Dual-based Online Learning of Dynamic Network Topologies

Seyed Saman Saboksayr and Gonzalo Mateos

Dept. of Electrical and Computer Engineering

University of Rochester

Email: gmateosb@ece.rochester.edu

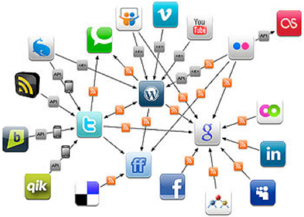
Web-page: http://hajim.rochester.edu/ece/sites/gmateos/

Acknowledgement: NSF awards CCF-1750428 and CCF-1934962

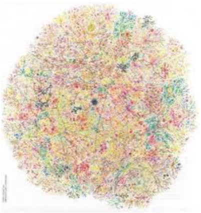
Rhodes Island, Greece, June 8, 2023

What is this talk about?

Online social media



Internet



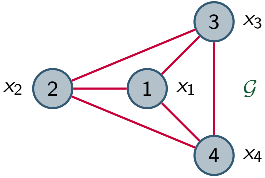
Clean energy and grid analytics

- Learning undirected graphs from nodal observations
  - ⇒ Ex: Functional brain connectivity from brain signals
- **Q:** What about streaming data from (possibly) dynamic networks?

G. B. Giannakis et al, "Topology identification and learning over graphs: Accounting for nonlinearities and dynamics," *Proc. IEEE*, 2018

Graph signal processing (GSP)

- Graph $\mathcal{G}$ with adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$
  - $\Rightarrow W_{ij}$ = proximity between i and j
- Define a signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph
  - $\Rightarrow x_i$ = signal value at node $i \in \mathcal{V}$

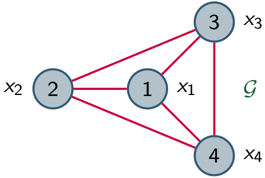


- Total variation of signal $\mathbf{x}$ with respect to Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$

$$\text{TV}(\mathbf{x}) = \mathbf{x}^\top \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{i \neq j} W_{ij}(x_i - x_j)^2$$

- Graph Signal Processing $\rightarrow$ Exploit structure encoded in $\mathbf{L}$ to process $\mathbf{x}$
  - $\Rightarrow$ Use GSP to learn the underlying $\mathcal{G}$ or a meaningful network model

- Graph $\mathcal{G}$ with adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$
  - $\Rightarrow W_{ij}$ = proximity between i and j
- Define a signal $\mathbf{x} \in \mathbb{R}^{N}$ on top of the graph
  - $\Rightarrow x_i$ = signal value at node $i \in \mathcal{V}$

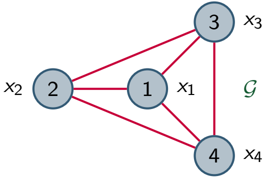


- Total variation of signal $\mathbf{x}$ with respect to Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$

$$\text{TV}(\mathbf{x}) = \mathbf{x}^\top \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{i \neq j} W_{ij} (x_i - x_j)^2$$

- Graph Signal Processing → Exploit structure encoded in $\mathbf{L}$ to process $\mathbf{x}$
  - $\Rightarrow$ Use GSP to learn the underlying $\mathcal{G}$ or a meaningful network model

- Graph $\mathcal{G}$ with adjacency matrix $\mathbf{W} \in \mathbb{R}^{N \times N}$
  - $\Rightarrow W_{ij}$ = proximity between i and j
- Define a signal $\mathbf{x} \in \mathbb{R}^N$ on top of the graph
  - $\Rightarrow x_i$ = signal value at node $i \in \mathcal{V}$



- Total variation of signal $\mathbf{x}$ with respect to Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{W}$

$$\mathrm{TV}(\mathbf{x}) = \mathbf{x}^\top \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{i \neq j} W_{ij} (x_i - x_j)^2$$

- Graph Signal Processing $\rightarrow$ Exploit structure encoded in $\mathbf{L}$ to process $\mathbf{x}$
  - $\Rightarrow$ Use GSP to learn the underlying $\mathcal{G}$ or a meaningful network model

Problem formulation

- Noteworthy GSP-based approaches for undirected graphs
  - Graphical models [Egilmez et al'16], [Rabbat'17], [Kumar et al'19], ...
  - Structural constraints [Nie et al'16], [Cardoso et al'21], ...
  - Smooth signals [Dong et al'15], [Kalofolias'16], [Sardellitti et al'17], ...
  - Streaming data [Shafipour et al'18], [Natali et al'20], [Saboksayr et al'21], ...

Setup and rationale

- Sparse network $\mathcal{G}$ with unknown adjacency matrix $\mathbf{W}$ (or $\mathbf{W}_t$ in dynamic setting)
- Observe streaming smooth signals $\{\mathbf{x}_t\}_{t=1}^{T}$ defined on $\mathcal{G}$
- Seek graphs on which data admit certain regularities

Problem statement

Given a set $\mathcal{X} = \{\mathbf{x}_t\}_{t=1}^{T}$ of graph signal observations acquired by time T, learn an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ such that the observations in $\mathcal{X}$ are smooth on $\mathcal{G}$.

- Noteworthy GSP-based approaches for undirected graphs
  - Graphical models [Egilmez et al'16], [Rabbat'17], [Kumar et al'19], ...
  - Structural constraints [Nie et al'16], [Cardoso et al'21], ...
  - Smooth signals [Dong et al'15], [Kalofolias'16], [Sardellitti et al'17], ...
  - Streaming data [Shafipour et al'18], [Natali et al'20], [Saboksayr et al'21], ...

Setup and rationale

- Sparse network $\mathcal{G}$ with unknown adjacency matrix $\mathbf{W}$ (or $\mathbf{W}_t$ in dynamic setting)
- Observe streaming smooth signals $\{\mathbf{x}_t\}_{t=1}^T$ defined on $\mathcal{G}$
- Seek graphs on which data admit certain regularities

Problem statement

Given a set $\mathcal{X} = \{\mathbf{x}_t\}_{t=1}^T$ of graph signal observations acquired by time T, learn an undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ such that the observations in $\mathcal{X}$ are smooth on $\mathcal{G}$.

- Given $\mathcal{X}$ one can form the data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$

  $\Rightarrow$ A link between smoothness and sparsity (considering $\mathbf{E} \in \mathbb{R}_+^{N \times N}$ as Euclidean-distance matrix)

$$\sum_{t=1}^{T} \mathrm{TV}(\mathbf{x}_t) = \mathrm{trace}(\mathbf{X}^\top \mathbf{L} \mathbf{X}) = \frac{1}{2} \|\mathbf{W} \circ \mathbf{E}\|_1$$

- Framework for learning graphs under a smoothness prior [Kalofolias'16]

$$\min_{\mathbf{W}} \left\{ \|\mathbf{W} \circ \mathbf{E}\|_1 - \alpha \mathbf{1}^\top \log(\mathbf{W}\mathbf{1}) + \frac{\beta}{2} \|\mathbf{W}\|_F^2 \right\} \text{ s. to } \quad \mathrm{diag}(\mathbf{W}) = \mathbf{0},\ W_{ij} = W_{ji} \geq 0,\ i \neq j$$

  $\Rightarrow$ Logarithmic barrier forces positive degrees

  $\Rightarrow$ Penalize large edge-weights to control sparsity

  $\Rightarrow$ Efficient algorithms with $\mathcal{O}(N^2)$ cost: primal-dual [Kalofolias'16], ADMM [Wang et al'21]

- Given $\mathcal{X}$ one can form the data matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_T] \in \mathbb{R}^{N \times T}$

  $\Rightarrow$ A link between smoothness and sparsity (considering $\mathbf{E} \in \mathbb{R}_+^{N \times N}$ as Euclidean-distance matrix)

$$\sum_{t=1}^{T} \mathrm{TV}(\mathbf{x}_t) = \mathrm{trace}(\mathbf{X}^\top \mathbf{L} \mathbf{X}) = \frac{1}{2} \|\mathbf{W} \circ \mathbf{E}\|_1$$

- Framework for learning graphs under a smoothness prior [Kalofolias'16]

$$\min_{\mathbf{W}} \left\{ \|\mathbf{W} \circ \mathbf{E}\|_1 - \alpha \mathbf{1}^\top \log(\mathbf{W}\mathbf{1}) + \frac{\beta}{2} \|\mathbf{W}\|_F^2 \right\} \text{ s. to } \quad \mathrm{diag}(\mathbf{W}) = \mathbf{0},\ W_{ij} = W_{ji} \geq 0,\ i \neq j$$

  $\Rightarrow$ Logarithmic barrier forces positive degrees

  $\Rightarrow$ Penalize large edge-weights to control sparsity

  $\Rightarrow$ Efficient algorithms with $\mathcal{O}(N^2)$ cost: primal-dual [Kalofolias'16], ADMM [Wang et al'21]

Equivalent reformulation

- Encoding constraints on entries of $\mathbf{W}$
  - $\Rightarrow$ Hollow and symmetric $\rightarrow$ Suffices to work with $\mathbf{w} := \mathrm{vec}[\mathrm{triu}[\mathbf{W}]] \in \mathbb{R}_+^{N(N-1)/2}$
  - $\Rightarrow$ Non-negativity via penalty function: $\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} = 0$ if $\mathbf{w} \succeq \mathbf{0}$, else $\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} = \infty$
- Equivalent unconstrained, non-differentiable reformulation

$$\min_{\mathbf{w}} \Big\{ \underbrace{\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} + 2\mathbf{w}^\top \mathbf{e} + \beta\|\mathbf{w}\|_2^2}_{:=f(\mathbf{w})} - \underbrace{\alpha \mathbf{1}^\top \log(\mathbf{S}\mathbf{w})}_{:=-g(\mathbf{S}\mathbf{w})} \Big\}$$

  - $\Rightarrow$ $\mathbf{S} \in \{0,1\}^{N \times N(N-1)/2}$ maps edge weights to nodal degrees, i.e., $\mathbf{d} = \mathbf{S}\mathbf{w}$
- Non-differentiable $f(\mathbf{w})$ is strongly convex, $g(\mathbf{d})$ is strictly convex $\Rightarrow$ Unique solution $\mathbf{w}^*$
  - $\Rightarrow$ Amenable to dual-based proximal gradient (DPG) solver

Dual problem and its properties

- The dual problem is $\min_{\boldsymbol{\lambda}} \{F(\boldsymbol{\lambda}) + G(\boldsymbol{\lambda})\}$, where

$$F(\boldsymbol{\lambda}) := \max_{\mathbf{w}} \left\{ \langle \mathbf{S}^\top \boldsymbol{\lambda}, \mathbf{w} \rangle - f(\mathbf{w}) \right\}, \quad G(\boldsymbol{\lambda}) := \max_{\mathbf{d}} \{ \langle -\boldsymbol{\lambda}, \mathbf{d} \rangle - g(\mathbf{d}) \}$$

- Strong convexity of f implies a Lipschitz gradient property for F

$\Rightarrow$ **Theorem** (Smoothness of F). $\nabla F(\boldsymbol{\lambda})$ is Lipschitz continuous with constant $L := \frac{N-1}{\beta}$

- Can be solved by the Proximal Gradient (PG) method $\Rightarrow$ Dual-PG (DPG) iterations

$$\boldsymbol{\lambda}_k = \mathbf{prox}_{L^{-1}G} \left(\boldsymbol{\lambda}_{k-1} - \frac{1}{L} \nabla F(\boldsymbol{\lambda}_{k-1}) \right)$$

$\Rightarrow$ **Theorem** (Convergence rate). Primal sequence $\hat{\mathbf{w}}_k = \text{argmax}_{\mathbf{w}} \left\{ \langle \mathbf{S}^\top \boldsymbol{\lambda}_k, \mathbf{w} \rangle - f(\mathbf{w}) \right\}$ satisfies

$$\|\hat{\mathbf{w}}_k - \mathbf{w}^\star\|_2 \leq \frac{\sqrt{2(N-1)}\|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^\star\|_2}{\beta k}.$$

S. S. Saboksayr and G. Mateos, "Accelerated graph learning from smooth signals," *IEEE Signal Process. Letters*, 2021.

Dual problem and its properties

- The dual problem is $\min_{\boldsymbol{\lambda}} \{F(\boldsymbol{\lambda}) + G(\boldsymbol{\lambda})\}$, where

$$F(\boldsymbol{\lambda}) := \max_{\mathbf{w}} \left\{ \langle \mathbf{S}^\top \boldsymbol{\lambda}, \mathbf{w} \rangle - f(\mathbf{w}) \right\}, \quad G(\boldsymbol{\lambda}) := \max_{\mathbf{d}} \{ \langle -\boldsymbol{\lambda}, \mathbf{d} \rangle - g(\mathbf{d}) \}$$

- Strong convexity of f implies a Lipschitz gradient property for F

  $\Rightarrow$ **Theorem** (Smoothness of F). $\nabla F(\boldsymbol{\lambda})$ is Lipschitz continuous with constant $L := \frac{N-1}{\beta}$

- Can be solved by the Proximal Gradient (PG) method $\Rightarrow$ Dual-PG (DPG) iterations

$$\boldsymbol{\lambda}_k = \mathbf{prox}_{L^{-1}G} \left(\boldsymbol{\lambda}_{k-1} - \frac{1}{L} \nabla F(\boldsymbol{\lambda}_{k-1}) \right)$$

  $\Rightarrow$ **Theorem** (Convergence rate). Primal sequence $\hat{\mathbf{w}}_k = \operatorname{argmax}_{\mathbf{w}} \left\{ \langle \mathbf{S}^\top \boldsymbol{\lambda}_k, \mathbf{w} \rangle - f(\mathbf{w}) \right\}$ satisfies

$$\|\hat{\mathbf{w}}_k - \mathbf{w}^\star\|_2 \leq \frac{\sqrt{2(N-1)} \|\boldsymbol{\lambda}_0 - \boldsymbol{\lambda}^\star\|_2}{\beta k}.$$

S. S. Saboksayr and G. Mateos, "Accelerated graph learning from smooth signals," *IEEE Signal Process. Letters*, 2021.

- **Q:** Online estimation from streaming data $\{\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots\}$?

  $\Rightarrow$ At time t solve the time-varying composite optimization

$$\mathbf{w}_t^\star \in \underset{\mathbf{w}}{\operatorname{argmin}} \Big\{ \overbrace{\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} + 2\mathbf{w}^\top \mathbf{e}_{1:t} + \beta\|\mathbf{w}\|_2^2}^{:= f_t(\mathbf{w})} \underbrace{-\alpha \mathbf{1}^\top \log(\mathbf{S}\mathbf{w}) \Big\}}_{:= -g(\mathbf{S}\mathbf{w})}$$

- **Step 1:** Recursively update the Euclidean-distance vector via exponential moving average

$$\mathbf{e}_{1:t} = (1-\gamma)\mathbf{e}_{1:t-1} + \gamma \mathbf{e}_t$$

- **Step 2:** Run a single iteration of the batch DPG algorithm $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} - L^{-1}(\mathbf{S}\mathbf{v}_t - \mathbf{u}_t)$, where

$$\mathbf{v}_t = \max\left(\mathbf{0}, \frac{\mathbf{S}^\top \boldsymbol{\lambda}_{t-1} - 2\mathbf{e}_{1:t}}{2\beta}\right) \text{ and } \mathbf{u}_t = \frac{\mathbf{S}\mathbf{v}_t - L\boldsymbol{\lambda}_{t-1} + \sqrt{(\mathbf{S}\mathbf{v}_t - L\boldsymbol{\lambda}_{t-1})^2 + 4\alpha L \mathbf{1}}}{2}$$

  $\Rightarrow$ Memory footprint and per time interval complexity of $\mathcal{O}(N^2)$ do not grow with t

- **Q:** Online estimation from streaming data $\{\mathbf{x}_1, \ldots, \mathbf{x}_t, \mathbf{x}_{t+1}, \ldots\}$?

  $\Rightarrow$ At time t solve the time-varying composite optimization

$$\mathbf{w}_t^\star \in \underset{\mathbf{w}}{\operatorname{argmin}} \Big\{ \overbrace{\mathbb{I}\{\mathbf{w} \succeq \mathbf{0}\} + 2\mathbf{w}^\top \mathbf{e}_{1:t} + \beta\|\mathbf{w}\|_2^2}^{:=f_t(\mathbf{w})} \underbrace{-\alpha \mathbf{1}^\top \log(\mathbf{S}\mathbf{w})\Big\}}_{:=-g(\mathbf{S}\mathbf{w})}$$

- **Step 1:** Recursively update the Euclidean-distance vector via exponential moving average

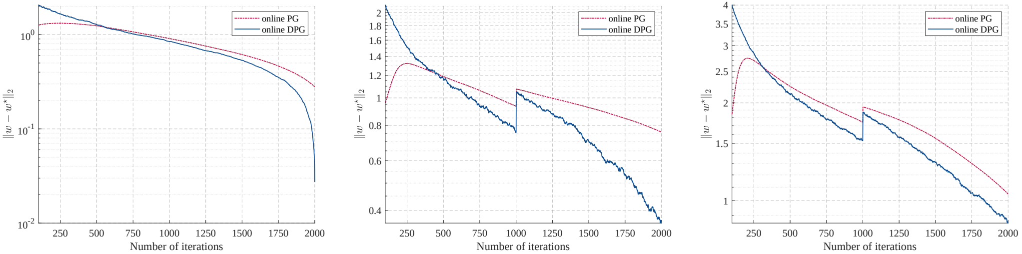
$$\mathbf{e}_{1:t} = (1-\gamma)\mathbf{e}_{1:t-1} + \gamma \mathbf{e}_t$$

- **Step 2:** Run a single iteration of the batch DPG algorithm $\boldsymbol{\lambda}_t = \boldsymbol{\lambda}_{t-1} - L^{-1}(\mathbf{S}\mathbf{v}_t - \mathbf{u}_t)$, where

$$\mathbf{v}_t = \max\left(\mathbf{0}, \frac{\mathbf{S}^\top \boldsymbol{\lambda}_{t-1} - 2\mathbf{e}_{1:t}}{2\beta}\right) \text{ and } \mathbf{u}_t = \frac{\mathbf{S}\mathbf{v}_t - L\boldsymbol{\lambda}_{t-1} + \sqrt{(\mathbf{S}\mathbf{v}_t - L\boldsymbol{\lambda}_{t-1})^2 + 4\alpha L \mathbf{1}}}{2}$$

  $\Rightarrow$ Memory footprint and per time interval complexity of $\mathcal{O}(N^2)$ do not grow with t

- Convergence behavior
  - ⇒ Graphs: (a) Stationary 100-node ER; (b) dynamic 50–node ER; (c) dynamic 100-node SBM
  - ⇒ Signals: $T = 2000$ i.i.d. Gaussian-distributed smooth signals $\mathbf{x}_t \sim \mathcal{N}\left(\mathbf{0}, \mathbf{L}_t^\dagger + 10^{-}2\mathbf{I}_N\right)$
  - ⇒ Monitor the evolution of the error metric $\|\hat{\mathbf{w}}_t - \mathbf{w}_t^\star\|_2$



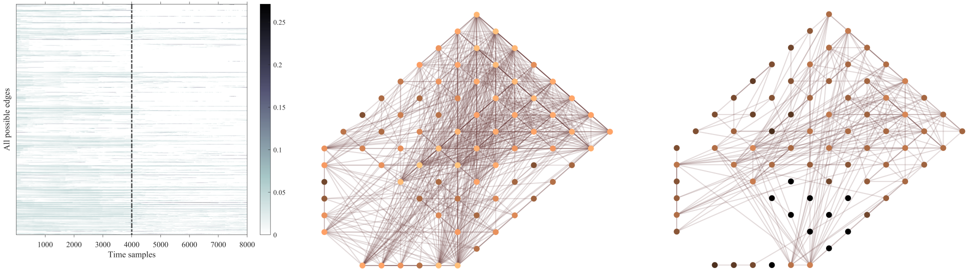
(a) (b) (c)

S. S. Saboksayr and G. Mateos, "Online graph learning under smoothness priors," *in Proc. EUSIPCO*, 2021.

- ECoG data acquired via $N = 76$ electrodes [Kramer et al'08]
  - $\Rightarrow$ 8×8 grid located at the cortical brain's surface
  - $\Rightarrow$ 12 electrodes are placed deeper

(a) Evolution of edge weights from pre-ictal to ictal; vertical line indicates seizure onset

(b) Recovered brain graphs 2.5s prior to seizure; and (c) 2.5s after $\Rightarrow$ Edge-thinning apparent



(a) (b) (c)

Closing remarks

UNIVERSITY of ROCHESTER

- Network topology inference cornerstone problem in Network Science
  - $\Rightarrow$ Most GSP works analyze how $\mathcal{G}$ affect signals and filters
  - $\Rightarrow$ In many cases the underlying $\mathcal{G}$ is not readily available
- Novel online algorithm to learn graphs from observations of streaming smooth signals
  - $\Rightarrow$ Cardinal property of many real-world graph signals
  - $\Rightarrow$ Show problem has favorable structure in the dual domain
- Online dual-based proximal gradient method
  - $\Rightarrow$ Tracks (possibly) dynamic network topology with affordable memory and complexity
  - $\Rightarrow$ Faster empirical convergence than state-of-the-art algorithms
- Try it out!: `http://hajim.rochester.edu/ece/sites/gmateos/code/ODPG.zip`