Graph Signal Processing: Fundamentals and Applications to Diffusion Processes

A. G. Marques†, G. Mateos⋆, A. Ribeiro, and S. Segarra

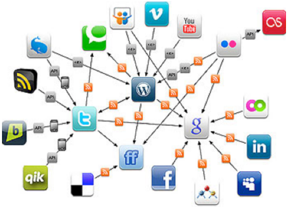
†King Juan Carlos University – antonio.garcia.marques@urjc.es
⋆University of Rochester – gmateosb@ece.rochester.edu

Thanks: Weiyu Huang and Geert Leus

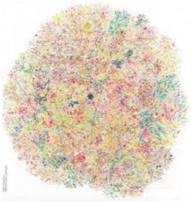
IEEE SAM Tutorial
Rio de Janeiro, Brazil
July 10, 2016

Network Science analytics

Online social media



Internet

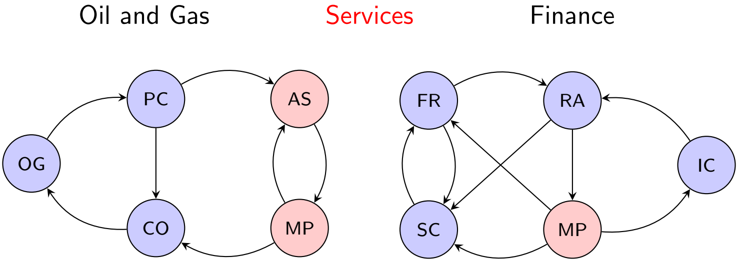


Clean energy and grid analytics

- Desiderata: Process, analyze and learn from network data [Kolaczyk'09]
- Network as graph $G = (\mathcal{V}, \mathcal{E})$: 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

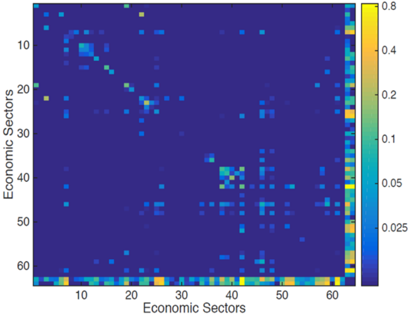
- Bureau of Economic Analysis of the U.S. Department of Commerce
- $\mathcal{E}$ = Output of sector i is an input to sector j (62 sectors in $\mathcal{V}$)



- Oil 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 $\mathcal{V}$)

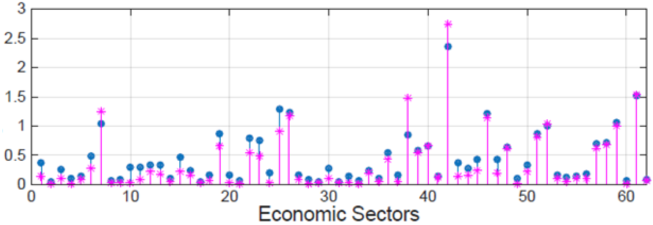


- A few sectors have widespread strong influence (services, finance, energy)
- Some sectors have strong indirect influences (oil)
- The heavy last row is final consumption

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

Disaggregated GDP of the United States

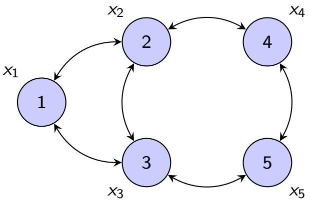
- Signal $\mathbf{x}$ = output per sector = disaggregated GDP
  - $\Rightarrow$ Network structure used to, e.g., reduce GDP estimation noise



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

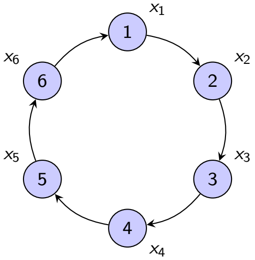


- **As.:** Signal properties related to topology of G (locality, smoothness)
  $\Rightarrow$ Algorithms that fruitfully leverage this relational structure
- Q: Why do we expect the graph structure to be useful in processing $\mathbf{x}$?

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 $\Rightarrow \tilde{\mathbf{x}} = \mathbf{F}^H \mathbf{x}$ $\left(F_{kn} = \frac{e^{j2\pi kn/N}}{\sqrt{N}} \right)$

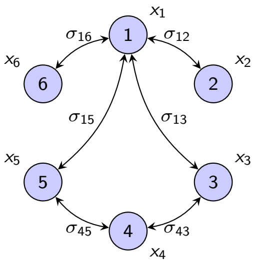
- Fourier transform $\Rightarrow$ Projection on eigenvector space of cycle

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

- 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 dependencies of Gaussian $\mathbf{x}$



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

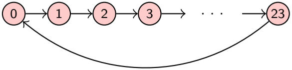
- PCA transform $\Rightarrow$ Projection on eigenvector space of (inverse) covariance

- Q: Can we extend these principles to general graphs and signals?

Graphs 101

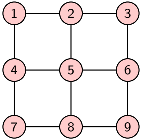
- Formally, a graph G (or a network) is a triplet $(\mathcal{V}, \mathcal{E}, W)$
- $\mathcal{V} = \{1, 2, \ldots, N\}$ is a finite set of N nodes or vertices
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is a set of edges defined as ordered pairs (n, m)
  - Write $\mathcal{N}(n) = \{m \in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

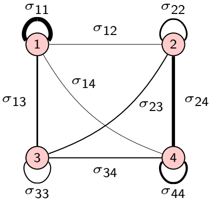
Graphs – examples



- 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

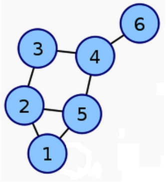
- Algebraic graph theory: matrices associated with a graph G
  - $\Rightarrow$ Adjacency $\mathbf{A}$ and Laplacian $\mathbf{L}$ matrices
  - $\Rightarrow$ Spectral graph theory: properties of G using spectrum of $\mathbf{A}$ or $\mathbf{L}$
- Given $G = (\mathcal{V}, \mathcal{E}, W)$, the adjacency 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: $\mathbf{d} = \mathbf{A}\mathbf{1} \rightarrow$ 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



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

Laplacian of a graph

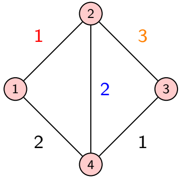
- Given undirected G with $\mathbf{A}$ and $\mathbf{D}$, the Laplacian matrix $\mathbf{L} \in \mathbb{R}^{N \times N}$ is

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

$\Rightarrow$ 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 $\mathbf{L}$
- $\mathbf{L}$ 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

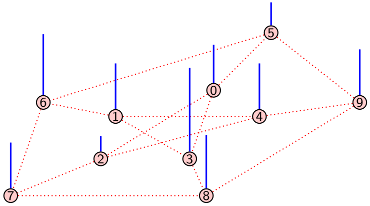
Part I: Fundamentals

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 $\mathbf{L}$)



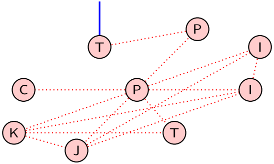
$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

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

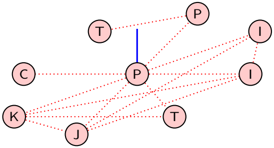
Graph signals – Genetic profiles

- Graphs representing gene-gene interactions
  - ⇒ Each node denotes a single gene (loosely speaking)
  - ⇒ Connected if their coded proteins participate in same metabolism
- Genetic profiles for each patient can be considered as a graph signal
  - ⇒ 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



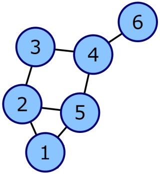
Sample patient 2 with subtype 1

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

- 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
- 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

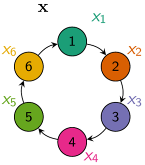


$$\mathbf{S} = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{23} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix}$$

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

Relevance of the graph-shift operator

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

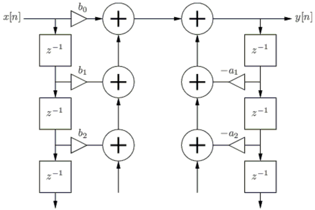


⇨ Set $\mathbf{S} = \mathbf{A_{dc}}$

What is $\mathbf{Sx}$? ⇨

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

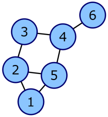
- **S** will be building block for GSP algorithms (More soon)
  - ⇒ Same is true in the time domain (filters and delay)



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 $(j, i) \in \mathcal{E}$



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

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

  $\Rightarrow$ Like powers of **A**: neighborhoods

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

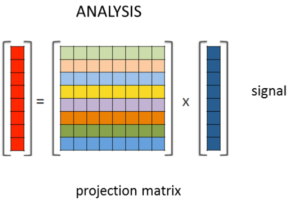
$$[\mathbf{S}^2]_{3,5} = S_{3,2}S_{2,5} + S_{3,4}S_{4,5}$$

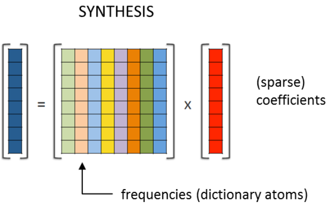
$$\mathbf{S}^2 = \begin{pmatrix} S_{11} & S_{12} & 0 & 0 & S_{15} & 0 \\ S_{21} & S_{22} & S_{23} & 0 & S_{25} & 0 \\ 0 & S_{32} & S_{33} & S_{34} & 0 & 0 \\ 0 & 0 & S_{43} & S_{44} & S_{45} & S_{46} \\ S_{51} & S_{52} & 0 & S_{54} & S_{55} & 0 \\ 0 & 0 & 0 & S_{64} & 0 & S_{66} \end{pmatrix} \begin{pmatrix}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

Graph Fourier Transform

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 $\mathbf{S} = \mathbf{V}\Lambda\mathbf{V}^{-1}$ 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

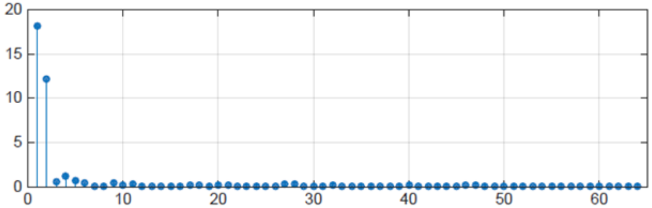
$$\mathbf{x} = \mathbf{V}\tilde{\mathbf{x}}$$

  $\Rightarrow$ 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



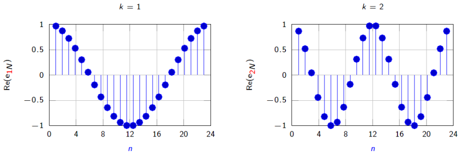
- GFT transform characterized by a few coefficients

  $\Rightarrow$ Notion of bandlimitedness: $\mathbf{x} = \sum_{k=1}^{K} \tilde{x}_k \mathbf{v}_k$

  $\Rightarrow$ Sampling, compression, filtering, pattern recognition

Eigenvalues as frequencies

- 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 $\lambda_k = e^{-j\frac{2\pi}{N}k}$
  - $\Rightarrow$ λ_k can be viewed as frequencies!!
- In time, well-defined relation between frequency and variation
  - $\Rightarrow$ Higher k $\Rightarrow$ higher 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 in general, but if $\mathbf{S} = \mathbf{L}$ there are interpretations for λ_k
  - $\Rightarrow$ $\{\lambda_k\}_{k=1}^N$ will be very important when analyzing graph filters

Interpretation of the Laplacian

- Consider a graph G, let $\mathbf{x}$ be a signal on G, and set $\mathbf{S} = \mathbf{L}$
  - $\Rightarrow$ $\mathbf{y} = \mathbf{S}\mathbf{x}$ is now $\mathbf{y} = \mathbf{L}\mathbf{x}$ $\Rightarrow$ $y_i = \sum_{j \in \mathcal{N}(i)} w_{ij}(x_i - x_j)$
  - $\Rightarrow$ j-th term is large if x_j is very different from neighboring x_i
  - $\Rightarrow$ y_i measures difference of x_i relative to its neighborhood

- We can also define the quadratic form $\mathbf{x}^T\mathbf{S}\mathbf{x}$

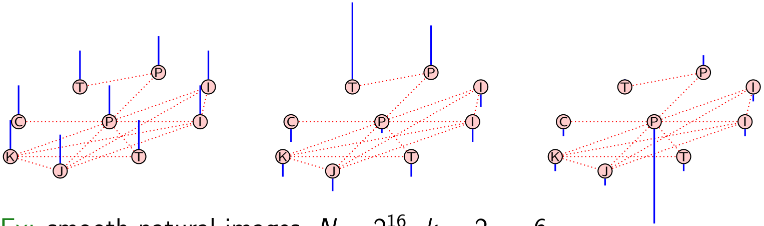
$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} w_{ij}(x_i - x_j)^2$$

  - $\Rightarrow$ $\mathbf{x}^T\mathbf{L}\mathbf{x}$ quantifies the (aggregated) local variation of signal $\mathbf{x}$
  - $\Rightarrow$ Natural measure of signal smoothness w.r.t. G

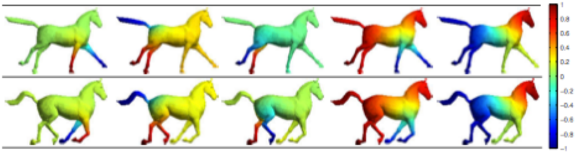
- Q: Interpretation of frequencies $\{\lambda_k\}_{k=1}^N$ when $\mathbf{S} = \mathbf{L}$?
  - $\Rightarrow$ If $\mathbf{x} = \mathbf{v}_k$, we get $\mathbf{x}^T\mathbf{L}\mathbf{x} = \lambda_k$ $\Rightarrow$ local variation of $\mathbf{v}_k$
  - $\Rightarrow$ Frequencies account for local variation, 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, $N = 2^{16}$, $k = 2, ..., 6$



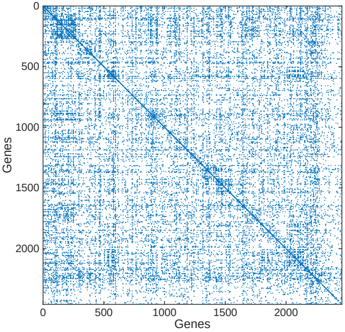
Application: Cancer subtype classification

- Patients diagnosed with same disease exhibit different behaviors
- Each patient has a genetic profile describing gene mutations
- Would be beneficial to infer phenotypes from genotypes
  $\Rightarrow$ Targeted treatments, more suitable suggestions, etc.
- Traditional approaches consider different genes to be independent
  $\Rightarrow$ Not ideal, as different genes may affect same metabolism
- Alternatively, consider genetic network
  $\Rightarrow$ Genetic profiles become graph signals on genetic network
  $\Rightarrow$ We will see how this consideration improves subtype classification

Genetic network

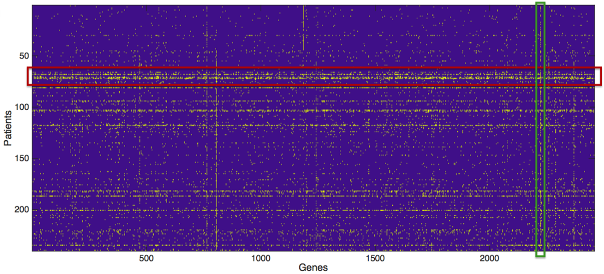
- Undirected and unweighted gene-to-gene interaction graph
  - 2458 nodes are genes in human DNA related to breast cancer
  - An edge between two genes represents interaction
    ⇒ Coded proteins participate in the same metabolic process

- Adjacency matrix of the gene-interaction network



Genetic profiles

- Genetic profile of 240 women with breast cancer
  - ⇒ 44 with serous subtype and 196 with endometrioid subtype
  - ⇒ Patient i has an associated profile $\mathbf{x}_i \in \{0,1\}^{2458}$
- Mutations are very varied across patients
  - ⇒ Some patients present a lot of mutations
  - ⇒ Some genes are consistently mutated across patients



- Q: Can we use genetic profiles to classify patients across subtypes?

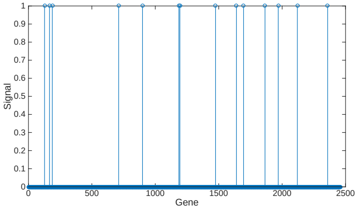
Improving k-nearest neighbor classification

- Distance between genetic profiles $\Rightarrow d(i,j) = \|\mathbf{x}_i - \mathbf{x}_j\|_2$
  $\Rightarrow$ N-fold cross-validation error from k-NN classification

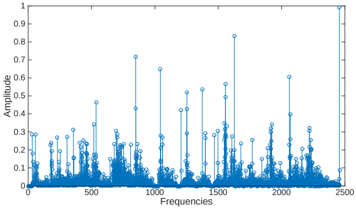
$$k = 3 \Rightarrow 13.3\%, \qquad k = 5 \Rightarrow 12.9\%, \qquad k = 7 \Rightarrow 14.6\%$$

- Q: Can we do any better using graph signal processing?
- Each genetic profile $\mathbf{x}_i$ is a graph signal on the genetic network
  $\Rightarrow$ Look at the frequency components $\tilde{\mathbf{x}}_i$ using the GFT
  $\Rightarrow$ Use as shift operator $\mathbf{S}$ the Laplacian of the genetic network

Example of signal $\mathbf{x}_i$



Frequency representation $\tilde{\mathbf{x}}_i$

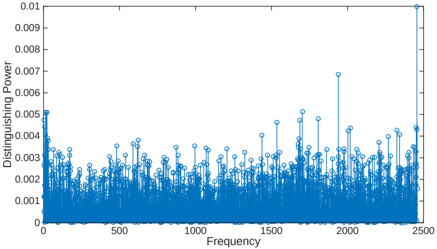


Distinguishing Power

- Define the distinguishing power of frequency $\mathbf{v}_k$ as

$$DP(\mathbf{v}_k) = \left| \frac{\sum_{i:y_i=1} \tilde{\mathbf{x}}_i(k)}{\sum_i \mathbf{1}\{y_i = 1\}} - \frac{\sum_{i:y_i=2} \tilde{\mathbf{x}}_i(k)}{\sum_i \mathbf{1}\{y_i = 2\}} \right| / \sum_i |\tilde{\mathbf{x}}_i(k)|,$$

- Normalized difference between the mean GFT coefficient for $\mathbf{v}_k$
  $\Rightarrow$ Among patients with serous and endometrioid subtypes
- Distinguishing power is not equal across frequencies



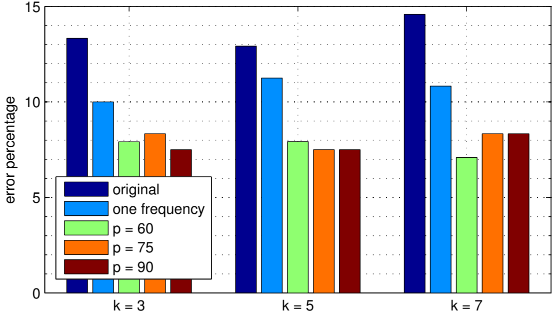
- The distinguishing power defined is one of many proper heuristics

Increasing accuracy by selecting the best frequencies

- Keep information in frequencies with higher distinguishing power
  $\Rightarrow$ Filter, i.e., multiply $\tilde{\mathbf{x}}_i$ by $\text{diag}(\tilde{\mathbf{h}}^p)$ where

$$[\tilde{\mathbf{h}}^p]_k = \begin{cases} 1, & \text{if } DP(\mathbf{v}_k) \geq p\text{-th percentile of } DP \\ 0, & \text{otherwise} \end{cases}$$

- Then perform inverse GFT to get the graph signal $\hat{\mathbf{x}}_i$

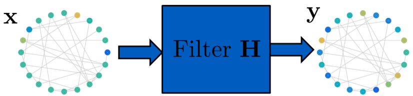


Graph filters and network processes

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

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 $\mathbf{y} = \mathbf{H}\mathbf{x}$

  ⇒ 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

  ⇒ $\mathbf{x}^{(l)}$ can be found locally if $\mathbf{x}^{(l-1)}$ is known

  ⇒ The output of the filter can be found in L local steps

- A graph filter represents a linear transformation that

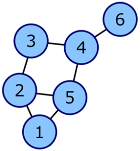
  ⇒ Accounts for local structure of the graph

  ⇒ Can be implemented distributedly in L steps

  ⇒ Only requires info in L-neighborhood [Shuman13, Sandyhaila14]

An example of a graph filter

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



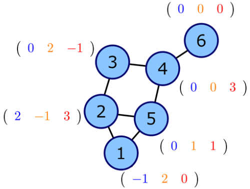
$$\mathbf{S}=\mathbf{A}=\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

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

$$\Downarrow$$

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

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

- Recalling that $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, we may write

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

- 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 frequency domain $\tilde{\mathbf{x}}$
  - $\Rightarrow$ $\tilde{\mathbf{H}} := \sum_{l=0}^{L} h_l \mathbf{\Lambda}^l$ modifies the frequency coefficients 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 $\tilde{\mathbf{H}} =: diag(\tilde{\mathbf{h}})$
  - $\Rightarrow$ $\tilde{\mathbf{h}}$ 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

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

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

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

  - $\Rightarrow$ Since $\mathbf{\Psi}$ is Vandermonde, invertible as long as $\lambda_k \neq \lambda_{k'}$ for $k \neq k'$

More on the frequency response

- Since $\mathbf{h} = \mathbf{\Psi}^{-1}\tilde{\mathbf{h}} \Rightarrow$ 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 $\mathbf{h} = \mathbf{\Psi}\tilde{\mathbf{h}} \Rightarrow$ 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 $\lambda_k = e^{-\mathrm{j}\frac{2\pi}{N}(k-1)}$

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

  $\Rightarrow$ 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

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

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

System identification and impulse response

- Suppose that our goal is to find $\mathbf{h}$ given $\mathbf{x}$ and $\mathbf{y}$
  $\Rightarrow$ Using the previous expressions

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

- In time, if we set $\mathbf{x} = [1, 0, ..., 0]^T = \mathbf{e}_1$ (i.e., $\tilde{\mathbf{x}} = \mathbf{1}$), we have
  $\Rightarrow \mathbf{h} = \mathbf{F}\text{diag}^{-1}(\mathbf{1})\mathbf{F}^H\mathbf{y} = \mathbf{y} \rightarrow \mathbf{h}$ is the impulse response
- In the graph domain
- If we set $\mathbf{x} = \mathbf{e}_i$, then $\mathbf{h} = \mathbf{\Psi}^{-1}\text{diag}^{-1}(\tilde{\mathbf{e}}_i)\mathbf{V}^{-1}\mathbf{y}$, where
  $\Rightarrow \tilde{\mathbf{e}}_i := \mathbf{V}^{-1}\mathbf{e}_i \equiv$ how strongly node i expresses each of the freqs.
  $\Rightarrow$ Problem if $\tilde{\mathbf{e}}_i$ has zero entries
- Alternatively we can get $\tilde{\mathbf{x}} = \mathbf{1}$ by setting $\mathbf{x} = \mathbf{V}\mathbf{1}$ and then
  $\Rightarrow \mathbf{h} = \mathbf{\Psi}^{-1}\text{diag}^{-1}(\tilde{\mathbf{x}})\mathbf{V}^{-1}\mathbf{y} = \mathbf{\Psi}^{-1}\mathbf{V}^{-1}\mathbf{y}$

Implementing graph filters: frequency or space

- Frequency or space?

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

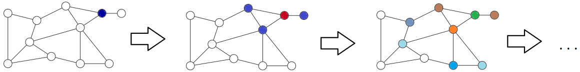
- In space: leverage the fact that $\mathbf{Sx}$ can be computed locally
  - ⇒ Signal $\mathbf{x}$ is percolated L times to find $\{\mathbf{x}^{(l)}\}_{l=0}^{L}$
  - ⇒ Every node finds its own y_i by computing $\sum_{l=0}^{L} h_l[\mathbf{x}^{(l)}]_i$
- Frequency implementation useful for processing if,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

Linear network processes via graph filters

- Consider a linear dynamics of the form

$$\mathbf{x}_t - \mathbf{x}_{t-1} = \alpha \mathbf{J} \mathbf{x}_{t-1} \;\Rightarrow\; \mathbf{x}_t = (\mathbf{I} - \alpha \mathbf{J}) \mathbf{x}_{t-1}$$

- If $\mathbf{x}$ is network process $\Rightarrow$ $[\mathbf{x}_t]_i$ depends only on $[\mathbf{x}_{t-1}]_j$, $j \in \mathcal{N}(i)$



$$[\mathbf{S}]_{ij} = [\mathbf{J}]_{ij} \Rightarrow \; \mathbf{x}_t = (\mathbf{I} - \alpha \mathbf{S}) \mathbf{x}_{t-1} \;\Rightarrow\; \mathbf{x}_t = (\mathbf{I} - \alpha \mathbf{S})^t \mathbf{x}_0$$

$\Rightarrow$ $\mathbf{x}_t = \mathbf{H} \mathbf{x}_0$, with $\mathbf{H}$ a polynomial of $\mathbf{S}$ $\Rightarrow$ linear graph filter

- If the system has memory $\Rightarrow$ output weighted sum of previous exchanges (opinion dynamics) $\Rightarrow$ still a polynomial of $\mathbf{S}$

$$\mathbf{y} = \sum_{t=0}^{T} \beta^t \mathbf{x}_t \;\Rightarrow\; \mathbf{y} = \sum_{t=0}^{T} (\beta \mathbf{I} - \beta \alpha \mathbf{S})^t \mathbf{x}_0$$

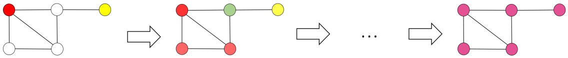
- Everything holds true if α_t or β_t are time varying

Diffusion dynamics and AR (IIR) filters

- Before finite-time dynamics (FIR filters)
- Consider now a diffusion dynamics $\mathbf{x}_t = \alpha \mathbf{S} \mathbf{x}_{t-1} + \mathbf{w}$

$$\mathbf{x}_t = \alpha^t \mathbf{S}^t \mathbf{x}_0 + \sum_{t'=0}^{t} \alpha^t \mathbf{S}^{t'} \mathbf{w}$$

$\Rightarrow$ When $t \to \infty$: $\mathbf{x}_\infty = (\mathbf{I} - \alpha \mathbf{S})^{-1} \mathbf{w} \Rightarrow$ AR graph filter



- Higher orders [Isufi-Leus16]
  - $\Rightarrow$ M successive diffusion dynamics $\Rightarrow$ AR of order M
  - $\Rightarrow$ Process is the sum of M parallel diffusions $\Rightarrow$ ARMA order M

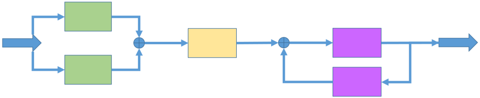
$$\mathbf{x}_\infty = \prod_{m=1}^{M} (\mathbf{I} - \alpha_m \mathbf{S})^{-1} \mathbf{w} \qquad \mathbf{x}_\infty = \sum_{m=1}^{M} (\mathbf{I} - \alpha_m \mathbf{S})^{-1} \mathbf{w}$$

General linear network processes

- Combinations of all the previous are possible

$$\mathbf{x}_t = \mathbf{H}_t^a(\mathbf{S})\mathbf{x}_{t-1} + \mathbf{H}_t^b(\mathbf{S})\mathbf{w} \Rightarrow \mathbf{x}_t = \mathbf{H}_t^A(\mathbf{S})\mathbf{x}_0 + \mathbf{H}_t^B(\mathbf{S})\mathbf{w}$$

  $\Rightarrow$ $\mathbf{y} = \mathbf{x}_t$, sequential/parallel application, linear combination



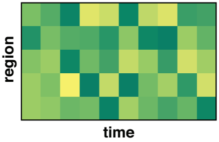
  $\Rightarrow$ Expands range of processes that can be modeled via GSP
  $\Rightarrow$ Coefficients can change according to some control inputs

- A number of linear processes can be modeled using graph filters
  $\Rightarrow$ Theoretical GSP results can be applied to distributed networking
  $\Rightarrow$ Deconvolution, filtering, system id, ...
  $\Rightarrow$ Beyond linearity possible too (more at the end of the talk)

- Links with control theory (of networks and complex systems)
  $\Rightarrow$ Controllability, observability

Application: Explaining human learning rates

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

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

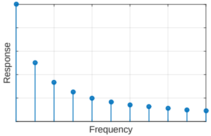
$\Rightarrow$ where Laplacian $\mathbf{L} = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^{-1}$ and β represents the diffusion rate

- Analyzing diffusion in the frequency domain

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

$\Rightarrow$ where $\tilde{h}_i = 1/(1 + \beta \lambda_i)$

- 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 $\mathbf{x}_{\text{null}}$

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

⇒ where π_t is a random permutation of the time indices

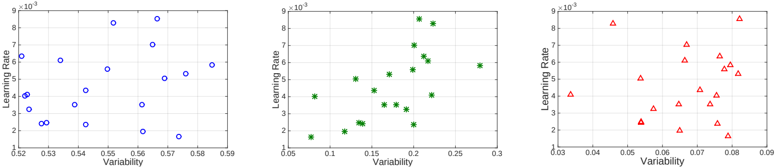
- Correlation between variability (m_1, m_2, and m_3) and learning?
- We consider three time series of brain activity
  - ⇒ The original fMRI data $\mathbf{x}$
  - ⇒ The filtered data $\mathbf{x}_{\text{diff}}$
  - ⇒ The null signal $\mathbf{x}_{\text{null}}$

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

Motivation and preliminaries

Part I: Fundamentals

Graph signals and the shift operator

Graph Fourier Transform (GFT)

Graph filters and network processes

Part II: Applications

Filter design for network operators

Sampling graph signals

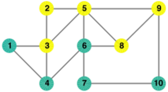
Blind identification of graph filters

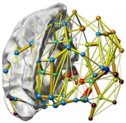
Network topology inference

Concluding remarks

Application domains

- Design graph filters to approximate desired network operators
- Sampling bandlimited graph signals
- Blind graph filter identification
  - ⇒ Infer diffusion coefficients from observed output
- Network topology inference
  - ⇒ Infer shift from collection of network diffused signals



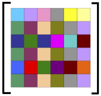


- Many more (not covered, glad to discuss or redirect):
  - ⇒ Statistical GSP, stationarity and spectral estimation
  - ⇒ Filter banks
  - ⇒ Windowing, convolution, duality...
  - ⇒ Nonlinear GSP

Distributed network operators

- Design graph filters to implement a given linear transformation
  - ⇒ Implementation is distributed by construction
  - ⇒ Conditions for perfect and approximate implementation
  - ⇒ [Shuman11], [Sandryhaila14], [Safavi15], [Chen15]
- Given a linear transformation $\mathbf{B}$, find the filter coefficients $\mathbf{h}$ s. t.

$$\mathbf{B} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$$

$\mathbf{B} =$ 

- Graph-shift operator $\mathbf{S}$ is given
  - ⇒ Well-suited for cases where $\mathbf{S}$ is a network process
  - ⇒ E.g., diffusion in a social network
  - ⇒ Agents exchange information and weigh info observed
  - ⇒ Choosing $\mathbf{h}$ ⇒ fixing the weights

Conditions for perfect implementation

Perfect implementation of linear graph operators [Segarra15]

The linear transformation $\mathbf{B}$ can be implemented using a graph filter $\mathbf{H}$ if the following conditions hold true:
i) Matrices $\mathbf{B}$ and $\mathbf{S}$ are simultaneously diagonalizable.
ii) If $\lambda_{k_1} = \lambda_{k_2}$, then $\gamma_{k_1} = \gamma_{k_2}$; and $L \geq \# \{\lambda_k\}_{k=1}^N$ distinct.

- i) $\Rightarrow$ frequency basis of $\mathbf{B}$ and $\mathbf{S}$ the same $\Rightarrow$ necessary
- ii) $\Rightarrow$ two equal freqs. in $\mathbf{S}$ must be equal in $\mathbf{B}$ $\Rightarrow$ necessary

- Restrictive conditions but not impossible to satisfy $\Rightarrow$ Consensus $\mathbf{B}_{\text{con}} = \mathbf{1}\mathbf{1}^T$ favors i) and ii) because it is rank-one

In time:
i) $\Leftrightarrow$ $\mathbf{B}$ circulant

- If satisfied: $\mathbf{h}^* = \mathbf{\Psi}^{-1}\boldsymbol{\gamma}$, where $\boldsymbol{\gamma} = [\gamma_1, ..., \gamma_N]^T$ are eigvals. of $\mathbf{B}$

Approximate design

- When perfect reconstruction is infeasible $\Rightarrow$ minimize error metric
  - $\Rightarrow$ Design $\mathbf{Hx}$ to resemble $\mathbf{Bx}$ (or $\mathbf{H}$ to resemble $\mathbf{B}$)
  - $\Rightarrow$ Minimizing $\|(\mathbf{H}-\mathbf{B})\mathbf{R_x}(\mathbf{H}-\mathbf{B})^T\|_z$ (with $\mathbf{R}_x = \mathbf{I}$ if unknown)
- MSE coefficients: $\mathbf{h}^* = \mathbf{\Theta}_{\mathbf{R_x}}^{\dagger}\mathbf{b}_{\mathbf{R_x}} = (\mathbf{\Theta}_{\mathbf{R_x}}^T\mathbf{\Theta}_{\mathbf{R_x}})^{-1}\mathbf{\Theta}_{\mathbf{R_x}}^T\mathbf{b}_{\mathbf{R_x}}$
  - $\Rightarrow$ with $\mathbf{\Theta}_{\mathbf{R_x}} := [\mathrm{vec}(\mathbf{I}\mathbf{R}_\mathbf{x}^{1/2}), ..., \mathrm{vec}(\mathbf{S}^{L-1}\mathbf{R}_\mathbf{x}^{1/2})]$, $\mathbf{b}_{\mathbf{R_x}} := \mathrm{vec}(\mathbf{B}\mathbf{R}_\mathbf{x}^{1/2})$
- Worst-case error coefficients:

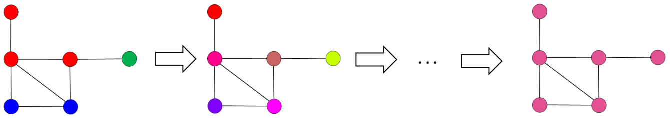
$$\{\mathbf{h}^*, s^*\} = \underset{\{\mathbf{h}, s\}}{\mathrm{argmin}} \quad s$$
$$\text{s. to} \begin{bmatrix} s\mathbf{I} & \mathbf{V}\mathrm{diag}(\mathbf{\Psi h})\mathbf{V}^{-1} - \mathbf{B} \\ (\mathbf{V}\mathrm{diag}(\mathbf{\Psi h})\mathbf{V}^{-1} - \mathbf{B})^T & s\mathbf{R}_\mathbf{x}^{-1} \end{bmatrix} \succeq 0.$$

- Additional assumptions can be incorporated

Consensus and rank-1 transformations

Consensus

- Local implementation of the consensus operator $\mathbf{B}_{\text{con}} = \mathbf{1}\mathbf{1}^T/N$



Proposition [Segarra16]

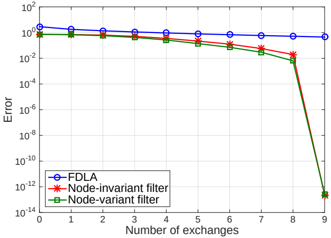
If $\mathcal{G}$ is connected and the desired operator $\mathbf{B}_{\text{rk1}}$ is rank one, then there exists an $\mathbf{S}$ such that $\mathbf{B}_{\text{rk1}}$ can be written as a graph filter $\sum_{l=0}^{N-1} h_l \mathbf{S}^l$.

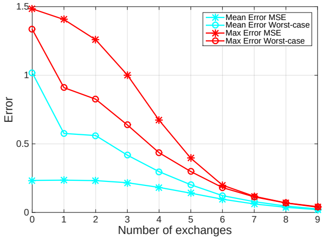
- Constructive proof, for consensus $\mathbf{S} = \mathbf{L}$
- Consensus is achieved in finite time [Sandryhaila-Kar-Moura14]
- Key: $\mathbf{B}$ low-rank (repeated eigenvalues) $\Rightarrow$ well-suited for approx.

- We compare the performance of: 1) Asymptotic fastest distributed linear averaging (FDLA), 2) Graph filter approx.

Finite-time consensus

- Define the graph-shift operator $\mathbf{S} = \mathbf{W}$
  $\Rightarrow$ Where $\lim_{k\to\infty} \mathbf{W}^k = \mathbf{B}_{\text{con}}$ with fastest convergence
- Plot average errors across the 100 graphs with 10 nodes
- Compare worst-case and mean error design (50 nodes)





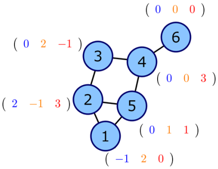
- Smaller error than FDLA for intermediate K
  $\Rightarrow$ When $K = N - 1 = 9$, perfect recovery
- The price to pay is that $\{\lambda_k\}_{k=1}^N$ need to be known
- Consistent performance of mean error and worst case designs

Node-variant graph filters: definition

- A generalization of graph filters [Segarra16]:

$$\mathbf{H}_{\mathrm{nv}} := \sum_{l=0}^{L-1} \mathrm{diag}(\mathbf{h}^{(l)})\mathbf{S}^{l}$$

$\Rightarrow$ When $\mathbf{h}^{(l)} = h_l \mathbf{1} \Rightarrow$ regular (node-invariant) filter



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

$$\mathbf{y} = h^{(0)}\mathbf{x}^{(0)} + h^{(1)}\mathbf{x}^{(1)} + h^{(2)}\mathbf{x}^{(2)}$$

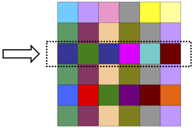
$$y_i = h_i^{(0)} x_i^{(0)} + h_i^{(1)} x_i^{(1)} + h_i^{(2)} x_i^{(2)}$$

- In general, when $\mathbf{H}_{\mathrm{nv}}$ is applied to a signal $\mathbf{x}$
  - $\Rightarrow$ Each node applies different weights to the shifted signals $\mathbf{S}^{l}\mathbf{x}$
  - $\Rightarrow$ More flexible and still distributed, not shift-invariant

Node-variant graph filters: frequency response

- Collect the coefficients of node i in $\mathbf{h}_i$, such that $[\mathbf{h}_i]_l = [\mathbf{h}^{(l)}]_i$
- Focus on the filter output at node i, $\mathbf{e}_i^T \mathbf{H}_{\text{nv}} \mathbf{x}$

$$\boldsymbol{\eta}_i^T = \mathbf{e}_i^T \mathbf{H}_{\text{nv}} = \sum_{l=0}^{L-1} [\mathbf{h}_i]_l \mathbf{e}_i^T \mathbf{V} \boldsymbol{\Lambda}^l \mathbf{V}^{-1}$$



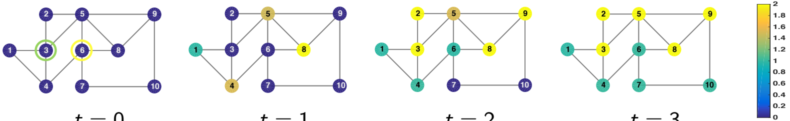
- Defining $\mathbf{u}_i := \mathbf{V}^T \mathbf{e}_i$

$$\boldsymbol{\eta}_i^T = \mathbf{u}_i^T \left(\sum_{l=0}^{L-1} [\mathbf{h}_i]_l \boldsymbol{\Lambda}^l \right) \mathbf{V}^{-1} = \mathbf{u}_i^T \text{diag}(\boldsymbol{\Psi} \mathbf{h}_i) \mathbf{V}^{-1}$$

- The output of the filter at node i, $\boldsymbol{\eta}_i^T \mathbf{x}$ is the inner product of
  - $\Rightarrow \mathbf{V}^{-1}\mathbf{x} \Rightarrow$ the frequency representation of the input, and
  - $\Rightarrow \mathbf{u}_i \Rightarrow$ how strongly the frequencies are expressed by node i
  - $\Rightarrow$ Modulated by $\boldsymbol{\Psi}\mathbf{h}_i \Rightarrow$ Frequency response associated to i

Perfect reconstruction with node-variant filters

- Node-variant filters can implement a large class of transformations
  - $\Rightarrow$ Pick $\mathbf{h}^{(l)}$ for $l = 0, \cdots, L-1$ so that $\mathbf{B} = \sum_{l=0}^{L-1} \text{diag}(\mathbf{h}^{(l)})\mathbf{S}^l$
  - $\Rightarrow$ TH: Always possible if $\mathbf{V}$ non-zero and $\{\lambda_k\}$ distinct
- Application in distributed processing: analog network coding
  - $\Rightarrow$ $\mathbf{B}$ is a binary matrix (input-output pairs)
- Example: G undirected, with $N = 10$, $\mathbf{S} = \mathbf{A}$, sources 3 and 6
  - $\Rightarrow$ Node 3 tx to 1, 4, 6, 7, and 10; node 6 to the remaining ones
  - $\Rightarrow$ Node invariant unable to implement $\mathbf{B}$



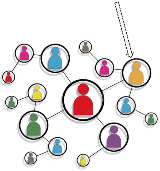
$t = 0$ $t = 1$ $t = 2$ $t = 3$

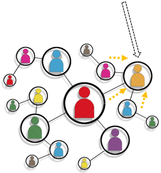
Sampling bandlimited graph signals

Motivation and preliminaries

- Sampling and interpolation are cornerstone problems in classical SP
  - ⇒ How recover a signal using only a few observations?
  - ⇒ Need to limit the degrees of freedom: subspace, smoothness
- Graph signals: sampling thoroughly investigated
  - ⇒ Most works assume only a few values are observed
  - ⇒ [Anis14, Chen15, Tsitsvero15, Puy15, Wang15]

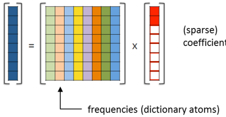
- Alternative approach [Marques16, Segarra16]
  - ⇒ GSP is well-suited for distributed networking
  - ⇒ Incorporate local graph structure into the observation model
  - ⇒ Recover signal using distributed local graph operators





Sampling bandlimited graph signals: Overview

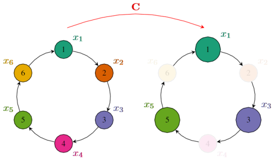
- Sampling is likely to be most important inverse problem
  - $\Rightarrow$ How to find $\mathbf{x} \in \mathbb{R}^N$ using $P < N$ observations?
- Our focus on bandlimited signals, but other models possible
  - $\Rightarrow$ $\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$ sparse
  - $\Rightarrow$ $\mathbf{x} = \sum_{k \in \mathcal{K}} \tilde{x}_k \mathbf{v}_k$, with $|\mathcal{K}| = K < N$
  - $\Rightarrow$ $\mathbf{S}$ involved in generation of $\mathbf{x}$
  - $\Rightarrow$ Agnostic to the particular form of $\mathbf{S}$



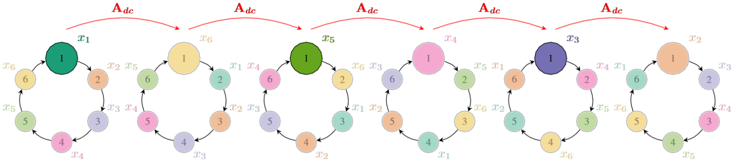
- Two sampling schemes were introduced in the literature
  - $\Rightarrow$ Selection [Anis14, Chen15, Tsitsvero15, Puy15, Wang15]
  - $\Rightarrow$ Aggregation [Segarra15], [Marques15]
  - $\Rightarrow$ Hybrid scheme combining both $\Rightarrow$ Space-shift sampling
- More involved, theoretical benefits, practical benefits in distr. setups

Revisiting sampling in time

- There are two ways of interpreting sampling of time signals
- We can either freeze the signal and sample values at different times

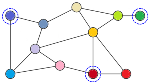


- We can fix a point (present) and sample the evolution of the signal

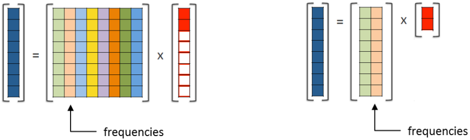


- Both strategies coincide for time signals but not for general graphs
  $\Rightarrow$ Give rise to selection and aggregation sampling

Selection sampling: Definition



- Intuitive generalization to graph signals
  - $\Rightarrow$ $\mathbf{C} \in \{0,1\}^{P \times N}$ (matrix P rows of $\mathbf{I}_N$)
  - $\Rightarrow$ Sampled signal is $\bar{\mathbf{x}} = \mathbf{C}\mathbf{x}$
- Goal: recover $\mathbf{x}$ based on $\bar{\mathbf{x}}$
  - $\Rightarrow$ Assume that the support of $\mathcal{K}$ is known (w.l.o.g. $\mathcal{K} = \{k\}_{k=1}^{K}$)
  - $\Rightarrow$ Since $\tilde{x}_k = 0$ for $k > K$, define $\tilde{\mathbf{x}}_K := [\tilde{x}_1, ..., \tilde{x}_K]^T = \mathbf{E}_K^T \tilde{\mathbf{x}}$



- Approach: use $\bar{\mathbf{x}}$ to find $\tilde{\mathbf{x}}_K$, and then recover $\mathbf{x}$ as

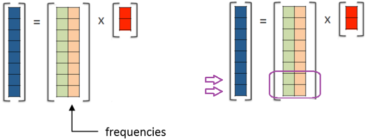
$$\mathbf{x} = \mathbf{V}(\mathbf{E}_K \tilde{\mathbf{x}}_K) = (\mathbf{V}\mathbf{E}_K)\tilde{\mathbf{x}}_K = \mathbf{V}_K \tilde{\mathbf{x}}_K$$

Selection sampling: Recovery

- Number of samples $P \geq K$

$$\bar{\mathbf{x}} = \mathbf{C}\mathbf{x} = \mathbf{C}\mathbf{V}_K \tilde{\mathbf{x}}_K$$

$\Rightarrow$ $(\mathbf{C}\mathbf{V}_K)$ submatrix of $\mathbf{V}$



Recovery of selection sampling

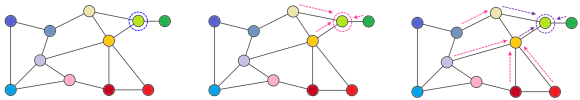
If $\text{rank}(\mathbf{C}\mathbf{V}_K) \geq K$, $\mathbf{x}$ can be recovered from the P values in $\bar{\mathbf{x}}$ as

$$\mathbf{x} = \mathbf{V}_K \tilde{\mathbf{x}}_K = \mathbf{V}_K (\mathbf{C}\mathbf{V}_K)^{\dagger} \bar{\mathbf{x}}$$

- With $P = K$, hard to check invertibility (by inspection)
  $\Rightarrow$ Columns of $\mathbf{V}_K(\mathbf{C}\mathbf{V}_K)^{-1}$ are the interpolators
- In time ($\mathbf{S} = \mathbf{A}_{dc}$), if the samples in $\mathbf{C}$ are equally spaced
  $\Rightarrow$ $(\mathbf{C}\mathbf{V}_K)$ is Vandermonde (DFT) and $\mathbf{V}_K(\mathbf{C}\mathbf{V}_K)^{-1}$ are sincs

Aggregation sampling: Definition

- Idea: incorporating $\mathbf{S}$ to the sampling procedure
  - $\Rightarrow$ Reduces to classical sampling for time signals
- Consider shifted (aggregated) signals $\mathbf{y}^{(l)} = \mathbf{S}^l \mathbf{x}$
  - $\Rightarrow \mathbf{y}^{(l)} = \mathbf{S}\mathbf{y}^{(l-1)} \Rightarrow$ found sequentially with only local exchanges
- Form $\mathbf{y}_i = [y_i^{(0)}, y_i^{(1)}, ..., y_i^{(N-1)}]^T$ (obtained locally by node i)



- The sampled signal is

$$\bar{\mathbf{y}}_i = \mathbf{C}\mathbf{y}_i$$

- Goal: recover $\mathbf{x}$ based on $\bar{\mathbf{y}}_i$

Aggregation sampling: Recovery

- Goal: recover $\mathbf{x}$ based on $\bar{\mathbf{y}}_i \Rightarrow$ Same approach than before
  $\Rightarrow$ Use $\bar{\mathbf{y}}_i$ to find $\tilde{\mathbf{x}}_K$, and then recover $\mathbf{x}$ as $\mathbf{x} = \mathbf{V}_K \tilde{\mathbf{x}}_K$
- Define $\bar{\mathbf{u}}_i := \mathbf{V}_K^T \mathbf{e}_i$ and recall $\Psi_{kl} = \lambda_k^{l-1}$

Recovery of aggregation sampling

Signal $\mathbf{x}$ can be recovered from the first K samples in $\bar{\mathbf{y}}_i$ as

$$\mathbf{x} = \mathbf{V}_K \tilde{\mathbf{x}}_K = \mathbf{V}_K \text{diag}^{-1}(\bar{\mathbf{u}}_i)(\mathbf{C}\boldsymbol{\Psi}^T \mathbf{E}_K)^{-1} \bar{\mathbf{y}}_i$$

provided that $[\bar{\mathbf{u}}_i]_k \neq 0$ and all $\{\lambda_k\}_{k=1}^K$ are distinct.

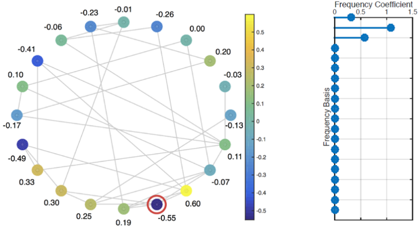
- If $\mathbf{C} = \mathbf{E}_K^T$, node i can recover $\mathbf{x}$ with info from $K-1$ hops!
  $\Rightarrow$ Node i has to be able to capture frequencies in $\mathcal{K}$
  $\Rightarrow$ The frequencies have to distinguishable
- **Bandlimited signals:** Signals that can be well estimated locally

Aggregation and selection sampling: Example

- In time ($\mathbf{S} = \mathbf{A}_{dc}$), selection and aggregation are equivalent

  $\Rightarrow$ Differences for a more general graph?

- Erdős-Rényi
  $p = 0.2$, $\mathbf{S} = \mathbf{A}$,
  $K = 3$,
  non-smooth



- First 3 observations at node 4: $\mathbf{y}_4 = [0.55, 1.27, 2.94]^T$

  $\Rightarrow$ $[\mathbf{y}_4]_1 = x_4 = -0.55$, $[\mathbf{y}_4]_2 = x_2 + x_3 + x_5 + x_6 + x_7 = 1.27$

  $\Rightarrow$ For this example, any node guarantees recovery

  $\Rightarrow$ Selection sampling fails if, e.g., $\{1, 3, 4\}$

Sampling: Discussion and extensions

- Discussion on aggregation sampling
  - ⇒ Observation matrix: diagonal times Vandermonde
  - ⇒ Very appropriate in distributed scenarios
  - ⇒ Different nodes will lead to different performance (soon)
  - ⇒ Types of signals that are actually bandlimited (role of $\mathbf{S}$)
- Three extensions:
  - ⇒ Sampling in the presence of noise
  - ⇒ Unknown frequency support
  - ⇒ Space-shift sampling (hybrid)

Presence of noise

- Linear observation model: $\bar{\mathbf{z}}_i = \mathbf{C}\mathbf{\Psi}_i\tilde{\mathbf{x}}_K + \mathbf{C}\mathbf{w}_i$ and $\mathbf{x} = \mathbf{V}_K\tilde{\mathbf{x}}_K$
- BLUE interpolation ($\mathbf{\Psi}_i$ either selection or aggregation)

$$\hat{\tilde{\mathbf{x}}}_K^{(i)} = [\mathbf{\Psi}_i^H\mathbf{C}^H(\bar{\mathbf{R}}_w^{(i)})^{-1}\mathbf{C}\mathbf{\Psi}_i]^{-1}\mathbf{\Psi}_i^H\mathbf{C}^H(\bar{\mathbf{R}}_w^{(i)})^{-1}\bar{\mathbf{z}}_i$$

$\Rightarrow$ If $P = K$, then $\hat{\mathbf{x}}^{(i)} = \mathbf{V}_K\left(\mathbf{C}\mathbf{\Psi}_i\right)^{-1}\bar{\mathbf{z}}_i$

- Error covariances ($\mathbf{R}_e^{(i)}$,$\tilde{\mathbf{R}}_e^{(i)}$) in closed form $\Rightarrow$ Noise covariances?

  $\Rightarrow$ Colored, different models: white noise in $\mathbf{z}_i$, in $\mathbf{x}$, or in $\tilde{\mathbf{x}}_K$

- Metric to optimize?

  $\Rightarrow \mathrm{trace}(\mathbf{R}_e^{(i)})$, $\lambda_{\max}(\mathbf{R}_e^{(i)})$, $\log\det(\tilde{\mathbf{R}}_e^{(i)})$, $\left[\mathrm{trace}\left(\tilde{\mathbf{R}}_e^{(i)^{-1}}\right)\right]^{-1}$

- Select i and $\mathbf{C}$ to min. error $\Rightarrow$ Depends on metric and noise [Marques16]

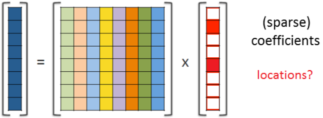
Unknown frequency support

- Falls into the class of sparse reconstruction: observation matrix?

$\Rightarrow$ Selec. $\Rightarrow$ submatrix of unitary $\mathbf{V}_{\mathcal{K}}$

$\Rightarrow$ Aggr. $\Rightarrow$ Vander. $\times$ diag

$[\mathbf{u}_i]_k \neq 0$ and $\lambda_k \neq \lambda_{k'} \Rightarrow$ full-spark



- Joint recovery and support identification (noiseless)

$$\tilde{\mathbf{x}}^* := \arg\min_{\tilde{\mathbf{x}}} \quad \|\tilde{\mathbf{x}}\|_0$$
$$\text{s.t.} \quad \mathbf{C}\mathbf{y}_i = \mathbf{C}\boldsymbol{\Psi}_i\tilde{\mathbf{x}},$$

- If full spark $\Rightarrow P = 2K$ samples suffice

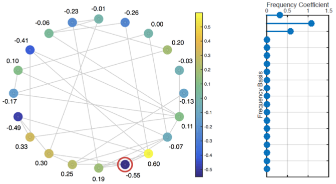
  $\Rightarrow$ Different relaxations are possible

  $\Rightarrow$ Conditioning will depend on $\boldsymbol{\Psi}_i$ (e.g., how different $\{\lambda_k\}$ are)

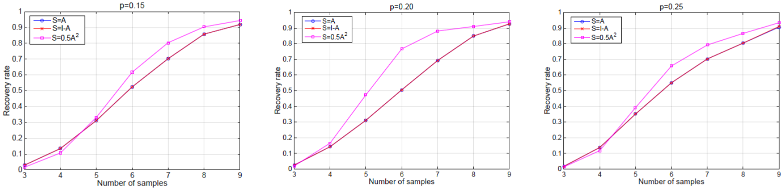
- Noisy case: sampling nodes critical

Recovery with unknown support: Example

- Erdős-Rényi
  $p = 0.15, 0.20, 0.25$,
  $K = 3$, non-smooth



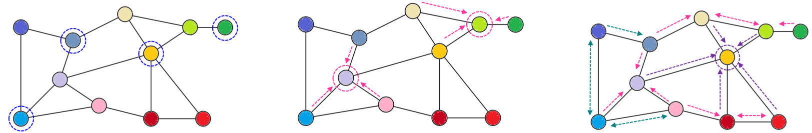
- Three different shifts: $\mathbf{A}$, $(\mathbf{I} - \mathbf{A})$ and $\frac{1}{2}\mathbf{A}^2$



Space-shift sampling

- Space-shift sampling (hybrid) $\Rightarrow$ Multiple nodes and multiple shifts

Selection: 4 nodes, 1 sample — Space-shift: 2 nodes, 2 samples — Aggregat.: 1 node, 4 samples



- Section and aggregation sampling as particular cases
- With $\bar{\mathbf{U}} := [\text{diag}(\bar{\mathbf{u}}_1), ..., \text{diag}(\bar{\mathbf{u}}_N)]^T$, the sampled signal is

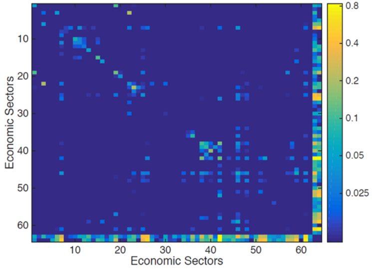
$$\bar{\underline{\mathbf{z}}} = \mathbf{C}\Big(\mathbf{I}\otimes(\mathbf{\Psi}^T\mathbf{E}_K)\Big)\bar{\mathbf{U}}\tilde{\mathbf{x}}_K + \mathbf{C}\underline{\mathbf{w}}$$

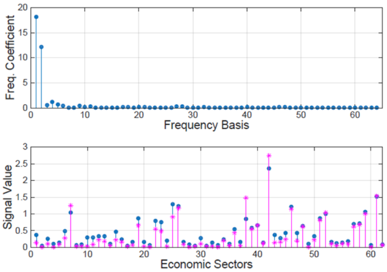
- As before, BLUE and error covariance in close-form
- Optimizing sample selection more challenging
- More structured schemes easier: e.g., message passing

$\Rightarrow$ Node i knows $y_i^{(l)}$ $\Rightarrow$ node i knows $y_j^{(l')}$ for all $j \in \mathcal{N}_i$ and $l' < l$

Sampling the US economy

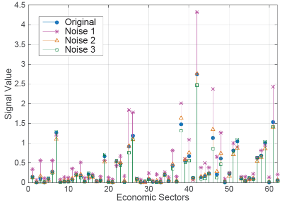
- 62 economic sectors in USA + 2 artificial sectors
  - $\Rightarrow$ Graph: average flows in 2007-2010, $\mathbf{S} = \mathbf{A}$
  - $\Rightarrow$ Signal $\mathbf{x}$: production in 2011
  - $\Rightarrow$ $\mathbf{x}$ is approximately bandlimited with $K = 4$

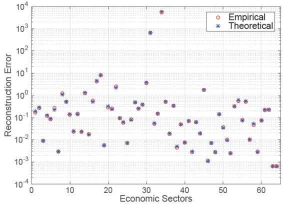




Sampling the US economy: Results

- Setup 1: we add different types of noise
  - $\Rightarrow$ Error depends on sampling node: better if more connected





- Setup 2: we try different shift-space strategies

Sampling strategy				Min. error	Median error
$[\mathbf{x}]_i$	$[\mathbf{Sx}]_i$	$[\mathbf{S}^2\mathbf{x}]_i$	$[\mathbf{S}^3\mathbf{x}]_i$	.0035	.019
$[\mathbf{x}]_i$	$[\mathbf{x}]_j$	$[\mathbf{x}]_k$	$[\mathbf{x}]_l$	.0039	4.2
$[\mathbf{Sx}]_i$	$[\mathbf{Sx}]_j$	$[\mathbf{Sx}]_k$	$[\mathbf{Sx}]_l$	.0035	.030
$[\mathbf{S}^2\mathbf{x}]_i$	$[\mathbf{S}^2\mathbf{x}]_j$	$[\mathbf{S}^2\mathbf{x}]_k$	$[\mathbf{S}^2\mathbf{x}]_l$	.0035	.0055
$[\mathbf{S}^3\mathbf{x}]_i$	$[\mathbf{S}^3\mathbf{x}]_j$	$[\mathbf{S}^3\mathbf{x}]_k$	$[\mathbf{S}^3\mathbf{x}]_l$	.0035	.0040
$[\mathbf{x}]_i$	$[\mathbf{Sx}]_i$	$[\mathbf{x}]_j$	$[\mathbf{Sx}]_j$	.0035	.039

More on sampling graph signals

- Beyond bandlimitedness
  - ⇒ Smooth signals [Chen15]
  - ⇒ Parsimonious in kernelized domain [Romero-Giannakis16]
- Strategies to select the sampling nodes
  - ⇒ Random (sketching) [Varma15]
  - ⇒ Optimal reconstruction [Marques16, Chepuri-Leus16]
  - ⇒ Designed based on posterior task [Gama16]
- And more...
  - ⇒ Low-complexity implementations [Tremblay16, Anis16]
  - ⇒ Local implementations [Wang14, Segarra15]
  - ⇒ Unknown spectral decomposition [Anis16]

Blind identification of graph filters

Diffusion processes as graph filter outputs

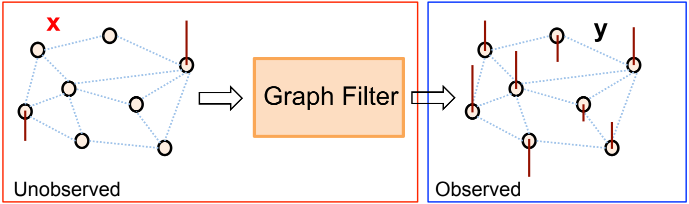
- Q: Upon observing a graph signal $\mathbf{y}$, how was this signal generated?
- Postulate the following generative model
  - ⇒ An originally sparse signal $\mathbf{x} = \mathbf{x}^{(0)}$
  - ⇒ Diffused via linear graph dynamics $\mathbf{S}$ ⇒ $\mathbf{x}^{(l)} = \mathbf{S}\mathbf{x}^{(l-1)}$
  - ⇒ Observed $\mathbf{y}$ is a linear combination of the diffused signals $\mathbf{x}^{(l)}$

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

- Model: Observed network process as output of a graph filter
  - ⇒ View few elements in $\text{supp}(\mathbf{x}) =: \{i : x_i \neq 0\}$ as seeds

Motivation and problem statement

- Ex: Global opinion/belief profile formed by spreading a rumor
  - ⇒ What was the rumor? Who started it?
  - ⇒ How do people weigh in peers' opinions to form their own?



- Problem: Blind identification of graph filters with sparse inputs
- Q: Given $\mathbf{S}$, can we find $\mathbf{x}$ and the combination weights $\mathbf{h}$ from $\mathbf{y} = \mathbf{H}\mathbf{x}$?
  - ⇒ Extends classical blind deconvolution to graphs

Blind graph filter identification

- Leverage frequency response of graph filters ($\mathbf{U} := \mathbf{V}^{-1}$)

$$\mathbf{y} = \mathbf{H}\mathbf{x} \;\Rightarrow\; \mathbf{y} = \mathbf{V}\text{diag}(\mathbf{\Psi}\mathbf{h})\mathbf{U}\mathbf{x}$$

$\Rightarrow$ $\mathbf{y}$ is a bilinear function of the unknowns $\mathbf{h}$ and $\mathbf{x}$

- Problem is ill-posed $\Rightarrow$ $(L+1)+N$ unknowns and N observations

$\Rightarrow$ **As.:** $\mathbf{x}$ is S-sparse i.e., $\|\mathbf{x}\|_0 := |\text{supp}(\mathbf{x})| \leq S$

- Blind graph filter identification $\Rightarrow$ Non-convex feasibility problem

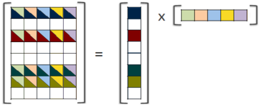
$$\text{find } \{\mathbf{h}, \mathbf{x}\}, \quad \text{s. to} \quad \mathbf{y} = \mathbf{V}\text{diag}\big(\mathbf{\Psi}\mathbf{h}\big)\mathbf{U}\mathbf{x}, \;\; \|\mathbf{x}\|_0 \leq S$$

"Lifting" the bilinear inverse problem

- Key observation: Use the Khatri-Rao product $\odot$ to write $\mathbf{y}$ as

$$\mathbf{y} = \mathbf{V}(\boldsymbol{\Psi}^T \odot \mathbf{U}^T)^T \text{vec}(\mathbf{x}\mathbf{h}^T)$$

- Reveals $\mathbf{y}$ is a linear combination of the entries of $\mathbf{Z} := \mathbf{x}\mathbf{h}^T$



- $\mathbf{Z}$ is of rank-1 and row-sparse $\Rightarrow$ Linear matrix inverse problem

$$\min_{\mathbf{Z}} \text{rank}(\mathbf{Z}), \quad \text{s. to } \mathbf{y} = \mathbf{V}\left(\boldsymbol{\Psi}^T \odot \mathbf{U}^T\right)^T \text{vec}\left(\mathbf{Z}\right), \quad \|\mathbf{Z}\|_{2,0} \leq S$$

$\Rightarrow$ Pseudo-norm $\|\mathbf{Z}\|_{2,0}$ counts the nonzero rows of $\mathbf{Z}$

$\Rightarrow$ Matrix "lifting" for blind deconvolution [Ahmed etal'14]

- Rank minimization s. to row-cardinality constraint is NP-hard. Relax!

Algorithmic approach via convex relaxation

- Key property: ℓ_1-norm minimization promotes sparsity [Tibshirani'94]
  - Nuclear norm $\|\mathbf{Z}\|_* := \sum_i \sigma_i(\mathbf{Z})$ a convex proxy of rank [Fazel'01]
  - $\ell_{2,1}$ norm $\|\mathbf{Z}\|_{2,1} := \sum_i \|\mathbf{z}_i^T\|_2$ surrogate of $\|\mathbf{Z}\|_{2,0}$ [Yuan-Lin'06]

- Convex relaxation

$$\min_{\mathbf{Z}} \|\mathbf{Z}\|_* + \alpha\|\mathbf{Z}\|_{2,1}, \quad \text{s. to } \mathbf{y} = \mathbf{V}\left(\mathbf{\Psi}^T \odot \mathbf{U}^T\right)^T \text{vec}(\mathbf{Z})$$

$\Rightarrow$ Scalable algorithm using method of multipliers

- Refine estimates $\{\mathbf{h}, \mathbf{x}\}$ via iteratively-reweighted optimization

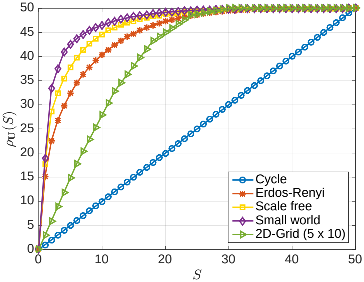
$\Rightarrow$ Weights $\alpha_i(k) = (\|\mathbf{z}_i(k)^T\|_2 + \delta)^{-1}$ per row i, per iteration k

- Noisy and partial observations $\Rightarrow$ Adjust constraints
  - Noise in $\mathbf{y}$: $\|\mathbf{y} - \mathbf{V}\left(\mathbf{\Psi}^T \odot \mathbf{U}^T\right)^T \text{vec}(\mathbf{Z})\| \leq \varepsilon$
  - Sampling via selection matrix $\mathbf{C}$: $\mathbf{y}_C = \mathbf{C}\mathbf{V}\left(\mathbf{\Psi}^T \odot \mathbf{U}^T\right)^T \text{vec}(\mathbf{Z})$

Exact recovery guarantees

- Exact recovery $\Rightarrow$ Success of the convex relaxation
  - Random model on the graph structure [Ling-Stromher'15]
  - Probabilistic guarantees depend on graph spectrum

$$P_{\text{rec}} \geq 1 - N^{-O\left(\rho_{\mathbf{U}}^{-1}(S)\right)}, \quad \rho_{\mathbf{U}}(S) := \max_{I \in \{1,\ldots,N\}} \max_{\Omega \in \Omega_S^N} \|\mathbf{u}_{I,\Omega}\|_2^2$$

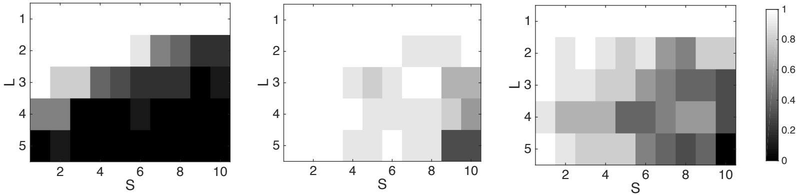


- Blind deconvolution (in time) most favorable graph setting

Details in arXiv:1604.07234v1 [cs.IT]

Numerical tests: Recovery rates

- Recovery rates over an (L, S) grid and 20 trials
  - Successful recovery when $\|\mathbf{x}^*(\mathbf{h}^*)^T - \mathbf{x}\mathbf{h}^T\|_F < 10^{-3}$
- ER (left), ER reweighted $\ell_{2,1}$ (center), brain reweighted $\ell_{2,1}$ (right)



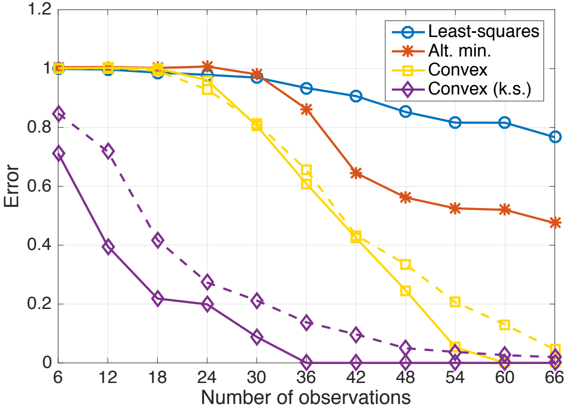
- Exact recovery over non-trivial (L, S) region

$\Rightarrow$ Reweighted optimization markedly improves performance

$\Rightarrow$ Encouraging results even for real-world graphs

Numerical tests: Brain graph

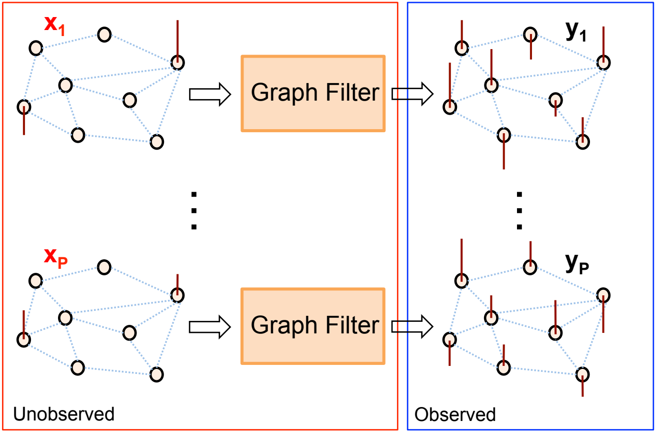
- Human brain graph with $N = 66$ regions, $L = 3$ and $S = 3$



- Proposed method also outperforms alternating-minimization solver
  - $\Rightarrow$ Unknown supp($\mathbf{x}$) $\approx$ Need twice as many observations
  - $\Rightarrow$ Stable to Gaussian noise in $\mathbf{y}$ ($\sigma^2 = 0.01$)

Multiple output signals

- Suppose we have access to P output signals $\{\mathbf{y}_p\}_{p=1}^{P}$



- Goal: Identify common filter $\mathbf{H}$ fed by multiple unobserved inputs $\mathbf{x}_p$

Formulation

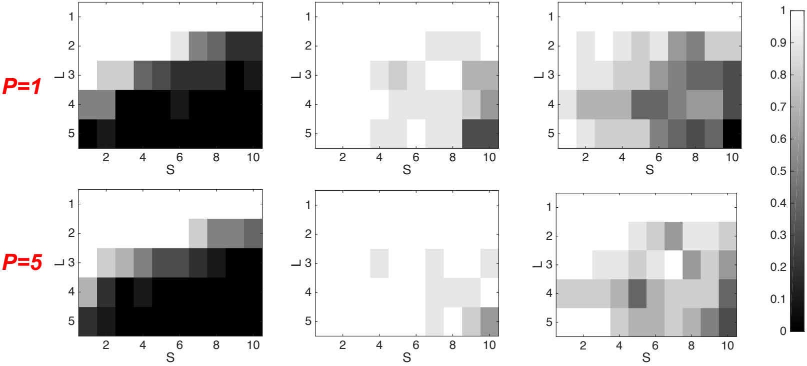
- **As.:** $\{\mathbf{x}_p\}_{p=1}^P$ are S-sparse with common support
- Concatenate outputs $\bar{\mathbf{y}} := [\mathbf{y}_1^T, \ldots, \mathbf{y}_P^T]^T$ and inputs $\bar{\mathbf{x}} := [\mathbf{x}_1^T, \ldots, \mathbf{x}_P^T]^T$
- Unknown rank-one matrices $\mathbf{Z}_p := \mathbf{x}_p \mathbf{h}^T$. Stack them
  - $\Rightarrow$ Vertically in rank one $\bar{\mathbf{Z}}_v := [\mathbf{Z}_1^T, ..., \mathbf{Z}_P^T]^T = \bar{\mathbf{x}}\mathbf{h}^T \in \mathbb{R}^{NP \times L}$
  - $\Rightarrow$ Horizontally in row sparse $\bar{\mathbf{Z}}_h := [\mathbf{Z}_1, ..., \mathbf{Z}_P] \in \mathbb{R}^{N \times PL}$
- Convex formulation

$$\min_{\{\mathbf{Z}_p\}_{p=1}^P} \|\bar{\mathbf{Z}}_v\|_* + \tau \|\bar{\mathbf{Z}}_h\|_{2,1}, \quad \text{s. to } \bar{\mathbf{y}} = \left(\mathbf{I}_P \otimes \left(\mathbf{V}\left(\boldsymbol{\Psi}^T \odot \mathbf{U}^T\right)^T\right)\right) \text{vec}\left(\bar{\mathbf{Z}}_h\right)$$

$\Rightarrow$ Relax **(As.)**: $\|\bar{\mathbf{Z}}_h\|_{2,1} \leftrightarrow \|\bar{\mathbf{Z}}_v\|_{2,1} = \sum_{p=1}^P \|\mathbf{Z}_p\|_{2,1}$

Numerical tests: Multiple signals, recovery rates

- Recovery rates over an (L, S) grid and 20 trials
  - Successful recovery when $\|\hat{\bar{\mathbf{x}}}\hat{\mathbf{h}}^T - \bar{\mathbf{x}}\mathbf{h}^T\|_F < 10^{-3}$
- ER (left), ER reweighted $\ell_{2,1}$ (center), brain reweighted $\ell_{2,1}$ (right)



- Leveraging multiple output signals aids the blind identification task

Blind ID: Takeaways

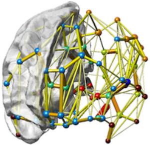
- Extended blind deconvolution of space/time signals to graphs
  $\Rightarrow$ Key: model diffusion process as output of graph filter
- Rank and sparsity minimization subject to model constraints
  $\Rightarrow$ "Lifting" and convex relaxation yield efficient algorithms
- Exact recovery conditions $\Rightarrow$ Success of the convex relaxation
  $\Rightarrow$ Probabilistic guarantees that depend on the graph spectrum
- Consideration of multiple sparse inputs aids recovery
- Envisioned application domains
  (a) Opinion formation in social networks
  (b) Identify sources of epileptic seizure
  (c) Trace "patient zero" for an epidemic outbreak
- Unknown shift $\mathbf{S}$ $\Rightarrow$ Network topology inference

Network topology inference

Motivation and context

- Network topology inference from nodal observations
  - ⇒ Approaches use Pearson correlations to construct graphs
  - ⇒ Partial correlations and conditional dependence

- Paramount importance in neuroscience
  - ⇒ Functional net inferred from activity



- Most GSP works assume that **S** (hence the graph) is known
  - ⇒ Analyze how the characteristics of **S** affect signals and filters

- We take the reverse path
  - ⇒ How to use GSP to infer the graph topology?
  - ⇒ [Dong15, Mei15, Pavez16, Pasdeloup16]

Generating structure of a diffusion process

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

$$\mathbf{x} \;=\; \alpha_0 \prod_{l=1}^{\infty}(\mathbf{I} - \alpha_l \mathbf{S})\mathbf{w} \;=\; \sum_{l=0}^{\infty} \beta_l \mathbf{S}^l \mathbf{w}$$

$\Rightarrow$ Common generative model. Heat diffusion if α_k constant

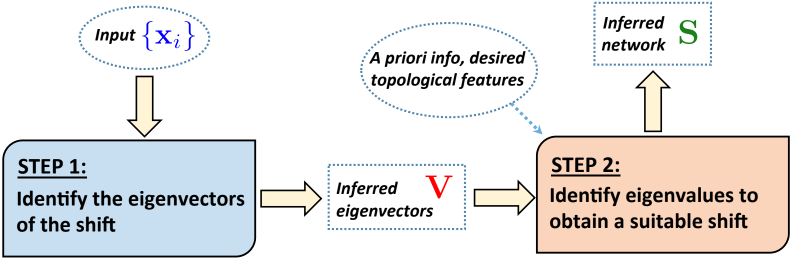
- We say the graph shift $\mathbf{S}$ explains the structure of signal $\mathbf{x}$
- It follows from Cayley Hamilton that we can write diffusion as

$$\mathbf{x} = \left(\sum_{l=0}^{N-1} h_l \mathbf{S}^l\right)\mathbf{w} := \mathbf{H}\mathbf{w}$$

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

Our approach for topology inference

- We propose a two-step approach for graph topology identification



- Beyond diffusion $\Rightarrow$ alternative sources for spectral templates $\mathbf{V}$

STEP 1: Obtaining the eigenvectors

- The covariance matrix of the signal $\mathbf{x}$ is

$$\mathbf{C}_x = \mathbb{E}\left[\left(\mathbf{Hw}\left(\mathbf{Hw}\right)^H\right)\right] = \mathbf{H}\mathbb{E}\left[\left(\mathbf{ww}^H\right)\right]\mathbf{H}^H = \mathbf{HH}^H$$

- Since $\mathbf{H}$ is diagonalized by $\mathbf{V}$, so is the covariance $\mathbf{C}_x$

$$\mathbf{C}_x \;=\; \mathbf{V}\left|\sum_{l=0}^{L-1} h_l \boldsymbol{\Lambda}^l\right|^2 \mathbf{V}^H \;=\; \mathbf{V}\,\mathrm{diag}(|\tilde{\mathbf{h}}|^2)\,\mathbf{V}^H$$

- Any shift with eigenvectors $\mathbf{V}$ can explain $\mathbf{x}$
  - $\Rightarrow$ G and its specific eigenvalues have been obscured by diffusion

Observations

(a) There are many shifts that can explain a signal $\mathbf{x}$

(b) Identifying the shift $\mathbf{S}$ is just a matter of identifying the eigenvalues

(c) In correlation methods the eigenvalues are kept unchanged

(d) In precision methods the eigenvalues are inverted

Other sources of spectral templates

1) Implementation of linear network operators

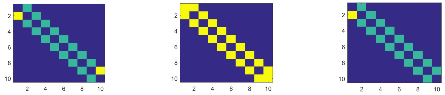
- Goal: distributed implementation of linear operator $\mathbf{B}$ via graph filter
  - $\Rightarrow$ $\mathbf{B}$ and $\mathbf{S}$ sharing $\mathbf{V}$ is beneficial for implementation
- Given a pre-specified $\mathbf{B}$
  - $\Rightarrow$ Use its eigenvectors as spectral templates to generate a shift $\mathbf{S}$
  - $\Rightarrow$ The goal here not to identify a shift, but to design one

Ex.: consensus $\Rightarrow$ Laplacian of the smallest connected graph

2) Relationship between nodes of a signal

- Particular transforms $\mathbf{T}$ are known to work well on specific data
  - $\Rightarrow$ Such transform assumes an implicit relation among data $\Rightarrow$ $\mathbf{S}$
  - $\Rightarrow$ Identification of that relation can provide insights $\mathbf{V}^H = \mathbf{T}$

DCTs: i–iii



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

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

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

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

  - $\Rightarrow$ Can accommodate Laplacian matrices as well
- Problem is convex if we select a convex objective $f(\mathbf{S}, \boldsymbol{\lambda})$
  - $\Rightarrow$ Minimum energy ($f(\mathbf{S}) = \|\mathbf{S}\|_F$), Fast mixing ($f(\boldsymbol{\lambda}) = -\lambda_2$)

Size of the feasibility set

- The feasibility set in (1) is generally small
  - $\Rightarrow$ Define $\mathbf{W} := \mathbf{V} \odot \mathbf{V}$ where $\odot$ is the Khatri-Rao product
  - $\Rightarrow$ Denote by $\mathcal{D}$ the index set such that $\mathrm{vec}(\mathbf{S})_{\mathcal{D}} = \mathrm{diag}(\mathbf{S})$

Assume that (1) is feasible, then it holds that $\mathrm{rank}(\mathbf{W}_{\mathcal{D}}) \leq N - 1$.
If $\mathrm{rank}(\mathbf{W}_{\mathcal{D}}) = N - 1$, then the feasible set of (1) is a singleton.

- Convex feasibility set $\Rightarrow$ Search for the optimal solution may be easy
- Simulations will show that $\mathrm{rank}(\mathbf{W}_{\mathcal{D}}) = N - 1$ arises in practice

Sparse recovery

- Whenever the feasibility set of (1) is non-trivial
  $\Rightarrow f(\mathbf{S}, \boldsymbol{\lambda})$ determines the features of the recovered graph

Ex: Identify the sparsest shift $\mathbf{S}_0^*$ that explains observed signal structure
  $\Rightarrow$ Set the cost $f(\mathbf{S}, \boldsymbol{\lambda}) = \|\mathbf{S}\|_0$

- Problem is not convex, but can relax to ℓ_1 norm minimization

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

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

Recovery guarantee

- Denoting by $\mathbf{m}_i^T$ the i-th row of $\mathbf{M} := (\mathbf{I} - \mathbf{W}\mathbf{W}^{\dagger})_{\mathcal{D}^c}$
  - $\Rightarrow$ Construct $\mathbf{R} := [\mathbf{m}_2 - \mathbf{m}_1, \ldots, \mathbf{m}_{N-1} - \mathbf{m}_1, \mathbf{m}_N, \ldots, \mathbf{m}_{|\mathcal{D}^c|}]^T$
  - $\Rightarrow$ Denote by $\mathcal{K}$ the indices of the support of $\mathbf{s}_0^* = \mathrm{vec}(\mathbf{S}_0^*)$

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

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

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

Noisy and incomplete spectral templates

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

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

- Recovery result similar to the noiseless case can be derived
  $\Rightarrow$ Conditions under which we are guaranteed $d(\mathbf{S}^*, \mathbf{S}_0^*) \leq C\epsilon$

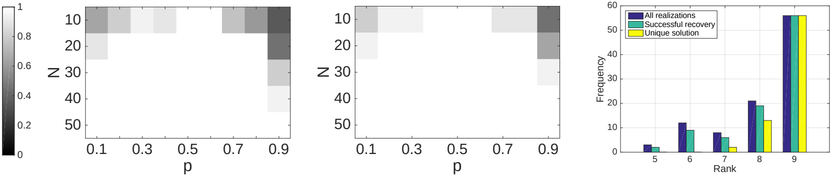
- Partial access to $\mathbf{V}$ $\Rightarrow$ Only K known eigenvectors $[v_1, \ldots, v_K]$

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

- Incomplete and noisy scenarios can be combined

Topology inference in random graphs

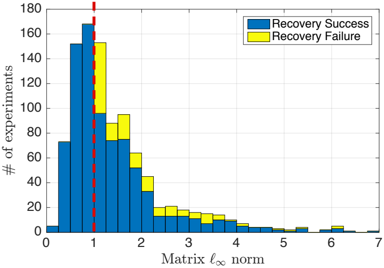
- Erdős-Rényi graphs of varying size $N \in \{10, 20, \ldots, 50\}$
  - $\Rightarrow$ Edge probabilities $p \in \{0.1, 0.2, \ldots, 0.9\}$
- Recovery rates for adjacency (left) and normalized Laplacian (mid)



- Recovery is easier for intermediate values of p
- Rate of recovery related to the $\text{rank}(\mathbf{W}_{\mathcal{D}})$ (histogram $N = 10, p = 0.2$)
  - $\Rightarrow$ When rank is $N - 1$, recovery is guaranteed
  - $\Rightarrow$ As rank decreases, there is a detrimental effect on recovery

Sparse recovery guarantee

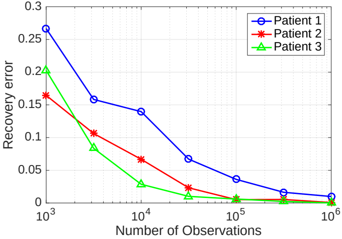
- Generate 1000 ER random graphs ($N = 20$, $p = 0.1$) such that
  - $\Rightarrow$ Feasible set is not a singleton
  - $\Rightarrow$ Cond. 1) in sparse recovery theorem is satisfied
- Noiseless case: ℓ_1 norm guarantees recovery as long as $\psi_{\mathbf{R}} < 1$



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

Inferring brain graphs from noisy templates

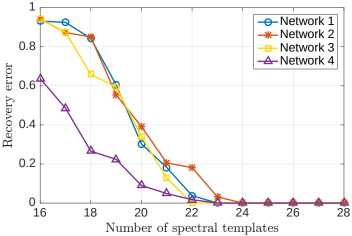
- Identification of structural brain graphs $N = 66$
- Test recovery for noisy spectral templates $\hat{\mathbf{V}}$
  - $\Rightarrow$ Obtained from sample covariances of diffused signals



- Recovery error decreases with increasing number of observed signals
  - $\Rightarrow$ More reliable estimate of the covariance $\Rightarrow$ Less noisy $\hat{\mathbf{V}}$
- Brain of patient 1 is consistently the hardest to identify
  - $\Rightarrow$ Robustness for identification in noisy scenarios
- Traditional methods like graphical lasso fail to recover $\mathbf{S}$

Inferring social graphs from incomplete templates

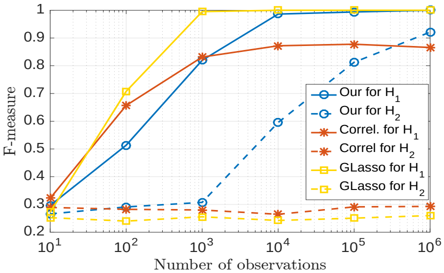
- Identification of multiple social networks $N = 32$
  - $\Rightarrow$ Defined on the same node set of students from Ljubljana
- Test recovery for incomplete spectral templates $\hat{\mathbf{V}} = [\mathbf{v}_1, \ldots, \mathbf{v}_K]$
  - $\Rightarrow$ Obtained from a low-pass diffusion process
  - $\Rightarrow$ Repeated eigenvalues in $\mathbf{C}_\mathbf{x}$ introduce rotation ambiguity in $\mathbf{V}$



- Recovery error decreases with increasing nr. of spectral templates
  - $\Rightarrow$ Performance improvement is sharp and precipitous

Performance comparisons

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



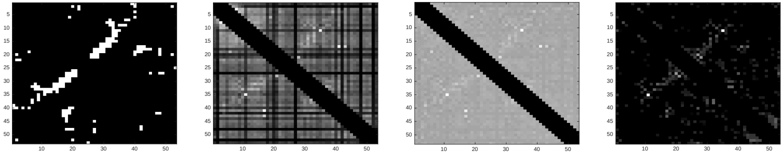
- Graphical lasso implicitly assumes a filter $\mathbf{H}_1 = (\rho\mathbf{I} + \mathbf{S})^{-1/2}$

  $\Rightarrow$ For this filter spectral templates work, but not as well (MLE)

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

Inferring direct relations

- Our method can be used to sparsify a given network
- 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 templates
- 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

Topology ID: Takeaways

- Network topology inference cornerstone problem in Network Science
  - Most GSP works analyze how $\mathbf{S}$ affect signals and filters
  - Here, reverse path: How to use GSP to infer the graph topology?
- Our GSP approach to network topology inference

  $\Rightarrow$ 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 too: net operators, data transforms
- Infer $\mathbf{S}$ via convex optimization

  $\Rightarrow$ Objectives promotes desirable properties

  $\Rightarrow$ Constraints encode structure a priori info and structure

  $\Rightarrow$ Formulations for perfect and imperfect templates

  $\Rightarrow$ Sparse recovery results for both adjacency and Laplacian

Wrapping up

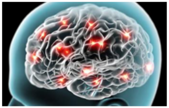
Concluding remarks

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

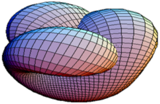
Concluding remarks

- GSP results can be applied to solve practical problems
  - ⇒ Filter design (design of distributed operators)
  - ⇒ Sampling, interpolation (network control)
  - ⇒ Blind deconvolution (source ID), 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

Looking ahead

- First step to challenging problems: social nets, brain signals
- Motivates further research:
  ⇒ Statistical modeling
  ⇒ Space-time variation
  ⇒ Changing topologies
  ⇒ Nonlinear approaches
  ⇒ Local, reduced-complexity algorithms
- **Thanks!**
  ⇒ Contact: antonio.garcia.marques@urjc.es gmateosb@ece.rochester.edu ssegarra@seas.upenn.edu aribeiro@seas.upenn.edu
  ⇒ Slides on stationarity available at:
  `http://tsc.urjc.es/~amarques/papers/ssamglar_sam16_slides.pdf`

References

We include a list of our published work in graph signal processing (GSP) categorized by topic. We also include relevant works by other authors. This latter list is not intended to be exhaustive but rather its purpose is to guide the interested reader to pertinent publications in different areas of graph signal processing.

References: our work

Sampling bandlimited graph signals

A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Sampling of Graph Signals with Successive Local Aggregations", IEEE Trans. on Signal Process., vol. 64, no. 7, pp. 1832 - 1843, Apr. 2016.

S. Segarra, A. G. Marques, G. Leus and A. Ribeiro, "Space-shift sampling of graph signals", Proc. of IEEE Intl. Conf. on Acoustics, Speech and Signal Process., Shanghai, China, March 20-25, 2016.

S. Segarra, A. G. Marques, G. Leus and A. Ribeiro, "Aggregation Sampling of Graph Signals in the Presence of Noise", Proc. of IEEE Intl. Wrksp. on Computational Advances in Multi-Sensor Adaptive Processing, Cancun, Mexico, Dec. 13-16, 2015.

S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Sampling of Graph Signals: Successive Local Aggregations at a Single Node", Proc. of 49th Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, CA, Nov. 8-11, 2015.

F. Gama, A. G. Marques, G. Mateos, and A. Ribeiro, "Rethinking Sketching as Sampling: Efficient Approximate Solution to Linear Inverse Problems", Proc. of IEEE of Global Conf. on Signal and Info. Process., Washington DC, Dec. 7-9, 2016.

F. Gama, A. G. Marques, G. Mateos, and A. Ribeiro, "Rethinking Sketching as Sampling: Linear Transforms of Graph Signals", Proc. of 50th Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, CA, Nov. 6-9, 2016.

References: our work

Interpolating graph signals

S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Reconstruction of Graph Signals: Percolation from a Single Seeding Node", Proc. of IEEE of Global Conf. on Signal and Info. Process., Orlando, FL, Dec. 14-16, 2015.

S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Interpolation of Graph Signals Using Shift-Invariant Graph Filters", Proc. of European Signal Process. Conf., Nice, France, Aug. 31-Sep. 4, 2015.

S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Reconstruction of Graph Signals through Percolation from Seeding Nodes", IEEE Trans. on Signal Process., vol. 64, no. 16, pp. 4363 4378, Aug. 2016.

Graph filter design and network operators

S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed Implementation of Network Linear Operators using Graph Filters", Proc. of 53rd Allerton Conf. on Commun. Control and Computing, Univ. of Illinois at U-C, Monticello, IL, Sept. 30- Oct. 2, 2015.

S. Segarra, A. G. Marques, and A. Ribeiro, "Distributed Network Linear Operators using Node Variant Graph Filters", Proc. of IEEE Intl. Conf. on Acoustics, Speech and Signal Process., Shanghai, China, March 20-25, 2016.

References: our work

Blind graph deconvolution

S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind Identification of Graph Filters with Sparse Inputs: Unknown support", Proc. of IEEE Intl. Conf. on Acoustics, Speech and Signal Process., Shanghai, China, March 20-25, 2016.

S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind Identification of Graph Filters with Sparse Inputs", Proc. of IEEE Intl. Wrksp. on Computational Advances in Multi-Sensor Adaptive Processing, Cancun, Mexico, Dec. 13-16, 2015.

S. Segarra, G. Mateos, A. G. Marques, and A. Ribeiro, "Blind Identification of Graph Filters", IEEE Trans. Signal Process. (arXiv:1604.07234 [cs.IT])

GSP-Based Network Topology Inference

S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network Topology Identification from Imperfect Spectral Templates", Proc. of 50th Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, CA, Nov. 6-9, 2016.

S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network Topology Identification from Spectral Templates", Proc. of IEEE Intl. Wrksp. on Statistical Signal Process., Palma de Mallorca, Spain, June 26-29, 2016.

S. Segarra, A. G. Marques, G. Mateos, and A. Ribeiro, "Network Topology Inference from Spectral Templates", IEEE Trans. Signal Process., (submitted).

References: our work

Stationary Graph Processes

A. G. Marques, S. Segarra, G. Leus, and A. Ribeiro, "Stationary Graph Processes and Spectral Estimation", IEEE Trans. Signal Process. (arXiv:1603.04667 [cs.SY])

S. Segarra, A. G. Marques, G. Leus, and A. Ribeiro, "Stationary Graph Processes: Nonparametric Power Spectral Estimation ", Proc. of IEEE Sensor Array and Multichannel Signal Process. Wrksp., Rio de Janeiro, Brazil, July 10-13, 2016.

Median Graph Filters

S. Segarra, A. Marques, G. Arce, and Alejandro Ribeiro, "Center-weighted Median Graph Filters", Proc. of IEEE of Global Conf. on Signal and Info. Process., Washington DC, Dec. 7-9, 2016.

References: GSP

General references

D. Shuman, S. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, The emerging field of signal processing on graphs: Extending highdimensional data analysis to networks and other irregular domains, IEEE Signal Process. Mag., vol. 30, no. 3, pp. 83-98, Mar. 2013.

A. Sandryhaila and J. Moura, Discrete signal processing on graphs, IEEE Trans. Signal Process., vol. 61, no. 7, pp. 1644-1656, Apr. 2013.

A. Sandryhaila and J. Moura, Discrete signal processing on graphs: Frequency analysis, IEEE Trans. Signal Process., vol. 62, no. 12, pp. 3042-3054, June 2014.

A. Sandryhaila and J. M. F. Moura, Big Data analysis with signal processing on graphs, IEEE Signal Process. Mag., vol. 31, no. 5, pp. 80-90, 2014.

M. Rabbat and V. Gripon, Towards a spectral characterization of signals supported on small-world networks, in IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP), May 2014, pp. 4793-4797.

A. Agaskar, Y.M. Lu, A spectral graph uncertainty principle, IEEE Trans. Info. Theory, vol. 59, no. 7, pp. 4338-4356, 2013.

References: GSP

Filtering

D. I. Shuman, P. Vandergheynst, and P. Frossard, Distributed signal processing via Chebyshev polynomial approximation, CoRR, vol. abs/1111.5239, 2011.

S. Safavi and U. Khan, Revisiting finite-time distributed algorithms via successive nulling of eigenvalues, IEEE Signal Process. Lett., vol. 22, no. 1, pp. 54-57, Jan. 2015.

Sampling

A. Anis, A. Gadde, and A. Ortega, Towards a sampling theorem for signals on arbitrary graphs, in IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP), May 2014, pp. 3864-3868.

I. Shomorony and A. S. Avestimehr, Sampling large data on graphs, arXiv preprint arXiv:1411.3017, 2014.

S. Chen, R. Varma, A. Sandryhaila, and J. Kovacevic, Discrete signal processing on graphs: Sampling theory, IEEE. Trans. Signal Process., vol. 63, no. 24, pp. 6510-6523, Dec. 2015.

M. Tsitsvero, Mikhail, S. Barbarossa, and P. Di Lorenzo. Signals on graphs: Uncertainty principle and sampling, arXiv preprint arXiv:1507.08822, 2015.

References: GSP

Interpolation and Reconstruction

S. Narang, A. Gadde, and A. Ortega, Signal processing techniques for interpolation in graph structured data, in IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP), May 2013, pp. 5445-5449.

S. Narang, A. Gadde, E. Sanou, and A. Ortega, Localized iterative methods for interpolation in graph structured data, in Global Conf. on Signal and Info. Process. (GlobalSIP), Dec. 2013, pp. 491-494.

S. Chen, A. Sandryhaila, J. M. Moura, and J. Kovacevic, Signal recovery on graphs: Variation on Graphs, IEEE Trans. Signal Process., vol. 63, no. 17, pp. 4609-4624, Sep. 2015.

X. Wang, P. Liu, and Y. Gu, Local-set-based graph signal reconstruction, IEEE Trans. Signal Process., vol. 63, no. 9, pp. 2432-2444, Sept. 2015.

X. Wang, M. Wang, and Y. Gu, A distributed tracking algorithm for reconstruction of graph signals, IEEE J. Sel. Topics Signal Process., vol. 9, no. 4, pp. 728-740, June 2015.

D. Romero, M. Meng, and G. Giannakis. Kernel-based Reconstruction of Graph Signals, arXiv preprint arXiv:1605.07174, 2016.

References: GSP

Topology Inference

V. Kalofolias, How to learn a graph from smooth signals, arXiv preprint arXiv:1601.02513, 2016.

X. Dong, D. Thanou, P. Frossard, and P. Vandergheynst, Learning laplacian matrix in smooth graph signal representations, arXiv preprint arXiv:1406.7842, 2014.

B. Pasdeloup, V. Gripon, G. Mercier, D. Pastor, M. Rabbat, Characterization and inference of weighted graph topologies from observations of diffused signals, arXiv preprint arXiv:1605.02569, 2016.

J. Mei and J. Moura, Signal processing on graphs: Estimating the structure of a graph, in IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP), 2015, pp. 54955499.

E. Pavez and A. Ortega, Generalized Laplacian precision matrix esti- mation for graph signal processing, in IEEE Intl. Conf. Acoust., Speech and Signal Process. (ICASSP), Shanghai, China, Mar. 20-25, 2016.

Stationarity

N. Perraudin and P. Vandergheynst, Stationary signal processing on graphs, arXiv preprint arXiv:1601.02522, 2016.

B. Girault, Stationary graph signals using an isometric graph translation," Proc. IEEE European Signal Process. Conf. (EUSIPCO), 2015.

GlobalSIP'16 Symposium on Networks

Symposium on Signal and Information Processing over Networks

Topics of interest

- Graph-signal transforms and filters
- Non-linear graph SP
- Statistical graph SP
- Prediction and learning in graphs
- Network topology inference
- Network tomography
- Control of network processes
- Signals in high-order graphs
- Graph algorithms for network analytics
- Graph-based distributed SP algorithms
- Graph-based image and video processing
- Communications, sensor and power networks
- Neuroscience and other medical fields
- Web, economic and social networks

Symposium dates: **December 8-9, 2016**

Organizers:

Michael Rabbat (McGill Univ.)

Gonzalo Mateos (Univ. of Rochester)

Antonio Marques (King Juan Carlos Univ.)