# Online Learning of Time-Varying Signals and Graphs

Stefania Sardellitti, Sergio Barbarossa, Paolo Di Lorenzo

Sapienza University of Rome, Italy

Ack's: 2020 EU/Taiwan Project 5G CONNI, Nr. AMD-861459-3
MIUR under the PRIN Liquid-Edge contract

IEEE ICASSP 2021, Toronto, June 6-11, 2021

# Outline

- Motivating remarks
- Online learning of time-varying graphs and signals from noisy observations over a graph
- Design of an efficient learning strategy hinging on small perturbation analysis of Laplacian eigenvectors
- Numerical results
- Conclusions

# Motivating remarks

Motivations

- Graph-based representation: Data can be associated with the vertices of a graph to capture pairwise relations encoded by the presence of links
- In many applications (brain networks, social and communication networks) some links can be altered and the graph topology associated with data may evolve over time

Problem: learning time-varying graphs from the observed noisy signals

Assumptions: Each graph alteration involves a few edges at a time

# Motivations

- State of art:
  - V. Kalofolias et al. (2017), G.B. Giannakis et al. (2018), J. Lee (2020), K. Yamada et al. (2019), J. Mei et al. (2017), Y. Lin et al. (2019), E. Ceci and S. Barbarossa (2020), P. Di Lorenzo et al. (2018)...

# Motivations

- State of art:
  - V. Kalofolias et al. (2017), G.B. Giannakis et al. (2018), J. Lee (2020), K. Yamada et al. (2019), J. Mei et al. (2017), Y. Lin et al. (2019), E. Ceci and S. Barbarossa (2020), P. Di Lorenzo et al. (2018)...

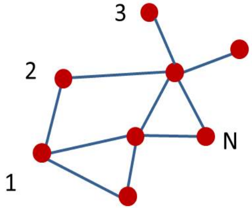
- Our novel contribution

  Joint online learning of time-varying graphs *and* signals from noisy observations of smooth graph signals hinging on small perturbation analysis of the Laplacian eigenvectors

# Introduction to GSP

- Graph signal processing (GSP): provides tools for the processing of signals defined over the vertices of a graph

- A graph $\mathcal{G}$ is represented by a set $\mathcal{V}$ of $N$ vertices and a set of edge $\mathcal{E}$



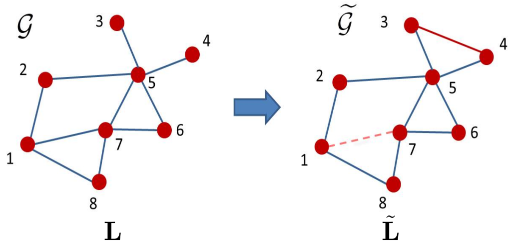

- Algebraic representation for undirected graph:
  - Adjacency matrix $\mathbf{A} \in \mathbb{R}^{N \times N}$ with entries $a_{ij} = 1$ if there is a link between nodes $i$ and $j$ and $a_{ij} = 0$ otherwise
  - Laplacian matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$ with $\mathbf{D} \in \mathbb{R}^{N \times N}$ the diagonal degree matrix
  - $\mathbf{L} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$ where $\mathbf{U}$ collects all the eigenvectors $\{\boldsymbol{u}_i\}_{i=1}^N$ and $\boldsymbol{\Lambda}$ is a diagonal matrix containing its eigenvalues $\lambda_i$

# Introduction to GSP

- A signal $\boldsymbol{x}$ on a graph is defined as the mapping $\boldsymbol{x} : \mathcal{V} \rightarrow \mathbb{R}$
- Edges between nodes capture relations between graph signals
- For undirected graph the Graph Fourier Transform (GFT) $\boldsymbol{s}$ of a graph signal $\boldsymbol{x}$ is
$$\boldsymbol{s} = \mathbf{U}^T \boldsymbol{x}$$
- A $\mathcal{K}$-bandlimited graph signal is a signal whose GFT $\boldsymbol{s}$ is $|\mathcal{K}| = K$ sparse, i.e.
$$\boldsymbol{x} = \mathbf{U}_{\mathcal{K}} \boldsymbol{s}_{\mathcal{K}}$$
where $\mathbf{U}_{\mathcal{K}} \in \mathbb{R}^{N \times K}$ collects the columns of $\mathbf{U}$ associated with the subset of indices $\mathcal{K}$

# Small perturbation of graph Laplacian

Given a nominal graph with Laplacian $\mathbf{L}$, assume that a few edges are added or removed



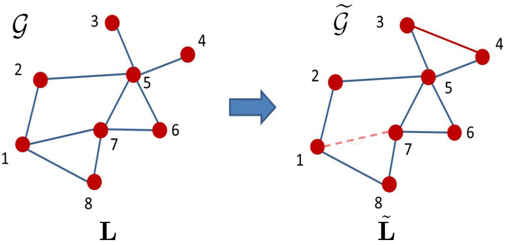

Perturbed Laplacian: $\tilde{\mathbf{L}} = \mathbf{L} + \Delta\mathbf{L} = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^T$

Perturbed eigenvectors and eigenvalues:

$$\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_i + \delta\boldsymbol{u}_i, \quad \tilde{\lambda}_i = \lambda_i + \delta\lambda_i$$

# Small perturbation of graph Laplacian

Given a nominal graph with Laplacian $\mathbf{L}$, assume that a few edges are added or removed




Perturbed Laplacian: $\tilde{\mathbf{L}} = \mathbf{L} + \Delta\mathbf{L} = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{U}}^T$

Perturbed eigenvectors and eigenvalues:

$$\tilde{\boldsymbol{u}}_i = \boldsymbol{u}_i + \delta\boldsymbol{u}_i, \quad \tilde{\lambda}_i = \lambda_i + \delta\lambda_i$$

What is the impact of the graph perturbation on the eigen-decomposition of the Laplacian $\mathbf{L}$?

# Small perturbation of graph Laplacian

If all $\lambda_i(\mathbf{L})$ are distinct, we may use the first-order analysis developed in [1]

Perturbation of the $m$th link: $\Delta\mathbf{L}^{(m)} = \sigma_m \; \boldsymbol{a}_m \; \boldsymbol{a}_m^T$

- $\sigma_m = 1, -1$ if the edge $m$ is added or removed from graph
- $\boldsymbol{a}_m \in \mathbb{R}^N$: all zeros vector with entries $\pm 1$ associated with the vertices of edge $m$

[1] E. Ceci and S. Barbarossa, "Graph Signal Processing in the Presence of Topology Uncertainties", IEEE Trans. on Signal Process., vol. 68, pp. 1558–1573, Feb. 2020

# Small perturbation of graph Laplacian

If all $\lambda_i(\mathbf{L})$ are distinct, we may use the first-order analysis developed in [1]

Perturbation of the $m$th link: $\Delta\mathbf{L}^{(m)} = \sigma_m \; \boldsymbol{a}_m \; \boldsymbol{a}_m^T$

- $\sigma_m = 1, -1$ if the edge $m$ is added or removed from graph
- $\boldsymbol{a}_m \in \mathbb{R}^N$: all zeros vector with entries $\pm 1$ associated with the vertices of edge $m$

Approximate perturbations:

$$\delta\lambda_i^{(m)} = \boldsymbol{u}_i^T \Delta\mathbf{L}^{(m)} \boldsymbol{u}_i = \sigma_m \boldsymbol{u}_i^T \; \boldsymbol{a}_m \; \boldsymbol{a}_m^T \boldsymbol{u}_i$$

$$\delta\boldsymbol{u}_i^{(m)} = \sigma_m \sum_{j=2, j\neq i}^{N} \frac{\boldsymbol{u}_j^T \; \boldsymbol{a}_m \; \boldsymbol{a}_m^T \boldsymbol{u}_i}{\lambda_i - \lambda_j} \boldsymbol{u}_j = \sigma_m \sum_{j=2, j\neq i}^{N} b_{ji}^{(m)} \boldsymbol{u}_j$$

[1] E. Ceci and S. Barbarossa, "Graph Signal Processing in the Presence of Topology Uncertainties", IEEE Trans. on Signal Process., vol. 68, pp. 1558–1573, Feb. 2020

# Small perturbation of graph Laplacian

If all $\lambda_i(\mathbf{L})$ are distinct, we may use the first-order analysis developed in [1]

Perturbation of the $m$th link: $\mathbf{\Delta L}^{(m)} = \sigma_m \; \boldsymbol{a}_m \; \boldsymbol{a}_m^T$

- $\sigma_m = 1, -1$ if the edge $m$ is added or removed from graph
- $\boldsymbol{a}_m \in \mathbb{R}^N$: all zeros vector with entries $\pm 1$ associated with the vertices of edge $m$

Approximate perturbations:

$$\delta\lambda_i^{(m)} = \boldsymbol{u}_i^T \mathbf{\Delta L}^{(m)} \boldsymbol{u}_i = \sigma_m \boldsymbol{u}_i^T \; \boldsymbol{a}_m \; \boldsymbol{a}_m^T \boldsymbol{u}_i$$

$$\delta\boldsymbol{u}_i^{(m)} = \sigma_m \sum_{j=2, j\neq i}^{N} \frac{\boldsymbol{u}_j^T \; \boldsymbol{a}_m \; \boldsymbol{a}_m^T \boldsymbol{u}_i}{\lambda_i - \lambda_j} \boldsymbol{u}_j = \sigma_m \sum_{j=2, j\neq i}^{N} b_{ji}^{(m)} \boldsymbol{u}_j$$

Approximate sum of the perturbations:

$$\delta\lambda_i = \sum_{m\in\mathcal{E}_p} \delta\lambda_i^{(m)}, \qquad \delta\boldsymbol{u}_i = \sum_{m\in\mathcal{E}_p} \delta\boldsymbol{u}_i^{(m)}$$

assuming

$$-g_i^- \ll \sum_{m\in\mathcal{E}_p} \sigma_m \boldsymbol{u}_i^T \; \boldsymbol{a}_m \; \boldsymbol{a}_m^T \boldsymbol{u}_i \ll g_i^+$$

with $g_i^+ := \lambda_{i+1} - \lambda_i$, $g_i^- := \lambda_i - \lambda_{i-1}$ the eigenvalues gaps

[1] E. Ceci and S. Barbarossa, “Graph Signal Processing in the Presence of Topology Uncertainties”, IEEE Trans. on Signal Process., vol. 68, pp. 1558–1573, Feb. 2020

# Online learning of signals and graphs

Our goals:

- Use small perturbation analysis to write a time-varying model in closed form for the perturbed Laplacian
- Track the graph perturbations and the graph signal evolution *jointly* from noisy observations

Proposed method:

Online learning method for time-varying graphs affected by small edges perturbations

# Graph signal model

- $\mathcal{G} := \{\mathcal{V}, \mathcal{E}\}$: the nominal graph with Laplacian $\mathbf{L}$
- $\mathcal{G}[n] := \{\mathcal{V}, \mathcal{E}[n]\}$: the time-varying graph at the time index $n$ with instantaneous Laplacian $\mathbf{L}[n]$
- Noisy observed graph signal at time $n$:

$$\boldsymbol{y}[n] = \boldsymbol{x}[n] + \boldsymbol{v}[n] = \mathbf{U}_{\mathcal{K}}[n]\boldsymbol{s}_{\mathcal{K}}[n] + \boldsymbol{v}[n]$$

  - $\boldsymbol{x}[n] \in \mathbb{R}^N$: $\mathcal{K}$-bandlimited graph signal
  - $\mathbf{U}_{\mathcal{K}}[n]$: matrix collecting the first $K$ eigenvectors of $\mathbf{L}[n]$
  - $\boldsymbol{v}[n] \in \mathbb{R}^N$: Gaussian random vector with $\boldsymbol{v}[n] \sim \mathcal{N}(\mathbf{0}, \sigma_v^2\mathbf{I})$

# Graph signal model

At each time $n$, the instantaneous Laplacian $\tilde{\mathbf{L}}[n]$ is a small perturbation of the Laplacian at the previous time $n-1$

$$\tilde{\mathbf{L}}[n] = \tilde{\mathbf{L}}[n-1] + \Delta\tilde{\mathbf{L}}[n-1]$$

The eigenvectors of $\tilde{\mathbf{L}}[n]$ are expressed as perturbations of the eigenvectors of $\tilde{\mathbf{L}}[n-1]$

$$\tilde{\mathbf{U}}_{\mathcal{K}}[n] = \tilde{\mathbf{U}}_{\mathcal{K}}[n-1] + \Delta\tilde{\mathbf{U}}_{\mathcal{K}}[n-1]$$

Using small perturbation analysis we have

$$\Delta\tilde{\mathbf{U}}_{\mathcal{K}}[n-1] = \sum_{m\in\mathcal{E}} z_m[n]\tilde{\mathbf{U}}[n-1]\mathbf{B}_{m,\mathcal{K}}[n-1]$$

- $z_m[n] \in \{1, -1, 0\}$ is 1 or $-1$ if edge $m$ at time $n$ is, respectively, added or removed and 0 if it remains unaltered
- $\mathbf{B}_{m,\mathcal{K}}[n-1]$ is derived using $\tilde{\mathbf{U}}[n-1]$ and $\tilde{\mathbf{\Lambda}}[n-1]$

# Online learning strategy

Proposed strategy:

Joint online estimation of:

1) the sparse vector $\boldsymbol{z}[n] = [z_1[n]; \ldots ; z_E[n]]$, identifying the perturbed edges over time
2) the GFT coefficients vector $\boldsymbol{s}_{\mathcal{K}}[n]$

to minimize the mean-square error plus a $l_1$-norm penalty

# Online learning strategy

Proposed strategy:

Joint online estimation of:

1) the sparse vector $\boldsymbol{z}[n] = [z_1[n]; \ldots ; z_E[n]]$, identifying the perturbed edges over time
2) the GFT coefficients vector $\boldsymbol{s}_{\mathcal{K}}[n]$

to minimize the mean-square error plus a $l_1$-norm penalty

Average data fitting error: $E[f^{(n)}(\boldsymbol{s}_{\mathcal{K}}, \boldsymbol{z})]$

with

$$f^{(n)}(\boldsymbol{s}_{\mathcal{K}}, \boldsymbol{z}) := \left\|\boldsymbol{y}[n] - \tilde{\mathbf{U}}_{\mathcal{K}}[n]\boldsymbol{s}_{\mathcal{K}}\right\|^2$$

$$= \left\|\boldsymbol{y}[n] - \Big(\tilde{\mathbf{U}}_{\mathcal{K}}[n-1] + \sum_{m\in\mathcal{E}} z_m \tilde{\mathbf{U}}[n-1]\mathbf{B}_{m,\mathcal{K}}[n-1]\Big)\boldsymbol{s}_{\mathcal{K}}\right\|^2$$

# Online learning strategy

Optimization problem: Using a least-mean square (LMS) approach, the problem is

$$\{\boldsymbol{s}_{\mathcal{K}}[n], \boldsymbol{z}[n]\} = \arg\min_{\boldsymbol{s}_{\mathcal{K}}, \boldsymbol{z}} \quad f^{(n)}(\boldsymbol{s}_{\mathcal{K}}, \boldsymbol{z}) + \gamma \parallel \boldsymbol{z} \parallel_1 \qquad (\mathcal{P})$$

$$\begin{aligned}
s.t. \quad & \text{a) } z_m \in \{-1, 0, 1\}, \;\; m = 1, \ldots, E, \\
& \text{b) } \boldsymbol{z}^T \boldsymbol{c}_i[n-1] \leq \epsilon g_i^+[n-1], \quad i = 2, \ldots, K, \\
& \text{c) } \boldsymbol{z}^T \boldsymbol{c}_i[n-1] \geq -\epsilon g_i^-[n-1], \; i = 3, \ldots, K, \\
& \text{d) } \boldsymbol{z}^T \boldsymbol{c}_2[n-1] \geq \delta - \tilde{\lambda}_2[n-1], \\
& \text{e) } \tilde{\mathbf{L}}_{ij}[n-1] + \sum_{m \in \mathcal{E}} z_m(\, \boldsymbol{a}_m \; \boldsymbol{a}_m^T)_{ij} \leq 0, \; \forall i, j, \; i \neq j.
\end{aligned}$$

- b)-d) force the small perturbation condition
- the coefficient $\delta > 0$ in d) ensures graph connectivity
- e) forces $\tilde{\mathbf{L}}[n]$ to be a valid Laplacian

# Online learning strategy

Optimization problem: Using a least-mean square (LMS) approach, the problem is

$$\{\boldsymbol{s}_{\mathcal{K}}[n], \boldsymbol{z}[n]\} = \arg\min_{\boldsymbol{s}_{\mathcal{K}}, \boldsymbol{z}} \quad f^{(n)}(\boldsymbol{s}_{\mathcal{K}}, \boldsymbol{z}) + \gamma \parallel \boldsymbol{z} \parallel_1 \qquad (\mathcal{P})$$

$$\begin{aligned} s.t. \quad & \text{a) } z_m \in \{-1, 0, 1\}, \;\; m = 1, \ldots, E, \\ & \text{b) } \boldsymbol{z}^T \boldsymbol{c}_i[n-1] \leq \epsilon g_i^+[n-1], \quad i = 2, \ldots, K, \\ & \text{c) } \boldsymbol{z}^T \boldsymbol{c}_i[n-1] \geq -\epsilon g_i^-[n-1], \; i = 3, \ldots, K, \\ & \text{d) } \boldsymbol{z}^T \boldsymbol{c}_2[n-1] \geq \delta - \tilde{\lambda}_2[n-1], \\ & \text{e) } \tilde{\mathbf{L}}_{ij}[n-1] + \sum_{m \in \mathcal{E}} z_m(\, \boldsymbol{a}_m \; \boldsymbol{a}_m^T)_{ij} \leq 0, \; \forall i, j, \; i \neq j. \end{aligned}$$

- b)-d) force the small perturbation condition
- the coefficient $\delta > 0$ in d) ensures graph connectivity
- e) forces $\tilde{\mathbf{L}}[n]$ to be a valid Laplacian

Problem $\mathcal{P}$ is a mixed integer nonconvex problem!

# Online learning strategy

Proposed algorithm for solving $\mathcal{P}$:

1) First relax the integer constraint to a box constraint $z_m \in [-1, 1]$ and rewrite $\mathcal{P}$ as

$$\begin{aligned} \min_{\boldsymbol{s}_{\mathcal{K}}, \boldsymbol{z}} \quad & f^{(n)}(\boldsymbol{s}_{\mathcal{K}}, \boldsymbol{z}) + \gamma \parallel \boldsymbol{z} \parallel_1 \\ s.t. \quad & \boldsymbol{g}_l[n-1] \leq \mathbf{H}[n-1]\boldsymbol{z} \leq \boldsymbol{g}_u[n-1] \end{aligned} \qquad (\mathcal{P}_r)$$

   The set $\mathcal{S}$ is a convex polyhedron but $\mathcal{P}_r$ is still non-convex in the objective function

2) LMS-type iterative algorithm: decouple $\mathcal{P}_r$ into two simpler convex optimization problems by alternating between the minimization with respect to $\boldsymbol{s}_{\mathcal{K}}$ and $\boldsymbol{z}$

# Online learning strategy

At each iteration $n$:

1) Estimate the GFT vector $\hat{\boldsymbol{s}}_{\mathcal{K}}[n]$ through a steepest-descent procedure

$$\hat{\boldsymbol{s}}_{\mathcal{K}}[n] = \hat{\boldsymbol{s}}_{\mathcal{K}}[n-1] - \mu \nabla_{\boldsymbol{s}_{\mathcal{K}}} f^{(n)}(\hat{\boldsymbol{s}}_{\mathcal{K}}[n-1], \hat{\boldsymbol{z}}[n-1])$$

2) Given $\hat{\boldsymbol{s}}_{\mathcal{K}}[n]$, find $\boldsymbol{z}$ using proximal splitting decomposition method:

- Compute $\hat{\boldsymbol{z}}[n] = \hat{\boldsymbol{z}}[n-1] - \mu \nabla_{\boldsymbol{z}} f^{(n)}(\hat{\boldsymbol{s}}_{\mathcal{K}}[n], \hat{\boldsymbol{z}}[n-1])$
- Given $\hat{\boldsymbol{z}}[n]$, the proximal operator

$$\text{prox}_{\mathbb{I}_{\mathcal{S}}} = \underset{\boldsymbol{y} \in \mathcal{S}}{\arg\min} \frac{1}{2} \parallel \boldsymbol{y} - \hat{\boldsymbol{z}}[n] \parallel^2$$

  is the iterative projection $\Pi_{\text{HS}_i}(\hat{\boldsymbol{z}}[n])$ of $\hat{\boldsymbol{z}}[n]$ onto the halfspaces $\text{HS}_i$

- Given the projection $\bar{\boldsymbol{z}}[n]$, the proximal $\hat{\boldsymbol{z}}[n]$ is

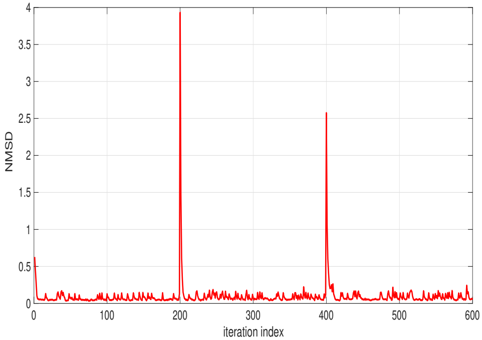
$$\hat{\boldsymbol{z}}[n] = \underset{\boldsymbol{y}}{\arg\min} \quad \parallel \boldsymbol{y} \parallel_1 + \frac{1}{2\gamma\mu} \parallel \boldsymbol{y} - \bar{\boldsymbol{z}}[n] \parallel^2 = \boldsymbol{T}_{\gamma\mu}(\bar{\boldsymbol{z}}[n])$$

  where $\boldsymbol{T}_{\gamma\mu}(\bar{\boldsymbol{z}})$ is a thresholding function

3) If a given stopping rule is satisfied, then update the eigenvectors $\tilde{\mathbf{U}}_{\mathcal{K}}[n]$ of $\tilde{\mathbf{L}}[n]$

# Numerical results

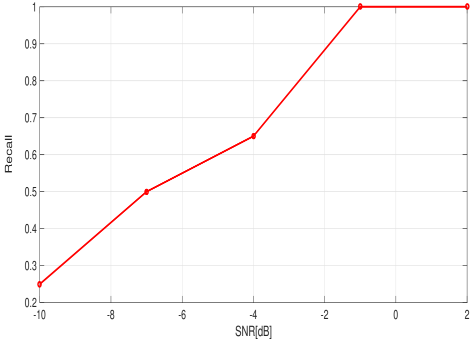
We generate graphs with $N = 30$ nodes forming 3 clusters of 10 nodes. The perturbation affects only the inter-clusters edges. Graph signal bandwidth: $K = 3$.




A nominal graph is perturbed at the iteration indexes 200 and 400. The proposed online learning strategy is able to track the graph perturbation with a very small graph signal estimation error.

$\text{NMSD} = \frac{\|\hat{\boldsymbol{x}}[n]-\boldsymbol{x}[n]\|^2}{\|\boldsymbol{x}[n]\|^2}$ vs the iteration index

# Numerical results




$$\text{Recall} = \frac{|\mathcal{E}_p \cap \hat{\mathcal{E}}_p|}{|\mathcal{E}_p|} \text{ vs SNR}$$

The capability of the method to identify the perturbed edges tends to one as SNR increases.

# Conclusions

- We proposed an online strategy to track *jointly* the perturbation of a time-varying graph and the associated signal

- Hinging on the small perturbation analysis a time-varying model for the perturbed Laplacian is derived in closed form

- Future developments:
  - enforce some kind of smoothness in both the graph and the corresponding signal evolution over time
  - improve performance by developing a second-order approximation of the eigenvectors of the perturbed Laplacian

Thanks for your attention!