An embedding-based distance for temporal graphs

Lorenzo Dall'Amico

Ciro Cattuto

ISI Foundation

Alain Barrat

cnrs

Aix Marseille université
Initiative d'excellence

Problem statement

Temporal graphs

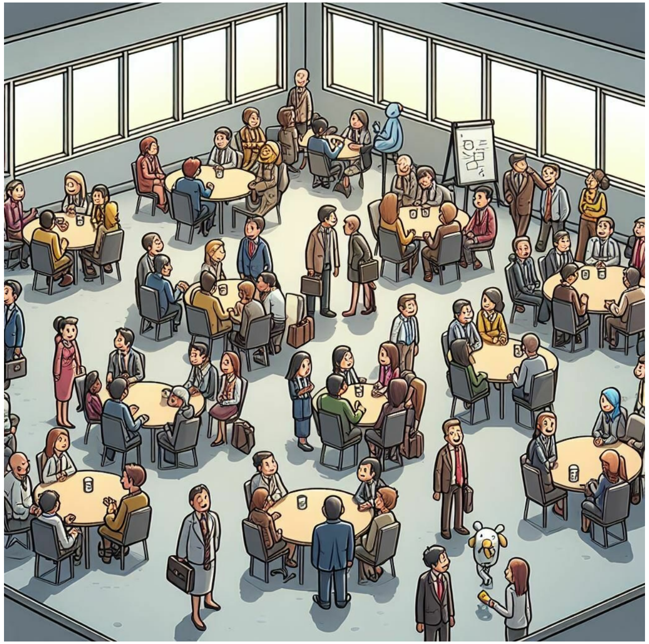
- Snapshot representation
- A set of n nodes $\mathcal{V}$
- A set of temporal edges

$$(i , j , t)$$

first node $i \in \mathcal{V}$

second node $j \in \mathcal{V}$

interaction timestamp



Temporal graphs

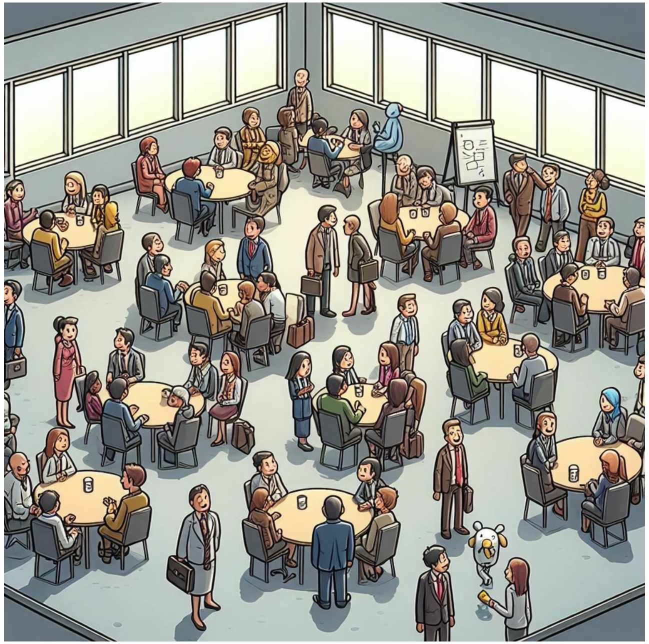
- Snapshot representation
- A set of n nodes $\mathcal{V}$
- A set of temporal edges

$$(i , j , t)$$

first node $i \in \mathcal{V}$

second node $j \in \mathcal{V}$

interaction timestamp



Temporal graphs can encode complex dynamic relations between entities

Question

Can we define a
<u>distance</u> *to compare*
temporal graphs?

Question

Can we define a ***distance*** *to compare temporal graphs?*

Only few related works:

- Bail et al (2023)
- Froese et al (2020),
- Zhanet al (2021)

Desiderata

Desiderata

- Usual distance properties

1. non negativity
2. separation axiom
3. symmetry
4. triangle inequality

Desiderata

- Usual distance properties
  1. non negativity
  2. separation axiom
  3. symmetry
  4. triangle inequality
- Capture topological and temporal structure of the interactions

Desiderata

- Usual distance properties
  1. non negativity
  2. separation axiom
  3. symmetry
  4. triangle inequality

- Capture topological and temporal structure of the interactions

topological → how the nodes are connected

Desiderata

- Usual distance properties
  1. non negativity
  2. separation axiom
  3. symmetry
  4. triangle inequality

- Capture topological and temporal structure of the interactions

  topological → how the nodes are connected

  temporal → when they get connected

Desiderata

- Usual distance properties
  1. non negativity
  2. separation axiom
  3. symmetry
  4. triangle inequality

- Capture topological and temporal structure of the interactions

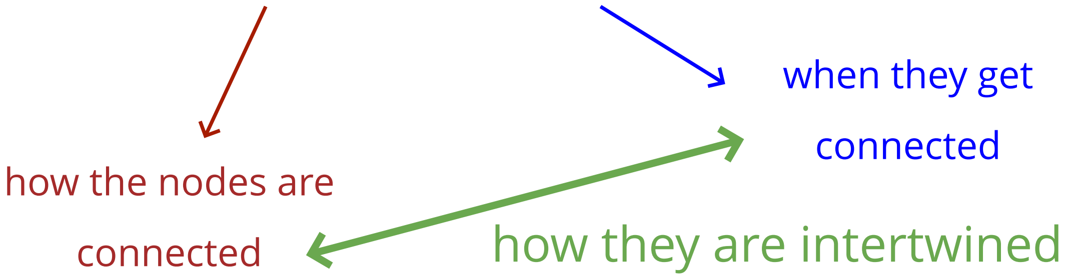
how the nodes are connected

when they get connected

how they are intertwined

Desiderata

- Usual distance properties
  1. non negativity
  2. separation axiom
  3. symmetry
  4. triangle inequality
- Capture topological and temporal structure of the interactions



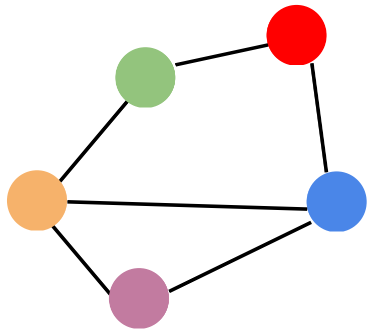
- Can compare graphs with different observation times T

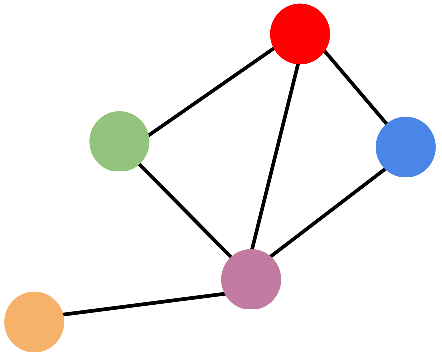
(Un)matched graphs

(Un)matched graphs

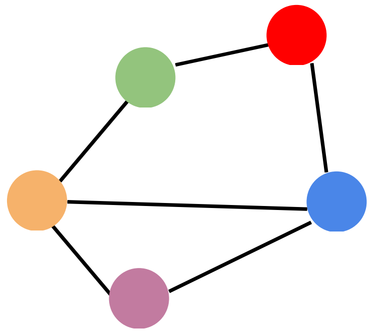
Matched graphs:

a **known** bijection between nodes





(Un)matched graphs

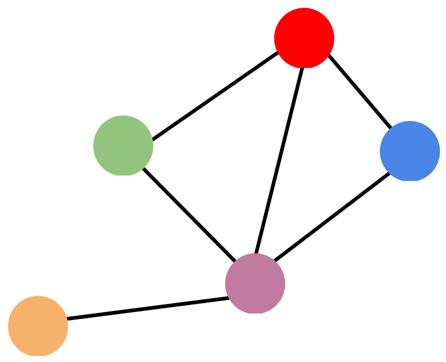


Matched graphs:

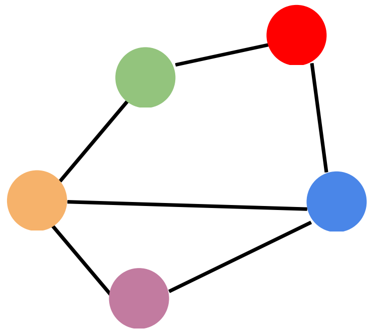
a **known** bijection between nodes

Unmatched graphs:

Also different number of nodes



(Un)matched graphs

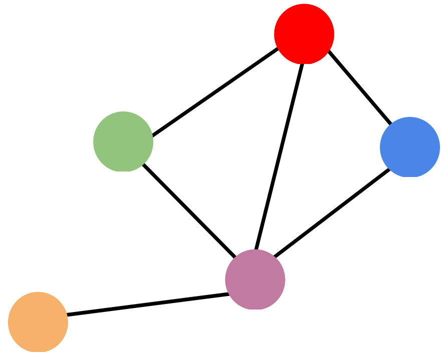


__Matched graphs:__

a **known** bijection between nodes

__Unmatched graphs__:

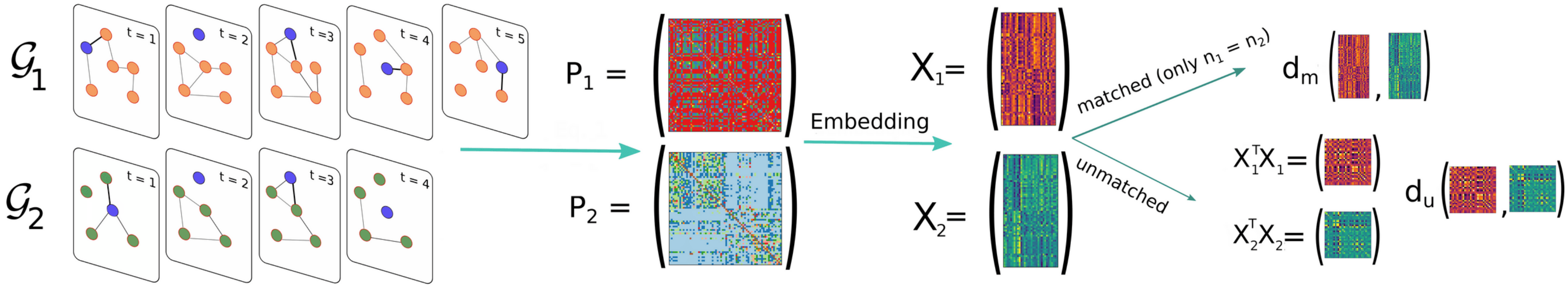
Also different number of nodes



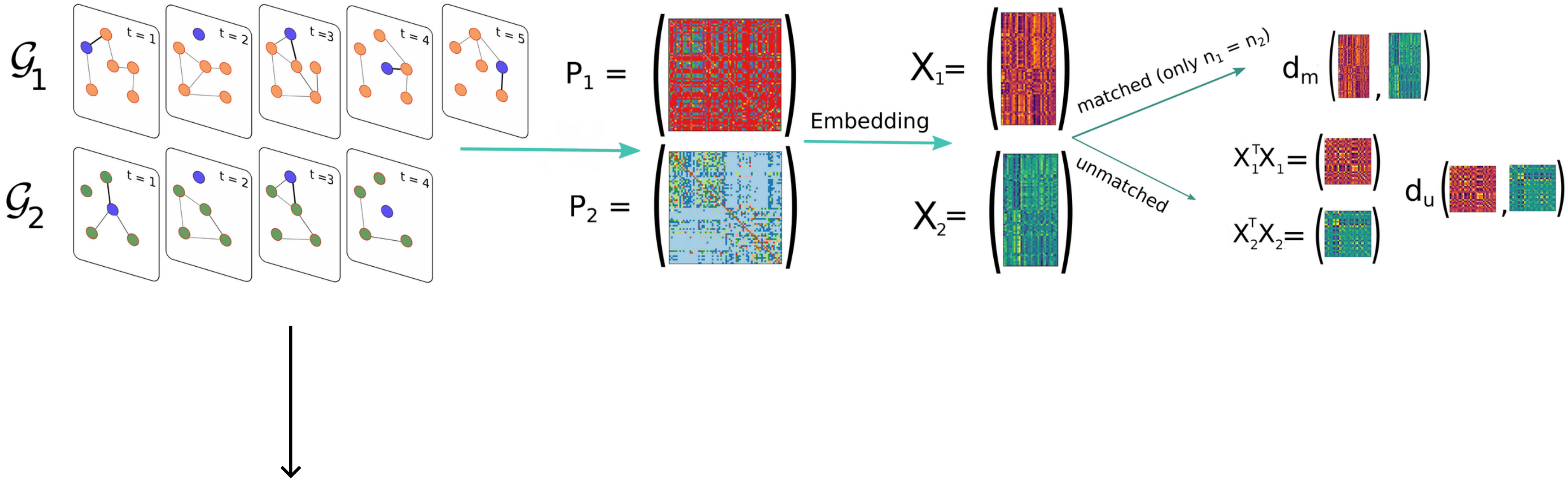
Two distances to handle both cases

Distance definition

The strategy

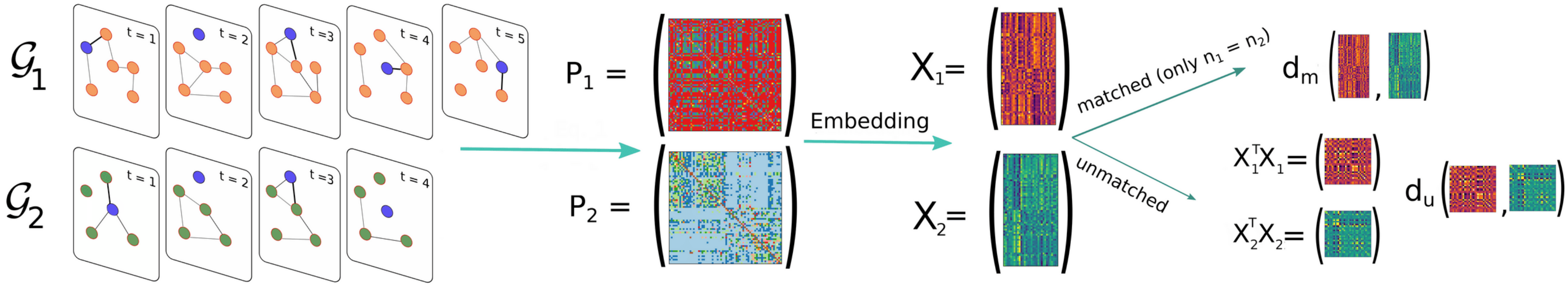


The strategy



input graphs

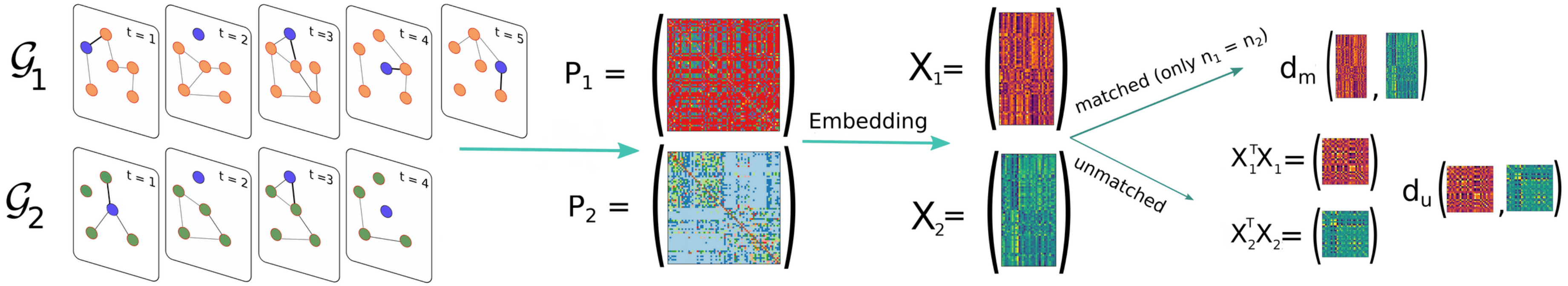
The strategy



input graphs

transition probability matrix

The strategy

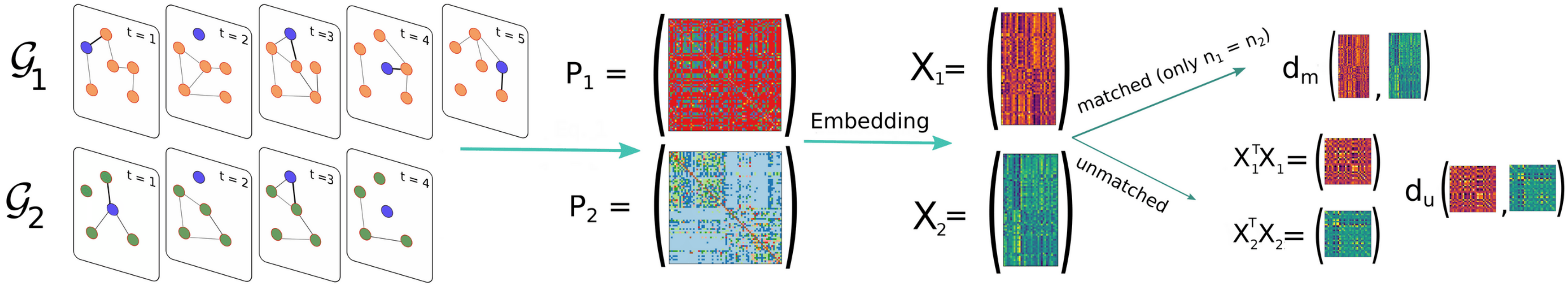


input graphs

transition probability matrix

embedding in vector space

The strategy



input graphs

transition probability matrix

embedding in vector space

distance in embedded space

Transition probability matrix

$$P = \frac{1}{T}\sum_{\tau=1}^{T} L_\tau L_{\tau-1}\cdots L_2 L_1$$

- T : total number of snapshots
- L_τ: row-normalized instantaneous adjaceny matrix with self-loops

P_{ij} *: limiting probability to go from i to j with time-respecting random walks*

Transition probability matrix

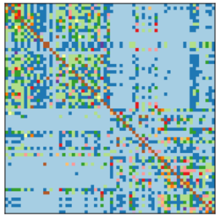
$$P = \frac{1}{T} \sum_{\tau=1}^{T} L_{\tau} L_{\tau-1} \cdots L_2 L_1$$

Depends on tological and temporal network structure

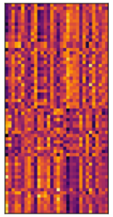
- T : total number of snapshots
- L_{τ}: row-normalized instantaneous adjaceny matrix with self-loops

P_{ij} *: limiting probability to go from i to j with time-respecting random walks*

Embedding

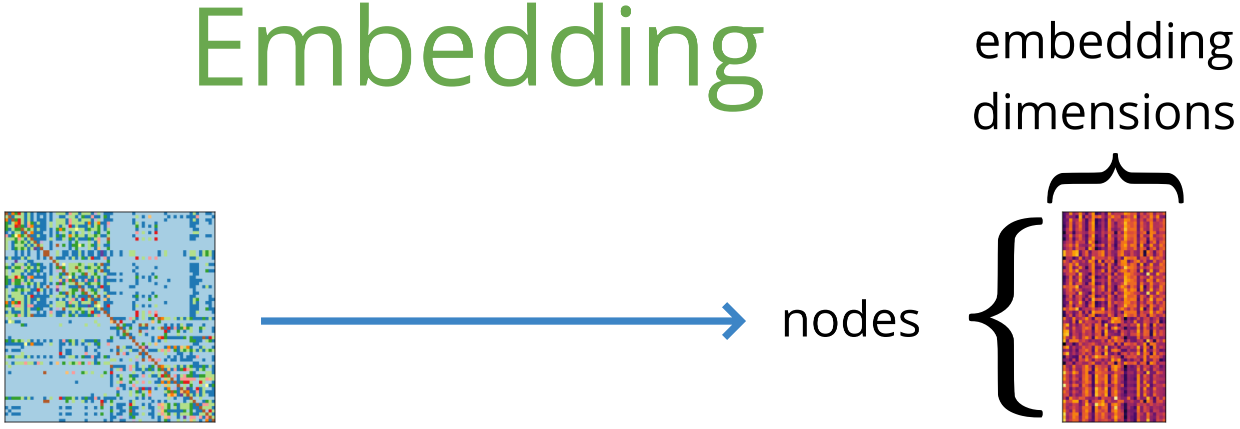


$\boldsymbol{P} \in \mathbb{R}^{n \times n}$



$\boldsymbol{X} \in \mathbb{R}^{n \times d}$

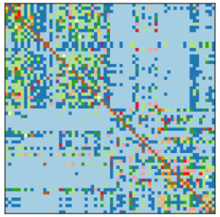
Embedding



$\boldsymbol{P} \in \mathbb{R}^{n \times n}$

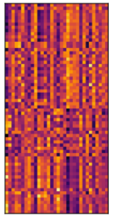
$\boldsymbol{X} \in \mathbb{R}^{n \times d}$

Embedding



embedding dimensions

nodes



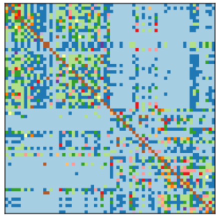
$P \in \mathbb{R}^{n \times n}$ $\qquad$ $X \in \mathbb{R}^{n \times d}$

Minimize

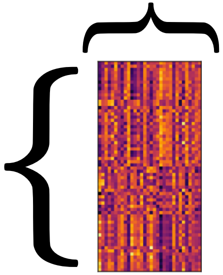
$$\mathcal{L}(X) = -\sum_{i,j \in \mathcal{V}} \left(P_{ij} - \frac{1}{n} \right) X_i^T X_j - \log \underbrace{\left(\sum_{k \in \mathcal{V}} e^{X_i^T X_k} \right)}_{Z_i}$$

Embedding

embedding dimensions



nodes



$P \in \mathbb{R}^{n \times n}$

$X \in \mathbb{R}^{n \times d}$

Minimize

$$\mathcal{L}(X) = -\sum_{i,j \in \mathcal{V}} \left(P_{ij} - \frac{1}{n} \right) X_i^T X_j - \log \underbrace{\left(\sum_{k \in \mathcal{V}} e^{X_i^T X_k} \right)}_{Z_i}$$

Embedding

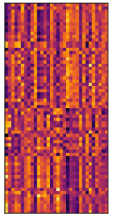
EDRep algorithm

- Fast
- Low dependence on d
- No need to compute P explicitely

https://arxiv.org/abs/2303.17475

embedding dimensions

nodes



$X \in \mathbb{R}^{n \times d}$

Minimize

$$\mathcal{L}(X) = -\sum_{i,j \in \mathcal{V}} \left(P_{ij} - \frac{1}{n} \right) X_i^T X_j - \log \underbrace{\left(\sum_{k \in \mathcal{V}} e^{X_i^T X_k} \right)}_{Z_i}$$

Matched graphs

$$d_{\mathrm{m}} = \overbrace{\|X_1 X_1^T - X_2 X_2^T\|_{\mathrm{F}}}^{\text{expensive}}$$

Entry-wise comparison of similarity between node pairs

Matched graphs

$$d_{\mathrm{m}} = \overbrace{\|X_1 X_1^T - X_2 X_2^T\|_{\mathrm{F}}}^{\text{expensive}}$$

$$= \underbrace{\sqrt{\|X_1^T X_1\|_{\mathrm{F}} + \|X_2^T X_2\|_{\mathrm{F}} - 2\|X_1^T X_2\|_{\mathrm{F}}}}_{\text{cheap}}$$

Entry-wise comparison of similarity between node pairs

Unmatched graphs

$$d_{\mathrm{u}} = \left\| \lambda\left(\frac{X_1^T X_1}{n_1}\right) - \lambda\left(\frac{X_2^T X_2}{n_2}\right) \right\|_2$$

$\lambda \in \mathbb{R}^d$: set of ordered eigenvalues

- *Invariant under node permutations*
- *Independent of graph size*

Evaluation

Method 1

Method 1

1. **Generate** several instances of static **random graphs** of varying size (SBM, ER, CM, GM)

Method 1

1. **Generate** several instances of static **random graphs** of varying size (SBM, ER, CM, GM)
2. Assign each **edge** a **time series** from a real proximity graph (SocioPatterns)

Method 1

1. **Generate** several instances of static **random graphs** of varying size (SBM, ER, CM, GM)
2. Assign each **edge** a **time series** from a real proximity graph (SocioPatterns)
3. Build a **distance matrix D** between each graph pair

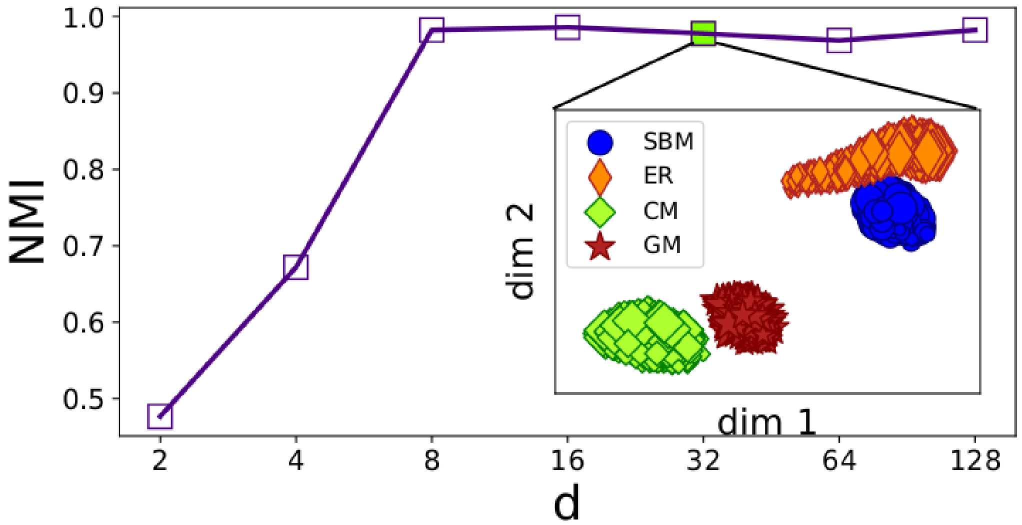
Method 1

1. **Generate** several instances of static **random graphs** of varying size (SBM, ER, CM, GM)
2. Assign each **edge** a **time series** from a real proximity graph (SocioPatterns)
3. Build a **distance matrix D** between each graph pair
4. Run **unsupervised clustering on D**

Method 1

1. **Generate** several instances of static **random graphs** of varying size (SBM, ER, CM, GM)
2. Assign each **edge** a **time series** from a real proximity graph (SocioPatterns)
3. Build a **distance matrix D** between each graph pair
4. Run **unsupervised clustering on D**
5. **Compare** inferred cluster label with known generative model

Method 1



Reconstruction accuracy as function of embedding dimension

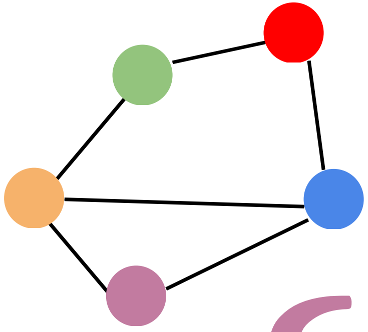
Method 2

Randomizations of real graphs

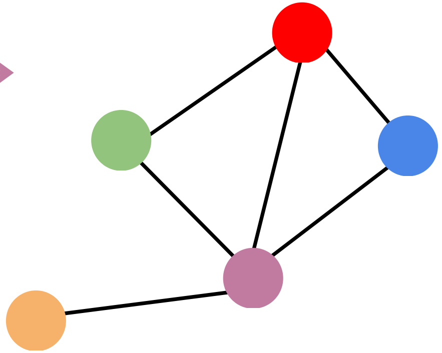
Method 2

Gauvin et al, *Randomized Reference Models for Temporal Networks*

Randomizations of real graphs



Randomization preserving Q



Q preserved

- *Random:* # temporal edges
- *Random delta:* interaction duration distribution
- *Active snapshot:* node activity state, # of edges at t
- *Time:* aggregated graph
- *Sequence:* structure of each snapshot
- *Weighted degree:* # of interactions per node

Method 2

Method 2

1. Consider a **empirical temporal graph**

Method 2

1. Consider a **empirical temporal graph**
2. Generate **N randomizations** preserving Q

Method 2

1. Consider a **empirical temporal graph**
2. Generate **N randomizations** preserving Q
3. For all **pairs of randomizations** build distance matrix **D** between 2N graphs

Method 2

1. Consider a **empirical temporal graph**
2. Generate **N randomizations** preserving Q
3. For all **pairs of randomizations** build distance matrix **D** between 2N graphs
4. Run **unsupervised clustering** on D

Method 2

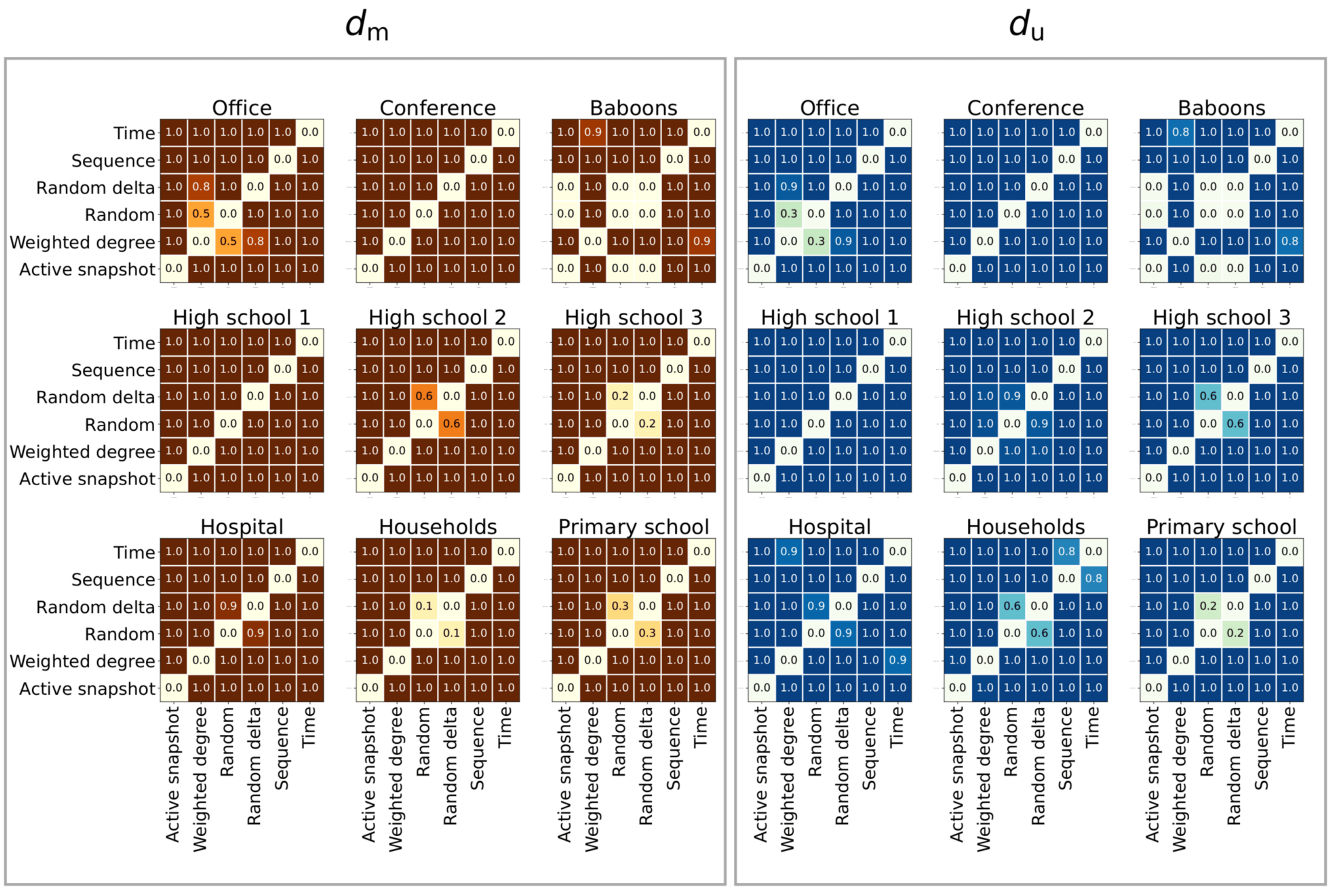
1. Consider a **empirical temporal graph**
2. Generate **N randomizations** preserving Q
3. For all **pairs of randomizations** build distance matrix **D** between 2N graphs
4. Run **unsupervised clustering** on D
5. **Compare** inferred cluster label with randomization type

Method 2

1. Consider a **empirical temporal graph**
2. Generate **N randomizations** preserving Q
3. For all **pairs of randomizations** build distance matrix **D** between 2N graphs
4. Run **unsupervised clustering** on D
5. **Compare** inferred cluster label with randomization type

Distinguishing randomizations means to be sensitive to Q

Method 2



We can distinguish all shufflings pairs for almost all graphs

Conclusion

Contribution

- We introduce a **distance** between temporal graphs
- Both **matched** and **unmacthed** cases
- Disciminate (synthetic, empirical) temporal networks with complex topological and temporal structures

Contribution

- We introduce a **distance** between temporal graphs
- Both **matched** and **unmacthed** cases
- Disciminate (synthetic, empirical) temporal networks with complex topological and temporal structures

Outlook

- Data analysis
- Evaluate generative model

THANK YOU

ISI Foundation

Mail: lorenzo.dallamico@isi.i

Website: lorenzodallamico.github.io

fondation BOTNAR

EPFL COVID-19 Real Time Epidemiology I-DAIR Pathfinder

periscope

European Union's Horizon 2020 No. 101016233

Fondazione CRT

Lagrange project

PAPER

CODE

arXiv:2401.12843