Online Proximal ADMM for Graph Learning from Streaming Smooth Signals

Hector Chahuara

Department of Electrical Engineering
Pontificia Universidad Católica del Perú, Lima, Peru

Co-author: Gonzalo Mateos

Department of Electrical and Computer Engineering
University of Rochester, NY, USA

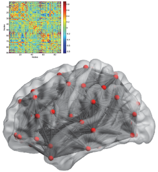
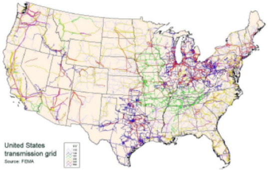
Acknowledgments: IEEE SPS ME-UYR Program

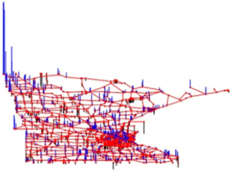
Hyderabad, India, April 2025

Contents

Overview

Overview

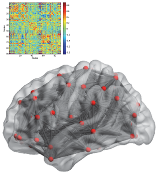
 

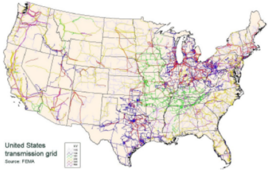


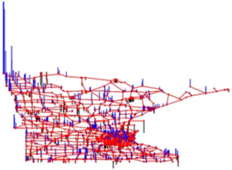
- **Context:** Information from graph-structured data is of interest in applications such as network neuroscience, traffic networks, power grid networks, among others

[1] Xiaolu Wang, Chaorui Yao, and Anthony Man-Cho So. "A Linearly Convergent Optimization Framework for Learning Graphs From Smooth Signals". In: *IEEE Trans. Signal Inf. Process. Netw.* 9 (2023), pp. 490–504.

Overview



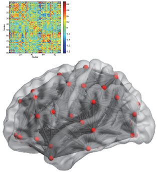


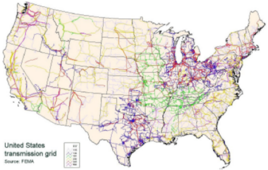


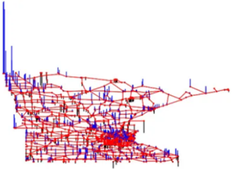
- **Context:** Information from graph-structured data is of interest in applications such as network neuroscience, traffic networks, power grid networks, among others
- **Idea:** Formulate an ADMM-based method for online identification of dynamic networks from streaming signals

[1] Xiaolu Wang, Chaorui Yao, and Anthony Man-Cho So. "A Linearly Convergent Optimization Framework for Learning Graphs From Smooth Signals". In: *IEEE Trans. Signal Inf. Process. Netw.* 9 (2023), pp. 490–504.

Overview







- **Context:** Information from graph-structured data is of interest in applications such as network neuroscience, traffic networks, power grid networks, among others
- **Idea:** Formulate an ADMM-based method for online identification of dynamic networks from streaming signals
- **Motivation:** Proximal ADMM (PADMM) exhibits local linear convergence (fast graph learning) in batch settings[1], do benefits carry over online?

---

[1] Xiaolu Wang, Chaorui Yao, and Anthony Man-Cho So. "A Linearly Convergent Optimization Framework for Learning Graphs From Smooth Signals". In: *IEEE Trans. Signal Inf. Process. Netw.* 9 (2023), pp. 490–504.

Graph learning from smooth signals

Graph learning from smooth signals

Problem statement

Estimate the undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ given the network measurements $\mathbf{X} = \{\mathbf{x}^{(k)}\}$ (smooth on $\mathcal{G}$), with a potentially time-varying weight matrix $\mathbf{W}$.

[2] Vassilis Kalofolias. "How to Learn a Graph from Smooth Signals". In: *Proc. Int. Conf. Artif. Intell. Statist.* 2016, pp. 920–929.

Graph learning from smooth signals

Problem statement

Estimate the undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ given the network measurements $\mathbf{X} = \{\mathbf{x}^{(k)}\}$ (smooth on $\mathcal{G}$), with a potentially time-varying weight matrix $\mathbf{W}$.

- Distance matrix: $\mathbf{Z}_{ij} = \|\mathbf{X}_{i,:} - \mathbf{X}_{j,:}\|_2^2$, $(i,j) \in \mathcal{V}$

[2] Vassilis Kalofolias. "How to Learn a Graph from Smooth Signals". In: *Proc. Int. Conf. Artif. Intell. Statist.* 2016, pp. 920–929.

Graph learning from smooth signals

Problem statement

Estimate the undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ given the network measurements $\mathbf{X} = \{\mathbf{x}^{(k)}\}$ (smooth on $\mathcal{G}$), with a potentially time-varying weight matrix $\mathbf{W}$.

- Distance matrix: $\mathbf{Z}_{ij} = \|\mathbf{X}_{i,:} - \mathbf{X}_{j,:}\|_2^2$, $(i,j) \in \mathcal{V}$
- Total variation of signal $\mathbf{x}$: $\mathrm{TV}(\mathbf{x}) := \mathbf{x}^\top \mathbf{L}\mathbf{x}$ (Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{W}$)

[2] Vassilis Kalofolias. "How to Learn a Graph from Smooth Signals". In: *Proc. Int. Conf. Artif. Intell. Statist.* 2016, pp. 920–929.

Graph learning from smooth signals

Problem statement

Estimate the undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ given the network measurements $\mathbf{X} = \{\mathbf{x}^{(k)}\}$ (smooth on $\mathcal{G}$), with a potentially time-varying weight matrix $\mathbf{W}$.

- Distance matrix: $\mathbf{Z}_{ij} = \|\mathbf{X}_{i,:} - \mathbf{X}_{j,:}\|_2^2$, $(i,j) \in \mathcal{V}$
- Total variation of signal $\mathbf{x}$: $\mathrm{TV}(\mathbf{x}) := \mathbf{x}^\top \mathbf{L}\mathbf{x}$ (Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{W}$)
- Sparsity and smoothness are linked: $\sum_{t=1}^{T} \mathrm{TV}\left(\mathbf{x}^{(t)}\right) = \mathrm{tr}\left(\mathbf{X}^\top \mathbf{L}\mathbf{X}\right) = \frac{1}{2}\|\mathbf{W} \odot \mathbf{Z}\|_1$

[2] Vassilis Kalofolias. "How to Learn a Graph from Smooth Signals". In: *Proc. Int. Conf. Artif. Intell. Statist.* 2016, pp. 920–929.

Graph learning from smooth signals

Problem statement

Estimate the undirected graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ given the network measurements $\mathbf{X} = \{\mathbf{x}^{(k)}\}$ (smooth on $\mathcal{G}$), with a potentially time-varying weight matrix $\mathbf{W}$.

- Distance matrix: $\mathbf{Z}_{ij} = \|\mathbf{X}_{i,:} - \mathbf{X}_{j,:}\|_2^2$, $(i,j) \in \mathcal{V}$
- Total variation of signal $\mathbf{x}$: $\mathrm{TV}(\mathbf{x}) := \mathbf{x}^\top \mathbf{L}\mathbf{x}$ (Laplacian: $\mathbf{L} = \mathbf{D} - \mathbf{W}$)
- Sparsity and smoothness are linked: $\sum_{t=1}^{T} \mathrm{TV}\left(\mathbf{x}^{(t)}\right) = \mathrm{tr}\left(\mathbf{X}^\top \mathbf{L}\mathbf{X}\right) = \frac{1}{2}\|\mathbf{W} \odot \mathbf{Z}\|_1$

Graph learning problem[2]

$$\min_{\mathbf{W} \in \mathbb{R}^{n \times n}} \quad \|\mathbf{Z} \odot \mathbf{W}\|_{1,1} - \alpha \mathbf{1}_n^\top \log\left(\mathbf{W}\mathbf{1}_n\right) + \frac{\beta}{2}\|\mathbf{W}\|_F^2$$

$$\text{subject to } \operatorname{diag}(\mathbf{W}) = \mathbf{0}_n, \mathbf{W}_{ij} = \mathbf{W}_{ji} \geq 0, i \neq j$$

[2] Vassilis Kalofolias. “How to Learn a Graph from Smooth Signals”. In: *Proc. Int. Conf. Artif. Intell. Statist.* 2016, pp. 920–929.

Batch graph learning via PADMM

Reformulated problem

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^r,\mathbf{v}\in\mathbb{R}^n} \quad & f(\mathbf{w}) + g(\mathbf{v}) \\ \text{subject to} \quad & \mathbf{S}\mathbf{w} - \mathbf{v} = \mathbf{0} \end{aligned}$$

- $\mathbf{w} = \text{vec}\left[\text{triu}(\mathbf{W})\right] \in \mathbb{R}^r$, $r = \frac{n(n-1)}{2}$
- Variable-splitting constraint: $\mathbf{S}\mathbf{w} - \mathbf{v}$
- $\mathbf{v}$: vector of nodal degrees
- $f(\mathbf{w}) = 2\mathbf{z}^\top \mathbf{w} + \beta \|\mathbf{w}\|_2^2 + \iota_{\mathbf{w}\geq \mathbf{0}}$
- $g(\mathbf{v}) = -\alpha \mathbf{1}^\top \mathbf{log}(\mathbf{v})$

- $\mathcal{L}_\rho(\mathbf{w}, \mathbf{v}, \boldsymbol{\lambda}) = f(\mathbf{w}) + g(\mathbf{v}) + \boldsymbol{\lambda}^\top (\mathbf{S}\mathbf{w} - \mathbf{v}) + \frac{\rho}{2} \|\mathbf{S}\mathbf{w} - \mathbf{v}\|_2^2$

Batch graph learning via PADMM

Reformulated problem

$$\min_{\mathbf{w}\in\mathbb{R}^r,\mathbf{v}\in\mathbb{R}^n} \quad f(\mathbf{w}) + g(\mathbf{v})$$
$$\text{subject to} \quad \mathbf{S}\mathbf{w} - \mathbf{v} = \mathbf{0}$$

- $\mathbf{w} = \text{vec}\left[\text{triu}(\mathbf{W})\right] \in \mathbb{R}^r$, $r = \frac{n(n-1)}{2}$
- Variable-splitting constraint: $\mathbf{S}\mathbf{w} - \mathbf{v}$
- $\mathbf{v}$: vector of nodal degrees
- $f(\mathbf{w}) = 2\mathbf{z}^\top\mathbf{w} + \beta\|\mathbf{w}\|_2^2 + \iota_{\mathbf{w}\geq\mathbf{0}}$
- $g(\mathbf{v}) = -\alpha\mathbf{1}^\top\mathbf{log}(\mathbf{v})$

- $\mathcal{L}_\rho(\mathbf{w},\mathbf{v},\boldsymbol{\lambda}) = f(\mathbf{w}) + g(\mathbf{v}) + \boldsymbol{\lambda}^\top(\mathbf{S}\mathbf{w} - \mathbf{v}) + \frac{\rho}{2}\|\mathbf{S}\mathbf{w} - \mathbf{v}\|_2^2$
- $\tilde{\mathcal{L}}_\rho(\mathbf{w},\mathbf{v},\boldsymbol{\lambda}) = \mathcal{L}_\rho(\mathbf{w},\mathbf{v},\boldsymbol{\lambda}) + \frac{1}{2}\left\|\mathbf{w} - \mathbf{w}^{(k)}\right\|_{\mathbf{G}}^2 + \frac{1}{2}\left\|\mathbf{v} - \mathbf{v}^{(k)}\right\|_{\mathbf{H}}^2$

Batch graph learning via PADMM

Reformulated problem

$$\boxed{\begin{aligned}&\min_{\mathbf{w}\in\mathbb{R}^r,\mathbf{v}\in\mathbb{R}^n} && f(\mathbf{w}) + g(\mathbf{v})\\ &\text{subject to} && \mathbf{S}\mathbf{w} - \mathbf{v} = \mathbf{0}\end{aligned}}$$

- $\mathbf{w} = \text{vec}\left[\text{triu}(\mathbf{W})\right] \in \mathbb{R}^r,\ r = \frac{n(n-1)}{2}$
- Variable-splitting constraint: $\mathbf{S}\mathbf{w} - \mathbf{v}$
- $\mathbf{v}$: vector of nodal degrees
- $f(\mathbf{w}) = 2\mathbf{z}^\top\mathbf{w} + \beta\left\|\mathbf{w}\right\|_2^2 + \iota_{\mathbf{w}\geq\mathbf{0}}$
- $g(\mathbf{v}) = -\alpha\mathbf{1}^\top\mathbf{log}(\mathbf{v})$

- $\mathcal{L}_\rho(\mathbf{w},\mathbf{v},\boldsymbol{\lambda}) = f(\mathbf{w}) + g(\mathbf{v}) + \boldsymbol{\lambda}^\top(\mathbf{S}\mathbf{w} - \mathbf{v}) + \frac{\rho}{2}\left\|\mathbf{S}\mathbf{w} - \mathbf{v}\right\|_2^2$
- $\tilde{\mathcal{L}}_\rho(\mathbf{w},\mathbf{v},\boldsymbol{\lambda}) = \mathcal{L}_\rho(\mathbf{w},\mathbf{v},\boldsymbol{\lambda}) + \frac{1}{2}\left\|\mathbf{w} - \mathbf{w}^{(k)}\right\|_{\mathbf{G}}^2 + \frac{1}{2}\left\|\mathbf{v} - \mathbf{v}^{(k)}\right\|_{\mathbf{H}}^2$
- Proximity terms use the following matrices:

$$\mathbf{G} = \tau_1^{-1}\mathbf{I} - \rho\mathbf{S}^\top\mathbf{S} \qquad \mathbf{H} = \left(\tau_2^{-1} - \rho\right)\mathbf{I}$$

PADMM updates

- Parameters: $0 < \tau_1 < \frac{1}{\rho}\left\|\mathbf{S}\right\|_2$, $0 < \tau_2 < \frac{1}{\rho}$

PADMM iteration

$$\mathbf{w}^{(k+1)} = \underset{\mathbf{w}\in\mathbb{R}^r}{\arg\min}\ \tilde{\mathcal{L}}_\rho\left(\mathbf{w}, \mathbf{v}^{(k)}, \boldsymbol{\lambda}^{(k)}\right)$$

$$\mathbf{v}^{(k+1)} = \underset{\mathbf{v}\in\mathbb{R}^n}{\arg\min}\ \tilde{\mathcal{L}}_\rho\left(\mathbf{w}^{(k+1)}, \mathbf{v}, \boldsymbol{\lambda}^{(k)}\right)$$

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \rho\left(\mathbf{S}\mathbf{w}^{(k+1)} - \mathbf{v}^{(k+1)}\right)$$

- PADMM enjoys local linear convergence[3].

[3] Xiaolu Wang, Chaorui Yao, and Anthony Man-Cho So. "A Linearly Convergent Optimization Framework for Learning Graphs From Smooth Signals". In: *IEEE Trans. Signal Inf. Process. Netw.* 9 (2023), pp. 490–504.

PADMM updates

- Parameters: $0 < \tau_1 < \frac{1}{\rho}\|\mathbf{S}\|_2$, $0 < \tau_2 < \frac{1}{\rho}$

PADMM iteration

$$\mathbf{w}^{(k+1)} = \arg\min_{\mathbf{w}\in\mathbb{R}^r} \tilde{\mathcal{L}}_\rho\left(\mathbf{w}, \mathbf{v}^{(k)}, \boldsymbol{\lambda}^{(k)}\right)$$

$$\mathbf{v}^{(k+1)} = \arg\min_{\mathbf{v}\in\mathbb{R}^n} \tilde{\mathcal{L}}_\rho\left(\mathbf{w}^{(k+1)}, \mathbf{v}, \boldsymbol{\lambda}^{(k)}\right)$$

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \rho\left(\mathbf{S}\mathbf{w}^{(k+1)} - \mathbf{v}^{(k+1)}\right)$$

Primary variable update

$$\overline{\mathbf{w}} = \mathbf{w}^{(k)} - \rho\tau_1\mathbf{S}^\top\left[\mathbf{S}\mathbf{w}^{(k)} - \mathbf{v}^{(k)} + \frac{\boldsymbol{\lambda}^{(k)}}{\rho}\right]$$

$$\mathbf{w}^{(k+1)} = \max\left(\frac{\overline{\mathbf{w}} - 2\tau_1\mathbf{z}}{2\tau_1\beta + 1}, \mathbf{0}\right)$$

- PADMM enjoys local linear convergence[3].

[3] Xiaolu Wang, Chaorui Yao, and Anthony Man-Cho So. "A Linearly Convergent Optimization Framework for Learning Graphs From Smooth Signals". In: *IEEE Trans. Signal Inf. Process. Netw.* 9 (2023), pp. 490–504.

PADMM updates

- Parameters: $0 < \tau_1 < \frac{1}{\rho}\|\mathbf{S}\|_2$, $0 < \tau_2 < \frac{1}{\rho}$

PADMM iteration

$$\mathbf{w}^{(k+1)} = \underset{\mathbf{w}\in\mathbb{R}^r}{\arg\min}\ \tilde{\mathcal{L}}_\rho\left(\mathbf{w}, \mathbf{v}^{(k)}, \boldsymbol{\lambda}^{(k)}\right)$$

$$\mathbf{v}^{(k+1)} = \underset{\mathbf{v}\in\mathbb{R}^n}{\arg\min}\ \tilde{\mathcal{L}}_\rho\left(\mathbf{w}^{(k+1)}, \mathbf{v}, \boldsymbol{\lambda}^{(k)}\right)$$

$$\boldsymbol{\lambda}^{(k+1)} = \boldsymbol{\lambda}^{(k)} + \rho\left(\mathbf{S}\mathbf{w}^{(k+1)} - \mathbf{v}^{(k+1)}\right)$$

Auxiliary variable update

$$\bar{\mathbf{v}} = \mathbf{v}^{(k)} - \rho\tau_2\left[\mathbf{S}\mathbf{w}^{(k+1)} - \mathbf{v}^{(k)} - \frac{\boldsymbol{\lambda}^{(k)}}{\rho}\right]$$

$$\mathbf{v}^{(k+1)} = \frac{\bar{\mathbf{v}} + \sqrt{\bar{\mathbf{v}}^2 + 4\tau_2\alpha\mathbf{1}}}{2}$$

- PADMM enjoys local linear convergence[3].

[3] Xiaolu Wang, Chaorui Yao, and Anthony Man-Cho So. "A Linearly Convergent Optimization Framework for Learning Graphs From Smooth Signals". In: *IEEE Trans. Signal Inf. Process. Netw.* 9 (2023), pp. 490–504.

Proposed method

OPADMM: Algorithm outline

Input: $\mathbf{S}$; $\overline{\mathbf{z}}^{(k)}$; reg. hyp. α and β; PADMM hyp. ρ, τ_1 and τ_2; $\mathbf{w}^{(0)}$, $\mathbf{v}^{(0)}$ and $\boldsymbol{\lambda}^{(0)}$
Output: Tracking solution available $\mathbf{w}^{(k)}$
for $k = 1, 2, \dots$ **do**

Update $\gamma^{(k)}$

$\mathbf{z}_{1:k} \leftarrow \left(1 - \gamma^{(k)}\right) \mathbf{z}_{1:k-1} + \gamma^{(k)} \overline{\mathbf{z}}^{(k)}$

$\overline{\mathbf{w}} \leftarrow \mathbf{w}^{(k-1)} - \tau_1 \rho \mathbf{S}^\top \left(\mathbf{S}\mathbf{w}^{(k-1)} - \mathbf{v}^{(k-1)} + \frac{\boldsymbol{\lambda}^{(k-1)}}{\rho} \right)$

$\mathbf{w}^{(k)} \leftarrow \frac{1}{2\tau_1\beta+1} \max\left(\overline{\mathbf{w}} - 2\tau_1 \mathbf{z}_{1:k}, \mathbf{0}_r\right)$

$\overline{\mathbf{v}} \leftarrow \left(1 + \rho\tau_2\right) \mathbf{v}^{(k-1)} - \rho\tau_2 \mathbf{S}\mathbf{w}^{(k)} + \tau_2 \boldsymbol{\lambda}^{(k-1)}$

$\mathbf{v}^{(k)} \leftarrow \frac{1}{2} \left(\overline{\mathbf{v}} + \sqrt{\overline{\mathbf{v}}^2 + 4\tau_2\alpha \mathbf{1}_n} \right)$

$\boldsymbol{\lambda}^{(k)} \leftarrow \boldsymbol{\lambda}^{(k-1)} + \rho \left(\mathbf{S}\mathbf{w}^{(k)} - \mathbf{v}^{(k)} \right)$

end

OPADMM: Algorithm outline

Input: $\mathbf{S}$; $\overline{\mathbf{z}}^{(k)}$; reg. hyp. α and β; PADMM hyp. ρ, τ_1 and τ_2; $\mathbf{w}^{(0)}$, $\mathbf{v}^{(0)}$ and $\boldsymbol{\lambda}^{(0)}$
Output: Tracking solution available $\mathbf{w}^{(k)}$
for $k = 1, 2, \ldots$ **do**

Update $\gamma^{(k)}$

$$\mathbf{z}_{1:k} \leftarrow \left(1 - \gamma^{(k)}\right) \mathbf{z}_{1:k-1} + \gamma^{(k)} \overline{\mathbf{z}}^{(k)}$$

$$\overline{\mathbf{w}} \leftarrow \mathbf{w}^{(k-1)} - \tau_1 \rho \mathbf{S}^\top \left(\mathbf{S}\mathbf{w}^{(k-1)} - \mathbf{v}^{(k-1)} + \frac{\boldsymbol{\lambda}^{(k-1)}}{\rho} \right)$$

$$\mathbf{w}^{(k)} \leftarrow \frac{1}{2\tau_1\beta+1} \max\left(\overline{\mathbf{w}} - 2\tau_1 \mathbf{z}_{1:k}, \mathbf{0}_r\right)$$

$$\overline{\mathbf{v}} \leftarrow (1 + \rho\tau_2)\, \mathbf{v}^{(k-1)} - \rho\tau_2 \mathbf{S}\mathbf{w}^{(k)} + \tau_2 \boldsymbol{\lambda}^{(k-1)}$$

$$\mathbf{v}^{(k)} \leftarrow \frac{1}{2} \left(\overline{\mathbf{v}} + \sqrt{\overline{\mathbf{v}}^2 + 4\tau_2 \alpha \mathbf{1}_n} \right)$$

$$\boldsymbol{\lambda}^{(k)} \leftarrow \boldsymbol{\lambda}^{(k-1)} + \rho \left(\mathbf{S}\mathbf{w}^{(k)} - \mathbf{v}^{(k)} \right)$$

end

Standard PADMM iteration
Only one iteration is run per time instant k

OPADMM: Algorithm outline

Input: $\mathbf{S}$; $\bar{\mathbf{z}}^{(k)}$; reg. hyp. α and β; PADMM hyp. ρ, τ_1 and τ_2; $\mathbf{w}^{(0)}$, $\mathbf{v}^{(0)}$ and $\boldsymbol{\lambda}^{(0)}$
Output: Tracking solution available $\mathbf{w}^{(k)}$
for $k = 1, 2, \ldots$ **do**

Update $\gamma^{(k)}$

$\mathbf{z}_{1:k} \leftarrow \left(1 - \gamma^{(k)}\right) \mathbf{z}_{1:k-1} + \gamma^{(k)} \bar{\mathbf{z}}^{(k)}$

$\overline{\mathbf{w}} \leftarrow \mathbf{w}^{(k-1)} - \tau_1 \rho \mathbf{S}^\top \left(\mathbf{S}\mathbf{w}^{(k-1)} - \mathbf{v}^{(k-1)} + \frac{\boldsymbol{\lambda}^{(k-1)}}{\rho} \right)$

$\mathbf{w}^{(k)} \leftarrow \frac{1}{2\tau_1\beta+1} \max\left(\overline{\mathbf{w}} - 2\tau_1 \mathbf{z}_{1:k}, \mathbf{0}_r\right)$

$\bar{\mathbf{v}} \leftarrow \left(1 + \rho\tau_2\right) \mathbf{v}^{(k-1)} - \rho\tau_2 \mathbf{S}\mathbf{w}^{(k)} + \tau_2 \boldsymbol{\lambda}^{(k-1)}$

$\mathbf{v}^{(k)} \leftarrow \frac{1}{2} \left(\bar{\mathbf{v}} + \sqrt{\bar{\mathbf{v}}^2 + 4\tau_2\alpha \mathbf{1}_n} \right)$

$\boldsymbol{\lambda}^{(k)} \leftarrow \boldsymbol{\lambda}^{(k-1)} + \rho \left(\mathbf{S}\mathbf{w}^{(k)} - \mathbf{v}^{(k)} \right)$

end

Online Adaptation

$$\gamma^{(k)} = \begin{cases} \frac{1}{k} & \text{stationary graphs} \\ 2 \times 10^{-3} & \text{time-varying graphs} \end{cases}$$

OPADMM: Analysis

- **Tracking**: Proximity terms on $\mathbf{w}$ and $\mathbf{v}$ apply temporal-variation regularization

[4] Huahua Wang and Arindam Banerjee. "Online Alternating Direction Method". In: *Proc. Int. Conf. Mach. Learn.* Edinburgh, Scotland, 2012, 1699–1706.

OPADMM: Analysis

- **Tracking**: Proximity terms on $\mathbf{w}$ and $\mathbf{v}$ apply temporal-variation regularization
  $\Rightarrow$ This allows OPADMM to yield enhanced tracking capabilities

[4] Huahua Wang and Arindam Banerjee. "Online Alternating Direction Method". In: *Proc. Int. Conf. Mach. Learn.* Edinburgh, Scotland, 2012, 1699–1706.

OPADMM: Analysis

- **Tracking**: Proximity terms on **w** and **v** apply temporal-variation regularization
  $\Rightarrow$ This allows OPADMM to yield enhanced tracking capabilities
- **Computational cost per iteration:** $\mathcal{O}(r)$, $r = \frac{n(n-1)}{2}$

[4] Huahua Wang and Arindam Banerjee. "Online Alternating Direction Method". In: *Proc. Int. Conf. Mach. Learn.* Edinburgh, Scotland, 2012, 1699–1706.

OPADMM: Analysis

- **Tracking**: Proximity terms on $\mathbf{w}$ and $\mathbf{v}$ apply temporal-variation regularization
  $\Rightarrow$ This allows OPADMM to yield enhanced tracking capabilities
- **Computational cost per iteration:** $\mathcal{O}(r)$, $r = \frac{n(n-1)}{2}$
- **Online efficiency:** Memory storage and computation cost do not increase

[4] Huahua Wang and Arindam Banerjee. "Online Alternating Direction Method". In: *Proc. Int. Conf. Mach. Learn.* Edinburgh, Scotland, 2012, 1699–1706.

OPADMM: Analysis

- **Tracking**: Proximity terms on $\mathbf{w}$ and $\mathbf{v}$ apply temporal-variation regularization
  $\Rightarrow$ This allows OPADMM to yield enhanced tracking capabilities
- **Computational cost per iteration:** $\mathcal{O}(r)$, $r = \frac{n(n-1)}{2}$
- **Online efficiency:** Memory storage and computation cost do not increase
- **Convergence guarantees:** OPADMM achieves a sublinear static regret[4]

[4] Huahua Wang and Arindam Banerjee. "Online Alternating Direction Method". In: *Proc. Int. Conf. Mach. Learn.* Edinburgh, Scotland, 2012, 1699–1706.

Experiments and results

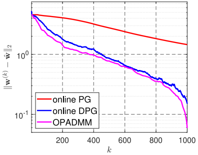
Experiments settings

- **Tech. details:** MATLAB R2023b, Intel i7-7700HQ CPU @ 2.8 GHz, 8 GB RAM
- **Reg. hyperparameters:** α and β chosen by grid search on batch graph learning
- **OPADMM hyperparameters:** ρ, τ_1 and τ_2 chosen by grid search
- **Suboptimality (tracking error):** $\left\|\mathbf{w}^{(k)} - \hat{\mathbf{w}}\right\|_2$
- **Methods:** Online PG[5], online DPG[6], OPADMM (proposed)
- **Data:** Computer-simulated graphs and real world data

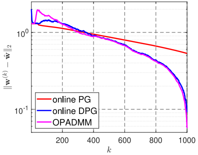
[5] Seyed Saman Saboksayr, Gonzalo Mateos, and Mujdat Cetin. "Online Graph Learning under Smoothness Priors". In: *Proc. of European Signal Process. Conf.* 2021, pp. 1820–1824.

[6] Seyed Saman Saboksayr and Gonzalo Mateos. "Dual-Based Online Learning of Dynamic Network Topologies". In: *Proc. Int. Conf. Acoustics, Speech, Signal Process.* 2023, pp. 1–5.

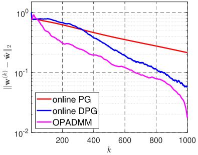
Results: Computer-simulated stationary graphs



(a) Gaussian



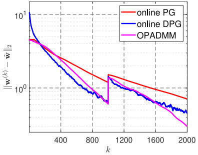
(b) ER



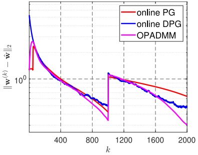
(c) PA

- 1000 signals corrupted with Gaussian noise ($\mu = 0$, $\sigma^2 = 0.01$), 100 nodes
- We used three random models:
  - $\Rightarrow$ Gaussian: threshold 0.8, scale 0.2
  - $\Rightarrow$ Erdös-Rényi (ER): edge probability 0.1
  - $\Rightarrow$ P. attachment (PA): 2 initial nodes
- OPADMM outperforms both DPG and DPG in convergence speed

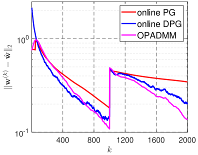
Results: Computer-simulated dynamic graphs



(a) Gaussian



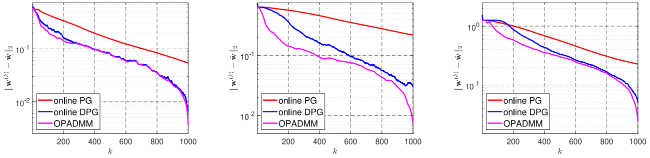
(b) ER



(c) PA

- 2000 signals corrupted with Gaussian noise ($\mu = 0$, $\sigma^2 = 0.01$), 100 nodes
- Piecewise-stationary graphs (10% of edges resampled after 1000 samples)
- Dynamic graphs use same models and parameters as stationary graphs
- OPADMM adapts better to abrupt topology changes

Results: Real-world data



(a) mesh1e1 (b) bcspwr03 (c) lshp_265

- Datasets from the SuiteSparse Matrix Collection[7]:
  - ⇒ **Structural engineering data**: mesh1e1 (48 nodes)
  - ⇒ **Power network data**: bcspwr03 (118 nodes)
  - ⇒ **Thermal network data**: lshp_265 (265 nodes)
- 1000 synthetic smooth signals corrupted with Gaussian noise ($\mu = 0$, $\sigma^2 = 0.01$)
- OPADMM yields a faster convergence than online PG and online DPG

[7] Timothy A. Davis and Yifan Hu. "The University of Florida Sparse Matrix Collection". In: *ACM Trans. Math. Softw.* 38.1 (Dec. 2011).

Conclusions

Concluding remarks

- We propose OPADMM, an efficient online method for graph learning

Concluding remarks

- We propose OPADMM, an efficient online method for graph learning
- It excels at tracking due to temporal-variation regularization in topology updates

Concluding remarks

- We propose OPADMM, an efficient online method for graph learning
- It excels at tracking due to temporal-variation regularization in topology updates
- OPADMM exhibits sublinear static regret under simplifying assumptions

Concluding remarks

- We propose OPADMM, an efficient online method for graph learning
- It excels at tracking due to temporal-variation regularization in topology updates
- OPADMM exhibits sublinear static regret under simplifying assumptions
- It outperforms state-of-the-art online algorithms in synthetic tests

Concluding remarks

- We propose OPADMM, an efficient online method for graph learning
- It excels at tracking due to temporal-variation regularization in topology updates
- OPADMM exhibits sublinear static regret under simplifying assumptions
- It outperforms state-of-the-art online algorithms in synthetic tests
- OPADMM is effective in both stationary and dynamic settings

Concluding remarks

- We propose OPADMM, an efficient online method for graph learning
- It excels at tracking due to temporal-variation regularization in topology updates
- OPADMM exhibits sublinear static regret under simplifying assumptions
- It outperforms state-of-the-art online algorithms in synthetic tests
- OPADMM is effective in both stationary and dynamic settings

  $\Rightarrow$ Robust performance on real-world datasets

Thank you!

References

[1] Xiaolu Wang, Chaorui Yao, and Anthony Man-Cho So. "A Linearly Convergent Optimization Framework for Learning Graphs From Smooth Signals". In: *IEEE Trans. Signal Inf. Process. Netw.* 9 (2023), pp. 490–504.

[2] Vassilis Kalofolias. "How to Learn a Graph from Smooth Signals". In: *Proc. Int. Conf. Artif. Intell. Statist.* 2016, pp. 920–929.

[3] Huahua Wang and Arindam Banerjee. "Online Alternating Direction Method". In: *Proc. Int. Conf. Mach. Learn.* Edinburgh, Scotland, 2012, 1699–1706.

[4] Seyed Saman Saboksayr, Gonzalo Mateos, and Mujdat Cetin. "Online Graph Learning under Smoothness Priors". In: *Proc. of European Signal Process. Conf.* 2021, pp. 1820–1824.

[5] Seyed Saman Saboksayr and Gonzalo Mateos. "Dual-Based Online Learning of Dynamic Network Topologies". In: *Proc. Int. Conf. Acoustics, Speech, Signal Process.* 2023, pp. 1–5.

[6] Timothy A. Davis and Yifan Hu. "The University of Florida Sparse Matrix Collection". In: *ACM Trans. Math. Softw.* 38.1 (Dec. 2011).