Kindly view the PDF using Adobe Reader to view the animations properly!

Detecting communities in space-time graphs

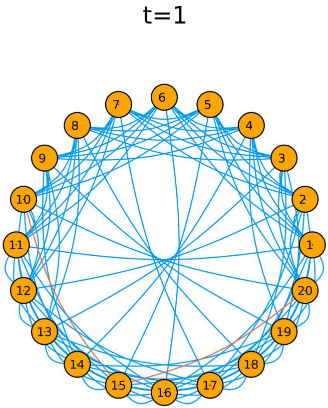
The inflated dynamic Laplacian for temporal networks

Manu Kalia (Freie Universität Berlin)
Joint work with: Gary Froyland (UNSW Sydney)
Péter Koltai (Universität Bayreuth)

SIAM DS 23

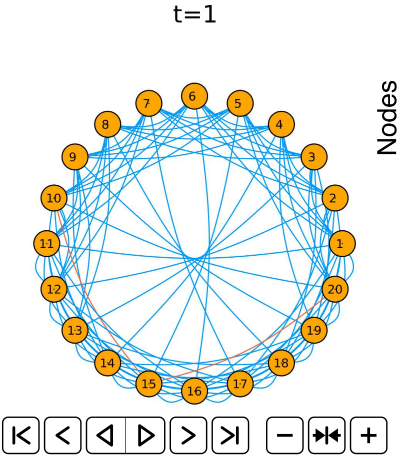
May 14, 2023

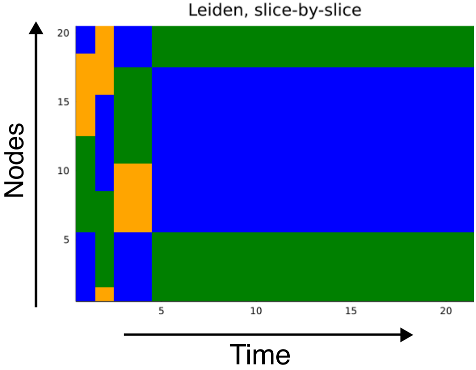
Community detection in temporal networks



- Consider a temporal network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with 20 nodes and edges that evolve in time $t = 1 \dots 21$.
- Network shows transition from 11-regular with no clear communities $\to$ two distinct d-regular communities.
- **Challenge:** How can we detect the single community computationally, without *a priori* information?

Community detection in temporal networks



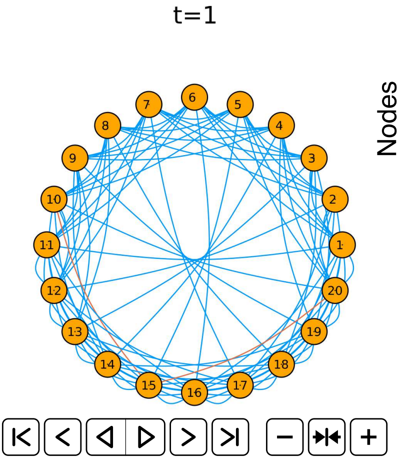


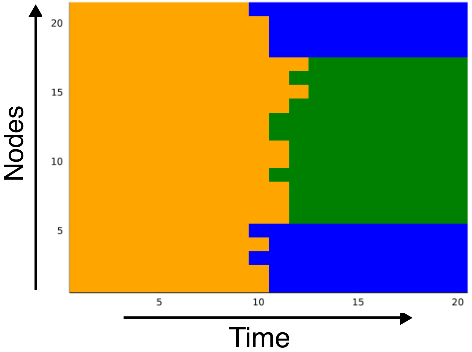
Slice-by-slice identification of communities using *Leiden* [1]

State-of-the-art allocates communities within regular graph!

[1] Traag, Waltman, & van Eck (2019). *Scientific Reports 9-1*.

Community detection in temporal networks





Community detection using the inflated dynamic Laplacian reveals better allocation!

Key idea: Construct the inflated dynamic Laplacian

$$\Delta_{G_0,a}(F(t,x)) = a^2\partial_{tt}F(t,x) + \Delta_{g_t}F(t,x)$$

for temporal networks, and analyze the eigenproblem to detect communities.

The inflated dynamic Laplacian on space-time graphs

- Consider a space-time graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{E}$ is the edge set connecting vertices $\mathcal{V} \subset \mathbb{N} \times \mathbb{N}$.
- Define an edge-weight function $\mathcal{W} : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}_0^+$,

$$\mathcal{W} = ((x,t),(y,s)) \mapsto \begin{cases} \mathcal{W}_{(x,t),(y,s)}, & ((x,t),(y,s)) \in \mathcal{E} \\ 0, & \text{otherwise.} \end{cases}$$

- Now define the supra Laplacian $\mathcal{L} : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}$

$$\mathcal{L} = ((x,t),(y,s)) \mapsto \begin{cases} \mathcal{L}_{(x,t),(y,s)} = \sum_{y,s} \mathcal{W}_{(x,t),(y,s)}, & x = y \text{ and } t = s \\ \mathcal{L}_{(x,t),(y,s)} = -\mathcal{W}_{(x,t),(y,s)} & \text{otherwise.} \end{cases}$$

The inflated dynamic Laplacian on space-time graphs

- Analogous to the continuous setting, the spatial and temporal components of $\mathcal{W}$ and $\mathcal{L}$ can be split as follows,

$$\mathcal{W} = \mathcal{W}^{\text{spat}} + a^2 \mathcal{W}^{\text{temp}}$$
$$\mathcal{L} = \mathcal{L}^{\text{spat}} + a^2 \mathcal{L}^{\text{temp}},$$

- The splitting is achieved by defining

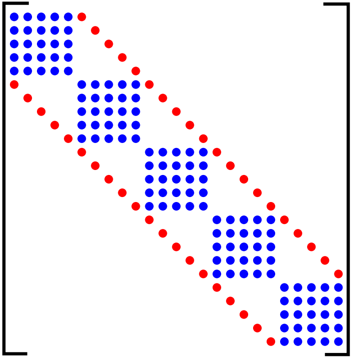
$$\mathcal{W}^{\text{spat}}_{(x,t),(y,s)} = 0, \; t \neq s,$$
$$\mathcal{W}^{\text{temp}}_{(x,t),(y,s)} = 0, \; t = s.$$

Temporal network structure of $\mathcal{W}$

- **Spatial:** $\mathcal{W}^{\text{spat}} = \bigoplus_t \mathcal{W}_{(x,t),(y,t)}$
- **Temporal**: Given a $T \times T$ matrix W^{temp},

$$\mathcal{W}^{\text{temp}} = W^{\text{temp}} \otimes I.$$

- Example: W^{temp} has nonzero terms on super- and subdiagonal only $\rightarrow$ temporal network!



$$\mathcal{W} = \mathcal{W}^{\text{spat}} + a^2 \mathcal{W}^{\text{temp}}$$

How does the inflated dynamic Laplacian $\mathcal{L}$ relate to graph partitioning?
Spectral partitioning and Cheeger constants.

Balanced graph cuts and Cheeger constants

- Consider a pairwise disjoint space-time partition $\mathcal{X}_1 \dots \mathcal{X}_K$ of $\mathcal{V}$. Cheeger constant h_K determines quality of partition,

$$h_K = \min_{\mathcal{X}_1 \dots \mathcal{X}_K} \max_k \frac{\mathrm{cut}(\mathcal{X}_k, \overline{\mathcal{X}_k})}{\min\{|\mathcal{X}_k|, |\overline{\mathcal{X}_k}|\}}.$$

- $\mathrm{cut}(X, Y) =$ sum of cut edge-weights between X and Y.
- Classical Cheeger inequality [1,2],

$$h_2 \leq \sqrt{2\lambda_2}$$
$$h_K \leq 2^{3/2} K^2 \sqrt{\lambda_K}$$

- λ_K is the K-th smallest eigenvalue of $\mathcal{L}$.

---

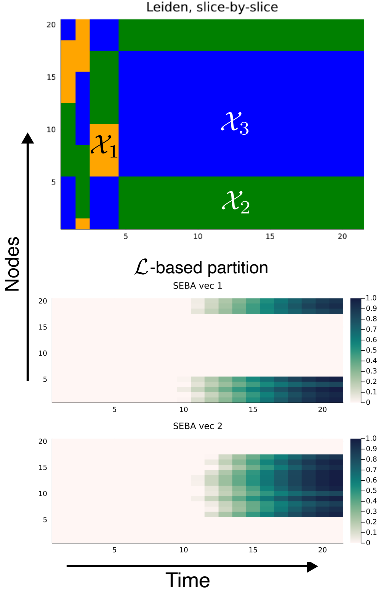
[1] Chung (1996). Laplacians of graphs and cheeger's inequalities. *Combinatorics, Paul Erdos is Eighty*.

[2] Lee, Gharan, Trevisan (2014). *J. ACM.*

$\mathcal{L}$ versus slice-by-slice

- How well does $\mathcal{L}$-based partitioning perform against slice-by-slice? Example $\rightarrow$ 3-partition in intro graph.
- Consider slice-by-slice partitioning, marked as $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{X}_3$ with Cheeger constant h_3^{slice}.
- Spectral partitioning with $\mathcal{L}$ gives a 3-partition. Let h_3 be the associated Cheeger constant
- In general we prove that

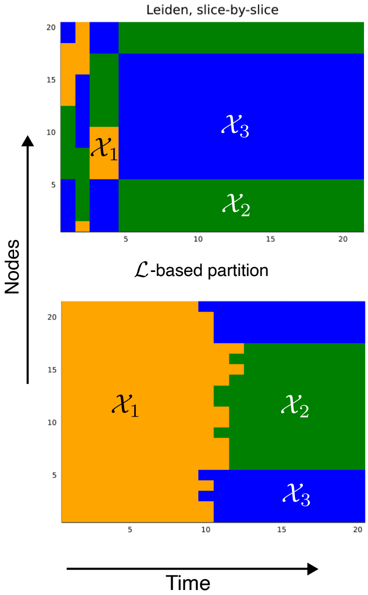
$$h_K \leq h_K^{\text{slice}}$$



$\mathcal{L}$ versus slice-by-slice

- How well does $\mathcal{L}$-based partitioning perform against slice-by-slice? Example $\rightarrow$ 3-partition in intro graph.
- Consider slice-by-slice partitioning, marked as $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{X}_3$ with Cheeger constant h_3^{slice}.
- Spectral partitioning with $\mathcal{L}$ gives a 3-partition. Let h_3 be the associated Cheeger constant
- In general we prove that

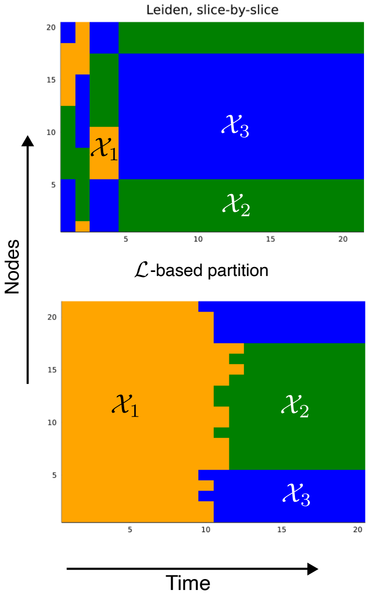
$$h_K \leq h_K^{\text{slice}}$$



$\mathcal{L}$ versus slice-by-slice

- How well does $\mathcal{L}$-based partitioning perform against slice-by-slice? Example $\rightarrow$ 3-partition in intro graph.
- Consider slice-by-slice partitioning, marked as $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{X}_3$ with Cheeger constant h_3^{slice}.
- Spectral partitioning with $\mathcal{L}$ gives a 3-partition. Let h_3 be the associated Cheeger constant
- In general we prove that

$$h_K \leq h_K^{\text{slice}}$$



Cheeger constants are bounded by k-smallest eigenvalues λ_k of $\mathcal{L}$.
How do λ_k behave?

Characterizing the spectrum of $\mathcal{L}$

- Eigenvalues $\lambda_{k,a}$ are either **(spatial)** $\lambda_{k,a}^{\text{spat}}$ or **(temporal)** $\lambda_{k,a}^{\text{temp}}$.
- Recall: W^{temp} is a $T \times T$ matrix with nonzero terms on supra/sub diagonal only. Let L^{temp} be its Laplacian.
- Temporal eigenvalues have the structure $\boldsymbol{\lambda_{k,a}^{temp} = a^2 \nu_k}$ where ν_k are eigenvalues of Laplacian L^{temp} .
- No simple structure for $\lambda_{k,a}^{\text{spat}}$.
- $\lim_{a\to\infty} \lambda_{k,a}^{\text{temp}} = \infty$, what happens to $\lambda_{k,a}^{\text{spat}}$?

Characterizing the spectrum of $\mathcal{L}$

- Eigenvalues $\lambda_{k,a}$ are either **(spatial)** $\lambda_{k,a}^{\text{spat}}$ or **(temporal)** $\lambda_{k,a}^{\text{temp}}$.
- Recall: W^{temp} is a $T \times T$ matrix with nonzero terms on supra/sub diagonal only. Let L^{temp} be its Laplacian.
- Temporal eigenvalues have the structure $\boldsymbol{\lambda}_{\mathbf{k,a}}^{\mathbf{temp}} = \mathbf{a^2}\boldsymbol{\nu}_{\mathbf{k}}$ where ν_k are eigenvalues of Laplacian L^{temp} .
- No simple structure for $\lambda_{k,a}^{\text{spat}}$.
- $\lim_{a\to\infty} \lambda_{k,a}^{\text{temp}} = \infty$, what happens to $\lambda_{k,a}^{\text{spat}}$?

Characterizing the spectrum of $\mathcal{L}$

Theorem

Let N be the number of vertices per time fiber. Then for the first N spatial eigenvalues $\lambda_{k,a}^{\text{spat}}$ we have,

$$\lim_{a\to\infty} \lambda_{k,a}^{\text{spat}} = \lambda_k^D, \text{ where } \mathcal{L}^D f_k^D = \lambda_k^D f_k^D.$$

$\mathcal{L}^D$ is the dynamic Laplacian,

$$\mathcal{L}^D = \frac{1}{T} \sum_t L_t^{\text{spat}}.$$

Note: L_t^{spat} is the Laplacian corresponding to W_t^{spat} for fixed t.

First N eigenvalues remain finite even in the a-limit!

Algorithm for community detection in temporal networks

(1) For fixed a, compute $\mathcal{L} = \mathcal{L}^{\text{spat}} + a^2\mathcal{L}^{\text{temp}}$. Find $a = a_{\text{crit}} \rightarrow$ transition from leading temporal to leading spatial eigenvalue.

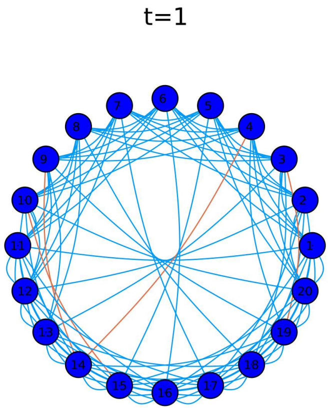
(2) Estimate K ($\#$ partitions) $\rightarrow$ Ex. using spectral gap theorem from [1].

(3) Perform SEBA (Sparse Eigenbasis Approximation) [2] on eigenvectors $f^{\text{spat}}_{k,a_{\text{crit}}}$, which define clusters.

[1] Lee, Gharan & Trevisan (2014). *J. ACM.*

[2] Froyland, Rock & Sakellariou (2019). *Commun Nonlinear Sci Numer Simulat*.

Example: d-regular → 2 clusters → 3 clusters

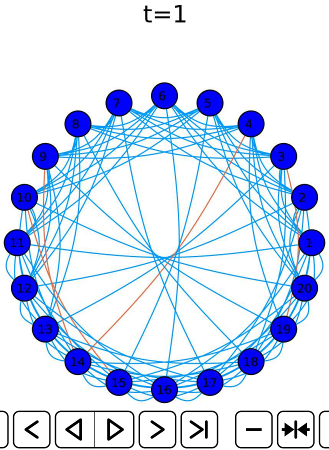


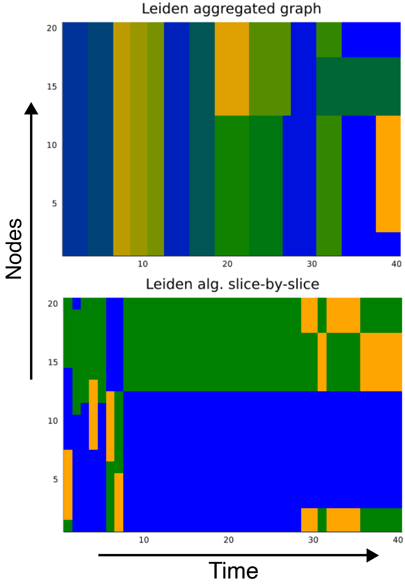
Graphs generated to show the following edge dynamics

- **Appearance**
- **Disappearance**
- **Merge**
- **Split**

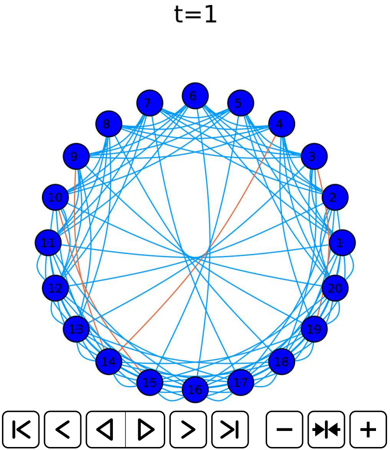
← Graph with 20 nodes, evolving over 40 time steps. Transition: **regular graph** splits to **2 clusters** which further splits to **3 clusters**.

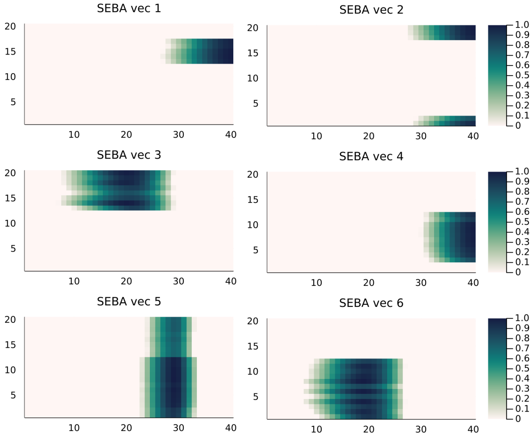
Example: d-regular → 2 clusters → 3 clusters





Example: d-regular → 2 clusters → 3 clusters

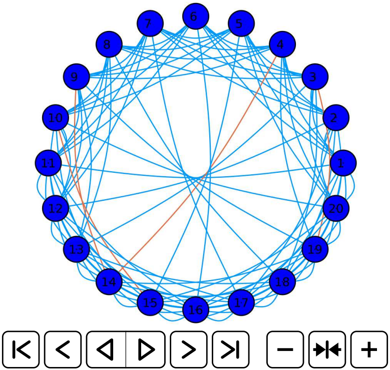


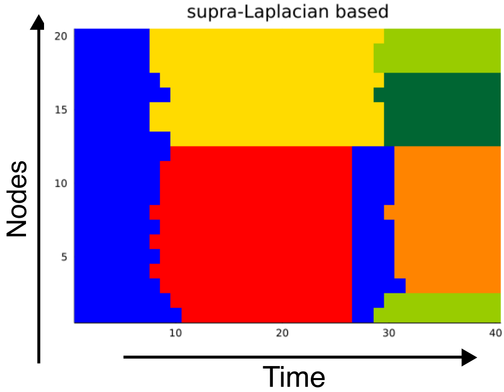


Note 'bad' SEBA vector 5.

Example: d-regular $\rightarrow$ 2 clusters $\rightarrow$ 3 clusters

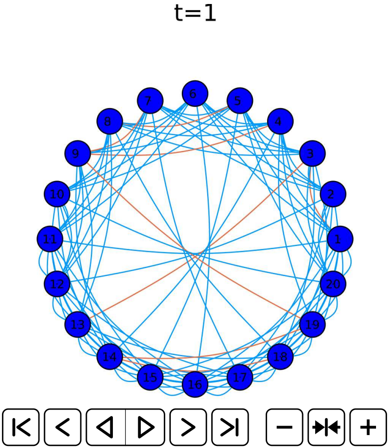
t=1





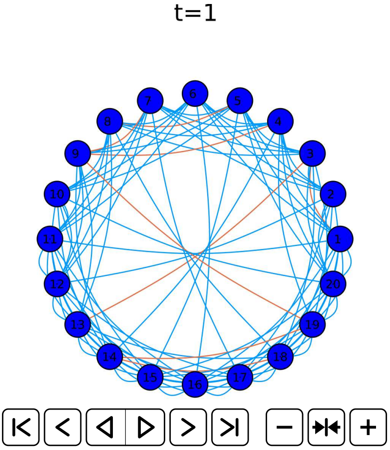
Clustering done without SEBA vector 5.

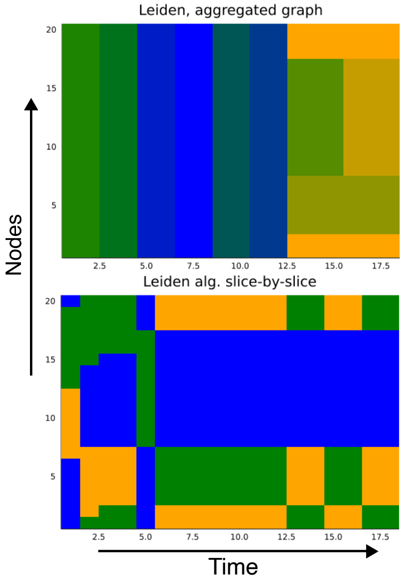
Example: d-regular $\rightarrow$ 2 clusters



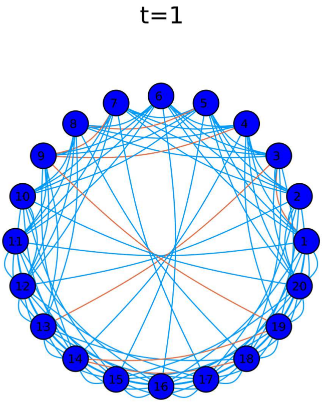
$\leftarrow$ Graph with 20 nodes, evolving over 18 time steps. Transition: Emergence of **3 clusters** from a **regular graph**.

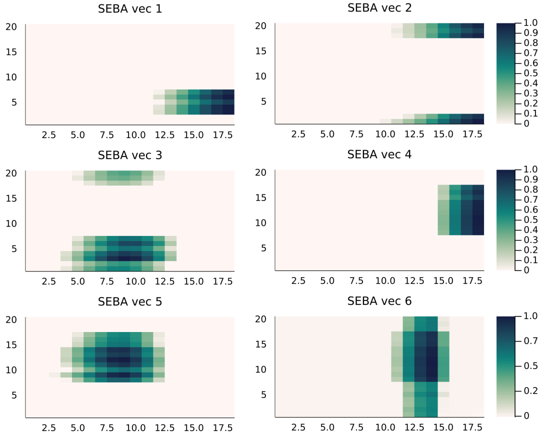
Example: d-regular $\to$ 2 clusters





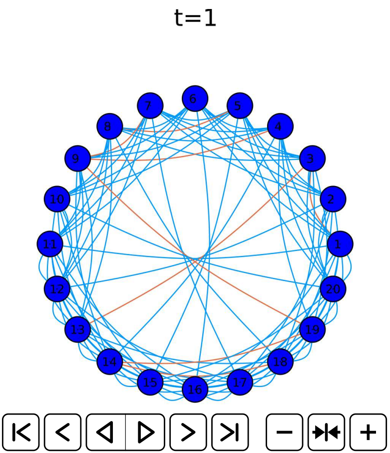
Example: d-regular $\to$ 2 clusters

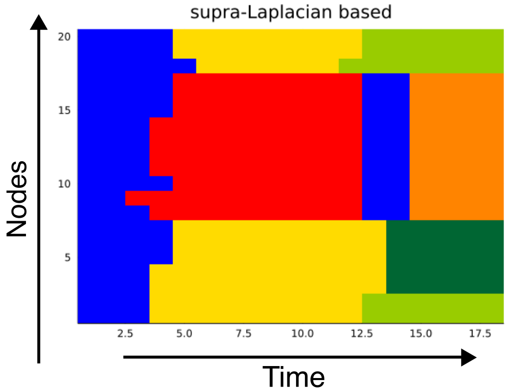




Note ‘bad’ SEBA vector 6.
Intermediate split visible (vecs 3,5).

Example: d-regular $\to$ 2 clusters





Clustering done without SEBA vector 6.

Outlook

- **Summary:** Spectral clustering with the inflated dynamic Laplacian reveals better balanced cuts in temporal networks comparted to state-of-the-art.
- **Challenge:** Finding 'optimal' a by considering $a \equiv a(x,t)$ and formulating an appropriate minimisation problem.
- **Future work:** Continuum limit: Constructing edge-based dynamical system to compare with continuous case.

Questions?