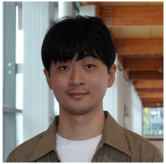
Relational Learning via Covariance Information

Andrea Cavallo

TU Delft

May 1, 2025



Elvin Isufi
TU Delft

Mohammad Sabbaqi
TU Delft

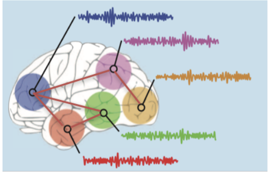
Zhan Gao
Cambridge University

Madeline Navarro
Rice University

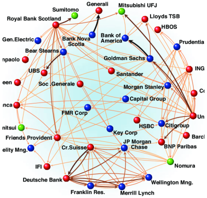
Santiago Segarra
Rice University

Covariance relationships between data points

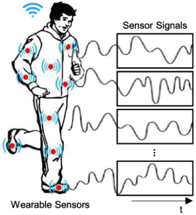
- Data usually contains hidden interconnections



Brain



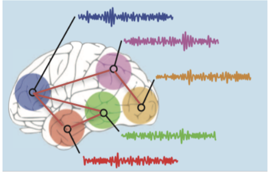
Financial



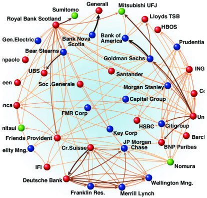
Human Action Recognition

Covariance relationships between data points

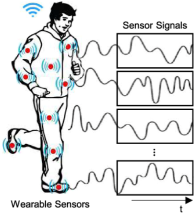
- Data usually contains hidden interconnections



Brain



Financial

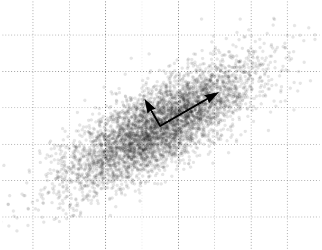


Human Action Recognition

- One way to capture these relations is through the covariance matrix
  $\Rightarrow \mathbf{C} = \mathbb{E}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^{\mathsf{T}}]$

Principal Component Analysis (PCA)

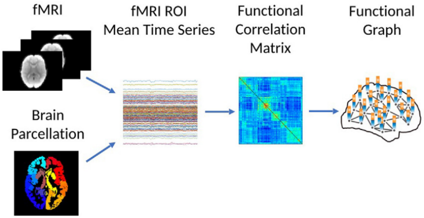
- Project the data onto the covariance eigenspace
  - ⇒ $\mathbf{C} = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^\top$
  - ⇒ $\tilde{\mathbf{x}} = \mathbf{V}^\top \mathbf{x}$
- Find the directions that maximize the variance of data points
  - ⇒ Used for dimensionality reduction by selecting only a few eigenvectors



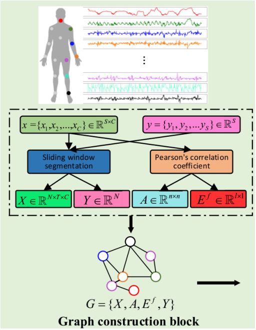
Principal directions

Topology Inference

- Graphical Lasso, Collaborative Filtering, Graph Stationarity



Brain fMRI data



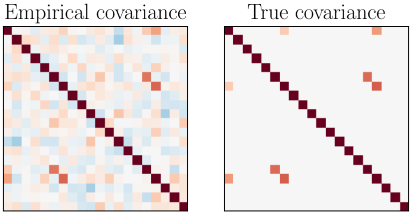
Human motion sensor recordings

- Use this topology for downstream processing

Wang et al., MhaGNN, IEEE Transactions on Instrumentation and Measurement, 2023
Li et al., BrainGNN: Interpretable brain graph neural network for fMRI analysis. Medical Image Analysis, 2021

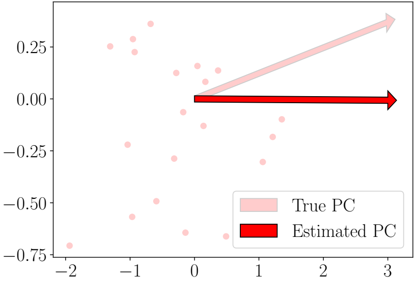
Finite-data effect

- In practice, we work with estimates
  $\Rightarrow \hat{\mathbf{C}} = \frac{1}{t}\sum_{t'=1}^{t} \mathbf{x}_{t'}\mathbf{x}_{t'}^{\mathsf{T}}$
- May not reflect the true underlying structure
  $\Rightarrow$ Sample estimator $\hat{\mathbf{C}}$ is noisy
  $\Rightarrow$ When number of samples t is of same order of data dimension N



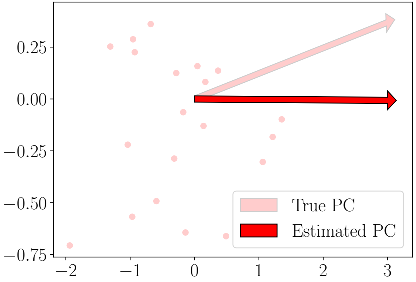
Finite-data effect on PCA

- PCA is unstable to covariance estimation errors



Finite-data effect on PCA

- PCA is unstable to covariance estimation errors



- Problem setting: learn with covariances
  - ⇒ in a stable way
  - ⇒ efficiently
  - ⇒ In a variety of settings: static, temporal, and biased data

Outline

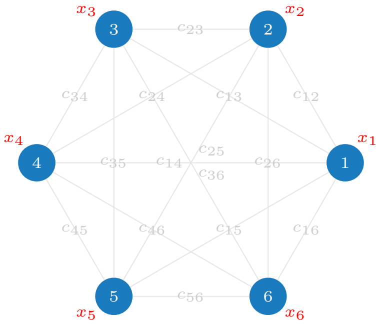
- Covariance Neural Networks
- Sparse VNNs
- Spatiotemporal VNNs
- Fair VNNs
- Conclusions

Outline

- **Covariance Neural Networks**
- Sparse VNNs
- Spatiotemporal VNNs
- Fair VNNs
- Conclusions

Covariance Data Relationships

- Data sample $\mathbf{x} = [x_1, \ldots, x_N]^\mathsf{T}$ with covariance $\mathbf{C}$
  - $\Rightarrow$ Build a graph where:
  - $\Rightarrow$ the features are the node signals x_i
  - $\Rightarrow$ the edges are the covariance values c_{ij} $\rightarrow$ fully-connected graph



Covariance Filters

- Definition: Graph convolution covariance filters

$$\mathbf{z} = \mathbf{H}(\hat{\mathbf{C}})\mathbf{x} = \sum_{k=0}^{K} h_k \hat{\mathbf{C}}^k \mathbf{x}$$

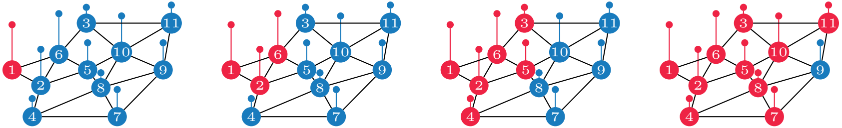
$\Rightarrow$ learnable parameters: h_k

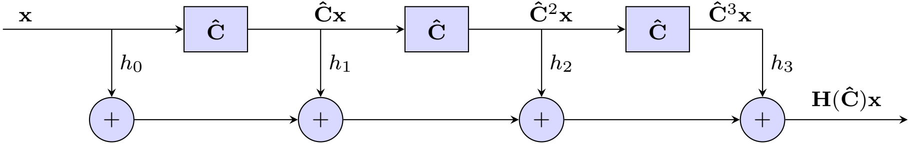
Covariance Filters

- Definition: Graph convolution covariance filters

$$\mathbf{z} = \mathbf{H}(\hat{\mathbf{C}})\mathbf{x} = \sum_{k=0}^{K} h_k \hat{\mathbf{C}}^k \mathbf{x}$$

$\Rightarrow$ learnable parameters: h_k



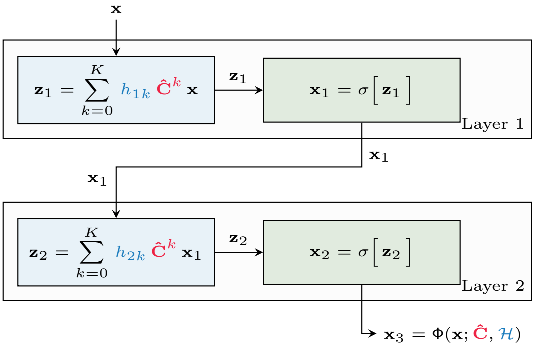


- $\hat{\mathbf{C}}^k\mathbf{x}$ shifts signal $\mathbf{x}$ k times over the covariance graph $\hat{\mathbf{C}}$

Covariance Neural Networks (VNNs)

- Definition: Covariance filters followed by pointwise nonlinearities σ

$$\mathbf{x}^l = \sigma\left(\mathbf{H}^l(\hat{\mathbf{C}})\mathbf{x}^{l-1}\right) \quad l = 1, \ldots, L.$$



Stability of VNNs

- We use the VNN on the noisy covariance matrix $\hat{\mathbf{C}}$

  $\Rightarrow$ How does this affect the performance?

- Stability for true $\mathbf{H}(\mathbf{C})$ vs. estimated $\mathbf{H}(\hat{\mathbf{C}})$ covariance filter:

$$\|\mathbf{H}(\hat{\mathbf{C}}) - \mathbf{H}(\mathbf{C})\| \leq \frac{P}{\sqrt{t}} \mathcal{O}\left(\sqrt{N} + \frac{\|\mathbf{C}\| \sqrt{\log(Nt)}}{\nu t}\right)$$

Stability of VNNs

- We use the VNN on the noisy covariance matrix $\hat{\mathbf{C}}$

  $\Rightarrow$ How does this affect the performance?

- Stability for true $\mathbf{H}(\mathbf{C})$ vs. estimated $\mathbf{H}(\hat{\mathbf{C}})$ covariance filter:

$$\|\mathbf{H}(\hat{\mathbf{C}}) - \mathbf{H}(\mathbf{C})\| \leq \frac{P}{\sqrt{t}} \mathcal{O}\left(\sqrt{N} + \frac{\|\mathbf{C}\|\sqrt{\log(Nt)}}{\nu t}\right)$$

- t and N: number of samples and data dimension

Stability of VNNs

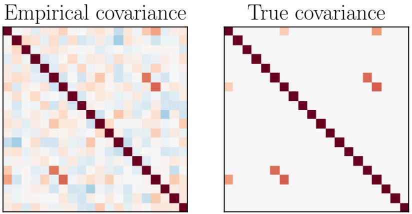
- We use the VNN on the noisy covariance matrix $\hat{\mathbf{C}}$
  $\Rightarrow$ How does this affect the performance?
- Stability for true $\mathbf{H}(\mathbf{C})$ vs. estimated $\mathbf{H}(\hat{\mathbf{C}})$ covariance filter:

$$\|\mathbf{H}(\hat{\mathbf{C}}) - \mathbf{H}(\mathbf{C})\| \leq \frac{P}{\sqrt{t}} \mathcal{O}\left(\sqrt{N} + \frac{\|\mathbf{C}\|\sqrt{\log(Nt)}}{\nu t}\right)$$

- t and N: number of samples and data dimension
- Data distribution

Limitations of VNNs

- However:
  - ⇒ Low-data high-dimensional settings → sample covariance estimator is really bad!



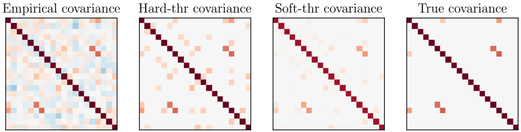
  - ⇒ Computationally inefficient → complexity $\mathcal{O}(N^2)$
  - ⇒ Time-unaware → ignores temporal dependencies and distribution shifts
  - ⇒ Unfair → if the data is biased

Outline

- Covariance Neural Networks
- **Sparse VNNs**
- Spatiotemporal VNNs
- Fair VNNs
- Conclusions

Sparse Covariance Matrices

- In practice, the true covariance matrix $\mathbf{C}$ is sparse



- Better estimators:

$\Rightarrow$ Hard thresholding:

$$[\bar{\mathbf{C}}]_{ij} = \begin{cases} \hat{c}_{ij} & \text{if } |\hat{c}_{ij}| \geq \tau/\sqrt{t} \\ 0 & \text{otherwise} \end{cases}$$

$\Rightarrow$ Soft thresholding:

$$[\bar{\mathbf{C}}]_{ij} = \begin{cases} \hat{c}_{ij} - \text{sign}(\hat{c}_{ij})\tau/\sqrt{t} & \text{if } |\hat{c}_{ij}| \geq \tau/\sqrt{t} \\ 0 & \text{otherwise} \end{cases}$$

$\Rightarrow$ Stochastic sparsification: no need to choose a threshold

Sparse Covariance Neural Networks (S-VNNs)

- Hard-thresholding:
  - $\Rightarrow$ Stability for true $\mathbf{H}(\mathbf{C})$ vs. hard-thresholded $\mathbf{H}(\bar{\mathbf{C}})$ covariance filter:

$$\|\mathbf{H}(\bar{\mathbf{C}}) - \mathbf{H}(\mathbf{C})\| \leq \frac{1}{\sqrt{t}} P c_0 \sqrt{N \log N}(1 + \sqrt{2N}) + \mathcal{O}\left(\frac{1}{t}\right).$$

- c_0: number of non-zero elements in $\bar{\mathbf{C}}$
  - $\Rightarrow$ $c_0 \ll \|\mathbf{C}\| \rightarrow$ tighter bound!
  - $\Rightarrow$ We need fewer data

Sparse Covariance Neural Networks (S-VNNs)

- Soft-thresholding
  $\Rightarrow$ Stability for true $\mathbf{H}(\mathbf{C})$ vs. soft-thresholded $\mathbf{H}(\bar{\mathbf{C}})$ covariance filter:

$$\|\mathbf{H}(\bar{\mathbf{C}}) - \mathbf{H}(\mathbf{C})\| \leq \frac{1}{\sqrt{t}} P\sqrt{N} C c_0 \max(1, \lambda_{max}) \sqrt{\max\left(\log(N/c_0^2), 1\right)} (1 + \sqrt{2N}) + \mathcal{O}\left(\frac{1}{t}\right).$$

- $\sqrt{\max\left(\log(N/c_0^2), 1\right)} \leq \sqrt{\log N} \rightarrow$ even tighter bound!

Sparse Covariance Neural Networks (S-VNNs)

- Stochastic sparsification
  - ⇒ Stability for true $\mathbf{H}(\mathbf{C})$ vs. stochastically sparsified $\mathbf{H}(\tilde{\mathbf{C}})$ covariance filter:

$$\mathbb{E}[\|\mathbf{H}(\mathbf{C})\mathbf{x} - \mathbf{H}(\tilde{\mathbf{C}})\mathbf{x}\|^2] \leq NP^2Q + \mathcal{O}((1-p_1)(1-p_2)) + \frac{P^2}{t^{1-2\epsilon}}\mathcal{O}\left(N + \frac{\|\mathbf{C}\|^2 \log(Nt)}{\nu^2 t^{4\epsilon}}\right)$$

Sparse Covariance Neural Networks (S-VNNs)

- Stochastic sparsification
  ⇒ Stability for true $\mathbf{H}(\mathbf{C})$ vs. stochastically sparsified $\mathbf{H}(\tilde{\mathbf{C}})$ covariance filter:

$$\mathbb{E}[\|\mathbf{H}(\mathbf{C})\mathbf{x} - \mathbf{H}(\tilde{\mathbf{C}})\mathbf{x}\|^2] \leq NP^2Q + \mathcal{O}((1-p_1)(1-p_2)) + \frac{P^2}{t^{1-2\epsilon}}\mathcal{O}\left(N + \frac{\|\mathbf{C}\|^2 \log(Nt)}{\nu^2 t^{4\epsilon}}\right)$$

- Covariance estimation error

Sparse Covariance Neural Networks (S-VNNs)

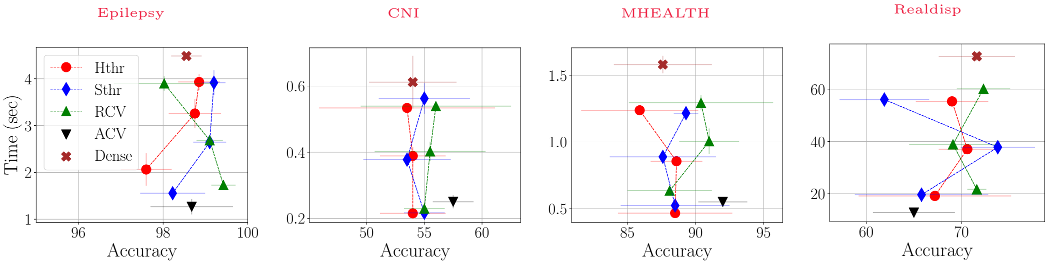
- Stochastic sparsification
  ⇒ Stability for true $\mathbf{H}(\mathbf{C})$ vs. stochastically sparsified $\mathbf{H}(\tilde{\mathbf{C}})$ covariance filter:

$$\mathbb{E}[\|\mathbf{H}(\mathbf{C})\mathbf{x} - \mathbf{H}(\tilde{\mathbf{C}})\mathbf{x}\|^2] \leq NP^2Q + \mathcal{O}((1-p_1)(1-p_2)) + \frac{P^2}{t^{1-2\epsilon}}\mathcal{O}\left(N + \frac{\|\mathbf{C}\|^2 \log(Nt)}{\nu^2 t^{4\epsilon}}\right)$$

- Covariance estimation error
- Sparsification error

Numerical Results

- Brain recordings: classify patient condition
- Human Action Recognition: classify action performed



- S-VNNs are always faster and more accurate due to spurious correlation removal

Limitations of VNNs

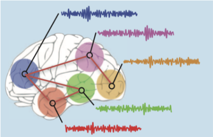
- However:
  - ⇒ Low-data high-dimensional settings → sample covariance estimator is really bad!
  - ⇒ Computationally inefficient → complexity $\mathcal{O}(N^2)$
  - ⇒ Time-unaware → ignores temporal dependencies and distribution shifts
  - ⇒ Unfair → if the data is biased

Outline

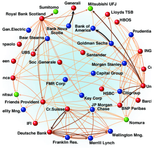
- Covariance Neural Networks
- Sparse VNNs
- **Spatiotemporal VNNs**
- Fair VNNs
- Conclusions

VNNs and Temporal Data

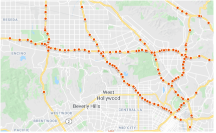
- Setting: data are often time-varying and non-stationary



Brain



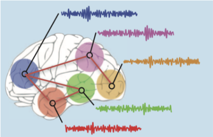
Financial



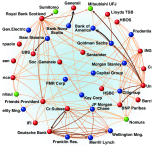
Traffic

VNNs and Temporal Data

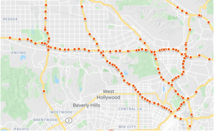
- Setting: data are often time-varying and non-stationary



Brain

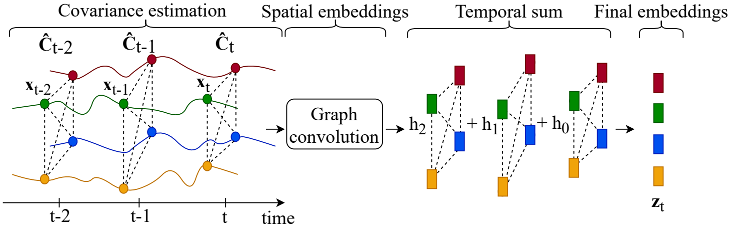


Financial



Traffic

- SpatioTemporal coVariance Neural Networks



Limitations of VNNs

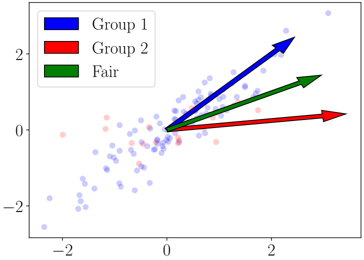
- However:
  - ⇒ Low-data high-dimensional settings → sample covariance estimator is really bad!
  - ⇒ Computationally inefficient → complexity $\mathcal{O}(N^2)$
  - ⇒ Time-unaware → ignores temporal dependencies and distribution shifts
  - ⇒ Unfair → if the data is biased

Outline

- Covariance Neural Networks
- Sparse VNNs
- Spatiotemporal VNNs
- **Fair VNNs**
- Conclusions

Fair PCA

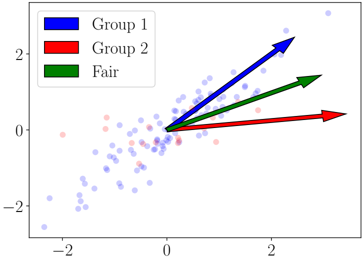
- Principal Component Analysis (PCA) on unbalanced data
  - ⇒ Principal component favors the majority group
  - ⇒ Fair PCA: balance contributions of both groups

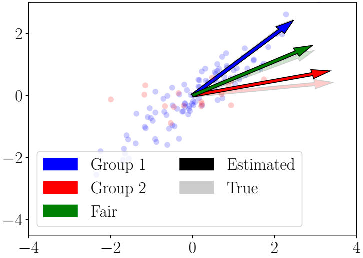


Fair PCA

- Principal Component Analysis (PCA) on unbalanced data
  - ⇒ Principal component favors the majority group
  - ⇒ Fair PCA: balance contributions of both groups

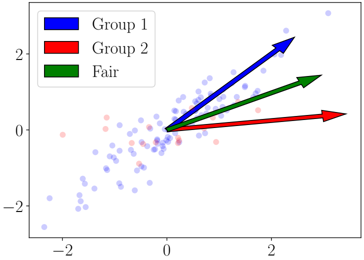
- PCA estimation in low-data setting is difficult
  - ⇒ Minority group estimation is worse
  - ⇒ Estimation errors lead to unfair treatment!

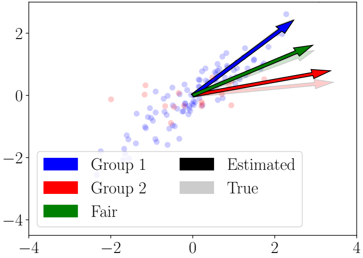




Fair PCA

- Principal Component Analysis (PCA) on unbalanced data
  - ⇒ Principal component favors the majority group
  - ⇒ Fair PCA: balance contributions of both groups
- PCA estimation in low-data setting is difficult
  - ⇒ Minority group estimation is worse
  - ⇒ Estimation errors lead to unfair treatment!
- Solution: Fair Covariance Neural Networks
  - ⇒ Intrinsically mitigate bias due to PCA estimation
  - ⇒ Achieve fair group treatments





Outline

- Covariance Neural Networks
- Sparse VNNs
- Spatiotemporal VNNs
- Fair VNNs
- **Conclusions**

Conclusions

- Learning with covariance matrices
  - ⇒ Capture hidden data relations
  - ⇒ Unstable and ineffective in low-data high-dimensional regimes

Conclusions

- Learning with covariance matrices
  - ⇒ Capture hidden data relations
  - ⇒ Unstable and ineffective in low-data high-dimensional regimes
- coVariance Neural Networks (VNNs)
  - ⇒ Process covariance matrices and are stable
  - ⇒ Computationally inefficient and may fail in sparse setting → Sparse VNNs
  - ⇒ Cannot handle non-stationary temporal data → Spatiotemporal VNNs
  - ⇒ Suffer data imbalance → Fair VNNs

Conclusions

- Learning with covariance matrices
  - ⇒ Capture hidden data relations
  - ⇒ Unstable and ineffective in low-data high-dimensional regimes
- coVariance Neural Networks (VNNs)
  - ⇒ Process covariance matrices and are stable
  - ⇒ Computationally inefficient and may fail in sparse setting → Sparse VNNs
  - ⇒ Cannot handle non-stationary temporal data → Spatiotemporal VNNs
  - ⇒ Suffer data imbalance → Fair VNNs
- Thank you for the attention! Main references and code:

⇒ Sihag, Mateos, McMillan & Ribeiro. coVariance Neural Networks. NeurIPS, 2022

⇒ Cavallo, Gao & Isufi. Sparse Covariance Neural Networks. under review, 2024

⇒ Cavallo, Sabbaqi & Isufi. Spatiotemporal Covariance Neural Networks. ECML PKDD, 2024

⇒ Cavallo, Navarro, Segarra & Isufi. Fair Covariance Neural Networks. ICASSP 2025