Online Proximal Gradient for Learning Graphs from Streaming Signals

Rasoul Shafipour and **Gonzalo Mateos**

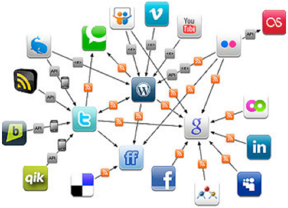
Dept. of ECE and Goergen Institute for Data Science
University of Rochester
gmateosb@ece.rochester.edu
http://www.ece.rochester.edu/~gmateosb/

Acknowledgment: NSF Awards CCF-1750428, ECCS-1809356, CCF-1934962

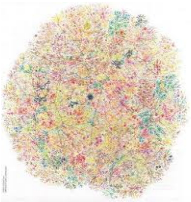
Amsterdam, Netherlands, January 18-22, 2021

Network Science analytics

Online social media



Internet



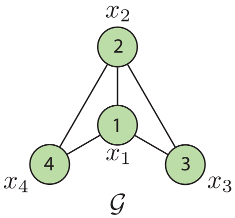
Clean energy and grid analytics

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

  $\Rightarrow$ Use G to study graph signals, data associated with nodes in $\mathcal{V}$
- Ex: Opinion profile, buffer congestion levels, neural activity, epidemic
- **Q:** What about streaming data from (possibly) dynamic networks?

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



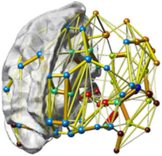
- Associated with $\mathcal{G}$ is the graph-shift operator (GSO) $\mathbf{S} = \mathbf{V \Lambda V}^T$
  - $\Rightarrow S_{ij} = 0$ for $i \neq j$ and $(i, j) \notin \mathcal{E}$ (local structure in $\mathcal{G}$)
  - $\Rightarrow$ Ex: $\mathbf{A}$, degree $\mathbf{D}$ and Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ matrices
- Graph Signal Processing $\rightarrow$ Exploit structure encoded in $\mathbf{S}$ to process $\mathbf{x}$
  - $\Rightarrow$ 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]

- Key in neuroscience [Sporns'10]

  $\Rightarrow$ Functional network from BOLD signal



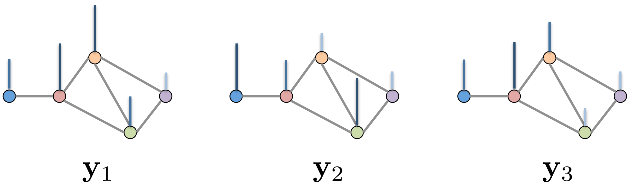
- Noteworthy GSP-based approaches
  - Graphical models [Egilmez et al'16], [Rabbat'17], [Kumar et al'19], . . .
  - Smooth signals [Dong et al'15], [Kalofolias'16], [Sardellitti et al'17], . . .
  - Stationary signals [Pasdeloup et al'15], [Segarra et al'16], . . .
  - Dynamic graphs [Shen et al'16], [Kalofolias et al'17], [Cardoso et al'20], . . .
  - Streaming data [Shafipour et al'18], [Vlaski et al'18], [Natali et al'20], . . .

- **Our contribution:** graph learning from streaming stationary signals
  - Topology inference via convergent online proximal gradient (PG) iterations

Problem formulation

Setup

- Sparse network $\mathcal{G}$ with unknown graph shift $\mathbf{S}$ (even dynamic $\mathbf{S}_t$)
- Observe
  - $\Rightarrow$ Streaming stationary signals $\{\mathbf{y}_t\}_{t=1}^{T}$ defined on $\mathbf{S}$
  - $\Rightarrow$ Edge status s_{ij} for $(i, j) \in \Omega \subset \mathcal{V} \times \mathcal{V}$



Problem statement

Given observations $\{\mathbf{y}_t\}_{t=1}^{T}$ and edge status in Ω, determine the network $\mathbf{S}$ knowing that $\{\mathbf{y}_t\}_{t=1}^{T}$ are generated via diffusion on $\mathbf{S}$.

Generating structure of a diffusion process

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

$$\mathbf{y}_t = \alpha_0 \prod_{l=1}^{\infty}(\mathbf{I} - \alpha_l \mathbf{S})\mathbf{x}_t = \sum_{l=0}^{\infty} \beta_l \mathbf{S}^l \mathbf{x}_t, \quad t = 1, \ldots, T$$

$\Rightarrow$ Common generative model, e.g., heat diffusion, consensus

- Cayley-Hamilton asserts we can write diffusion as $(L \leq N)$

$$\mathbf{y}_t = \left(\sum_{l=0}^{L-1} h_l \mathbf{S}^l\right)\mathbf{x}_t := \mathbf{H}\mathbf{x}_t, \quad t = 1, \ldots, T$$

$\Rightarrow$ Graph filter $\mathbf{H}$ is shift invariant [Sandryhaila-Moura'13]

- **Goal**: estimate undirected network $\mathbf{S}$ online from signals $\{\mathbf{y}_t\}_{t=1}^{T}$

$\Rightarrow$ Unknowns: filter order L, coefficients $\{h_l\}_{l=1}^{L-1}$, inputs $\{\mathbf{x}_t\}_{t=1}^{T}$

Batch topology inference under stationarity

- Suppose that the input is white, i.e., $\mathbf{C}_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right] = \mathbf{I}$

  $\Rightarrow$ The covariance matrix of $\mathbf{y} = \mathbf{H}\mathbf{x}$ is a polynomial in $\mathbf{S}$

$$\mathbf{C}_y = \mathbb{E}\left[\mathbf{H}\mathbf{x}\left(\mathbf{H}\mathbf{x}\right)^T\right] = \mathbf{H}^2 = h_0^2\mathbf{I} + 2h_0h_1\mathbf{S} + h_1^2\mathbf{S}^2 + \ldots$$

- Implies $\mathbf{C}_y\mathbf{S} = \mathbf{S}\mathbf{C}_y$, shift-invariant second-order statistics (stationarity)

- **Formulation:** given $\hat{\mathbf{C}}_y$, search for $\mathbf{S}$ that is sparse and feasible

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

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

$$\mathcal{S} := \{\mathbf{S} \,|\, S_{ij} \geq 0, \mathbf{S}^T = \mathbf{S}, S_{ii} = 0, S_{ij} = s_{ij}, \ (i,j) \in \Omega\}$$

Batch proximal gradient algorithm

- Dualize the constraint to arrive at the convex, composite cost $F(\mathbf{S})$

$$\mathbf{S}^\star \in \underset{\mathbf{S}\in\mathcal{S}}{\text{argmin}}\, F(\mathbf{S}) := \|\mathbf{S}\|_1 + \underbrace{\frac{\mu}{2}\|\mathbf{S}\hat{\mathbf{C}}_y - \hat{\mathbf{C}}_y\mathbf{S}\|_F^2}_{g(\mathbf{S})}$$

- Smooth component $g(\mathbf{S})$ has an $M = 4\mu\lambda^2_{\max}(\hat{\mathbf{C}}_y)$-Lipschitz gradient

$$\nabla g(\mathbf{S}) = \mu\big[(\mathbf{S}\hat{\mathbf{C}}_y - \hat{\mathbf{C}}_y\mathbf{S})\hat{\mathbf{C}}_y - \hat{\mathbf{C}}_y(\mathbf{S}\hat{\mathbf{C}}_y - \hat{\mathbf{C}}_y\mathbf{S})\big]$$

- Convergent PG updates with stepsize $\gamma < \frac{2}{M}$ at iteration $k = 1, 2, \ldots$

$$\mathbf{S}_{k+1} = \text{prox}_{\gamma\|\cdot\|_1,\mathcal{S}}\left(\mathbf{S}_k - \gamma\nabla g(\mathbf{S}_k)\right)$$

- Proximal operator $(\mathbf{D}_k := \mathbf{S}_k - \gamma\nabla g(\mathbf{S}_k))$

$$[\mathbf{S}_{k+1}]_{ij} = \begin{cases} 0, & i = j \\ s_{ij}, & (i,j) \in \Omega \\ \max(0, [\mathbf{D}_k]_{ij} - \gamma), & \text{otherwise.} \end{cases}$$

Online proximal gradient algorithm

- **Q:** Online estimation from streaming data $\mathbf{y}_1, \ldots, \mathbf{y}_t, \mathbf{y}_{t+1}, \ldots$?
  - At time t solve the time-varying composite optimization

$$\mathbf{S}_t^\star \in \operatorname*{argmin}_{\mathbf{S}\in\mathcal{S}} F_t(\mathbf{S}) := \|\mathbf{S}\|_1 + \underbrace{\frac{\mu}{2}\|\mathbf{S}\hat{\mathbf{C}}_{\mathbf{y},t} - \hat{\mathbf{C}}_{\mathbf{y},t}\mathbf{S}\|_F^2}_{g_t(\mathbf{S})}$$

- **Step 1:** Recursively update the sample covariance $\hat{\mathbf{C}}_{y,t}$

$$\hat{\mathbf{C}}_{y,t} = \frac{1}{t}\left((t-1)\hat{\mathbf{C}}_{y,t-1} + \mathbf{y}_t\mathbf{y}_t^T\right)$$

  - Track $\mathbf{S}_t$ $\Rightarrow$ Sliding window or exponentially-weighted moving average

- **Step 2:** Run a single iteration of the PG algorithm [Madden et al'18]

$$\mathbf{S}_{t+1} = \text{prox}_{\gamma_t\|\cdot\|_1,\mathcal{S}}\left(\mathbf{S}_t - \gamma_t \nabla g_t(\mathbf{S}_t)\right)$$

  - Memory footprint and computational complexity does not grow with t

Convergence analysis

Theorem (Madden et al'18)

Let $\nu_t := \left\| \mathbf{S}^\star_{t+1} - \mathbf{S}^\star_t \right\|_F$ capture the variability of the optimal solution. If g_t is strongly convex with constant m_t (details in the paper), then for all $t \geq 1$ the iterates $\mathbf{S}_t$ generated by the online PG algorithm satisfy

$$\|\mathbf{S}_t - \mathbf{S}^\star_t\|_F \leq \tilde{L}_{t-1} \left(\|\mathbf{S}_0 - \mathbf{S}^*_0\|_F + \sum_{\tau=0}^{t-1} \frac{\nu_\tau}{\tilde{L}_\tau} \right),$$

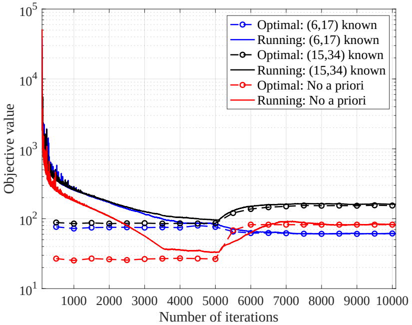
where $L_t = \max\left\{ |1 - \gamma_t m_t|, |1 - \gamma_t M_t| \right\}, \tilde{L}_t = \prod_{\tau=0}^{t} L_\tau$.

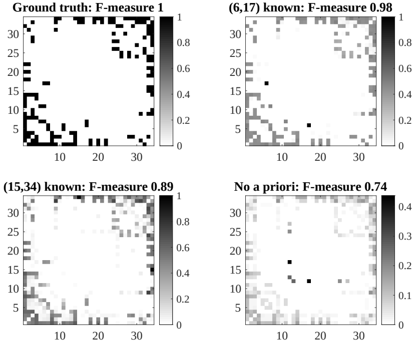
- **Corollary:** Define $\hat{L}_t := \max_{\tau=0,\ldots,t} L_\tau$, $\hat{\nu}_t := \max_{\tau=0,\ldots,t} \nu_\tau$. Then

$$\|\mathbf{S}_t - \mathbf{S}^\star_t\|_F \leq \left(\hat{L}_{t-1}\right)^t \|\mathbf{S}_0 - \mathbf{S}^\star_0\|_F + \frac{\hat{\nu}_t}{1 - \hat{L}_{t-1}}$$

  - For $m_\tau \geq m$, $M_\tau \leq M$, and $\gamma_\tau = 2/(m_\tau + M_\tau)$ $\Rightarrow \hat{L}_t \leq \frac{M-m}{M+m} < 1$
  - Misadjustment grows with $\hat{\nu}_t$ and bad conditioning ($M \to \infty$ or $m \to 0$)

- Zachary's karate club social network with $N = 34$ nodes
  - Diffusion filter $\mathbf{H} = \sum_{l=0}^{2} h_l \mathbf{A}^l$, $h_l \sim \mathcal{U}[0,1]$
  - Generate streaming signals $\mathbf{y}_1, \ldots, \mathbf{y}_t, \mathbf{y}_{t+1}, \ldots$ via $\mathbf{y}_t = \mathbf{H}\mathbf{x}_t$
  - Both batch and online inference for different Ω (one edge observed)
  - Dynamic $\mathbf{S}_t$: flip 10% of the edges at random at $t = 5000$





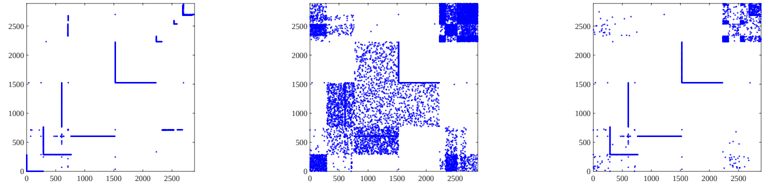
- The online scheme attains the performance of its batch counterpart

Facebook friendship graph

- Facebook friendship graph with $N = 2888$ nodes. Ego-nets of 7 users

Number of observations	10^3	10^4	10^5	10^6
F-measure	0.45	0.77	0.87	0.94

- Ground-truth $\mathbf{A}$ (left) and $\mathbf{S}_t$ for $t = 10^4$ (center) and $t = 10^6$ (right)



- Scalable to graphs with several thousand nodes

Closing remarks

- Topology inference from streaming diffused graph signals
  - Graph shift $\mathbf{S}$ and covariance $\mathbf{C}_y$ commute
  - Promote desirable properties on $\mathbf{S}$ via convex criteria
- Online PG algorithm with quantifiable performance
  - Estimates hover around the optimal time-varying batch solution
  - Iterations scale to graphs with several thousand nodes
  - Tacks the network's dynamic behavior
- Ongoing work
  - Task-oriented (i.e., classification) discriminative graph learning
  - Nesterov-type accelerated algorithms
  - Observations of streaming signals that are smooth on $\mathbf{S}$

Extended version https://doi.org/10.3390/a13090228