Low Pass Graph Signal Processing: Modeling Data, Inference, and Beyond

Hoi-To Wai

Department of SEEM, The Chinese University of Hong Kong

June 12, 2023, GSP Workshop 2023

Acknowledgement: CUHK Direct Grant #4055135, HKRGC Project #24203520
Motivation: Network (Graph) Data

- Graph signal processing (GSP): tool to analyze network data (graph signals).
- Processes affected by irregular + relational parameters: social, economic, biological, electric power, transportation, gas, etc.
Dealing with Network Data

- **Statistics**: Gauss Markov random fields, graphical models
  — *statistical association of data*

- **Machine learning**: dimensionality reduction
  — *graph representation of data*

- **SP**: Graph Signal Processing
  — *input/output association of data*
  —* generative, interpretable model*
Dealing with Network Data

- **Statistics:** Gauss Markov random fields, graphical models  
  — statistical association of data

- **Machine learning:** dimensionality reduction  
  — graph representation of data

- **SP:** Graph Signal Processing  
  — input/output association of data  
  ⟷ generative, interpretable model
Low Pass GSP

- SP cares about the **frequency content** in a (time domain) signal — *low frequency vs high frequency*:

![Comparison of high and low frequency signals](image)

- Similar notion carries over to **graph signal processing (GSP)** — *low pass graph signals vs non low pass graph signals*:

![Graph signals with different frequencies](image)

**Takehome Point:** *Low pass* graph signals are prevalent + entail structure that enables (blind) **graph topology learning**.
Agenda

Background

Basics of GSP Models
  A Quick Introduction
  Low Pass Graph Signals

Graph Learning from Network Data
  Smoothness and Graph Learning
  Low-rank Model and Graph Feature Learning
  Learning with Partial Observation

Beyond Inference Problems & Wrapping Up

References
Agenda

Background

Basics of GSP Models
  A Quick Introduction
  Low Pass Graph Signals

Graph Learning from Network Data
  Smoothness and Graph Learning
  Low-rank Model and Graph Feature Learning
  Learning with Partial Observation

Beyond Inference Problems & Wrapping Up

References
Network Data = Filter + Excitation

- Consider a *undirected graph* \( G = (V, E, A) \) with \( N \) nodes.

- Graph signals = vectors defined on \( V \), *i.e.*, \( x \in \mathbb{R}^N \).

- As in SP, filter encodes the responses of a system to excitation.

- As SP-ers, what is our favorite form of filter?
- *Linear time invariant* filter = ‘shifting’ + ‘linear combination’.
Graph Shift Operator (GSO)

- **Starting point:** Periodic signals \( x = (x_1, \ldots, x_N) \) is ‘shifted’ on a cycle

\[
\begin{pmatrix}
  x_1 \\
  x_2 \\
  \vdots \\
  x_N
\end{pmatrix}
\rightarrow
\begin{pmatrix}
  x_N \\
  x_1 \\
  \vdots \\
  x_{N-1}
\end{pmatrix}
\]

Applying \( A \) is analogous to shifting the signal.

- **Generalization to graphs:** GSO mixes adjacent elements on \( G \).

- **Common choice of GSO:** Laplacian matrix, \( L = \text{Diag}(A1) - A \).

  — for the rest of the talk, we focus on undirected graph.

- Denote the EVD \( L = U \Lambda U^T \) with \( 0 = \lambda_1 < \cdots < \lambda_N \).

---

Graph Filters

- Consider the **graph filter** as a matrix polynomial:

\[
\mathcal{H}(L) := \sum_{\ell=0}^{+\infty} h_\ell L^\ell.
\]

**Shift-invariant prop:** \( y = \mathcal{H}(L)x \rightarrow Ly = L\mathcal{H}(L)x \equiv \mathcal{H}(L)Lx \)

- **SP/GSP Perspective:** network data are **filtered** graph signals,

\[
y = \mathcal{H}(L)x = \sum_{\ell=0}^{+\infty} h_\ell L^\ell x.
\]

- The signal/observation is \( y \) while \( x \) is viewed as the **excitation**.
What are low and high frequencies basis on graph?

- High frequency graph signal → *large variation* in adjacent entries:

\[ S(x) := \sum_{i,j} A_{ij} (x_i - x_j)^2 = x^\top L x. \]

- Intuition: if \( S(x) \) is small, the graph signal \( x \) is *smooth*. It holds \( S(u_i) = u_i^\top L u_i = \lambda_i \), as seen:

\[
\begin{array}{cccc}
\lambda_1 = 0 & \lambda_2 = 0.4706 & \lambda_{10} = 5.2813 & \lambda_{15} = 8.0818 \\
\end{array}
\]

\[
\begin{array}{cccc}
\text{lowest frequency} & \cdots & \cdots & \text{highest frequency} \\
\end{array}
\]

\[ U = (u_1 \ u_2 \ \cdots \ \ u_N) \text{ form the right basis for graph signals on } G. \]
Frequency Analysis via Graph Fourier Transform

- Graph Fourier Transform (GFT) calculates the frequency components of a signal:
  \[ \tilde{y} = U^\top y \leftarrow \tilde{y}_i = \langle u_i, y \rangle. \]

- The transfer/frequency response function of the graph filter is:
  \[ \tilde{h} = h(\lambda) \text{ where } \tilde{h}_i = h(\lambda_i) := \sum_\ell h_\ell \lambda_\ell^i. \]

- We have the convolution theorem:
  \[ y = H(L)x \iff \tilde{y} = \tilde{h} \odot \tilde{x} \leftarrow \odot \text{ is element-wise product.} \]

- Graph filter can be classified as either low-pass\(^2\), band-pass, or high-pass, depending on its graph frequency response, also see\(^3\).

\(^2\)E.g., an ideal low-pass \(\tilde{h}_1, \ldots, \tilde{h}_K = 1, \tilde{h}_{K+1}, \ldots, \tilde{h}_N = 0.\)

Low Pass Graph Filter (LPGF)

**Def.** For $1 \leq K \leq N - 1$, define
\[ \eta_K := \frac{\max\{|h(\lambda_{K+1})|, \ldots, |h(\lambda_N)|\}}{\min\{|h(\lambda_1)|, \ldots, |h(\lambda_K)|\}}. \]

If the low-pass ratio satisfies $\eta_K < 1$, then $\mathcal{H}(L)$ is $K$-low-pass.

- Integer $K$ characterizes the *bandwidth*, or the cut-off frequency.
- We say that $y$ is $K$ low pass signal provided that $y = \mathcal{H}(L)x$, where $\mathcal{H}(L)$ is $K$-low pass & $x$ satisfies some mild cond..
- Graph frequencies are non-uniformly distributed: $\lambda_K \ll \lambda_{K+1}$ tends to induce $K$-low-pass filters, e.g., stochastic block model (SBM).
Physical Models lead to Low Pass Signals

Social Network Opinions\(^4\)

- \( V = \text{individuals}, \; E = \text{friends}. \)
- DeGroot model for opinions:
  \[ y_{t+1} = (1 - \beta)(I - \alpha L)y_t + \beta x_t. \]
- **Observed** steady state:
  \[ y_\infty = (I + \tilde{\alpha} L)^{-1} x = \mathcal{H}(L)x, \]
  where \( \tilde{\alpha} = \beta (1 - \alpha) / \alpha > 0. \)

Prices in Stock Market\(^5\)

- \( V = \text{financial inst.}, \; E = \text{ties}. \)
- Business performances evolve as:
  \[ y_{t+1} = (1 - \beta)\mathcal{H}(L)y_t + \beta Bx, \]
  e.g., stock return.
- **Observed** steady state:
  \[ y_\infty = \left( \frac{1}{\beta} I - \frac{\bar{\beta}}{\beta} \mathcal{H}(L) \right)^{-1} Bx = \tilde{\mathcal{H}}(L)Bx. \]

Fact\(^6\): Both \( \mathcal{H}(L), \; \tilde{\mathcal{H}}(L) \) are **low pass** graph filters.

---

Agenda

Background

Basics of GSP Models
  A Quick Introduction
  Low Pass Graph Signals

Graph Learning from Network Data
  Smoothness and Graph Learning
  Low-rank Model and Graph Feature Learning
  Learning with Partial Observation

Beyond Inference Problems & Wrapping Up

References
Graph Learning from Network Data

- **Goal:** estimate $L$ or some information about it.
- **Working hypothesis:** a number of graph signals $y^{(t)}$ are available as observed low pass graph signals.

### GSP Model

- **Unknown Graph**
- **Observed Low Pass Graph Signals**

**Observed graph signals:**

$$y^{(t)} \approx \mathcal{H}(L)Bz^{(t)}, \quad t = 0, \ldots, T - 1$$

- $\mathcal{H}(L)$ is low pass, $z^{(t)}$ is 0-mean, $B$ is pattern of (low rank) excitation.

- **Graph learning relies on two properties of low pass signals:**
  - **Smoothness** → graph topology learning.
  - **Low-rankness** → graph feature learning (e.g., community, centrality).
Smoothness and Graph Learning

**Insight:** For $K$-low-pass graph signals ($\eta_K \ll 1$) with full-rank excitation satisfying $B = I$, we observe that

$$\mathbb{E}[y_\ell^T Ly_\ell] \approx \sum_{i=1}^{K} \lambda_i |h(\lambda_i)|^2 + \sigma^2 \text{Tr}(L) \text{low pass filter} \approx 0,$$

i.e., the low pass filtered graph signals are smooth w.r.t. $L$.

**Idea:** Fit a graph optimizing for smoothness (GL-SigRep)$^7$:

$$\min_{z_\ell, \ell=1,\ldots,m,\hat{L}} \frac{1}{m} \sum_{\ell=1}^{m} \left\{ \frac{1}{\sigma^2} \|z_\ell - y_\ell\|_2^2 + z_\ell^T \hat{L} z_\ell \right\} \leftarrow \text{note } z \approx y$$

s.t. \(\text{Tr}(\hat{L}) = N, \hat{L}_{ji} = \hat{L}_{ij} \leq 0, \forall i \neq j, \hat{L}1 = 0,\)

---

Numerical Experiment: GL-SigRep

- Topology learnt using GL-SigRep from the synthetic data generated through a low pass graph filter:

\[
y_\ell = \sqrt{L}^{-1} x_\ell, \quad x_\ell \sim \mathcal{N}(0, I),
\]

- Alternative approaches:
  - [Friedman et al., 2008] Graphical LASSO: ML estimation for GMRF.
  - [Mei and Moura, 2016] Causal modeling: time series data

---

8Image credits: [Dong et al., 2016].
Low-rank-ness and Graph Feature Learning

**Issue**: with low-rank excitation \((B \in \mathbb{R}^{N \times R} \text{ with } R < N) \rightarrow \text{graph learning} = \text{difficult} \because \text{data is nearly rank deficient}...

**Insight**: Suppose \(\mathcal{H}(L)\) is \((\eta, K)\) **low pass**, then

\[
C_y = \mathbb{E}[yy^\top] = \mathcal{H}(L)UC_xU^\top\mathcal{H}(L)^\top \approx U_KC_{\tilde{x}}U_K^\top.
\]

Thus \(C_y\) is also **low rank**!

**Approximation** holds if \(\eta \ll 1 \Rightarrow \text{low rank } \mathcal{H}(\cdot), \text{ rank}(\mathcal{H}(L)) \approx K \ll N \text{ and range space } \approx U_K.

**Idea**: spectral method to extract principal components in \(U_K\) from \(C_y\).

\[\implies \text{Can (still) learn **communities** and **centrality** of the graph.}\]
Blind community detection (Blind CD)

**Idea:** spectral clustering applied on empirical covariance \( \hat{C}_y \approx C_y \):

(i) find the top-\( k \) \( \hat{U}_K \in \mathbb{R}^{N \times K} \) of \( \hat{C}_y = \frac{1}{m} \sum_{\ell=1}^{m} y_\ell y_\ell^\top \);

(ii) apply \( k \)-means on the rows of \( \hat{U}_K \).

**Theorem:** Denote the detected clusters as \( \hat{N}_1, \ldots, \hat{N}_K \), then

\[
\underbrace{K(\hat{N}_1, \ldots, \hat{N}_K; U_K)}_{\text{K-means obj. based on } U_K} - \underbrace{K^*}_{\text{Optimal } K\text{-means obj.}} = O(\eta_k + m^{-1/2}).
\]

\( \dagger \rightarrow \) performance of spectral clustering (with known topology) if \( \eta_k \rightarrow 0 \).

**Learning of high-level structure is robust** to low-rank excitation.

**Extensions:** exact community recovery on multi-graphs

[Roddenberry et al., 2020], dynamic observations [Schaub et al., 2020], ...

---

Blind community detection (Blind CD)

Problem: What if $\eta_K \approx 1$? Let’s try $\tilde{\mathcal{H}}_{\rho}(L) := \mathcal{H}(L) - \rho I$ ($\rho > 0$).

- Original ratio: $\eta = \frac{0.7}{0.85} \approx 0.82$.
- Boosted ratio: $\tilde{\eta} = \frac{0.05}{0.25} = 0.2$.

Suppose that $Z$ is known,

$$YZ^\dagger = \mathcal{H}(L)B = \tilde{\mathcal{H}}_{\rho}(L)B + \rho B$$

- Typically, $B$ is sparse
  $\implies$ low-rank + sparse decomposition!

Freq. response

Robust PCA formulation:
$$\min_{L,B} \| YZ^\dagger - L - B \|_F^2 + \gamma \| L \|_* + \mu \| B \|_1$$
(a) As $R = \text{rank}(C_x)$ increases, Blind CD approaches the performance of spectral clustering on the true GSO.
Blind Centrality Learning

- Eigen-centrality $= \text{TopEV}(A)$ is revealed by $\text{TopEV}(C_y)$ for 1-low pass signals $\implies$ a simple PCA procedure suffices:

\[
C_y = \frac{1}{m} \sum_{t=1}^{m} y^t (y^t)^T
\]

Sample Covariance

$\hat{v}_1 := \text{TopEV}(C_y)$

Centrality Estimation

Detected K possible central nodes

- **Theorem**\(^{10}\): let $v_1$ be the true eig. centrality,

\[
\|\hat{v}_1 - u_1\|_2 = \mathcal{O}(\eta_1 + m^{-1/2}).
\]

To obtain a robust formulation against $\eta_1 \approx 1$, assume that $B$ is *sparse* and use similar idea as Blind CD:

$$Y = \mathcal{H}(A)BZ = \left( (\mathcal{H}(A) - \rho I)B + \rho B \right)Z$$

$$= \text{(Low-rank + Sparse)} \times Z$$

**Structured factor analysis**: if $Z$ is unknown,

1. decompose $Y$ via NMF,
2. Robust PCA.

**Theoretical analysis** (for NMF): good performance if (i) $N/\text{rank}(Z)$ is large, (ii) $\text{rank}(Z)$ is large.

— trade-off between low-pass-ness and NMF performance.
— derived from [Fu et al., 2019].

**Related Works**: centrality ranking [Roddenberry and Segarra, 2021].
Numerical Experiment: Blind Centrality Learning

- For each model: (L) $H_{\text{weak}}$ and (R) $H_{\text{strong}}$.
- The error rate for most methods decreases with $k$.
- RPCA and proposed algorithm outperform other algorithms.

Graph filter $H(\cdot)$ is (left) ‘weak’ low pass, i.e., $\eta \approx 1$; (right) ‘strong’ low pass, i.e., $\eta \ll 1$.

- Proposed **Algorithm 1** with NMF outperforms SOTA in the considered setting for ‘weak’ low pass; and similarly as PCA for ‘strong’ low pass.
Numerical Experiment: Blind Centrality Learning

(left) ‘Strong’ low pass, (right) ‘Weak’ low pass
Numerical Experiment: Blind Centrality Learning

(a) Stock Dataset\(^1\)

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-10 Estimated Central Stocks (sorted left-to-right)</th>
<th>Average Correlation Score: 0.56 ± 0.154</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>ALL</td>
<td>ACN</td>
</tr>
<tr>
<td></td>
<td>0.43</td>
<td>0.56</td>
</tr>
<tr>
<td>PCA (11)</td>
<td>NVIDIA</td>
<td>NFLX</td>
</tr>
<tr>
<td></td>
<td>0.56</td>
<td>0.60</td>
</tr>
<tr>
<td>GL-SigRep [13]</td>
<td>GOOGLE</td>
<td>GOOG</td>
</tr>
<tr>
<td></td>
<td>0.63</td>
<td>0.63</td>
</tr>
<tr>
<td>KNN</td>
<td>ACN</td>
<td>HON</td>
</tr>
<tr>
<td></td>
<td>0.56</td>
<td>0.51</td>
</tr>
<tr>
<td>SpecTemp [14]</td>
<td>ACN</td>
<td>ORCL</td>
</tr>
<tr>
<td></td>
<td>0.56</td>
<td>0.70</td>
</tr>
<tr>
<td>Kalofolias [44]</td>
<td>ACN</td>
<td>HON</td>
</tr>
<tr>
<td></td>
<td>0.56</td>
<td>0.51</td>
</tr>
</tbody>
</table>

(b) Senate Dataset\(^1\)

<table>
<thead>
<tr>
<th>Method</th>
<th>Top-10 Estimated Central States (sorted left-to-right)</th>
<th>Average Correlation Score: 0.66 ± 0.099</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algorithm</td>
<td>MI</td>
<td>MT</td>
</tr>
<tr>
<td></td>
<td>0.79</td>
<td>0.66</td>
</tr>
<tr>
<td>PCA (11)</td>
<td>CA</td>
<td>DE</td>
</tr>
<tr>
<td></td>
<td>0.55</td>
<td>0.46</td>
</tr>
<tr>
<td>GL-SigRep [13]</td>
<td>CA</td>
<td>DE</td>
</tr>
<tr>
<td></td>
<td>0.55</td>
<td>0.46</td>
</tr>
<tr>
<td>KNN</td>
<td>ND</td>
<td>CA</td>
</tr>
<tr>
<td></td>
<td>0.72</td>
<td>0.55</td>
</tr>
<tr>
<td>SpecTemp [14]</td>
<td>AL</td>
<td>ND</td>
</tr>
<tr>
<td></td>
<td>0.61</td>
<td>0.72</td>
</tr>
<tr>
<td>Kalofolias [44]</td>
<td>AL</td>
<td>AK</td>
</tr>
<tr>
<td></td>
<td>0.61</td>
<td>0.63</td>
</tr>
</tbody>
</table>

\(^{11}\)The number below each stock/state shows its normalized correlation score with the S&P100 index and number of ‘Yay’ in the voting result [cf. (36)]. The average correlation scores are taken over the set of central nodes found and the number after ‘±’ is the standard deviation.

(a) Detected central nodes with performance measured on correlation of nodes with (left) S&P500 index, (right) voting outcomes.

Extension: *Multiple graph* learning from streaming data\(^{11}\).

In many settings, we do not observe **complete graph signals** on every nodes, e.g., large social network, power network, etc.

Hidden nodes remain **influential** and affect the observations:

\[ y = \mathcal{H}(L)x \quad \text{with} \quad y = \begin{bmatrix} y_{\text{obs}} \\ y_{\text{hid}} \end{bmatrix}, \quad L = \begin{bmatrix} L_{o,o} & L_{o,h} \\ L_{h,o} & L_{h,h} \end{bmatrix} \]
Learning with Partial Observation

▶ **Goal:** infer about the subgraph of observable nodes, $L_{o,o}$:

\[ y = \mathcal{H}(L)x = \begin{bmatrix} y_{\text{obs}} \\ y_{\text{hid}} \end{bmatrix}, \quad C_y = \begin{bmatrix} C_y^o & C_y^{o,h} \\ C_y^{h,o} & C_y^h \end{bmatrix}, \quad L = \begin{bmatrix} L_{o,o} & L_{o,h} \\ L_{h,o} & L_{h,h} \end{bmatrix} \]

1. **Leveraging Smoothness:** observing that\(^{12}\)

\[
\frac{1}{m} \sum_{i=1}^{m} \mathbf{y}_\ell^\top L \mathbf{y}_\ell \approx \text{Tr}(C_y^o L_{o,o}) + \text{Tr}(2C_y^{o,h} L_{o,h}^\top) + \text{Tr}(C_y^h L_{h,h}) \geq 0
\]

low rank if $|V_{\text{hid}}| \ll N \geq 0$

\[\implies \min_{L_{o,o},K,R} \text{Tr}(C_y^o L_{o,o}) + \text{Tr}(K) + \text{Tr}(R) + \alpha g(L_{o,o}) + \gamma\|K\|_* \]

s.t. \[\text{Tr}(C_y^o L_{o,o}) + \text{Tr}(K) + \text{Tr}(R) \geq 0, \quad \text{Tr}(R) \geq 0, \quad L_{o,o} \in \mathcal{L}, \]

where $g(\cdot)$, $\mathcal{L}$ are respectively regularization, constraint for $L_{o,o}$ to be a proper sub-matrix of Laplacian.

Learning with Partial Observation

Goal: infer about the subgraph of observable nodes, $L_{o,o}$:

$$y = \mathcal{H}(L)x = \begin{bmatrix} y_{\text{obs}} \\ y_{\text{hid}} \end{bmatrix}, \quad C_y = \begin{bmatrix} C_y^o & C_y^{o,h} \\ C_y^{h,o} & C_y^h \end{bmatrix}, \quad L = \begin{bmatrix} L_{o,o} & L_{o,h} \\ L_{h,o} & L_{h,h} \end{bmatrix}$$

I. Leveraging Smoothness: observing that\(^{12}\)

$$\frac{1}{m} \sum_{i=1}^{m} y_{\ell}^\top Ly_{\ell} \approx \text{Tr}(C_y^o L_{o,o}) + \text{Tr}(2C_y^{o,h} L_{o,h}^\top) + \text{Tr}(C_y^h L_{h,h}) \geq 0$$

low rank if $|V_{\text{hid}}| \ll N$ ≥0

$$\implies \min_{L_{o,o}, K, R} \text{Tr}(C_y^o L_{o,o}) + \text{Tr}(K) + \text{Tr}(R) + \alpha g(L_{o,o}) + \gamma \|K\|_*$$

s.t. $\text{Tr}(C_y^o L_{o,o}) + \text{Tr}(K) + \text{Tr}(R) \geq 0$, $\text{Tr}(R) \geq 0$, $L_{o,o} \in \mathcal{L}$,

where $g(\cdot)$, $\mathcal{L}$ are respectively regularization, constraint for $L_{o,o}$ to be a proper sub-matrix of Laplacian.

Learning with Partial Observation

- **Goal**: infer about the subgraph of observable nodes, $L_{o,o}$:

$$y = \mathcal{H}(L)x = \begin{bmatrix} y_{obs} \\ y_{hid} \end{bmatrix}, \quad C_y = \begin{bmatrix} C_y^o & C_{y, h}^o \\ C_y^h & C_y^h \end{bmatrix}, \quad L = \begin{bmatrix} L_{o,o} & L_{o,h} \\ L_{h,o} & L_{h,h} \end{bmatrix}$$

II. Leveraging Lowrank-ness: provided $\mathcal{H}(L)$ is $(\eta, K)$ low pass,

$$C_y^o = E_o C_y E_o^T \approx (E_o U_K) C_x (E_o U_K)^T$$

where $E_o$ is row-selection matrix for $V_{obs}$. ↑ can estimate $E_o U_K \approx U_{K,o}$

- **Key observation**: low-rankness of $\mathcal{H}(L)$ supersedes partial obs.

- Straightforward extension for graph feature learning: partial community detection\textsuperscript{12}, partial centrality inference\textsuperscript{13}


\textsuperscript{13}[He and Wai, 2023a] Y. He, H.-T., Central nodes detection from partially observed graph signals, in ICASSP 2023.
Complete Learning with Partial Observation

- **Goal**: inferring the graph features of the whole \( A \),

\[
y = \mathcal{H}(A)x = \begin{bmatrix} y_{obs} \\ y_{hid} \end{bmatrix}, \quad C_y = \begin{bmatrix} C_y^o & C_y^{o,h} \\ C_y^{h,o} & C_y^h \end{bmatrix}, \quad A = \begin{bmatrix} A_{o,o} & A_{o,h} \\ A_{h,o} & A_{h,h} \end{bmatrix}
\]

- Requires *side information or sub-graph topology*:

- We rely on *low-rankness* and aim to learn community or centrality.
Complete Learning with Partial Observation

▶ **Goal**: inferring the graph features of the whole $A$,

$$y = \mathcal{H}(A)x = \begin{bmatrix} y_{\text{obs}} \\ y_{\text{hid}} \end{bmatrix}, \quad C_y = \begin{bmatrix} C_y^o & C_y^{o,h} \\ C_y^{h,o} & C_y^h \end{bmatrix}, \quad A = \begin{bmatrix} A_{o,o} & A_{o,h} \\ A_{h,o} & A_{h,h} \end{bmatrix}$$

▶ Requires *side information or sub-graph topology*:

(I) **If $A_{o,h}$ is known**: Nyström method [Fowlkes et al., 2004] to ‘interpolate’ eigenvectors,

(i) top-$K$ $\hat{U}_K$ of $\hat{C}_y^{\text{obs}}$, (ii) $\hat{V}_K := \begin{pmatrix} \hat{U}_K \\ A_{h,o} \frac{\hat{U}_K}{\hat{\lambda}} \end{pmatrix}$, (iii) k-means on $\hat{V}_K$.

▶ Assume that $V_{\text{obs}}$ is chosen at random, then w.h.p.,

$$F(\tilde{\mathcal{N}}_1, \ldots, \tilde{\mathcal{N}}_K; V_K) \rightarrow F^* = \mathcal{O} \left( \eta_K + \frac{1}{\sqrt{m}} + \frac{1}{\sqrt{|V_{\text{obs}}|}} + \frac{|V_{\text{hid}}|}{|V|} \right).$$

---

Complete Learning with Partial Observation

- **Goal**: inferring the graph features of the whole $A$,

\[
y = \mathcal{H}(A)x = \begin{bmatrix} y_{\text{obs}} \\ y_{\text{hid}} \end{bmatrix}, \quad C_y = \begin{bmatrix} C_y^o & C_y^{o,h} \\ C_y^{h,o} & C_y^h \end{bmatrix}, \quad A = \begin{bmatrix} A_{o,o} & A_{o,h} \\ A_{h,o} & A_{h,h} \end{bmatrix}
\]

- Requires *side information or sub-graph topology*:
  
  (II) Excitation signal is known\(^\text{14}\): recall $x^{(t)} = Bz^{(t)}$ and we know $B, z^{(t)}$.

\[
Y_{\text{obs}}Z^\dagger = \tilde{h}_\rho(\lambda_1)c_{\text{obs}}c^\top B + \rho E_0B + \mathcal{O}(\tilde{\eta}), \quad \text{holds} \quad \forall \rho > 0
\]

- **Full eigen-centrality $c$** can be estimated if

\[
\text{Excitation rank} = \text{rank}(B) = K \geq |V_{\text{hid}}| + 1
\]

\(^{14}\)[He and Wai, 2023a] Y. He, H.-T., Central nodes detection from partially observed graph signals, in ICASSP 2023.
Numerical Experiment: Complete Graph Learning

Increasing the excitation rank $K$ improves the detection performances.
Agenda

Background

Basics of GSP Models

A Quick Introduction

Low Pass Graph Signals

Graph Learning from Network Data

Smoothness and Graph Learning

Low-rank Model and Graph Feature Learning

Learning with Partial Observation

Beyond Inference Problems & Wrapping Up

References
Detecting Low-pass Signals

**Question**: How do we know if a set of graph signals are low pass?

- Topology inferred from non low pass signals can be **deceptive**.

![Graphs showing ground truth and topology learnt by GL-SigRep on non-low-pass signals.](image)

- **Challenges**: graph topology $A$ and filter $\mathcal{H}(A)$ are **unknown**.
- **Warning**: an **ill posed** problem – graph signals is **smooth** on one graph, but **non-smooth** on another.
Detecting Low-pass Signals

- **Assume**: no. of dense clusters, $K$, in the graph is known a-priori.
  \[ \Rightarrow \lambda_1, \ldots, \lambda_K \approx 0 \Rightarrow \text{if the filter is low pass, it will be } K \text{ low pass.} \]

- **Observation**: graph signals from $K$ low pass filter exhibit particular spectral signature. E.g., SBM graph with $K = 3$ clusters,

![Graph showing spectral signature](image)

**Idea**: Measure *clusterability* of principal eigenvectors.
Application: Robustifying Graph Learning

What if graph signals are corrupted with non-low-pass observations? ⇒ screen them out by a blind detector and apply [Dong et al., 2016].

(a) Ground truth graph learnt from clean data.
(b) Graph learnt from corrupted data (mixed w/ high-pass signals).
(c) Graph learnt after the pre-screening procedure.

▶ Other applications: blind detection of network dynamics, blind anomaly detection, etc.\textsuperscript{15}

Stability of Graph Filter with Edge Rewiring

- Graph filter is an important building block of Graph Convolutional Neural Network (GCN) → trained on $\mathcal{H}(L)$, but applied on $\mathcal{H}(\hat{L})$.
- **Stability**\(^{16}\) is related to *transferability* of GCNs. Existing results require small no. of edge rewrites.

**Frequency-domain bound:** If $\mathcal{H}(L)$ is low pass, then

$$\|\mathcal{H}(L) - \mathcal{H}(\hat{L})\| = O(\eta + \|U_k - \hat{U}_k\| + \|\Lambda_k - \hat{\Lambda}_k\|),$$

where $U_k - \hat{U}_k$, $\Lambda_k - \hat{\Lambda}_k$ are perturbations of top eigenvectors/values.

- Residuals $\rightarrow 0$ for edge rewiring on SBMs perturbations\(^{17}\).
  — Proof: depends on convergence of graph filter on SBM.


Stability of Graph Filter with Edge Rewiring

*Frequency-domain bound:* If $H(L)$ is **low pass**, then

$$
\|H(L) - H(\hat{L})\| = O(\eta + \|U_k - \hat{U}_k\| + \|\Lambda_k - \hat{\Lambda}_k\|),
$$

where $U_k - \hat{U}_k$, $\Lambda_k - \hat{\Lambda}_k$ are perturbations of top eigenvectors/values.

- Low pass filters are **insensitive** to no. of rewiring vs. high pass filters.
Generalization Bound

- **Sample complexity** of MPNN (GCN) learning\(^{18}\) analyzed via

\[ \mathcal{E}_m^n = \mathbb{E}_{\mu_G}^m \left[ \sup_{\Theta} \left( \frac{1}{m} \sum_{i=1}^m \mathcal{L}(\Theta_{G^i}(x^i), y^i) - \mathbb{E}_{\mu_G} [\mathcal{L}(\Theta_G(x), y)] \right)^2 \right] \leq \frac{C}{m} n^{-\frac{1}{D+1}} \]

where \(m = \) no. of training sets, \(n = \) no. of nodes, and \(G^i, x^i, y^i\) is the \(i\)th training set of graph, attributes (signals), labels.

- **Proof**: MPNN \(\rightarrow\) graphon limit as \(n \rightarrow \infty\) [Keriven et al., 2020].

- \(C\) depends on Lipschitz-ness of message (activation) functions, etc. \(\leftarrow\) no explicit dependence on graph filter.

- **Recent work**\(^{19}\) provide transferability bound utilizing the spectrum of graph filter similar to [Keriven et al., 2020] \(\leftarrow\) open problem?

---


Wrapping Up

Takehome Point: *Low pass* graph signals are prevalent + entail structure that enables (blind) graph topology learning.

- **Smoothness** → graph topology learning.
- **Low-rankness** → topology feature learning (centrality, community).
  - also for learning from partial observation, ...

- Related problems: how to detect low pass signals, application to graph ML, ...
Perspectives

- Graph learning from partial observations with many hidden nodes. — it is the case for observations on social/economics networks.

- Learning from multi-attribute signal: graphs do not live in isolation, e.g., multiplex networks in ecology, social systems, etc. — needs new notion for graph filter:

\[ \mathcal{H}(L^C, L^G) = \sum_{i,j} h_{ij}(L^C)^i \otimes (L^G)^j, \]

and interpretation for low pass multi-layer graph filter [Zhang et al., 2023b, Kadambari and Chepuri, 2021, Einizade and Sardouie, 2022].

Thank you!
Questions & comments are welcomed.
Agenda

Background

Basics of GSP Models
  A Quick Introduction
  Low Pass Graph Signals

Graph Learning from Network Data
  Smoothness and Graph Learning
  Low-rank Model and Graph Feature Learning
  Learning with Partial Observation

Beyond Inference Problems & Wrapping Up

References
Econometric measures of connectedness and systemic risk in the finance and insurance sectors.

Learning graphs from smooth and graph-stationary signals with hidden variables.
IEEE Transactions on Signal and Information Processing over Networks, 8:273–287.

Reaching a consensus.

Learning Laplacian matrix in smooth graph signal representations.

Learning product graphs from spectral templates.

Spectral grouping using the nystrom method.

Sparse inverse covariance estimation with the graphical lasso.
Biostatistics, 9(3):432–441.
Nonnegative matrix factorization for signal and data analytics: Identifiability, algorithms, and applications.


Detecting central nodes from low-rank excited graph signals via structured factor analysis.
IEEE Transactions on Signal Processing.

Central nodes detection from partially observed graph signals.
In ICASSP. IEEE.

Online inference for mixture model of streaming graph signals with non-white excitation.

Graph filters for signal processing and machine learning on graphs.

Product graph learning from multi-domain data with sparsity and rank constraints.

Convergence and stability of graph convolutional networks on large random graphs.
Generalization analysis of message passing neural networks on large random graphs.
In Advances in Neural Information Processing Systems.

Signal processing on graphs: Causal modeling of unstructured data.

On the stability of low pass graph filter with a large number of edge rewrites.
In IEEE ICASSP.

The algebraic approach to the discrete cosine and sine transforms and their fast algorithms.

A user guide to low-pass graph signal processing and its applications.

Exact blind community detection from signals on multiple graphs.

Blind inference of eigenvector centrality rankings.

Transferability properties of graph neural networks.

Discrete signal processing on graphs.

Blind identification of stochastic block models from dynamical observations.

Network topology inference from spectral templates.

Community inference from partially observed graph signals: Algorithms and analysis.

Blind community detection from low-rank excitations of a graph filter.

Detecting low pass graph signals via spectral pattern: Sampling complexity and applications.
*ArXiv*.

Product graph learning from multi-attribute graph signals with inter-layer coupling.
In *ICASSP*. 

38 / 38