Graph Signal Processing: Fundamentals and Applications to Diffusion Processes

Antonio G. Marques†, Santiago Segarra‡, Alejandro Ribeiro*

†King Juan Carlos University
‡Massachusetts Institute of Technology
*University of Pennsylvania

antonio.garcia.marques@urjc.es

Thanks: Gonzalo Mateos, Geert Leus, and Weiyu Huang
Spanish MINECO grant No TEC2013-41604-R

EUSIPCO16
Budapest, Hungary - August 29, 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** $G = (\mathcal{V}, \mathcal{E}, W)$: encode pairwise relationships

▶ **Interest here not in** $G$ **itself, but in** data **associated with nodes** in $\mathcal{V}$
  => Object of study is a **graph signal** $x$

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

Oil and Gas  Services  Finance

- Oil extraction (OG), Petroleum and coal products (PC), Construction (CO)
- Administrative services (AS), Professional services (MP)
- Credit intermediation (FR), Securities (SC), Real state (RA), Insurance (IC)
- Only interactions stronger than a threshold are shown
Network of economic sectors of the United States

- Bureau of Economic Analysis of the U.S. Department of Commerce
- $\mathcal{E} = \text{Output of sector } i \text{ is an input to sector } j$ (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** $x = \text{output per sector} = \text{disaggregated GDP}$
  - 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]

  ⇒ **Our view**: GSP well suited to study network (diffusion) processes

- **As.**: Signal properties related to **topology** of $G$ (locality, smoothness)
  ⇒ 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
  - Time \( n \) follows time \( n - 1 \)
  - Signal value \( x_n \) similar to \( x_{n-1} \)

- Formalized with the notion of frequency

- Cyclic structure \( \Rightarrow \) Fourier transform \( \Rightarrow \) \( \tilde{x} = F^H x \)

\[
F_{kn} = \frac{e^{j2\pi kn/N}}{\sqrt{N}}
\]

- Fourier transform \( \Rightarrow \) Projection on eigenvector space of cycle
Random signal with mean $\mathbb{E}[x] = 0$ and covariance $C_x = \mathbb{E}[x x^H]$

$\Rightarrow$ Eigenvector decomposition $C_x = V \Lambda V^H$

Covariance matrix $C_x$ is a graph

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

Precision matrix $C_x^{-1}$ a common graph too

$\Rightarrow$ Conditional dependencies of Gaussian $x$

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

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

Q: Can we extend these principles to general graphs and signals?
Formally, a graph $G$ (or a network) is a triplet $(\mathcal{V}, \mathcal{E}, \mathcal{W})$

- $\mathcal{V} = \{1, 2, \ldots, N\}$ 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 $n$
- $\mathcal{W} : \mathcal{E} \to \mathbb{R}$ 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, $\mathcal{W} : \mathcal{E} \to \mathbb{R}^{++}$

- **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 $(n, m)$
• **Algebraic graph theory:** matrices associated with a graph $G$
  ⇒ Adjacency $A$ and Laplacian $L$ matrices
  ⇒ **Spectral graph theory:** properties of $G$ using spectrum of $A$ or $L$

• Given $G = (\mathcal{V}, \mathcal{E}, W)$, the adjacency matrix $A \in \mathbb{R}^{N \times N}$ 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 $G$
  ⇒ For **unweighted** graphs, positive entries represent connected pairs
  ⇒ 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 $|\mathcal{N}(i)|$
  - ⇒ In **directed** graphs, have **out-degree** and an **in-degree**

- Using the adjacency matrix in the **undirected** case
  - ⇒ For node $i$: $\deg(i) = \sum_{j \in \mathcal{N}(i)} A_{ij} = \sum_{j} A_{ij}$
  - ⇒ For all $N$ nodes: $d = \mathbf{A1}$ → **Degree matrix**: $\mathbf{D} := \text{diag}(d)$

- **Q**: Can this be extended to $k$-hop neighbors? → **Powers** of $\mathbf{A}$
  - ⇒ $[\mathbf{A}^k]_{ij}$ non-zero only if there exists a path of length $k$ from $i$ to $j$
  - ⇒ **Support** of $\mathbf{A}^k$: pairs that can be reached in $k$ hops
Given undirected $G$ with $A$ and $D$, the Laplacian matrix $L \in \mathbb{R}^{N \times N}$ is

$$L = D - A$$

Equivalently, $L$ can be defined element-wise as

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

Normalized Laplacian: $\mathcal{L} = D^{-1/2}LD^{-1/2}$ (we will focus on $L$)
Denote by $\lambda_i$ and $v_i$ the eigenvalues and eigenvectors of $L$.

$L$ is positive semi-definite

\[ x^T L x = \frac{1}{2} \sum_{(i,j) \in E} w_{ij} (x_i - x_j)^2 \geq 0, \text{ for all } x \]

\[ \Rightarrow \text{All eigenvalues are nonnegative, i.e. } \lambda_i \geq 0 \text{ for all } i \]

A constant vector $1$ is an eigenvector of $L$ with eigenvalue $0$

\[ [L1]_i = \sum_{j \in \mathcal{N}(i)} w_{ij} (1 - 1) = 0 \]

\[ \Rightarrow \text{Thus, } \lambda_1 = 0 \text{ and } v_1 = (1/\sqrt{N}) \cdot 1 \]

In connected graphs, it holds that $\lambda_i > 0$ for $i = 2, \ldots, N$

\[ \Rightarrow \text{Multiplicity} \{ \lambda = 0 \} = \text{number of connected components} \]
Part I: Fundamentals

Motivation and preliminaries

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

Part II: Applications
  Sampling graph signals
  Stationarity of graph processes
  Network topology inference

Concluding remarks
Consider graph $G = (\mathcal{V}, \mathcal{E}, \mathcal{W})$. **Graph signals** are mappings $x : \mathcal{V} \to \mathbb{R}$

⇒ 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 $x \in \mathbb{R}^N$

⇒ $x_n$ denotes the signal value at the $n$-th vertex in $\mathcal{V}$

⇒ Implicit ordering of vertices (same as in $A$ or $L$)

Data associated with links of $G$ ⇒ Use line graph of $G$

\[
x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}
\]
Graph signals – 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

\[
\begin{bmatrix}
0 \\
1 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

Sample patient 1 with subtype 1

\[
\begin{bmatrix}
1 \\
0 \\
0 \\
\vdots \\
0
\end{bmatrix}
\]

Sample patient 2 with subtype 1

- To understand a graph signal, the structure of $G$ must be considered
To understand and analyze $\mathbf{x}$, useful to account for $G$’s structure

Associated with $G$ is the graph-shift operator $\mathbf{S} \in \mathbb{R}^{N \times N}$

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

$\mathbf{S}$ can take nonzero values in the edges of $G$ or in its diagonal

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

$S$ 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 \( y \) is defined as \( y = Sx \), then node \( i \) can compute \( y_i \) if it has access to \( x_j \) at \( j \in \mathcal{N}(i) \).

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

- **What if** \( y = S^2x \)?
  - \( \Rightarrow \) Like powers of \( A \): neighborhoods
  - \( \Rightarrow y_i \) found using values within 2-hops

\[
[S^2]_{3,5} = S_{3,2}S_{2,5} + S_{3,4}S_{4,5}
\]
Motivation and preliminaries

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

Part II: Applications
- Sampling graph signals
- Stationarity of graph processes
- Network topology inference

Concluding remarks
Discrete Fourier Transform (DFT)

Let $\mathbf{x}$ be a temporal signal, its DFT is $\tilde{\mathbf{x}} = \mathbf{F}^H \mathbf{x}$, with $F_{kn} = \frac{1}{\sqrt{N}} e^{+j \frac{2\pi}{N} kn}$

⇒ Equivalent description, provides insights
⇒ Oftentimes, more parsimonious (bandlimited)
⇒ 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? ⇒ $S$ involved in generation/description of $x$
  ⇒ Let $S = \mathbf{VΛV}^{-1}$ be the shift associated with $G$

- The **Graph Fourier Transform (GFT)** of $x$ is defined as
  \[ \tilde{x} = \mathbf{V}^{-1}x \]

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

  ⇒ Eigenvectors $\mathbf{V} = [v_1, ..., v_N]$ are the frequency basis (atoms)

- Additional structure
  ⇒ If $S$ is normal, then $\mathbf{V}^{-1} = \mathbf{V}^H$ and $\tilde{x}_k = v_k^Hx = \langle v_k, x \rangle$
  ⇒ Parseval holds, $\|x\|^2 = \|\tilde{x}\|^2$

- **GFT** ⇒ Projection on eigenvector space of shift operator $S$
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**: $x = \sum_{k=1}^{K} \tilde{x}_k v_k$
  $\Rightarrow$ 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 $\lambda_k = \Lambda_{kk}$
  - When $\mathbf{S} = \mathbf{A}_{dc}$, we get $\lambda_k = e^{-j\frac{2\pi}{N}k}$
  - $\lambda_k$ can be viewed as frequencies!!

- In time, well-defined relation between frequency and variation
  - Higher $k$ $\Rightarrow$ higher oscillations
  - Bounds on total-variation: $TV(\mathbf{x}) = \sum_n (x_n - x_{n-1})^2$

- Q: Does this carry over for graph signals?
  - No in general, but if $\mathbf{S} = \mathbf{L}$ there are interpretations for $\lambda_k$
  - $\left\{\lambda_k\right\}_{k=1}^{N}$ 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 $\mathbf{S} = \mathbf{L}$
  \[ \mathbf{y} = \mathbf{S}\mathbf{x} \text{ is now } \mathbf{y} = \mathbf{L}\mathbf{x} \Rightarrow y_i = \sum_{j \in \mathcal{N}(i)} w_{ij}(x_i - x_j) \]
  \[ \Rightarrow j\text{-th term is large if } x_j \text{ is very different from neighboring } x_i \]
  \[ \Rightarrow y_i \text{ measures difference of } x_i \text{ relative to its neighborhood} \]

- We can also define the quadratic form $\mathbf{x}^T \mathbf{S}\mathbf{x}$
  \[ \mathbf{x}^T \mathbf{L}\mathbf{x} = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} w_{ij}(x_i - x_j)^2 \]
  \[ \Rightarrow \mathbf{x}^T \mathbf{L}\mathbf{x} \text{ quantifies the (aggregated) local variation of signal } \mathbf{x} \]
  \[ \Rightarrow \text{Natural measure of signal smoothness w.r.t. } G \]

- Q: Interpretation of frequencies $\{\lambda_k\}_{k=1}^N$ when $\mathbf{S} = \mathbf{L}$?
  \[ \Rightarrow \text{If } \mathbf{x} = \mathbf{v}_k, \text{ we get } \mathbf{x}^T \mathbf{L}\mathbf{x} = \lambda_k \Rightarrow \text{local variation of } \mathbf{v}_k \]
  \[ \Rightarrow \text{Frequencies account for local variation, they can be ordered} \]
  \[ \Rightarrow \text{Eigenvector associated with eigenvalue 0 is constant} \]
Frequencies of the Laplacian

- Laplacian eigenvalue $\lambda_k$ accounts for the local variation of $\mathbf{v}_k$
  - Let us plot some of the eigenvectors of $\mathbf{L}$ (also graph signals)
- **Ex:** gene network, $N = 10$, $k = 1, 2, 9$
- **Ex:** smooth natural images, $N = 2^{16}$, $k = 2, \ldots, 6$
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.

⇒ Targeted treatments, more suitable suggestions, etc.

Traditional approaches consider different genes to be independent.

⇒ Not ideal, as different genes may affect same metabolism.

Alternatively, consider genetic network.

⇒ Genetic profiles become graph signals on genetic network.

⇒ 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 $x_i \in \{0, 1\}^{2458}$

- 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 $\Rightarrow d(i,j) = \|x_i - x_j\|_2$
  $\Rightarrow N$-fold cross-validation error from $k$-NN classification

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

- Q: Can we do any better using graph signal processing?

- Each genetic profile $x_i$ is a graph signal on the genetic network
  $\Rightarrow$ Look at the frequency components $\tilde{x}_i$ using the GFT
  $\Rightarrow$ Use as shift operator $S$ the Laplacian of the genetic network

Example of signal $x_i$  

Frequency representation $\tilde{x}_i$
Define the distinguishing power of frequency $v_k$ as

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

Normalized difference between the mean GFT coefficient for $v_k$

$\Rightarrow$ 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 \(\tilde{x}_i\) by \(\text{diag}(\tilde{h}^p)\) where
  \[
  [\tilde{h}^p]_k = \begin{cases} 
  1, & \text{if } DP(v_k) \geq p\text{-th percentile of } DP \\
  0, & \text{otherwise}
  \end{cases}
  \]

- Then perform inverse GFT to get the graph signal \(\hat{x}_i\)
Motivation and preliminaries

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

Part II: Applications
  Sampling graph signals
  Stationarity of graph processes
  Network topology inference

Concluding remarks
A graph filter $H : \mathbb{R}^N \rightarrow \mathbb{R}^N$ is a map between graph signals. Focus on linear filters ⇒ map represented by an $N \times N$ matrix.

**DEF1:** Polynomial in $S$ of degree $L$, with coeff. $h = [h_0, \ldots, h_L]^T$

$$H := h_0S^0 + h_1S^1 + \ldots + h_LS^L = \sum_{l=0}^{L} h_l S^l \quad \text{[Sandryhaila13]}$$

**DEF2:** Orthogonal operator in the frequency domain

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

With $[\Psi]_{k,l} := \lambda_k^{l-1}$, we have $\tilde{h} = \Psi h \Rightarrow$ Defs can be rendered equivalent ⇒ More on this later, now focus on DEF1.
DEF1 says $H = \sum_{l=0}^{L} h_l S^l$

Suppose $H$ acts on a graph signal $x$ to generate $y = Hx$

If we define $x^{(l)} := S^l x = Sx^{(l-1)}$

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

$y$ is a linear combination of successive shifted versions of $x$

After introducing $S$, we stressed that $y = Sx$ can be computed locally

$x^{(l)}$ can be found locally if $x^{(l-1)}$ 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

\[ \mathbf{x} = \begin{bmatrix} -1, 2, 0, 0, 0, 0 \end{bmatrix}^T, \quad \mathbf{h} = \begin{bmatrix} 1, 1, 0.5 \end{bmatrix}^T, \quad \mathbf{y} = \left( \sum_{l=0}^{L} h_l \mathbf{S} \right) \mathbf{x} = \sum_{l=0}^{L} h_l \mathbf{x}^{(l)} \]

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

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

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

Given \( \mathbf{x} = \begin{bmatrix} -1, 2, 0, 0, 0, 0 \end{bmatrix}^T \) and \( \mathbf{h} = \begin{bmatrix} 1, 1, 0.5 \end{bmatrix}^T \) ⇒ Find \( \{ \mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \mathbf{x}^{(2)} \} \) ⇒ Find \( \mathbf{y} \)

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

\[ \mathbf{y} = 1 \mathbf{x}^{(0)} + 1 \mathbf{x}^{(1)} + 0.5 \mathbf{x}^{(2)} = \begin{pmatrix}
1.0 \\
2.5 \\
1.5 \\
1.5 \\
1.5 \\
0.0
\end{pmatrix} \]
Recalling that $S = \mathbf{V}\Lambda\mathbf{V}^{-1}$, we may write

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

The application $H\mathbf{x}$ of filter $H$ to $\mathbf{x}$ can be split into three parts

$\Rightarrow \mathbf{V}^{-1}$ takes signal $\mathbf{x}$ to the graph frequency domain $\tilde{\mathbf{x}}$

$\Rightarrow \tilde{H} := \sum_{l=0}^{L} h_l \Lambda^l$ 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 $\tilde{H}$ is diagonal, define $\tilde{H} =: \text{diag}(\tilde{h})$

$\Rightarrow \tilde{h}$ is the frequency response of the filter $H$

$\Rightarrow$ Output at frequency $k$ depends only on input at frequency $k$

$$\tilde{y}_k = \tilde{h}_k \tilde{x}_k$$
Relation between $\tilde{h}$ and $h$ in a more friendly manner?

⇒ Since $\tilde{h} = \text{diag}(\sum_{l=0}^{L} h_l \Lambda^l)$, we have that $\tilde{h}_k = \sum_{l=0}^{L} h_l \lambda_k^l$

⇒ Define the Vandermonde matrix $\Psi$ as

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

Frequency response of a graph filter

If $h$ are the coefficients of a graph filter, its frequency response is

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

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

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

⇒ Since $\Psi$ is Vandermonde, invertible as long as $\lambda_k \neq \lambda_{k'}$ for $k \neq k'$
More on the frequency response

- Since \( h = \psi^{-1} \tilde{h} \Rightarrow \) If all \( \{\lambda_k\}_{k=1}^N \) distinct, then
  \[ \Rightarrow \text{Any} \tilde{h} \text{ can be implemented with at most } L+1 = N \text{ coefficients} \]

- Since \( h = \psi \tilde{h} \Rightarrow \) If \( \lambda_k = \lambda_{k'} \), then
  \[ \Rightarrow \text{The corresponding frequency response will be the same} \tilde{h}_k = \tilde{h}_{k'} \]

- For the particular case when \( S = A_{dc} \), we have that \( \lambda_k = e^{-j \frac{2\pi}{N} (k-1)} \)

\[ \psi = \begin{pmatrix}
1 & 1 & \cdots & 1 \\
1 & e^{-j \frac{2\pi(1)(1)}{N}} & \cdots & e^{-j \frac{2\pi(1)(N-1)}{N}} \\
\vdots & \vdots & \ddots & \vdots \\
1 & e^{-j \frac{2\pi(N-1)(1)}{N}} & \cdots & e^{-j \frac{2\pi(N-1)(N-1)}{N}}
\end{pmatrix} = F^H \]

\[ \Rightarrow \text{The frequency response is the DFT of the impulse response} \]

\[ \tilde{h} = F^H h \]
Suppose that we have a signal $x$ and filter coefficients $h$.

For time signals, it holds that the output $y$ is

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

For graph signals, the output $y$ in the frequency domain is

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

The GFT for filters is different from the GFT for signals:

- Symmetry is lost, but both depend on spectrum of $S$.
- Many of the properties are not true for graphs.
- Several options to generalize operations.
Suppose that our goal is to find $h$ given $x$ and $y$

Using the previous expressions

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

In time, if we set $x = [1, 0, \ldots, 0]^T = e_1$ (i.e., $\tilde{x} = 1$), we have

$$h = F \text{diag}^{-1}(1) F^H y = y \rightarrow h \text{ is the impulse response}$$

In the graph domain

If we set $x = e_i$, then $h = \Psi^{-1} \text{diag}^{-1}(\tilde{e}_i)V^{-1}y$, where

$$\tilde{e}_i := V^{-1}e_i \equiv \text{how strongly node } i \text{ expresses each of the freqs.}$$

Problem if $\tilde{e}_i$ has zero entries

Alternatively we can get $\tilde{x} = 1$ by setting $x = V1$ and then

$$h = \Psi^{-1} \text{diag}^{-1}(\tilde{x})V^{-1}y = \Psi^{-1}V^{-1}y$$
Implementing graph filters: frequency or space

- Frequency or space?

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

- **In space:** leverage the fact that \( \mathbf{Sx} \) can be computed locally
  \( \Rightarrow \) Signal \( \mathbf{x} \) is percolated \( L \) times to find \( \{\mathbf{x}^{(l)}\}_{l=0}^{L} \)
  \( \Rightarrow \) Every node finds its own \( y_i \) by computing \( \sum_{l=0}^{L} h_l [\mathbf{x}^{(l)}]_i \)

- **Frequency implementation** useful for processing if, e.g.,
  \( \Rightarrow \) Filter bandlimited and eigenvectors easy to find
  \( \Rightarrow \) Low complexity [Anis16, Tremblay16]

- **Space definition** useful for modeling
  \( \Rightarrow \) Diffusion, percolation, opinion formation, ... (more on this soon)

- More on filter design
  \( \Rightarrow \) Chebyshev polyn. [Shuman12]; AR-MA [Isufi-Leus15]; Node-var. [Segarra15]; Time-var. [Isufi-Leus16]; Median filters [Segarra16]
Consider a linear dynamics of the form

\[ x_t - x_{t-1} = \alpha J x_{t-1} \implies x_t = (I - \alpha J)x_{t-1} \]

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

\[ [S]_{ij} = [J]_{ij} \implies x_t = (I - \alpha S)x_{t-1} \implies x_t = (I - \alpha S)^t x_0 \]

\( \Rightarrow x_t = H x_0 \), with \( H \) a polynomial of \( S \) \( \Rightarrow \) linear graph filter

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

\[ y = \sum_{t=0}^{T} \beta^t x_t \implies y = \sum_{t=0}^{T} (\beta I - \beta \alpha S)^t x_0 \]

Everything holds true if \( \alpha_t \) or \( \beta_t \) are time varying
Before finite-time dynamics (FIR filters)

Consider now a diffusion dynamics $x_t = \alpha S x_{t-1} + w$

$$x_t = \alpha^t S^t x_0 + \sum_{t'=0}^{t} \alpha^{t-t'} S^{t'} w$$

$\Rightarrow$ When $t \rightarrow \infty$: $x_\infty = (I - \alpha S)^{-1} w \Rightarrow$ 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 $M$

$$x_\infty = \prod_{m=1}^{M} (I - \alpha_m S)^{-1} w \quad x_\infty = \sum_{m=1}^{M} (I - \alpha_m S)^{-1} w$$
General linear network processes

- Combinations of all the previous are possible

\[ x_t = H^a_t(S)x_{t-1} + H^b_t(S)w \Rightarrow x_t = H^A_t(S)x_0 + H^B_t(S)w \]

\[ \Rightarrow y = x_t, \text{ sequential/parallel application, linear combination} \]

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

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

- Links with control theory (of networks and complex systems)
  - Controllability, observability
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
⇒ \( x(t) \in \mathbb{R}^{112} \) describing brain states

Does brain state variability correlate with learning?
Measuring brain state variability

- We propose three different measures capturing different time scales
  ⇒ Changes in micro, meso, and macro scales
- Micro: instantaneous changes higher than a threshold $\alpha$
  \[
  m_1(x) = \sum_{t=1}^{T} 1 \left\{ \frac{\|x(t) - x(t-1)\|_2}{\|x(t)\|_2} > \alpha \right\}
  \]
- Meso: Cluster brain states and count the changes in clusters
  \[
  m_2(x) = \sum_{t=1}^{T} 1 \{ c(t) \neq c(t-1) \}
  \]
  ⇒ where $c(t)$ is the cluster to which $x(t)$ belongs.
- Macro: Sample entropy. Measure of complexity of time series
  \[
  m_3(x) = -\log \left( \frac{\sum_t \sum_{s \neq t} 1\{\|\bar{x}_3(t) - \bar{x}_3(s)\|_\infty > \alpha\}}{\sum_t \sum_{s \neq t} 1\{\|\bar{x}_2(t) - \bar{x}_2(s)\|_\infty > \alpha\}} \right)
  \]
  ⇒ Where $\bar{x}_r(t) = [x(t), x(t+1), \ldots, x(t+r-1)]$
Diffusion as low-pass filtering

- We diffuse each time signal $x(t)$ across the brain graph

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

⇒ where Laplacian $L = V\Lambda V^{-1}$ and $\beta$ represents the diffusion rate

- Analyzing diffusion in the frequency domain

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

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

- Diffusion acts as low-pass filtering
- High freq. components are attenuated
- $\beta$ 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 $x_{null}$

\[ x_{null}(t) = x_{diff}(\pi_t) \]

$\Rightarrow$ where $\pi_t$ 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 $x$
  - **The filtered** data $x_{diff}$
  - **The null** signal $x_{null}$
Low-pass filtering reveals correlation

Correlation coeff. between learning rate and brain state variability

<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Filtered</th>
<th>Null</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m_1$</td>
<td>0.211</td>
<td>0.568</td>
<td>0.182</td>
</tr>
<tr>
<td>$m_2$</td>
<td>0.226</td>
<td>0.611</td>
<td>0.174</td>
</tr>
<tr>
<td>$m_3$</td>
<td>0.114</td>
<td>0.382</td>
<td>0.113</td>
</tr>
</tbody>
</table>

Correlation is clear when the signal is filtered

⇒ Result for original signal similar to null signal

Scatter plots for original, filtered, and null signals ($m_2$ variability)
Motivation and preliminaries

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

Part II: Applications
- Sampling graph signals
- Stationarity of graph processes
- 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
Motivation and preliminaries

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

Part II: Applications
- Sampling graph signals
- Stationarity of graph processes
- Network topology inference

Concluding remarks
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 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
  ⇒ How to find \( x \in \mathbb{R}^N \) using \( P < N \) observations?

- Our focus on bandlimited signals, but other models possible
  ⇒ \( \tilde{x} = V^{-1}x \) sparse
  ⇒ \( x = \sum_{k \in \mathcal{K}} \tilde{x}_k v_k \), with \( |\mathcal{K}| = K < N \)
  ⇒ \( S \) involved in generation of \( x \)
  ⇒ Agnostic to the particular form of \( S \)

- Two sampling schemes were introduced in the literature
  ⇒ **Selection** [Anis14, Chen15, Tsitsvero15, Puy15, Wang15]
  ⇒ **Aggregation** [Segarra15], [Marques15]
  ⇒ **Hybrid** scheme combining both ⇒ **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

Both strategies coincide for time signals but not for general graphs
⇒ Give rise to selection and aggregation sampling
Intuitive generalization to graph signals

⇒ \( C \in \{0, 1\}^{P \times N} \) (matrix \( P \) rows of \( I_N \))

⇒ Sampled signal is \( \bar{x} =Cx \)

Goal: recover \( x \) based on \( \bar{x} \)

⇒ Assume that the support of \( K \) is known (w.l.o.g. \( K = \{k\}_{k=1}^{K} \))

⇒ Since \( \tilde{x}_k = 0 \) for \( k > K \), define \( \tilde{x}_K := [\tilde{x}_1, \ldots, \tilde{x}_K]^T = E_K^T\tilde{x} \)

Approach: use \( \bar{x} \) to find \( \tilde{x}_K \), and then recover \( x \) as

\[
x = V(E_K\tilde{x}_K) = (VE_K)\tilde{x}_K = V_K\tilde{x}_K
\]
Selection sampling: Recovery

- Number of samples \( P \geq K \)

\[
\tilde{x} = Cx = CV_K \tilde{x}_K
\]

\( \Rightarrow (CV_K) \) submatrix of \( V \)

\[
\text{Recovery of selection sampling}
\]

If \( \text{rank}(CV_K) \geq K \), \( x \) can be recovered from the \( P \) values in \( \tilde{x} \) as

\[
x = V_K \tilde{x}_K = V_K(CV_K)^\dagger \tilde{x}
\]

- With \( P = K \), hard to check invertibility (by inspection)
  \( \Rightarrow \) Columns of \( V_K(CV_K)^{-1} \) are the interpolators

- In time \( (S = A_{dc}) \), if the samples in \( C \) are equally spaced
  \( \Rightarrow (CV_K) \) is Vandermonde (DFT) and \( V_K(CV_K)^{-1} \) are sincs
Aggregation sampling: Definition

- Idea: incorporating $S$ to the sampling procedure
  - Reduces to classical sampling for time signals

- Consider shifted (aggregated) signals $y^{(l)} = S^l x$
  - $y^{(l)} = Sy^{(l-1)} \Rightarrow$ found sequentially with only local exchanges

- Form $y_i = [y_i^{(0)}, y_i^{(1)}, ..., y_i^{(N-1)}]^T$ (obtained locally by node $i$)

- The sampled signal is
  $$\bar{y}_i = C y_i$$

- Goal: recover $x$ based on $\bar{y}_i$
Aggregation sampling: Recovery

- **Goal:** recover \( x \) based on \( \bar{y}_i \) ⇒ Same approach than before
  ⇒ Use \( \bar{y}_i \) to find \( \tilde{x}_K \), and then recover \( x \) as \( x = V_K \tilde{x}_K \)

- Define \( \bar{u}_i := V_K^T e_i \) and recall \( \psi_{kl} = \lambda_k^{l-1} \)

**Recovery of aggregation sampling**

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

\[
x = V_K \tilde{x}_K = V_K \text{diag}^{-1}(\bar{u}_i)(C\psi^T E_K)^{-1}\bar{y}_i
\]

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

- If \( C = E_K^T \), node \( i \) can recover \( x \) with info from \( K - 1 \) hops!
  ⇒ Node \( i \) has to be able to capture frequencies in \( \mathcal{K} \)
  ⇒ The frequencies have to distinguishable

- **Bandlimited signals:** Signals that can be well estimated locally
Aggregation and selection sampling: Example

- In time ($S = A_{dc}$), selection and aggregation are equivalent
  ⇒ Differences for a more general graph?

- Erdős-Rényi
  $p = 0.2$, $S = A$, $K = 3$, non-smooth

- First 3 observations at node 4: $y_4 = [0.55, 1.27, 2.94]^T$
  ⇒ $[y_4]_1 = x_4 = -0.55$, $[y_4]_2 = x_2 + x_3 + x_5 + x_6 + x_7 = 1.27$
  ⇒ For this example, any node guarantees recovery
  ⇒ 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 $S$)

- Three extensions:
  - Sampling in the presence of noise
  - Unknown frequency support
  - Space-shift sampling (hybrid)
Presence of noise

- Linear observation model: \( \bar{z}_i = C\Psi_i\bar{x}_K + Cw_i \) and \( x = V_K\bar{x}_K \)

- BLUE interpolation (\( \Psi_i \) either selection or aggregation)

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

⇒ If \( P = K \), then \( \hat{x}^{(i)} = V_K (C\Psi_i)^{-1} \bar{z}_i \)

- Error covariances (\( R^{(i)}_e, \tilde{R}^{(i)}_e \)) in closed form ⇒ Noise covariances?
  ⇒ Colored, different models: white noise in \( z_i \), in \( x \), or in \( \bar{x}_K \)

- Metric to optimize?

  ⇒ \( \text{trace}(R^{(i)}_e), \lambda_{\text{max}}(R^{(i)}_e), \log \det(\tilde{R}^{(i)}_e), \left[ \text{trace} \left( \tilde{R}^{(i)}_e^{-1} \right) \right]^{-1} \)

- Select \( i \) and \( C \) to min. error ⇒ Depends on metric and noise [Marques16]
Unknown frequency support

- Falls into the class of sparse reconstruction: observation matrix?
  - Selec. ⇒ submatrix of unitary $V_K$
  - Aggr. ⇒ Vander. $\times$ diag
    $[u_i]_k \neq 0$ and $\lambda_k \neq \lambda_{k'}$ ⇒ full-spark

- Joint recovery and support identification (noiseless)
  $\tilde{x}^* := \arg\min_{\tilde{x}} \|\tilde{x}\|_0$
  s.t. $Cy_i = C\psi_i\tilde{x}$

- If full spark ⇒ $P = 2K$ samples suffice
  - Different relaxations are possible
  - Conditioning will depend on $\psi_i$ (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: \( A, (I - A) \) and \( \frac{1}{2}A^2 \)
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 \(\tilde{U} := [\text{diag}(\tilde{u}_1), ..., \text{diag}(\tilde{u}_N)]^T\), the sampled signal is

\[
\tilde{z} = C \left( I \otimes (\Psi^T E_K) \right) \tilde{U} \tilde{x}_K + Cw
\]

- 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
  - Graph: average flows in 2007-2010, \( S = A \)
  - Signal \( x \): production in 2011
  - \( x \) is approximately bandlimited with \( K = 4 \)
Sampling the US economy: Results

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

- Setup 2: we try different shift-space strategies

<table>
<thead>
<tr>
<th>Sampling strategy</th>
<th>Min. error</th>
<th>Median error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[x]_i$</td>
<td>.0035</td>
<td>.019</td>
</tr>
<tr>
<td>$[Sx]_i$</td>
<td>.0039</td>
<td>4.2</td>
</tr>
<tr>
<td>$[S^2x]_i$</td>
<td>.0035</td>
<td>.030</td>
</tr>
<tr>
<td>$[S^3x]_i$</td>
<td>.0035</td>
<td>.0055</td>
</tr>
<tr>
<td>$[x]_j$</td>
<td>.0035</td>
<td>.0040</td>
</tr>
<tr>
<td>$[Sx]_j$</td>
<td>.0035</td>
<td>.039</td>
</tr>
</tbody>
</table>
More on sampling graph signals

- **Beyond bandlimitedness**
  - Smooth signals [Chen15]
  - Parsimonious in kernelized domain [Romero16]

- **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]
Motivation and preliminaries

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

Part II: Applications
   Sampling graph signals
   Stationarity of graph processes
   Network topology inference

Concluding remarks
Motivation and context

We frequently encounter **stochastic** processes

Statistical SP ⇒ tools for their understanding

**Stationarity** facilitates the analysis of random signals in time
⇒ Statistical properties are time-invariant

We seek to **extend** the concept of **stationarity** to **graph processes**
⇒ Network data and irregular domains motivate this
⇒ Lack of regularity leads to multiple definitions

Classical SSP can be generalized: **spectral estimation, periodograms,**
⇒ Better understanding and estimation of graph processes
⇒ Related works: [Girault 15], [Perraudin 16]

*Segarra, Marques, Leus, Ribeiro, *Stationary Graph Processes: Nonparametric Spectral Estimation*, SAM16
*Marques, Segarra, Leus, Ribeiro, *Stationary Graph Processes and Spectral Estimation*, IEEE TSP (sub.)
Prelimaries: Weak stationarity in time

(1) **Correlation** of stationary discrete time signals is invariant to shifts

\[ \mathbf{C}_x := \mathbb{E}[\mathbf{x}\mathbf{x}^H] = \mathbb{E}[\mathbf{x}^H(n-l)\mathbf{x}(n-l)] = \mathbb{E}[\mathbf{S'}x(\mathbf{S'}x)^H] \]

(2) Signal is the output of a LTI filter \( \mathbf{H} \) excited with white noise \( \mathbf{w} \)

\[ \mathbf{x} = \mathbf{Hw}, \quad \text{with} \quad \mathbb{E}[\mathbf{ww}^H] = \mathbf{I} \]

(3) The covariance matrix \( \mathbf{C}_x \) is diagonalized by the Fourier matrix

\[ \mathbf{C}_x = \mathbf{F}\text{diag}(\mathbf{p})\mathbf{F}^H \]

- The process has a power spectral density \( \Rightarrow \mathbf{p} := \text{diag}(\mathbf{F}^H\mathbf{C}_x\mathbf{F}) \)
- Each of these definitions can be generalized to graph signals
Stationary graph processes: Shifts

Definition (shift invariance)
Process $x$ is weakly stationary with respect to $S$ if and only if $(b > c)$

$$
\mathbb{E}\left[(S^a x)(S^H)^b x^H\right] = \mathbb{E}\left[(S^{a+c} x)(S^H)^{b-c} x^H\right]
$$

- Use $a$ and $b$ shifts as reference. Shift by $c$ forward and backward
  ⇒ Signal is stationary if these shifts do not alter its covariance

- It reduces to $\mathbb{E}[xx^H] = \mathbb{E}[S'x(S'x)^H]$ when $S$ is a directed cycle
- Time shift is orthogonal, $S^H = S^{-1}$ ($a = 0$, $b = N$ and $c = l$)

- Need reference shifts because $S$ can change energy of the signal
Definition (filtering of white noise)

Process $\mathbf{x}$ is weakly stationary with respect to $\mathbf{S}$ if it can be written as the output of linear shift invariant filter $\mathbf{H}$ with white input $\mathbf{w}$

$$\mathbf{x} = \mathbf{H}\mathbf{w}, \quad \text{with } \mathbb{E}[\mathbf{w}\mathbf{w}^H] = \mathbf{I}$$

- The filter $\mathbf{H}$ is linear shift invariant if $\Rightarrow \mathbf{H}(\mathbf{S}\mathbf{x}) = \mathbf{S}(\mathbf{H}\mathbf{x})$

- Equivalently, $\mathbf{H}$ polynomial on the shift operator $\Rightarrow \mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{S}^l$

- Filter $\mathbf{H}$ determines color $\Rightarrow \mathbf{C}_x = \mathbb{E}[(\mathbf{H}\mathbf{w})(\mathbf{H}\mathbf{w})^H] = \mathbf{H}\mathbf{H}^H$
Definition (Simultaneous diagonalization)
Process $x$ is weakly stationary with respect to $S$ if the covariance $C_x$ and the shift $S$ are simultaneously diagonalizable

$$S = V \Lambda V^H \implies C_x = V \text{diag}(p) V^H$$

- Equivalent to time definition because $F$ diagonalizes cycle graph
- The process has a power spectral density $\implies p := \text{diag}(V^H C_x V)$
Equivalence of definitions and PSD

- Have introduced three equally valid definitions of weak stationarity
  ⇒ They are different but, pleasingly, equivalent

**Proposition**

*Process $x$ has shift invariant correlation matrix $⇔$ it is the output of a linear shift invariant filter $⇔$ Covariance jointly diagonalizable with shift*

- Shift and Filtering $⇒$ How stationary signals look like (local invariance)

- Simultaneous Diagonalization $⇒$ A PSD exists $⇒ p := \text{diag}(V^H C_x V)$
  $⇒$ The PSD collects the eigenvalues of $C_x$ and is nonnegative

**Proposition**

*Let $x$ be stationary in $S$ and define the process $\tilde{x} := V^H x$. Then, it holds that $\tilde{x}$ is uncorrelated with covariance matrix $C_{\tilde{x}} = \mathbb{E} [\tilde{x} \tilde{x}^H] = \text{diag}(p)$.*
Weak stationary graph processes examples

Example (White noise)
- White noise \( w \) is stationary in any graph shift \( S = \mathbf{V}\Lambda\mathbf{V}^H \)
- Covariance \( C_w = \sigma^2 I \) simultaneously diagonalizable with all \( S \)

Example (Covariance matrix graphs and Precision matrices)
- Every process is stationary in the graph defined by its covariance matrix
- If \( S = C_x \), shift \( S \) and covariance \( C_x \) diagonalized by same basis
- Process is also stationary on precision matrix \( S = C_x^{-1} \)

Example (Heat diffusion processes and ARMA processes)
- Heat diffusion process in a graph \( \Rightarrow x = \alpha_0(I - \alpha L)^{-1}w \)
- Stationary in \( L \) since \( \alpha_0(I - \alpha L)^{-1} \) is a polynomial on \( L \)
- Any autoregressive moving average (ARMA) process on a graph
Power spectral density examples

Example (White noise)

- Power spectral density \( \Rightarrow p = \text{diag}(V^H(\sigma^2 I)V) = \sigma^2 1 \)

Example (Covariance matrix graphs and Precision matrices)

- Power spectral density \( \Rightarrow p = \text{diag}(V^H(V \Lambda V^H)V) = \text{diag}(\Lambda) \)

Example (Heat diffusion processes and ARMA processes)

- Power spectral density \( \Rightarrow p = \text{diag}\left[\alpha_0^2 (I - \alpha \Lambda)^{-2}\right] \)
Given a process $x$, the covariance of $\tilde{x} = V^H x$ is given by

$$C_{\tilde{x}} := \mathbb{E} \left[ \tilde{x}\tilde{x}^H \right] = \mathbb{E} \left[ (V^H x)(V^H x)^H \right] = \text{diag}(p)$$

Periodogram $\Rightarrow$ Given samples $\{x_r\}_{r=1}^R$, average GFTs of samples

$$\hat{p}_{pg} := \frac{1}{R} \sum_{r=1}^R |\tilde{x}_r|^2 = \frac{1}{R} \sum_{r=1}^R |V^H x_r|^2$$

Correlogram $\Rightarrow$ Replace $C_x$ in PSD definition by sample covariance

$$\hat{p}_{cg} := \text{diag} \left( V^H \hat{C}_x V \right) := \text{diag} \left[ V^H \left[ \frac{1}{R} \sum_{r=1}^R x_r x_r^H \right] V \right]$$

Periodogram and correlogram lead to identical estimates $\hat{p}_{pg} = \hat{p}_{cg}$
**Theorem**

If the process $x$ is Gaussian, periodogram estimates have bias and variance

- **Bias** $\Rightarrow b_{pg} := E[\hat{p}_{pg}] - p = 0$

- **Variance** $\Rightarrow \Sigma_{pg} := E[(\hat{p}_{pg} - p)(\hat{p}_{pg} - p)^H] = \frac{2}{R} \text{diag}^2(p)$

- The periodogram is **unbiased** but the **variance** is not too good $\Rightarrow$ **Quadratic in** $p$. Same as time processes

- Alternative nonparametric methods to reduce variance
  $\Rightarrow$ **Average windowed periodogram**
  $\Rightarrow$ **Filterbanks**
  $\Rightarrow$ **Bias - variance tradeoff** characterized [Marques16,Segarra16]
Parametric PSD estimation

- PG and CG are examples of non-parametric estimators

- **Parametric ARMA estimation**
  - Model \( \mathbf{x} = \sum_{l=0}^{L} h_l \mathbf{S}^l \mathbf{w} \) with \( \mathbf{w} \) white
  - PSD is \( \mathbf{p}_x(h) = |\mathbf{\Psi} h|^2 \)
  - Given \( \mathbf{x}_r \), compute \( \hat{\mathbf{p}}_{pg} \) and find \( \hat{h} = \text{arg min}_h d(\hat{\mathbf{p}}_{pg}, \mathbf{p}_x(h)) \)
  - Set \( \hat{\mathbf{p}}_{MA} = \mathbf{p}_x(\hat{h}) = |\mathbf{\Psi} \hat{h}|^2 \)
  - General estimation problem nonconvex
  - Particular cases (\( \mathbf{S} \succeq 0 \) and \( \mathbf{h} \succeq 0 \)) tractable

- Other parametric models (sum of frequency basis) possible too
Average periodogram

- MSE of periodogram as a function of the nr. of observations $R$
- Baseline ER random graph ($N = 100$ and $p = 0.05$) and $S = A$
- Observe filtered white Gaussian noise and estimate PSD

- Normalized MSE evolves as $2/R$ as expected
  ⇒ Invariant to size, topology, and PSD
- Same behavior observed in non-Gaussian processes (theory not valid)
Windowed average periodogram

- Performance of local windows and random windows
- Block stochastic graph ($N = 100$, 10 communities) and small world
- Process filters white noise with different number of taps

The use of windows introduces bias but reduces total error (MSE)

Local windows work better than random windows

⇒ Advantage of local windows is larger for local processes
Opinion source identification

- Opinion diffusion in Zachary’s karate club network \((N = 34)\)
- Observed opinion \(x\) obtained by diffusing sparse white rumor \(w\)

- Given \(\{x_r\}_{r=1}^{R}\) generated from unknown \(\{w_r\}_{r=1}^{R}\) 
  \(\Rightarrow\) Diffused through filter of unknown nonnegative coefficients \(\beta\)
- Goal \(\Rightarrow\) Identify the support of each rumor \(w_r\)
- First \(\Rightarrow\) Estimate \(\beta\) from Moving Average PSD estimation
- Second \(\Rightarrow\) Solve \(R\) sparse linear regressions to recover \(\text{supp}(w_r)\)

![Source identification error graph](image-url)
PSD of face images

- PSD estimation for spectral signatures of faces of different people
- 100 grayscale face images \( \{x_i\}_{i=1}^{100} \in \mathbb{R}^{10304} \) (10 images \( \times \) 10 people)

- Consider \( x_i \) as realization graph process that is Stationary on \( \hat{C}_x \)
- Construct \( \hat{C}_x^{(j)} = V^{(j)} \Lambda^{(j)} V^H(j) \) based on images of person \( j \)

- Process of person \( j \) approximately stationary in \( \hat{C}_x \) (left)
- Use windowed average periodogram to estimate PSD of new face
Stationarity: Takeaways

- Extended the notion of weak stationarity for graph processes
- **Three definitions** inspired in stationary time processes
  - Shown all of them to be equivalent

- Defined power spectral density and studied its estimation
- Generalized classical non-parametric estimation methods
  - Periodogram and correlogram where shown to be equivalent
  - Windowed average periodogram, filter banks
- Generalized classical ARMA parametric estimation methods
  - Particular cases tractable

- **Extensions**
  - Other parametric schemes
  - Space-time variation
Motivation and preliminaries

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

Part II: Applications
  Sampling graph signals
  Stationarity of graph processes
  Network topology inference

Concluding remarks
Motivation and context

- **Network topology inference** from nodal observations [Kolaczyk’09]
  - Approaches use Pearson correlations to construct graphs [Brovelli04]
  - Partial correlations and conditional dependence [Friedman08, Karanikolas16]

- Key in neuroscience [Sporns’10]
  - 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]

†Segarra, Marques, Mateos, Ribeiro, *Network Topology Identification from Spectral Templates*, IEEE SSP16
‡Segarra, Marques, Mateos, Ribeiro, *Network Topology Inference from Spectral Templates*, IEEE TSP (sub.)
Structure from graph stationarity

- Given a set of signals \( \{x_r\}_{r=1}^R \) find \( S \)
  - We view signals as samples of random graph process \( x \)
  - \( AS. \ x \) is stationary in \( S \)

- Equivalent to “\( x \) is the linear diffusion of a white input”
  \[
  x = \alpha_0 \prod_{l=1}^{\infty} (I - \alpha_I S)w = \sum_{l=0}^{\infty} \beta_l S^l w
  \]
  - Examples: Heat diffusion, structural equation models

- We say the graph shift \( S \) explains the structure of signal \( x \)

- Key point after assuming stationarity: eigenvectors of the covariance
The covariance matrix of the stationary signal $x = Hw$ is

$$C_x = \mathbb{E} \left[ xx^H \right] = H \mathbb{E} \left[ (ww^H) \right] H^H = HH^H$$

Since $H$ is diagonalized by $V$, so is the covariance $C_x$

$$C_x = V \left| \sum_{l=0}^{L-1} h_l \Lambda^l \right|^2 V^H = V \text{diag}(p) V^H$$

Any shift with eigenvectors $V$ can explain $x$

⇒ $G$ and its specific eigenvalues have been obscured by diffusion

**Observations**

(a) There are many shifts that can explain a signal $x$

(b) Identifying the shift $S$ 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
We propose a **two-step approach** for graph topology identification.

- **STEP 1:** Identify the eigenvectors of the shift
- **STEP 2:** Identify eigenvalues to obtain a suitable shift

**Beyond diffusion** ⇒ alternative sources for spectral templates $V$

⇒ Graph sparsification, network deconvolution,...
STEP 1: Other sources of spectral templates

1) Graph sparsification
   - Goal: given $S_f$ find sparser $S$ with same eigenvectors
     $\Rightarrow$ Find $S_f = V_f \Lambda_f V_f^H$ and set $V = V_f$
     $\Rightarrow$ Often times referred to as network deconvolution problem

2) Nodal relation assumed by a given transform
   - GSP: decompose $S = \mathbf{V} \mathbf{\Lambda} \mathbf{V}^H$ and set $\mathbf{V}^H$ as GFT
   - SP: some transforms $\mathbf{T}$ known to work well on specific data
   - Goal: given $\mathbf{T}$, set $\mathbf{V}^H = \mathbf{T}$ and identify $S$ $\Rightarrow$ intuition on data relation

DCTs: i–iii

3) Implementation of linear network operators
   - Goal: distributed implementation of linear operator $\mathbf{B}$ via graph filter
     $\Rightarrow$ Feasible if $\mathbf{B}$ and $S$ share eigenvectors $\Rightarrow$ Like 1) with $S_f = \mathbf{B}$
STEP 2: Obtaining the eigenvalues

- Given $V$, there are many possible $S = V \text{diag}(\lambda) V^H$
  
  ⇒ We can use extra knowledge/assumptions to choose one graph
  
  ⇒ Of all graphs, select one that is optimal in some sense

\[
S^* := \arg\min_{S,\lambda} f(S, \lambda) \quad \text{s. to} \quad S = \sum_{k=1}^{N} \lambda_k v_k v_k^H, \quad S \in S
\]  

- Set $S$ contains all admissible scaled adjacency matrices

\[
S := \{S \mid S_{ij} \geq 0, \quad S \in \mathcal{M}^N, \quad S_{ii} = 0, \quad \sum_j S_{1j} = 1\}
\]

⇒ Can accommodate Laplacian matrices as well

- Problem is convex if we select a convex objective $f(S, \lambda)$
  
  ⇒ Minimum energy ($f(S) = \|S\|_F$), Fast mixing ($f(\lambda) = -\lambda_2$)
The feasibility set in (1) is generally small ⇒ Why?
⇒ We search over $\lambda \in \mathbb{R}^N$, we have $N$ linear constraints $S_{ii} = 0$

This helps in the optimization, to be rigorous
⇒ Define $W := V \odot V$ where $\odot$ is the Khatri-Rao product
⇒ Denote by $D$ the index set such that $\text{vec}(S)_D = \text{diag}(S)$

Assume that (1) is feasible, then it holds that $\text{rank}(W_D) \leq N - 1$. If $\text{rank}(W_D) = N - 1$, then the feasible set of (1) is a singleton.

Convex feasibility set ⇒ Search for the optimal solution may be easy

Simulations will show that $\text{rank}(W_D) = N - 1$ arises in practice
Sparse recovery

- Whenever the feasibility set of (1) is non-trivial
  \[ f(S, \lambda) \text{ determines the features of the recovered graph} \]

Ex: Identify the sparsest shift $S^*_0$ that explains observed signal structure
  \[ f(S, \lambda) = \|S\|_0 \]

\[
S^*_0 = \operatorname{argmin}_{S, \lambda} \|S\|_0 \quad \text{s. to} \quad S = \sum_{k=1}^{N} \lambda_k v_k v_k^T, \quad S \in S
\]

- Problem is not convex, but can relax to $\ell_1$ norm minimization

\[
S^*_1 := \operatorname{argmin}_{S, \lambda} \|S\|_1 \quad \text{s. to} \quad S = \sum_{k=1}^{N} \lambda_k v_k v_k^H, \quad S \in S
\]

- Does the solution $S^*_1$ coincide with the $\ell_0$ solution $S^*_0$?
Recovery guarantee

- Denoting by $m_i^T$ the $i$-th row of $M := (I - WW^\dagger)_{Dc}$
  - Construct $R := [m_2 - m_1, \ldots, m_{N-1} - m_1, m_N, \ldots, m|_{Dc}]^T$
  - Denote by $\mathcal{K}$ the indices of the support of $s_0^* = \text{vec}(S_0^*)$

$S_1^*$ and $S_0^*$ coincide if the two following conditions are satisfied:
1) $\text{rank}(R_{\mathcal{K}}) = |\mathcal{K}|$; and
2) There exists a constant $\delta > 0$ such that

$$\psi_R := \|I_{\mathcal{K}}(\delta^{-2}RR^T + I_{\mathcal{K}}^T I_{\mathcal{K}})^{-1} I_{\mathcal{K}}^T\|_\infty < 1.$$ 

- Cond. 1) ensures uniqueness of solution $S_1^*$
- Cond. 2) guarantees existence of a dual certificate for $\ell_0$ optimality
Noisy and incomplete spectral templates

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

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

- Recovery result similar to the noiseless case can be derived
  
  ⇒ Conditions under which we are guaranteed $d(S^*, S_0^*) \leq C\epsilon$

- Partial access to $V$ ⇒ Only $K$ known eigenvectors $[v_1, \ldots, v_K]$

  \[
  \min_{\{S, S_{\bar{K}}, \lambda\}} \|S\|_1 \quad \text{s. to} \quad S = S_{\bar{K}} + \sum_{k=1}^{K} \lambda_k v_k v_k^T, \quad S \in S, \quad S_{\bar{K}} v_k = 0
  \]

- Incomplete and noisy scenarios can be combined
Topology inference in random graphs

- Erdős-Rényi graphs of varying size $N \in \{10, 20, \ldots, 50\}$
  - 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 $\text{rank}(W_D)$ (histogram $N=10, p=0.2$)
  - When rank is $N - 1$, recovery is guaranteed
  - 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
  - Feasible set is not a singleton
  - Cond. 1) in sparse recovery theorem is satisfied

- Noiseless case: $\ell_1$ norm guarantees recovery as long as $\psi_R < 1$

- Condition is sufficient but not necessary
  - 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 $\hat{V}$
  ⇒ Obtained from sample covariances of diffused signals

<table>
<thead>
<tr>
<th>Number of Observations</th>
<th>Recovery error</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10^3$</td>
<td>0.3</td>
</tr>
<tr>
<td>$10^4$</td>
<td>0.25</td>
</tr>
<tr>
<td>$10^5$</td>
<td>0.2</td>
</tr>
<tr>
<td>$10^6$</td>
<td>0.15</td>
</tr>
</tbody>
</table>

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

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

![Graph displaying recovery error vs. number of spectral templates]

- Recovery error decreases with increasing nr. of spectral templates
  - 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 comparison](image)

- Graphical lasso *implicitly assumes a filter* $H_1 = (\rho I + S)^{-1/2}$
  - For this filter spectral templates work, but not as well (MLE)

- For general diffusion filters $H_2$ 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
  - **Discard indirect relations** that can be explained by direct ones
- Use eigenvectors $\hat{V}$ of given network as noisy templates
- Infer **contact between amino-acid residues** in BPT1 BOVIN
  - Use mutual information of amino-acid covariation as input

---

**Ground truth**

**Mutual info.**

**Network deconv.**

**Our approach**

- Network deconvolution assumes a specific filter model [Feizi13]
  - We achieve better performance by being agnostic to this
Network topology inference cornerstone problem in Network Science

Most GSP works analyze how $S$ affect signals and filters

Here, reverse path: How to use GSP to infer the graph topology?

Our GSP approach to network topology inference

Two step approach: i) Obtain $V$; ii) Estimate $S$ given $V$

How to obtain the spectral templates $V$

Based on covariance of diffused signals

Other sources too: net operators, data transforms

Infer $S$ via convex optimization

Objectives promotes desirable properties

Constraints encode structure a priori info and structure

Formulations for perfect and imperfect templates

Sparse recovery results for both adjacency and Laplacian
Wrapping up

Motivation and preliminaries

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

Part II: Applications
- Sampling graph signals
- Stationarity of graph processes
- Network topology inference

Concluding remarks
Concluding remarks

- **Network science** and big data pose new challenges
  - \( \Rightarrow \) **GSP** can contribute to solve some of those challenges
  - \( \Rightarrow \) Well suited for **network (diffusion) processes**

- **Central elements in GSP**: graph-shift operator and Fourier transform

- **Graph filters**: operate graph signals
  - \( \Rightarrow \) Polynomials of the shift operator that can be implemented locally

- **Network diffusion/percolations processes via graph filters**
  - \( \Rightarrow \) Successive/parallel combination of local linear dynamics
  - \( \Rightarrow \) Possibly time-varying diffusion coefficients
  - \( \Rightarrow \) Accurate to model certain setups
  - \( \Rightarrow \) GSP yields insights on how those processes behave
Concluding remarks

- **GSP results** can be applied to solve practical problems
  - Sampling, interpolation (network control)
  - Input and system ID (rumor 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:
  - Space-time variation
  - Changing topologies
  - Nonlinear approaches
  - Local, reduced-complexity algorithms

- Thanks!
  - If you have questions, feel free to contact me by e-mail antonio.garcia.marques@urjc.es or any of the other authors.
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.
Sampling bandlimited graph signals


Interpolating graph signals


Graph filter design and network operators


**Blind graph deconvolution**


**GSP-based network topology inference**


References: our work

**Stationary graph processes**

**Median graph filters**
General references


Filtering


Sampling


Interpolation and reconstruction


**Topology inference**


**Stationarity**
