Network Science analytics

Online social media  Internet  Clean energy and grid analytics

- **Desiderata**: Process, analyze and learn from network data [Kolaczyk’09]
Network Science analytics

- **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 $\mathcal{V}$
  - The object of study is a graph signal

- Ex: Opinion profile, buffer congestion levels, neural activity, epidemic

Online social media  | Internet  | Clean energy and grid analytics
Motivating examples – Graph signals

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

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

▶ **As.**: Signal properties related to topology of $G$ (e.g., smoothness)

⇒ **Algorithms** that fruitfully leverage this relational structure
Graph signals

Consider a graph $G(V, E)$. Graph signals are mappings $x : V \to \mathbb{R}$

$\Rightarrow$ Defined on the vertices of the graph (data tied to nodes)

May be represented as a vector $x \in \mathbb{R}^N$

$\Rightarrow x_n$ denotes the signal value at the $n$-th vertex in $V$

$\Rightarrow$ Implicit ordering of vertices

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$
Graph-shift operator

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

- Graph $G$ is endowed with a **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

- **Ex:** Adjacency $\mathbf{A}$, degree $\mathbf{D}$, and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices
- **S** is a linear operator that can be computed locally at the nodes in $\mathcal{V}$

- Consider the graph signal $y = Sx$ and node $i$'s neighborhood $\mathcal{N}_i$
  \[ y_i = \sum_{j \in \mathcal{N}_i} S_{ij} x_j, \quad i \in \mathcal{V} \]

- Recall $S_{ij} \neq 0$ only if $i = j$ or $(j, i) \in E$

- If $y = S^2x \Rightarrow y_i$ found using values $x_j$ within 2 hops
- **As.:** $S$ related to generation (description) of the signals of interest

  $\Rightarrow$ Spectrum of $S = V \Lambda V^{-1}$ will be especially useful to analyze $x$

- The **Graph Fourier Transform (GFT)** of $x$ is defined as

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

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

  $x = V \tilde{x}$

  $\Rightarrow$ Eigenvectors $V = [v_1, ..., v_N]$ are the frequency basis (atoms)

- **Ex:** For the directed cycle (temporal signal) $\Rightarrow$ GFT $\equiv$ DFT
A graph filter $H : \mathbb{R}^N \to \mathbb{R}^N$ is a map between graph signals. Focus on linear filters, which are maps represented by an $N \times N$ matrix.

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

**Graph filter [Sandryhaila-Moura’13]**

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

Key properties: shift-invariance and distributed implementation.

$H(Sx) = S(Hx)$, no other can be linear and shift-invariant. Each application of $S$ only local info $\Rightarrow$ only $L$-hop info for $y = Hx$. 
Frequency response of a graph filter

- Using $S = V \Lambda V^{-1}$, filter is $H = \sum_{l=0}^{L} h_l S^l = V \left( \sum_{l=0}^{L} h_l \Lambda^l \right) V^{-1}$
Frequency response of a graph filter

- Using \( S = V \Lambda V^{-1} \), filter is
  \[
  H = \sum_{l=0}^{L} h_l S' = V \left( \sum_{l=0}^{L} h_l \Lambda^l \right) V^{-1}
  \]

- Since \( \Lambda^l \) are diagonal, use GFT-iGFT to write \( y = Hx \) as
  \[
  \tilde{y} = \text{diag}(\tilde{h})\tilde{x}
  \]

  \( \Rightarrow \) Output at frequency \( k \) depends only on input at frequency \( k \)

- Frequency response of the filter \( H \) is \( \tilde{h} = \Psi h \), with Vandermonde \( \Psi \)

  \[
  \Psi := 
  \begin{pmatrix}
  1 & \lambda_1 & \ldots & \lambda_1^L \\
  \vdots & \vdots & \ddots & \vdots \\
  1 & \lambda_N & \ldots & \lambda_N^L 
  \end{pmatrix}
  \]
Using $S = V\Lambda V^{-1}$, filter is $H = \sum_{l=0}^{L} h_{l} S^{l} = V \left( \sum_{l=0}^{L} h_{l} \Lambda^{l} \right) V^{-1}$.

Since $\Lambda^{l}$ are diagonal, use GFT-iGFT to write $y = Hx$ as $\tilde{y} = \text{diag}(\tilde{h})\tilde{x}$.

⇒ Output at frequency $k$ depends only on input at frequency $k$.

Frequency response of the filter $H$ is $\tilde{h} = \Psi h$, with Vandermonde $\Psi$:

$$\Psi := \begin{pmatrix} 1 & \lambda_{1} & \ldots & \lambda_{1}^{L} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \lambda_{N} & \ldots & \lambda_{N}^{L} \end{pmatrix}$$

GFT for signals ($\tilde{x} = V^{-1}x$) and filters ($\tilde{h} = \Psi h$) is different.
Q: Upon observing a graph signal $y$, how was this signal generated?

Postulate the following generative model:

1. An originally sparse signal $x = x(0)$
2. Diffused via linear graph dynamics $S$
   
   $x(l) = Sx(l-1)$
3. Observed $y$ is a linear combination of the diffused signals
   
   $y = \sum_{l=0}^{L} h_l x(l) = Hx$

Model: Observed network process as output of a graph filter.
Q: Upon observing a graph signal $y$, how was this signal generated?

Postulate the following generative model:

- An originally sparse signal $x = x^{(0)}$
- Diffused via linear graph dynamics $S$ $\Rightarrow$ $x^{(l)} = Sx^{(l-1)}$
- Observed $y$ is a linear combination of the diffused signals $x^{(l)}$

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

Model: Observed network process as output of a graph filter

- View few elements in $\text{supp}(x) =: \{i : x_i \neq 0\}$ 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
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 $S$, can we find $x$ and the combination weights $h$ from $y = Hx$?
    - Extends classical blind deconvolution to graphs
Leverage frequency response of graph filters ($U := V^{-1}$)

\[ y = Hx \implies y = V \text{diag}(\psi h)Ux \]

\[ \implies y \text{ is a bilinear function of the unknowns } h \text{ and } x \]
Leverage frequency response of graph filters ($U := V^{-1}$)

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

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

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

$\Rightarrow$ As.: $x$ is $S$-sparse i.e., $\|x\|_0 := |\text{supp}(x)| \leq S$
Leverage frequency response of graph filters ($U := V^{-1}$)

$y = Hx \Rightarrow y = V \text{diag}(\Psi h) Ux$

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

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

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

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

find $\{h, x\}$, s. to $y = V \text{diag}(\Psi h) Ux$, $\|x\|_0 \leq S$
Key observation: Use the Khatri-Rao product to write $y$ as

$$y = V(\Psi^T \odot U^T)^T \text{vec}(xh^T)$$

Reveals $y$ is a linear combination of the entries of $Z := xh^T$
“Lifting” the bilinear inverse problem

➤ **Key observation:** Use the Khatri-Rao product $\odot$ to write $y$ as

$$y = V(\Psi^T \odot U^T)^T \text{vec}(xh^T)$$

➤ Reveals $y$ is a **linear** combination of the entries of $Z := xh^T$

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

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

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

$\Rightarrow$ Matrix “lifting” for blind deconvolution [Ahmed et al.’14]

➤ Rank minimization s. to row-cardinality constraint is NP-hard
“Lifting” the bilinear inverse problem

- **Key observation:** Use the Khatri-Rao product \( \odot \) to write \( y \) as

\[
y = V(\Psi^T \odot U^T)^T \text{vec}(xh^T)
\]

- Reveals \( y \) is a linear combination of the entries of \( Z := xh^T \)

- \( Z \) is of rank-1 and row-sparse \( \Rightarrow \) Linear matrix inverse problem

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

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

\( \Rightarrow \) 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:** $\ell_1$-norm minimization promotes sparsity [Tibshirani’94]
  - Nuclear norm $\|Z\|_* := \sum_i \sigma_i(Z)$ a convex proxy of rank [Fazel’01]
  - $\ell_{2,1}$ norm $\|Z\|_{2,1} := \sum_i \|z_i^T\|_2$ surrogate of $\|Z\|_{2,0}$ [Yuan-Lin’06]
Algorithmic approach via convex relaxation

- **Key property**: $\ell_1$-norm minimization promotes sparsity [Tibshirani’94]
  - Nuclear norm $\|Z\|_* := \sum_i \sigma_i(Z)$ a convex proxy of rank [Fazel’01]
  - $\ell_{2,1}$ norm $\|Z\|_{2,1} := \sum_i \|z_i^T\|_2$ surrogate of $\|Z\|_{2,0}$ [Yuan-Lin’06]

- Convex relaxation

\[
\min_{Z} \|Z\|_* + \alpha \|Z\|_{2,1}, \quad \text{s. to } y = V (\Psi^T \otimes U^T)^T \text{vec}(Z)
\]

$\Rightarrow$ Scalable algorithm using method of multipliers
Algorithmic approach via convex relaxation

- **Key property:** $\ell_1$-norm minimization promotes sparsity [Tibshirani’94]
  - Nuclear norm $\|Z\|_* := \sum_i \sigma_i(Z)$ a convex proxy of rank [Fazel’01]
  - $\ell_{2,1}$ norm $\|Z\|_{2,1} := \sum_i \|z_i^T\|_2$ surrogate of $\|Z\|_{2,0}$ [Yuan-Lin’06]

- **Convex relaxation**

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

  $\Rightarrow$ Scalable algorithm using method of multipliers

- Refine estimates $\{h, x\}$ via iteratively-reweighted optimization
  $\Rightarrow$ Weights $\alpha_i(k) = (\|z_i(k)^T\|_2 + \delta)^{-1}$ per row $i$, per iteration $k$

- **Exact recovery** conditions $\Rightarrow$ Success of the convex relaxation
  $\Rightarrow$ Random model on the graph structure $\Rightarrow$ Recovery conditions
  $\Rightarrow$ Probabilistic guarantees that depend on the graph spectrum
  $\Rightarrow$ Blind deconvolution (in time) is a favorable graph setting
Numerical tests: Recovery rates

- **Recovery rates** over an \((L, S)\) grid and 20 trials
  - Successful recovery when \(\|x^* (h^*)^T - xh^T\|_F < 10^{-3}\)

- **ER** (left), **ER reweighted** \(\ell_{2,1}\) (center), **brain reweighted** \(\ell_{2,1}\) (right)

- **Exact recovery over non-trivial** \((L, S)\) region
  - Reweighted optimization markedly improves performance
  - Encouraging results even for real-world graphs
Numerical tests: Brain graph

- Human brain graph with $N = 66$ regions, $L = 6$ and $S = 6$

- Proposed method also outperforms alternating-minimization solver
  ⇒ Unknown $\text{supp}(x) \approx$ Need twice as many observations
Multiple output signals

- Suppose we have access to $P$ output signals $\{y_p\}_{p=1}^P$.
Formulation

- As.: \( \{x_p\}_{p=1}^P \) are \( S \)-sparse with common support
Formulation

- **As.**: $\{x_p\}_{p=1}^P$ are $S$-sparse with common support

- Concatenate outputs $\bar{y} := [y_1^T, \ldots, y_P^T]^T$ and inputs $\bar{x} := [x_1^T, \ldots, x_P^T]^T$

- Unknown rank-one matrices $Z_p := x_p h^T$. Stack them
  - Vertically in rank one $\bar{Z}_v := [Z_1^T, \ldots, Z_P^T]^T = \bar{x} h^T \in \mathbb{R}^{NP \times L}$
  - Horizontally in row sparse $\bar{Z}_h := [Z_1, \ldots, Z_P] \in \mathbb{R}^{N \times PL}$
As.: \( \{x_p\}_{p=1}^P \) are \( S \)-sparse with common support

Concatenate outputs \( \bar{y} := [y_1^T, \ldots, y_P^T]^T \) and inputs \( \bar{x} := [x_1^T, \ldots, x_P^T]^T \)

Unknown rank-one matrices \( Z_p := x_p h^T \). Stack them

⇒ Vertically in rank one \( \bar{Z}_v := [Z_1^T, \ldots, Z_P^T]^T = \bar{x} h^T \in \mathbb{R}^{NP \times L} \)

⇒ Horizontally in row sparse \( \bar{Z}_h := [Z_1, \ldots, Z_P] \in \mathbb{R}^{N \times PL} \)

Convex formulation

\[
\min_{\{Z_p\}_{p=1}^P} \| \bar{Z}_v \|_* + \tau \| \bar{Z}_h \|_{2,1}, \quad \text{s. to } \bar{y} = \left( I_P \otimes \left( V (\psi^T \odot U^T)^T \right) \right) \text{vec}(\bar{Z}_h)
\]
Formulation

- **As.:** \( \{x_p\}_{p=1}^P \) are \( S \)-sparse with common support

- Concatenate outputs \( \bar{y} := [y_1^T, \ldots, y_P^T]^T \) and inputs \( \bar{x} := [x_1^T, \ldots, x_P^T]^T \)

- Unknown rank-one matrices \( Z_p := x_p h^T \). Stack them
  - Vertically in rank one \( \bar{Z}_v := [Z_1^T, \ldots, Z_P^T]^T = \bar{x} h^T \in \mathbb{R}^{NP \times L} \)
  - Horizontally in row sparse \( \bar{Z}_h := [Z_1, \ldots, Z_P] \in \mathbb{R}^{N \times PL} \)

- **Convex formulation**

\[
\min_{\{Z_p\}_{p=1}} \| \bar{Z}_v \|_* + \tau \| \bar{Z}_h \|_{2,1}, \quad \text{s. to} \quad \bar{y} = \left( I_P \otimes \left( V (\Psi^T \odot U^T)^T \right) \right) \text{vec}(\bar{Z}_h)
\]

\[\Rightarrow \text{Relax (As.):} \quad \| \bar{Z}_h \|_{2,1} \leftrightarrow \| \bar{Z}_v \|_{2,1} = \sum_{p=1}^P \| Z_p \|_{2,1} \]
Numerical tests: Multiple signals, recovery rates

- Recovery rates over an \((L, S)\) grid and 20 trials
  - Successful recovery when \(\|\hat{\mathbf{x}}h^T - \bar{\mathbf{x}}h^T\|_F < 10^{-3}\)

- ER (left), ER reweighted \(\ell_{2,1}\) (center), brain reweighted \(\ell_{2,1}\) (right)

- Leveraging multiple output signals aids the blind identification task
Summary and extensions

- Extended blind deconvolution of space/time signals to graphs
  - Key: model diffusion process as output of graph filter
- Rank and sparsity minimization subject to model constraints
  - “Lifting” and convex relaxation yield efficient algorithms
- Exact recovery conditions ⇒ Success of the convex relaxation
  ⇒ 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 $S$ ⇒ Network topology inference
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

Paper submission due: June 5, 2016

Organizers:
- Michael Rabbat (McGill Univ.)
- Antonio Marques (King Juan Carlos Univ.)
- Gonzalo Mateos (Univ. of Rochester)
Relevance of the graph-shift operator

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

Set $S = A_{de}$

What is $Sx$?

$$
\begin{pmatrix}
0 \\
0 \\
x_2 \\
x_3 \\
0 \\
0
\end{pmatrix} =
\begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 0 & 0 & 0 \\
x_2 & 0 & 1 & 0 & 0 & 0 \\
x_3 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
x_1 \\
x_6 \\
x_5 \\
x_4 \\
x_1 \\
x_2
\end{pmatrix}
$$
Relevance of the graph-shift operator

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

Set $S = A_{dc}$

What is $Sx$?

$S$ will be building block for GSP algorithms

⇒ Same is true in the time domain (filters and delay)