Block 5: Graph Signal Processing

ELEC 573: Network Science and Analytics

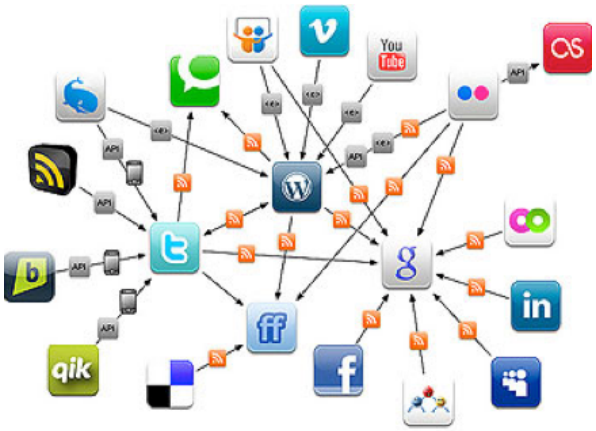
Santiago Segarra

Electrical and Computer Engineering
Rice University
segarra@rice.edu

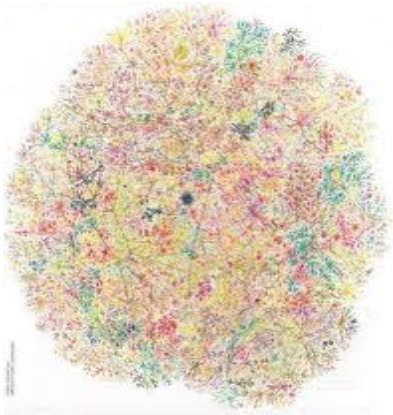
Fall 2021

Network Science analytics

Online social media



Internet

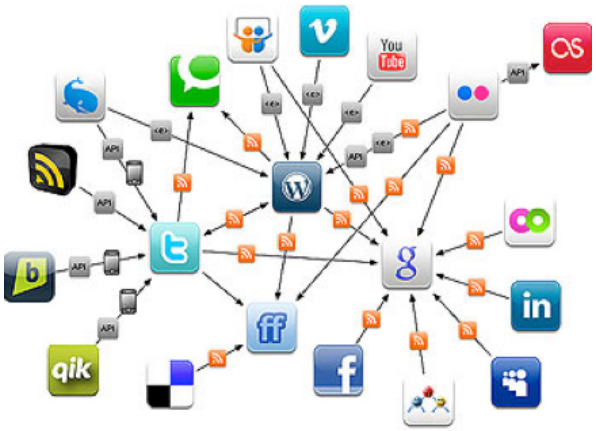


Clean energy and grid analytics

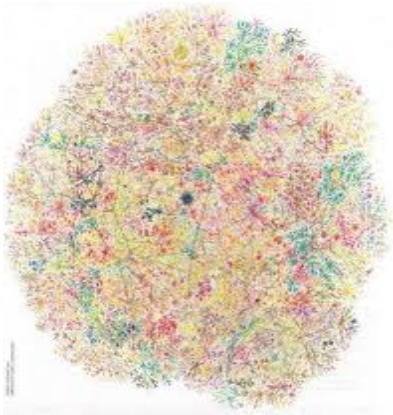
- Desiderata: Process, analyze and learn from network data [Kolaczyk'09]

Network Science analytics

Online social media



Internet



Clean energy and grid analytics

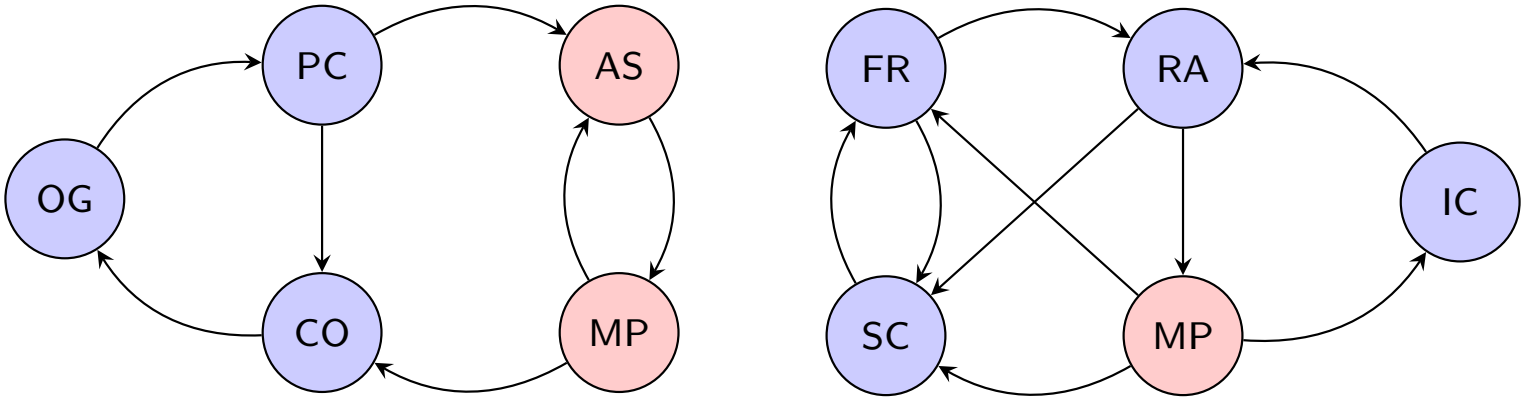
- Desiderata: Process, analyze and learn from network data [Kolaczyk'09]
- Network as graph $G = (\mathcal{V}, \mathcal{E}, W)$: encode pairwise relationships
- Interest here not in G itself, but in data associated with nodes in $\mathcal{V}$

  $\Rightarrow$ Object of study is a graph signal $\mathbf{x}$
- Q: Graph signals common and interesting as networks are?

Network of economic sectors of the United States

- Bureau of Economic Analysis of the U.S. Department of Commerce
- $\mathcal{E}$ = Output of sector i is an input to sector j (62 sectors in $\mathcal{V}$)

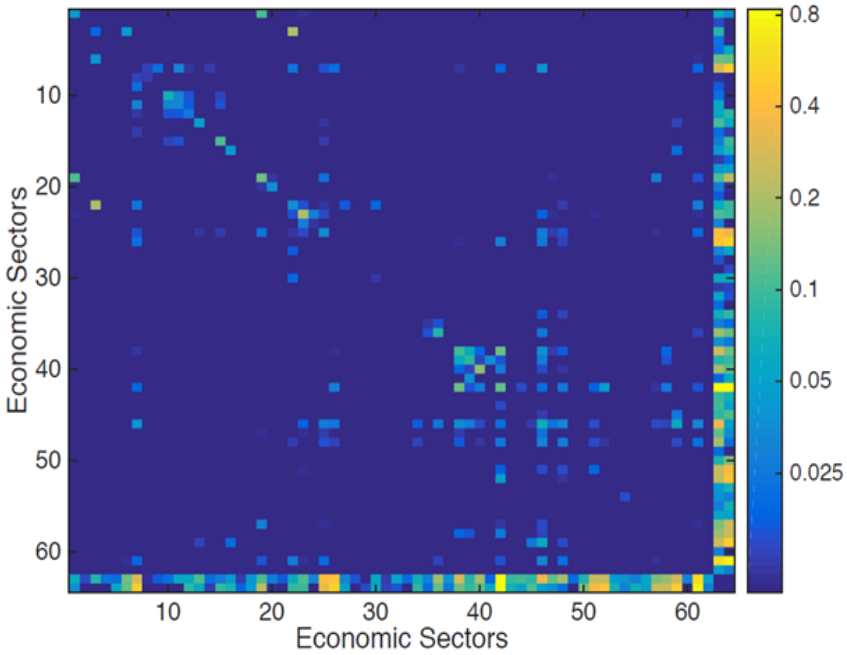
Oil and Gas Services Finance



- 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
- $\mathcal{E}$ = Output of sector i is an input to sector j (62 sectors in $\mathcal{V}$)

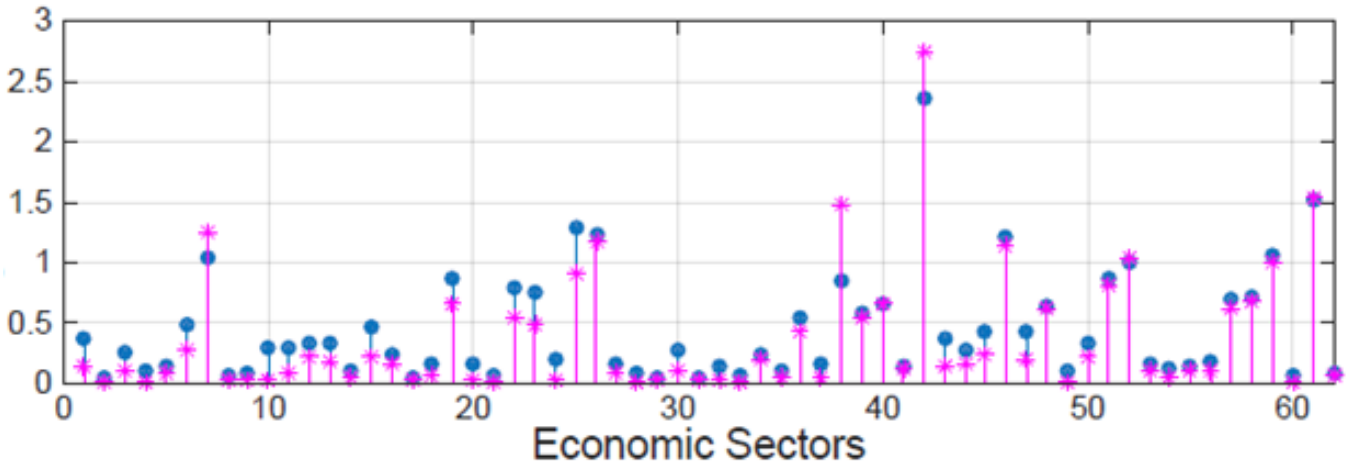


- A few sectors have widespread strong influence (services, finance, energy)
- Some sectors have strong indirect influences (oil)
- The heavy last row is final consumption

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

Disaggregated GDP of the United States

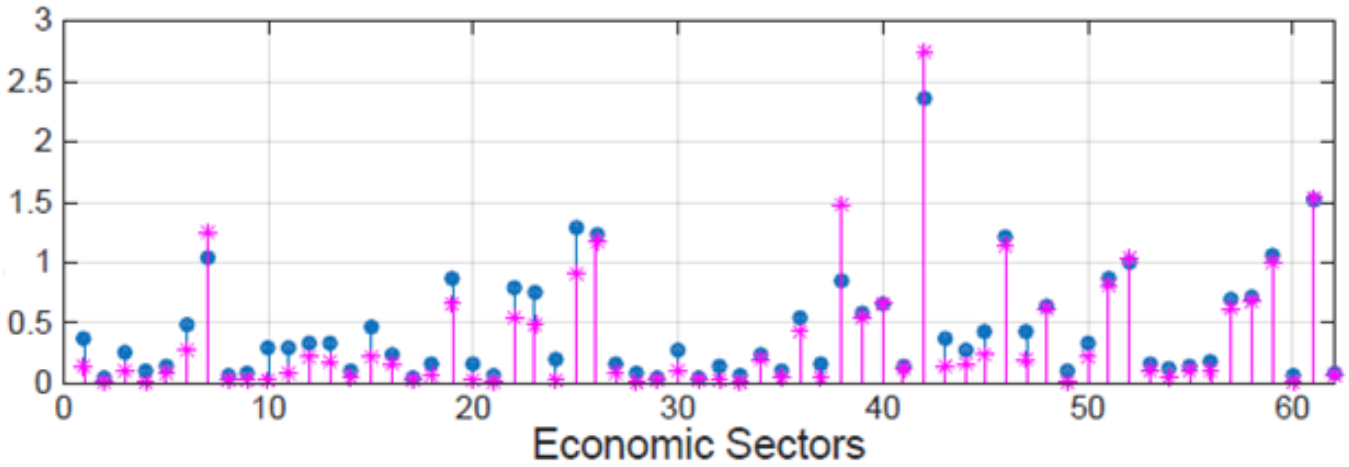
- Signal $\mathbf{x}$ = output per sector = disaggregated GDP
  - $\Rightarrow$ Network structure used to, e.g., reduce GDP estimation noise



- 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

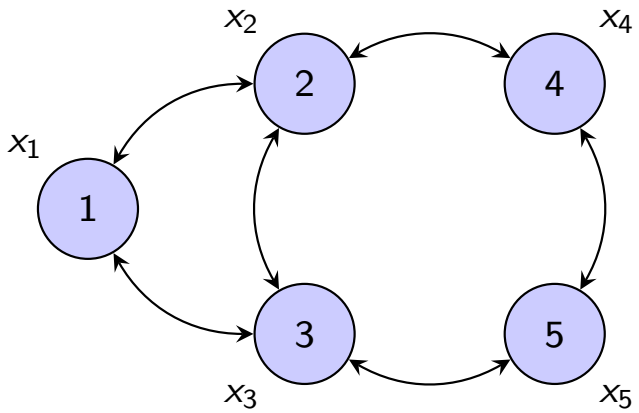


- Signal is as interesting as the network itself. Arguably more
  - Same is true on brain connectivity and fMRI brain signals, ...
  - Gene regulatory networks and gene expression levels, ...
  - Online social networks and information cascades, ...
  - Alignment of customer preferences and product ratings, ...

Graph signal processing

- Graph SP: broaden classical SP to graph signals [Shuman et al.'13]

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



- **As.:** Signal properties related to topology of G (locality, smoothness)

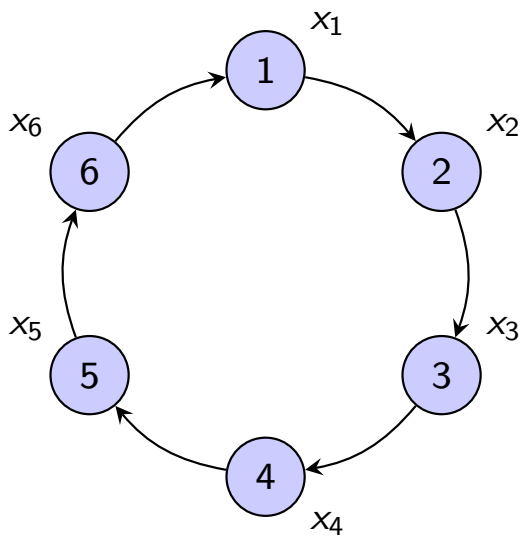
  $\Rightarrow$ Algorithms that fruitfully leverage this relational structure

- Q: Why do we expect the graph structure to be useful in processing $\mathbf{x}$?

Importance of signal structure in time

- Signal and Information Processing is about exploiting signal structure

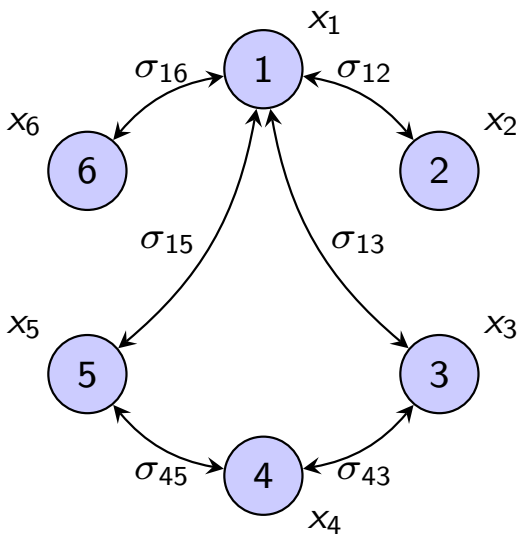
- Discrete time described by cyclic graph
  - $\Rightarrow$ Time n follows time $n-1$
  - $\Rightarrow$ Signal value x_n similar to x_{n-1}
- Formalized with the notion of frequency



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

Covariances and principal components

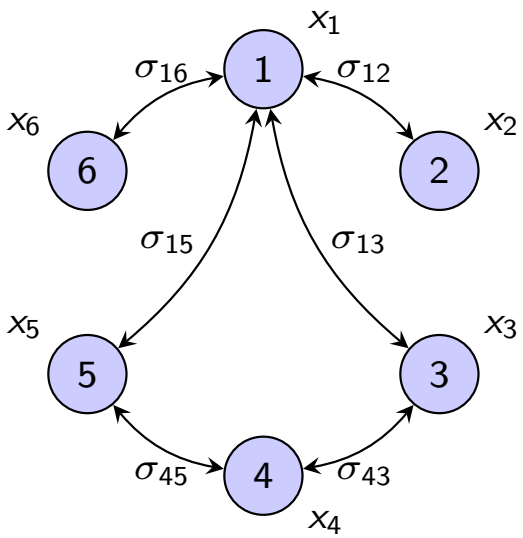
- Random signal with mean $\mathbb{E}[\mathbf{x}] = 0$ and covariance $\mathbf{C}_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^H\right]$
  - $\Rightarrow$ Eigenvector decomposition $\mathbf{C}_x = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$
- Covariance matrix $\mathbf{C}_x$ is a graph
  - $\Rightarrow$ Not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



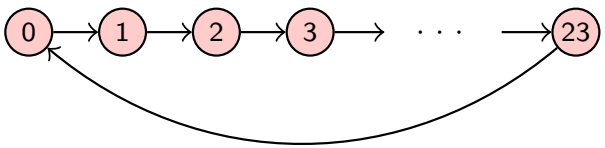
- Covariance matrix structure $\Rightarrow$ Principal components (PCA) $\Rightarrow \tilde{\mathbf{x}} = \mathbf{V}^H\mathbf{x}$
- PCA transform $\Rightarrow$ Projection on eigenvector space of (inverse) covariance
- Q: Can we extend these principles to general graphs and signals?

Graphs 101

- Formally, a graph G (or a network) is a triplet $(\mathcal{V}, \mathcal{E}, W)$
- $\mathcal{V} = \{1, 2, \ldots, N\}$ is a finite set of N nodes or vertices
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges defined as ordered pairs (n, m)
  - Write $\mathcal{N}(n) = \{m \in \mathcal{V} : (m, n) \in \mathcal{E}\}$ as the in-neighbors of n

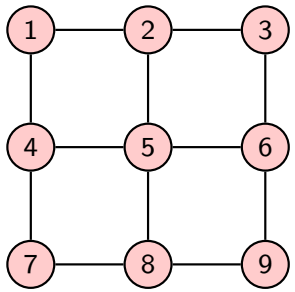
- Formally, a graph G (or a network) is a triplet $(\mathcal{V}, \mathcal{E}, W)$
- $\mathcal{V} = \{1, 2, \ldots, N\}$ is a finite set of N nodes or vertices
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges defined as ordered pairs (n, m)
  - Write $\mathcal{N}(n) = \{m \in \mathcal{V} : (m, n) \in \mathcal{E}\}$ as the in-neighbors of n
- $W : \mathcal{E} \to \mathbb{R}$ is a map from the set of edges to scalar values w_{nm}
  - Represents the level of relationship from n to m
  - Often weights are strictly positive, $W : \mathcal{E} \to \mathbb{R}_{++}$
- Unweighted graphs $\Rightarrow w_{nm} \in \{0, 1\}$, for all $(n, m) \in \mathcal{E}$
- Undirected graphs $\Rightarrow (n, m) \in \mathcal{E}$ if and only if $(m, n) \in \mathcal{E}$ and $w_{nm} = w_{mn}$, for all $(n, m) \in \mathcal{E}$

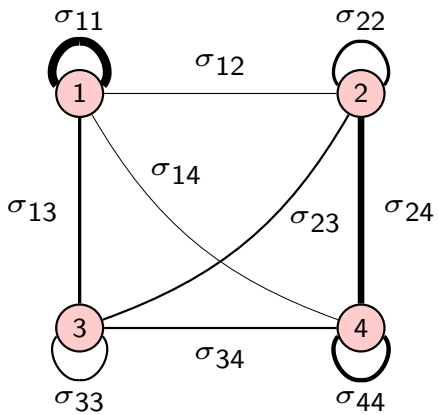
Graphs – examples



- Unweighted and directed graphs (e.g., time)
  - $\mathcal{V} = \{0, 1, \ldots, 23\}$
  - $\mathcal{E} = \{(0, 1), (1, 2), \ldots, (22, 23), (23, 0)\}$
  - $W : (n, m) \mapsto 1$, for all $(n, m) \in \mathcal{E}$

- Unweighted and undirected graphs (e.g., image)
  - $\mathcal{V} = \{1, 2, 3, \ldots, 9\}$
  - $\mathcal{E} = \{(1, 2), (2, 3), \ldots, (8, 9), (1, 4), \ldots, (6, 9)\}$
  - $W : (n, m) \mapsto 1$, for all $(n, m) \in \mathcal{E}$





- Weighted and undirected graphs (e.g., covariance)
  - $\mathcal{V} = \{1, 2, 3, 4\}$
  - $\mathcal{E} = \{(1, 1), (1, 2), \ldots, (4, 4)\} = \mathcal{V} \times \mathcal{V}$
  - $W : (n, m) \mapsto \sigma_{nm} = \sigma_{mn}$, for all (n, m)

Adjacency matrix

- Algebraic graph theory: matrices associated with a graph G
  - $\Rightarrow$ Adjacency $\mathbf{A}$ and Laplacian $\mathbf{L}$ matrices
  - $\Rightarrow$ Spectral graph theory: properties of G using spectrum of $\mathbf{A}$ or $\mathbf{L}$

Adjacency matrix

- Algebraic graph theory: matrices associated with a graph G
  - $\Rightarrow$ Adjacency $\mathbf{A}$ and Laplacian $\mathbf{L}$ matrices
  - $\Rightarrow$ Spectral graph theory: properties of G using spectrum of $\mathbf{A}$ or $\mathbf{L}$
- Given $G = (\mathcal{V}, \mathcal{E}, W)$, the adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ is

$$A_{nm} = \begin{cases} w_{nm}, & \text{if } (n,m) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}$$

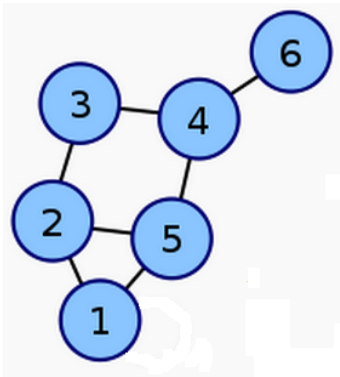
- Matrix representation incorporating all information about G
  - $\Rightarrow$ For unweighted graphs, positive entries represent connected pairs
  - $\Rightarrow$ For weighted graphs, also denote proximities between pairs

Degree and k-hop neighbors

- If G is unweighted and undirected, the degree of node i is $|\mathcal{N}(i)|$
  - $\Rightarrow$ In directed graphs, have out-degree and an in-degree
- Using the adjacency matrix in the undirected case
  - $\Rightarrow$ For node i: $\deg(i) = \sum_{j \in \mathcal{N}(i)} A_{ij} = \sum_j A_{ij}$
  - $\Rightarrow$ For all N nodes: $\mathbf{d} = \mathbf{A1} \rightarrow$ Degree matrix: $\mathbf{D} := \text{diag}(\mathbf{d})$

Degree and k-hop neighbors

- If G is unweighted and undirected, the degree of node i is $|\mathcal{N}(i)|$
  - $\Rightarrow$ In directed graphs, have out-degree and an in-degree
- Using the adjacency matrix in the undirected case
  - $\Rightarrow$ For node i: $\deg(i) = \sum_{j \in \mathcal{N}(i)} A_{ij} = \sum_j A_{ij}$
  - $\Rightarrow$ For all N nodes: $\mathbf{d} = \mathbf{A1} \rightarrow$ Degree matrix: $\mathbf{D} := \text{diag}(\mathbf{d})$
- Q: Can this be extended to k-hop neighbors? $\rightarrow$ Powers of $\mathbf{A}$
  - $\Rightarrow$ $[\mathbf{A}^k]_{ij}$ non-zero only if there exists a path of length k from i to j
  - $\Rightarrow$ Support of $\mathbf{A}^k$: pairs that can be reached in k hops



$[A^2]_{3,5}$

$$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

Laplacian of a graph

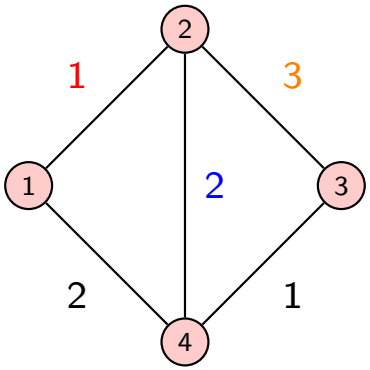
- Given undirected G with $\mathbf{A}$ and $\mathbf{D}$, the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ is

$$\mathbf{L} = \mathbf{D} - \mathbf{A}$$

$\Rightarrow$ Equivalently, $\mathbf{L}$ can be defined element-wise as

$$L_{ij} = \begin{cases} \deg(i), & \text{if } i = j \\ -w_{ij}, & \text{if } (i,j) \in \mathcal{E} \\ 0, & \text{otherwise} \end{cases}$$

- Normalized Laplacian: $\mathcal{L} = \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2}$ (we will focus on $\mathbf{L}$)



$$\mathbf{L} = \begin{bmatrix} 3 & -1 & 0 & -2 \\ -1 & 6 & -3 & -2 \\ 0 & -3 & 4 & -1 \\ -2 & -2 & -1 & 5 \end{bmatrix}$$

Spectral properties of the Laplacian

- Denote by λ_i and $\mathbf{v}_i$ the eigenvalues and eigenvectors of $\mathbf{L}$

Spectral properties of the Laplacian

- Denote by λ_i and $\mathbf{v}_i$ the eigenvalues and eigenvectors of $\mathbf{L}$
- $\mathbf{L}$ is positive semi-definite
  - $\Rightarrow \mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} w_{ij}(x_i - x_j)^2 \geq 0$, for all $\mathbf{x}$
  - $\Rightarrow$ All eigenvalues are nonnegative, i.e. $\lambda_i \geq 0$ for all i

Spectral properties of the Laplacian

- Denote by λ_i and $\mathbf{v}_i$ the eigenvalues and eigenvectors of $\mathbf{L}$
- $\mathbf{L}$ is positive semi-definite
  - $\Rightarrow \mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} w_{ij}(x_i - x_j)^2 \geq 0$, for all $\mathbf{x}$
  - $\Rightarrow$ All eigenvalues are nonnegative, i.e. $\lambda_i \geq 0$ for all i
- A constant vector $\mathbf{1}$ is an eigenvector of $\mathbf{L}$ with eigenvalue 0

$$[\mathbf{L}\mathbf{1}]_i = \sum_{j\in\mathcal{N}(i)} w_{ij}(1-1) = 0$$

  - $\Rightarrow$ Thus, $\lambda_1 = 0$ and $\mathbf{v}_1 = (1/\sqrt{N})\ \mathbf{1}$

Spectral properties of the Laplacian

- Denote by λ_i and $\mathbf{v}_i$ the eigenvalues and eigenvectors of $\mathbf{L}$
- $\mathbf{L}$ is positive semi-definite
  - $\Rightarrow \mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{(i,j)\in\mathcal{E}} w_{ij}(x_i - x_j)^2 \geq 0$, for all $\mathbf{x}$
  - $\Rightarrow$ All eigenvalues are nonnegative, i.e. $\lambda_i \geq 0$ for all i
- A constant vector $\mathbf{1}$ is an eigenvector of $\mathbf{L}$ with eigenvalue 0

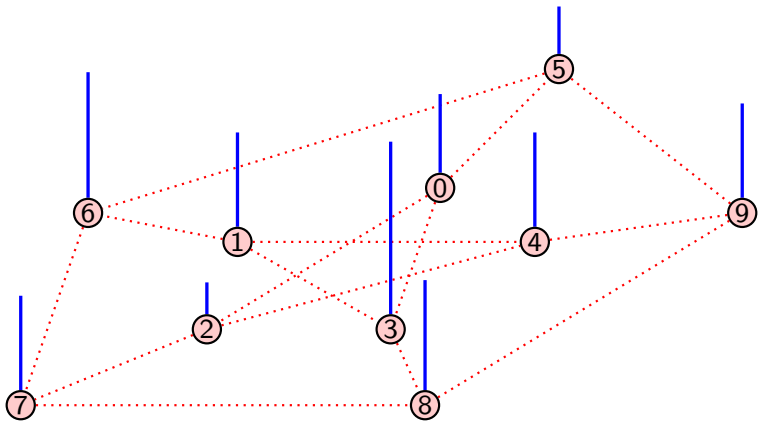
$$[\mathbf{L}\mathbf{1}]_i = \sum_{j\in\mathcal{N}(i)} w_{ij}(1-1) = 0$$

  - $\Rightarrow$ Thus, $\lambda_1 = 0$ and $\mathbf{v}_1 = (1/\sqrt{N})\,\mathbf{1}$
- In connected graphs, it holds that $\lambda_i > 0$ for $i = 2, \ldots, N$
  - $\Rightarrow$ Multiplicity$\{\lambda = 0\}$ = number of connected components

Graph signals

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

  Ex: Opinion profile, buffer congestion levels, neural activity, epidemic
- May be represented as a vector $\mathbf{x} \in \mathbb{R}^N$
  $\Rightarrow$ x_n denotes the signal value at the n-th vertex in $\mathcal{V}$
  $\Rightarrow$ Implicit ordering of vertices (same as in $\mathbf{A}$ or $\mathbf{L}$)

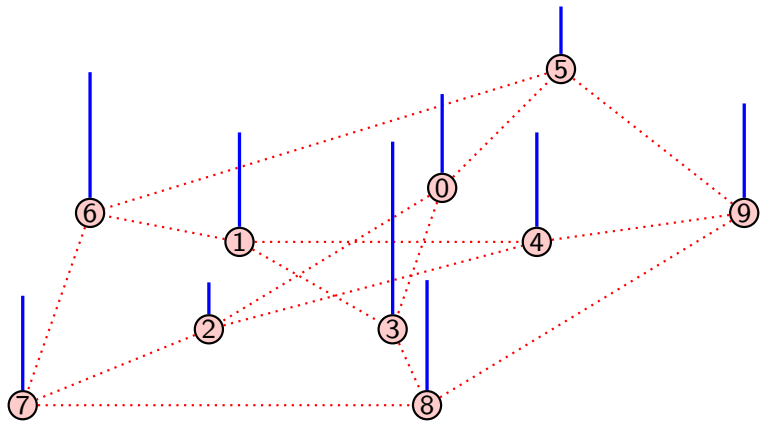


$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

Graph signals

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

  Ex: Opinion profile, buffer congestion levels, neural activity, epidemic
- May be represented as a vector $\mathbf{x} \in \mathbb{R}^N$
  $\Rightarrow$ x_n denotes the signal value at the n-th vertex in $\mathcal{V}$
  $\Rightarrow$ Implicit ordering of vertices (same as in $\mathbf{A}$ or $\mathbf{L}$)



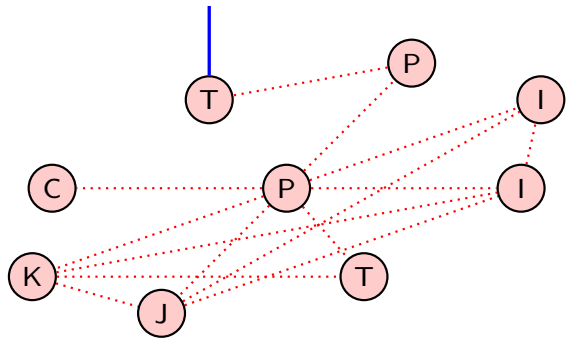
$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

- Data associated with links of G $\Rightarrow$ Use line graph of G

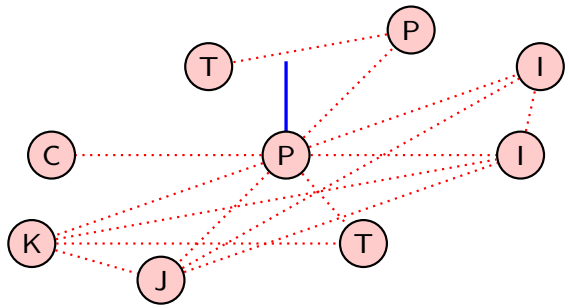
Graph signals – Genetic profiles

- Graphs representing gene-gene interactions
  - $\Rightarrow$ Each node denotes a single gene (loosely speaking)
  - $\Rightarrow$ Connected if their coded proteins participate in same metabolism
- Genetic profiles for each patient can be considered as a graph signal
  - $\Rightarrow$ Signal on each node is 1 if mutated and 0 otherwise

$$\mathbf{x}_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$



Sample patient 1 with subtype 1



Sample patient 2 with subtype 1

$$\mathbf{x}_2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

- To understand a graph signal, the structure of G must be considered

Graph-shift operator

- To understand and analyze $\mathbf{x}$, useful to account for G's structure

Graph-shift operator

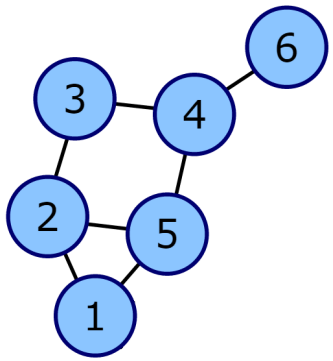
- To understand and analyze $\mathbf{x}$, useful to account for G's structure
- Associated with G is the graph-shift operator $\mathbf{S} \in \mathbb{R}^{N \times N}$

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

Graph-shift operator

- To understand and analyze $\mathbf{x}$, useful to account for G's structure
- Associated with G is the graph-shift operator $\mathbf{S} \in \mathbb{R}^{N \times N}$

  $\Rightarrow S_{ij} = 0$ for $i \neq j$ and $(i, j) \notin \mathcal{E}$ (captures local structure in G)
- $\mathbf{S}$ can take nonzero values in the edges of G or in its diagonal



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

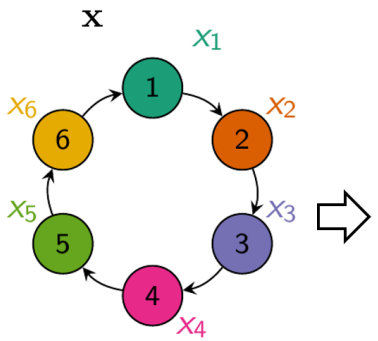
- Ex: Adjacency $\mathbf{A}$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices

Relevance of the graph-shift operator

- Q: Why is **S** called shift?

Relevance of the graph-shift operator

- Q: Why is **S** called shift? A: Resemblance to time shifts



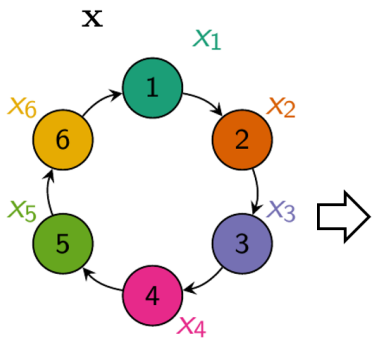
Set $\mathbf{S} = \mathbf{A_{dc}}$

What is $\mathbf{Sx}$?

$$\begin{pmatrix} 0 \\ 0 \\ x_2 \\ x_3 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ x_2 \\ x_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Relevance of the graph-shift operator

- Q: Why is **S** called shift? A: Resemblance to time shifts



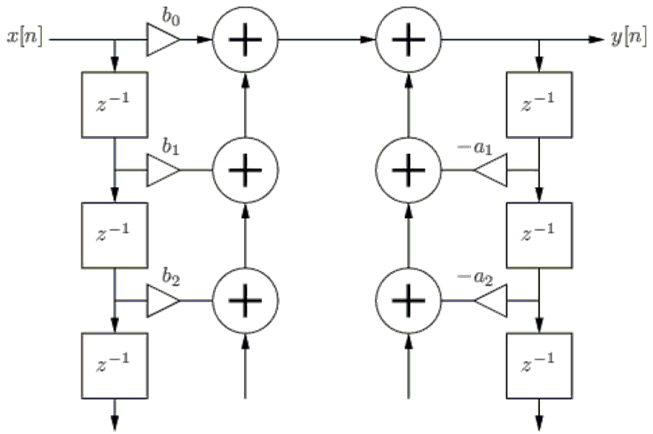
Set $\mathbf{S} = \mathbf{A_{dc}}$

What is $\mathbf{Sx}$?

$$\begin{pmatrix} 0 \\ 0 \\ x_2 \\ x_3 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 0 \\ x_2 \\ x_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

- **S** will be building block for GSP algorithms (More soon)

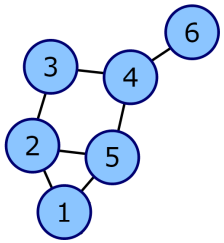
  $\Rightarrow$ Same is true in the time domain (filters and delay)



Local structure of graph-shift operator

S *represents a linear transformation that can be computed locally at the nodes of the graph. More rigorously, if* $\mathbf{y}$ *is defined as* $\mathbf{y} = \mathbf{S}\mathbf{x}$*, then node* i *can compute* y_i *if it has access to* x_j *at* $j \in \mathcal{N}(i)$*.*

- Straightforward because $[\mathbf{S}]_{ij} \neq 0$ only if $i = j$ or $(j, i) \in \mathcal{E}$

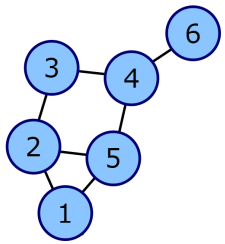


$$\Rightarrow \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

Local structure of graph-shift operator

S *represents a linear transformation that can be computed locally at the nodes of the graph. More rigorously, if* $\mathbf{y}$ *is defined as* $\mathbf{y} = \mathbf{S}\mathbf{x}$*, then node* i *can compute* y_i *if it has access to* x_j *at* $j \in \mathcal{N}(i)$*.*

- Straightforward because $[\mathbf{S}]_{ij} \neq 0$ only if $i = j$ or $(j, i) \in \mathcal{E}$



$$\Rightarrow \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \end{pmatrix} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{pmatrix}$$

- What if $\mathbf{y} = \mathbf{S}^2\mathbf{x}$?

  $\Rightarrow$ Like powers of $\mathbf{A}$: neighborhoods

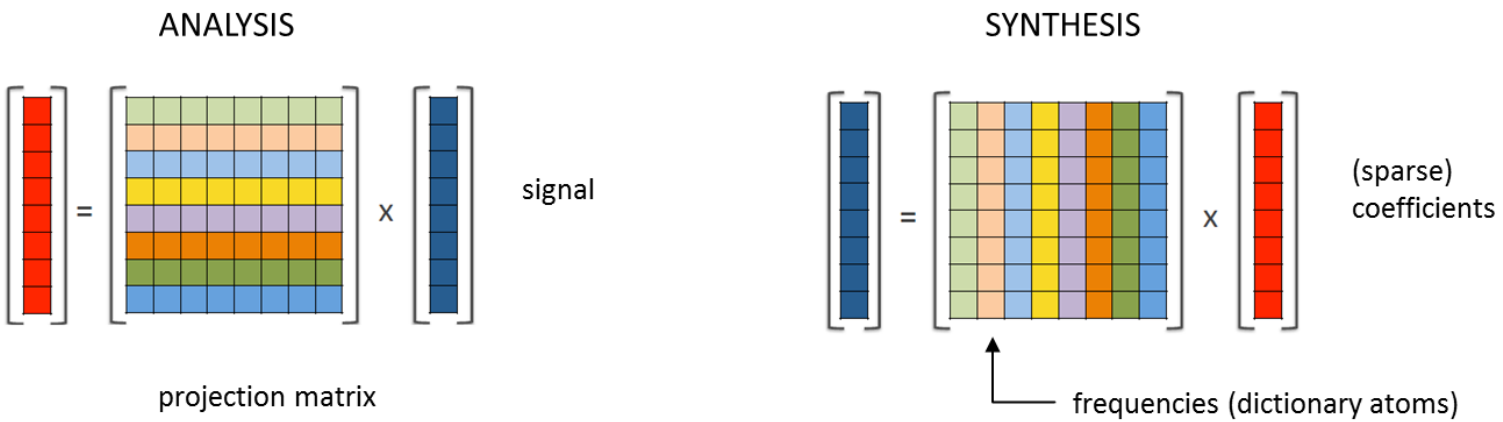
  $\Rightarrow$ y_i found using values within 2-hops

$$[\mathbf{S}^2]_{3,5} = S_{3,2}S_{2,5} + S_{3,4}S_{4,5}$$

$$\mathbf{S}^2 = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix} \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix}$$

Discrete Fourier Transform (DFT)

- Let $\mathbf{x}$ be a temporal signal, its DFT is $\tilde{\mathbf{x}} = \mathbf{F}^H \mathbf{x}$, with $F_{kn} = \frac{1}{\sqrt{N}} e^{+j\frac{2\pi}{N}kn}$
  - $\Rightarrow$ Equivalent description, provides insights
  - $\Rightarrow$ Oftentimes, more parsimonious (bandlimited)
  - $\Rightarrow$ Facilitates the design of SP algorithms: e.g., filters
- Many other transformations (orthogonal dictionaries) exist



- Q: What transformation is suitable for graph signals?

Graph Fourier Transform (GFT)

- Useful transformation? $\Rightarrow$ $\mathbf{S}$ involved in generation/description of $\mathbf{x}$
  - $\Rightarrow$ Let $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ 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

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

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

Graph Fourier Transform (GFT)

- Useful transformation? $\Rightarrow$ $\mathbf{S}$ involved in generation/description of $\mathbf{x}$
  - $\Rightarrow$ Let $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ be the shift associated with G
- The Graph Fourier Transform (GFT) of $\mathbf{x}$ is defined as

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

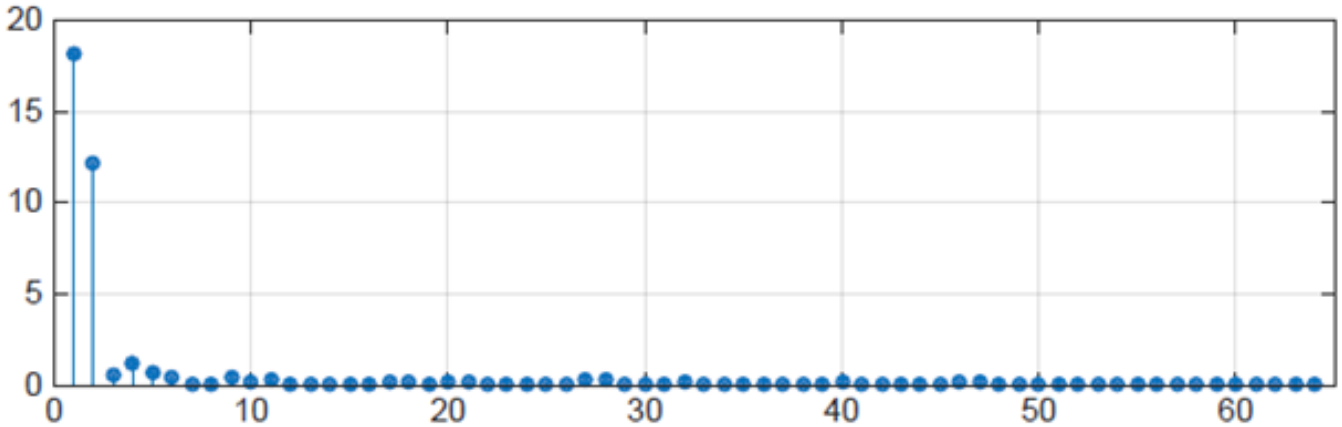
- While the inverse GFT (iGFT) of $\tilde{\mathbf{x}}$ is defined as

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

  - $\Rightarrow$ 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 shift operator $\mathbf{S}$

Is this a reasonable transform?

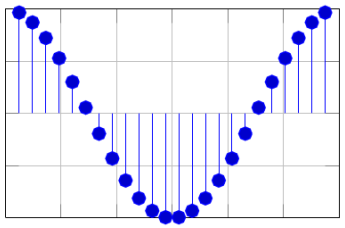
- Particularized to cyclic graphs $\Rightarrow$ GFT $\equiv$ Fourier transform
- Particularized to covariance matrices $\Rightarrow$ GFT $\equiv$ PCA transform
- But really, this is an empirical question. GFT of disaggregated GDP

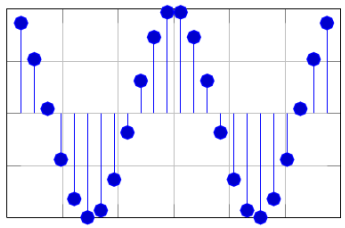


- GFT transform characterized by a few coefficients
  - $\Rightarrow$ Notion of bandlimitedness: $\mathbf{x} = \sum_{k=1}^{K} \tilde{x}_k \mathbf{v}_k$
  - $\Rightarrow$ Sampling, compression, filtering, pattern recognition

Meaning of the eigenvalues

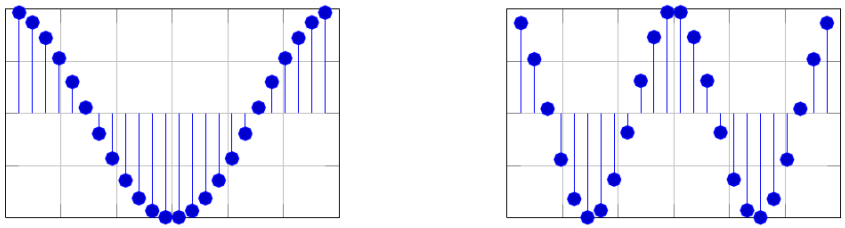
- Columns of $\mathbf{V}$ are the frequency atoms: $\mathbf{x} = \sum_k \tilde{x}_k \mathbf{v}_k$
- Q: What about the eigenvalues $\lambda_k = \Lambda_{kk}$?
  - $\Rightarrow$ When $\mathbf{S} = \mathbf{A}_{dc}$, we get $\lambda_k = e^{-j\frac{2\pi}{N}(k-1)}$
  - $\Rightarrow$ λ_k can be viewed as frequencies!!
- In time, well-defined relation between frequency and variation
  - $\Rightarrow$ Higher k $\Rightarrow$ faster oscillations
  - $\Rightarrow$ Bounds on total-variation: $TV(\mathbf{x}) = \sum_n (x_n - x_{n-1})^2$





Meaning of the eigenvalues

- Columns of $\mathbf{V}$ are the frequency atoms: $\mathbf{x} = \sum_k \tilde{x}_k \mathbf{v}_k$
- Q: What about the eigenvalues $\lambda_k = \Lambda_{kk}$?
  - $\Rightarrow$ When $\mathbf{S} = \mathbf{A}_{dc}$, we get $\lambda_k = e^{-j\frac{2\pi}{N}(k-1)}$
  - $\Rightarrow$ λ_k can be viewed as frequencies!!
- In time, well-defined relation between frequency and variation
  - $\Rightarrow$ Higher k $\Rightarrow$ faster oscillations
  - $\Rightarrow$ Bounds on total-variation: $TV(\mathbf{x}) = \sum_n (x_n - x_{n-1})^2$



- Q: Does this carry over for graph signals? $\Rightarrow$ No, only if $\mathbf{S} = \mathbf{L}$
  - $\Rightarrow$ $\{\lambda_k\}_{k=1}^N$ will be very important when analyzing graph filters

Interpretation for the Laplacian

- Consider a graph G, let $\mathbf{x}$ be a signal on G, and set $\mathbf{S} = \mathbf{L}$
- Define the quadratic form

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} w_{ij} (x_i - x_j)^2$$

$\Rightarrow$ $\mathbf{x}^T \mathbf{L} \mathbf{x}$ quantifies the (aggregated) local variation of signal $\mathbf{x}$

$\Rightarrow$ Natural measure of signal smoothness w.r.t. G

Interpretation for the Laplacian

- Consider a graph G, let $\mathbf{x}$ be a signal on G, and set $\mathbf{S} = \mathbf{L}$
- Define the quadratic form

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} w_{ij} (x_i - x_j)^2$$

  $\Rightarrow$ $\mathbf{x}^T \mathbf{L} \mathbf{x}$ quantifies the (aggregated) local variation of signal $\mathbf{x}$

  $\Rightarrow$ Natural measure of signal smoothness w.r.t. G

- Q: Interpretation of frequencies $\{\lambda_k\}_{k=1}^N$ when $\mathbf{S} = \mathbf{L}$?

  $\Rightarrow$ If $\mathbf{x} = \mathbf{v}_k$, we get $\mathbf{x}^T \mathbf{L} \mathbf{x} = \lambda_k$ $\Rightarrow$ local variation of $\mathbf{v}_k$

  $\Rightarrow$ Frequencies account for local variation, they can be ordered

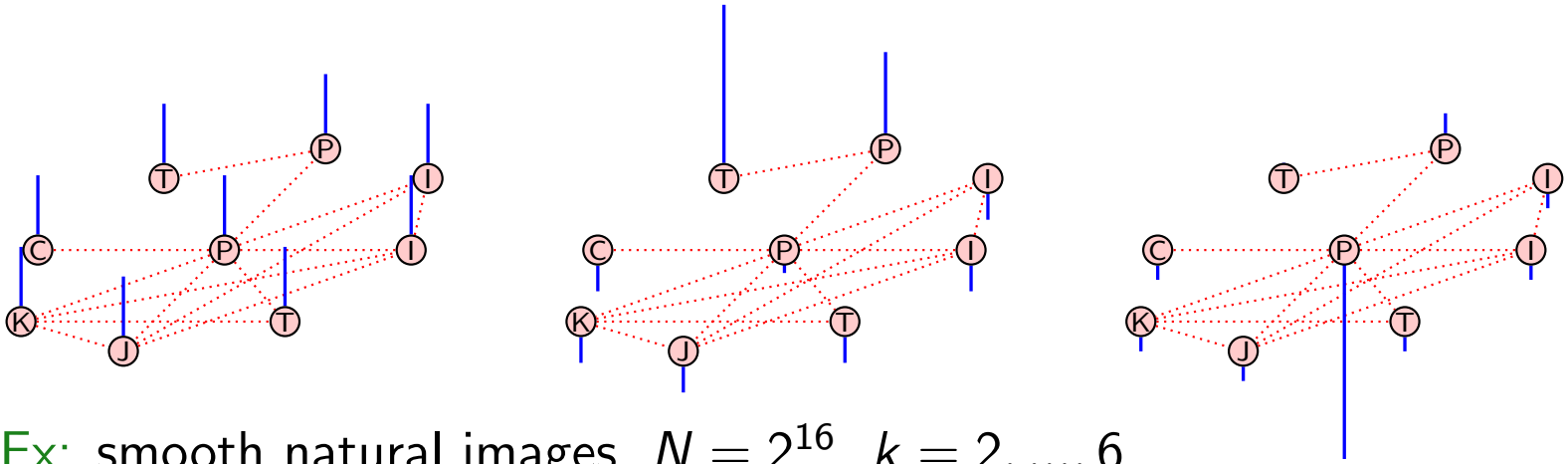
  $\Rightarrow$ Eigenvector associated with eigenvalue 0 is constant

Frequencies of the Laplacian

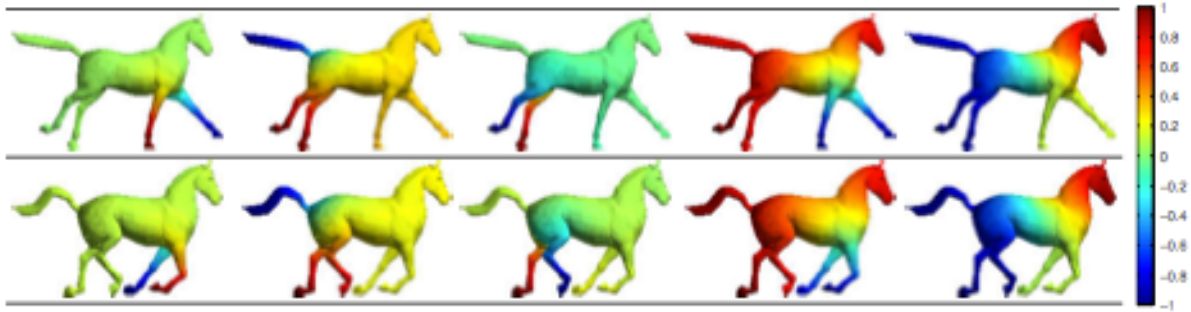
- Laplacian eigenvalue λ_k accounts for the local variation of $\mathbf{v}_k$

  $\Rightarrow$ Let us plot some of the eigenvectors of $\mathbf{L}$ (also graph signals)

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



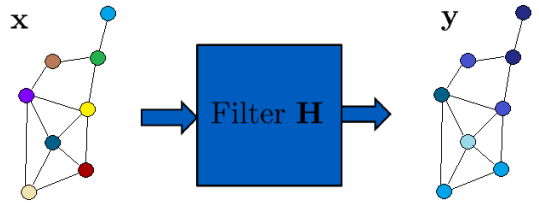
- Ex: smooth natural images, $N = 2^{16}$, $k = 2, \ldots, 6$



Linear (shift-invariant) graph filter

- A graph filter $H : \mathbb{R}^N \to \mathbb{R}^N$ is a map between graph signals

  Focus on linear filters
  $\Rightarrow$ map represented by an $N \times N$ matrix



Def1: Polynomial in $\mathbf{S}$ of degree L, with coeff. $\mathbf{h} = [h_0, \ldots, h_L]^T$

$$\mathbf{H} := h_0\mathbf{S}^0 + h_1\mathbf{S}^1 + \ldots + h_L\mathbf{S}^L = \sum_{l=0}^{L} h_l\mathbf{S}^l \quad \text{[Sandryhaila13]}$$

Def2: Orthogonal operator in the frequency domain

$$\mathbf{H} := \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^{-1}, \quad \tilde{h}_k = g(\lambda_k)$$

- With $[\mathbf{\Psi}]_{k,l} := \lambda_k^{l-1}$, we have $\tilde{\mathbf{h}} = \mathbf{\Psi}\mathbf{h} \Rightarrow$ Defs can be rendered equivalent
  $\Rightarrow$ More on this later, now focus on Def1
- If $\mathbf{y} := \mathbf{H}\mathbf{x}$, Def1 says $\mathbf{y}$ linear combination of shifted versions of $\mathbf{x}$

Graph filters as linear network operators

- Def1 says $\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{S}^l$
- Suppose $\mathbf{H}$ acts on a graph signal $\mathbf{x}$ to generate $\mathbf{y} = \mathbf{H}\mathbf{x}$
  - $\Rightarrow$ If we define $\mathbf{x}^{(l)} := \mathbf{S}^l \mathbf{x} = \mathbf{S}\mathbf{x}^{(l-1)}$

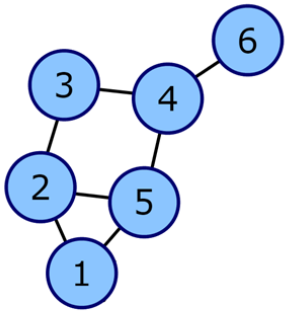
$$\mathbf{y} = \sum_{l=0}^{L} h_l \mathbf{x}^{(l)}$$

$\mathbf{y}$ is a linear combination of successive shifted versions of $\mathbf{x}$

- After introducing $\mathbf{S}$, we stressed that $\mathbf{y} = \mathbf{S}\mathbf{x}$ can be computed locally
  - $\Rightarrow$ $\mathbf{x}^{(l)}$ can be found locally if $\mathbf{x}^{(l-1)}$ is known
  - $\Rightarrow$ The output of the filter can be found in L local steps
- A graph filter represents a linear transformation that
  - $\Rightarrow$ Accounts for local structure of the graph
  - $\Rightarrow$ Can be implemented distributedly in L steps
  - $\Rightarrow$ Only requires info in L-neighborhood [Shuman13, Sandryhaila14]

An example of a graph filter

- $\mathbf{x}=[-1,2,0,0,0,0]^T$, $\mathbf{h}=[1,1,0.5]^T$, $\mathbf{y}=(\sum_{l=0}^{L} h_l\mathbf{S})\mathbf{x}=\sum_{l=0}^{L} h_l\mathbf{x}^{(l)}$

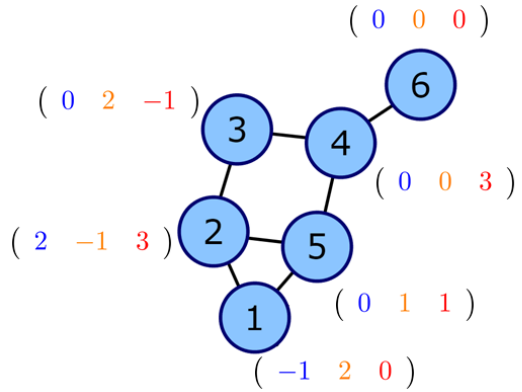


$$\mathbf{S}=\mathbf{A}=\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$$\mathbf{y}=\sum_{l=0}^{L} h_l\mathbf{S}^l\mathbf{x}=\sum_{l=0}^{L} h_l\mathbf{x}^{(l)}$$

$$\mathbf{y}=h_0\mathbf{x}^{(0)}+h_1\mathbf{x}^{(1)}+h_2\mathbf{x}^{(2)}$$

Given $\mathbf{x}=[-1,2,0,0,0,0]^T$ and $\mathbf{h}=[1,1,0.5]^T \Rightarrow$ Find $\{\mathbf{x}^{(0)},\mathbf{x}^{(1)},\mathbf{x}^{(2)}\} \Rightarrow$ Find $\mathbf{y}$



$$\mathbf{x}^{(0)}=\mathbf{x}=\begin{pmatrix} -1 \\ 2 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad \mathbf{x}^{(1)}=\mathbf{S}\mathbf{x}^{(0)}=\begin{pmatrix} 2 \\ -1 \\ 2 \\ 0 \\ 1 \\ 0 \end{pmatrix} \quad \mathbf{x}^{(2)}=\mathbf{S}\mathbf{x}^{(1)}=\begin{pmatrix} 0 \\ 3 \\ -1 \\ 3 \\ 1 \\ 0 \end{pmatrix}$$

$$\mathbf{y}=1\mathbf{x}^{(0)}+1\mathbf{x}^{(1)}+0.5\mathbf{x}^{(2)}=\begin{pmatrix} 1.0 \\ 2.5 \\ 1.5 \\ 1.5 \\ 1.5 \\ 0.0 \end{pmatrix}$$

Frequency response of a graph filter

- Def2 says $\mathbf{H} = \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^{-1}$
  - $\Rightarrow$ Since $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, Def1 $\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{S}^l$ yields

$$\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{S}^l = \sum_{l=0}^{L} h_l \mathbf{V}\mathbf{\Lambda}^l\mathbf{V}^{-1} = \mathbf{V}\left(\sum_{l=0}^{L} h_l \mathbf{\Lambda}^l\right)\mathbf{V}^{-1}$$

- The application $\mathbf{H}\mathbf{x}$ of filter $\mathbf{H}$ to $\mathbf{x}$ can be split into three parts
  - $\Rightarrow$ $\mathbf{V}^{-1}$ takes signal $\mathbf{x}$ to the graph freq. domain $\tilde{\mathbf{x}}$
  - $\Rightarrow$ $\tilde{\mathbf{H}} := \sum_{l=0}^{L} h_l \mathbf{\Lambda}^l$ modulates the freq. coefficients $\tilde{\mathbf{x}}$ to obtain $\tilde{\mathbf{y}}$
  - $\Rightarrow$ $\mathbf{V}$ brings the signal $\tilde{\mathbf{y}}$ back to the graph domain $\mathbf{y}$

Frequency response of a graph filter

- Def2 says $\mathbf{H} = \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^{-1}$
  - $\Rightarrow$ Since $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, Def1 $\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{S}^l$ yields

$$\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{S}^l = \sum_{l=0}^{L} h_l \mathbf{V}\mathbf{\Lambda}^l\mathbf{V}^{-1} = \mathbf{V}\left(\sum_{l=0}^{L} h_l \mathbf{\Lambda}^l\right)\mathbf{V}^{-1}$$

- The application $\mathbf{H}\mathbf{x}$ of filter $\mathbf{H}$ to $\mathbf{x}$ can be split into three parts
  - $\Rightarrow$ $\mathbf{V}^{-1}$ takes signal $\mathbf{x}$ to the graph freq. domain $\tilde{\mathbf{x}}$
  - $\Rightarrow$ $\tilde{\mathbf{H}} := \sum_{l=0}^{L} h_l \mathbf{\Lambda}^l$ modulates the freq. coefficients $\tilde{\mathbf{x}}$ to obtain $\tilde{\mathbf{y}}$
  - $\Rightarrow$ $\mathbf{V}$ brings the signal $\tilde{\mathbf{y}}$ back to the graph domain $\mathbf{y}$
- Since $\tilde{\mathbf{H}}$ is diagonal, define $\tilde{\mathbf{H}} =: \textit{diag}(\tilde{\mathbf{h}})$
  - $\Rightarrow$ $\tilde{\mathbf{h}}$ is the frequency response of the filter $\mathbf{H}$
  - $\Rightarrow$ Output at frequency k depends only on input at frequency k

$$\tilde{y}_k = \tilde{h}_k \tilde{x}_k$$

Frequency response and filter coefficients

- Relation between $\tilde{\mathbf{h}}$ and $\mathbf{h}$ in a more friendly manner?
  - $\Rightarrow$ Since $\tilde{\mathbf{h}} = \text{diag}(\sum_{l=0}^{L} h_l \mathbf{\Lambda}^l)$, we have that $\tilde{h}_k = \sum_{l=0}^{L} h_l \lambda_k^l$
  - $\Rightarrow$ Define the Vandermonde matrix $\mathbf{\Psi}$ as

$$\mathbf{\Psi} := \begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^L \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_N & \dots & \lambda_N^L \end{pmatrix}$$

Frequency response of a graph filter

If $\mathbf{h}$ are the coefficients of a graph filter, its frequency response is

$$\tilde{\mathbf{h}} = \mathbf{\Psi}\mathbf{h}$$

Frequency response and filter coefficients

- Relation between $\tilde{\mathbf{h}}$ and $\mathbf{h}$ in a more friendly manner?
  - $\Rightarrow$ Since $\tilde{\mathbf{h}} = \text{diag}(\sum_{l=0}^{L} h_l \mathbf{\Lambda}^l)$, we have that $\tilde{h}_k = \sum_{l=0}^{L} h_l \lambda_k^l$
  - $\Rightarrow$ Define the Vandermonde matrix $\mathbf{\Psi}$ as

$$\mathbf{\Psi} := \begin{pmatrix} 1 & \lambda_1 & \dots & \lambda_1^L \\ \vdots & \vdots & & \vdots \\ 1 & \lambda_N & \dots & \lambda_N^L \end{pmatrix}$$

Frequency response of a graph filter

If $\mathbf{h}$ are the coefficients of a graph filter, its frequency response is

$$\tilde{\mathbf{h}} = \mathbf{\Psi}\mathbf{h}$$

- Given a desired $\tilde{\mathbf{h}}$, we can find the coefficients $\mathbf{h}$ as

$$\mathbf{h} = \mathbf{\Psi}^{-1}\tilde{\mathbf{h}}$$

More on the frequency response

- Since $\mathbf{h} = \boldsymbol{\Psi}^{-1}\tilde{\mathbf{h}} \Rightarrow$ If all $\{\lambda_k\}_{k=1}^{N}$ distinct, then
  - $\Rightarrow$ Any $\tilde{\mathbf{h}}$ can be implemented with at most $L+1 = N$ coefficients
- Since $\tilde{\mathbf{h}} = \boldsymbol{\Psi}\mathbf{h} \Rightarrow$ If $\lambda_k = \lambda_{k'}$, then
  - $\Rightarrow$ The corresponding frequency response will be the same $\tilde{h}_k = \tilde{h}_{k'}$
- For the particular case when $\mathbf{S} = \mathbf{A}_{dc}$, we have that $\lambda_k = e^{-\mathrm{j}\frac{2\pi}{N}(k-1)}$

$$\boldsymbol{\Psi} = \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & e^{-\mathrm{j}\frac{2\pi(1)(1)}{N}} & \dots & e^{-\mathrm{j}\frac{2\pi(1)(N-1)}{N}} \\ \vdots & \vdots & & \vdots \\ 1 & e^{-\mathrm{j}\frac{2\pi(N-1)(1)}{N}} & \dots & e^{-\mathrm{j}\frac{2\pi(N-1)(N-1)}{N}} \end{pmatrix} = \mathbf{F}^H$$

- $\Rightarrow$ The frequency response is the DFT of the impulse response

$$\tilde{\mathbf{h}} = \mathbf{F}^H\mathbf{h}$$

Frequency response for graph signals and filters

- Suppose that we have a signal $\mathbf{x}$ and filter coefficients $\mathbf{h}$
- For time signals, it holds that the output $\mathbf{y}$ is

$$\tilde{\mathbf{y}} = \text{diag}(\mathbf{F}^H \mathbf{h}) \mathbf{F}^H \mathbf{x}$$

- For graph signals, the output $\mathbf{y}$ in the frequency domain is

$$\tilde{\mathbf{y}} = \text{diag}(\boldsymbol{\Psi} \mathbf{h}) \mathbf{V}^{-1} \mathbf{x}$$

- The GFT for filters is different from the GFT for signals
  - $\Rightarrow$ Symmetry is lost, but both depend on spectrum of $\mathbf{S}$
  - $\Rightarrow$ Some of the properties are not true for graphs
  - $\Rightarrow$ Several options to generalize operations

Implementing graph filters: frequency or space

- Frequency or space?

$$\mathbf{y} = \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^{-1}\mathbf{x} \quad \text{vs.} \quad \mathbf{y} = \sum_{l=0}^{L} h_l \mathbf{S}^l \mathbf{x}$$

- In space: leverage the fact that $\mathbf{Sx}$ can be computed locally
  - $\Rightarrow$ Signal $\mathbf{x}$ is percolated L times to find $\{\mathbf{x}^{(l)}\}_{l=0}^{L}$
  - $\Rightarrow$ Every node finds its own y_i by computing $\sum_{l=0}^{L} h_l [\mathbf{x}^{(l)}]_i$
- Frequency implementation useful for processing if, e.g.,
  - $\Rightarrow$ Filter bandlimited and eigenvectors easy to find
  - $\Rightarrow$ Low complexity [Anis16, Tremblay16]
- Space definition useful for modeling
  - $\Rightarrow$ Diffusion, percolation, opinion formation, ...

Implementing graph filters: frequency or space

- Frequency or space?

$$\mathbf{y} = \mathbf{V}\text{diag}(\tilde{\mathbf{h}})\mathbf{V}^{-1}\mathbf{x} \quad \text{vs.} \quad \mathbf{y} = \sum_{l=0}^{L} h_l \mathbf{S}^l \mathbf{x}$$

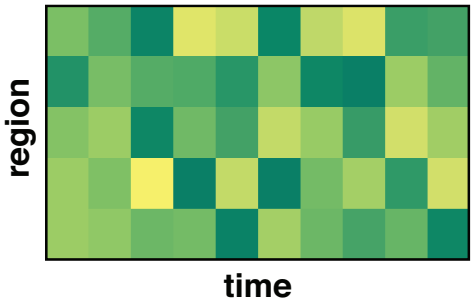
- In space: leverage the fact that $\mathbf{Sx}$ can be computed locally
  - $\Rightarrow$ Signal $\mathbf{x}$ is percolated L times to find $\{\mathbf{x}^{(l)}\}_{l=0}^{L}$
  - $\Rightarrow$ Every node finds its own y_i by computing $\sum_{l=0}^{L} h_l [\mathbf{x}^{(l)}]_i$
- Frequency implementation useful for processing if, e.g.,
  - $\Rightarrow$ Filter bandlimited and eigenvectors easy to find
  - $\Rightarrow$ Low complexity [Anis16, Tremblay16]
- Space definition useful for modeling
  - $\Rightarrow$ Diffusion, percolation, opinion formation, ...
- More on filter design
  - $\Rightarrow$ Chebyshev polyn. [Shuman12]; AR-MA [Isufi-Leus15]; Node-var. [Segarra15]; Time-var. [Isufi-Leus16]; Median filters [Segarra16]

Graph filters: Summary

- Linear graph-signal operators that account for the structure of G
  - $\Rightarrow$ Polynomials of $\mathbf{S}$, orthogonal operators on the frequency domain
- A few take-home messages
  - $\Rightarrow$ Output is weighted sum of shifted inputs
  - $\Rightarrow$ Shifting is not a translation, but a local diffusion
  - $\Rightarrow$ Distributed implementation across L-hop neighborhood
  - $\Rightarrow$ GFT given by Vandermonde matrix $\mathbf{\Psi}$, different from $\mathbf{V}^{-1}$
  - $\Rightarrow$ System identification more involved
- Graph filter design
  - $\Rightarrow$ Designing $\mathbf{h}$, designing $\tilde{\mathbf{h}}$, and... designing $\mathbf{S}$?
- Useful to process signals, but also to model networked phenomena

Application: Explaining human learning rates

- Why do some people learn faster than others?
  - $\Rightarrow$ Can we answer this by looking at their brain activity?
- Brain activity during learning of a motor skill in 112 cortical regions
  - $\Rightarrow$ fMRI while learning a piano pattern for 20 individuals
- Pattern is repeated, reducing the time needed for execution
  - $\Rightarrow$ Learning rate = rate of decrease in execution time
- Define a functional brain graph
  - $\Rightarrow$ Based on correlated activity
- fMRI outputs a series of graph signals
  - $\Rightarrow$ $\mathbf{x}(t) \in \mathbb{R}^{112}$ describing brain states
- Does brain state variability correlate with learning?



Measuring brain state variability

- We propose three different measures capturing different time scales
  $\Rightarrow$ Changes in micro, meso, and macro scales
- Micro: instantaneous changes higher than a threshold α

$$m_1(\mathbf{x}) = \sum_{t=1}^{T} \mathbf{1}\left\{ \frac{\|\mathbf{x}(t) - \mathbf{x}(t-1)\|_2}{\|\mathbf{x}(t)\|_2} > \alpha \right\}$$

- Meso: Cluster brain states and count the changes in clusters

$$m_2(\mathbf{x}) = \sum_{t=1}^{T} \mathbf{1}\left\{ \mathbf{c}(t) \neq \mathbf{c}(t-1) \right\}$$

  $\Rightarrow$ where $\mathbf{c}(t)$ is the cluster to which $\mathbf{x}(t)$ belongs.
- Macro: Sample entropy. Measure of complexity of time series

$$m_3(\mathbf{x}) = -\log\left(\frac{\sum_t \sum_{s \neq t} \mathbf{1}\{\|\bar{\mathbf{x}}_3(t) - \bar{\mathbf{x}}_3(s)\|_\infty > \alpha\}}{\sum_t \sum_{s \neq t} \mathbf{1}\{\|\bar{\mathbf{x}}_2(t) - \bar{\mathbf{x}}_2(s)\|_\infty > \alpha\}} \right)$$

  $\Rightarrow$ Where $\bar{\mathbf{x}}_r(t) = [\mathbf{x}(t), \mathbf{x}(t+1), \ldots, \mathbf{x}(t+r-1)]$

Diffusion as low-pass filtering

- We diffuse each time signal $\mathbf{x}(t)$ across the brain graph

$$\mathbf{x}_{\text{diff}}(t) = (\mathbf{I} + \beta\mathbf{L})^{-1}\mathbf{x}(t)$$

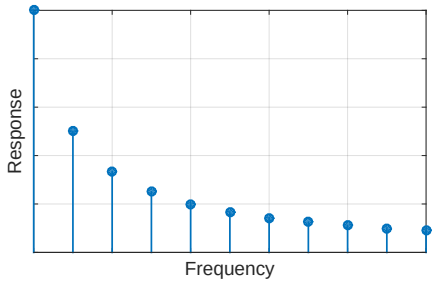
$\Rightarrow$ Laplacian $\mathbf{L} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ and β represents the diffusion rate

- Analyzing diffusion in the frequency domain

$$\tilde{\mathbf{x}}_{\text{diff}}(t) = (\mathbf{I} + \beta\mathbf{\Lambda})^{-1}\mathbf{V}^{-1}\mathbf{x}(t) = \text{diag}(\tilde{\mathbf{h}})\tilde{\mathbf{x}}(t)$$

$\Rightarrow$ Freq. response $\tilde{h}_k = 1/(1 + \beta\lambda_k)$

- Diffusion acts as low-pass filtering
- High freq. components are attenuated
- β controls the level of attenuation



Computing correlation for three signals

- Variability measures consider the order of brain signal activity
- As a control, we include in our analysis a null signal time series $\mathbf{x}_{\text{null}}$

$$\mathbf{x}_{\text{null}}(t) = \mathbf{x}_{\text{diff}}(\pi_t)$$

$\Rightarrow$ π_t is a random permutation of the time indices

- Correlation between variability (m_1, m_2, and m_3) and learning?
- We consider three time series of brain activity
  - $\Rightarrow$ The original fMRI data $\mathbf{x}$
  - $\Rightarrow$ The filtered data $\mathbf{x}_{\text{diff}}$
  - $\Rightarrow$ The null signal $\mathbf{x}_{\text{null}}$

Low-pass filtering reveals correlation

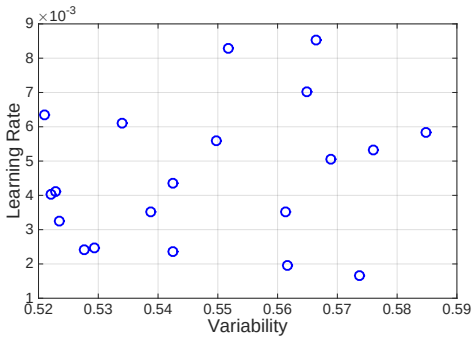
- Correlation coeff. between learning rate and brain state variability

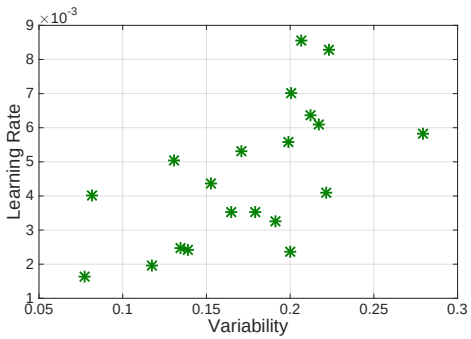
	Original	**Filtered**	Null
m_1	0.211	**0.568**	0.182
m_2	0.226	**0.611**	0.174
m_3	0.114	**0.382**	0.113

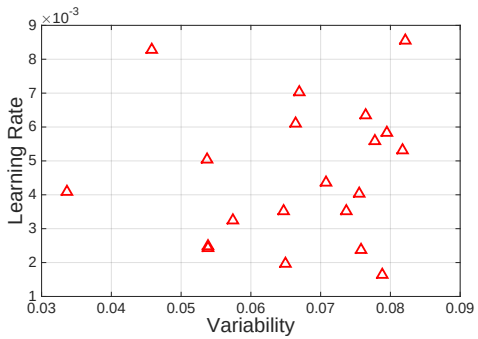
- Correlation is clear when the signal is filtered

  $\Rightarrow$ Result for original signal similar to null signal

- Scatter plots for original, filtered, and null signals (m_2 variability)



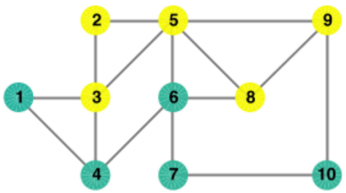


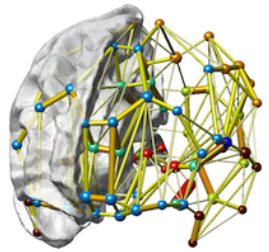


Part II: Applications

Application domains

- Design graph filters to approximate desired network operators
- Sampling bandlimited graph signals
- Blind graph filter identification
  - $\Rightarrow$ Infer diffusion coefficients from observed output
- Statistical GSP: understanding random graph signals
- Network topology inference
  - $\Rightarrow$ Infer shift from collection of network diffused signals



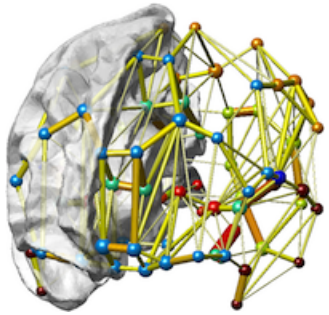


- Many more (glad to discuss or redirect):
  - $\Rightarrow$ Filter banks
  - $\Rightarrow$ Windowing, convolution, duality...
  - $\Rightarrow$ Nonlinear GSP

Network topology inference

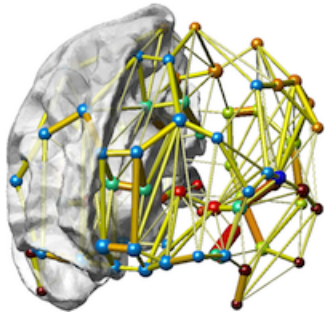
Motivation

- Learning graphs from nodal observations
- Key in neuroscience
  - $\Rightarrow$ Functional network from fMRI signals



Motivation

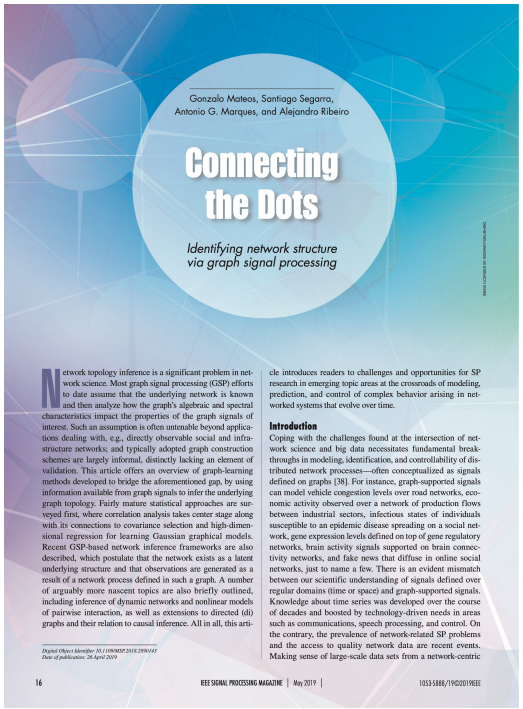
- Learning graphs from nodal observations
- Key in neuroscience
  - $\Rightarrow$ Functional network from fMRI signals

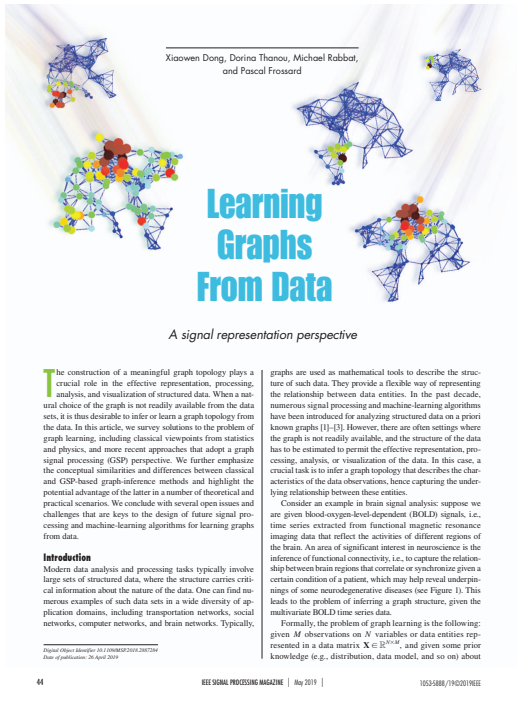


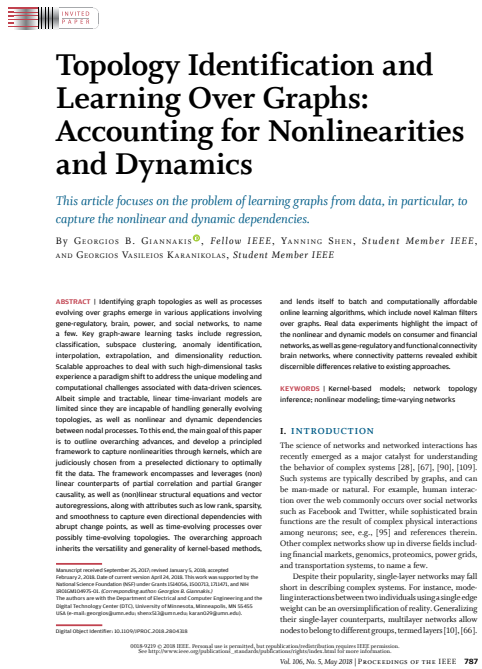
- Most GSP works: how known graph $\mathbf{S}$ affects signals and filters
- Here, reverse path: how to use GSP to infer the graph topology?
  - Gaussian graphical models [Egilmez et al'16], [Rabbat'17], ...
  - Smooth signals [Dong et al'15], [Kalofolias'16], [Sardellitti et al'17], ...
  - Graph filtering 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





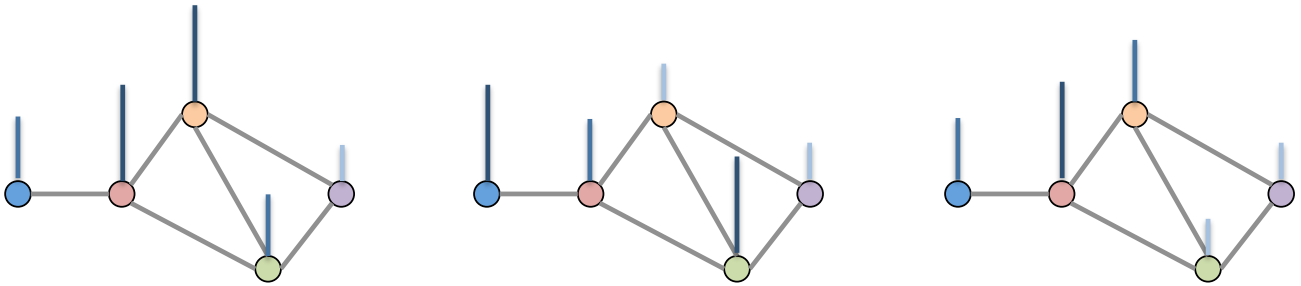


- IEEE Trans. on Signal and Information Processing over Networks
  - Special issue on Network Topology Inference (Jan. 2020)

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

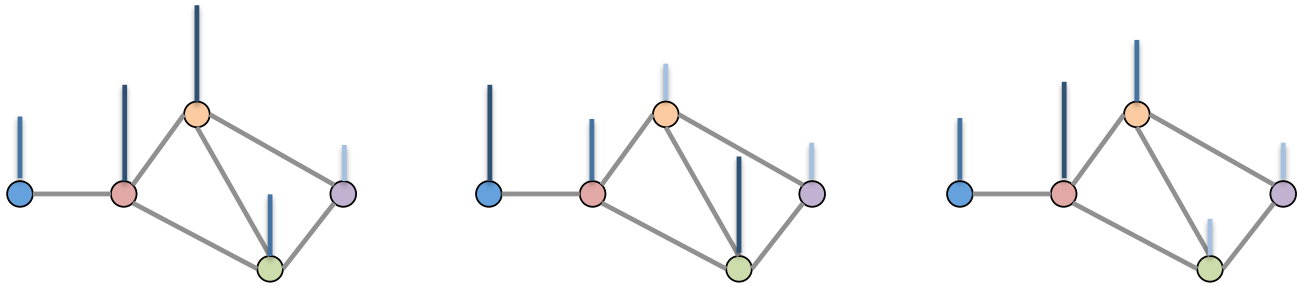


$\mathbf{y}_1$ $\mathbf{y}_2$ $\mathbf{y}_3$

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



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

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 ($L \leq N$)

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

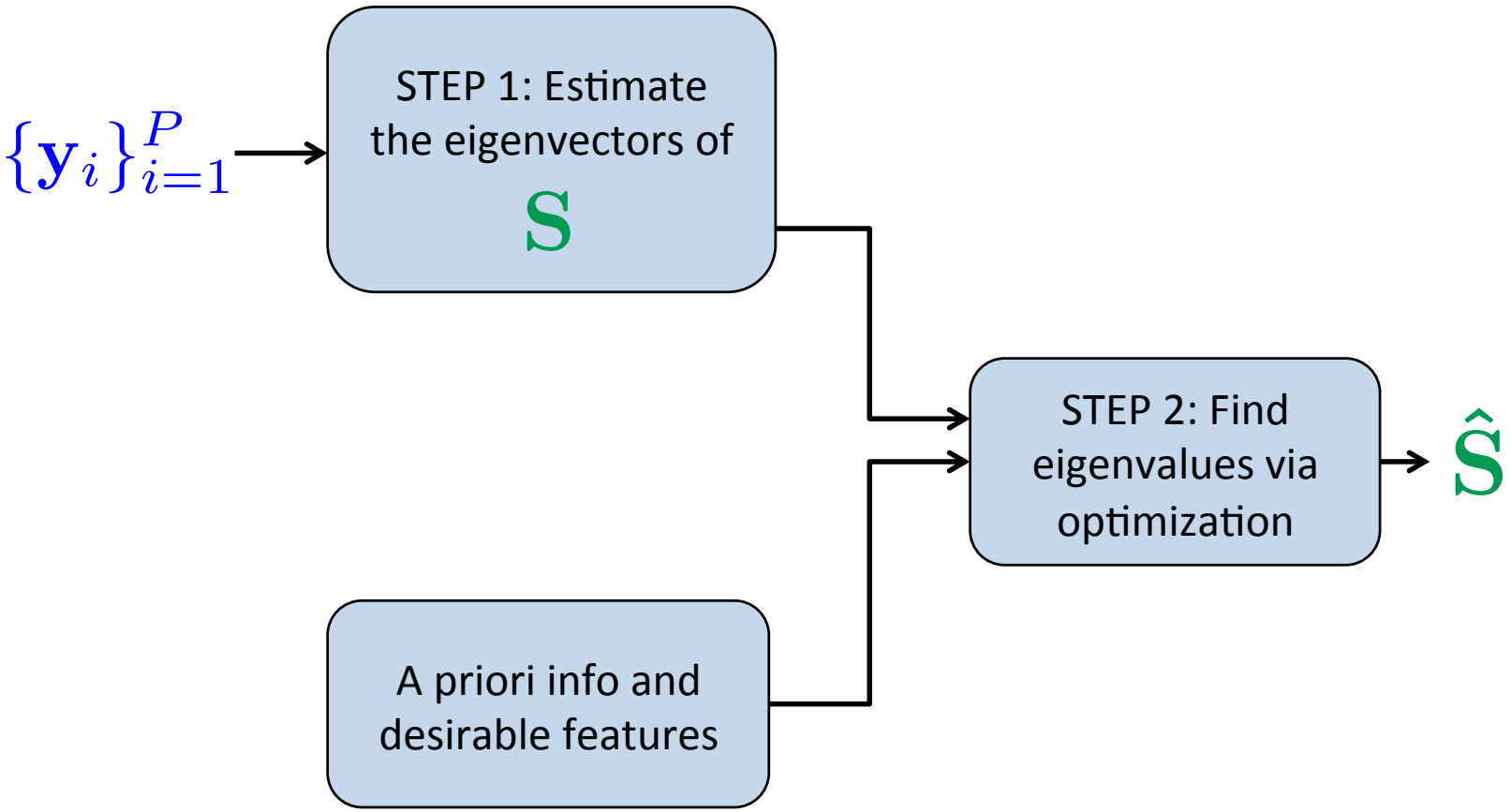
  $\Rightarrow$ Graph filter $\mathbf{H}$ is shift invariant [Sandryhaila-Moura'13]

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

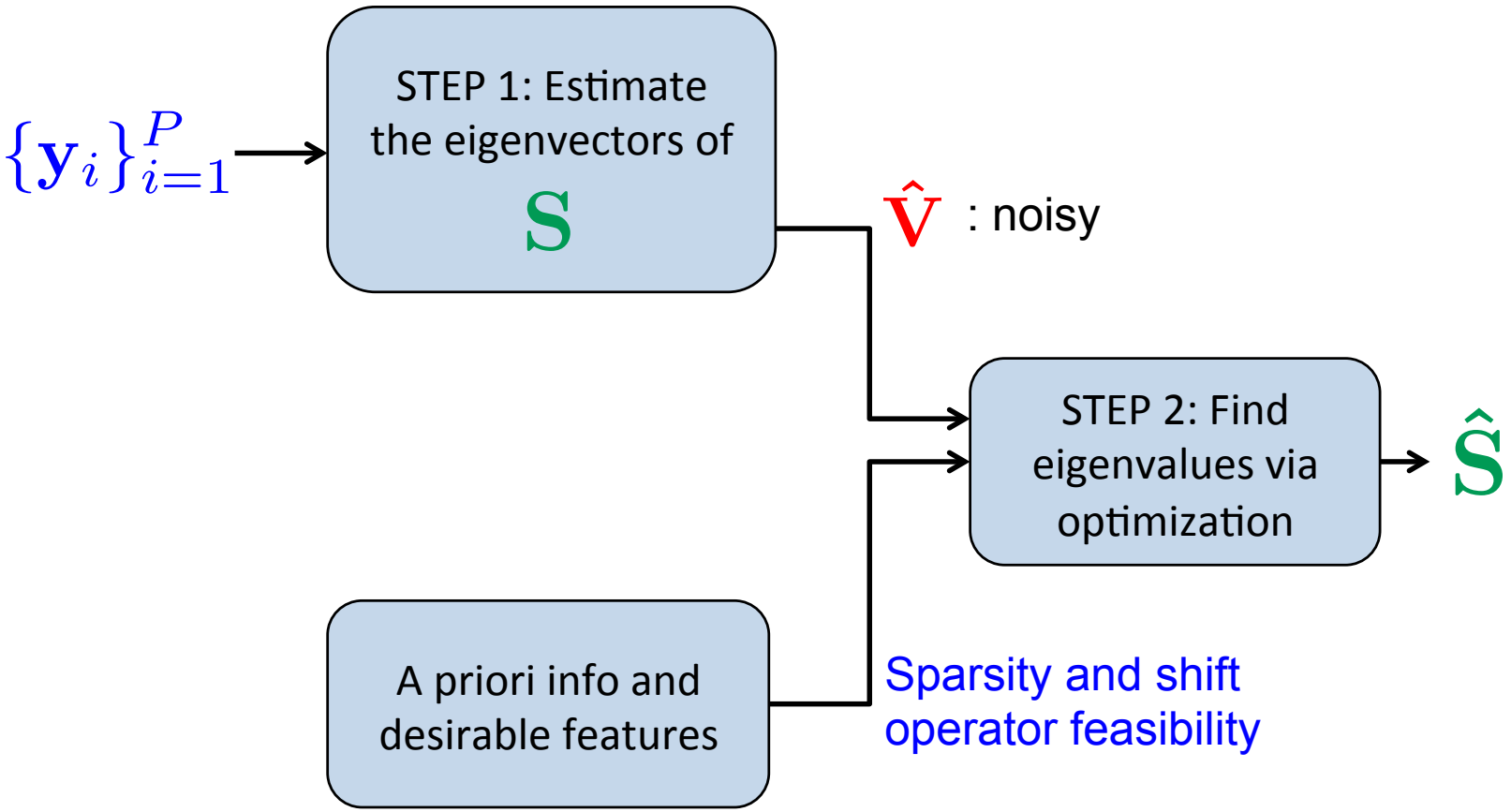
- **Goal**: estimate undirected network $\mathbf{S}$ from signal realizations $\{\mathbf{y}_i\}_{i=1}^P$

  $\Rightarrow$ 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

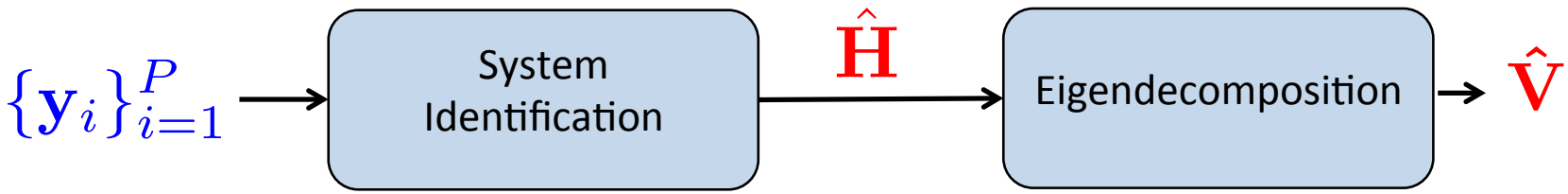
$$\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}$$

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

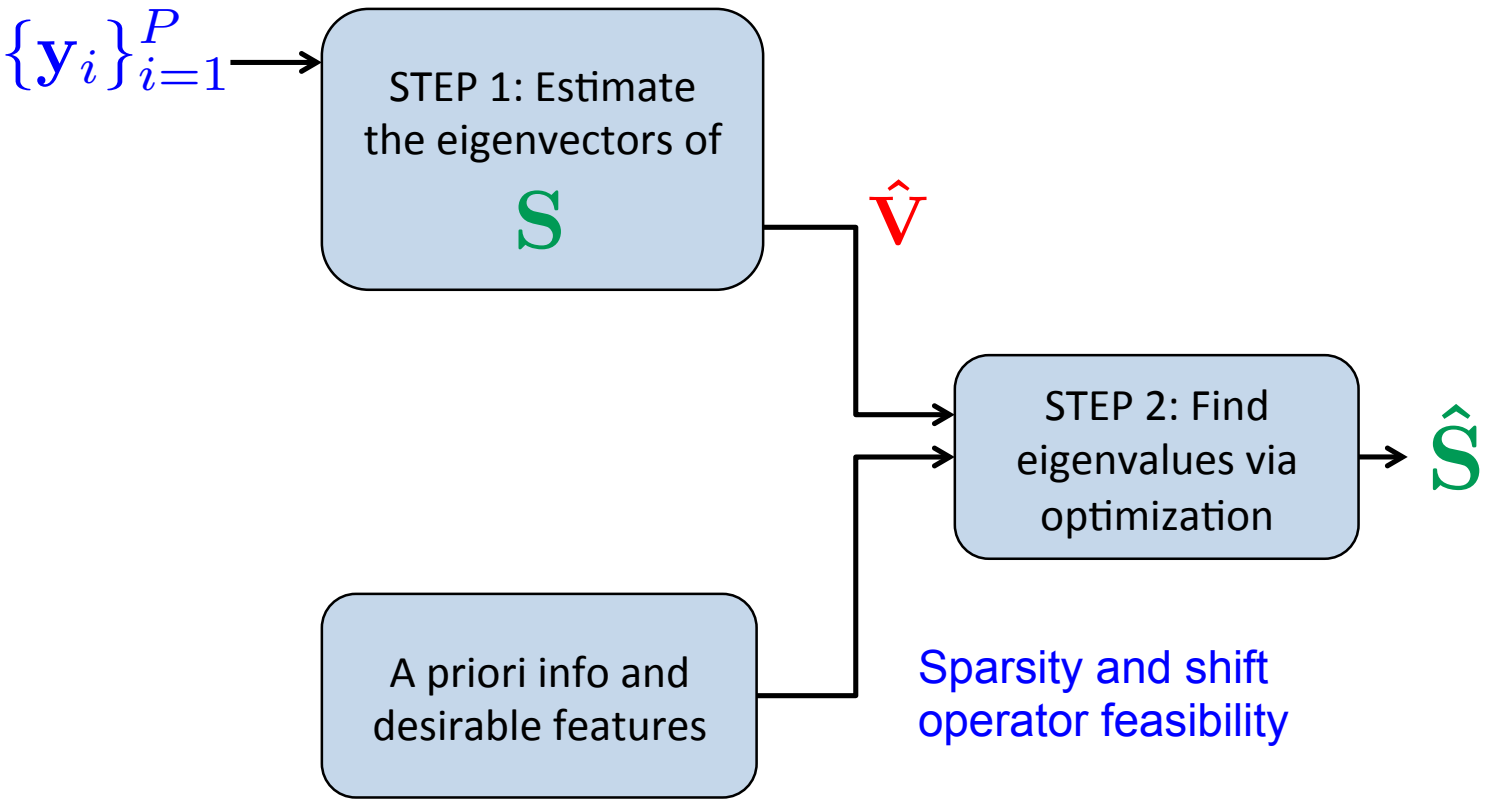
$$\mathbf{C}_y = \mathbf{H}\mathbf{C}_x\mathbf{H}$$

- 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

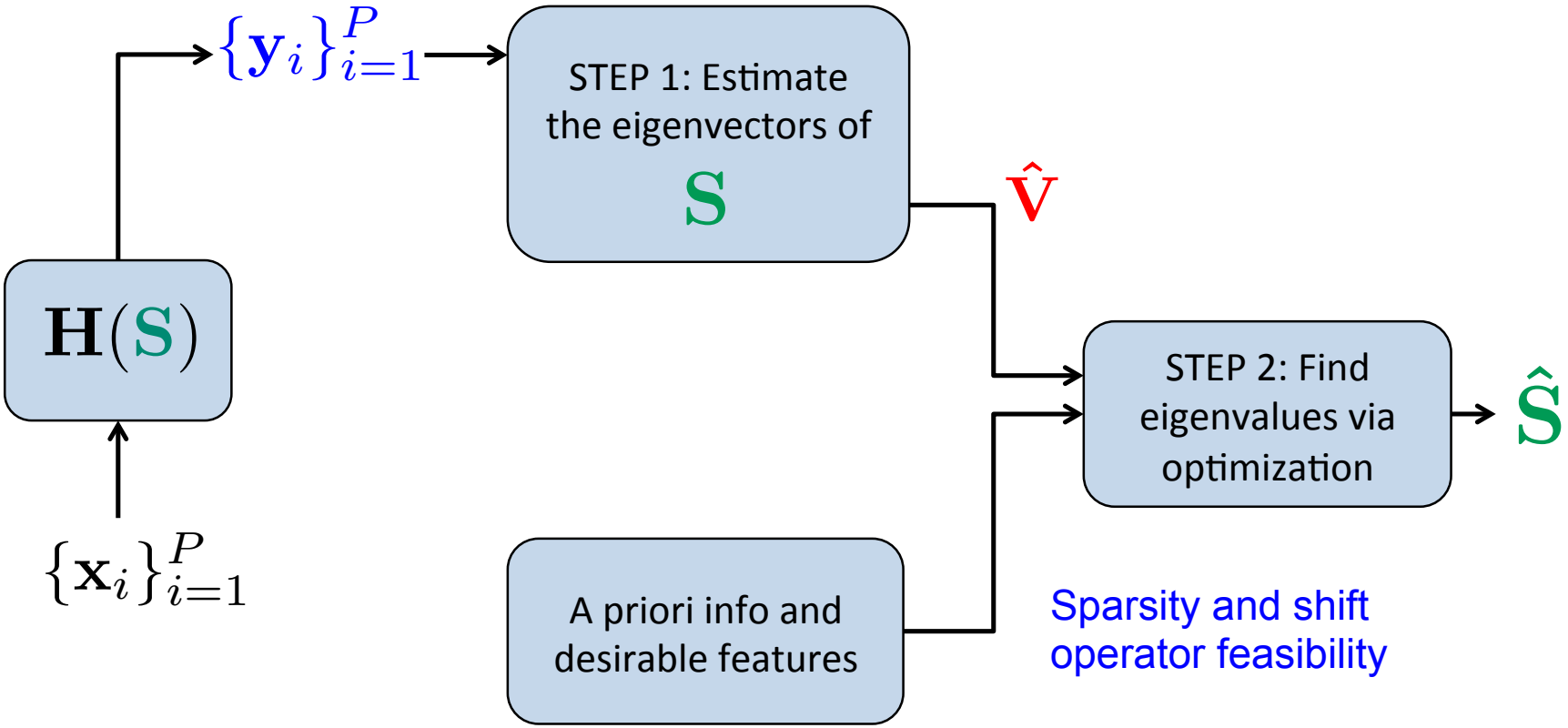


- I won't go into details on how to do this Sytem ID step

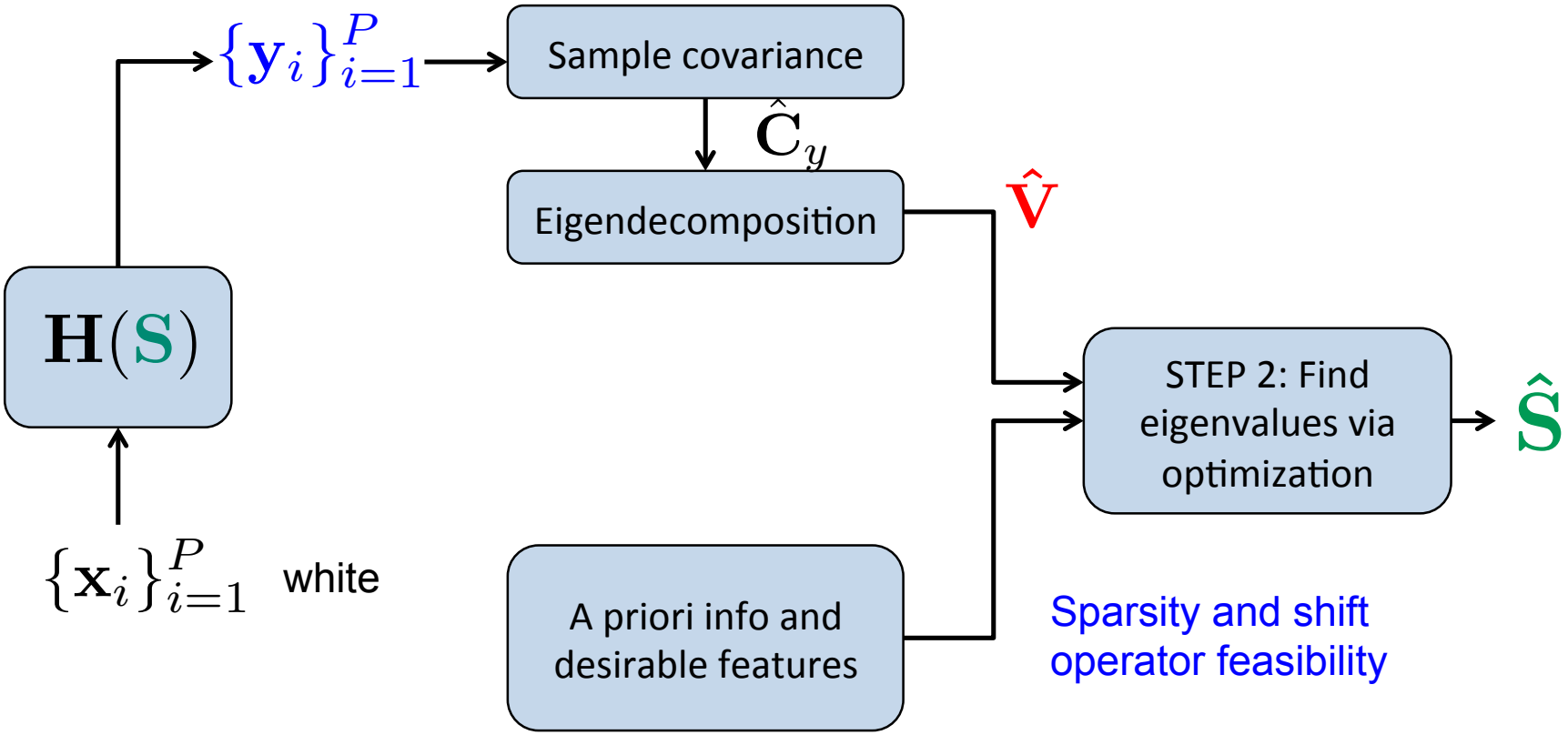
Summary of Step 1



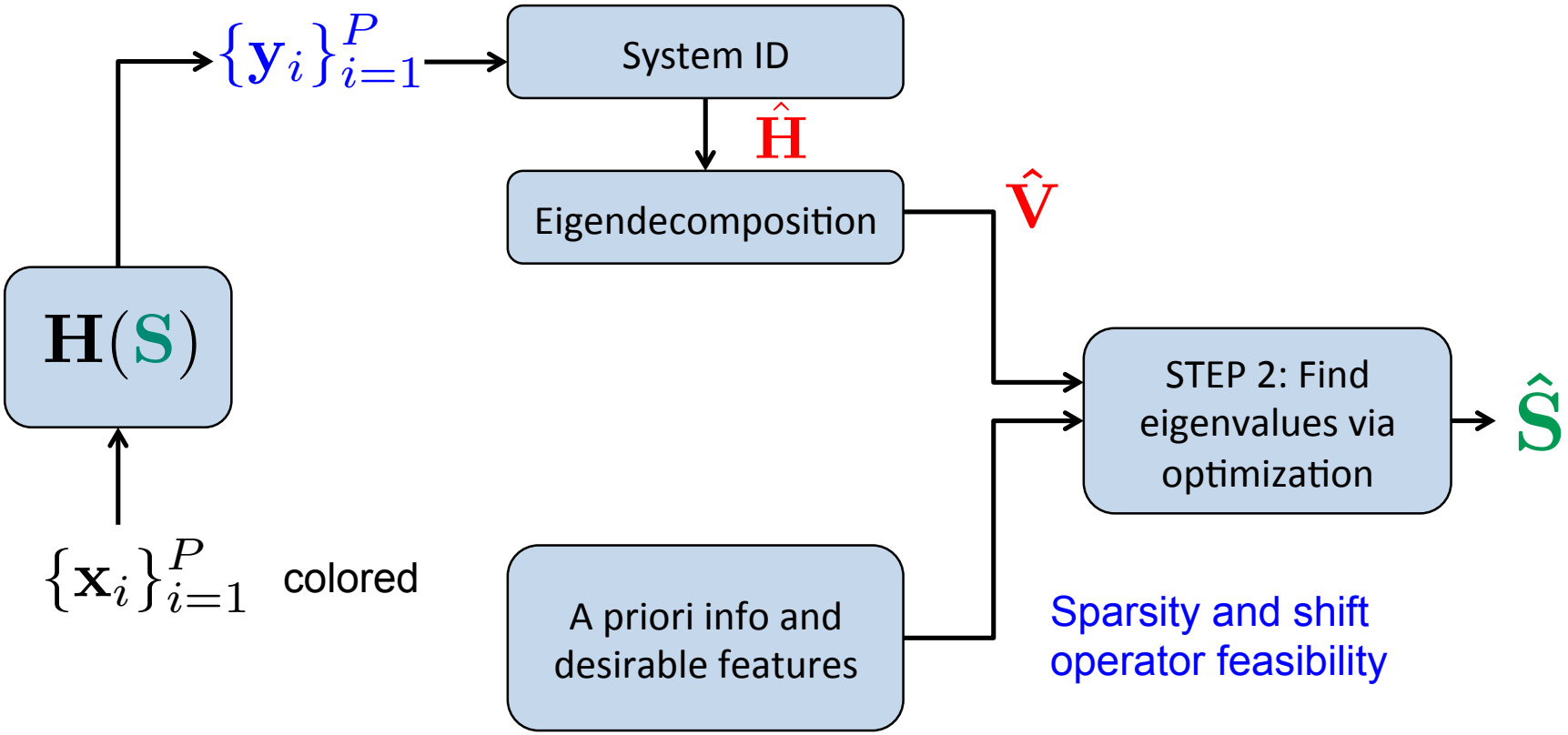
Summary of Step 1



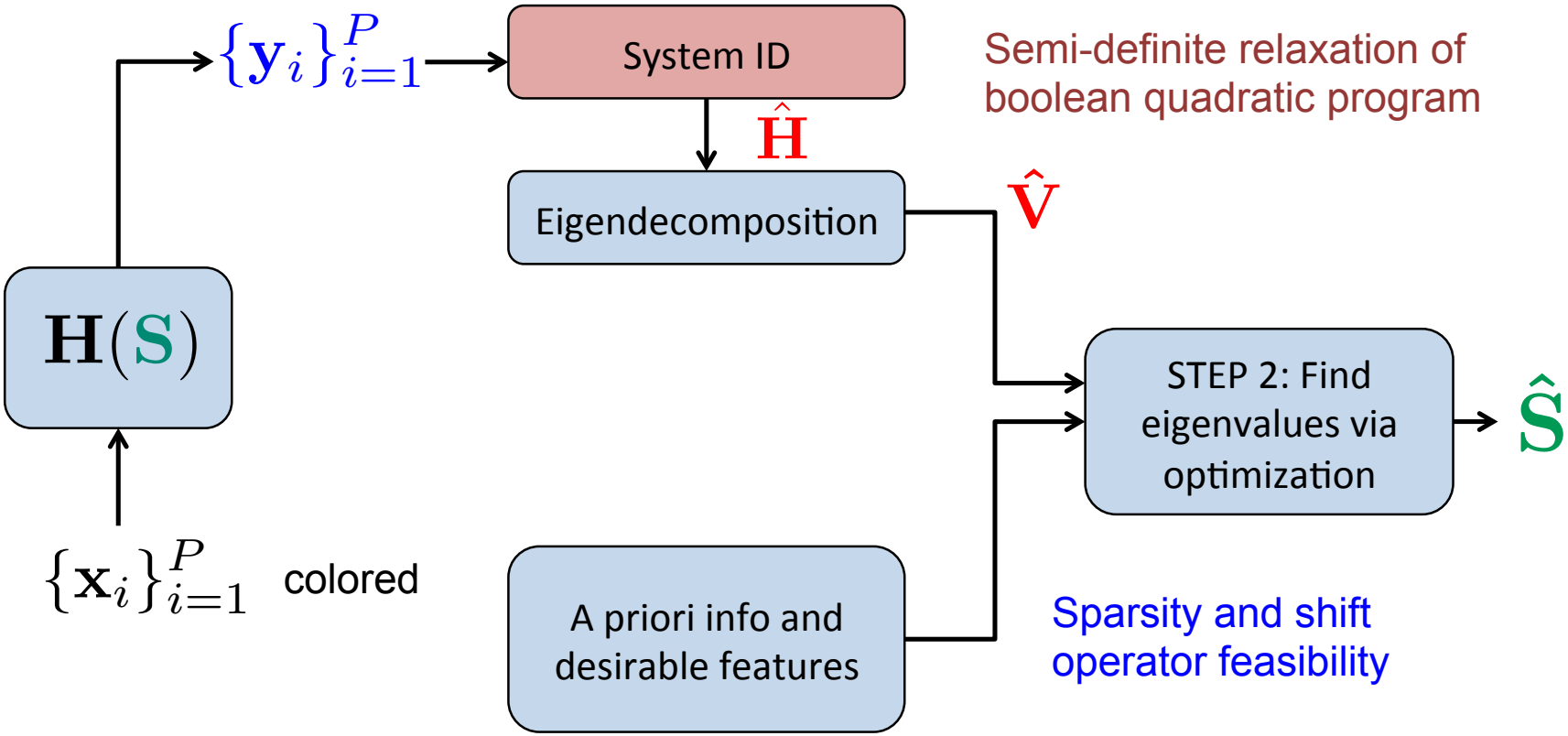
Summary of Step 1



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

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

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

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

- 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

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

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

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

  $\Rightarrow$ 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 $\mathbf{R} := [(\mathbf{I} - \mathbf{M}\mathbf{M}^\dagger)_{\mathcal{D}^c},\ \ \mathbf{e}_1 \otimes \mathbf{1}_{N-1}]$

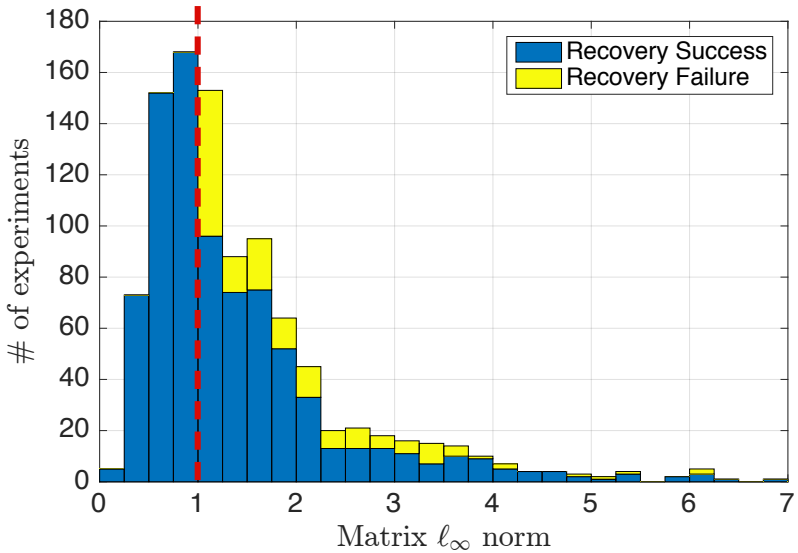
Theorem: $\mathbf{S}_1^* = \mathbf{S}_0^*$ if the two following conditions are satisfied
1) $\mathrm{rank}(\mathbf{R}_{\mathcal{K}}) = |\mathcal{K}|$; and
2) There exists a constant $\delta > 0$ such that

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

- 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

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

- **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

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

- **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 $\|\hat{\mathbf{R}}^T\mathbf{t} - \mathbf{b}\|_2 \leq \epsilon$
- 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 $\mathcal{K}$

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 $\mathbf{V}_K = [v_1, \ldots, v_K]$

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

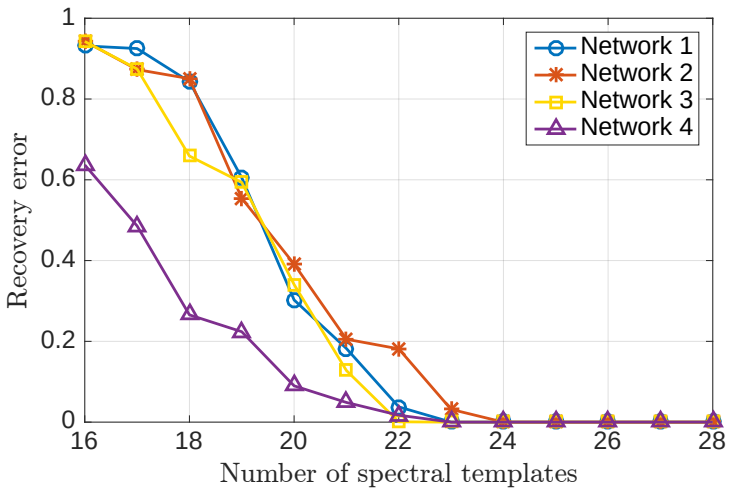
- **Q:** How does the (partial) knowledge of $\mathbf{V}_K$ affect the recovery?
- Define $\mathbf{P} := [\mathbf{P}_1, \ \mathbf{P}_2]$ in terms of $\mathbf{V}_K$, and $\boldsymbol{\Upsilon} := [\mathbf{I}_{N^2}, \mathbf{0}_{N^2 \times N^2}]$
  $\Rightarrow$ Reformulate problem as $\min_{\mathbf{t}} \|\boldsymbol{\Upsilon}\mathbf{t}\|_1$ s.to $\mathbf{P}^T\mathbf{t} = \mathbf{b}$

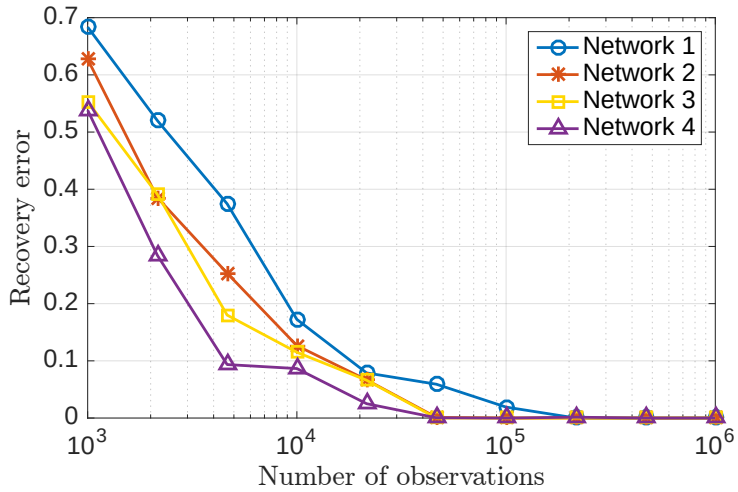
Theorem: $\mathbf{S}^* = \mathbf{S}_0^*$ if the two following conditions are satisfied
1) $\text{rank}([\mathbf{P}_1{}_{\mathcal{K}}^T, \mathbf{P}_2{}^T]) = |\mathcal{K}| + N^2$; and
2) There exists a constant $\delta > 0$ such that

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

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

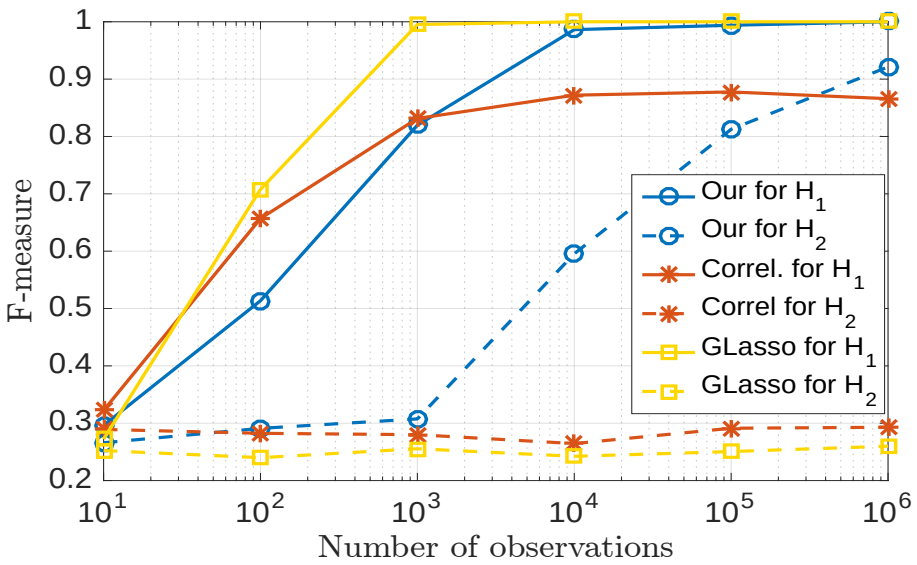




- Error (left) decreases with increasing nr. of spectral templates
- Error (right) decreases with increasing number of observed signals

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

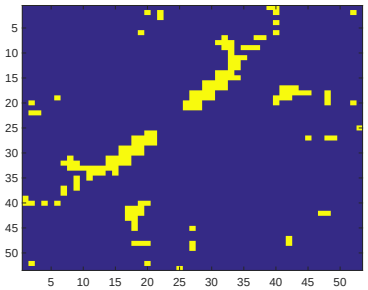
- For general diffusion filters $\mathbf{H}_2$ spectral templates still work fine

Inferring the structure of a protein

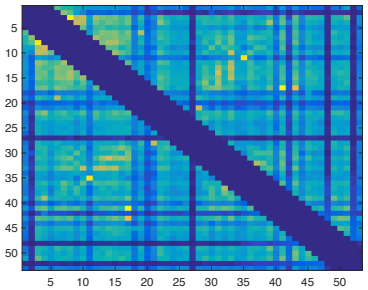
- Our method can be used to sparsify a given network
  - $\Rightarrow$ Keep direct and important edges or relations
  - $\Rightarrow$ Discard indirect relations that can be explained by direct ones
- Use eigenvectors $\hat{\mathbf{V}}$ of given network as noisy eigenvectors of $\mathbf{S}$

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

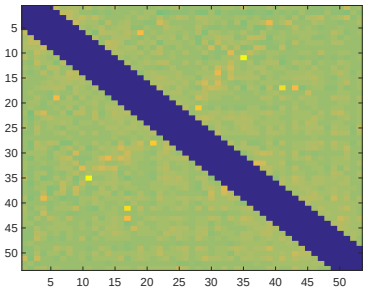
- $\Rightarrow$ Use mutual information of amino-acid covariation as input



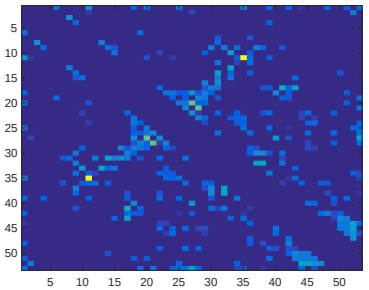
Ground truth



Mutual info.



Network deconv.



Our approach

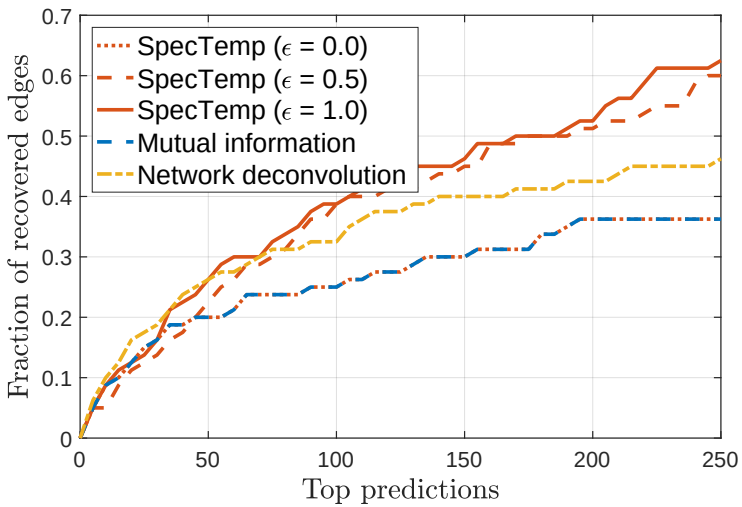
- Network deconvolution assumes a specific filter model [Feizi13]
  - $\Rightarrow$ We achieve better performance by being agnostic to this

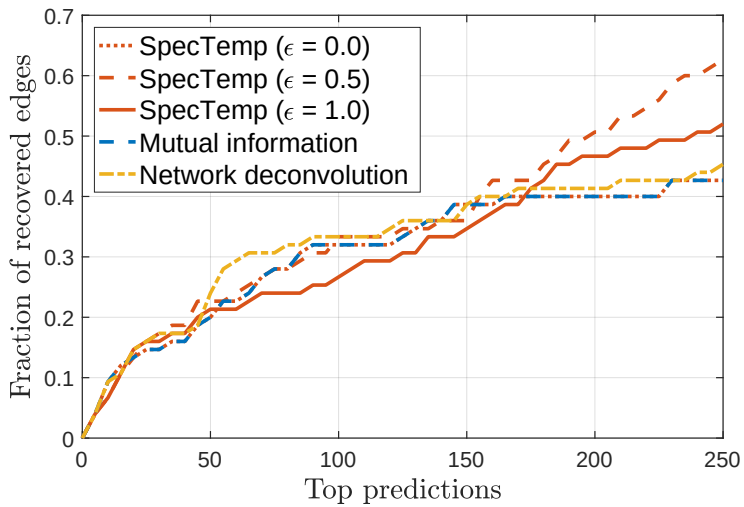
Sensitivity of recovered edges

- 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





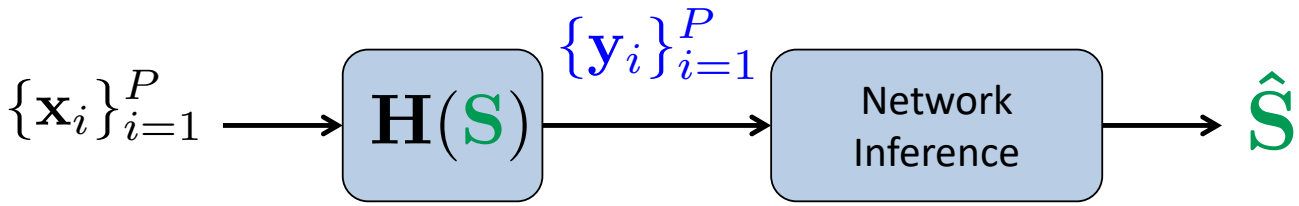
Summary

- 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

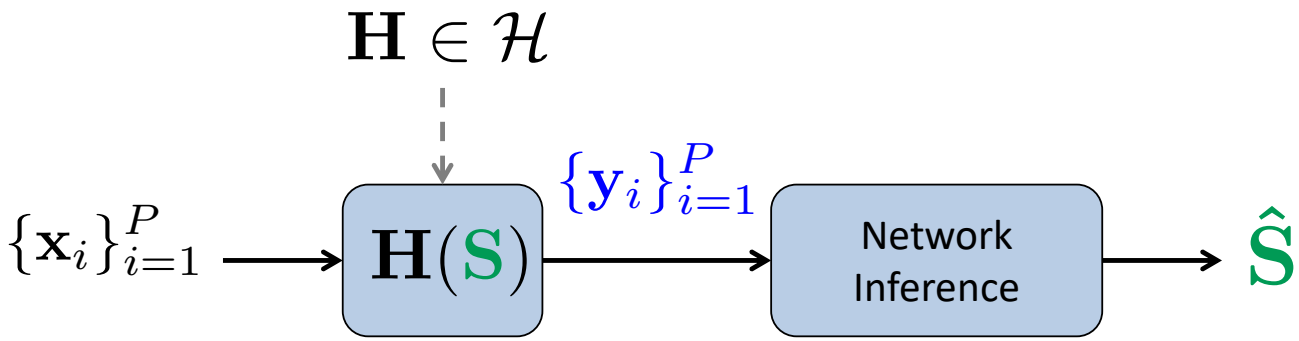
Summary

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

A rich framework for network inference

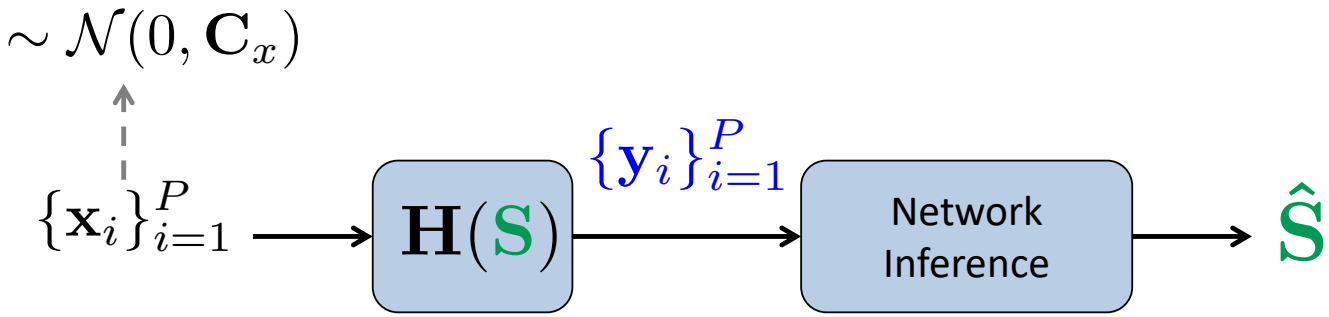


A rich framework for network inference



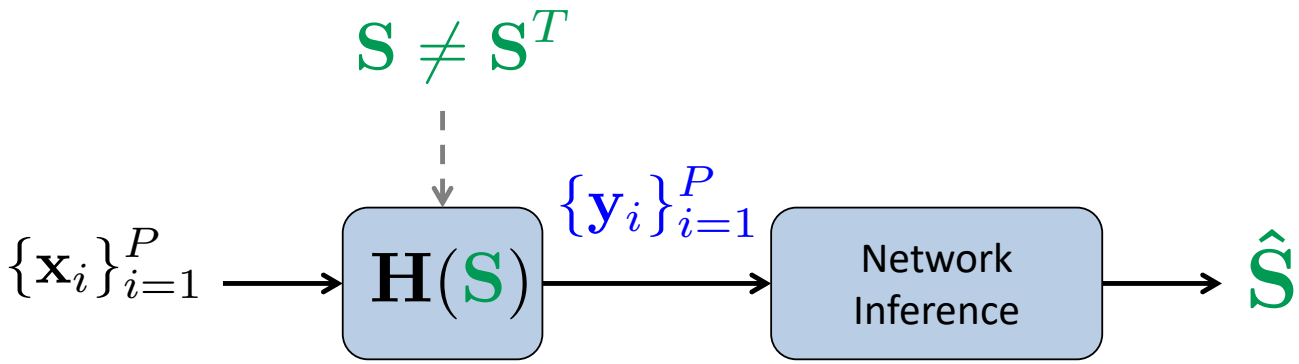
- Prior knowledge on the filter class [Segarra et al'17]

A rich framework for network inference



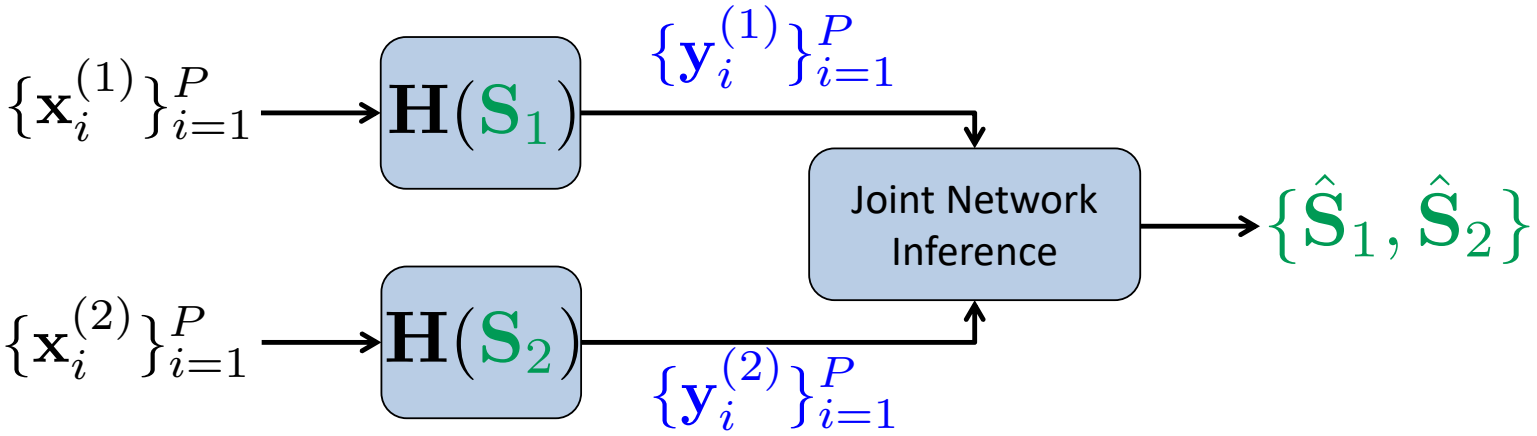
- Prior knowledge on the filter class [Segarra et al'17]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]

A rich framework for network inference



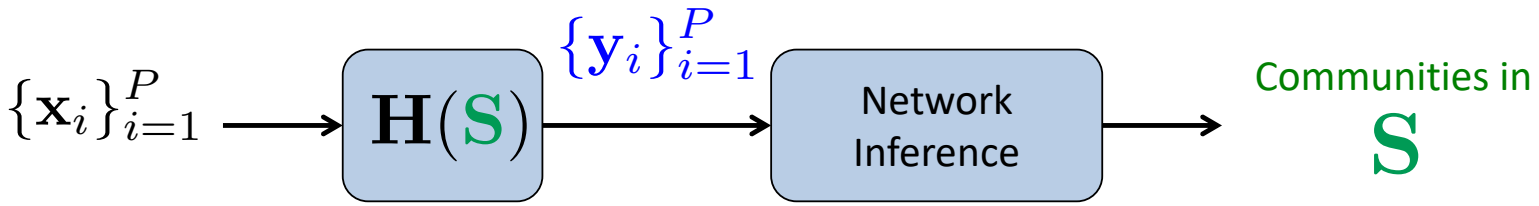
- Prior knowledge on the filter class [Segarra et al'17]
- Colored inputs to the diffusion process [Shafipour et al'17, '19]
- Inference for directed graphs [Shafipour et al'18]

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]

Wrapping up

Motivation and preliminaries

Part I: Fundamentals
- Graphs 101
- Graph signals and the shift operator
- Graph Fourier Transform (GFT)
- Graph filters and network processes

Part II: Applications
- Network topology inference

Concluding remarks

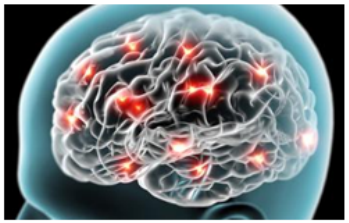
Concluding remarks

- Network science and big data pose new challenges
  - $\Rightarrow$ GSP can contribute to solve some of those challenges
  - $\Rightarrow$ Well suited for network (diffusion) processes
- Central elements in GSP: graph-shift operator and Fourier transform
- Graph filters: operate graph signals
  - $\Rightarrow$ Polynomials of the shift operator that can be implemented locally
- Network diffusion/percolations processes via graph filters
  - $\Rightarrow$ Successive/parallel combination of local linear dynamics
  - $\Rightarrow$ Possibly time-varying diffusion coefficients
  - $\Rightarrow$ Accurate to model certain setups
  - $\Rightarrow$ GSP yields insights on how those processes behave

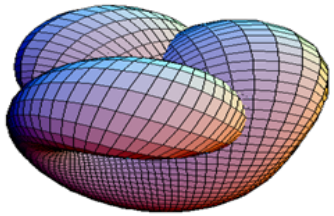
Concluding remarks

- GSP results can be applied to solve practical problems
  - $\Rightarrow$ Sampling, interpolation (network control)
  - $\Rightarrow$ Input and system ID (rumor ID)
  - $\Rightarrow$ Shift design (network topology ID)

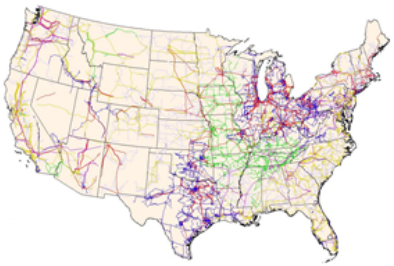
Interpolate a brain signal from local observations



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