Multivariate Analysis of Large-Scale Network Series

Michael Weylandt

2022-05-09

Department of Mathematics
University of Houston
Houston, TX USA

michael.weylandt@ufl.edu
https://michaelweylandt.github.io/
IC Postdoctoral Research Fellow
University of Florida Informatics Institute

Network Data

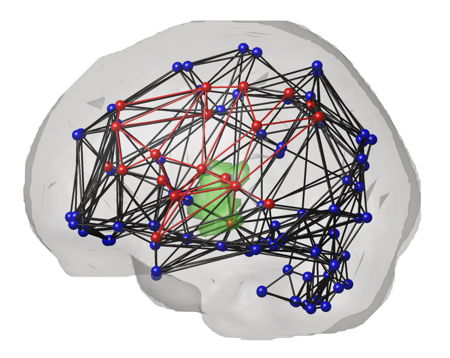
Ubiquitous Network Data: telecommunications, social media, neuroscience, sensor networks, transportation, *etc.*

Network Data

Ubiquitous Network Data: telecommunications, social media, neuroscience, sensor networks, transportation, *etc.*

Three types of network data:

- Networks as models of complex phenomena ("graphical models")



Network Data

Ubiquitous Network Data: telecommunications, social media, neuroscience, sensor networks, transportation, *etc.*

Three types of network data:

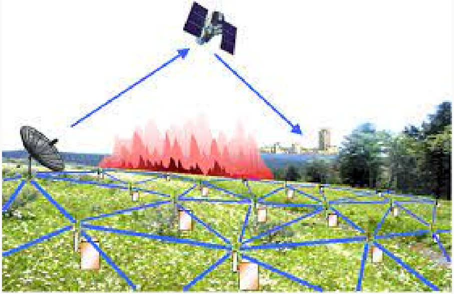
- Networks as models of complex phenomena ("graphical models")
- Observed network(s) ("network science")

Network Data

Ubiquitous Network Data: telecommunications, social media, neuroscience, sensor networks, transportation, *etc.*

Three types of network data:

- Networks as models of complex phenomena ("graphical models")
- Observed network(s) ("network science")
- Data observed on a network ("graph signal processing")



Network Data

Ubiquitous Network Data: telecommunications, social media, neuroscience, sensor networks, transportation, *etc.*

Three types of network data:

- Networks as models of complex phenomena ("graphical models")
- Observed network(s) ("network science")
- Data observed on a network ("graph signal processing")

Today: contributions to **network science** on multiple networks

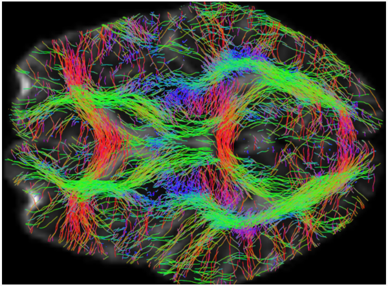
Multiple Network Data

Applications:

Multiple Network Data

Applications:

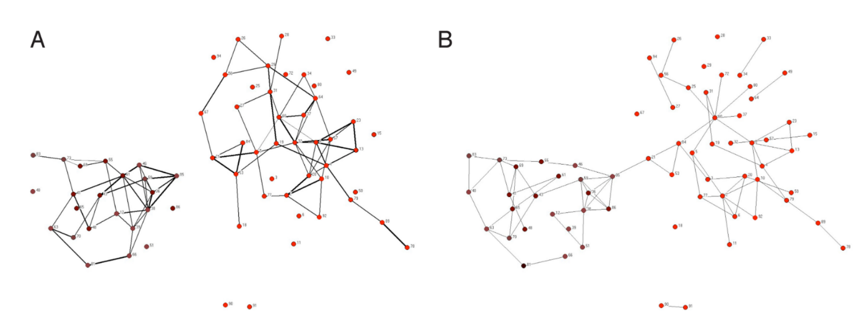
- Neuroscience (Zhang *et al.*, *NeuroImage*, 2019)



Multiple Network Data

Applications:

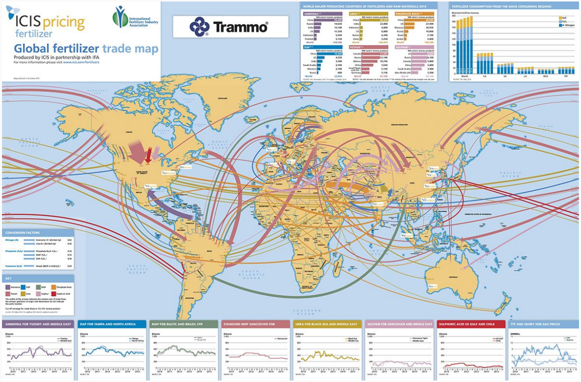
- Neuroscience (Zhang *et al.*, *NeuroImage*, 2019)
- Social Dynamics (Eagle *et al.*, *PNAS*, 2009)



Multiple Network Data

Applications:

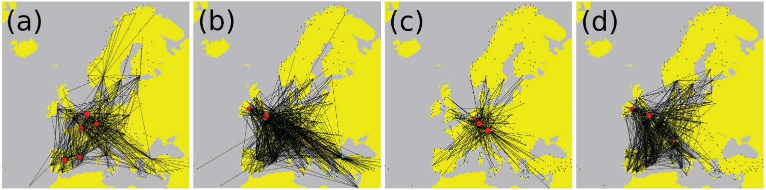
- Neuroscience (Zhang *et al.*, *NeuroImage*, 2019)
- Social Dynamics (Eagle *et al.*, *PNAS*, 2009)
- International Development (Hafner-Burton *et al.*, *International Organization*, 2009)



Multiple Network Data

Applications:

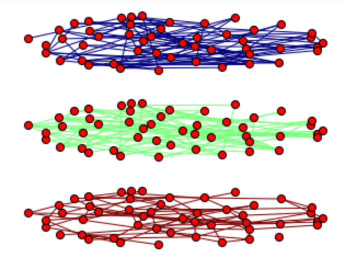
- Neuroscience (Zhang *et al.*, *NeuroImage*, 2019)
- Social Dynamics (Eagle *et al.*, *PNAS*, 2009)
- International Development (Hafner-Burton *et al.*, *International Organization*, 2009)
- Transportation (Cardillo *et al.*, *Sci. Reports*, 2013)



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: an ordered set of network observed on the same nodes

Special Case of "Multilayer Networks" (Kivelä *et al.*, *J. Complex Networks*, 2014)

Selected Related Work

Clustering:

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

Selected 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

Selected 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

Selected 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

Selected 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+

Selected 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

Multivariate Methods for Multiple Networks

Acknowledgements

Joint work with George Michailidis (U Florida)

Acknowledgements

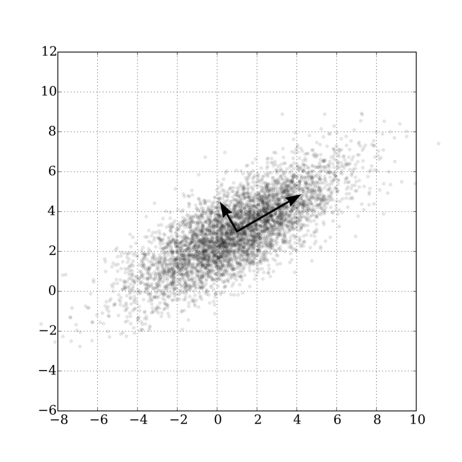
Joint work with George Michailidis (U Florida)

IC Advisors: Joe McCloskey (NCSC) + Steve H (GCHQ)

Principal Components Analysis

Principal Components Analysis:

- Exploratory Data Analysis
- Pattern Recognition
- Dimension Reduction
- Data Visualization



Principal Components Analysis

Data matrix: $\mathsf{X} \in \mathbb{R}^{n \times p}$

- n observations (rows)
- p features (columns)

Principal Components Analysis

Data matrix: $X \in \mathbb{R}^{n \times p}$

- n observations (rows)
- p features (columns)

$$X \approx d\,u\,v^T$$

Decompose X into a major pattern v and how much the pattern contributes to each observation u

Principal Components Analysis

Data matrix: $X \in \mathbb{R}^{n \times p}$

- n observations (rows)
- p features (columns)

$$X \approx d\,u\,v^T$$

Decompose X into a major pattern v and how much the pattern contributes to each observation u

All-purpose pattern recognition tool:

- Raw X - major patterns (trends)
- Centered X - variance components (covariance patterns)
- Differenced X - change-point identification (CUSUM analysis)

Tensor Analysis

Goal: Apply PCA to **multiple** networks

Tensor Analysis

Goal: Apply PCA to **multiple** networks

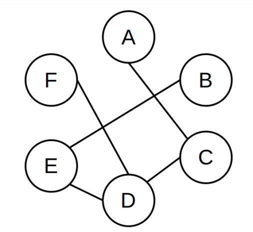
Numerical representation: given T networks on p vertices each:

Tensor Analysis

Goal: Apply PCA to **multiple** networks

Numerical representation: given T networks on p vertices each:

- Identify edges for each network

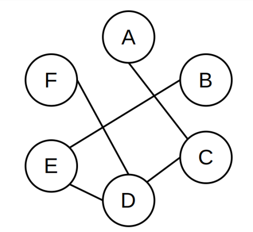


Tensor Analysis

Goal: Apply PCA to **multiple** networks

Numerical representation: given T networks on p vertices each:

- Identify edges for each network
- Create a $p \times p$ **adjacency matrix**



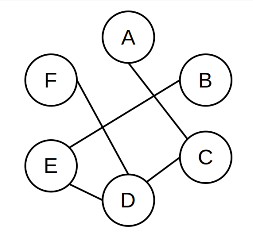
	A	B	C	D	E	F
A	0	0	1	0	0	0
B	0	0	0	0	1	0
C	1	0	0	1	1	0
D	0	0	1	0	1	1
E	0	1	0	1	0	0
F	0	0	0	1	0	0

Tensor Analysis

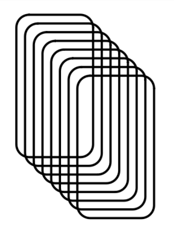
Goal: Apply PCA to **multiple** networks

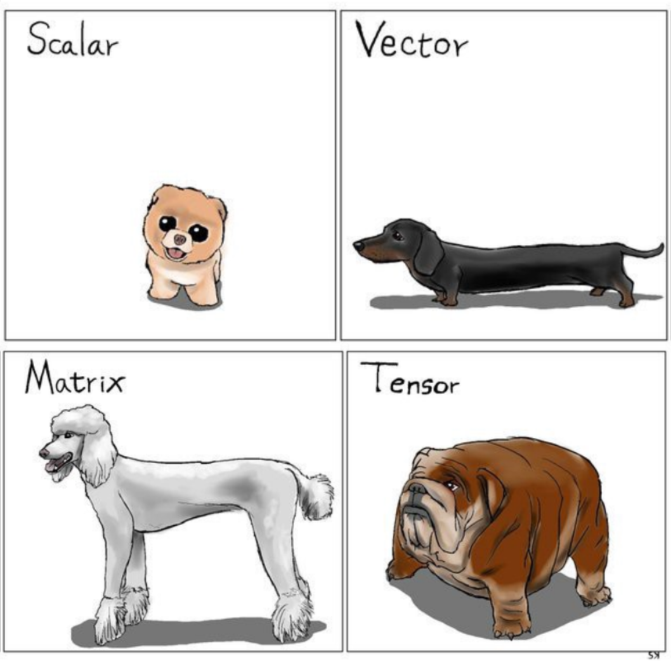
Numerical representation: given T networks on p vertices each:

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



	A	B	C	D	E	F
A	0	0	1	0	0	0
B	0	0	0	0	1	0
C	1	0	0	1	1	0
D	0	0	1	0	1	1
E	0	1	0	1	0	0
F	0	0	0	1	0	0




Scalar
Vector
Matrix
Tensor

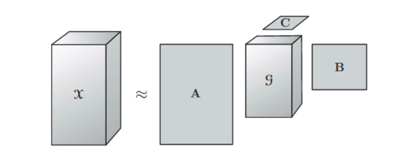
Tensor Decompositions

Given $p \times p \times T$ tensor, how to break it into "principal components"?

Tensor Decompositions

Given $p \times p \times T$ tensor, how to break it into "principal components"?

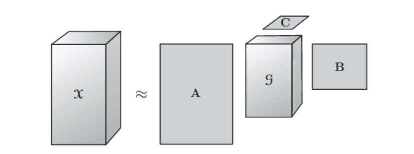
- Tucker Factors: Optimal for compression; limited interpretation



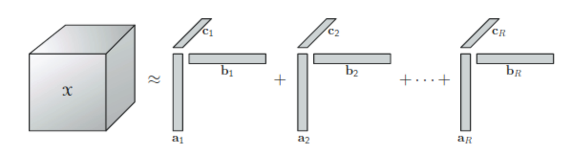
Tensor Decompositions

Given $p \times p \times T$ tensor, how to break it into "principal components"?

- Tucker Factors: Optimal for compression; limited interpretation



- CP Factors: Method of choice for PCA; (orthogonal) distinct factors

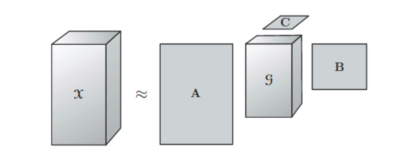


Regularized variants available (Allen, *AISTATS*, 2012)

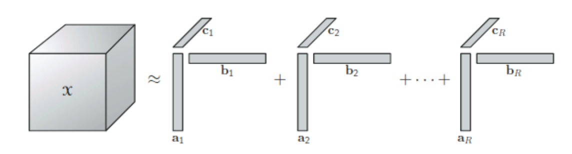
Tensor Decompositions

Given $p \times p \times T$ tensor, how to break it into "principal components"?

- Tucker Factors: Optimal for compression; limited interpretation



- CP Factors: Method of choice for PCA; (orthogonal) distinct factors



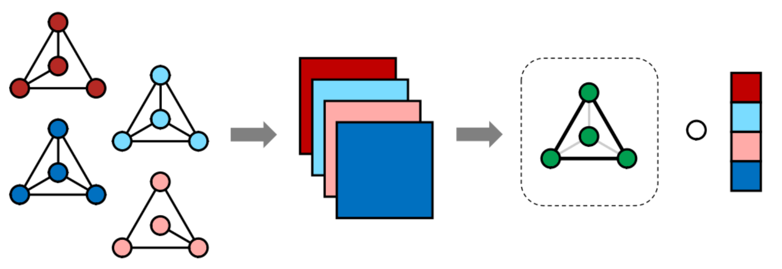
Regularized variants available (Allen, *AISTATS*, 2012)

Both well-studied: neither uses the network structure of our tensor

Semi-Symmetric Tensor Decomposition Needed

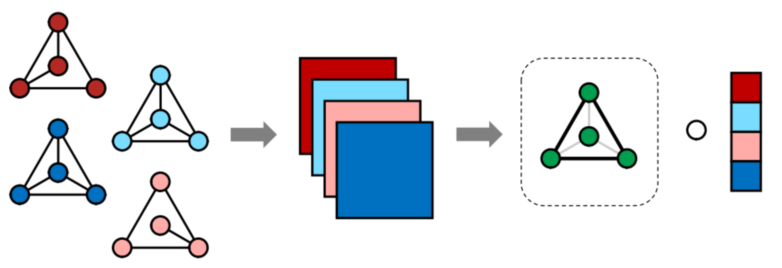
Semi-Symmetric Tensor Decomposition

Given a series of networks, arranged as a $p \times p \times T$ tensor, decompose into a "principal network" and loading vector



Semi-Symmetric Tensor Decomposition

Given a series of networks, arranged as a $p \times p \times T$ tensor, decompose into a "principal network" and loading vector



Goals:

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

Semi-Symmetric Tensor PCA

Semi-Symmetric Generalization of the CP decomposition:

$$\mathcal{X} \approx \sum_{i=1}^{k} d_i \, \mathsf{V}_i \circ \mathsf{V}_i \circ \mathsf{u}_i$$

where $\mathsf{V}_i \in \mathbb{R}^{p \times r_i}$ is orthogonal, $\mathsf{u}_i \in \mathbb{R}^{T}$.

$$(r_1, \dots, r_k)\text{-SS-TPCA}$$

Semi-Symmetric Tensor PCA

Semi-Symmetric Generalization of the CP decomposition:

$$\mathcal{X} \approx \sum_{i=1}^{k} d_i \, \mathsf{V}_i \circ \mathsf{V}_i \circ \mathsf{u}_i$$

where $\mathsf{V}_i \in \mathbb{R}^{p \times r_i}$ is orthogonal, $\mathsf{u}_i \in \mathbb{R}^T$.

$$(r_1, \dots, r_k)\text{-SS-TPCA}$$

Between classical CP and Tucker:

- Allows for components of rank ≥ 1 (RDPG: Athreya *et al.*, *JMLR* 2018)
- Still has a diagonal core (viewed as Tucker)
- Restriction of a very flexible CP: repeated u factors

Semi-Symmetric Tensor PCA

Semi-Symmetric Generalization of the CP decomposition:

$$\mathcal{X} \approx \sum_{i=1}^{k} d_i \, \mathrm{V}_i \circ \mathrm{V}_i \circ \mathrm{u}_i$$

where $\mathrm{V}_i \in \mathbb{R}^{p \times r_i}$ is orthogonal, $\mathrm{u}_i \in \mathbb{R}^T$.

$$(r_1, \dots, r_k)\text{-SS-TPCA}$$

Between classical CP and Tucker:

- Allows for components of rank ≥ 1 (RDPG: Athreya *et al.*, *JMLR* 2018)
- Still has a diagonal core (viewed as Tucker)
- Restriction of a very flexible CP: repeated u factors

Rank-1 model applied in multi-subject neuroimaging to find PC factors correlated with behavioral traits (Zhang *et al.*, *NeuroImage* 2019)

Computing the SS-TPCA

Tensor factors are, in general, NP-Hard to compute (Hillar and Lim, *J. ACM*, 2013)

Computing the SS-TPCA

Tensor factors are, in general, NP-Hard to compute (Hillar and Lim, *J. ACM*, 2013)

We define our estimator *computationally* rather than by optimality

Computing the SS-TPCA

Tensor factors are, in general, NP-Hard to compute (Hillar and Lim, *J. ACM*, 2013)

We define our estimator *computationally* rather than by optimality

Best approximation in Frobenius (ℓ_2) norm:

$$\underset{\mathsf{V},\mathsf{u},d}{\arg\min} \|\mathcal{X} - d\,\mathsf{V} \circ \mathsf{V} \circ \mathsf{u}\|_F^2 \iff \underset{\mathsf{V},\mathsf{u}}{\arg\max} \langle \mathcal{X}, \mathsf{V} \circ \mathsf{V} \circ \mathsf{u} \rangle$$

Computing the SS-TPCA

Tensor factors are, in general, NP-Hard to compute (Hillar and Lim, *J. ACM*, 2013)

We define our estimator *computationally* rather than by optimality

Best approximation in Frobenius (ℓ_2) norm:

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

Alternating maximization approach: u and V subproblems are tractable!

Computing the SS-TPCA

Semi-Symmetric Tensor Decomposition (Rank-k Case)

- Initialize u_0 to be random p-vector
- Repeat until convergence:
  - $\mathrm{V}_k = \text{leading } r\text{-eigenvectors}(\mathcal{X} \bar{\times}_3 \mathrm{u}_{k-1})$
  - $\mathrm{u}_k \propto \mathcal{X} \times_1 \mathrm{V}_k \times_2 \mathrm{V}_k$
- Return: Principal Matrix $\mathrm{V}_\infty \circ \mathrm{V}_\infty$ and Loading Vector u_∞

Computing the SS-TPCA

Semi-Symmetric Tensor Decomposition (Rank-k Case)

- Initialize u_0 to be random p-vector
- Repeat until convergence:
  - $\mathrm{V}_k = \text{leading } r\text{-eigenvectors}(\mathcal{X} \bar{\times}_3 \mathrm{u}_{k-1})$
  - $\mathrm{u}_k \propto \mathcal{X} \times_1 \mathrm{V}_k \times_2 \mathrm{V}_k$
- Return: Principal Matrix $\mathrm{V}_\infty \circ \mathrm{V}_\infty$ and Loading Vector u_∞

Extension of **power method** for eigenvalue calculations

Computing the SS-TPCA

Semi-Symmetric Tensor Decomposition (Rank-k Case)

- Initialize u_0 to be random p-vector
- Repeat until convergence:
  - $\mathrm{V}_k = \text{leading } r\text{-eigenvectors}(\mathcal{X} \bar{\times}_3 \mathrm{u}_{k-1})$
  - $\mathrm{u}_k \propto \mathcal{X} \times_1 \mathrm{V}_k \times_2 \mathrm{V}_k$
- Return: Principal Matrix $\mathrm{V}_\infty \circ \mathrm{V}_\infty$ and Loading Vector u_∞

Extension of **power method** for eigenvalue calculations

Advantages:

- Fast
- Adaptable to streaming, big-data, sparse networks *etc.* in usual ways
- V_k-Update takes advantage of semi-symmetric structure

Computing the SS-TPCA

Semi-Symmetric Tensor Decomposition (Rank-k Case)

- Initialize u_0 to be random p-vector
- Repeat until convergence:
  - $\mathrm{V}_k = \text{leading } r\text{-eigenvectors}(\mathcal{X} \bar{\times}_3 \mathrm{u}_{k-1})$
  - $\mathrm{u}_k \propto \mathcal{X} \times_1 \mathrm{V}_k \times_2 \mathrm{V}_k$
- Return: Principal Matrix $\mathrm{V}_\infty \circ \mathrm{V}_\infty$ and Loading Vector u_∞

Extension of **power method** for eigenvalue calculations

Advantages:

- Fast
- Adaptable to streaming, big-data, sparse networks *etc.* in usual ways
- V_k-Update takes advantage of semi-symmetric structure

Disadvantages:

- Non-Convex
- Mildly Sensitive to Initialization

Statistical Consistency

"Spiked Covariance / Low-Rank Mean" Model:

$$\mathcal{X} = d\,\mathsf{V}_* \circ \mathsf{V}_* \circ \mathsf{u}_* + \mathcal{E}$$

Does the Semi-Symmetric Tensor Power Method recover u_* and V_*?

Statistical properties: do our estimates reflect the population model?
(Not interested in optimally (over-)fitting the observed data)

Statistical Consistency

"Spiked Covariance / Low-Rank Mean" Model:

$$\mathcal{X} = d\,\mathsf{V}_* \circ \mathsf{V}_* \circ \mathsf{u}_* + \mathcal{E}$$

Does the Semi-Symmetric Tensor Power Method recover u_* and V_*?

Statistical properties: do our estimates reflect the population model?
(Not interested in optimally (over-)fitting the observed data)

Challenges:

Statistical Consistency

"Spiked Covariance / Low-Rank Mean" Model:

$$\mathcal{X} = d\,\mathsf{V}_* \circ \mathsf{V}_* \circ \mathsf{u}_* + \mathcal{E}$$

Does the Semi-Symmetric Tensor Power Method recover u_* and V_*?

Statistical properties: do our estimates reflect the population model?
(Not interested in optimally (over-)fitting the observed data)

Challenges:

- Non-convex problem
    - Initialization-dependent solution
    - Algorithm not guaranteed to reach *global* optimum

Statistical Consistency

"Spiked Covariance / Low-Rank Mean" Model:

$$\mathcal{X} = d\,\mathsf{V}_* \circ \mathsf{V}_* \circ \mathsf{u}_* + \mathcal{E}$$

Does the Semi-Symmetric Tensor Power Method recover u_* and V_*?

Statistical properties: do our estimates reflect the population model?
(Not interested in optimally (over-)fitting the observed data)

Challenges:

- Non-convex problem
  - Initialization-dependent solution
  - Algorithm not guaranteed to reach *global* optimum
- Limited theoretical tools:
  - Adversarial analysis of noisy power method (Hardt and Price, *NeurIPS*, 2014)
  - Davis-Kahan theorem (Yu *et al.*, *Biometrika*, 2015)
    - Control of $\hat{\mathsf{u}}, \hat{\mathsf{v}}$ from PCA on $\hat{\mathsf{X}} = \mathsf{X}_* + \mathsf{E}$ in terms of $\|\mathsf{E}\|_{\text{op}}$

Statistical Consistency

"Spiked Covariance / Low-Rank Mean" Model:

$$\mathcal{X} = d\,\mathsf{V}_* \circ \mathsf{V}_* \circ \mathsf{u}_* + \mathcal{E}$$

Does the Semi-Symmetric Tensor Power Method recover u_* and V_*?

Statistical properties: do our estimates reflect the population model?
(Not interested in optimally (over-)fitting the observed data)

Challenges:

- Non-convex problem
  - Initialization-dependent solution
  - Algorithm not guaranteed to reach *global* optimum
- Limited theoretical tools:
  - Adversarial analysis of noisy power method (Hardt and Price, *NeurIPS*, 2014)
  - Davis-Kahan theorem (Yu *et al.*, *Biometrika*, 2015)
    - Control of $\hat{\mathsf{u}}, \hat{\mathsf{v}}$ from PCA on $\hat{\mathsf{X}} = \mathsf{X}_* + \mathsf{E}$ in terms of $\|\mathsf{E}\|_{\text{op}}$
- Recovery up to orthogonal rotation

Selected Related Work

Statistics:

Selected Related Work

Statistics:

- Sun *et al.*, *JRSS-B*, 2017:
    - Statistical consistency of 3-way tensor decompositions without symmetry

Selected Related Work

Statistics:

- Sun *et al.*, *JRSS-B*, 2017:
  - Statistical consistency of 3-way tensor decompositions without symmetry
- Anandkumar *et al.*, *JMLR*, 2017:
  - Statistical consistency of fully-symmetric 3-way tensor decomposition

Selected Related Work

Statistics:

- Sun *et al.*, *JRSS-B*, 2017:
    - Statistical consistency of 3-way tensor decompositions without symmetry
- Anandkumar *et al.*, *JMLR*, 2017:
    - Statistical consistency of fully-symmetric 3-way tensor decomposition
- Wang *et al.*, *PAMI*, 2021:
    - Optimization convergence of a closely-related greedy algorithm
    - Asymptotic (Large Sample) Consistency for Erdő-Renyi graphs

Selected Related Work

Statistics:

- Sun *et al.*, *JRSS-B*, 2017:
  - Statistical consistency of 3-way tensor decompositions without symmetry
- Anandkumar *et al.*, *JMLR*, 2017:
  - Statistical consistency of fully-symmetric 3-way tensor decomposition
- Wang *et al.*, *PAMI*, 2021:
  - Optimization convergence of a closely-related greedy algorithm
  - Asymptotic (Large Sample) Consistency for Erdős-Renyi graphs

Applied Math:

Selected Related Work

Statistics:

- Sun *et al.*, *JRSS-B*, 2017:
  - Statistical consistency of 3-way tensor decompositions without symmetry
- Anandkumar *et al.*, *JMLR*, 2017:
  - Statistical consistency of fully-symmetric 3-way tensor decomposition
- Wang *et al.*, *PAMI*, 2021:
  - Optimization convergence of a closely-related greedy algorithm
  - Asymptotic (Large Sample) Consistency for Erdő-Renyi graphs

Applied Math:

- Sorensen and De Lathauwer, *SIMAX*, 2015
  - Considered our model as a $(L_r, L_r, 1)$-Multilinear Rank Decomposition
  - No "statistical" theory

Statistical Consistency

Theorem (Semi-Formal)

Suppose $\mathcal{X} = d\, \mathsf{V}_* \circ \mathsf{V}_* \circ \mathsf{u}_* + \mathcal{E}$ where

- u_* is a unit-norm T-vector
- V_* is a $p \times r$ orthogonal matrix satisfying $\mathsf{V}_*^T \mathsf{V}_* = \mathsf{I}_{r \times r}$
- $d \in \mathbb{R}_{>0}$ is a measure of signal strength
- $\mathcal{E}$ is a semi-symmetric noise tensor each free element of which is independently σ-sub-Gaussian.

With sufficiently good iteration, our Algorithm applied to $\mathcal{X}$ satisfies the following with high probability:

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

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

Proof Outline

Davis-Kahan theorem applied repeatedly + iteration:

V-update:

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

Proof Outline

Davis-Kahan theorem applied repeatedly + iteration:

V-update:

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

u-update:

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

Surprisingly tricky deal with normalization of u updates
$\implies$ apply DK to $\lambda\tilde{\mathrm{u}}_k \circ \tilde{\mathrm{u}}_k - \mathrm{u}_* \circ \mathrm{u}_*$ for suitable λ

Proof Outline

Davis-Kahan theorem applied repeatedly + iteration:

V-update:

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

u-update:

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

Surprisingly tricky deal with normalization of u updates
$\implies$ apply DK to $\lambda\tilde{\mathrm{u}}_k \circ \tilde{\mathrm{u}}_k - \mathrm{u}_* \circ \mathrm{u}_*$ for suitable λ

To combine these results, note that for small angles $2|1 - \cos\theta| < |\sin\theta|$

Assume we are in this range ($\approx [0, 53^\circ]$) *via* good initialization (actually a bit smaller to deal with noise)

Proof Outline

Davis-Kahan theorem applied repeatedly + iteration:

V-update:

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

u-update:

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

Surprisingly tricky deal with normalization of u updates
$\implies$ apply DK to $\lambda\tilde{\mathsf{u}}_k \circ \tilde{\mathsf{u}}_k - \mathsf{u}_* \circ \mathsf{u}_*$ for suitable λ

To combine these results, note that for small angles $2|1-\cos\theta| < |\sin\theta|$

Assume we are in this range ($\approx [0, 53^\circ]$) *via* good initialization (actually a bit smaller to deal with noise)

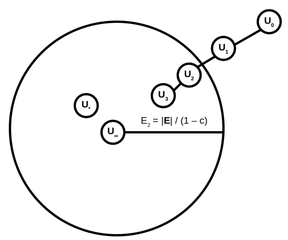
Chain + iterate to desired bound

Convergence Rate

More detailed work shows that:

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

where



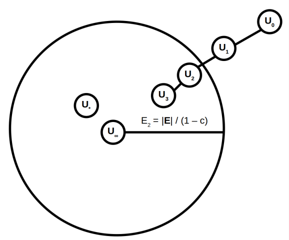
Convergence Rate

More detailed work shows that:

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

where

- E_1 is initialization error (depends only on $\angle(\mathrm{u}_0, \mathrm{u}_*)$)



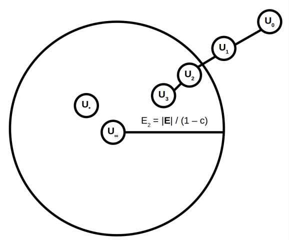
Convergence Rate

More detailed work shows that:

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

where

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



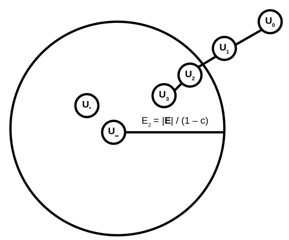
Convergence Rate

More detailed work shows that:

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

where

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



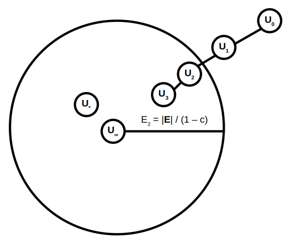
Convergence Rate

More detailed work shows that:

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

where

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



Implications:

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

Similar results obtained for sparse regression by Fan *et al.* (*AoS*, 2018)

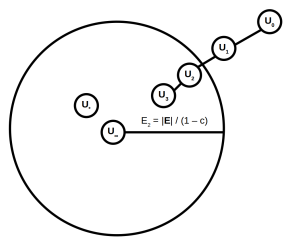
Convergence Rate

More detailed work shows that:

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

where

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



Implications:

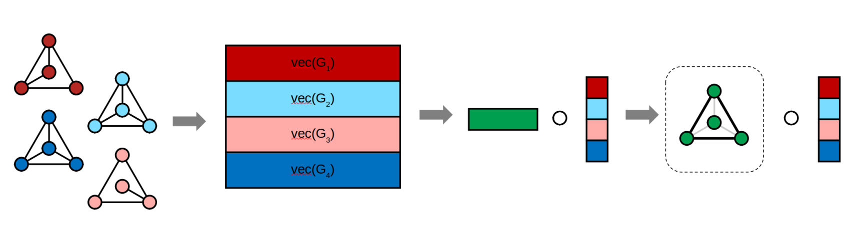
- Geometric convergence to "noise range"
- Possibly slow (or looping) after that

Similar results obtained for sparse regression by Fan *et al.* (*AoS*, 2018)

All this despite non-convexity: analyze *algorithm* not *problem*!

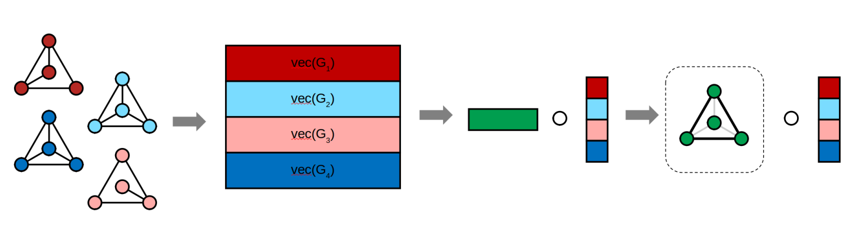
Is this just PCA?

Weighted networks + (real) inner product - perform a spectral decomposition in this space (Eaton, 1983)?



Is this just PCA?

Weighted networks + (real) inner product - perform a spectral decomposition in this space (Eaton, 1983)?



Does not enforce rank-r structure on "principal network:" SS-TPCA is equivalent to

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

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

Comparison with Classical PCA

Method	Dimension	u-MSE	v-MSE
Classical PCA	$T \times p$	$\frac{\sigma\sqrt{p}}{d}$	$\frac{\sigma\sqrt{T}}{d}$

$\mathcal{O}(\sqrt{\log T / T})$ terms omitted

Comparison with Classical PCA

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

$\mathcal{O}(\sqrt{\log T / T})$ terms omitted

Comparison with Classical PCA

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	$p \times p \times T$	$\frac{\sigma r\sqrt{p}}{d}$	$\frac{\sigma r\sqrt{T}}{d}$

$\mathcal{O}(\sqrt{\log T / T})$ terms omitted

Comparison with Classical PCA

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	$p \times p \times T$	$\frac{\sigma r\sqrt{p}}{d}$	$\frac{\sigma r\sqrt{T}}{d}$

Tensor approach:

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

$\mathcal{O}(\sqrt{\log T / T})$ terms omitted

Comparison with Classical PCA

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	$p \times p \times T$	$\frac{\sigma r\sqrt{p}}{d}$	$\frac{\sigma r\sqrt{T}}{d}$

Tensor approach:

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

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

$\mathcal{O}(\sqrt{\log T / T})$ terms omitted

Practicalities

Initialization:

- For many network problems, signal is roughly constant (or at least positive)
    - Stable initialization: $\mathrm{u}_0 = 1/\sqrt{T}$
    - Performs particularly well for trend-finding and change-point
- Random (re-)initialization works well for hardest problems
- See simulations in paper

Practicalities

Initialization:

- For many network problems, signal is roughly constant (or at least positive)
  - Stable initialization: $\mathrm{u}_0 = 1/\sqrt{T}$
  - Performs particularly well for trend-finding and change-point
- Random (re-)initialization works well for hardest problems
- See simulations in paper

Deflation: $\mathcal{X}^t \Leftarrow \mathcal{X}^{t-1} \times_1 [\mathrm{I}_p - \hat{\mathrm{V}} \circ \hat{\mathrm{V}}] \times_2 [\mathrm{I}_p - \hat{\mathrm{V}} \circ \hat{\mathrm{V}}] \times_3 [\mathrm{I} - \hat{\mathrm{u}}\hat{\mathrm{u}}^T]$

- Removes estimated signal: $\mathcal{X}^t \times_1 \mathrm{V}_t \circ \mathrm{V}_t = \mathcal{X}^t \bar{\times}_3 \mathrm{u}_t = 0$

Practicalities

Initialization:

- For many network problems, signal is roughly constant (or at least positive)
  - Stable initialization: $\mathrm{u}_0 = 1/\sqrt{T}$
  - Performs particularly well for trend-finding and change-point
- Random (re-)initialization works well for hardest problems
- See simulations in paper

Deflation: $\mathcal{X}^t \Leftarrow \mathcal{X}^{t-1} \times_1 [\mathrm{I}_p - \hat{\mathrm{V}} \circ \hat{\mathrm{V}}] \times_2 [\mathrm{I}_p - \hat{\mathrm{V}} \circ \hat{\mathrm{V}}] \times_3 [\mathrm{I} - \hat{\mathrm{u}}\hat{\mathrm{u}}^T]$

- Removes estimated signal: $\mathcal{X}^t \times_1 \mathrm{V}_t \circ \mathrm{V}_t = \mathcal{X}^t \bar{\times}_3 \mathrm{u}_t = 0$
- $\|\mathcal{X}^t\| > \|\mathcal{X}^{t-1}\|$; may not lower rank (Stegman and Comon, *Lin. Alg. Appl.*, 2010)

Practicalities

Initialization:

- For many network problems, signal is roughly constant (or at least positive)
    - Stable initialization: $\mathrm{u}_0 = 1/\sqrt{T}$
    - Performs particularly well for trend-finding and change-point
- Random (re-)initialization works well for hardest problems
- See simulations in paper

Deflation: $\mathcal{X}^t \Leftarrow \mathcal{X}^{t-1} \times_1 [\mathrm{I}_p - \hat{\mathrm{V}} \circ \hat{\mathrm{V}}] \times_2 [\mathrm{I}_p - \hat{\mathrm{V}} \circ \hat{\mathrm{V}}] \times_3 [\mathrm{I} - \hat{\mathrm{u}}\hat{\mathrm{u}}^T]$

- Removes estimated signal: $\mathcal{X}^t \times_1 \mathrm{V}_t \circ \mathrm{V}_t = \mathcal{X}^t \bar{\times}_3 \mathrm{u}_t = 0$
- $\|\mathcal{X}^t\| > \|\mathcal{X}^{t-1}\|$; may not lower rank (Stegman and Comon, *Lin. Alg. Appl.*, 2010)

Rank Selection:

Practicalities

Initialization:

- For many network problems, signal is roughly constant (or at least positive)
  - Stable initialization: $\mathsf{u}_0 = 1/\sqrt{T}$
  - Performs particularly well for trend-finding and change-point
- Random (re-)initialization works well for hardest problems
- See simulations in paper

Deflation: $\mathcal{X}^t \Leftarrow \mathcal{X}^{t-1} \times_1 [\mathsf{I}_p - \hat{\mathsf{V}} \circ \hat{\mathsf{V}}] \times_2 [\mathsf{I}_p - \hat{\mathsf{V}} \circ \hat{\mathsf{V}}] \times_3 [\mathsf{I} - \hat{\mathsf{u}}\hat{\mathsf{u}}^T]$

- Removes estimated signal: $\mathcal{X}^t \times_1 \mathsf{V}_t \circ \mathsf{V}_t = \mathcal{X}^t \bar{\times}_3 \mathsf{u}_t = 0$
- $\|\mathcal{X}^t\| > \|\mathcal{X}^{t-1}\|$; may not lower rank (Stegman and Comon, *Lin. Alg. Appl.*, 2010)

Rank Selection:

- Data-Driven or BIC used to select K (Sedighin, *J. Spec. Top. Sig. Proc.*, 2021)

Practicalities

Initialization:

- For many network problems, signal is roughly constant (or at least positive)
  - Stable initialization: $\mathrm{u}_0 = 1/\sqrt{T}$
  - Performs particularly well for trend-finding and change-point
- Random (re-)initialization works well for hardest problems
- See simulations in paper

Deflation: $\mathcal{X}^t \Leftarrow \mathcal{X}^{t-1} \times_1 [\mathrm{I}_p - \hat{\mathrm{V}} \circ \hat{\mathrm{V}}] \times_2 [\mathrm{I}_p - \hat{\mathrm{V}} \circ \hat{\mathrm{V}}] \times_3 [\mathrm{I} - \hat{\mathrm{u}}\hat{\mathrm{u}}^T]$

- Removes estimated signal: $\mathcal{X}^t \times_1 \mathrm{V}_t \circ \mathrm{V}_t = \mathcal{X}^t \bar{\times}_3 \mathrm{u}_t = 0$
- $\|\mathcal{X}^t\| > \|\mathcal{X}^{t-1}\|$; may not lower rank (Stegman and Comon, *Lin. Alg. Appl.*, 2010)

Rank Selection:

- Data-Driven or BIC used to select K (Sedighin, *J. Spec. Top. Sig. Proc.*, 2021)
- Selecting r too high usually harmless

Application: SCOTUS Voting


EQUAL JUSTICE UNDER LAW

Can we use network analysis to identify voting patterns among SCOTUS Justices?

SCOTUS Network Data

Each term SCOTUS decides ≈ 80 cases:

- Create a **weighted**, **undirected** network based on co-voting[1]
- Analyze by "seat" (AJ7 = Ginsburg = Barrett), not by Justice

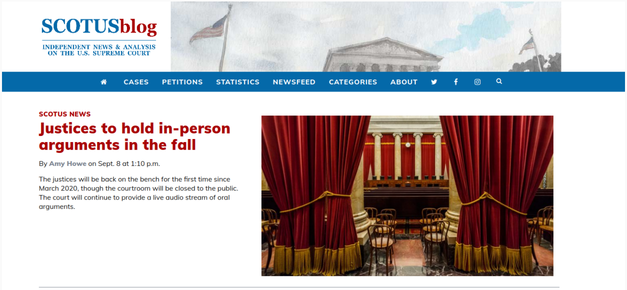
[1] We consider agreement *in the judgement*, not in reasoning.

SCOTUS Network Data

Each term SCOTUS decides ≈ 80 cases:

- Create a **weighted**, **undirected** network based on co-voting[1]
- Analyze by "seat" (AJ7 = Ginsburg = Barrett), not by Justice

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

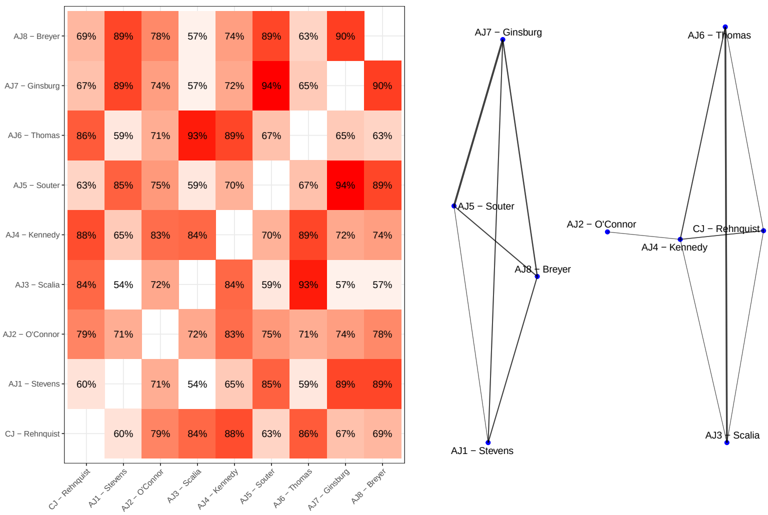


9×9 pairs $\times$ 25 terms $\equiv \mathcal{X} \in \mathbb{R}^{9 \times 9 \times 25}$

[1] We consider agreement *in the judgement*, not in reasoning.

Example - OT 2001

October Term 2001



Data from the SCOTUSBlog Stat Pack

Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

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 analysis)

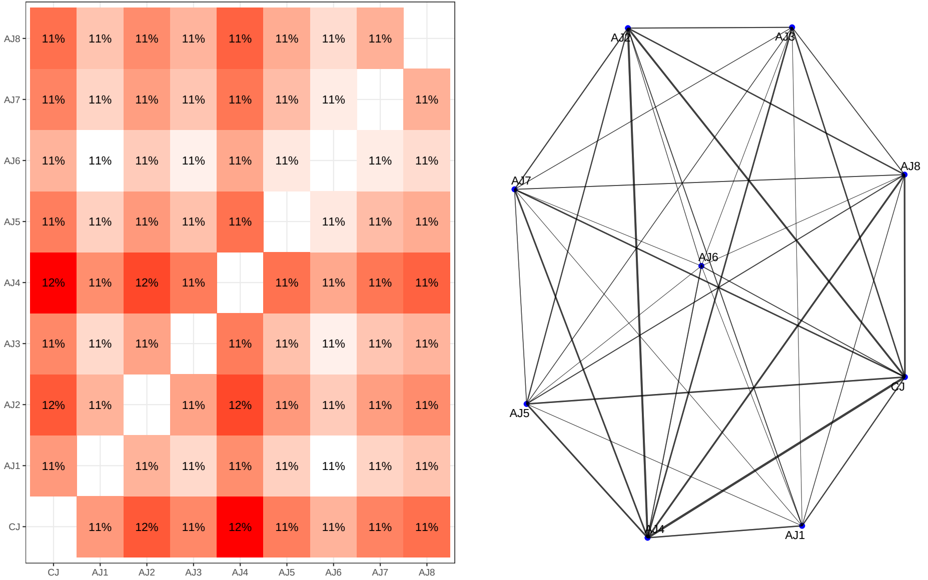
Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

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 analysis)

Baseline (Mean) Court Behavior

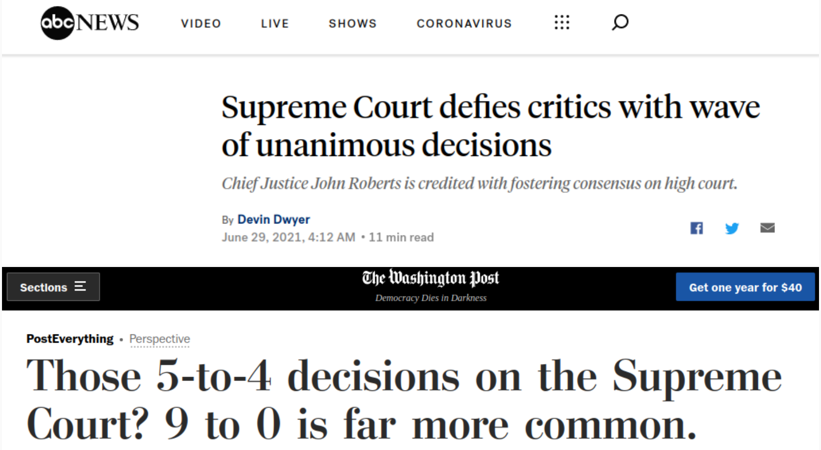
The majority of SCOTUS cases are decided (nearly) unanimously



Data from the SCOTUSBlog Stat Pack

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 analysis)


abcNEWS VIDEO LIVE SHOWS CORONAVIRUS

Supreme Court defies critics with wave of unanimous decisions

Chief Justice John Roberts is credited with fostering consensus on high court.

By **Devin Dwyer**
June 29, 2021, 4:12 AM • 11 min read

Sections
The Washington Post
Democracy Dies in Darkness
Get one year for $40

PostEverything • Perspective

Those 5-to-4 decisions on the Supreme Court? 9 to 0 is far more common.

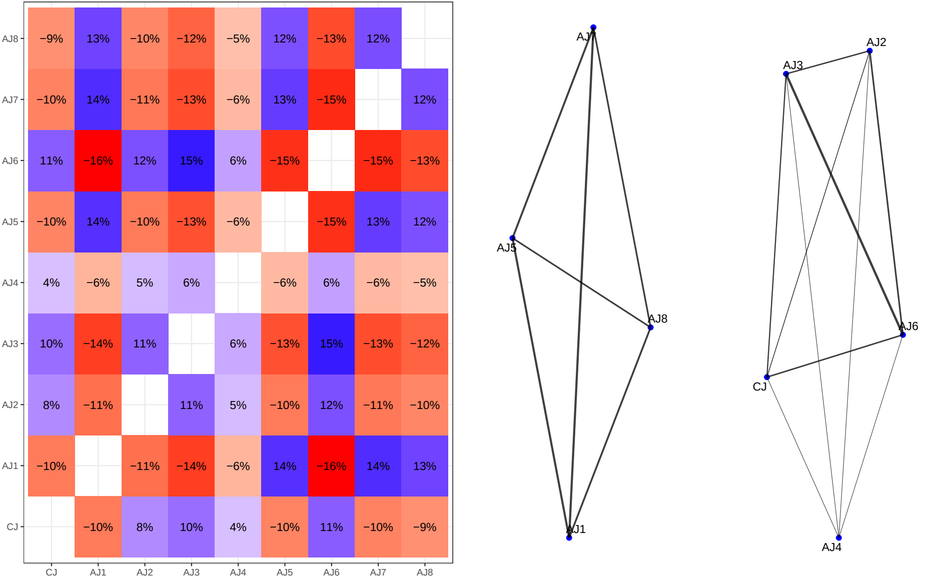
Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

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 analysis)

First Principal Component of Court Behavior

The most significant source of divided rulings is the familiar left/right split



Data from the SCOTUSBlog Stat Pack

Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

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 analysis)


AP

AP NEWS

9/11 A World Changed | AP Top 25 College Football Poll | Coronavirus pandemic | Politics | Sports | Entertainment | Photography

Justice Ginsburg warns of more 5-4 decisions ahead

The New York Times

Splitting 5 to 4, Supreme Court Backs Religious Challenge to Cuomo's Virus

Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

Semi-Symmetric PCA as a Flexible Pattern Recognition Tool:

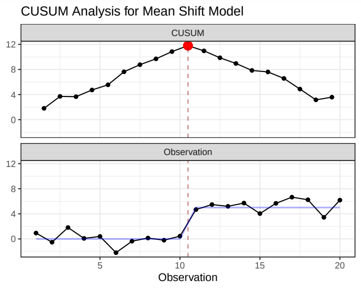
- Raw $\mathcal{X}$ - major patterns (trends)
- Centered $\mathcal{X}$ - variance components (covariance patterns)
- Differenced $\mathcal{X}$ - change-point identification (CUSUM analysis)

Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

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 analysis)

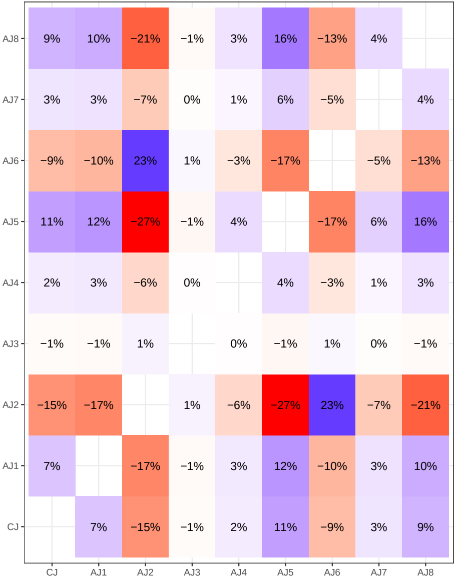
$$\text{cusum}(X)_t = \text{mean}(X_{<t}) - \text{mean}(X_{>t})$$

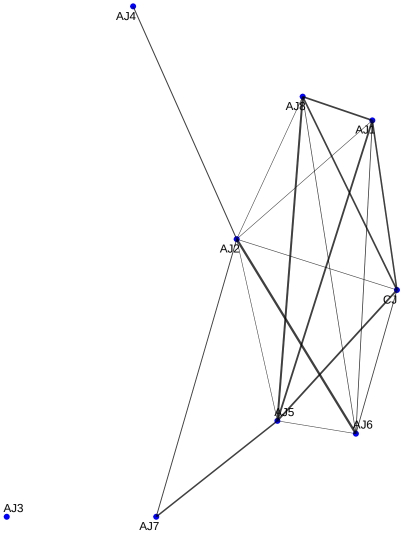


Wang and Samworth, *JRSS-B*, 2018: PCA + CUSUM for time series

First Principal Component of Tensor CUSUM Analysis

The most significant change in court dynamics is O'Connor / Alito (AJ2) seat





Data from the SCOTUSBlog Stat Pack

Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

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 analysis)

Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

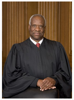
Semi-Symmetric PCA as a Flexible Pattern Recognition Tool:

- Raw $\mathcal{X}$ - major patterns (trends)
- Centered $\mathcal{X}$ - variance components (covariance patterns)
- Differenced $\mathcal{X}$ - change-point identification (CUSUM analysis)

Biggest change in court dynamics - AJ2 O'Connor $\rightarrow$ Alito:

$\Longrightarrow$

$\Longrightarrow$



Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

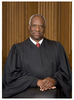
Semi-Symmetric PCA as a Flexible Pattern Recognition Tool:

- Raw $\mathcal{X}$ - major patterns (trends)
- Centered $\mathcal{X}$ - variance components (covariance patterns)
- Differenced $\mathcal{X}$ - change-point identification (CUSUM analysis)

Biggest change in court dynamics - AJ2 O'Connor $\rightarrow$ Alito:

$\Longrightarrow$

$\Longrightarrow$



Also: Scalia / Gorsuch seat (AJ3) essentially unchanged

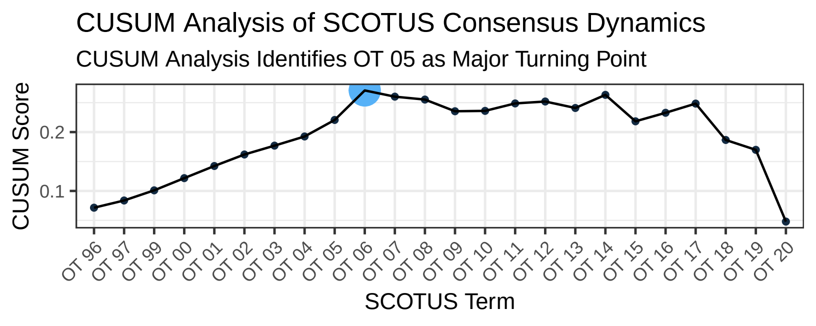
Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

SS-PCA gives both **principal network** and **(time) loading vector**

Semi-Symmetric PCA as a Flexible Pattern Recognition Tool

SS-PCA gives both **principal network** and **(time) loading vector**

For CUSUM analysis, loading vector identifies **when** change occurs



Ongoing and Future Work

Statistical Analysis of Partially Aligned Networks

Given two graphs $\mathcal{G}_1, \mathcal{G}_2$ from (different) graphons, test difference

- Full Alignment: Yes! (Higher criticism, multiple binomial test; Ghoshdastidar *et al.*, *AoS* 2020)
- No Alignment: Yes! (Sabanayagam *et al.*, ICLR 2022)
- Partial Alignment: ??

Statistical Analysis of Partially Aligned Networks

Given two graphs $\mathcal{G}_1, \mathcal{G}_2$ from (different) graphons, test difference

- Full Alignment: Yes! (Higher criticism, multiple binomial test; Ghoshdastidar *et al.*, *AoS* 2020)
- No Alignment: Yes! (Sabanayagam *et al.*, ICLR 2022)
- Partial Alignment: ??

Approach:

- Adapt Smooth-and-Sort estimator (Chan & Airoldi, *ICML* 2014) for partial alignment
    - Message-passing step forces estimates for paired vertices to match
    - Statistical consistency
- Connections to partially paired t-test, permuted regression, *etc.*
- Apply machinery to other statistical problems

Statistical Analysis of Partially Aligned Networks

Given two graphs $\mathcal{G}_1, \mathcal{G}_2$ from (different) graphons, test difference

- Full Alignment: Yes! (Higher criticism, multiple binomial test; Ghoshdastidar *et al.*, *AoS* 2020)
- No Alignment: Yes! (Sabanayagam *et al.*, ICLR 2022)
- Partial Alignment: ??

Approach:

- Adapt Smooth-and-Sort estimator (Chan & Airoldi, *ICML* 2014) for partial alignment
  - Message-passing step forces estimates for paired vertices to match
  - Statistical consistency
- Connections to partially paired t-test, permuted regression, *etc.*
- Apply machinery to other statistical problems

Related Work: "CONGA" (W., Michailidis, Roddenberry, *ICASSP*, 2021)

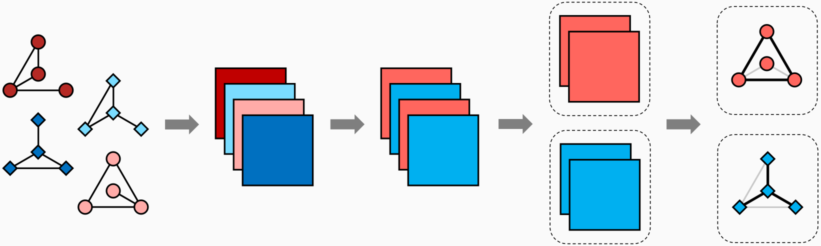
- Simultaneous community detection on two graphs via regularized PLS (sparsity + graph Laplacian smoothing) of graph signals

Unsupervised Learning

Methods for Regularized Matrix Decomposition and Clustering

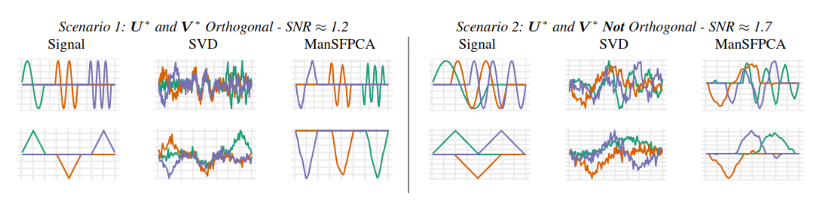
Unsupervised Learning

Methods for Regularized Matrix Decomposition and Clustering



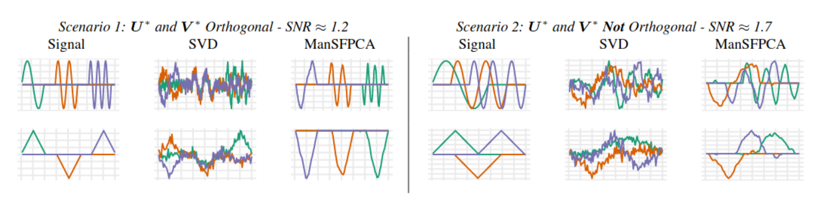
Unsupervised Learning

Methods for Regularized Matrix Decomposition and Clustering



Unsupervised Learning

Methods for Regularized Matrix Decomposition and Clustering



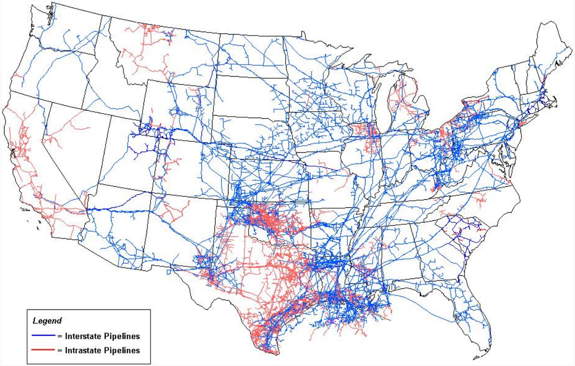
Related Work:

- Sparse + Smooth PCA (Allen and W., *DSW*, 2019)
- Simultaneous regularized PCs via nonsmooth manifold optimization (W., *CAMSAP*, 2019)
- Convex clustering of networks (W. *et al.*, 2022+)
- `MoMA` Software

Time Series Analysis

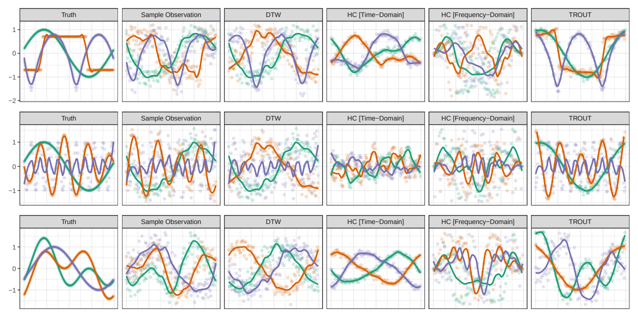
Structured multivariate time series - finance, neuroscience, envirometrics

Structured multivariate time series - finance, neuroscience, envirometrics



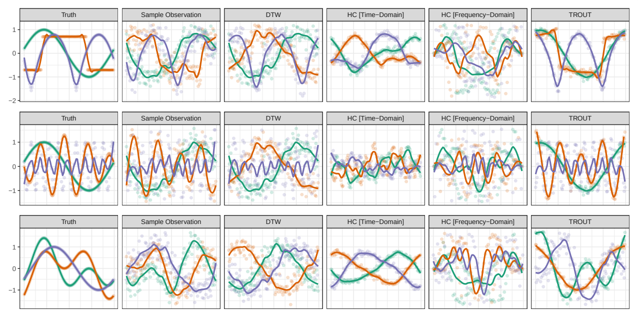
Time Series Analysis

Structured multivariate time series - finance, neuroscience, envirometrics



Time Series Analysis

Structured multivariate time series - finance, neuroscience, envirometrics

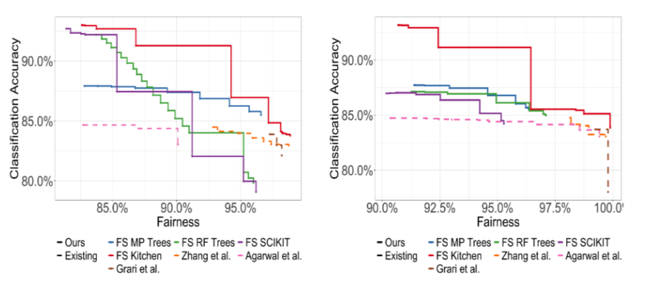


Related Work:

- Econometric modeling of NG futures markets (W. *et al.*, 2022+)
- Clustering of unaligned time series (W. & Michailidis, *ICASSP* 2021)
- Clustering + denoising time series (W. *et al.*, *DSLW*, 2021)
- Complex-Valued Graphical Models of Time Series Spectra (W., 2022+)

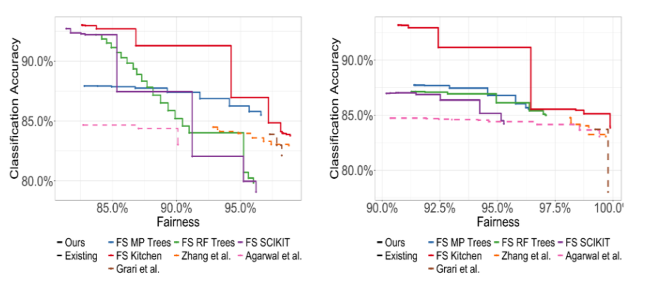
Machine Learning Fairness

Exploring Optimal Fairness-Accuracy Tradeoff (Pareto Frontier)



Machine Learning Fairness

Exploring Optimal Fairness-Accuracy Tradeoff (Pareto Frontier)



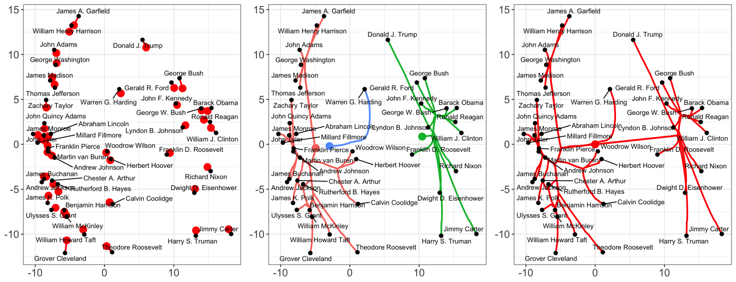
Related Work:

- Measuring, Optimizing, and Testing Fairness-Accuracy Tradeoff (W. *et al.*, 2022+)
- Fair PCA (W. and Allen, 2022+)
- Auditing individual fairness via metric learning (W. and Michailidis, 2022+)

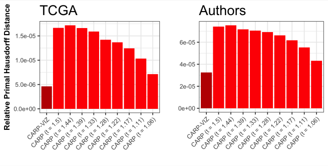
Statistical Computing

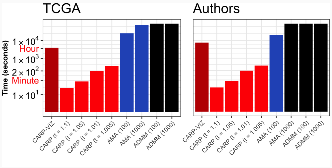
Development of efficient, robust, and “statistically sound” software

Development of efficient, robust, and "statistically sound" software

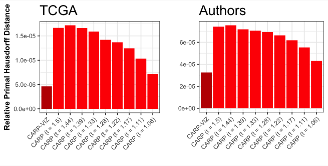


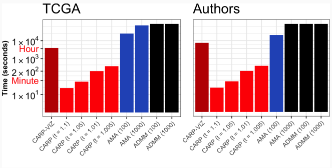
Development of efficient, robust, and "statistically sound" software





Development of efficient, robust, and "statistically sound" software





Related Work:

- Efficient algorithms for computing regularization paths (W. *et al.*, *JCGS*, 2020)
- Algorithms for higher-order convex clustering (W., *DSW*, 2019)
- Manifold optimization in unsupervised learning (W., *CAMSAP*, 2019)
- `clustRviz`, `MoMA`, `ExclusiveLasso`, *etc.* `R` packages

Conclusions

Statistical Analysis of Multiple Networks

Network Tensor PCA – Network Science meets PCA:

Network Tensor PCA – Network Science meets PCA:

- Pattern recognition across **aligned** multiple networks

Statistical Analysis of Multiple Networks

Network Tensor PCA – Network Science meets PCA:

- Pattern recognition across **aligned** multiple networks
- Trends, Variability, Changepoint Detection

Statistical Analysis of Multiple Networks

Network Tensor PCA – Network Science meets PCA:

- Pattern recognition across **aligned** multiple networks
- Trends, Variability, Changepoint Detection
- Efficient power method-inspired algorithm
    - Admits extensions for large, sparse, or streaming data

Statistical Analysis of Multiple Networks

Network Tensor PCA – Network Science meets PCA:

- Pattern recognition across **aligned** multiple networks
- Trends, Variability, Changepoint Detection
- Efficient power method-inspired algorithm
  - Admits extensions for large, sparse, or streaming data
- Theoretical analysis
  - Provable estimation consistency for non-convex problem
  - Comparable rates to classical PCA and better than naïve approach

Statistical Analysis of Multiple Networks

Network Tensor PCA – Network Science meets PCA:

- Pattern recognition across **aligned** multiple networks
- Trends, Variability, Changepoint Detection
- Efficient power method-inspired algorithm
  - Admits extensions for large, sparse, or streaming data
- Theoretical analysis
  - Provable estimation consistency for non-convex problem
  - Comparable rates to classical PCA and better than naïve approach

Tensor PCA: ArXiv 2202.04719 + Questions?

Backup Slides

Noise Model

Key notion of noise: operator noise of $\mathcal{E}$ considered as mapping from $\overline{\mathbb{B}}^T \times \mathcal{V}^{p \times r} \to \mathbb{R}_{\geq 0}$

$$\|\mathcal{E}\|_{r\text{-op}} = \max_{\mathrm{u},\mathrm{V}} \left| \left\langle \left(\mathrm{Tr}(\mathrm{V}^T \mathcal{E}_{\cdot\cdot i} \mathrm{V}) \right)_i, \mathrm{u} \right\rangle \right|$$

Deterministic upper bound:

$$\|\mathcal{E}\|_{r\text{-op}} \leq r\sqrt{T} \max_i \lambda_{\max}(\mathcal{E}_{\cdot\cdot i})$$

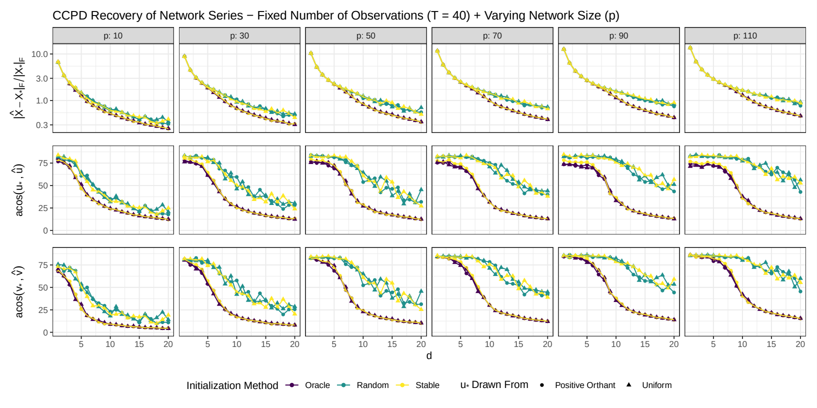
SS-Tensor Concentration bound:

$$\|\mathcal{E}\|_{r\text{-op}} \leq cr\sqrt{T}\sigma(\sqrt{p} + \sqrt{\log T} + \delta)$$

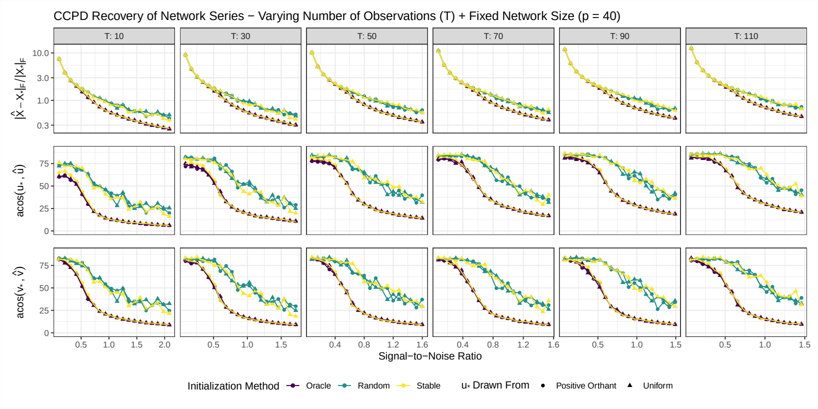
with probability at least $1 - 4e^{-\delta^2}$

c is small $\Leftrightarrow$ $c = 1$ for true Gaussians

Simulations



Simulations



Stock Market Application