Multivariate Analysis of Large-Scale Network Series

Michael Weylandt* and George Michailidis

University of Florida
Informatics Institute & Department of Statistics

Objectives

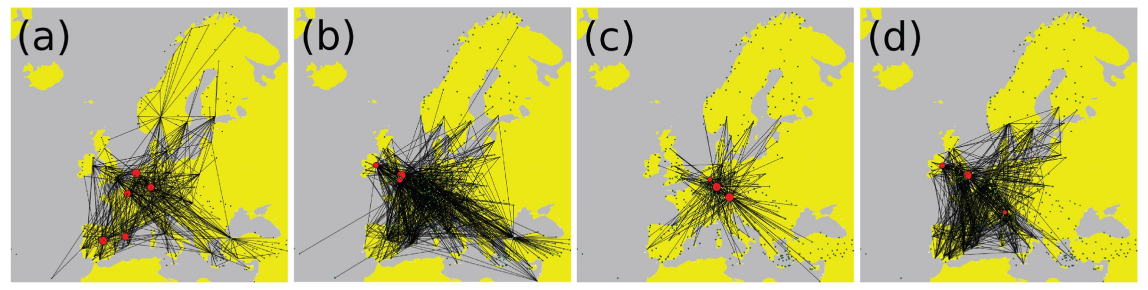
Principal Components Analysis of Network Data:

- Preserve Network Structure
- Computational AND Statistical Efficiency
- Flexibility - Capture Arbitrary Low-Rank Factors
- Nestability - Capture Multiple Principal Components

Multiple Network Data

Applications:

- Neuroscience (Zhang *et al.*, *NeuroImage*, 2019)
- Social Dynamics (Eagle *et al.*, *PNAS*, 2009)
- International Development (Hafner-Burton *et al.*, *International Organization*, 2009)
- Transportation (Cardillo *et al.*, *Sci. Reports*, 2013)



Network Series: ordered set of networks on the same nodes
Special Case of "Multilayer Networks" (Kivelä *et al.*, *J. Complex Networks*, 2014)

Related Work

Clustering:

- Sundar *et al.*, *NeurIPS*, 2017; Mantziou *et al.* (2022+); Signorelli and Wit, *Stat. Mod.*, 2020

Generative Modeling:

- Crane, *Bernoulli*, 2015; Crane, *AoAP*, 2016; Gollini *JCGS*, 2016; Durante *et al.*, *JASA*, 2017

Two-Sample Testing:

- Ginestet *et al.*, *AoAS*, 2017

Scalar-on-Network Regression:

- Relión *et al.*, *AoAS*, 2019; Guha and Rodriguez, *JASA*, 2021

Time Series Models:

- Hanneke *et al.*, *EJS*, 2010; Chen and Chen, 2019+

Joint Embeddings:

- Wang *et al.*, *PAMI*, 2021

Tensor Factorizations - Statistics:

- Sun *et al.*, *JRSS-B*, 2017; Anandkumar *et al.*, *JMLR*, 2017; Wang *et al.*, *PAMI*, 2021

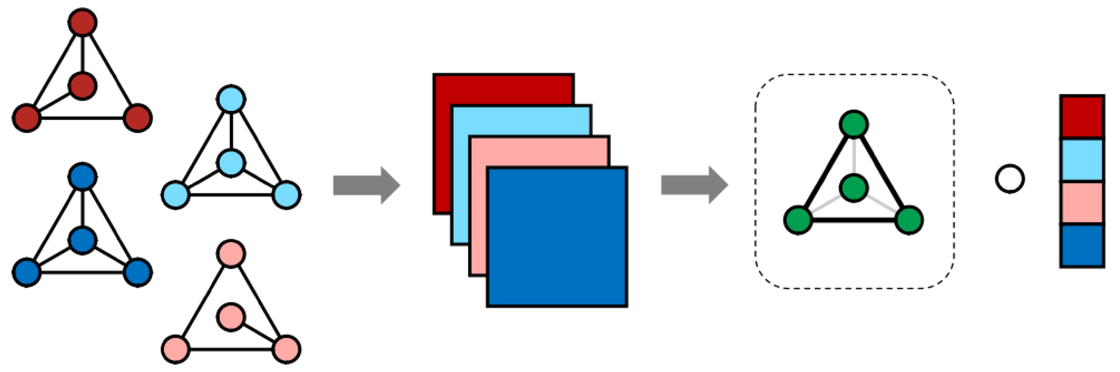
Tensor Factorizations - Applied Math:

- Sorensen and De Lathauwer, *SIMAX*, 2015
  - $(L_r, L_r, 1)$-Multilinear Rank Decomposition

Network PCA via Tensor Decompositions

Numerical representation: given T networks on p vertices each:

- Identify edges for each network
- Create a $p \times p$ **adjacency matrix**
- Align into a $p \times p \times T$ **tensor**



Symmetric rank-r variant of CP decomposition

Algorithm

Alternating maximization: $\mathbf{u}$ and $\mathbf{V}$-subproblems tractable!

$$\arg\min_{\mathbf{V},\mathbf{u},d} \|\mathcal{X} - d\,\mathbf{V} \circ \mathbf{V} \circ \mathbf{u}\|_F^2 \iff \arg\max_{\mathbf{V},\mathbf{u}} \langle \mathcal{X}, \mathbf{V} \circ \mathbf{V} \circ \mathbf{u} \rangle$$

Semi-Symmetric Tensor Power Method

- Initialize $\mathbf{u}_0$ to be random p-vector
- Repeat until convergence:
  - $\mathbf{V}_k = \text{leading } r\text{-eigenvectors}(\mathcal{X} \bar{\times}_3 \mathbf{u}_{k-1})$
  - $\mathbf{u}_k \propto \mathcal{X} \times_1 \mathbf{V}_k \times_2 \mathbf{V}_k$
- Return: Principal Matrix $\mathbf{V}_\infty \circ \mathbf{V}_\infty$ and Loadings $\mathbf{u}_\infty$

Extension of **power method** for eigenvalue calculations

Advantages: Fast; Streaming, Big-data, Sparse *etc.*
Disadvantages: Non-Convex; Mildly Sensitive to Initialization

Consistency of Semi-Symmetric Tensor PCA

Let $\mathcal{X} = d\,\mathbf{V}^* \circ \mathbf{V}^* \circ \mathbf{u}_* + \mathcal{E}$ for σ-sub-Gaussian $\mathcal{E}$. Then, with good initialization, the semi-symmetric tensor power method applied to $\mathcal{X}$ recovers $\mathbf{u}_*$ and $\mathbf{V}_*$ at the same rates as classical PCA with high probability:

$$\min_{\mathbf{O} \in \mathcal{V}^{k \times k}} \frac{\|\mathbf{V}^* - \hat{\mathbf{V}}\mathbf{O}\|_2}{\sqrt{pr}} \lesssim \frac{\sigma r \sqrt{T}}{d} \quad \text{and} \quad \min_{\epsilon \in \{\pm 1\}} \frac{\|\mathbf{u}^* - \hat{\mathbf{u}}\epsilon\|_2}{\sqrt{T}} \lesssim \frac{\sigma r \sqrt{p}}{d}$$

Furthermore the statistical convergence is linear (fast) before hitting the "noise barrier."

Proof Outline

Tools: **Davis-Kahan theorem + Iteration**

V-update:

$$\|\sin\angle(\mathbf{V}^*, \mathbf{V}^{(k+1)})\|_F \leq 2\left|1 - \cos\angle(\mathbf{u}^{(k+1)}, \mathbf{u}_*)\right| + \frac{2\,\|\mathcal{E}\|_{r\text{-op}}}{d}$$

u-update:

$$|\sin\angle(\mathbf{u}_*, \mathbf{u}^{(k+1)})| \leq 2\left|1 - \cos\angle(\mathbf{V}_*, \mathbf{V}^{(k)})^4\right| + \frac{8r^2\,\|\mathcal{E}\|_{r\text{-op}}}{d}$$

For small angles $2|1 - \cos\theta| < |\sin\theta| \Leftarrow$ **initialization!**

Chained iteration shows:

$$\text{Error at Iteration } k \approx c^k E_1 + E_2/(1-c)$$

where:

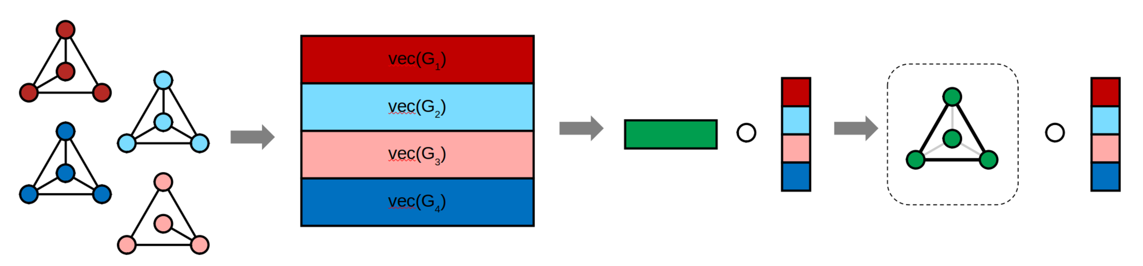
- E_1 is initialization error (depends only on $\angle(\mathbf{u}_0, \mathbf{u}_*)$)
- E_2 is stochastic error (depends on noise $\|\mathcal{E}\|_{r\text{-op}}$ & signal d)
- $c < 1$ depends on initialization quality

Implications:

- Statistical consistency
- Geometric convergence to "noise range"
- Possibly slow (or looping) after that

Similar results for sparse regression by Fan *et al.* (*AoS*, 2018)
All this despite non-convexity: analyze *algorithm* not *problem*!

Comparison with Naive PCA



Does not enforce rank-r structure on "principal network"

SS-TPCA is equivalent to

$$\arg\max_{\mathbf{u},\mathbf{v}} \mathbf{u}^T \mathcal{M}_3(\mathcal{X})\mathbf{v} \text{ such that } \text{rank}(\text{unvec}(\mathbf{v})) = r$$

Variant of Truncated Power Method for Sparse PCA (Yuan and Zhang, *JMLR* 2013) with "unvec-rank" instead of sparsity:

Method	Dimension	u-MSE	v-MSE
Classical PCA	$T \times p$	$\frac{\sigma\sqrt{p}}{d}$	$\frac{\sigma\sqrt{T}}{d}$
Vectorize + PCA	$T \times \binom{p}{2}$	$\frac{\sigma p}{d}$	$\frac{\sigma\sqrt{T}}{d}$
SS-TPCA	$p \times p \times T$	$\frac{\sigma r\sqrt{p}}{d}$	$\frac{\sigma r\sqrt{T}}{d}$

- Same rate as classical PCA (when $r = 1$)
- Better than naïve (vectorization) by factor of $\sqrt{p} \gg r$

Connection to "unvec-rank" constrained PCA highlights key role of Davis-Kahan in theoretical analysis

Application: SCOTUS Voting

Each term SCOTUS decides ≈ 80 cases:

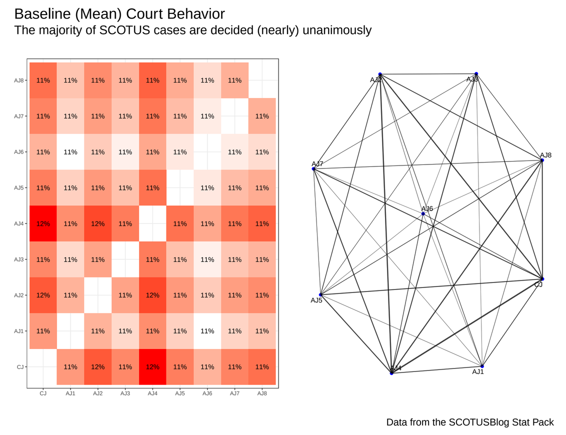
- **Weighted**, **undirected** network based on co-voting
- By "seat" (AJ7 = Ginsburg = Barrett), not by Justice

Data: SCOTUSblog annual "stat pack" - OT 1995 to OT 2020

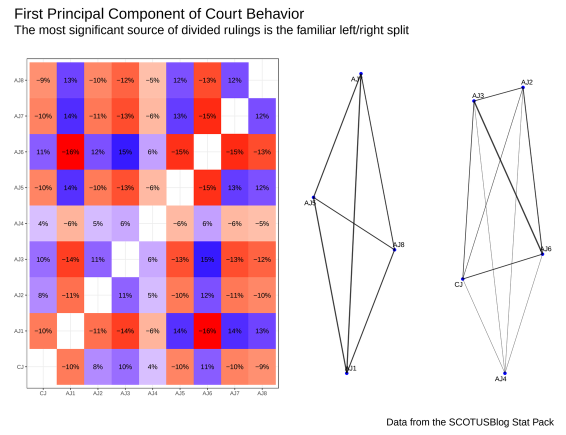
$$9 \times 9 \text{ pairs} \times 25 \text{ terms} \equiv \mathcal{X} \in \mathbb{R}^{9 \times 9 \times 25}$$

Semi-Symmetric PCA as a Flexible Pattern Recognition Tool:

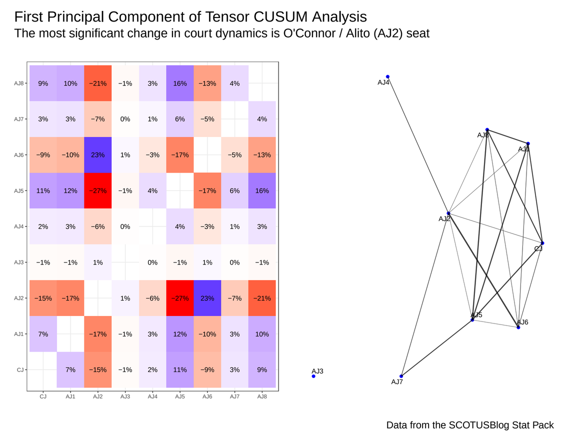
- Raw $\mathcal{X}$ - major patterns (trends)



- Centered $\mathcal{X}$ - variance components (covariance patterns)



- Differenced $\mathcal{X}$ - change-point identification (CUSUM)



Acknowledgements

MW's research is supported by an appointment to the Intelligence Community Postdoctoral Research Fellowship Program at the University of Florida Informatics Institute, administered by Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and the Office of the Director of National Intelligence. GM's research is supported by NSF/DMS grant 1821220. The authors also thank Genevera I. Allen for helpful discussions on the tensor power method.

More Information

- ArXiv: 2202.04719
- Web: https://michaelweylandt.github.io/
- Email: michael.weylandt@ufl.edu