University of BRISTOL

Universal Dynamic Network Embedding

How to Comprehend Changes in 20,000 Dimensions.

Ed Davis

PhD Student at the University of Bristol

September 2022

Work completed in collaboration with Patrick Rubin-Delanchy and Dan Lawson

In this Talk

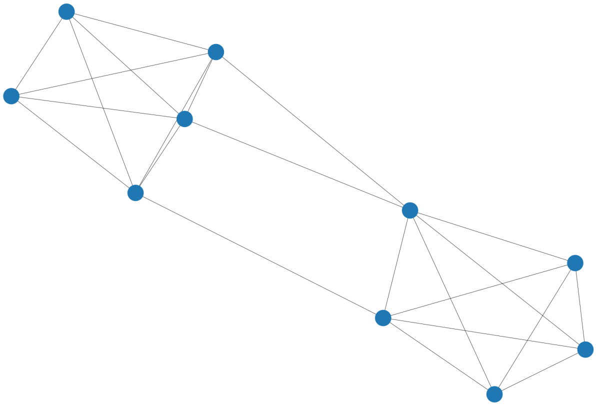
- What is network embedding.
- How can we do dynamic network embedding.
- Our contributions to dynamic embedding.
- A brief history of the world since 1266 as told by a dynamic embedding.

Spot the network communities

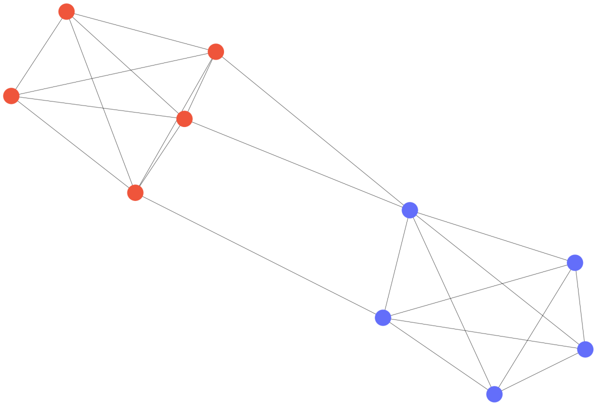
Spot the network communities

Groups of nodes which behave in a similar way

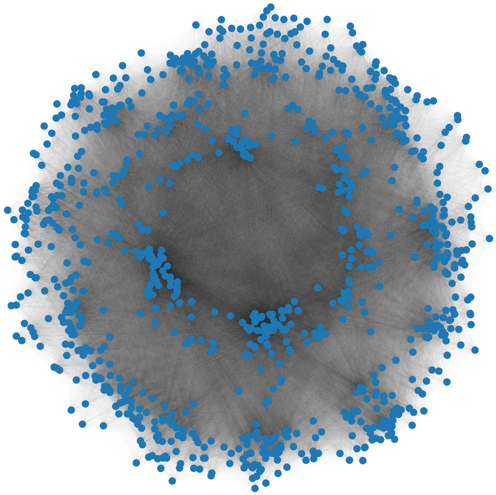
Spot the communities: n=10



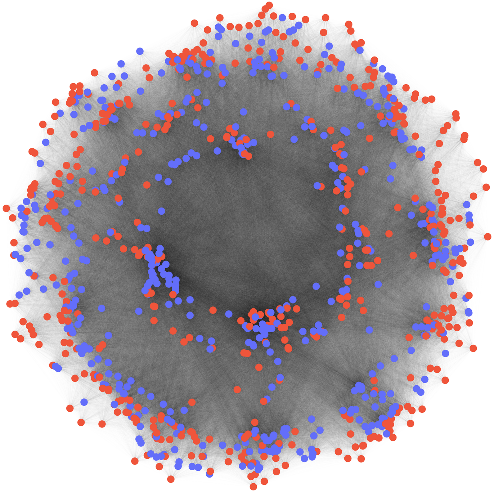
Spot the communities: n=10



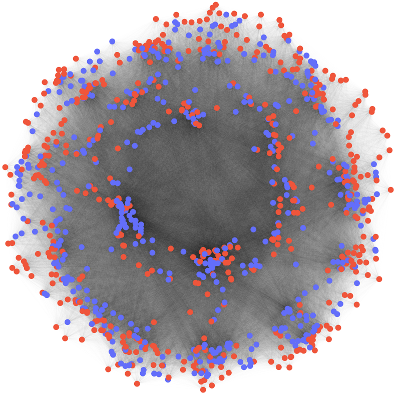
Spot the communities: n=1000



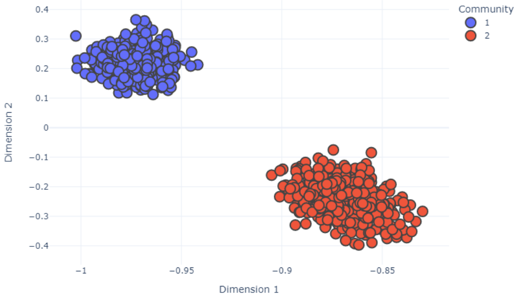
Spot the communities: n=1000



Spot the communities



(a) 1000 node network



(b) 2D embedding of network

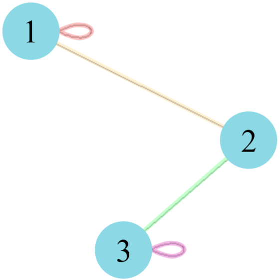
Network Basics

The Adjacency Matrix

We can define an n node graph, G, using an **adjacency matrix**, $\mathrm{A} \in \mathbb{R}^{n \times n}$, where for nodes $i, j \in V$,

$$\mathrm{A}_{ij} = \begin{cases} 1 & (i, j) \in \varepsilon \\ 0 & (i, j) \notin \varepsilon. \end{cases} \tag{1}$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$



Spectral Embedding

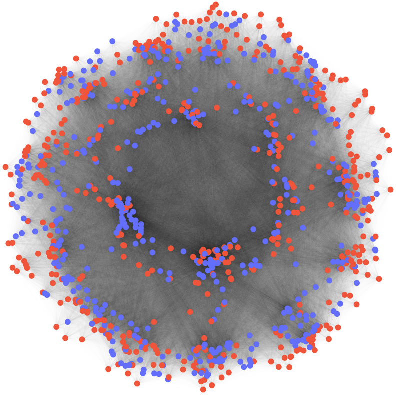
Take the spectral decomposition (eigendecomposition),

$$A = \underbrace{U_A S_A U_A{}^\top}_{\substack{\text{Largest } d \\ \text{by eigenvalue}}} + \underbrace{U_{A,\perp} S_{A,\perp} U_{A,\perp}{}^\top}_{\text{Discard}}. \tag{2}$$

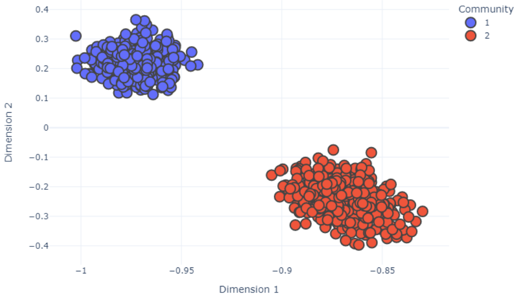
The spectral embedding is then given as

$$\hat{X}_A = U_A S_A{}^{\frac{1}{2}}. \tag{3}$$

Spectral Embedding



(a) 1000 node network



(b) 2D embedding of network

Dynamic Network Embedding

Dynamic Networks

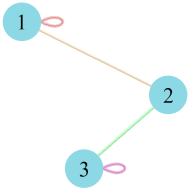
- We have a series of networks, $G^{(1)}, \ldots, G^{(T)}$, with adjacency matrices, $\mathrm{A}^{(1)}, \ldots, \mathrm{A}^{(T)}$.
- We want to have an embedding representation for each adjacency, $\hat{\mathrm{Y}}^{(1)}, \ldots, \hat{\mathrm{Y}}^{(T)}$.
- Want to answer two questions:
  - Does any node move over time?
  - Do any two nodes behave the same?

The Probability Matrix

The **probability matrix** is a noise-free representation of an adjacency matrix, $\mathrm{P} \in \mathbb{R}^{n \times n}$ such that

$$\mathrm{A}_{ij} \overset{\text{ind.}}{\sim} \text{Bernoulli}(\mathrm{P}_{ij}).$$

$$\boldsymbol{P} = \begin{bmatrix} 0.9 & 0.7 & 0.1 \\ 0.7 & 0.2 & 0.8 \\ 0.1 & 0.8 & 0.4 \end{bmatrix} \rightarrow \boldsymbol{A} = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \rightarrow$$



Stability in Dynamic Networks

We use $\mathrm{P}_i^{(t)}$ as a noise-free representation of the behaviour of the ith node at time t.

Stability in Dynamic Networks

We use $\mathrm{P}_i^{(t)}$ as a noise-free representation of the behaviour of the ith node at time t.

Definition (Spatial Stability)

If $\mathrm{P}_i^{(t)} = \mathrm{P}_j^{(t)}$, then $\hat{\mathrm{Y}}_i^{(t)} = \hat{\mathrm{Y}}_j^{(t)}$ for each $i \neq j \in [n]$.
If two nodes behave the same at some time, their embedded positions should be the same at that time.

Stability in Dynamic Networks

We use $\mathrm{P}_i^{(t)}$ as a noise-free representation of the behaviour of the ith node at time t.

Definition (Spatial Stability)

If $\mathrm{P}_i^{(t)} = \mathrm{P}_j^{(t)}$, then $\hat{\mathrm{Y}}_i^{(t)} = \hat{\mathrm{Y}}_j^{(t)}$ for each $i \neq j \in [n]$.
If two nodes behave the same at some time, their embedded positions should be the same at that time.

Definition (Temporal Stability)

If $\mathrm{P}_i^{(t)} = \mathrm{P}_i^{(u)}$, then $\hat{\mathrm{Y}}_i^{(t)} = \hat{\mathrm{Y}}_i^{(u)}$ for each $t \neq u \in [T]$.
If a node does not move, its embedded position should not move.

How Can we Embed Dynamic Networks

- **Separate spectral embeddings (SSE)**: new embedding space for each adjacency matrix (spatially stable).

How Can we Embed Dynamic Networks

- **Separate spectral embeddings (SSE)**: new embedding space for each adjacency matrix (spatially stable).
- **Omnibus embedding (OMNI)** [1]: compute a single embedding space for node to move in (temporally stable).

$$\mathrm{M} = \begin{bmatrix} \mathrm{A}^{(1)} & \frac{\mathrm{A}^{(1)}+\mathrm{A}^{(2)}}{2} & \cdots & \frac{\mathrm{A}^{(1)}+\mathrm{A}^{(T)}}{2} \\ \frac{\mathrm{A}^{(2)}+\mathrm{A}^{(1)}}{2} & \mathrm{A}^{(2)} & \cdots & \frac{\mathrm{A}^{(2)}+\mathrm{A}^{(T)}}{2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\mathrm{A}^{(T)}+\mathrm{A}^{(1)}}{2} & \frac{\mathrm{A}^{(T)}+\mathrm{A}^{(2)}}{2} & \cdots & \mathrm{A}^{(T)} \end{bmatrix}. \tag{4}$$

[1] K. Levin, A. Athreya, M. Tang, V. Lyzinski, Y. Park, and C. E. Priebe, “A central limit theorem for an omnibus embedding of multiple random graphs and implications for multiscale network inference,” *arXiv preprint arXiv:1705.09355*, 2017

How Can we Embed Dynamic Networks

- **Unfolded adjacency spectral embedding (UASE)** [2] [3]: computes a single embedding where nodes from different times cannot interact (spatially and temporally stable).

$$\boldsymbol{\mathcal{A}} = \left(\mathrm{A}^{(1)}, \ldots, \mathrm{A}^{(T)}\right) \in \mathbb{R}^{n \times nT}. \tag{5}$$

---

[2] A. Jones and P. Rubin-Delanchy, “The multilayer random dot product graph,” *arXiv preprint arXiv:2007.10455*, 2020

[3] I. Gallagher, A. Jones, and P. Rubin-Delanchy, “Spectral embedding for dynamic networks with stability guarantees,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 158–10 170, 2021

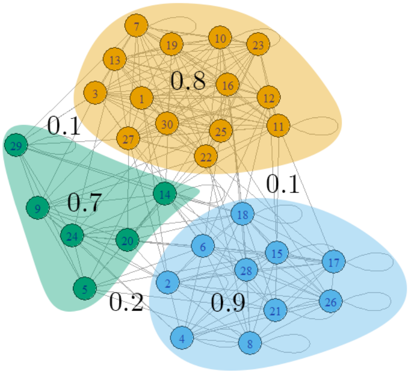
Our Contribution Part 1: Statistical Testing for Changes

Simulating what can happen in a dynamic network

- **IID**: All nodes behave the same.
- **Single move**: Some nodes behave the same, others change behaviour.
- **Merge**: Some nodes suddenly behave the same as others.

Simulating networks: The Stochastic Block Model (SBM)

$$\boldsymbol{B} = \begin{bmatrix} 0.8 & 0.1 & 0.1 \\ 0.1 & 0.9 & 0.2 \\ 0.1 & 0.2 & 0.7 \end{bmatrix}$$



Simulating what can happen in a dynamic network

- **IID**: All nodes behave the same.
- **Single move**: Some nodes behave the same, others change behaviour.
- **Merge**: Some nodes suddenly behave the same as others.

Experiment	SBM Matrix	Stability Required
IID	$\mathrm{B}^{(1)} = \begin{bmatrix} 0.8 & 0.2 \\ 0.2 & 0.4 \end{bmatrix} = \mathrm{B}^{(2)}$	Temporal
Single moving community	$\mathrm{B}^{(1)} = \begin{bmatrix} 0.5 & 0.2 \\ 0.2 & 0.5 \end{bmatrix}, \mathrm{B}^{(2)} = \begin{bmatrix} 0.5 & 0.2 \\ 0.2 & 0.503 \end{bmatrix}$	Temporal and Spatial
Community merge	$\mathrm{B}^{(1)} = \begin{bmatrix} 0.9 & 0.2 \\ 0.2 & 0.1 \end{bmatrix}, \mathrm{B}^{(2)} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & 0.5 \end{bmatrix}$	Spatial

Temporal Permutation Testing for Change

Hypotheses,

$$H_0 : \hat{Y}_1^{(1)}, \ldots, \hat{Y}_m^{(1)} \overset{d}{=} \hat{Y}_1^{(2)}, \ldots, \hat{Y}_m^{(2)}, \quad \textit{there is no change.} \tag{6}$$

$$H_1 : \hat{Y}_1^{(1)}, \ldots, \hat{Y}_m^{(1)} \overset{d}{\neq} \hat{Y}_1^{(2)}, \ldots, \hat{Y}_m^{(2)}, \quad \textit{there is a change.}$$

Temporal Permutation Testing for Change

Hypotheses,

$$H_0 : \hat{Y}_1^{(1)}, \ldots, \hat{Y}_m^{(1)} \overset{d}{=} \hat{Y}_1^{(2)}, \ldots, \hat{Y}_m^{(2)}, \quad \textit{there is no change.} \tag{7}$$

$$H_1 : \hat{Y}_1^{(1)}, \ldots, \hat{Y}_m^{(1)} \overset{d}{\neq} \hat{Y}_1^{(2)}, \ldots, \hat{Y}_m^{(2)}, \quad \textit{there is a change.}$$

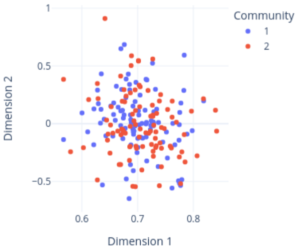
Mean change test statistic,

$$T(\mathrm{x}, \mathrm{y}) = |\bar{\mathrm{x}} - \bar{\mathrm{y}}|, \quad t^{\text{obs}} = T\left(\hat{Y}^{(1)}, \hat{Y}^{(2)}\right), \quad t^\star = T\left(\hat{Y}^{(1)\star}, \hat{Y}^{(2)\star}\right). \tag{8}$$

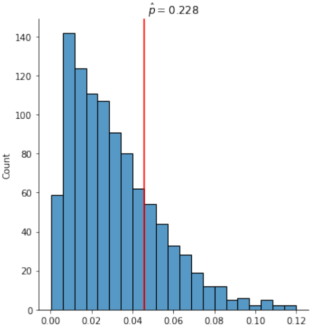
P-value after n_{sim} generations of $t^\star$,

$$\hat{p} = \frac{1}{n_{\text{sim}}} \sum_{n_{\text{sim}}} \mathbb{I}(t^\star \geqslant t^{\text{obs}}). \tag{9}$$

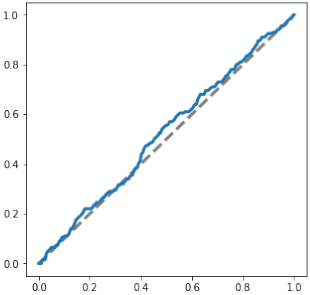
Permutation Testing for Change



(a) Example network with identical communities.

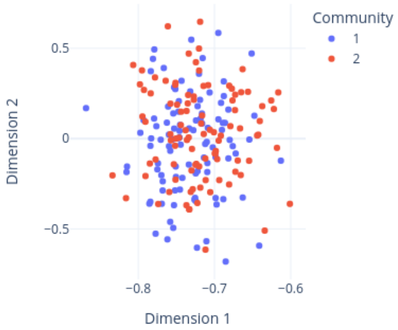


(b) Compute p-value.

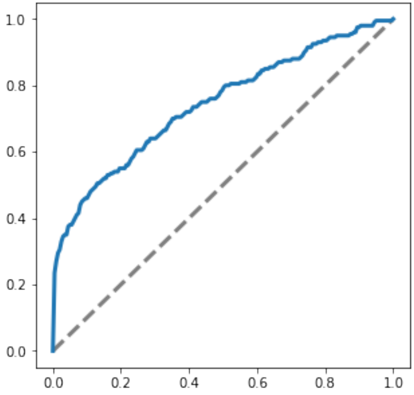


(c) Cumulative p-value distribution from the samples.

Permutation Testing for Change

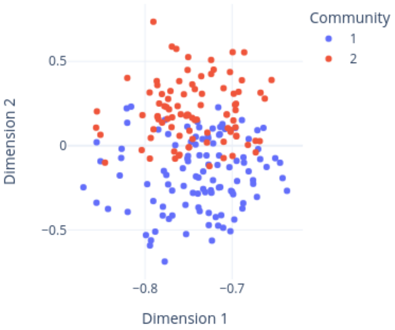


(a) Example network with close communities.

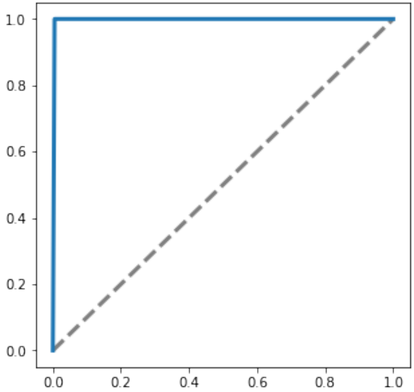


(b) Cumulative p-value distribution.

Permutation Testing for Change

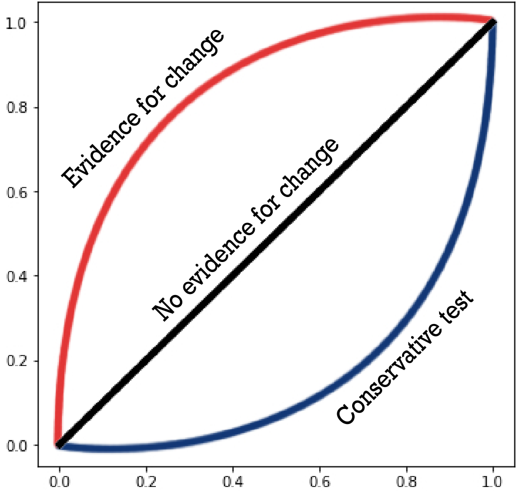


(a) Example network with different communities.

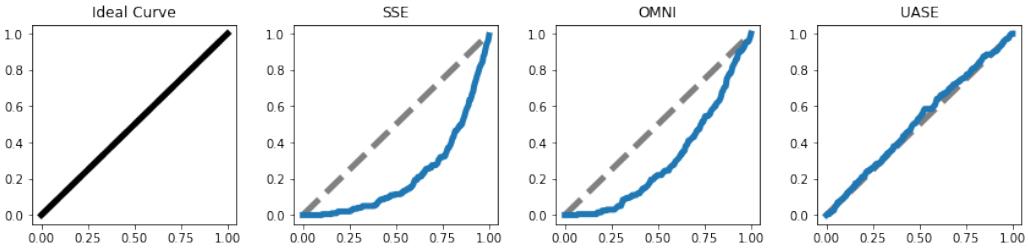


(b) Cumulative p-value distribution.

Permutation Testing for Change

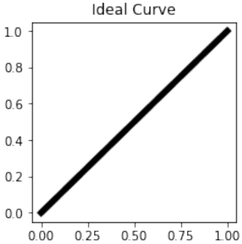


Permutation Testing Results: IID Communities

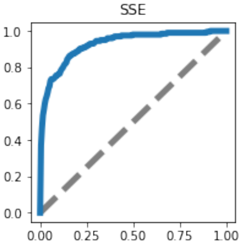


(a) Ideal (b) Temporal instability (c) Spatial instability (d) Stable

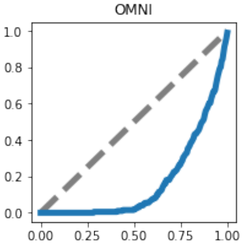
Permutation Testing Results: Single Moving Community (Static One)



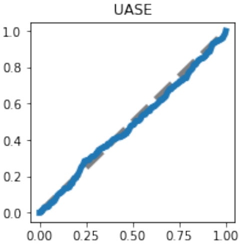
(a) Ideal



(b) Temporal instability

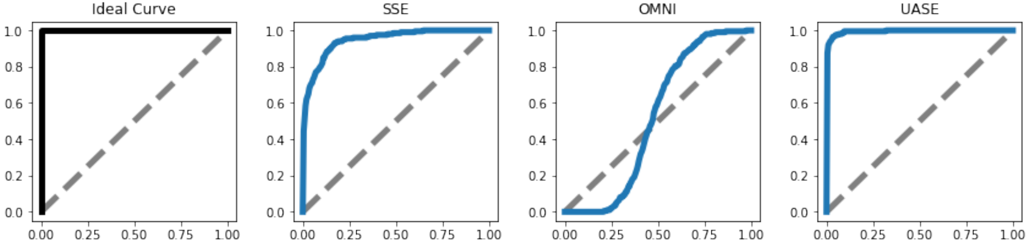


(c) Spatial instability



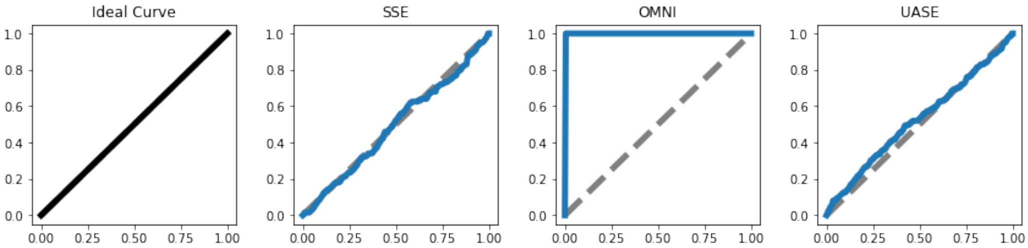
(d) Stable

Permutation Testing Results: Single Moving Community (Moving One)



(a) Ideal (b) Temporal instability (c) Spatial instability (d) Stable

Permutation Testing Results: Community Merge



(a) Ideal (b) Temporal instability (c) Spatial instability (d) Stable

Problem

- To our knowledge UASE is the only embedding method which is stable [4].
- Not robust to degree heterogeneity, sparsity, direction, overfitting...
- Choice:
  - Keep temporal information but lose robustness.
  - Have robust embeddings, but with no way of reliably getting temporal information.

[4] I. Gallagher, A. Jones, and P. Rubin-Delanchy, “Spectral embedding for dynamic networks with stability guarantees,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 158–10 170, 2021

Our Contribution Part 2: Universal Stable Dynamic Embedding

Universal Unfolded Adjacency Matrix

- Unfolded adjacency matrix: $\boldsymbol{\mathcal{A}} = \left(\mathrm{A}^{(1)}, \ldots, \mathrm{A}^{(T)}\right) \in \mathbb{R}^{(n \times nT)}$
- Most embedding methods require a square matrix.
- Define the universal unfolded adjacency matrix

$$\mathrm{A} = \begin{bmatrix} 0 & \left(\mathrm{A}^{(1)}, \ldots, \mathrm{A}^{(T)}\right) \\ \begin{pmatrix} \mathrm{A}^{(1)\top} \\ \vdots \\ \mathrm{A}^{(T)\top} \end{pmatrix} & 0 \end{bmatrix} = \begin{bmatrix} 0 & \boldsymbol{\mathcal{A}} \\ \boldsymbol{\mathcal{A}}^\top & 0 \end{bmatrix} \in \mathbb{R}^{(n+nT)\times(n+nT)}. \quad (10)$$

- Dynamic embedding then given by $\hat{\mathrm{Y}}_{\mathrm{A}}^{[n:n+nT,:]} \in \mathbb{R}^{nT \times d}$.

Universal Unfolded Adjacency Matrix

Define the universal unfolded adjacency matrix

$$\mathbf{A} = \begin{bmatrix} 0 & \boldsymbol{\mathcal{A}} \\ \boldsymbol{\mathcal{A}}^\top & 0 \end{bmatrix} \in \mathbb{R}^{(n+nT)\times(n+nT)}.$$

Theorem

Any single-graph embedding method which is invariant to node-labelling will be spatially and temporally stable when acting on the universal unfolded adjacency matrix.

Any valid embedding method can be used to produce stable embeddings

- Can use your favourite network embedding.
  - **Degree-corrected Regularised Laplacian Spectral Embedding**.
    - Spectral embedding robust to degree-heterogeneity and overfitting.
  - **Deepwalk**, **Node2Vec**.
    - Non-spectral. Iterative learning based on random walks. $\sim$ 8,000 citations.

Real Data: A History of World Alliances

A History of World Alliances

- Over time alliances between countries form and dissolve.

1914–1918	Central Powers	German Empire Austria-Hungary Ottoman Empire Bulgaria

Figure: An example alliance between countries.

- Have a network involving 163 nations (nodes), from the year 1266 to the present day (128 time points).
- This means we need $nT \sim 21,000$ node representations.

Real data: visualising the world stage

Cluster shows Italy and Russia changing their mind before WW1

Year	Country
1882	Russian Empire
1882	Germany
1882	Italy
1887	Germany
1887	Italy
⋮	⋮
1913	Germany
1913	Italy
1914	Germany
1914	Ottoman Empire

Table: One of 60 clusters.

Real data: visualising the world stage

Conclusion

- With stability, you can perform statistical tests for change in network embeddings.
- You can achieve stability using *any* valid single-graph embedding method by embedding the universal unfolded adjacency matrix.

Thank you

- Email: edward.davis@bristol.ac.uk
- Linkedin: https://www.linkedin.com/in/**edwarddavis941**/
- Twitter: @EdD8ta

(a) LinkedIn

(b) Twitter

References

K. Levin, A. Athreya, M. Tang, V. Lyzinski, Y. Park, and C. E. Priebe, "A central limit theorem for an omnibus embedding of multiple random graphs and implications for multiscale network inference," *arXiv preprint arXiv:1705.09355*, 2017.

A. Jones and P. Rubin-Delanchy, "The multilayer random dot product graph," *arXiv preprint arXiv:2007.10455*, 2020.

I. Gallagher, A. Jones, and P. Rubin-Delanchy, "Spectral embedding for dynamic networks with stability guarantees," *Advances in Neural Information Processing Systems*, vol. 34, pp. 10 158–10 170, 2021.