Online Topology Inference from Streaming Stationary Graph Signals

Rasoul Shafipour

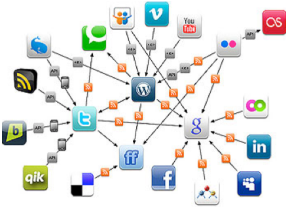
Dept. of Electrical and Computer Engineering
University of Rochester
rshafipo@ece.rochester.edu
http://www.ece.rochester.edu/~rshafipo/

Co-authors: Abolfazl Hashemi, Gonzalo Mateos, and Haris Vikalo

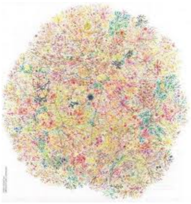
IEEE Data Science Workshop, June 4, 2019

Network Science analytics

Online social media



Internet

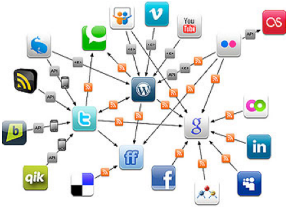


Clean energy and grid analytics

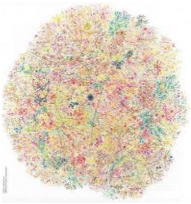
- Network as graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$: encode pairwise relationships
- Desiderata: Process, analyze and learn from network data [Kolaczyk'09]

Network Science analytics

Online social media



Internet



Clean energy and grid analytics

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

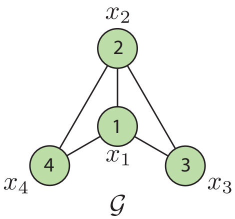
Graph signal processing (GSP)

- Undirected $\mathcal{G}$ with adjacency matrix $\mathbf{A}$

  $\Rightarrow A_{ij}$ = Proximity between i and j

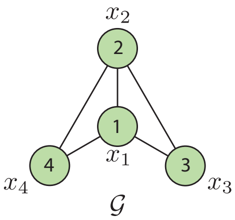
- Define a signal $\mathbf{x}$ on top of the graph

  $\Rightarrow x_i$ = Signal value at node i



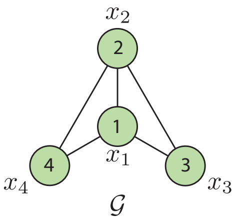
Graph signal processing (GSP)

- Undirected $\mathcal{G}$ with adjacency matrix $\mathbf{A}$
  - $\Rightarrow A_{ij}$ = Proximity between i and j
- Define a signal $\mathbf{x}$ on top of the graph
  - $\Rightarrow x_i$ = Signal value at node i



- Associated with $\mathcal{G}$ is the graph-shift operator (GSO) $\mathbf{S} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^T \in \mathcal{M}^N$
  - $\Rightarrow S_{ij} = 0$ for $i \neq j$ and $(i, j) \notin \mathcal{E}$ (local structure in G)
  - $\Rightarrow$ Ex: $\mathbf{A}$, degree $\mathbf{D}$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices

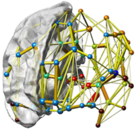
Graph signal processing (GSP)



- Undirected $\mathcal{G}$ with adjacency matrix $\mathbf{A}$
  - $\Rightarrow A_{ij}$ = Proximity between i and j
- Define a signal $\mathbf{x}$ on top of the graph
  - $\Rightarrow x_i$ = Signal value at node i
- Associated with $\mathcal{G}$ is the graph-shift operator (GSO) $\mathbf{S} = \mathbf{V \Lambda V}^T \in \mathcal{M}^N$
  - $\Rightarrow S_{ij} = 0$ for $i \neq j$ and $(i, j) \notin \mathcal{E}$ (local structure in G)
  - $\Rightarrow$ Ex: $\mathbf{A}$, degree $\mathbf{D}$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices
- Graph Signal Processing $\rightarrow$ Exploit structure encoded in $\mathbf{S}$ to process $\mathbf{x}$
  - $\Rightarrow$ GSP well suited to study (network) diffusion processes
- Use GSP to learn the underlying $\mathcal{G}$ or a meaningful network model

Topology inference: Motivation and context

- Network topology inference from nodal observations [Kolaczyk'09]
  - Partial correlations and conditional dependence [Dempster'74]
  - Sparsity [Friedman et al'07] and consistency [Meinshausen-Buhlmann'06]
  - [Banerjee et al'08], [Lake et al'10], [Slawski et al'15], [Karanikolas et al'16]



- Can be useful in neuroscience [Sporns'10]

  $\Rightarrow$ Functional net inferred from activity

- Noteworthy GSP-based approaches
  - Gaussian graphical models [Egilmez et al'16]
  - Smooth signals [Dong et al'15], [Kalofolias'16]
  - Stationary signals [Pasdeloup et al'15], [Segarra et al'16]
  - Non-stationary signals [Shafipour et al'17]
  - Directed graphs [Mei-Moura'15], [Shen et al'16]
  - Low-rank excitation [Wai et al'18]
  - Learning from sequential data [Vlaski et al'18]

- **Here:** online topology inference from streaming stationary graph signals

Generating structure of a diffusion process

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

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

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

- One can state that the graph shift $\mathbf{S}$ explains the structure of signal $\mathbf{y}$

Generating structure of a diffusion process

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

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

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

- One can state that the graph shift $\mathbf{S}$ explains the structure of signal $\mathbf{y}$

- Cayley-Hamilton asserts that we can write diffusion as

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

$\Rightarrow$ Degree $L \leq N$ depends on the dependency range of the filter

$\Rightarrow$ Shift invariant operator $\mathbf{H}$ is graph filter [Sandryhaila-Moura'13]

- Online topology inference: From $\mathcal{Y}=\{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(P)}, \cdots\}$, Find $\mathbf{S}$ efficiently

Topology inference under stationarity

Stationary graph signal [Marques et al'16]

Def: A graph signal $\mathbf{y}$ is stationary with respect to the shift $\mathbf{S}$ if and only if $\mathbf{y} = \mathbf{H}\mathbf{x}$, where $\mathbf{H} = \sum_{l=0}^{L-1} h_l \mathbf{S}^l$ and $\mathbf{x}$ is white.

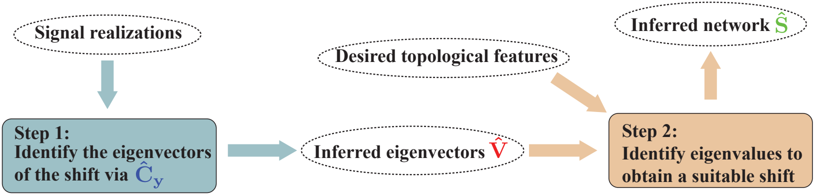
- The covariance matrix of the stationary signal $\mathbf{y}$ is

$$\mathbf{C}_y = \mathbb{E}\left[\mathbf{H}\mathbf{x}(\mathbf{H}\mathbf{x})^T\right] = \mathbf{H}\mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right]\mathbf{H}^T = \mathbf{H}\mathbf{H}^T$$

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

$$\mathbf{C}_y = \mathbf{V}\left|\sum_{l=0}^{L-1} h_l \mathbf{\Lambda}^l\right|^2 \mathbf{V}^T = \mathbf{V}\left(\mathcal{H}(\mathbf{\Lambda})\right)^2 \mathbf{V}^T$$

$\Rightarrow$ Estimate $\mathbf{V}$ from $\mathcal{Y}$ via Principal Component Analysis



- Step 2: Obtaining the eigenvalues of $\mathbf{S}$
- We can use extra knowledge/assumptions to choose one graph

  $\Rightarrow$ Of all graphs, select one that is optimal in the number of edges

$$\hat{\mathbf{S}} := \underset{\mathbf{S},\mathbf{\Lambda}}{\operatorname{argmin}} \ \|\mathbf{S}\|_1 \quad \text{subject to:} \quad \|\mathbf{S} - \hat{\mathbf{V}}\mathbf{\Lambda}\hat{\mathbf{V}}^T\|_F \leq \epsilon, \ \ \mathbf{S} \in \mathcal{S}$$

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

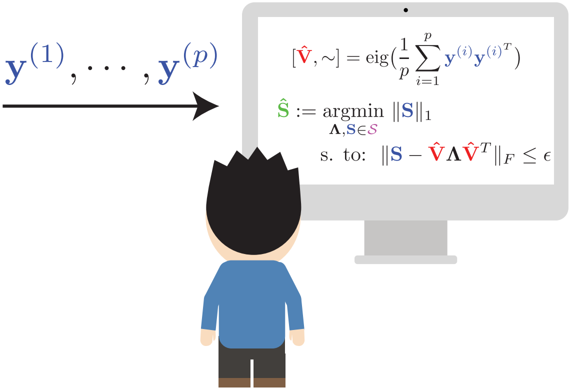
$$\mathcal{S} := \{\mathbf{S} \mid S_{ij} \geq 0, \ \ \mathbf{S} \in \mathcal{M}^N, \ S_{ii} = 0, \ \ \sum_j S_{1j} = 1\}$$

Online inference under stationarity

- Consider streaming stationary signals $\mathcal{Y}:=\{\mathbf{y}^{(1)},\cdots,\mathbf{y}^{(p)},\mathbf{y}^{(p+1)},\cdots\}$
- Assume that time differences of the signals arrival is relatively low

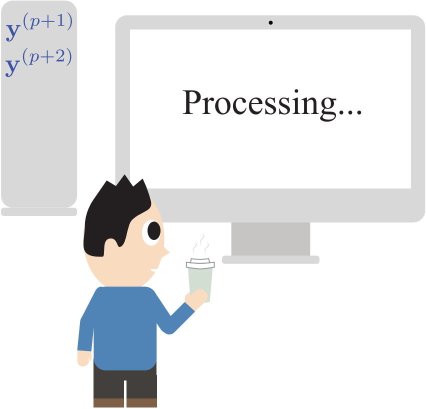
Online inference under stationarity

- Consider streaming stationary signals $\mathcal{Y}:=\{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(p)}, \mathbf{y}^{(p+1)}, \cdots\}$
- Assume that time differences of the signals arrival is relatively low



Online inference under stationarity

- Consider streaming stationary signals $\mathcal{Y}:=\{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(p)}, \mathbf{y}^{(p+1)}, \cdots\}$
- Assume that time differences of the signals arrival is relatively low



Online inference under stationarity

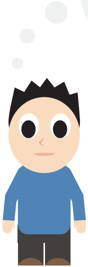
- Consider streaming stationary signals $\mathcal{Y} := \{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(p)}, \mathbf{y}^{(p+1)}, \cdots\}$
- Assume that time differences of the signals arrival is relatively low

Online inference under stationarity

- Consider streaming stationary signals $\mathcal{Y} := \{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(p)}, \mathbf{y}^{(p+1)}, \cdots\}$
- Assume that time differences of the signals arrival is relatively low

Online inference under stationarity

- Consider streaming stationary signals $\mathcal{Y} := \{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(p)}, \mathbf{y}^{(p+1)}, \cdots\}$
- Assume that time differences of the signals arrival is relatively low



- Develop an iterative algorithm for the topology inference
- Upon sensing new diffused output signals

$\Rightarrow$ - Update $\hat{\mathbf{V}}$ efficiently

- Take one or a few steps of the iterative algorithm

Topology inference via ADMM

- To apply ADMM, rewrite the problem as

$$\min_{\mathbf{S},\mathbf{\Lambda},\mathbf{D}} \quad \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\mathbf{\Lambda}\hat{\mathbf{V}}^\top\|_F^2$$

$$\text{s.to: } \mathbf{S} - \mathbf{D} = \mathbf{0}, \quad \mathbf{D} \in \mathcal{S} = \{\mathbf{S} \mid S_{ij} \geq 0,\ \mathbf{S} \in \mathcal{M}^N,\ S_{ii} = 0,\ \textstyle\sum_j S_{1j} = 1\}$$

$\Rightarrow$ Convex, thus ADMM would converge to a global minimizer

- Form the augmented Lagrangian

$$\mathcal{L}_{\rho_1}(\mathbf{S},\mathbf{D},\mathbf{\Lambda},\mathbf{U}_1) = \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\mathbf{\Lambda}\hat{\mathbf{V}}^\top\|_F^2 + \frac{\rho_1}{2}\|\mathbf{S} - \mathbf{D} + \mathbf{U}_1\|_F^2$$

- At k^{th} iteration, let $\mathbf{B}^{(k)} = \hat{\mathbf{V}}\mathbf{\Lambda}^{(k)}\hat{\mathbf{V}}^\top \Rightarrow$ ADMM consists of 4 iterative steps

Topology inference via ADMM

- To apply ADMM, rewrite the problem as

$$\min_{\mathbf{S},\boldsymbol{\Lambda},\mathbf{D}} \quad \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\boldsymbol{\Lambda}\hat{\mathbf{V}}^\top\|_F^2$$

$$\text{s.to: } \mathbf{S} - \mathbf{D} = \mathbf{0}, \quad \mathbf{D} \in \mathcal{S} = \{\mathbf{S} \mid S_{ij} \geq 0, \ \mathbf{S} \in \mathcal{M}^N, \ S_{ii} = 0, \ \textstyle\sum_j S_{1j} = 1\}$$

$\Rightarrow$ Convex, thus ADMM would converge to a global minimizer

- Form the augmented Lagrangian

$$\mathcal{L}_{\rho_1}(\mathbf{S}, \mathbf{D}, \boldsymbol{\Lambda}, \mathbf{U}_1) = \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\boldsymbol{\Lambda}\hat{\mathbf{V}}^\top\|_F^2 + \frac{\rho_1}{2}\|\mathbf{S} - \mathbf{D} + \mathbf{U}_1\|_F^2$$

- At k^{th} iteration, let $\mathbf{B}^{(k)} = \hat{\mathbf{V}}\boldsymbol{\Lambda}^{(k)}\hat{\mathbf{V}}^\top \Rightarrow$ ADMM consists of 4 iterative steps

- **Step 1.** $\mathbf{S}^{(k+1)} = \underset{\mathbf{S}}{\operatorname{argmin}} \mathcal{L}_{\rho_1}(\mathbf{S}, \mathbf{D}^{(k)}, \boldsymbol{\Lambda}^{(k)}, {\mathbf{U}_1}^{(k)}) = \mathcal{T}_{\frac{\lambda}{2+\rho_1}}\left(\frac{\mathbf{B}^{(k)} + \frac{\rho_1}{2}(\mathbf{D}^{(k)} - {\mathbf{U}_1}^{(k)})}{1 + \frac{\rho_1}{2}}\right)$,
where $\mathcal{T}_\eta(x) = (|x| - \eta)_+ \operatorname{sign}(x)$ is the element-wise soft-thresholding operator

Topology inference via ADMM

- To apply ADMM, rewrite the problem as

$$\min_{\mathbf{S},\mathbf{\Lambda},\mathbf{D}} \quad \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\mathbf{\Lambda}\hat{\mathbf{V}}^\top\|_F^2$$

$$\text{s.to: } \mathbf{S} - \mathbf{D} = \mathbf{0}, \quad \mathbf{D} \in \mathcal{S} = \{\mathbf{S} \,|\, S_{ij} \geq 0, \;\; \mathbf{S} \in \mathcal{M}^N, \; S_{ii} = 0, \;\; \textstyle\sum_j S_{1j} = 1\}$$

$\Rightarrow$ Convex, thus ADMM would converge to a global minimizer

- Form the augmented Lagrangian

$$\mathcal{L}_{\rho_1}(\mathbf{S}, \mathbf{D}, \mathbf{\Lambda}, \mathbf{U}_1) = \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\mathbf{\Lambda}\hat{\mathbf{V}}^\top\|_F^2 + \frac{\rho_1}{2}\|\mathbf{S} - \mathbf{D} + \mathbf{U}_1\|_F^2$$

- At k^{th} iteration, let $\mathbf{B}^{(k)} = \hat{\mathbf{V}}\mathbf{\Lambda}^{(k)}\hat{\mathbf{V}}^\top \Rightarrow$ ADMM consists of 4 iterative steps

- **Step 2.** $\mathbf{D}^{(k+1)} = \underset{\mathbf{D} \in \mathcal{S}}{\operatorname{argmin}} \quad \mathcal{L}_{\rho_1}(\mathbf{S}^{(k+1)}, \mathbf{D}, \mathbf{\Lambda}^{(k)}, {\mathbf{U}_1}^{(k)}) = \mathcal{P}_{\mathcal{S}}(\mathbf{S}^{(k+1)} + {\mathbf{U}_1}^{(k)})$,
where $\mathcal{P}_{\mathcal{S}}(.)$ is the projection operator onto $\mathcal{S}$

Topology inference via ADMM

- To apply ADMM, rewrite the problem as

$$\min_{\mathbf{S},\boldsymbol{\Lambda},\mathbf{D}} \quad \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\boldsymbol{\Lambda}\hat{\mathbf{V}}^\top\|_F^2$$

$$\text{s.to: } \mathbf{S} - \mathbf{D} = \mathbf{0}, \quad \mathbf{D} \in \mathcal{S} = \{\mathbf{S} \mid S_{ij} \geq 0,\ \ \mathbf{S} \in \mathcal{M}^N,\ S_{ii} = 0,\ \ \sum_j S_{1j} = 1\}$$

$\Rightarrow$ Convex, thus ADMM would converge to a global minimizer

- Form the augmented Lagrangian

$$\mathcal{L}_{\rho_1}(\mathbf{S}, \mathbf{D}, \boldsymbol{\Lambda}, \mathbf{U}_1) = \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\boldsymbol{\Lambda}\hat{\mathbf{V}}^\top\|_F^2 + \frac{\rho_1}{2}\|\mathbf{S} - \mathbf{D} + \mathbf{U}_1\|_F^2$$

- At k^{th} iteration, let $\mathbf{B}^{(k)} = \hat{\mathbf{V}}\boldsymbol{\Lambda}^{(k)}\hat{\mathbf{V}}^\top \Rightarrow$ ADMM consists of 4 iterative steps

- **Step 3.** $\boldsymbol{\Lambda}^{(k+1)} = \underset{\boldsymbol{\Lambda}}{\operatorname{argmin}} \quad \mathcal{L}_{\rho_1}(\mathbf{S}^{(k+1)}, \mathbf{D}^{(k+1)}, \boldsymbol{\Lambda}, \mathbf{U}_1{}^{(k)})$

Topology inference via ADMM

- To apply ADMM, rewrite the problem as

$$\min_{\mathbf{S},\boldsymbol{\Lambda},\mathbf{D}} \quad \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\boldsymbol{\Lambda}\hat{\mathbf{V}}^\top\|_F^2$$

$$\text{s.to: } \mathbf{S} - \mathbf{D} = \mathbf{0}, \quad \mathbf{D} \in \mathcal{S} = \{\mathbf{S} \mid S_{ij} \geq 0,\ \mathbf{S} \in \mathcal{M}^N,\ S_{ii} = 0,\ \textstyle\sum_j S_{1j} = 1\}$$

$\Rightarrow$ Convex, thus ADMM would converge to a global minimizer

- Form the augmented Lagrangian

$$\mathcal{L}_{\rho_1}(\mathbf{S}, \mathbf{D}, \boldsymbol{\Lambda}, \mathbf{U}_1) = \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\boldsymbol{\Lambda}\hat{\mathbf{V}}^\top\|_F^2 + \frac{\rho_1}{2}\|\mathbf{S} - \mathbf{D} + \mathbf{U}_1\|_F^2$$

- At k^{th} iteration, let $\mathbf{B}^{(k)} = \hat{\mathbf{V}}\boldsymbol{\Lambda}^{(k)}\hat{\mathbf{V}}^\top \Rightarrow$ ADMM consists of 4 iterative steps

- **Step 3.** $\boldsymbol{\Lambda}^{(k+1)} = \operatorname{argmin}_{\boldsymbol{\Lambda}} \quad \|\boldsymbol{\Lambda} - \hat{\mathbf{V}}^\top\mathbf{S}^{(k+1)}\hat{\mathbf{V}}\|_F^2 = \text{Diag}(\hat{\mathbf{V}}^\top\mathbf{S}^{(k+1)}\hat{\mathbf{V}})$

Topology inference via ADMM

- To apply ADMM, rewrite the problem as

$$\min_{\mathbf{S},\boldsymbol{\Lambda},\mathbf{D}} \quad \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\boldsymbol{\Lambda}\hat{\mathbf{V}}^\top\|_F^2$$

$$\text{s.to: } \mathbf{S} - \mathbf{D} = \mathbf{0}, \quad \mathbf{D} \in \mathcal{S} = \{\mathbf{S} \mid S_{ij} \geq 0, \ \mathbf{S} \in \mathcal{M}^N, \ S_{ii} = 0, \ \textstyle\sum_j S_{1j} = 1\}$$

$\Rightarrow$ Convex, thus ADMM would converge to a global minimizer

- Form the augmented Lagrangian

$$\mathcal{L}_{\rho_1}(\mathbf{S}, \mathbf{D}, \boldsymbol{\Lambda}, \mathbf{U}_1) = \lambda\|\mathbf{S}\|_1 + \|\mathbf{S} - \hat{\mathbf{V}}\boldsymbol{\Lambda}\hat{\mathbf{V}}^\top\|_F^2 + \frac{\rho_1}{2}\|\mathbf{S} - \mathbf{D} + \mathbf{U}_1\|_F^2$$

- At k^{th} iteration, let $\mathbf{B}^{(k)} = \hat{\mathbf{V}}\boldsymbol{\Lambda}^{(k)}\hat{\mathbf{V}}^\top \Rightarrow$ ADMM consists of 4 iterative steps

- **Step 4.** Dual gradient ascent update ${\mathbf{U}_1}^{(k+1)} = {\mathbf{U}_1}^{(k)} + \mathbf{S}^{(k+1)} - \mathbf{D}^{(k+1)}$

Topology inference algorithm

Input: estimated covariance eigenvectors $\hat{\mathbf{V}}$, penalty parameter ρ_1, regularization parameter λ, number of iterations T_1

Initialize: $\mathbf{\Lambda}^{(0)} = \mathrm{diag}(\mathbf{1}_N)$, $\mathbf{D}^{(0)} = \mathbf{0}$, $\mathbf{U}_1^{(0)} = \mathbf{0}$.

for $k = 0, \ldots, T_1 - 1$ **do**

$\quad \mathbf{B}^{(k)} = \hat{\mathbf{V}}\mathbf{\Lambda}^{(k)}\hat{\mathbf{V}}^\top$

$\quad \mathbf{S}^{(k+1)} = \mathcal{T}_{\frac{\lambda}{2+\rho_1}}\left(\frac{\mathbf{B}^{(k)} + \frac{\rho_1}{2}(\mathbf{D}^{(k)} - {\mathbf{U}_1}^{(k)})}{1+\frac{\rho_1}{2}}\right)$

$\quad \mathbf{D}^{(k+1)} = \mathcal{P}_{\mathcal{S}}(\mathbf{S}^{(k+1)} + {\mathbf{U}_1}^{(k)})$

$\quad \mathbf{\Lambda}^{(k+1)} = \mathrm{Diag}(\hat{\mathbf{V}}^\top \mathbf{S}^{(k+1)} \hat{\mathbf{V}})$

$\quad {\mathbf{U}_1}^{(k+1)} = {\mathbf{U}_1}^{(k)} + \mathbf{S}^{(k+1)} - \mathbf{D}^{(k+1)}$

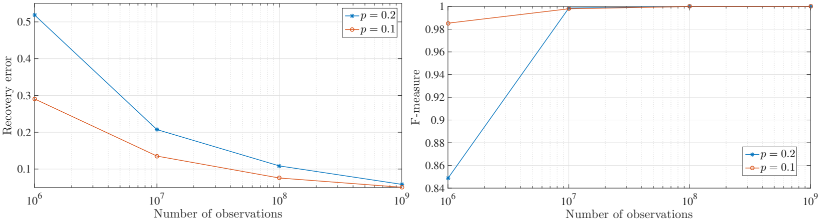
end for

return $\mathbf{S}^{(T_1)}$ and $\mathbf{\Lambda}^{(T_1)}$

- Develop an iterative algorithm for the topology inference
- Upon sensing new diffused output signals
  $\Rightarrow$ - Update $\hat{\mathrm{V}}$ efficiently
  - Take one or a few steps of the iterative algorithm

Inferring a large scale graph

- Consider an Erdős-Rényi graph with $N = 1000$ in an offline fashion
  - Edges are formed independently with probabilities $p = 0.1$ & 0.2
  - Signals diffused by $\mathbf{H} = \sum_{l=0}^{2} h_l \mathbf{A}^l$, $h_l \sim \mathcal{U}[0,1]$, $\mathbf{S} = \mathbf{A}$
  - Adopt sample covariance estimator for the Gaussian signals
  - Assess the recovery error $\xi_F := \|\hat{\mathbf{S}} - \mathbf{S}\|_F / \|\mathbf{S}\|_F$ and F-measure



- Increase in number of observations leads to a better performance
  - $\Rightarrow$ Performance enhances for sparser graphs (i.e., smaller p)

Online topology inference

- **Q:** How can we efficiently update the sample covariance eigenvectors $\hat{\mathbf{V}}$?
- Let $\hat{\mathbf{C}}_{\mathbf{y}}^{(P)}$ be sample covariance after receiving P streaming observations

  $\Rightarrow$ Updated sample covariance after receiving $\mathbf{y}^{(P+1)}$ takes the form

$$\hat{\mathbf{C}}_{\mathbf{y}}^{(P+1)} = \frac{1}{P+1}(P\hat{\mathbf{C}}_{\mathbf{y}}^{(P)} + \mathbf{y}^{(P+1)}\mathbf{y}^{(P+1)})$$

- Let $\mathbf{z} = \hat{\mathbf{V}}^{\top}\mathbf{y}^{(p+1)}$ and $\{d_j\}_{j=1}^{N}$ denote the eigenvalues of $\hat{\mathbf{C}}_{\mathbf{y}}^{(P)}$

  $\Rightarrow$ Eigenvalues of rank-one modification of $\hat{\mathbf{C}}_{\mathbf{y}}^{(P)}$ are the roots (γ) of

$$1 + \sum_{j=1}^{N} \frac{z_j^2}{Pd_j - \gamma} = 0 \quad \text{[Bunch et al'78]}$$

  $\Rightarrow$ Can be solved using the Newton method with $\mathcal{O}(N^2)$ complexity

- For the updated eigenvalue γ_j, the corresponding eigenvector $\mathbf{v}_j$ is given by

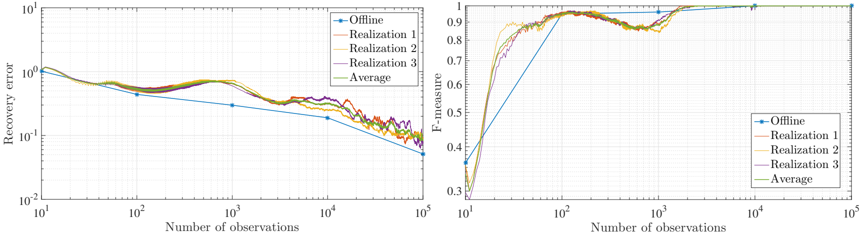
$$\mathbf{v}_j = \alpha_j \mathbf{y}^{(p+1)} \circ \mathbf{q}_j,$$

where $\mathbf{q}_j = [1/(Pd_1 - \gamma_j), \cdots, 1/(Pd_N - \gamma_j)]$ and α_j is a normalizing factor

Online inference of a brain network

- Consider a structural brain graph with $N = 66$ neural regions
  - Edge weights: Density of anatomical connections [Hagmann et al'08]
  - Signals diffused by $\mathbf{H} = \sum_{l=0}^{2} h_l \mathbf{A}^l$, $h_l \sim \mathcal{U}[0,1]$, $\mathbf{S} = \mathbf{A}$
  - Generate streaming signals $\{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(p)}, \mathbf{y}^{(p+1)}, \cdots\}$ via $\mathbf{y}^{(i)} = \mathbf{H}\mathbf{x}^{(i)}$
  - Upon sensing an observation $\mathbf{y}^{(p)}$

    $\Rightarrow$ Update $\hat{\mathbf{V}}$ efficiently and run the algorithm for $T_1 = 1$
  - Assess the recovery error $\xi_F := \|\hat{\mathbf{S}} - \mathbf{S}\|_F / \|\mathbf{S}\|_F$ and F-measure



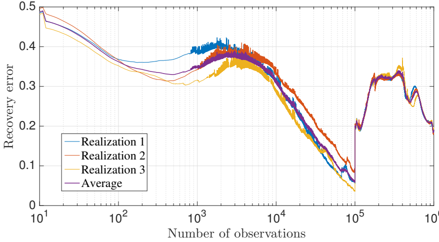
- The online scheme can track the performance of the batch inference

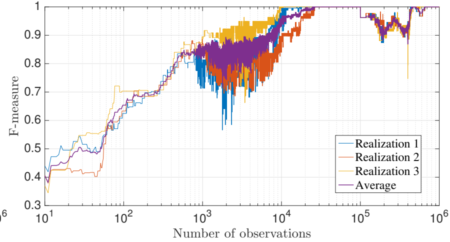
  $\Rightarrow$ The fluctuations are due to ADMM and online scheme

- Consider an Erdős-Rényi graph with $N=20$ and $p=0.2$
  - Signals diffused by $\mathbf{H} = \sum_{l=0}^{2} h_l \mathbf{A}^l$, $h_l \sim \mathcal{U}[0,1]$, $\mathbf{S}=\mathbf{A}$
  - Generate streaming signals $\{\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(p)}, \mathbf{y}^{(p+1)}, \cdots\}$ via $\mathbf{y}^{(i)} = \mathbf{H}\mathbf{x}^{(i)}$
  - Upon sensing an observation $\mathbf{y}^{(p)}$

    $\Rightarrow$ Update $\hat{\mathbf{V}}$ efficiently and run the algorithm for $T_1 = 1$
  - After 10^5 realizations

    $\Rightarrow$ Remove 10% of edges and add the same number of edges elsewhere
  - Assess the recovery error $\xi_F := \|\hat{\mathbf{S}} - \mathbf{S}\|_F / \|\mathbf{S}\|_F$ and F-measure





- The online algorithm can adapt and learn the new topology

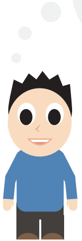
Closing remarks

- Online topology inference from streaming stationary graph signals
  - Graph shift $\mathbf{S}$ and covariance $\mathbf{C}_y$ are simultaneously diagonalizable
  - Promote desirable properties via convex losses on $\mathbf{S}$ $\Rightarrow$ Here: Sparsity

- Developed an iterative algorithm for the topology inference
- Upon sensing new diffused output signals

$\Rightarrow$ - Updated $\hat{\mathbf{V}}$ efficiently

- Took one or a few steps of the iterative algorithm



Back to the $\mathbf{D}$-update

- Recall **Step 2.** $\mathbf{D}^{(k+1)} = \operatorname*{argmin}_{\mathbf{D}\in\mathcal{S}} \quad \|\mathbf{D} - (\mathbf{S} + \mathbf{U}_1)\|_F^2$
- Define $\mathcal{C}_1 = \{\mathbf{M}|\mathbf{M} = \mathbf{M}^\top, \text{diag}(\mathbf{M}) = \mathbf{0}\}$ and $\mathcal{C}_2 = \{\mathbf{M}|\mathbf{M} \geq \mathbf{0}, \sum_{i=1}^{N} M_{1i} = 1\}$

  $\Rightarrow \mathcal{S} = \mathcal{C}_1 \cap \mathcal{C}_2$

  $\Rightarrow$ Establish an inner ADMM for the $\mathbf{D}$-update [Boyd et al'11]

$$\min_{\mathbf{E},\mathbf{Z}} \quad \|\mathbf{E} - (\mathbf{S}^{(k+1)} + \mathbf{U}_1{}^{(k)})\|_F^2 + g_1(\mathbf{E}) + g_2(\mathbf{Z})$$

$$\text{s.to:} \qquad \mathbf{E} - \mathbf{Z} = \mathbf{0},$$

Input: penalty parameter ρ_2, number of iterations T_2.
Initialize: $\mathbf{E}^{(0)} = \mathbf{Z}^{(0)} = \mathbf{U}_2{}^{(0)} = \mathbf{0}$.
for $i = 0, \ldots, T_2 - 1$ **do**
  $\mathbf{E}^{(i+1)} = \mathcal{P}_1\left(\frac{\mathbf{S}^{(k+1)} + \mathbf{U}_1{}^{(k)} + \frac{\rho_2}{2}(\mathbf{Z}^{(i)} - \mathbf{U}_2{}^{(i)})}{1 + \frac{\rho_2}{2}}\right) \Rightarrow \mathcal{P}_1(\mathbf{M}) = \frac{\mathbf{M}+\mathbf{M}^\top}{2} - \text{Diag}(\mathbf{M})$
  $\mathbf{Z}^{(i+1)} = \mathcal{P}_2(\mathbf{E}^{(i+1)} + \mathbf{U}_2{}^{(i)}) \Rightarrow$ Projection onto a simplex [Chen et al'11]
  $\mathbf{U}_2{}^{(i+1)} = \mathbf{U}_2{}^{(i)} + \mathbf{E}^{(i+1)} - \mathbf{Z}^{(i+1)}$
end for
return $\mathbf{D}^{(k+1)} := \mathbf{E}^{(T_2)}$