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 as graph $G = (\mathcal{V}, \mathcal{E})$: encode pairwise relationships

Desiderata: Process, analyze and learn from network data [Kolaczyk’09]
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 $A$
  \[ A_{ij} = \text{Proximity between } i \text{ and } j \]
- Define a signal $x$ on top of the graph
  \[ x_i = \text{Signal value at node } i \]
Graph signal processing (GSP)

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

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

Undirected $G$ with adjacency matrix $A$

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

Define a signal $x$ on top of the graph

$x_i = \text{Signal value at node } i$

Associated with $G$ is the graph-shift operator (GSO) $S = \mathbf{V}\Lambda\mathbf{V}^T \in \mathbb{M}^N$

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

Ex: $A$, degree $D$ and Laplacian $L = D - A$ matrices

Graph Signal Processing $\rightarrow$ Exploit structure encoded in $S$ to process $x$

$\Rightarrow$ GSP well suited to study (network) diffusion processes

Use GSP to learn the underlying $G$ or a meaningful network model
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]
  ⇒ 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 $y$ is the response of a linear diffusion process to an input $x$

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

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

- One can state that the graph shift $S$ explains the structure of signal $y$
Generating structure of a diffusion process

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

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

⇒ Common generative model. Heat diffusion if $\alpha_k$ constant

- One can state that the graph shift $S$ explains the structure of signal $y$

- Cayley-Hamilton asserts that we can write diffusion as

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

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

⇒ Shift invariant operator $H$ is graph filter [Sandryhaila-Moura’13]

- Online topology inference: From $\mathcal{Y} = \{ y^{(1)}, \ldots, y^{(P)}, \ldots \}$, Find $S$ efficiently
Def: A graph signal \( y \) is stationary with respect to the shift \( S \) if and only if \( y = Hx \), where \( H = \sum_{l=0}^{L-1} h_l S^l \) and \( x \) is white.

- The covariance matrix of the stationary signal \( y \) is

\[
C_y = \mathbb{E} \left[ Hx(Hx)^T \right] = HE \left[ xx^T \right] H^T = HH^T
\]

- Key: Since \( H \) is diagonalized by \( V \), so is the covariance \( C_y \)

\[
C_y = V \left| \sum_{l=0}^{L-1} h_l \Lambda^l \right|^2 V^T = V \left( \mathcal{H}(\Lambda) \right)^2 V^T
\]

⇒ Estimate \( V \) from \( \mathcal{Y} \) via Principal Component Analysis
Two-step approach [Segarra et al’17]

- **Step 1:** Identify the eigenvectors of the shift via $\hat{C}_Y$
  - **Inferred eigenvectors $\hat{V}$**

- **Step 2:** Identify eigenvalues to obtain a suitable shift
  - **Inferred network $\hat{S}$**

**Online Topology Inference from Streaming Stationary Graph Signals**

▶ **Step 2:** Obtaining the eigenvalues of $S$

▸ We can use extra knowledge/assumptions to choose one graph
  ⇒ Of all graphs, select one that is optimal in the number of edges

$$\hat{S} := \arg\min_{S,\Lambda} \|S\|_1 \quad \text{subject to:} \quad \|S - \hat{V}\Lambda\hat{V}^T\|_F \leq \epsilon, \; S \in S$$

▸ Set $S$ contains all admissible scaled adjacency matrices

$$S := \{S \mid S_{ij} \geq 0, \; S \in \mathcal{M}^N, \; S_{ii} = 0, \; \sum_j S_{1j} = 1\}$$
Consider streaming stationary signals $\mathcal{Y} := \{y^{(1)}, \ldots, y^{(p)}, y^{(p+1)}, \ldots\}$

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

Assume that time differences of the signals arrival is relatively low.
Online inference under stationarity

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

Assume that time differences of the signals arrival is relatively low.
Online inference under stationarity

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

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
  - Update \( \hat{V} \) efficiently
  - Take one or a few steps of the iterative algorithm
To apply ADMM, rewrite the problem as

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

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

⇒ Convex, thus ADMM would converge to a global minimizer

Form the augmented Lagrangian

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

At $k^{th}$ iteration, let $B^{(k)} = \hat{V} \Lambda^{(k)} \hat{V}^\top$ ⇒ ADMM consists of 4 iterative steps
To apply ADMM, rewrite the problem as

\[
\min_{S, \Lambda, D} \lambda \|S\|_1 + \|S - \hat{V}\Lambda\hat{V}^T\|_F^2
\]

s.to: \(S - D = 0, \quad D \in S = \{S \mid S_{ij} \geq 0, \ S \in \mathcal{M}_N, \ S_{ii} = 0, \ \sum_j S_{1j} = 1\}\)

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

Form the augmented Lagrangian

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

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

Step 1. \(S^{(k+1)} = \arg\min_S \mathcal{L}_{\rho_1}(S, D^{(k)}, \Lambda^{(k)}, U^{(k)}) = \mathcal{T}_{\frac{\lambda}{2+\rho_1}} \left( \frac{B^{(k)} + \frac{\rho_1}{2} (D^{(k)} - U^{(k)})}{1 + \frac{\rho_1}{2}} \right),\)

where \(\mathcal{T}_\eta(x) = (|x| - \eta)_+\) is the element-wise soft-thresholding operator.
To apply ADMM, rewrite the problem as

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

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

⇒ Convex, thus ADMM would converge to a global minimizer

Form the augmented Lagrangian

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

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

**Step 2.** $D^{(k+1)} = \arg\min_{D \in S} \mathcal{L}_{\rho_1}(S^{(k+1)}, D, \Lambda^{(k)}, U^{(k)}) = \mathcal{P}_S(S^{(k+1)} + U^{(k)})$, where $\mathcal{P}_S(.)$ is the projection operator onto $S$
To apply ADMM, rewrite the problem as

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

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

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

Form the augmented Lagrangian

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

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

Step 3. $\Lambda^{(k+1)} = \arg\min_{\Lambda} L_{\rho_1}(S^{(k+1)}, D^{(k+1)}, \Lambda, U^{(k)})$
To apply ADMM, rewrite the problem as

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

s.t. $S - D = 0$, $D \in S = \{S | S_{ij} \geq 0, S \in \mathcal{M}_N, S_{ii} = 0, \sum_j S_{1j} = 1\}$

⇒ Convex, thus ADMM would converge to a global minimizer.

Form the augmented Lagrangian

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

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

Step 3. $\Lambda^{(k+1)} = \arg\min_{\Lambda} \|\Lambda - \hat{V}^\top S^{(k+1)}\hat{V}\|_F^2 = \text{Diag}(\hat{V}^\top S^{(k+1)}\hat{V})$
Topology inference via ADMM

- To apply ADMM, rewrite the problem as

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

s.to: \( \mathbf{S} - \mathbf{D} = 0 \), \( \mathbf{D} \in \mathcal{S} = \{\mathbf{S} | S_{ij} \geq 0, \mathbf{S} \in \mathcal{M}_N^N, S_{ii} = 0, \sum_j S_{1j} = 1\} \)

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

- Form the augmented Lagrangian

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

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

- Step 4. Dual gradient ascent update \( \mathbf{U}^{(k+1)} = \mathbf{U}^{(k)} + \mathbf{S}^{(k+1)} - \mathbf{D}^{(k+1)} \)
1: Input: estimated covariance eigenvectors $\hat{V}$, penalty parameter $\rho_1$, regularization parameter $\lambda$, number of iterations $T_1$

2: Initialize: $\Lambda^{(0)} = \text{diag}(1_N)$, $D^{(0)} = 0$, $U^{(0)} = 0$.

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

4: $B^{(k)} = \hat{V} \Lambda^{(k)} \hat{V}^\top$

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

6: $D^{(k+1)} = \mathcal{P} S^{(k+1)} + U^{(k)}$

7: $\Lambda^{(k+1)} = \text{Diag}(\hat{V}^\top S^{(k+1)} \hat{V})$

8: $U^{(k+1)} = U^{(k)} + S^{(k+1)} - D^{(k+1)}$

9: end for

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

- Develop an iterative algorithm for the topology inference
- Upon sensing new diffused output signals
  - Update $\hat{V}$ efficiently
  - Take one or a few steps of the iterative algorithm
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 $H = \sum_{l=0}^{2} h_l A^l$, $h_l \sim \mathcal{U}[0, 1]$, $S = A$.
- Adopt sample covariance estimator for the Gaussian signals.
- Assess the recovery error $\xi_F := \|\hat{S} - S\|_F / \|S\|_F$ and F-measure.

Increase in number of observations leads to a better performance.

⇒ Performance enhances for sparser graphs (i.e., smaller $p$).
Online topology inference

- **Q:** How can we efficiently update the sample covariance eigenvectors \( \hat{V} \)?

- Let \( \hat{C}_y^{(P)} \) be sample covariance after receiving \( P \) streaming observations.

  \[ \hat{C}_y^{(P+1)} = \frac{1}{P+1} (P \hat{C}_y^{(P)} + y^{(P+1)} y^{(P+1)}) \]

  ⇒ Updated sample covariance after receiving \( y^{(P+1)} \) takes the form.
Online topology inference

- **Q:** How can we efficiently update the sample covariance eigenvectors \( \hat{V} \)?

- Let \( \hat{C}_y(P) \) be sample covariance after receiving \( P \) streaming observations
  
  \[ \Rightarrow \text{Updated sample covariance after receiving } y^{(P+1)} \text{ takes the form} \]
  
  \[ \hat{C}_y(P+1) = \frac{1}{P+1} \left( P\hat{C}_y(P) + y^{(P+1)}y^{(P+1)^T} \right) \]

- Let \( z = \hat{V}^T y^{(p+1)} \) and \( \{d_j\}_{j=1}^N \) denote the eigenvalues of \( \hat{C}_y(P) \)
  
  \[ \Rightarrow \text{Eigenvalues of rank-one modification of } \hat{C}_y(P) \text{ are the roots } (\gamma) \text{ of} \]
  
  \[ 1 + \sum_{j=1}^N \frac{z_j^2}{P d_j - \gamma} = 0 \quad \text{[Bunch et al’78]} \]

  \[ \Rightarrow \text{Can be solved using the } \textbf{Newton} \text{ method with } O(N^2) \text{ complexity} \]

- For the updated eigenvalue \( \gamma_j \), the corresponding eigenvector \( v_j \) is given by
  
  \[ v_j = \alpha_j y^{(p+1)} \odot q_j, \]

  where \( q_j = [1/(P d_1 - \gamma_j), \ldots, 1/(P d_N - \gamma_j)] \) and \( \alpha_j \) is a normalizing factor
Consider a structural brain graph with $N = 66$ neural regions.

- Edge weights: Density of anatomical connections [Hagmann et al'08]
- Signals diffused by $H = \sum_{l=0}^{2} h_l A^l$, $h_l \sim \mathcal{U}[0, 1]$, $S = A$
- Generate streaming signals $\{y^{(1)}, \ldots, y^{(p)}, y^{(p+1)}, \ldots\}$ via $y^{(i)} = Hx^{(i)}$
- Upon sensing an observation $y^{(p)}$
  - Update $\hat{V}$ efficiently and run the algorithm for $T_1 = 1$

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 $H = \sum_{l=0}^{2} h_l A^l$, $h_l \sim \mathcal{U}[0, 1]$, $S = A$
- Generate streaming signals $\{y^{(1)}, \ldots, y^{(p)}, y^{(p+1)}, \ldots\}$ via $y^{(i)} = H x^{(i)}$
- Upon sensing an observation $y^{(p)}$
  - Update $\hat{V}$ efficiently and run the algorithm for $T_1 = 1$
- After $10^5$ realizations
  - Remove 10% of edges and add the same number of edges elsewhere

The online algorithm can adapt and learn the new topology
Online topology inference from streaming stationary graph signals

- Graph shift $S$ and covariance $C_y$ are simultaneously diagonalizable
- Promote desirable properties via convex losses on $S$ ⇒ Here: Sparsity

- Developed an iterative algorithm for the topology inference
- Upon sensing new diffused output signals
  ⇒ - Updated $\hat{V}$ efficiently
  - Took one or a few steps of the iterative algorithm