Product Graph Learning from Multi-attribute Graph Signals with Inter-layer Coupling

Chenyue Zhang, Yiran He, Hoi-To Wai

Department of Systems Engineering & Engineering Management, CUHK

ICASSP 2023, Greece

Acknowledgement: CUHK Direct Grant #4055135

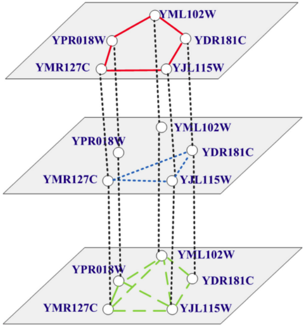
Motivation

[Source: NETWORK MANAGEMENT]

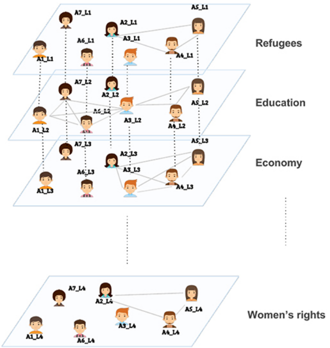
[Source: Sapien Labs]

- **Graphs:** Effectively depict data's spatial layout in diverse fields like **social**, **biology**, **transportation**, and **power networks**.
- **GSP:** A **flexible tool** that extends the concepts from **classical signal processing** to graphs and makes inference of data.

Motivation



Multilayer protein networks [Source: [Zhao et al., 2016]]

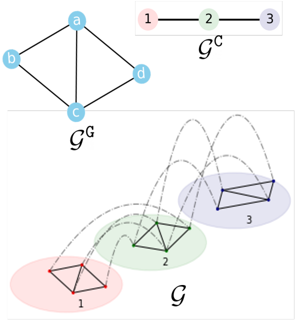


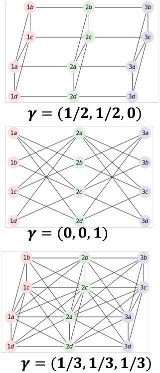
Multilayer social network [Source: [Hanteer and Rossi, 2019]]

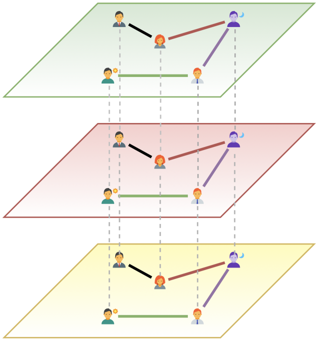
- **From single-way to multi-attribute data:** Complex systems are represented by multi-layer networks.
- **Multi-way graph signals:** E.g., opinion dynamics, multi-dimensional diffusion, protein-protein interactions, animal networks, and relations in image pixels.

Product Graph Model

- Coupling graph: $\mathcal{G}^{\mathtt{C}} = (\mathcal{V}^{\mathtt{C}}, \mathcal{E}^{\mathtt{C}}, \boldsymbol{A}^{\mathtt{C}})$ with $|\mathcal{V}^{\mathtt{C}}| = M$
- Physical graph: $\mathcal{G}^{\mathtt{G}} = (\mathcal{V}^{\mathtt{G}}, \mathcal{E}^{\mathtt{G}}, \boldsymbol{A}^{\mathtt{G}})$ with $|\mathcal{V}^{\mathtt{G}}| = N$







Adjacency matrix for **generalized** product graph $\mathcal{G} = (\mathcal{G}^{\mathtt{C}}, \mathcal{G}^{\mathtt{G}})$:

$$\boldsymbol{A} = \gamma_1 \boldsymbol{I} \otimes \boldsymbol{A}^{\mathtt{G}} + \gamma_2 \boldsymbol{A}^{\mathtt{C}} \otimes \boldsymbol{I} + \gamma_3 \boldsymbol{A}^{\mathtt{C}} \otimes \boldsymbol{A}^{\mathtt{G}}$$

Multi-attribute Graph Filter and Signals.

A **general** multi-layer graph filter model[1]:

$$\mathcal{H}(\boldsymbol{A}^{\mathtt{C}}, \boldsymbol{A}^{\mathtt{G}}) = \sum_{i=0}^{T_{\mathtt{G}}} \sum_{j=0}^{T_{\mathtt{C}}} h_{ij} (\boldsymbol{A}^{\mathtt{C}})^j \otimes (\boldsymbol{A}^{\mathtt{G}})^i$$

$$\boldsymbol{y}^{(s)} = \mathcal{H}(\boldsymbol{A}^{\mathtt{C}}, \boldsymbol{A}^{\mathtt{G}}) \boldsymbol{x}^{(s)} + \boldsymbol{w}^{(s)}$$

- Multi-dimensional opinion dynamics $\gamma = (0, 0, 1)$

$$\boldsymbol{y}^{(s)} = \lim_{t \to \infty} \boldsymbol{x}(t) = (\boldsymbol{I}_{NM} - \boldsymbol{A}^{\mathtt{C}} \otimes \boldsymbol{A}^{\mathtt{G}})^{-1} \boldsymbol{x}^{(s)}$$

- Diffusion Process $\gamma = (\frac{1}{2}, \frac{1}{2}, 0)$

$$\boldsymbol{y}^{(s)} = (\boldsymbol{I}_{NM} - \boldsymbol{A}^{\mathtt{C}} \otimes \boldsymbol{I}_N - \boldsymbol{I}_M \otimes \boldsymbol{A}^{\mathtt{G}})^{-1} \boldsymbol{x}^{(s)}$$

[1] Not necessarily a polynomial of $\boldsymbol{A}$.

Product Graph Learning

Task: Given $\boldsymbol{y}^{(s)}$, recover $\boldsymbol{A}^{\mathtt{C}}, \boldsymbol{A}^{\mathtt{G}}$ or their spectral features.

Idea: many graph features are embedded in the spectra:

$$\boldsymbol{A}^{\mathtt{C}} = \boldsymbol{V}^{\mathtt{C}} \boldsymbol{\Lambda}^{\mathtt{C}} (\boldsymbol{V}^{\mathtt{C}})^{\top}, \boldsymbol{A}^{\mathtt{G}} = \boldsymbol{V}^{\mathtt{G}} \boldsymbol{\Lambda}^{\mathtt{G}} (\boldsymbol{V}^{\mathtt{G}})^{\top}$$

If $\boldsymbol{V}^{\mathtt{C}}, \boldsymbol{V}^{\mathtt{G}}$ are known, then

- Topology Reconstruction [Segarra et al., 2017]

$$\min_{\boldsymbol{\lambda}^{\mathtt{G}}, \widehat{\boldsymbol{A}}^{\mathtt{G}}} \ \|\mathrm{vec}(\widehat{\boldsymbol{A}}^{\mathtt{G}})\|_1 + \frac{\rho}{2} \|\widehat{\boldsymbol{A}}^{\mathtt{G}} - \boldsymbol{V}^{\mathtt{G}} \mathrm{Diag}(\boldsymbol{\lambda}^{\mathtt{G}}) (\boldsymbol{V}^{\mathtt{G}})^{\top}\|_F^2 \longleftarrow \text{similar for } \boldsymbol{A}^{\mathtt{C}}$$

$$\text{s.t. } |\mathbf{diag}(\widehat{\boldsymbol{A}}^{\mathtt{G}})| \leq \epsilon \mathbf{1}, \widehat{\boldsymbol{A}}^{\mathtt{G}} \mathbf{1} \geq \mathbf{1}$$

- Centrality Estimation

$$\text{take } \ \widehat{\boldsymbol{c}}^{\mathtt{G}} = \boldsymbol{v}_c^{\mathtt{G}}, \ \widehat{\boldsymbol{c}}^{\mathtt{C}} = \boldsymbol{v}_c^{\mathtt{C}} \ \text{ where } \ \widehat{\boldsymbol{v}}_{i^\star} = \boldsymbol{v}_c^{\mathtt{C}} \otimes \widehat{\boldsymbol{v}}_c^{\mathtt{G}}$$

see [Roddenberry and Segarra, 2021, He and Wai, 2022]

Product Graph Learning - Algorithm

Step 1 **Recover $V^{\mathtt{C}}, V^{\mathtt{G}}$ from covariance matrix**[2]

$$C_y = \mathbb{E}[y^{(s)}(y^{(s)})^\top] = (V^{\mathtt{C}} \otimes V^{\mathtt{G}})|\mathcal{H}(\Lambda^{\mathtt{C}}, \Lambda^{\mathtt{G}})|^2 (V^{\mathtt{C}} \otimes V^{\mathtt{G}})^\top + \sigma^2 I$$

with $x^{(s)}, w^{(s)}$ satisfying white noise conditions.

Step 2 Spectral-based methods for graph (feature) learning

(P1) Topology Reconstruction [Segarra et al., 2017]

$$\min_{\lambda^{\mathtt{G}}, \widehat{A}^{\mathtt{G}}} \|\mathrm{vec}(\widehat{A}^{\mathtt{G}})\|_1 + \frac{\rho}{2}\|\widehat{A}^{\mathtt{G}} - \widehat{V}^{\mathtt{G}}\mathrm{Diag}(\lambda^{\mathtt{G}})(\widehat{V}^{\mathtt{G}})^\top\|_F^2$$

$$\text{s.t. } |\mathbf{diag}(\widehat{A}^{\mathtt{G}})| \leq \epsilon \mathbf{1}, \widehat{A}^{\mathtt{G}}\mathbf{1} \geq \mathbf{1}$$

(P2) Centrality Estimation [Roddenberry and Segarra, 2021, He and Wai, 2022]

$$\widehat{c}^{\mathtt{G}} = \widehat{v}_c^{\mathtt{G}}, \ \widehat{c}^{\mathtt{C}} = \widehat{v}_c^{\mathtt{C}} \quad \text{where} \quad \widehat{v}_{i^\star} = \widehat{v}_c^{\mathtt{C}} \otimes \widehat{v}_c^{\mathtt{G}}$$

[2] **A1**: Assume magnitudes of frequency response $|h(\lambda_j^{\mathtt{C}}, \lambda_i^{\mathtt{G}})|$ are distinct.

Exact Solution by NKD to step 1

- Calculate sample covariance's eigenvectors

$$\widehat{\boldsymbol{C}}_y^S = (1/S)\sum_{s=1}^{S} \boldsymbol{y}^{(s)}(\boldsymbol{y}^{(s)})^\top \quad \widehat{\boldsymbol{V}}^{\text{noisy}} = \text{EVD}(\widehat{\boldsymbol{C}}_y^S)$$

- **Columns** of $\widehat{\boldsymbol{V}}$ can be expressed as **Kronecker products**

$$\widehat{\boldsymbol{V}} = (\boldsymbol{V}^{\mathtt{C}} \otimes \boldsymbol{V}^{\mathtt{G}})\boldsymbol{\Pi} = \left[\cdots \ \boldsymbol{v}^{\mathtt{C}}_{\pi^{\mathtt{C}}(i)} \otimes \boldsymbol{v}^{\mathtt{G}}_{\pi^{\mathtt{G}}(i)} \cdots \right]$$

- NKD[3] recovers $(\widehat{\boldsymbol{v}}_i^{\mathtt{C}}, \widehat{\boldsymbol{v}}_i^{\mathtt{G}})$ from $\widehat{\boldsymbol{v}}_i^{\text{noisy}}$ via SVD.

Corollary: Under **A1**, noiseless observations, and $S \to \infty$, the NKD procedure exact recovers $\boldsymbol{V}^{\mathtt{C}}, \boldsymbol{V}^{\mathtt{G}}$.

- Further step: Gram-Schmidt (GS) to obtain the orthogonal matrices:

$$\widehat{\boldsymbol{V}}^{\mathtt{C}} = \text{GS}([\widehat{\boldsymbol{v}}_1^{\mathtt{C}}, \dots, \widehat{\boldsymbol{v}}_{NM}^{\mathtt{C}}]), \ \widehat{\boldsymbol{V}}^{\mathtt{G}} = \text{GS}([\widehat{\boldsymbol{v}}_1^{\mathtt{G}}, \dots, \widehat{\boldsymbol{v}}_{NM}^{\mathtt{G}}])$$

[3] [Van Loan and Pitsianis, 1993] *Nearest Kronecker product decomposition*

Simplified Solution by Unfolding to step 1

Recover $\boldsymbol{V}^{\mathtt{G}}, \boldsymbol{V}^{\mathtt{C}}$ by unfolding[4] $\boldsymbol{y} = (\boldsymbol{y}_1; \cdots; \boldsymbol{y}_M) \in \mathbb{R}^{NM}$:

- Set $\boldsymbol{Y}^{(s)} = [\boldsymbol{y}_1^{(s)}, \ldots, \boldsymbol{y}_M^{(s)}] \in \mathbb{R}^{N \times M}$
- $\boldsymbol{C}_y^{\text{layer}} = \mathbb{E}[(\boldsymbol{Y}^{(s)})^\top \boldsymbol{Y}^{(s)}]$, $\boldsymbol{C}_y^{\text{node}} = \mathbb{E}[\boldsymbol{Y}^{(s)} (\boldsymbol{Y}^{(s)})^\top]$
- Estimate $\widetilde{\boldsymbol{V}}^{\mathtt{C}} = \text{EVD}(\boldsymbol{C}_y^{\text{layer}})$, $\widetilde{\boldsymbol{V}}^{\mathtt{G}} = \text{EVD}(\boldsymbol{C}_y^{\text{node}})$

Issues for unfolding:

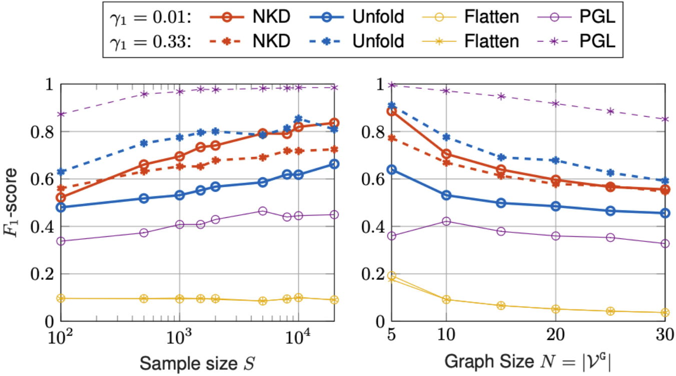
- $\widetilde{\boldsymbol{V}}^{\mathtt{G}}, \widetilde{\boldsymbol{V}}^{\mathtt{C}} \to \boldsymbol{V}^{\mathtt{G}}, \boldsymbol{V}^{\mathtt{C}}$ if eigenvalues of $\boldsymbol{C}_y^{\text{layer}}, \boldsymbol{C}_y^{\text{node}}$ are distinct, and

$$\boldsymbol{C}_y^{\text{node}} = \sum_{j=1}^{N} \boldsymbol{v}_j^{\mathtt{G}} (\boldsymbol{v}_j^{\mathtt{G}})^\top \sum_{i=1}^{M} |h(\lambda_i^{\mathtt{C}}, \lambda_j^{\mathtt{G}})|^2$$

  \- **repeated eigenvalues** are common even under **A1**.
- Sufficient condition for exact recovery: **separable filter** with $h(\lambda^{\mathtt{C}}, \lambda^{\mathtt{G}}) = h^{\mathtt{C}}(\lambda^{\mathtt{C}}) h^{\mathtt{G}}(\lambda^{\mathtt{G}})$

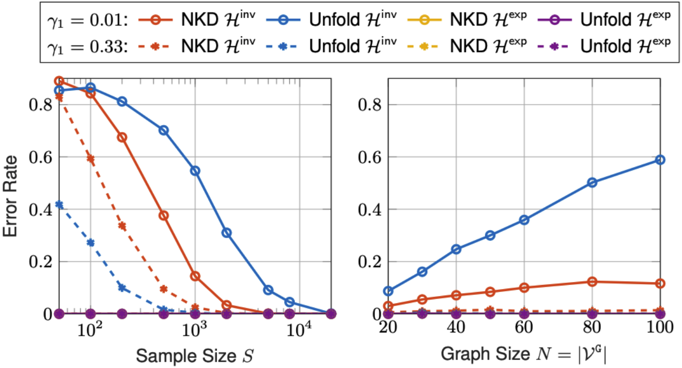
[4] Inspired from [Sandryhaila and Moura, 2014, Zhang et al., 2021], also used recently in [Kadambari and Chepuri, 2021, Einizade and Sardouie, 2022].

Experiment 1: Topology Reconstruction



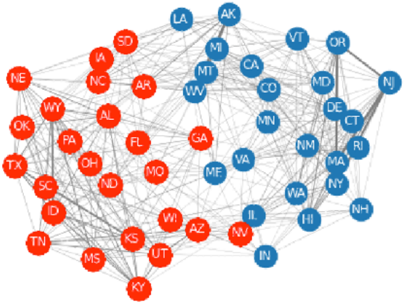
- $\boldsymbol{A} = \gamma_1 \boldsymbol{I} \otimes \boldsymbol{A}^{\mathtt{G}} + 2\gamma_1 \boldsymbol{A}^{\mathtt{C}} \otimes \boldsymbol{I} + (1 - 3\gamma_1)\boldsymbol{A}^{\mathtt{C}} \otimes \boldsymbol{A}^{\mathtt{G}}$
- $\gamma_1 \downarrow 0 \rightarrow$ stronger **inter-layer coupling** of graph.
- 'NKD' recovers topology effectively regardless of γ_1 **robustly**.
- 'Unfold' & 'PGL' **sensitive** to γ_1; 'Flatten' fails consistently.

Experiment 2: Central Nodes Detection

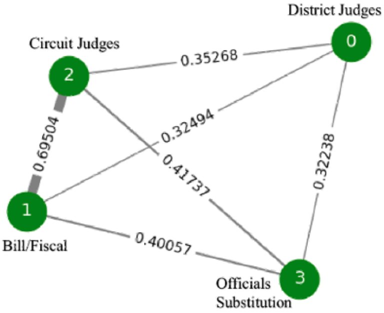


- $\mathcal{H}^{exp}$ works well for both 'NKD' and 'Unfold'.
- 'NKD' achieves lower error rate with fewer samples in strong coupling ($\gamma_1 = 0.01$).
- 'Unfold' method requires large samples with close eigenvalues ($\mathcal{H}^{inv}, \gamma_1 = 0.01$).

Real Data: US Senate Roll Calls



Estimated Senate topology G^G



Estimated coupling graph of topics G^C

- State graph $\mathcal{G}^{\mathtt{G}}$ shows two clusters (Republican&Democratic).
- Coupling graph $\mathcal{G}^{\mathtt{C}}$ reveals strong connection between 'circuit judges' & 'fiscal' topics.

Conclusion

- Generalize product graph filter for multi-attribute signals.
- Develop inference algorithms for topology learning & centrality detection.
- Study layer/node-wise unfolding and NKD for spectral estimation leading to graph inference.
- Product graph learning based on NKD delivers more robust performance.

Thank you!

References

[Einizade and Sardouie, 2022] Einizade, A. and Sardouie, S. H. (2022).
Learning product graphs from spectral templates.
arXiv preprint arXiv:2211.02893.

[Hanteer and Rossi, 2019] Hanteer, O. and Rossi, L. (2019).
An innovative way to model twitter topic-driven interactions using multiplex networks.
Frontiers in big data, 2:9.

[He and Wai, 2022] He, Y. and Wai, H.-T. (2022).
Detecting central nodes from low-rank excited graph signals via structured factor analysis.
IEEE Transactions on Signal Processing.

[Kadambari and Chepuri, 2021] Kadambari, S. K. and Chepuri, S. P. (2021).
Product graph learning from multi-domain data with sparsity and rank constraints.
IEEE Transactions on Signal Processing, 69:5665–5680.

[Roddenberry and Segarra, 2021] Roddenberry, T. M. and Segarra, S. (2021).
Blind inference of eigenvector centrality rankings.
IEEE Transactions on Signal Processing, 69:3935–3946.

[Sandryhaila and Moura, 2014] Sandryhaila, A. and Moura, J. M. (2014).
Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure.
IEEE signal processing magazine, 31(5):80–90.

[Segarra et al., 2017] Segarra, S., Marques, A. G., Mateos, G., and Ribeiro, A. (2017).
Network topology inference from spectral templates.
IEEE Transactions on Signal and Information Processing over Networks, 3(3):467–483.

[Van Loan and Pitsianis, 1993] Van Loan, C. F. and Pitsianis, N. (1993).
Approximation with kronecker products.
In *Linear algebra for large scale and real-time applications*, pages 293–314. Springer.

[Zhang et al., 2021] Zhang, S., Deng, Q., and Ding, Z. (2021).
Graph signal processing over multilayer networks–part i: Foundations and spectrum analysis.
arXiv preprint arXiv:2108.13638.

[Zhao et al., 2016] Zhao, B., Hu, S., Li, X., Zhang, F., Tian, Q., and Ni, W. (2016).
An efficient method for protein function annotation based on multilayer protein networks.
Human genomics, 10:1–15.