Blind Identification of Graph Filters with Multiple Sparse Inputs

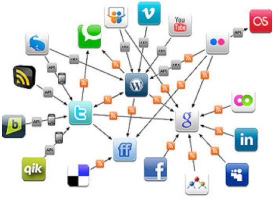
Santiago Segarra, Antonio G. Marques, Gonzalo Mateos & Alejandro Ribeiro

Dept. of Electrical and Systems Engineering
University of Pennsylvania
ssegarra@seas.upenn.edu
http://www.seas.upenn.edu/~ssegarra/

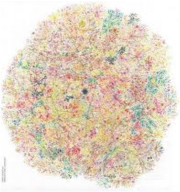
ICASSP, March 24, 2016

Network Science analytics

Online social media



Internet

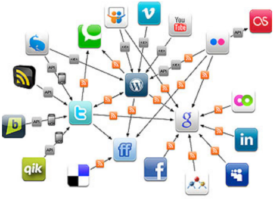


Clean energy and grid analytics

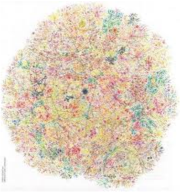
- Desiderata: Process, analyze and learn from network data [Kolaczyk'09]

Network Science analytics

Online social media



Internet

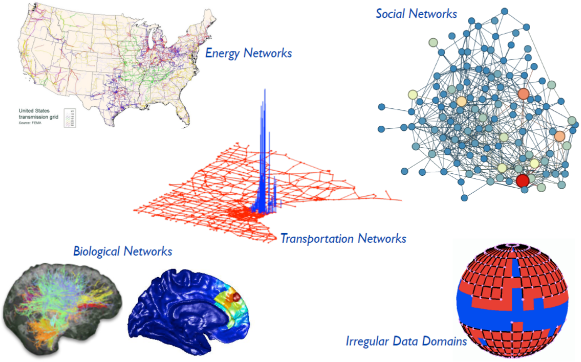


Clean energy and grid analytics

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

  $\Rightarrow$ The object of study is a graph signal
- Ex: Opinion profile, buffer congestion levels, neural activity, epidemic

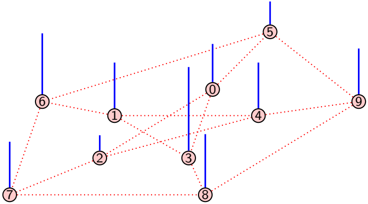
Motivating examples – Graph signals



- Graph SP: broaden classical SP to graph signals [Shuman etal'13]
  - $\Rightarrow$ Our view: GSP well suited to study network processes
- **As.:** Signal properties related to topology of G (e.g., smoothness)
  - $\Rightarrow$ Algorithms that fruitfully leverage this relational structure

Graph signals

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

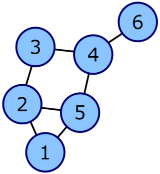


$$\mathbf{x} = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_9 \end{bmatrix} = \begin{bmatrix} 0.6 \\ 0.7 \\ 0.3 \\ \vdots \\ 0.7 \end{bmatrix}$$

Graph-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



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

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

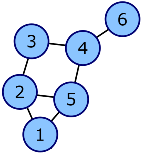
Locality of the graph-shift operator

- $\mathbf{S}$ is a linear operator that can be computed locally at the nodes in $\mathcal{V}$
- Consider the graph signal $\mathbf{y} = \mathbf{S}\mathbf{x}$ and node i's neighborhood $\mathcal{N}_i$

  $\Rightarrow$ Node i can compute y_i if it has access to x_j at $j \in \mathcal{N}_i$

$$y_i = \sum_{j \in \mathcal{N}_i} S_{ij} x_j, \quad i \in \mathcal{V}$$

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



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

- If $\mathbf{y} = \mathbf{S}^2\mathbf{x} \Rightarrow y_i$ found using values x_j within 2 hops

Graph Fourier transform (GFT)

- **As.:** $\mathbf{S}$ related to generation (description) of the signals of interest
  $\Rightarrow$ Spectrum of $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$ will be especially useful to analyze $\mathbf{x}$
- 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

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

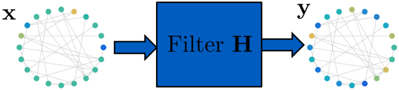
  $\Rightarrow$ Eigenvectors $\mathbf{V} = [\mathbf{v}_1, ..., \mathbf{v}_N]$ are the frequency basis (atoms)

- Ex: For the directed cycle (temporal signal) $\Rightarrow$ GFT $\equiv$ DFT

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



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

Graph filter [Sandryhaila-Moura'13]

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

- Key properties: shift-invariance and distributed implementation
  - $\Rightarrow$ $\mathbf{H}(\mathbf{S}\mathbf{x}) = \mathbf{S}(\mathbf{H}\mathbf{x})$, no other can be linear and shift-invariant
  - $\Rightarrow$ Each application of $\mathbf{S}$ only local info $\Rightarrow$ only L-hop info for $\mathbf{y} = \mathbf{H}\mathbf{x}$

Frequency response of a graph filter

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

Frequency response of a graph filter

- Using $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, filter is $\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{S}^l = \mathbf{V}\left(\sum_{l=0}^{L} h_l \mathbf{\Lambda}^l\right)\mathbf{V}^{-1}$
- Since $\mathbf{\Lambda}^l$ are diagonal, use GFT-iGFT to write $\mathbf{y} = \mathbf{H}\mathbf{x}$ as

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

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

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

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

Frequency response of a graph filter

- Using $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^{-1}$, filter is $\mathbf{H} = \sum_{l=0}^{L} h_l \mathbf{S}^l = \mathbf{V}\left(\sum_{l=0}^{L} h_l \mathbf{\Lambda}^l\right)\mathbf{V}^{-1}$
- Since $\mathbf{\Lambda}^l$ are diagonal, use GFT-iGFT to write $\mathbf{y} = \mathbf{H}\mathbf{x}$ as

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

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

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

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

- GFT for signals ($\tilde{\mathbf{x}} = \mathbf{V}^{-1}\mathbf{x}$) and filters ($\tilde{\mathbf{h}} = \mathbf{\Psi}\mathbf{h}$) is different

Diffusion processes as graph filter outputs

- Q: Upon observing a graph signal $\mathbf{y}$, how was this signal generated?

Diffusion processes as graph filter outputs

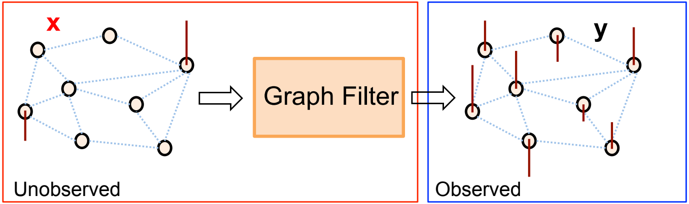
- Q: Upon observing a graph signal $\mathbf{y}$, how was this signal generated?
- Postulate the following generative model
  - $\Rightarrow$ An originally sparse signal $\mathbf{x} = \mathbf{x}^{(0)}$
  - $\Rightarrow$ Diffused via linear graph dynamics $\mathbf{S}$ $\Rightarrow \mathbf{x}^{(l)} = \mathbf{S}\mathbf{x}^{(l-1)}$
  - $\Rightarrow$ 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
  - $\Rightarrow$ View few elements in $\text{supp}(\mathbf{x}) =: \{i : x_i \neq 0\}$ as seeds

Motivation and problem statement

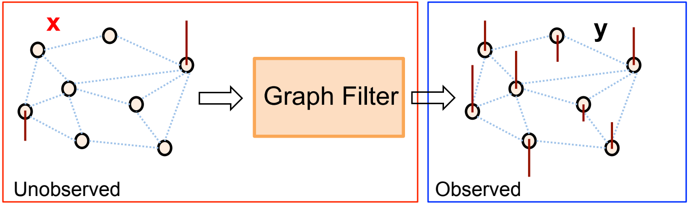
- Ex: Global opinion/belief profile formed by spreading a rumor
  - $\Rightarrow$ What was the rumor? Who started it?
  - $\Rightarrow$ How do people weigh in peers' opinions to form their own?



- Problem: Blind identification of graph filters with sparse inputs

Motivation and problem statement

- Ex: Global opinion/belief profile formed by spreading a rumor
  - $\Rightarrow$ What was the rumor? Who started it?
  - $\Rightarrow$ 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 $\mathbf{y} = \mathbf{H}\mathbf{x}$?
  - $\Rightarrow$ Extends classical blind deconvolution to graphs

Blind graph filter identification

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

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

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

Blind graph filter identification

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

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

  $\Rightarrow$ $\mathbf{y}$ 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

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

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

$\Rightarrow$ $\mathbf{y}$ 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

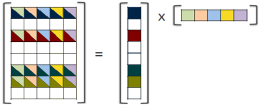
$$\text{find} \;\; \{\mathbf{h}, \mathbf{x}\}, \quad \text{s. to} \;\; \mathbf{y} = \mathbf{V}\text{diag}\big(\boldsymbol{\Psi}\mathbf{h}\big)\mathbf{U}\mathbf{x}, \;\; \|\mathbf{x}\|_0 \leq S$$

"Lifting" the bilinear inverse problem

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

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

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

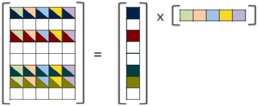


"Lifting" the bilinear inverse problem

- 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]

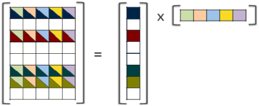
- Rank minimization s. to row-cardinality constraint is NP-hard

"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]

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

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

$\Rightarrow$ Scalable algorithm using method of multipliers

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

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

$\Rightarrow$ 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

- Exact recovery conditions $\Rightarrow$ Success of the convex relaxation

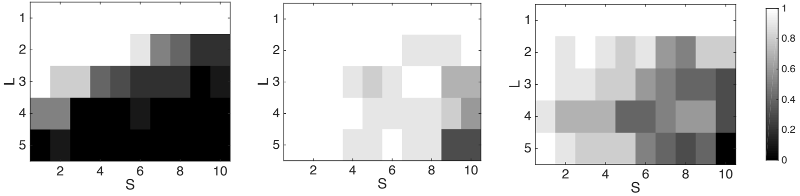
$\Rightarrow$ Random model on the graph structure $\Rightarrow$ Recovery conditions

$\Rightarrow$ 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 $\|\mathbf{x}^*(\mathbf{h}^*)^T - \mathbf{x}\mathbf{h}^T\|_F < 10^{-3}$
- ER (left), ER reweighted $\ell_{2,1}$ (center), brain reweighted $\ell_{2,1}$ (right)



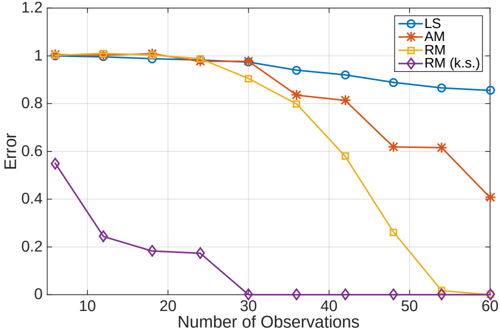
- Exact recovery over non-trivial (L, S) region

  $\Rightarrow$ Reweighted optimization markedly improves performance

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

Numerical tests: Brain graph

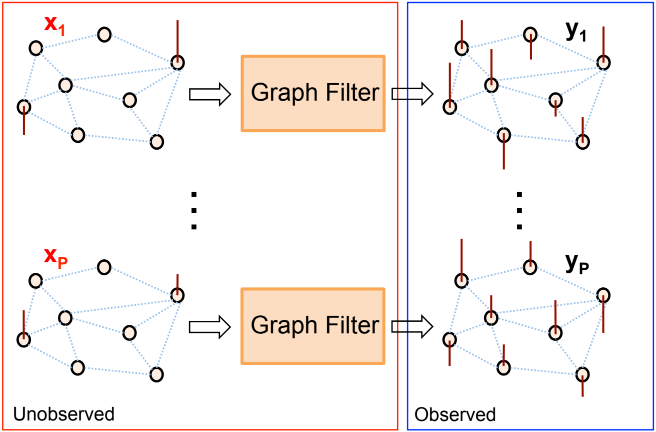
- Human brain graph with $N = 66$ regions, $L = 6$ and $S = 6$



- Proposed method also outperforms alternating-minimization solver
  - $\Rightarrow$ Unknown supp($\mathbf{x}$) $\approx$ Need twice as many observations

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

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 $\bar{\mathbf{Z}}_h := [\mathbf{Z}_1, ..., \mathbf{Z}_P] \in \mathbb{R}^{N \times PL}$

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

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

Formulation

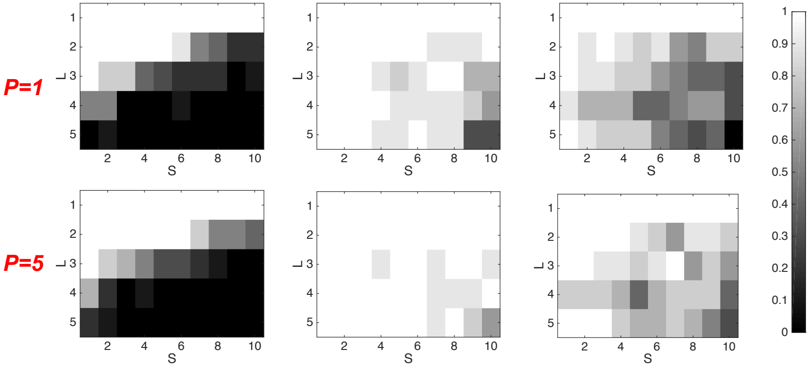
- **As.:** $\{\mathbf{x}_p\}_{p=1}^P$ are S-sparse with common support
- Concatenate outputs $\bar{\mathbf{y}} := [\mathbf{y}_1^T, \ldots, \mathbf{y}_P^T]^T$ and inputs $\bar{\mathbf{x}} := [\mathbf{x}_1^T, \ldots, \mathbf{x}_P^T]^T$
- Unknown rank-one matrices $\mathbf{Z}_p := \mathbf{x}_p \mathbf{h}^T$. 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

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

$\Rightarrow$ 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

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 $\mathbf{S}$ ⇒ Network topology inference

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.)

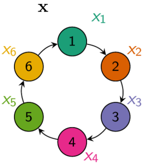
Gonzalo Mateos (Univ. of Rochester)

Relevance of the graph-shift operator

- Q: Why is $\mathbf{S}$ called shift?

Relevance of the graph-shift operator

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



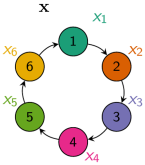
Set $\mathbf{S} = \mathbf{A_{dc}}$

What is $\mathbf{Sx}$?

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

Relevance of the graph-shift operator

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



⇨ Set $\mathbf{S} = \mathbf{A}_{\mathbf{dc}}$

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