interactions

- Genes are segments of DNA encoding information about cell functions
- Such information used in the expression of genes

  $\Rightarrow$ Creation of biochemical products, i.e., RNA or proteins

Case study: Inferring gene-regulatory interactions

- Genes are segments of DNA encoding information about cell functions
- Such information used in the expression of genes
  - $\Rightarrow$ Creation of biochemical products, i.e., RNA or proteins
- Regulation of a gene refers to the control of its expression

  Ex: regulation exerted during transcription, copy of DNA to RNA
  - $\Rightarrow$ Controlling genes are transcription factors (TFs)
  - $\Rightarrow$ Controlled genes are termed targets
  - $\Rightarrow$ Regulation type: activation or repression

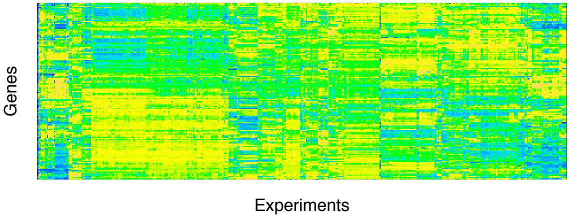
Case study: Inferring gene-regulatory interactions

- Genes are segments of DNA encoding information about cell functions
- Such information used in the expression of genes
  - $\Rightarrow$ Creation of biochemical products, i.e., RNA or proteins
- Regulation of a gene refers to the control of its expression

  Ex: regulation exerted during transcription, copy of DNA to RNA
  - $\Rightarrow$ Controlling genes are transcription factors (TFs)
  - $\Rightarrow$ Controlled genes are termed targets
  - $\Rightarrow$ Regulation type: activation or repression
- Regulatory interactions among genes basic to the workings of organisms
  - $\Rightarrow$ Inference of interactions $\rightarrow$ Finding TF/target gene pairs
- Such relational information summarized in gene-regulatory networks

Regulatory interactions among E. coli genes

- Use microarray data and correlation methods to infer TF/target pairs



- Dataset: relative log expression RNA levels, for genes in E. coli
  - 4,345 genes measured under 445 different experimental conditions
- Ground truth: 153 TFs, and TF/target pairs from database RegulonDB

Methods to infer TF/target gene pairs

- Three correlation based methods to infer TF/target gene pairs
  - $\Rightarrow$ Interactions declared if suitable p-values fall below a threshold

Method 1: Pearson correlation between TF and potential target gene

Method 2: Partial correlation, controlling for shared effects of one ($m = 1$) other TF, across all 152 other TFs

Method 3: Full partial correlation, simultaneously controlling for shared effects of all ($m = 152$) other TFs

Methods to infer TF/target gene pairs

- Three correlation based methods to infer TF/target gene pairs

  $\Rightarrow$ Interactions declared if suitable p-values fall below a threshold

  Method 1: Pearson correlation between TF and potential target gene

  Method 2: Partial correlation, controlling for shared effects of one ($m = 1$) other TF, across all 152 other TFs

  Method 3: Full partial correlation, simultaneously controlling for shared effects of all ($m = 152$) other TFs

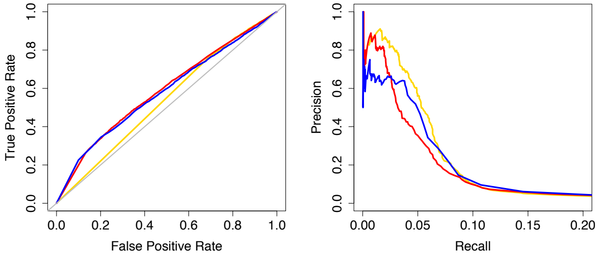
- In all cases applied Fisher transformation to obtain z-scores

  $\Rightarrow$ Asymptotic Gaussian distributions for p-values, with $P = 445$

- Compared inferred graphs to ground-truth network from RegulonDB

Performance comparisons

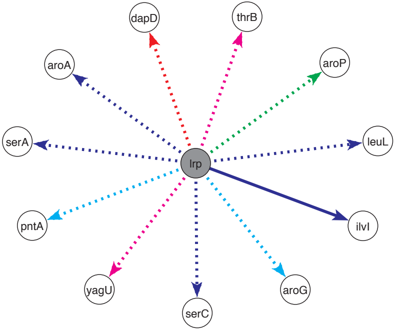
- ROC and Precision/Recall curves for Methods 1, 2, and 3
  - ⇒ Precision: fraction of predicted links that are true
  - ⇒ Recall: fraction of true links that are correctly predicted



- Method 1 performs worst, but none is stellar
  - ⇒ Correlation not strong indicator of regulation in this data
- All methods share a region of high precision, but a very small recall
  - ⇒ Limitations in number/diversity of profiles [Faith et al'07]

Predicting new TF/target gene pairs

- In biology, often interest is in predicting new interactions



- 11 interactions found for TF *lrp*, 10 experimentally confirmed (dotted)
  - ⇒ 5 interacting target genes were new (magenta, red, cyan)
  - ⇒ 4 present in RegulonDB (magenta, cyan), but not as *lrp* targets

Undirected Gaussian graphical models

- Suppose variables $\{x_i\}_{i \in \mathcal{V}}$ have multivariate Gaussian distribution
  - $\Rightarrow$ Consider $\rho_{ij|\mathcal{V}\setminus\{i,j\}}$ conditioning on all other vertices ($m = N - 2$)

Theorem

Under the Gaussian assumption, vertices $i, j \in \mathcal{V}$ have partial correlation

$$\rho_{ij|\mathcal{V}\setminus\{i,j\}} = 0$$

if and only if x_i and x_j are conditionally independent given $\{x_k\}_{k \in \mathcal{V}\setminus\{i,j\}}$

Undirected Gaussian graphical models

- Suppose variables $\{x_i\}_{i\in\mathcal{V}}$ have multivariate Gaussian distribution
  - $\Rightarrow$ Consider $\rho_{ij|\mathcal{V}\setminus\{i,j\}}$ conditioning on all other vertices ($m = N - 2$)

Theorem

Under the Gaussian assumption, vertices $i, j \in \mathcal{V}$ have partial correlation

$$\rho_{ij|\mathcal{V}\setminus\{i,j\}} = 0$$

if and only if x_i and x_j are conditionally independent given $\{x_k\}_{k\in\mathcal{V}\setminus\{i,j\}}$

- **Def:** the conditional independence graph $G(\mathcal{V},\mathcal{E})$ has edge set

$$\mathcal{E} = \left\{(i,j) \in \mathcal{V}^{(2)} : \rho_{ij|\mathcal{V}\setminus\{i,j\}} \neq 0\right\}$$

  - $\Rightarrow$ A special and popular case of partial correlation networks

- Also known as Gaussian Markov random field (GMRF)

Covariance selection

- Let $\mathbf{\Sigma}$ be the covariance matrix of $\mathbf{x} = [x_1, \ldots, x_N]^T$
  Def: the precision matrix is $\mathbf{\Theta} := \mathbf{\Sigma}^{-1}$ with entries θ_{ij}

- Key result: For GMRFs, the partial correlations can be expressed as

$$\rho_{ij|V\setminus\{i,j\}} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}}$$

$\Rightarrow$ Non-zero entries in $\mathbf{\Theta}$ $\Leftrightarrow$ Edges in the graph G

Covariance selection

- Let $\mathbf{\Sigma}$ be the covariance matrix of $\mathbf{x} = [x_1, \ldots, x_N]^T$
  Def: the precision matrix is $\mathbf{\Theta} := \mathbf{\Sigma}^{-1}$ with entries θ_{ij}

- Key result: For GMRFs, the partial correlations can be expressed as

$$\rho_{ij|V\setminus\{i,j\}} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}}$$

$\Rightarrow$ Non-zero entries in $\mathbf{\Theta} \Leftrightarrow$ Edges in the graph G

- Inferring G from $\mathcal{X}$ known as covariance selection [Dempster'74]
  $\Rightarrow$ Classical methods are 'network-agnostic,' and effectively test

$$H_0 : \rho_{ij|\mathcal{V}\setminus\{i,j\}} = 0 \quad \text{versus} \quad H_1 : \rho_{ij|\mathcal{V}\setminus\{i,j\}} \neq 0$$

- Often not scalable, and $P \ll N$ so estimation of $\hat{\mathbf{\Sigma}}$ challenging

Graphical Lasso

- Sparsity-regularized maximum-likelihood estimator of $\boldsymbol{\Theta}$ [Yuan-Lin'07]

$$\hat{\boldsymbol{\Theta}} \in \arg\max_{\boldsymbol{\Theta} \succeq \mathbf{0}} \left\{ \log\det \boldsymbol{\Theta} - \operatorname{trace}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta}) - \lambda \|\boldsymbol{\Theta}\|_1 \right\}$$

$\Rightarrow$ Effective when $P \ll N$, encourages interpretable models

$\Rightarrow$ Scalable solvers using coordinate-descent [Friedman et al'08]

Graphical Lasso

- Sparsity-regularized maximum-likelihood estimator of $\boldsymbol{\Theta}$ [Yuan-Lin'07]

$$\hat{\boldsymbol{\Theta}} \in \arg\max_{\boldsymbol{\Theta}\succeq \mathbf{0}} \left\{ \log\det\boldsymbol{\Theta} - \text{trace}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta}) - \lambda\|\boldsymbol{\Theta}\|_1 \right\}$$

$\Rightarrow$ Effective when $P \ll N$, encourages interpretable models

$\Rightarrow$ Scalable solvers using coordinate-descent [Friedman et al'08]

- Performance guarantee: Graphical lasso with $\lambda = 2\sqrt{\frac{\log N}{P}}$ satisfies

$$\|\hat{\boldsymbol{\Theta}} - \boldsymbol{\Theta}_0\|_2 \leq \sqrt{\frac{d_{\max}^2 \log N}{P}} \quad \text{w.h.p.}$$

$\Rightarrow$ Ground-truth $\boldsymbol{\Theta}_0$, maximum nodal degree $d_{\max}$

- Support consistency for $P = \Omega(d_{\max}^2 \log N)$ [Ravikumar et al'11]

GMRFs with Laplacian constraints

- Graphical model selection with Laplacian constraints $\mathbf{\Theta} = \mathbf{L}$
  - Off-diagonal entries $\theta_{ij} = L_{ij} = -A_{ij} \leq 0 \Rightarrow$ Attractive GMRF
  - Laplacian is singular ($\mathbf{L1} = \mathbf{0}$) $\Rightarrow$ Improper GMRF
- Estimate a proper GMRF via diagonal loading [Lake-Tenembaum'07]

$$\max_{\mathbf{\Theta} \succeq \mathbf{0}, \gamma \geq 0} \left\{ \log \det \mathbf{\Theta} - \text{trace}(\hat{\mathbf{\Sigma}}\mathbf{\Theta}) - \lambda \|\mathbf{\Theta}\|_1 \right\}$$

$$\text{s. to } \mathbf{\Theta} = \mathbf{L} + \gamma \mathbf{I}$$

$$\mathbf{L1} = \mathbf{0},\ L_{ij} \leq 0,\ i \neq j$$

$\Rightarrow$ Interpret γ^{-1} as variance of Gaussian isotropic fluctuations

- Favors graphs over which the signals are smooth (more later)

$$\text{trace}(\hat{\mathbf{\Sigma}}\mathbf{L}) \propto \sum_{p=1}^{P} \mathbf{x}_p^T \mathbf{L} \mathbf{x}_p = \sum_{p=1}^{P} \text{TV}(\mathbf{x}_p)$$

Covariance selection meets linear regression

- **Idea:** separately estimate neighborhoods $\mathcal{N}_i := \{j : (i,j) \in \mathcal{E}\}$, $i \in \mathcal{V}$
- Conditional mean of x_i given $\mathbf{x}_{\setminus i} := [x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N]^T$ is

$$\mathbb{E}\left[x_i \,\middle|\, \mathbf{x}_{\setminus i}\right] = \mathbf{x}_{\setminus i}^T \boldsymbol{\beta}^{(i)}$$

Covariance selection meets linear regression

- **Idea:** separately estimate neighborhoods $\mathcal{N}_i := \{j : (i,j) \in \mathcal{E}\}$, $i \in \mathcal{V}$
- Conditional mean of x_i given $\mathbf{x}_{\setminus i} := [x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N]^T$ is

$$\mathbb{E}\left[x_i \,\middle|\, \mathbf{x}_{\setminus i}\right] = \mathbf{x}_{\setminus i}^T \boldsymbol{\beta}^{(i)}$$

- Entries of $\boldsymbol{\beta}^{(i)}$ expressible in terms of those in $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$, namely

$$\beta_j^{(i)} = -\frac{\theta_{ij}}{\theta_{ii}}$$

$\Rightarrow$ Non-zero $\beta_j^{(i)} \Leftrightarrow$ Non-zero θ_{ij} in $\boldsymbol{\Theta} \Leftrightarrow$ Edge (i,j) in G

$\Rightarrow$ In other words, $\text{supp}(\boldsymbol{\beta}^{(i)}) := \{j : \beta_j^{(i)} \neq 0\} \equiv \mathcal{N}_i$

Covariance selection meets linear regression

- **Idea:** separately estimate neighborhoods $\mathcal{N}_i := \{j : (i,j) \in \mathcal{E}\}$, $i \in \mathcal{V}$
- Conditional mean of x_i given $\mathbf{x}_{\setminus i} := [x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_N]^T$ is

$$\mathbb{E}\left[x_i \,\middle|\, \mathbf{x}_{\setminus i}\right] = \mathbf{x}_{\setminus i}^T \boldsymbol{\beta}^{(i)}$$

- Entries of $\boldsymbol{\beta}^{(i)}$ expressible in terms of those in $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$, namely

$$\beta_j^{(i)} = -\frac{\theta_{ij}}{\theta_{ii}}$$

$\Rightarrow$ Non-zero $\beta_j^{(i)} \Leftrightarrow$ Non-zero θ_{ij} in $\boldsymbol{\Theta} \Leftrightarrow$ Edge (i,j) in G

$\Rightarrow$ In other words, $\text{supp}(\boldsymbol{\beta}^{(i)}) := \{j : \beta_j^{(i)} \neq 0\} \equiv \mathcal{N}_i$

- Suggests inference of G via least-squares (LS) regression, since

$$\boldsymbol{\beta}^{(i)} = \arg\min_{\boldsymbol{\beta}} \mathbb{E}\left[(x_i - \mathbf{x}_{\setminus i}^T \boldsymbol{\beta})^2\right], \quad i \in \mathcal{V}$$

Neighborhood-based sparse regression

- Cycle over vertices $i \in \mathcal{V}$ and estimate $\hat{\mathcal{N}}_i = \text{supp}(\hat{\boldsymbol{\beta}}^{(i)})$, where

$$\hat{\boldsymbol{\beta}}^{(i)} \in \arg \min_{\boldsymbol{\beta} \in \mathbb{R}^{N-1}} \left\{ \sum_{p=1}^{P} (x_{pi} - \mathbf{x}_{p,\setminus i}^T \boldsymbol{\beta})^2 + \lambda \|\boldsymbol{\beta}\|_1 \right\}$$

  $\Rightarrow$ Separable lasso problems per vertex

- No guarantee that $\hat{\beta}_j^{(i)} \neq 0$ implies $\hat{\beta}_i^{(j)} \neq 0$ and vice versa

  $\Rightarrow$ Combine information in $\hat{\mathcal{N}}_i$ and $\hat{\mathcal{N}}_j$ to enforce symmetry

  $\Rightarrow$ OR rule: $(i,j) \in \mathcal{E}$ if $\beta_j^{(i)} \neq 0$ or $\beta_i^{(j)} \neq 0$. Likewise, AND rule

- Support consistency for either rule [Meinshausen-Bühlmann'06]
  - Suitable choice of λ, sparsity of $\boldsymbol{\Theta}_0$, and sample complexity $P \ll N$

Conceptual roadmap for GMRF model selection

Testing partial correlations

For each $(i,j) \in \mathcal{V} \times \mathcal{V}$, test the hypothesis

$$H_0 : \rho_{ij|\mathcal{V}\backslash ij} = 0 \quad \text{versus} \quad H_1 : \rho_{ij|\mathcal{V}\backslash ij} \neq 0$$

$$\rho_{ij|\mathcal{V}\backslash ij} = -\frac{\theta_{ij}}{\sqrt{\theta_{ii}\theta_{jj}}} \rightarrow \rho_{ij|\mathcal{V}\backslash ij} \neq 0 \Leftrightarrow \theta_{ij} \neq 0$$

Covariance selection

Infer non-zero entries $\theta_{ij} \neq 0$ of the precision matrix

$$\boldsymbol{\Theta} := \boldsymbol{\Sigma}^{-1}$$

$$\beta_j^{(i)} = -\frac{\theta_{ij}}{\theta_{ii}} \rightarrow \beta_j^{(i)} \neq 0 \Leftrightarrow \theta_{ij} \neq 0$$

Neighborhood-based regression

For each $i \in \mathcal{V}$, infer non-zero regression coefficients $\beta_j^{(i)} \neq 0$ in

$$\boldsymbol{\beta}^{(i)} = \arg\min_{\boldsymbol{\beta}} \mathbb{E}\left[(x_i - \mathbf{x}_{\backslash i}^T \boldsymbol{\beta})^2\right]$$

Comparative summary

- Parallelizable neighborhood-based regression (NBR)
  - $\Rightarrow$ Conditional likelihood per vertex $i \in \mathcal{V}$, disregards $\boldsymbol{\Theta} \succeq \mathbf{0}$
  - $\Rightarrow$ Tends to be computationally faster
- Graphical Lasso minimizes a (regularized) global likelihood

$$\mathcal{L}(\boldsymbol{\Theta}; \mathcal{X}) = \log\det\boldsymbol{\Theta} - \text{trace}(\hat{\boldsymbol{\Sigma}}\boldsymbol{\Theta})$$

  - $\Rightarrow$ Tends to be (statistically) more efficient
- NBR method tractable even for discrete or mixed graphical models
  - $\Rightarrow$ Ising-model selection for $\mathbf{x} \in \{-1, +1\}^N$ [Ravikumar'10]

Learning graphs from smooth signals

Problem formulation

Rationale

- Seek graphs on which data admit certain regularities
  - Nearest-neighbor prediction (a.k.a. graph smoothing)
  - Semi-supervised learning
  - Efficient information-processing transforms

- Many real-world graph signals are smooth
  - Graphs based on similarities among vertex attributes
  - Network formation driven by homophily, proximity in latent space

Problem formulation

Rationale

- Seek graphs on which data admit certain regularities
  - Nearest-neighbor prediction (a.k.a. graph smoothing)
  - Semi-supervised learning
  - Efficient information-processing transforms
- Many real-world graph signals are smooth
  - Graphs based on similarities among vertex attributes
  - Network formation driven by homophily, proximity in latent space

Problem statement

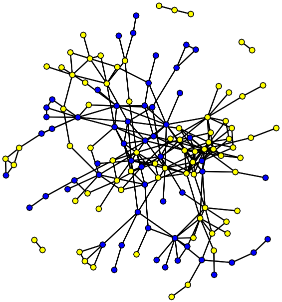
Given observations $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^{P}$, identify a graph G such that signals in $\mathcal{X}$ are smooth on G.

- Criterion: Dirichlet energy on the graph $\mathcal{G}$ with Laplacian $\mathbf{L}$

$$\text{TV}(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x}$$

Example: Predicting protein function

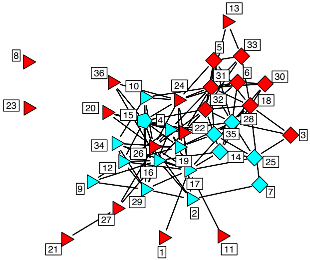
- Baker's yeast data, formally known as *Saccharomyces cerevisiae*
  - Graph: 134 vertices (proteins) and 241 edges (protein interactions)



- Signal: functional annotation intracellular signaling cascade (ICSC)
  - Signal transduction, how cells react to the environment
  - $x_i = 1$ if protein i annotated ICSC (yellow), $x_i = 0$ otherwise (blue)

Example: Predicting law practice

- Working relationships among lawyers [Lazega'01]
  - Graph: 36 partners, edges indicate partners worked together



- Signal: various node-level attributes $\mathbf{x} = \{x_i\}_{i \in \mathcal{V}}$ including

  $\Rightarrow$ Type of practice, i.e., litigation (red) and corporate (cyan)

- Suspect lawyers collaborate more with peers in same legal practice

  $\Rightarrow$ Knowledge of collaboration useful in predicting type of practice

Laplacian-based factor analysis model

- Consider an unknown graph G with Laplacian $\mathbf{L} = \mathbf{V}\boldsymbol{\Lambda}\mathbf{V}^T$

  $\Rightarrow$ Adopt GFT basis $\mathbf{V}$ as signal representation matrix

- Factor-analysis model for the observed graph signal [Dong et al'16]

$$\mathbf{x} = \mathbf{V}\boldsymbol{\chi} + \boldsymbol{\epsilon}$$

  $\Rightarrow$ Latent variables $\boldsymbol{\chi} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Lambda}^\dagger)$ ($\approx$ GFT coefficients)

  $\Rightarrow$ Isotropic error term $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$

- Smoothness: prior encourages low-pass bandlimited $\mathbf{x}$

  $\Rightarrow$ Small eigenvalues of $\mathbf{L}$ (low freq.) $\rightarrow$ High-power factor loadings

Inference as denoising via graph kernel regression

- Maximum a posteriori (MAP) estimator of the latent variables χ

$$\hat{\chi}_{\text{MAP}} = \arg\min_{\chi} \left\{ \|\mathbf{x} - \mathbf{V}\chi\|^2 + \alpha \chi^T \mathbf{\Lambda} \chi \right\}$$

$\Rightarrow$ Parameterized by the unknown $\mathbf{V}$ and $\mathbf{\Lambda}$

- Define predictor $\mathbf{y} := \mathbf{V}\chi$, regularizer expressible as

$$\chi^T \mathbf{\Lambda} \chi = \mathbf{y}^T \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T \mathbf{y} = \mathbf{y}^T \mathbf{L} \mathbf{y} = \text{TV}(\mathbf{y})$$

$\Rightarrow$ Laplacian-based TV denoiser of $\mathbf{x}$, smoothness prior on $\mathbf{y}$

$\Rightarrow$ Kernel-ridge regression with unknown $\mathbf{K} := \mathbf{L}^{\dagger}$ (graph filter)

Inference as denoising via graph kernel regression

- Maximum a posteriori (MAP) estimator of the latent variables χ

$$\hat{\chi}_{\text{MAP}} = \arg\min_{\chi} \left\{ \|\mathbf{x} - \mathbf{V}\chi\|^2 + \alpha \chi^T \mathbf{\Lambda} \chi \right\}$$

   $\Rightarrow$ Parameterized by the unknown $\mathbf{V}$ and $\mathbf{\Lambda}$

- Define predictor $\mathbf{y} := \mathbf{V}\chi$, regularizer expressible as

$$\chi^T \mathbf{\Lambda} \chi = \mathbf{y}^T \mathbf{V} \mathbf{\Lambda} \mathbf{V}^T \mathbf{y} = \mathbf{y}^T \mathbf{L} \mathbf{y} = \text{TV}(\mathbf{y})$$

   $\Rightarrow$ Laplacian-based TV denoiser of $\mathbf{x}$, smoothness prior on $\mathbf{y}$

   $\Rightarrow$ Kernel-ridge regression with unknown $\mathbf{K} := \mathbf{L}^{\dagger}$ (graph filter)

- **Idea:** jointly search for $\mathbf{L}$ and denoised representation $\mathbf{y} = \mathbf{V}\chi$

$$\min_{\mathbf{L}, \mathbf{y}} \left\{ \|\mathbf{x} - \mathbf{y}\|^2 + \alpha \mathbf{y}^T \mathbf{L} \mathbf{y} \right\}$$

Formulation and algorithm

- Given signals $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^{P}$ in $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times P}$, solve

$$\min_{\mathbf{L},\mathbf{Y}} \left\{ \|\mathbf{X} - \mathbf{Y}\|_F^2 + \alpha \text{trace}\left(\mathbf{Y}^T \mathbf{L} \mathbf{Y}\right) + \frac{\beta}{2} \|\mathbf{L}\|_F^2 \right\}$$
$$\text{s. to} \quad \text{trace}(\mathbf{L}) = N,\ \mathbf{L}\mathbf{1} = \mathbf{0},\ L_{ij} = L_{ji} \leq 0,\ i \neq j$$

$\Rightarrow$ Objective function: Fidelity + smoothness + edge sparsity

$\Rightarrow$ Not jointly convex in $\mathbf{L}$ and $\mathbf{Y}$, but bi-convex

Formulation and algorithm

- Given signals $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^{P}$ in $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times P}$, solve

$$\min_{\mathbf{L},\mathbf{Y}} \left\{ \|\mathbf{X} - \mathbf{Y}\|_F^2 + \alpha \text{trace}\left(\mathbf{Y}^T \mathbf{L} \mathbf{Y}\right) + \frac{\beta}{2} \|\mathbf{L}\|_F^2 \right\}$$
$$\text{s. to} \quad \text{trace}(\mathbf{L}) = N,\ \mathbf{L1} = \mathbf{0},\ L_{ij} = L_{ji} \leq 0,\ i \neq j$$

  $\Rightarrow$ Objective function: Fidelity + smoothness + edge sparsity

  $\Rightarrow$ Not jointly convex in $\mathbf{L}$ and $\mathbf{Y}$, but bi-convex

- Algorithmic approach: alternating minimization (AM), $O(N^3)$ cost

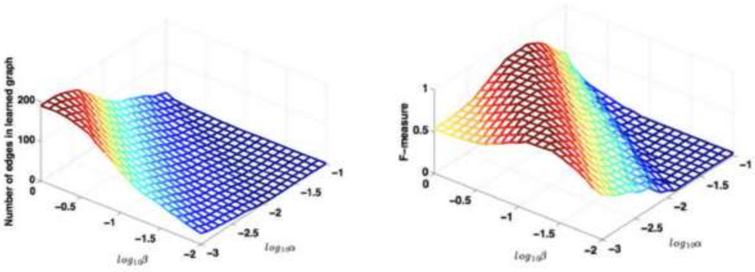
  (S1) Fixed $\mathbf{Y}$: solve for $\mathbf{L}$ via interior-point method, ADMM (more soon)

  (S2) Fixed $\mathbf{L}$: low-pass, graph filter-based smoother of the signals in $\mathbf{X}$

$$\mathbf{Y} = (\mathbf{I} + \alpha \mathbf{L})^{-1} \mathbf{X}$$

Impact of regularizers on sparsity and accuracy

- Generate multiple signals on a synthetic Erdős-Rényi graph
- Recover the graph for different values of α and β

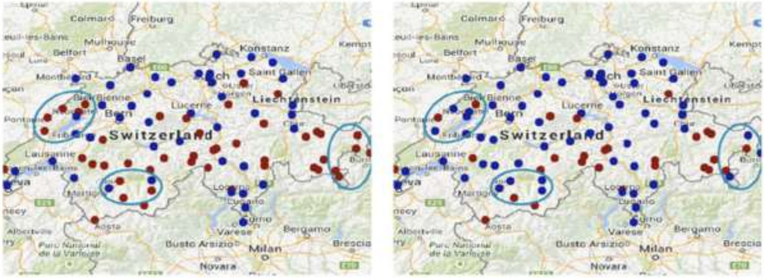


- More edges promoted by increasing β and decreasing α
- In the low noise regime, the ratio β/α determines behavior

Learning a temperature graph in Switzerland

- 89 stations measuring monthly temperature averages (1981-2010)
- Learn a graph on which the temperatures vary smoothly
- Geographical distance not a good idea $\Rightarrow$ different altitudes
- Recover altitude partition from spectral clustering

  $\Rightarrow$ Red (high stations) and blue (low stations) clusters
- k-means applied directly to the temperatures (right) fails



Signal smoothness meets edge sparsity

- Recall $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times P}$, let $\bar{\mathbf{x}}_i^T \in \mathbb{R}^{1 \times P}$ denote its i-th row
  $\Rightarrow$ Euclidean distance matrix $\mathbf{Z} \in \mathbb{R}_+^{N \times N}$, where $Z_{ij} := \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|^2$

- Neat trick: link between smoothness and sparsity [Kalofolias'16]

$$\sum_{p=1}^{P} \mathrm{TV}(\mathbf{x}_p) = \mathrm{trace}(\mathbf{X}^T \mathbf{L} \mathbf{X}) = \frac{1}{2} \|\mathbf{A} \circ \mathbf{Z}\|_1$$

  $\Rightarrow$ Sparse $\mathcal{E}$ when data come from a smooth manifold
  $\Rightarrow$ Favor candidate edges (i, j) associated with small Z_{ij}

- Shows that edge sparsity on top of smoothness is redundant

Signal smoothness meets edge sparsity

- Recall $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times P}$, let $\bar{\mathbf{x}}_i^T \in \mathbb{R}^{1 \times P}$ denote its i-th row
  - $\Rightarrow$ Euclidean distance matrix $\mathbf{Z} \in \mathbb{R}_+^{N \times N}$, where $Z_{ij} := \|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|^2$
- Neat trick: link between smoothness and sparsity [Kalofolias'16]

$$\sum_{p=1}^{P} \text{TV}(\mathbf{x}_p) = \text{trace}(\mathbf{X}^T \mathbf{L} \mathbf{X}) = \frac{1}{2} \|\mathbf{A} \circ \mathbf{Z}\|_1$$

  - $\Rightarrow$ Sparse $\mathcal{E}$ when data come from a smooth manifold
  - $\Rightarrow$ Favor candidate edges (i, j) associated with small Z_{ij}
- Shows that edge sparsity on top of smoothness is redundant
- Parameterize graph learning problems in terms of $\mathbf{A}$ (instead of $\mathbf{L}$)
  - $\Rightarrow$ Advantageous since constraints on $\mathbf{A}$ are decoupled

Scalable topology identification framework

- General purpose model for learning graphs [Kalofolias'16]

$$\min_{\mathbf{A}} \left\{ \|\mathbf{A} \circ \mathbf{Z}\|_1 - \alpha \mathbf{1}^T \log(\mathbf{A}\mathbf{1}) + \frac{\beta}{2}\|\mathbf{A}\|_F^2 \right\}$$
$$\text{s. to} \quad \text{diag}(\mathbf{A}) = \mathbf{0},\ A_{ij} = A_{ji} \geq 0,\ i \neq j$$

$\Rightarrow$ Logarithmic barrier forces positive degrees

$\Rightarrow$ Penalize large edge-weights to control sparsity

Scalable topology identification framework

- General purpose model for learning graphs [Kalofolias'16]

$$\min_{\mathbf{A}} \left\{ \|\mathbf{A} \circ \mathbf{Z}\|_1 - \alpha \mathbf{1}^T \log(\mathbf{A}\mathbf{1}) + \frac{\beta}{2}\|\mathbf{A}\|_F^2 \right\}$$
$$\text{s. to} \quad \text{diag}(\mathbf{A}) = \mathbf{0},\ A_{ij} = A_{ji} \geq 0,\ i \neq j$$

$\Rightarrow$ Logarithmic barrier forces positive degrees

$\Rightarrow$ Penalize large edge-weights to control sparsity

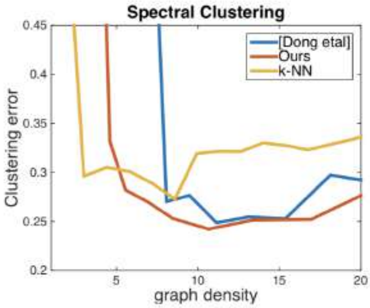
- Primal-dual solver amenable to parallelization, $O(N^2)$ cost

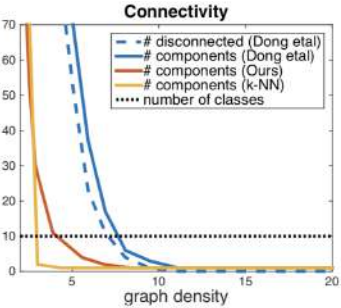
- Laplacian-based factor analysis encore. Tackle **(S1)** as

$$\min_{\mathbf{A}} \left\{ \|\mathbf{A} \circ \mathbf{Z}\|_1 - \log(\mathbb{I}\left\{\|\mathbf{A}\|_1 = N\right\}) + \frac{\beta}{2}\left(\|\mathbf{A}\mathbf{1}\|^2 + \|\mathbf{A}\|_F^2\right) \right\}$$
$$\text{s. to} \quad \text{diag}(\mathbf{A}) = \mathbf{0},\ A_{ij} = A_{ji} \geq 0,\ i \neq j$$

Learning the graph of USPS digits

- 1001 images of the 10 digits, but highly imbalanced ($2.6i^2$)
- 10 classes via graph recovery plus spectral clustering
- Compare two methods based on smoothness and k-NN graph





- Performance more robust to graph density
- Likely attributable to non-singleton nodes

Graph learning via edge subset selection

- **Idea:** parameterize the unknown topology via an edge indicator vector

Graph learning via edge subset selection

- **Idea:** parameterize the unknown topology via an edge indicator vector
- Complete graph on N nodes, having $M := \binom{N}{2}$ edges
  $\Rightarrow$ Incidence matrix $\mathbf{B} := [\mathbf{b}_1, \ldots, \mathbf{b}_M] \in \mathbb{R}^{N \times M}$
- Laplacian of a candidate graph $G(\mathcal{V}, \mathcal{E})$ [Chepuri et al'17]

$$\mathbf{L}(\boldsymbol{\omega}) = \sum_{m=1}^{M} \omega_m \mathbf{b}_m \mathbf{b}_m^T$$

$\Rightarrow$ Binary edge indicator vector $\boldsymbol{\omega} := [\omega_1, \ldots, \omega_M]^T \in \{0, 1\}^M$
$\Rightarrow$ Offers an explicit handle on the number of edges $\|\boldsymbol{\omega}\|_0 = |\mathcal{E}|$

Graph learning via edge subset selection

- **Idea:** parameterize the unknown topology via an edge indicator vector
- Complete graph on N nodes, having $M := \binom{N}{2}$ edges
  $\Rightarrow$ Incidence matrix $\mathbf{B} := [\mathbf{b}_1, \ldots, \mathbf{b}_M] \in \mathbb{R}^{N \times M}$
- Laplacian of a candidate graph $G(\mathcal{V}, \mathcal{E})$ [Chepuri et al'17]

$$\mathbf{L}(\boldsymbol{\omega}) = \sum_{m=1}^{M} \omega_m \mathbf{b}_m \mathbf{b}_m^T$$

$\Rightarrow$ Binary edge indicator vector $\boldsymbol{\omega} := [\omega_1, \ldots, \omega_M]^T \in \{0,1\}^M$

$\Rightarrow$ Offers an explicit handle on the number of edges $\|\boldsymbol{\omega}\|_0 = |\mathcal{E}|$

Problem: Given observations $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^{P}$, learn an unweighted graph $G(\mathcal{V}, \mathcal{E})$ such that signals in $\mathcal{X}$ are smooth on G and $|\mathcal{E}| = K$.

Cardinality-constrained Boolean optimization

- Natural formulation is to solve the non-convex problem

$$\min_{\boldsymbol{\omega}\in\{0,1\}^M} \text{trace}(\mathbf{X}^T\mathbf{L}(\boldsymbol{\omega})\mathbf{X}), \quad \text{s. to } \|\boldsymbol{\omega}\|_0 = K$$

- Solution obtained through a simple rank-ordering procedure
  - Compute edge scores $c_m := \text{trace}(\mathbf{X}^T(\mathbf{b}_m\mathbf{b}_m^T)\mathbf{X})$
  - Set $\omega_m = 1$ for those K edges having the smallest scores

Cardinality-constrained Boolean optimization

- Natural formulation is to solve the non-convex problem

$$\min_{\boldsymbol{\omega}\in\{0,1\}^M} \text{trace}(\mathbf{X}^T\mathbf{L}(\omega)\mathbf{X}), \quad \text{s. to } \|\boldsymbol{\omega}\|_0 = K$$

- Solution obtained through a simple rank-ordering procedure
  - Compute edge scores $c_m := \text{trace}(\mathbf{X}^T(\mathbf{b}_m\mathbf{b}_m^T)\mathbf{X})$
  - Set $\omega_m = 1$ for those K edges having the smallest scores

- More pragmatic AWGN setting where $\mathbf{x}_p = \mathbf{y}_p + \boldsymbol{\epsilon}_p$, $p = 1, \ldots, P$

$$\min_{\mathbf{Y},\boldsymbol{\omega}\in\{0,1\}^M} \left\{\|\mathbf{X}-\mathbf{Y}\|_F^2 + \alpha\text{trace}(\mathbf{Y}^T\mathbf{L}(\omega)\mathbf{Y})\right\}, \quad \text{s. to } \|\boldsymbol{\omega}\|_0 = K$$

$\Rightarrow$ Tackle via AM or semidefinite relaxation (SDR)

Comparative summary

- Noteworthy features of the edge subset selection approach
  - ✓ Direct control on edge sparsity
  - ✓ Simple algorithm in the noise-free case
  - ✓ Devoid of Laplacian feasibility constraints
  - ✗ Does not guarantee connectivity of G
  - ✗ No room for optimizing edge weights

Comparative summary

- Noteworthy features of the edge subset selection approach
  - ✓ Direct control on edge sparsity
  - ✓ Simple algorithm in the noise-free case
  - ✓ Devoid of Laplacian feasibility constraints
  - ✗ Does not guarantee connectivity of G
  - ✗ No room for optimizing edge weights

- Scalable framework in [Kalofolias'16] also quite flexible

$$\min_{\mathbf{A}} \{\|\mathbf{A} \circ \mathbf{Z}\|_1 + g(\mathbf{A})\}$$
$$\text{s. to} \quad \text{diag}(\mathbf{A}) = \mathbf{0},\ A_{ij} = A_{ji} \geq 0,\ i \neq j$$

$\Rightarrow$ Subsumes the factor-analysis model [Dong et al'16]

$\Rightarrow$ Recovers Gaussian kernel weights $A_{ij} := \exp\left(-\frac{\|\bar{\mathbf{x}}_i - \bar{\mathbf{x}}_j\|^2}{\sigma^2}\right)$ for

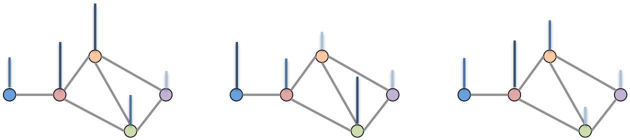
$$g(\mathbf{A}) = \sigma^2 \sum_{i,j} A_{ij}(\log(A_{ij}) - 1)$$

Learning graphs from diffused signals

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

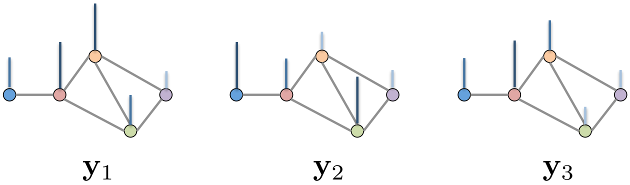


$\mathbf{y}_1$ $\mathbf{y}_2$ $\mathbf{y}_3$

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



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 $(L \leq N)$

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

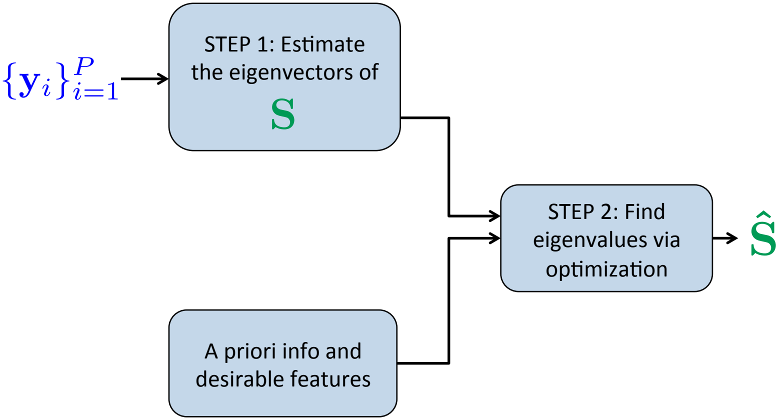
$\Rightarrow$ Graph filter $\mathbf{H}$ is shift invariant [Sandryhaila-Moura'13]

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

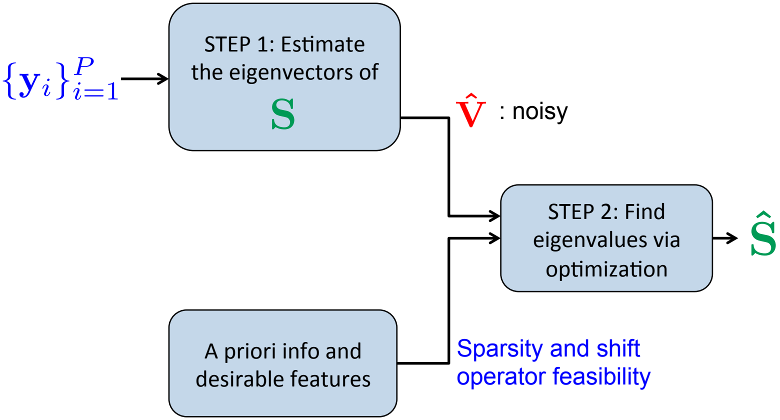
- **Goal**: estimate undirected network $\mathbf{S}$ from signal realizations $\{\mathbf{y}_i\}_{i=1}^{P}$

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

Blueprint of our solution



Blueprint of our solution



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 $\mathbf{S}$

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

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

Correlated input signals

- **Q:** What if the signal $\mathbf{x}$ is colored?

  $\Rightarrow$ Matrices $\mathbf{S}$ and $\mathbf{C}_y$ no longer simultaneously diagonalizable since

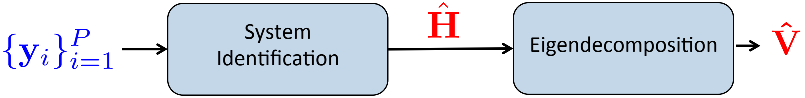
$$\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}$$

Correlated input signals

- **Q:** What if the signal $\mathbf{x}$ is colored?
  - $\Rightarrow$ Matrices $\mathbf{S}$ and $\mathbf{C}_y$ no longer simultaneously diagonalizable since

$$\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}$$

- Key: still $\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$ diagonalized by the eigenvectors $\mathbf{V}$ of $\mathbf{S}$
  - $\Rightarrow$ Infer $\mathbf{V}$ by estimating the unknown diffusion (graph) filter $\mathbf{H}$
  - $\Rightarrow$ Step 1 boils down to system identification + eigendecomposition



- Henceforth assume $\mathbf{C}_x$ is non-singluar and known

System ID as matrix quadratic equation

- **Q:** What are the solutions of the quadratic equation $\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}$?

Proposition: Define $\mathbf{C}_{xyx} := \mathbf{C}_x^{1/2}\mathbf{C}_y\mathbf{C}_x^{1/2}$, with eigenvectors $\mathbf{V}_{xyx}$. Then all admissible symmetric graph filters $\mathbf{H}$ are of the form

$$\mathbf{H} = \mathbf{C}_x^{-1/2}\mathbf{C}_{xyx}^{1/2}\mathbf{V}_{xyx}\text{diag}(\mathbf{b})\mathbf{V}_{xyx}^T\mathbf{C}_x^{-1/2},$$

where $\mathbf{b} \in \{-1, 1\}^N$ is a binary (signed) vector.

System ID as matrix quadratic equation

- **Q:** What are the solutions of the quadratic equation $\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}$?

Proposition: Define $\mathbf{C}_{xyx} := \mathbf{C}_x^{1/2}\mathbf{C}_y\mathbf{C}_x^{1/2}$, with eigenvectors $\mathbf{V}_{xyx}$. Then all admissible symmetric graph filters $\mathbf{H}$ are of the form

$$\mathbf{H} = \mathbf{C}_x^{-1/2}\mathbf{C}_{xyx}^{1/2}\mathbf{V}_{xyx}\text{diag}(\mathbf{b})\mathbf{V}_{xyx}^T\mathbf{C}_x^{-1/2},$$

where $\mathbf{b} \in \{-1, 1\}^N$ is a binary (signed) vector.

- Even if we know $\mathbf{C}_y$ perfectly, $\mathbf{H}$ is not identifiable
  - $\Rightarrow$ Not surprising since we only have second-moment information
  - $\Rightarrow$ Unique solution $\mathbf{H} = \mathbf{C}_x^{-1/2}\mathbf{C}_{xyx}^{1/2}\mathbf{C}_x^{-1/2}$ for positive semidefinite $\mathbf{H}$
- Consider having access to multiple input distributions $\{\mathbf{C}_{x,m}\}_{m=1}^M$

Boolean quadratic program

- Define $\mathbf{A}_m := (\mathbf{C}_{x,m}^{-1/2}\mathbf{V}_{xyx,m}) \odot (\mathbf{C}_{x,m}^{-1/2}\mathbf{C}_{xyx,m}^{1/2}\mathbf{V}_{xyx,m})$ and form

$$\boldsymbol{\Psi} := \begin{bmatrix} \mathbf{A}_1 & -\mathbf{A}_2 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & -\mathbf{A}_3 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{M-1} & -\mathbf{A}_M \end{bmatrix}$$

- With $\mathbf{b}_m \in \{-1,1\}^N$ and $\mathbf{b} = [\mathbf{b}_1^T, \mathbf{b}_2^T, \ldots, \mathbf{b}_M^T]^T$, then $\boldsymbol{\Psi}\mathbf{b}^* = \mathbf{0}$

Boolean quadratic program

- Define $\mathbf{A}_m := (\mathbf{C}_{x,m}^{-1/2}\mathbf{V}_{xyx,m}) \odot (\mathbf{C}_{x,m}^{-1/2}\mathbf{C}_{xyx,m}^{1/2}\mathbf{V}_{xyx,m})$ and form

$$\boldsymbol{\Psi} := \begin{bmatrix} \mathbf{A}_1 & -\mathbf{A}_2 & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{A}_2 & -\mathbf{A}_3 & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{A}_{M-1} & -\mathbf{A}_M \end{bmatrix}$$

- With $\mathbf{b}_m \in \{-1,1\}^N$ and $\mathbf{b} = [\mathbf{b}_1^T, \mathbf{b}_2^T, \ldots, \mathbf{b}_M^T]^T$, then $\boldsymbol{\Psi}\mathbf{b}^* = \mathbf{0}$
- In practice only $\{\hat{\mathbf{C}}_{y,m}\}_{m=1}^M$ are available $\Rightarrow$ Estimate $\mathbf{b}^*$ as

$$\hat{\mathbf{b}}^* = \operatorname*{argmin}_{\mathbf{b}\in\{-1,1\}^{NM}} \mathbf{b}^T\hat{\boldsymbol{\Psi}}^T\hat{\boldsymbol{\Psi}}\mathbf{b}$$

- Solution $\hat{\mathbf{b}}^*$ of binary quadratic program (BQP) $\Rightarrow$ Filter estimate

$$\hat{\mathbf{H}} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{C}_{x,m}^{-1/2}\hat{\mathbf{C}}_{xyx,m}^{1/2}\hat{\mathbf{V}}_{xyx,m}\text{diag}(\hat{\mathbf{b}}_m^*)\hat{\mathbf{V}}_{xyx,m}^T\mathbf{C}_{x,m}^{-1/2}$$

Semidefinite relaxation

- System identification reduces to solving the NP-hard BQP

$$\hat{\mathbf{b}}^* = \underset{\mathbf{b}\in\{-1,1\}^{NM}}{\operatorname{argmin}} \; \mathbf{b}^T \hat{\boldsymbol{\Psi}}^T \hat{\boldsymbol{\Psi}} \mathbf{b}$$

- Define $\hat{\mathbf{W}} = \hat{\boldsymbol{\Psi}}^T \hat{\boldsymbol{\Psi}}$ and $\mathbf{B} = \mathbf{b}\mathbf{b}^T$, BQP equivalent to

$$\min_{\mathbf{B}\succeq\mathbf{0}} \operatorname{tr}(\hat{\mathbf{W}}\mathbf{B}) \quad \text{s. to} \;\; \operatorname{rank}(\mathbf{B}) = 1,\; B_{ii} = 1,\; i = 1, \ldots, NM$$

- Drop source of non-convexity $\Rightarrow$ Semidefinite relaxation (SDR)

$$\mathbf{B}^* = \underset{\mathbf{B}\succeq\mathbf{0}}{\operatorname{argmin}} \operatorname{tr}(\hat{\mathbf{W}}\mathbf{B}) \quad \text{s. to} \;\; B_{ii} = 1,\; i = 1, \ldots, NM$$

Performance guarantee

- For $l = 1, \ldots, L$, draw $\mathbf{z}_l \sim \mathcal{N}(\mathbf{0}, \mathbf{B}^*)$, round $\tilde{\mathbf{b}}_l = \text{sign}(\mathbf{z}_l)$, to obtain

$$l^* = \underset{l=1,\ldots,L}{\text{argmin}}\, \tilde{\mathbf{b}}_l^T \hat{\mathbf{W}} \tilde{\mathbf{b}}_l$$

Performance guarantee

- For $l = 1, \ldots, L$, draw $\mathbf{z}_l \sim \mathcal{N}(\mathbf{0}, \mathbf{B}^*)$, round $\tilde{\mathbf{b}}_l = \text{sign}(\mathbf{z}_l)$, to obtain

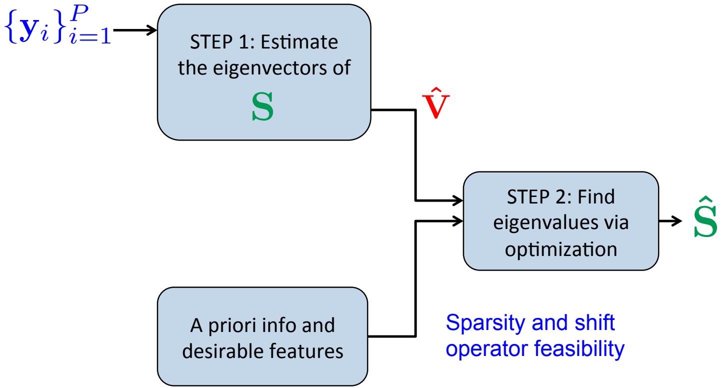
$$l^* = \underset{l=1,\ldots,L}{\text{argmin}}\, \tilde{\mathbf{b}}_l^T \hat{\mathbf{W}} \tilde{\mathbf{b}}_l$$

Theorem: Let $\hat{\mathbf{b}}^*$ be the BQP solution and $\tilde{\mathbf{b}}_{l^*}$ the SDR output. Then,

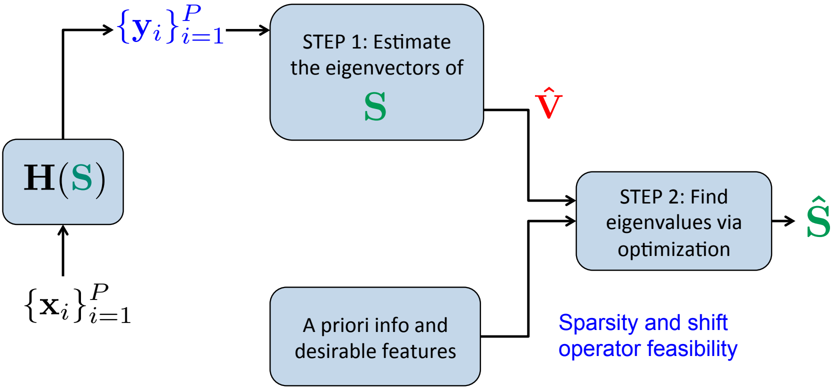
$$(\hat{\mathbf{b}}^*)^T \hat{\mathbf{W}} \hat{\mathbf{b}}^* \leq \mathbb{E}\left[(\tilde{\mathbf{b}}_{l^*})^T \hat{\mathbf{W}} \tilde{\mathbf{b}}_{l^*}\right] \leq \frac{2}{\pi}(\hat{\mathbf{b}}^*)^T \hat{\mathbf{W}} \hat{\mathbf{b}}^* + \gamma,$$

where $\gamma = \left(1 - \frac{2}{\pi}\right) \lambda_{\max}(\hat{\mathbf{W}}) NM$.

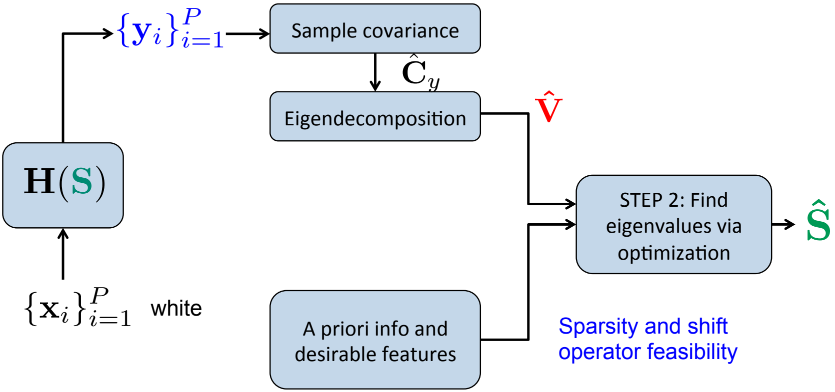
Summary of Step 1



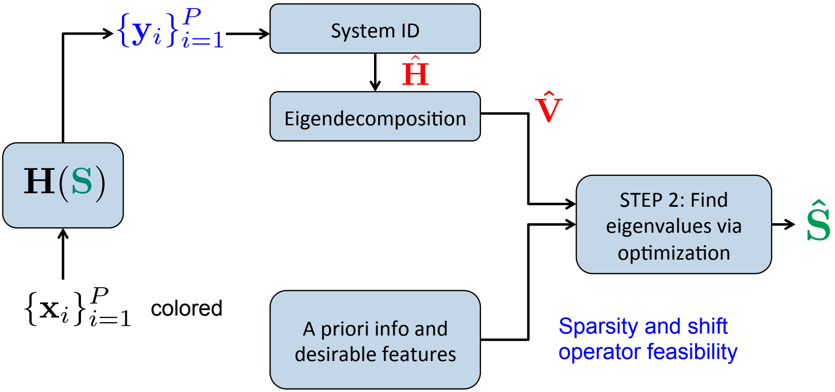
Summary of Step 1



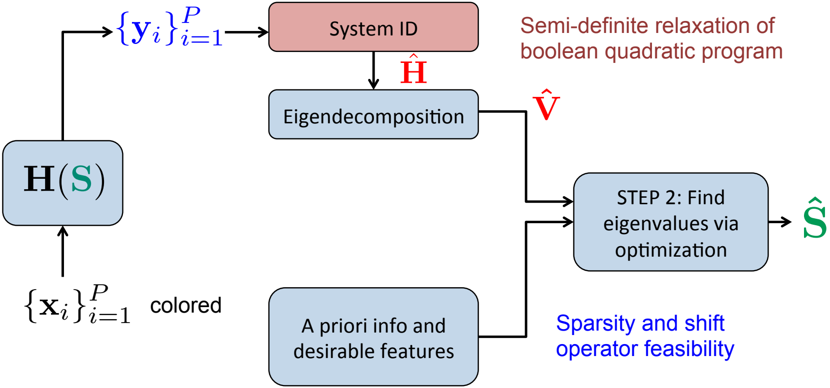
Summary of Step 1



Summary of Step 1



Summary of Step 1



Step 2: Obtaining the eigenvalues

- We can use extra knowledge/assumptions to choose one graph

  $\Rightarrow$ Of all graphs, select one that is optimal in some sense

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

Step 2: Obtaining the eigenvalues

- We can use extra knowledge/assumptions to choose one graph

  $\Rightarrow$ Of all graphs, select one that is optimal in some sense

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

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

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

  $\Rightarrow$ Can accommodate Laplacian matrices as well

Step 2: Obtaining the eigenvalues

- We can use extra knowledge/assumptions to choose one graph

  $\Rightarrow$ Of all graphs, select one that is optimal in some sense

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

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

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

  $\Rightarrow$ Can accommodate Laplacian matrices as well

- Problem is convex if we select a convex objective $f(\mathbf{S},\boldsymbol{\lambda})$

  Ex: Sparsity ($f(\mathbf{S}) = \|\mathbf{S}\|_1$), min. energy ($f(\mathbf{S}) = \|\mathbf{S}\|_F$), mixing ($f(\boldsymbol{\lambda}) = -\lambda_2$)

Sparse graph recovery

- Whenever the problem's feasibility set is non-trivial

  $\Rightarrow$ $f(\mathbf{S}, \boldsymbol{\lambda})$ determines the features of the recovered graph

Ex: Identify sparsest shift $\mathbf{S}_0^*$ that explains observed signal structure

$\Rightarrow$ Set the objective $f(\mathbf{S}, \boldsymbol{\lambda}) = \|\mathbf{S}\|_0 = |\text{supp}(\mathbf{S})|$

Sparse graph recovery

- Whenever the problem's feasibility set is non-trivial

  $\Rightarrow f(\mathbf{S}, \boldsymbol{\lambda})$ determines the features of the recovered graph

  Ex: Identify sparsest shift $\mathbf{S}_0^*$ that explains observed signal structure

  $\Rightarrow$ Set the objective $f(\mathbf{S}, \boldsymbol{\lambda}) = \|\mathbf{S}\|_0 = |\text{supp}(\mathbf{S})|$

- Non-convex problem, relax to ℓ_1-norm minimization, e.g., [Tropp'06]

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

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

Recovery guarantee for ℓ_1 relaxation

- $\mathcal{D}$ is the index set such that $\mathrm{vec}(\mathbf{S})_{\mathcal{D}} = \mathrm{diag}(\mathbf{S})$
- $\mathcal{K}$ indexes the support of $\mathbf{s}_0^* = \mathrm{vec}(\mathbf{S}_0^*)$
- Define $\mathbf{M} := \mathbf{V} \odot \mathbf{V}$, where $\odot$ is the Khatri-Rao product
  $\Rightarrow$ Form $\mathbf{R} := [(\mathbf{I} - \mathbf{M}\mathbf{M}^{\dagger})_{\mathcal{D}^c}, \ \mathbf{e}_1 \otimes \mathbf{1}_{N-1}]$

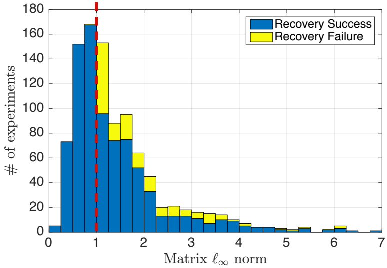
Theorem: $\mathbf{S}_1^* = \mathbf{S}_0^*$ if the two following conditions are satisfied
1) $\mathrm{rank}(\mathbf{R}_{\mathcal{K}}) = |\mathcal{K}|$; and
2) There exists a constant $\delta > 0$ such that

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

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

Sparse recovery guarantee

- Generate 1000 ER random graphs ($N = 20$, $p = 0.1$) such that
  - $\Rightarrow$ Feasible set is not a singleton
  - $\Rightarrow$ Cond. 1) in sparse recovery theorem is satisfied
- Noiseless case: ℓ_1 norm guarantees recovery as long as $\psi_{\mathbf{R}} < 1$



- Condition is sufficient but not necessary
  - $\Rightarrow$ Tightest possible bound on this matrix norm

Noisy spectral templates

- Step 1 actually yields $\hat{\mathbf{V}}$, a noisy version of the spectral templates

  $\Rightarrow$ With $d(\cdot,\cdot)$ denoting a (convex) distance between matrices

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

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

Noisy spectral templates

- Step 1 actually yields $\hat{\mathbf{V}}$, a noisy version of the spectral templates

  $\Rightarrow$ With $d(\cdot,\cdot)$ denoting a (convex) distance between matrices

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

- **Q:** How does the noise in $\hat{\mathbf{V}}$ affect the recovery?
- Stable recovery can be established $\Rightarrow$ depends on noise level

  $\Rightarrow$ Reformulate problem as $\min_{\mathbf{t}} \|\mathbf{t}\|_1$ s. to $\|\hat{\mathbf{R}}^T\mathbf{t} - \mathbf{b}\|_2 \le \epsilon$
- Conditions 1) and 2) but based on $\hat{\mathbf{R}}$, guaranteed $d(\mathbf{S}^*, \mathbf{S}_0^*) \le C\epsilon$

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

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

Incomplete spectral templates

- Partial access to $\mathbf{V}$ $\Rightarrow$ Only K known eigenvectors $\mathbf{V}_K = [v_1, \ldots, v_K]$

$$\min_{\{\mathbf{S}, \mathbf{S}_{\bar{K}}, \boldsymbol{\lambda}\}} \|\mathbf{S}\|_1 \text{ s. to } \mathbf{S} = \mathbf{S}_{\bar{K}} + \sum_{k=1}^{K} \lambda_k \mathbf{v}_k \mathbf{v}_k^T, \ \mathbf{S} \in \mathcal{S}, \ \mathbf{S}_{\bar{K}} \mathbf{V}_K = \mathbf{0}$$

- **Q:** How does the (partial) knowledge of $\mathbf{V}_K$ affect the recovery?

Incomplete spectral templates

- Partial access to $\mathbf{V}$ $\Rightarrow$ Only K known eigenvectors $\mathbf{V}_K = [v_1, \ldots, v_K]$

$$\min_{\{\mathbf{S}, \mathbf{S}_{\bar{K}}, \boldsymbol{\lambda}\}} \|\mathbf{S}\|_1 \text{ s. to } \mathbf{S} = \mathbf{S}_{\bar{K}} + \sum_{k=1}^{K} \lambda_k \mathbf{v}_k \mathbf{v}_k^T, \ \mathbf{S} \in \mathcal{S}, \ \mathbf{S}_{\bar{K}} \mathbf{V}_K = \mathbf{0}$$

- **Q:** How does the (partial) knowledge of $\mathbf{V}_K$ affect the recovery?

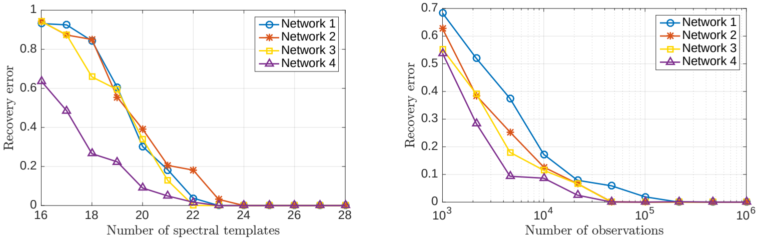
- Define $\mathbf{P} := [\mathbf{P}_1, \ \mathbf{P}_2]$ in terms of $\mathbf{V}_K$, and $\boldsymbol{\Upsilon} := [\mathbf{I}_{N^2}, \mathbf{0}_{N^2 \times N^2}]$
  $\Rightarrow$ Reformulate problem as $\min_{\mathbf{t}} \|\boldsymbol{\Upsilon}\mathbf{t}\|_1$ s.to $\mathbf{P}^T \mathbf{t} = \mathbf{b}$

Theorem: $\mathbf{S}^* = \mathbf{S}_0^*$ if the two following conditions are satisfied
1) $\text{rank}([\mathbf{P}_1{}_{\mathcal{K}}^T, \mathbf{P}_2{}^T]) = |\mathcal{K}| + N^2$; and
2) There exists a constant $\delta > 0$ such that

$$\eta_{\mathbf{P}} := \|\boldsymbol{\Upsilon}_{\mathcal{K}^c}(\delta^{-2}\mathbf{P}\mathbf{P}^T + \boldsymbol{\Upsilon}_{\mathcal{K}^c}^T \boldsymbol{\Upsilon}_{\mathcal{K}^c})^{-1} \boldsymbol{\Upsilon}_{\mathcal{K}}^T\|_\infty < 1$$

Social graphs from imperfect templates

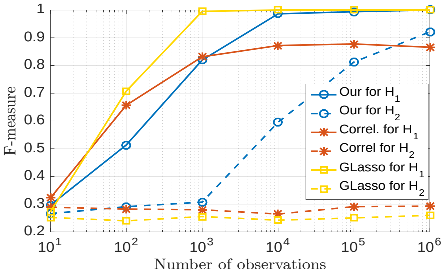
- Identification of multiple social networks with $N = 32$
  - $\Rightarrow$ Defined on the same node set of students from Ljubljana
  - $\Rightarrow$ Synthetic signals from diffusion processes in the graphs
- Recovery for incomplete (left) and noisy (right) spectral templates



- Error (left) decreases with increasing nr. of spectral templates
- Error (right) decreases with increasing number of observed signals

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 spectral templates work, but not as well

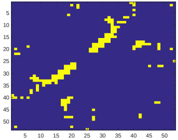
- For general diffusion filters $\mathbf{H}_2$ spectral templates still work fine

Inferring the structure of a protein

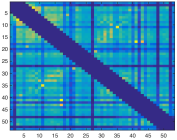
- Our method can be used to sparsify a given network
  - ⇒ Keep direct and important edges or relations
  - ⇒ Discard indirect relations that can be explained by direct ones
- Use eigenvectors $\hat{\mathbf{V}}$ of given network as noisy eigenvectors of $\mathbf{S}$

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

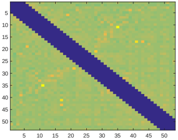
⇒ Use mutual information of amino-acid covariation as input



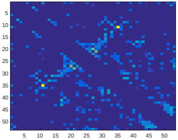
Ground truth



Mutual info.



Network deconv.



Our approach

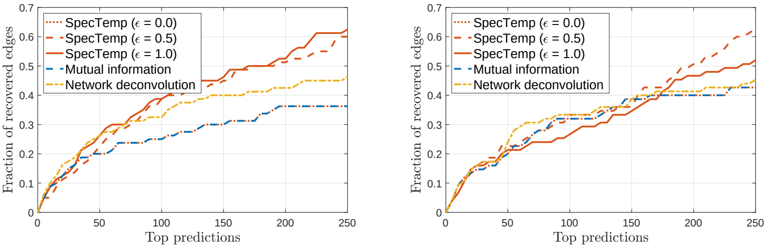
- Network deconvolution assumes a specific filter model [Feizi13]
  - ⇒ We achieve better performance by being agnostic to this

Sensitivity of recovered edges

- Sensitivity of the top edge predictions

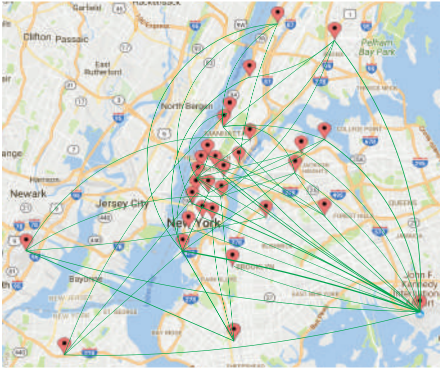
  $\Rightarrow$ Fraction of the real contact edges recovered

- For $\epsilon = 0$ we force $\mathbf{S}$ to be mutual information matrix $\mathbf{S}'$
- For larger values of ϵ, we get a better recovery



Unveiling urban mobility patterns

- Detect mobility patterns in New York City from Uber pickup data
  - Times and locations ($N = 30$) from January 1st to June 29th 2015
  - Pickups within 6-11am as input signal $\mathbf{x}$ and 3-8pm as output $\mathbf{y}$
  - $M = 2$ graph processes: weekday ($m = 1$) and weekend ($m = 2$) pickups



- Most edges between Manhattan and the other boroughs
- Few edges within Manhattan
  $\Rightarrow$ Uber mostly for commute
- Hubs at JFK, Newark and LaGuardia airports

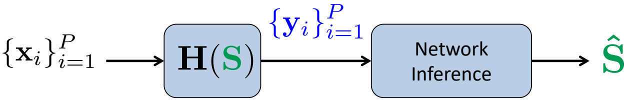
Summary

- 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

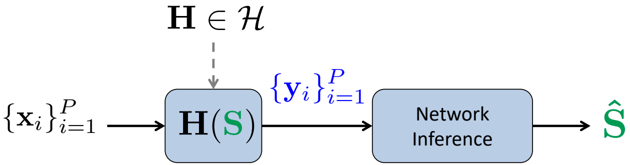
Summary

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

A rich framework for network inference

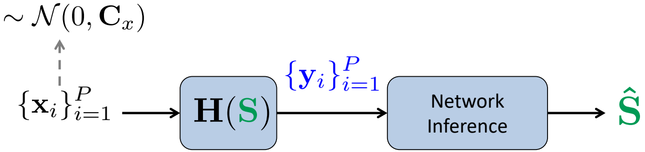


A rich framework for network inference



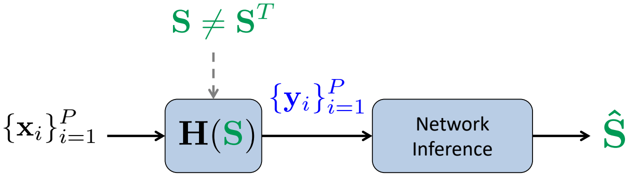
- Prior knowledge on the filter class [Segarra et al'17]

A rich framework for network inference



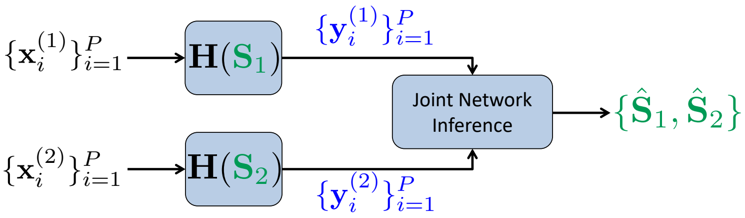
- Prior knowledge on the filter class [Segarra et al'17]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]

A rich framework for network inference



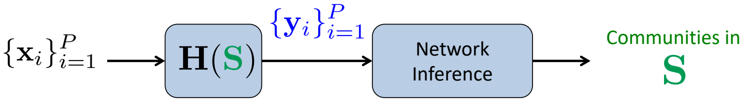
- Prior knowledge on the filter class [Segarra et al'17]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]

A rich framework for network inference



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

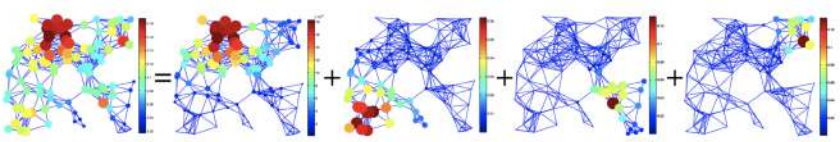
A rich framework for network inference



- Prior knowledge on the filter class [Segarra et al'17]
- 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 et al'18, '19]

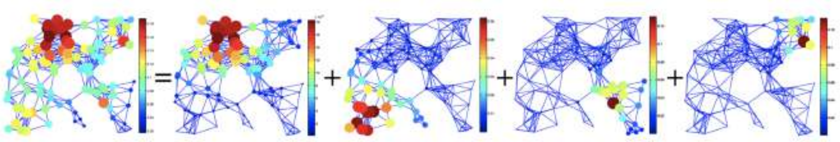
Learning heat diffusion graphs

- Superimposed heat diffusion processes on G [Thanou et al'17]



Learning heat diffusion graphs

- Superimposed heat diffusion processes on G [Thanou et al'17]



- Dictionary consisting of heat diffusion filters with different rates
  - $\Rightarrow$ Signals modeled as a linear combination of few (sparse) atoms
- Graph learning task as a regularized inverse problem
  - $\Rightarrow$ The graph (hence, the filters) is unknown
  - $\Rightarrow$ The sparse combination coefficients are unknown

Learning heat diffusion graphs: Formulation

- Heat rates $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_S]^T$ of the S filters $\mathbf{H}_s = e^{-\tau_s \mathbf{L}} = \sum_{l=0}^{\infty} \frac{(-\tau_s \mathbf{L})^l}{l!}$
- Given signals $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^P$ in $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times P}$, solve

$$\min_{\mathbf{L},\mathbf{R},\boldsymbol{\tau}} \left\{ \left\| \mathbf{X} - \mathbf{K}\mathbf{R} \right\|_F^2 + \alpha \sum_{p=1}^{P} \|\mathbf{r}_p\|_1 + \beta \|\mathbf{L}\|_F^2 \right\}$$

$$\text{s. to} \quad \mathbf{K} = \left[e^{-\tau_1 \mathbf{L}}, e^{-\tau_2 \mathbf{L}}, \ldots, e^{-\tau_S \mathbf{L}} \right]$$

$$\text{trace}(\mathbf{L}) = N, \quad \mathbf{L}\mathbf{1} = \mathbf{0}, \quad L_{ij} = L_{ji} \leq 0,\ i \neq j, \quad \tau_i \geq 0$$

$\Rightarrow$ $\mathbf{R} \in \mathbb{R}^{NS \times P}$ are sparse combination coefficients

$\Rightarrow$ Objective function: Fidelity + sparsity + regularizer

Learning heat diffusion graphs: Formulation

- Heat rates $\boldsymbol{\tau} = [\tau_1, \ldots, \tau_S]^T$ of the S filters $\mathbf{H}_s = e^{-\tau_s \mathbf{L}} = \sum_{l=0}^{\infty} \frac{(-\tau_s \mathbf{L})^l}{l!}$
- Given signals $\mathcal{X} := \{\mathbf{x}_p\}_{p=1}^P$ in $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_P] \in \mathbb{R}^{N \times P}$, solve

$$\begin{aligned} \min_{\mathbf{L},\mathbf{R},\boldsymbol{\tau}} & \left\{ \left\| \mathbf{X} - \mathbf{K}\mathbf{R} \right\|_F^2 + \alpha \sum_{p=1}^{P} \|\mathbf{r}_p\|_1 + \beta \|\mathbf{L}\|_F^2 \right\} \\ \text{s. to} \quad & \mathbf{K} = \left[e^{-\tau_1 \mathbf{L}}, e^{-\tau_2 \mathbf{L}}, \ldots, e^{-\tau_S \mathbf{L}} \right] \\ & \text{trace}(\mathbf{L}) = N, \quad \mathbf{L}\mathbf{1} = \mathbf{0}, \quad L_{ij} = L_{ji} \leq 0, \; i \neq j, \quad \tau_i \geq 0 \end{aligned}$$

$\Rightarrow$ $\mathbf{R} \in \mathbb{R}^{NS \times P}$ are sparse combination coefficients

$\Rightarrow$ Objective function: Fidelity + sparsity + regularizer

- Non-convex optimization, challenged by matrix exponentials
  - Proximal alternating linearized minimization (PALM)
  - Savings via low-degree polynomial approximation of $\mathbf{H}_s$

Comparative summary

- Main distinctive points of this model
  - $\Rightarrow$ Assumes a specific filter type: heat diffusion
  - $\Rightarrow$ Parametrized by a single scalar: the diffusion rate
  - $\Rightarrow$ Inputs to these filters are required to be sparse

Comparative summary

- Main distinctive points of this model
  - ⇒ Assumes a specific filter type: heat diffusion
  - ⇒ Parametrized by a single scalar: the diffusion rate
  - ⇒ Inputs to these filters are required to be sparse
- In comparison, for the spectral templates method
  - ⇒ Filters are arbitrary, not just diffusion
  - ⇒ Information about inputs is statistical instead of structural

Comparative summary

- Main distinctive points of this model
  - ⇒ Assumes a specific filter type: heat diffusion
  - ⇒ Parametrized by a single scalar: the diffusion rate
  - ⇒ Inputs to these filters are required to be sparse
- In comparison, for the spectral templates method
  - ⇒ Filters are arbitrary, not just diffusion
  - ⇒ Information about inputs is statistical instead of structural
- Inherent trade-off between model and data driven approaches

Discussion

Concluding remarks

- How to use the information in $\mathcal{X}$ to identify $G(\mathcal{V}, \mathcal{E})$
  - $\Rightarrow$ Focus on static and undirected graphs
  - $\Rightarrow$ GSP offers some novel insights and tools
- Emerging topic areas we did not cover
  - $\Rightarrow$ Directed graphs and causal structure identification
  - $\Rightarrow$ Dynamic networks and multi-layer graphs
  - $\Rightarrow$ Nonlinear models of interaction

Concluding remarks

- How to use the information in $\mathcal{X}$ to identify $G(\mathcal{V}, \mathcal{E})$
  - $\Rightarrow$ Focus on static and undirected graphs
  - $\Rightarrow$ GSP offers some novel insights and tools
- Emerging topic areas we did not cover
  - $\Rightarrow$ Directed graphs and causal structure identification
  - $\Rightarrow$ Dynamic networks and multi-layer graphs
  - $\Rightarrow$ Nonlinear models of interaction
- Open research directions
  - $\Rightarrow$ Performance guarantees such as those for graphical lasso
  - $\Rightarrow$ Does smoothness alone suffice? Can sparsity be forgone?
  - $\Rightarrow$ Bi-level network inference: graphs for higher-level tasks
  - $\Rightarrow$ Discrete signals, non-linear graph filter based models
  - $\Rightarrow$ Scalability via online and/or parallel algorithms