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

Desiderata: Process, analyze and learn from network data [Kolaczyk’09]
⇒ Use $\mathcal{G}$ 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 $G$ with adjacency matrix $A$
  $A_{ij} =$ Proximity between $i$ and $j$

- Define a signal $x$ on top of the graph
  $x_i =$ Signal value at node $i$

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

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

Key in neuroscience [Sporns’10]
  ⇒ 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 $G$ with unknown graph shift $S$ (even dynamic $S_t$)
- Observe
  $$\Rightarrow \text{Streaming stationary signals } \{y_t\}_{t=1}^T \text{ defined on } S$$
  $$\Rightarrow \text{Edge status } s_{ij} \text{ for } (i,j) \in \Omega \subset \mathcal{V} \times \mathcal{V}$$

Problem statement

Given observations $\{y_t\}_{t=1}^T$ and edge status in $\Omega$, determine the network $S$ knowing that $\{y_t\}_{t=1}^T$ are generated via diffusion on $S$. 
Generating structure of a diffusion process

- Signal $y_t$ is the response of a linear diffusion process to input $x_t$

\[
y_t = \alpha_0 \prod_{l=1}^{\infty} (I - \alpha_l S) x_t = \sum_{l=0}^{\infty} \beta_l S^l x_t, \quad t = 1, \ldots, T
\]

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

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

\[
y_t = \left( \sum_{l=0}^{L-1} h_l S^l \right) x_t := H x_t, \quad t = 1, \ldots, T
\]

$\Rightarrow$ Graph filter $H$ is shift invariant [Sandryhaila-Moura’13]

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

$\Rightarrow$ **Unknowns**: filter order $L$, coefficients $\{h_l\}_{l=1}^{L-1}$, inputs $\{x_t\}_{t=1}^{T}$
Suppose that the input is white, i.e., \( C_x = \mathbb{E} \left[ xx^T \right] = I \)

\( \Rightarrow \) The covariance matrix of \( y = Hx \) is a polynomial in \( S \)

\[
C_y = \mathbb{E} \left[ Hx(Hx)^T \right] = H^2 = h_0^2I + 2h_0h_1S + h_1^2S^2 + \ldots
\]

\( \Rightarrow \) Implies \( C_yS = SC_y \), shift-invariant second-order statistics (stationarity)

**Formulation:** given \( \hat{C}_y \), search for \( S \) that is sparse and feasible

\[
\hat{S} := \arg\min_S \|S\|_1 \quad \text{subject to:} \quad \|SC_y - \hat{C}_yS\|_F \leq \epsilon, \quad S \in S
\]

\( \Rightarrow \) Set \( S \) contains all admissible scaled adjacency matrices

\[
S := \{S \mid S_{ij} \geq 0, S^T = S, S_{ii} = 0, S_{ij} = s_{ij}, (i, j) \in \Omega\}
\]
Batch proximal gradient algorithm

- Dualize the constraint to arrive at the convex, composite cost $F(S)$
  \[ S^* \in \arg \min_{S \in \mathcal{S}} F(S) := \|S\|_1 + \frac{\mu}{2} \|SC_y - C_yS\|_F^2 \]

- Smooth component $g(S)$ has an $M = 4\mu \lambda^2 \max(\hat{C}_y)$-Lipschitz gradient
  \[ \nabla g(S) = \mu [(SC_y - C_yS)C_y - C_y(SC_y - C_yS)] \]

- Convergent PG updates with stepsize $\gamma < \frac{2}{M}$ at iteration $k = 1, 2, \ldots$
  \[ S_{k+1} = \text{prox}_{\gamma\|\cdot\|_1, S} (S_k - \gamma \nabla g(S_k)) \]

- Proximal operator $(D_k := S_k - \gamma \nabla g(S_k))$

  \[ [S_{k+1}]_{ij} = \begin{cases} 
  0, & i = j \\
  s_{ij}, & (i, j) \in \Omega \\
  \max(0, [D_k]_{ij} - \gamma), & \text{otherwise.} 
  \end{cases} \]
Online proximal gradient algorithm

- **Q:** Online estimation from streaming data $y_1, \ldots, y_t, y_{t+1}, \ldots$?
  - At time $t$ solve the time-varying composite optimization

$$
S_t^* \in \arg\min_{S \in S} F_t(S) := \|S\|_1 + \frac{\mu}{2} \|S \hat{C}_{y,t} - \hat{C}_{y,t} S\|_F^2 + \mu g_t(S)
$$

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

$$
\hat{C}_{y,t} = \frac{1}{t} \left( (t-1)\hat{C}_{y,t-1} + y_t y_t^T \right)
$$
  - Track $S_t \Rightarrow$ Sliding window or exponentially-weighted moving average

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

$$
S_{t+1} = \text{prox}_{\gamma_t \| \cdot \|_1, S} \left( S_t - \gamma_t \nabla g_t(S_t) \right)
$$
  - Memory footprint and computational complexity does not grow with $t$
Convergence analysis

**Theorem (Madden et al’18)**

Let \( \nu_t := \|S_{t+1}^* - S_t^*\|_F \) 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 \( S_t \) generated by the online PG algorithm satisfy

\[
\|S_t - S_t^*\|_F \leq \tilde{L}_{t-1} \left( \|S_0 - S_0^*\|_F + \sum_{\tau=0}^{t-1} \frac{\nu_{\tau}}{\tilde{L}_{\tau}} \right),
\]

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

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

\[
\|S_t - S_t^*\|_F \leq \left( \hat{L}_{t-1} \right)^t \|S_0 - S_0^*\|_F + \frac{\hat{\nu}_t}{1 - \hat{L}_{t-1}}
\]

- For \( m_\tau \geq m, M_\tau \leq M \), and \( \gamma_\tau = \frac{2}{(m_\tau + M_\tau)} \Rightarrow \hat{L}_t \leq \frac{M-m}{M+m} < 1 \)
- **Misadjustment** grows with \( \hat{\nu}_t \) and bad conditioning \( (M \to \infty \text{ or } m \to 0) \)
Zachary’s karate club network

- Zachary’s karate club social network with $N = 34$ nodes
  - Diffusion filter $H = \sum_{l=0}^{2} h_l A^l$, $h_l \sim \mathcal{U}[0, 1]$
  - Generate streaming signals $y_1, \ldots, y_t, y_{t+1}, \ldots$ via $y_t = Hx_t$
  - Both batch and online inference for different $\Omega$ (one edge observed)
  - Dynamic $S_t$: flip 10% of the edges at random at $t = 5000$

The online scheme attains the performance of its batch counterpart

---

**Ground truth:** F-measure 1

- **Optimal:** (6,17) known
- **Running:** (6,17) known

- **Optimal:** (15,34) known
- **Running:** (15,34) known

- **Optimal:** No a priori
- **Running:** No a priori

**F-measures:**
- (6,17) known: F-measure 0.98
- (15,34) known: F-measure 0.89
- No a priori: F-measure 0.74
Facebook friendship graph

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

<table>
<thead>
<tr>
<th>Number of observations</th>
<th>$10^3$</th>
<th>$10^4$</th>
<th>$10^5$</th>
<th>$10^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-measure</td>
<td>0.45</td>
<td>0.77</td>
<td>0.87</td>
<td>0.94</td>
</tr>
</tbody>
</table>

- Ground-truth $A$ (left) and $S_t$ 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 $S$ and covariance $C_y$ commute
  - Promote desirable properties on $S$ 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 $S$

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