Sparsification of the Magnetic Laplacian and A CyclePopping Random Walk

Michaël Fanuel
joint work with Rémi Bardenet

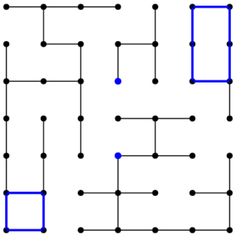
Université de Lille, CNRS, Centrale Lille
UMR 9189 – CRIStAL, F-59000 Lille, France

WINQ 29th April - 3rd May 2024
Week 1 - Dynamics and Topology of Complex Network Systems

Related paper 1: Complex valued graph Laplacian.

In this talk.

- M. Fanuel and R. Bardenet, Sparsification of the Regularized Magnetic Laplacian with Multi-Type Spanning Forests, arxiv 2208.14797



Related paper 2: Monte-Carlo estimator for inverse Laplacian.

Not in this talk.

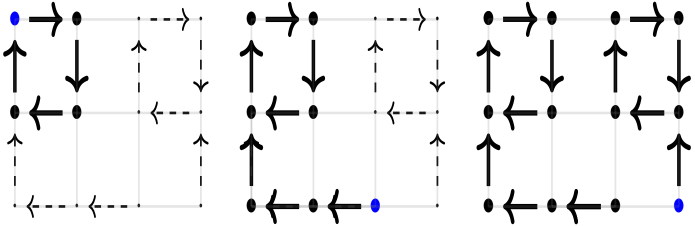
- H. Jaquard, M. Fanuel, P.-O. Amblard, R. Bardenet, S. Barthelmé, N. Tremblay, Smoothing Complex-Valued Signals on Graphs with Monte-Carlo, International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2023.

$$(\Delta + q\mathbb{I})^{-1} = \mathbb{E}_{\text{ forest } \mathcal{F}}[\text{estimator}(\mathcal{F})]$$

Related paper 3: Random walk with cyclepopping.

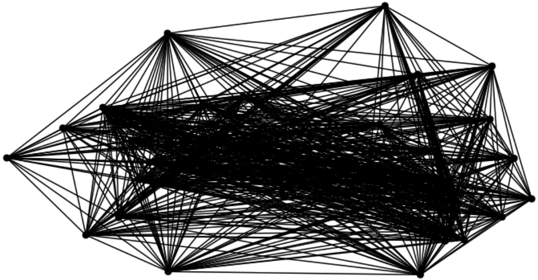
In this talk.

- M. Fanuel and R. Bardenet, On the Number of Steps of CyclePopping in Weakly Inconsistent U(1)-Connection Graphs,
  arxiv 2404.14803



Sparsification setting

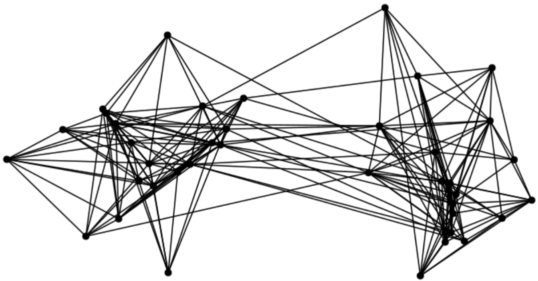
Connected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$.



In this talk, all the edge weights of $\mathcal{G}$ are equal to 1.

Goal

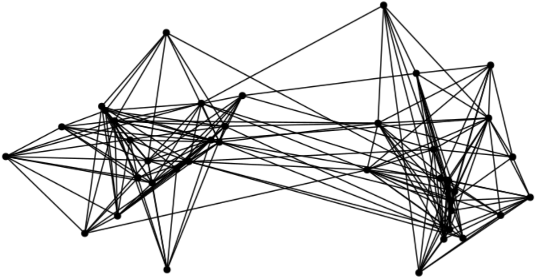
We aim to find a sparse approximation of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.



To do so, we sample edges in $\mathcal{E}$ and give them positive weights.

Goal

We aim to find a sparse approximation of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

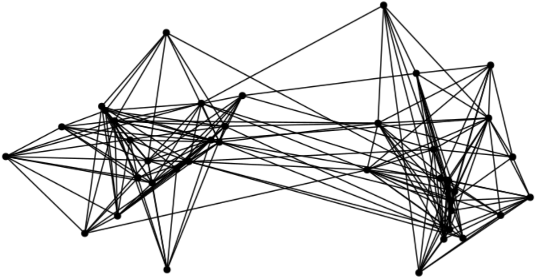


To do so, we sample edges in $\mathcal{E}$ and give them positive weights.

Actually, we approximate a **graph Laplacian** of $\mathcal{G}$.

Goal

We aim to find a sparse approximation of $\mathcal{G} = (\mathcal{V}, \mathcal{E})$.

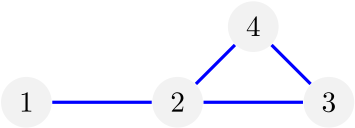


To do so, we sample edges in $\mathcal{E}$ and give them positive weights.

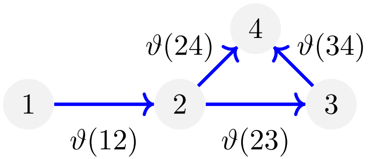
Actually, we approximate a **graph Laplacian** of $\mathcal{G}$.

We consider a case where edges come with extra information.

U(1)-connection graph

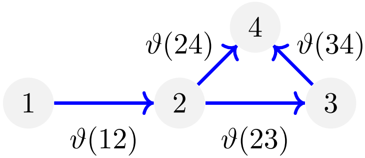


U(1)-connection graph



$\forall$ oriented edge uv, we have an **angle** $\vartheta(uv)$ s.t. $\vartheta(vu) = -\vartheta(uv)$.

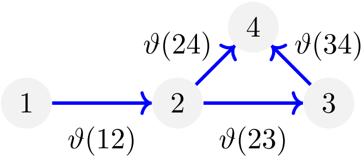
U(1)-connection graph



$\forall$ oriented edge uv, we have an **angle** $\vartheta(uv)$ s.t. $\vartheta(vu) = -\vartheta(uv)$.

Synchronization of the nodes:
Can we find h_u for $u \in \mathcal{V}$ s.t. $\vartheta(uv) \approx (h_u - h_v) \mod 2\pi$?

U(1)-connection graph

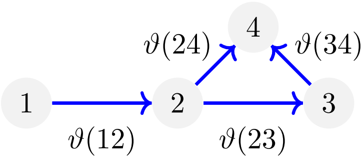


$\forall$ oriented edge uv, we have an **angle** $\vartheta(uv)$ s.t. $\vartheta(vu) = -\vartheta(uv)$.

Synchronization of the nodes:
Can we find h_u for $u \in \mathcal{V}$ s.t. $\vartheta(uv) \approx (h_u - h_v) \mod 2\pi$?

- Angular synchronization problem (cryo-electron microscopy, Singer 2011).
- Robust ranking from pairwise comparisons (Cucuringu 2016).

U(1)-connection graph



$\forall$ oriented edge uv, we have an **angle** $\vartheta(uv)$ s.t. $\vartheta(vu) = -\vartheta(uv)$.

Synchronization of the nodes:
Can we find h_u for $u \in \mathcal{V}$ s.t. $\vartheta(uv) \approx (h_u - h_v) \mod 2\pi$?

- Angular synchronization problem (cryo-electron microscopy, Singer 2011).
- Robust ranking from pairwise comparisons (Cucuringu 2016).

Magnetic Laplacian Δ associated with this connection graph.

Motivation for a sparsifier

Originally for solving Laplacian systems (e.g. Spielman and Srivastava, 2011)

Solving linear systems of the form

$$(\Delta + q\mathbb{I}_n)\boldsymbol{f} = q\boldsymbol{y},$$

where Δ is a Laplacian and $q \geq 0$, which originates e.g. from semi-supervised learning

$$\min_{\boldsymbol{f}} \boldsymbol{f}^* \Delta \boldsymbol{f} + q\|\boldsymbol{f} - \boldsymbol{y}\|_2^2.$$

Motivation for a sparsifier

Originally for solving Laplacian systems (e.g. Spielman and Srivastava, 2011)

Solving linear systems of the form

$$(\Delta + q\mathbb{I}_n)\boldsymbol{f} = q\boldsymbol{y},$$

where Δ is a Laplacian and $q \geq 0$, which originates e.g. from semi-supervised learning

$$\min_{\boldsymbol{f}} \boldsymbol{f}^* \Delta \boldsymbol{f} + q\|\boldsymbol{f} - \boldsymbol{y}\|_2^2.$$

The difficulty/sensitivity of this numerical problem

$$\mathrm{cond}(\Delta + q\mathbb{I}_n) \triangleq \frac{\lambda_{\max}(\Delta + q\mathbb{I}_n)}{\lambda_{\min}(\Delta + q\mathbb{I}_n)}.$$

Motivation for a sparsifier

Originally for solving Laplacian systems (e.g. Spielman and Srivastava, 2011)

Solving linear systems of the form

$$(\Delta + q\mathbb{I}_n)\boldsymbol{f} = q\boldsymbol{y},$$

where Δ is a Laplacian and $q \geq 0$, which originates e.g. from semi-supervised learning

$$\min_{\boldsymbol{f}} \boldsymbol{f}^* \Delta \boldsymbol{f} + q\|\boldsymbol{f} - \boldsymbol{y}\|_2^2.$$

The difficulty/sensitivity of this numerical problem

$$\operatorname{cond}(\Delta + q\mathbb{I}_n) \triangleq \frac{\lambda_{\max}(\Delta + q\mathbb{I}_n)}{\lambda_{\min}(\Delta + q\mathbb{I}_n)}.$$

If $\widetilde{\Delta} + q\mathbb{I}_n$ is a (sparse) approximation of $\Delta + q\mathbb{I}_n$, the system

$$(\widetilde{\Delta} + q\mathbb{I}_n)^{-1}(\Delta + q\mathbb{I}_n)\boldsymbol{f} = (\widetilde{\Delta} + q\mathbb{I}_n)^{-1}\boldsymbol{b},$$

is expected to have a smaller condition number.

Motivation for a sparsifier

Originally for solving Laplacian systems (e.g. Spielman and Srivastava, 2011)

Solving linear systems of the form

$$(\Delta + q\mathbb{I}_n)\boldsymbol{f} = q\boldsymbol{y},$$

where Δ is a Laplacian and $q \geq 0$, which originates e.g. from semi-supervised learning

$$\min_{\boldsymbol{f}} \boldsymbol{f}^* \Delta \boldsymbol{f} + q\|\boldsymbol{f} - \boldsymbol{y}\|_2^2.$$

The difficulty/sensitivity of this numerical problem

$$\mathrm{cond}(\Delta + q\mathbb{I}_n) \triangleq \frac{\lambda_{\max}(\Delta + q\mathbb{I}_n)}{\lambda_{\min}(\Delta + q\mathbb{I}_n)}.$$

If $\widetilde{\Delta} + q\mathbb{I}_n$ is a (sparse) approximation of $\Delta + q\mathbb{I}_n$, the system

$$(\widetilde{\Delta} + q\mathbb{I}_n)^{-1}(\Delta + q\mathbb{I}_n)\boldsymbol{f} = (\widetilde{\Delta} + q\mathbb{I}_n)^{-1}\boldsymbol{b},$$

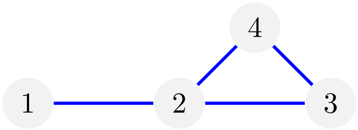
is expected to have a smaller condition number.

NB: There is a technicality if $q = 0$ and Δ is the combinatorial Laplacian.

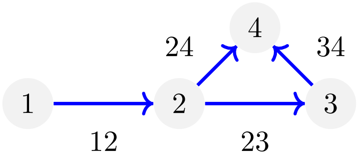
Outline

1. Combinatorial Laplacian and sparsification
2. Magnetic Laplacian and sparsification
3. Sampling edges with a loop-erased random walk
4. Numerical simulations

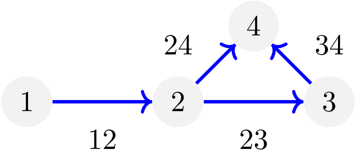
Combinatorial Laplacian and sparsification


4
1
2
3

Fix edge orientations.



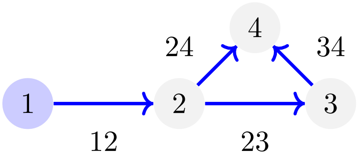
Fix edge orientations.



Edge-vertex incidence matrix $(m \times n)$ s.t. row uv is $(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^\top$.

$$B_0 = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array} \right] \end{array}.$$

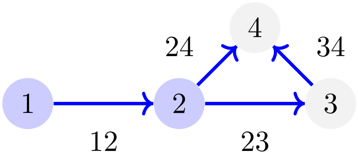
Fix edge orientations.



Edge-vertex incidence matrix $(m \times n)$ s.t. row uv is $(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^\top$.

$$B_0 = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 1 & & & \\ 0 & & & \\ 0 & & & \\ 0 & & & \end{array} \right] \end{array}.$$

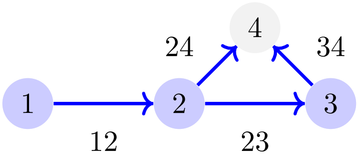
Fix edge orientations.



Edge-vertex incidence matrix $(m \times n)$ s.t. row uv is $(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^\top$.

$$B_0 = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 1 & -1 & & \\ 0 & 1 & & \\ 0 & 0 & & \\ 0 & 1 & & \end{array} \right] \end{array}.$$

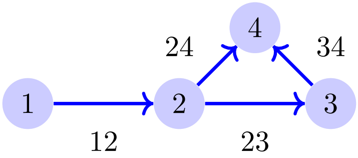
Fix edge orientations.



Edge-vertex incidence matrix $(m \times n)$ s.t. row uv is $(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^\top$.

$$B_0 = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 1 & -1 & 0 & \\ 0 & 1 & -1 & \\ 0 & 0 & 1 & \\ 0 & 1 & 0 & \end{array} \right] \end{array}.$$

Fix edge orientations.



Edge-vertex incidence matrix $(m \times n)$ s.t. row uv is $(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^\top$.

$$B_0 = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 1 & 0 & -1 \end{bmatrix} \end{array}.$$

Combinatorial Laplacian

Combinatorial Laplacian:

$$L = B_0^* B_0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \end{array}.$$

Combinatorial Laplacian

Combinatorial Laplacian:

$$L = B_0^* B_0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \end{array}.$$

- $\boldsymbol{f}^* L \boldsymbol{f} \propto \sum_{uv \in \mathcal{E}} |f(u) - f(v)|^2$

Combinatorial Laplacian

Combinatorial Laplacian:

$$L = B_0^* B_0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \end{array}.$$

- $\boldsymbol{f}^* L \boldsymbol{f} \propto \sum_{uv \in \mathcal{E}} |f(u) - f(v)|^2$
- Let $\mathcal{G}$ be connected. We have $\text{null}(L) = \text{span}(\mathbf{1})$.

Combinatorial Laplacian

Combinatorial Laplacian:

$$L = B_0^* B_0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 3 & -1 & -1 \\ 0 & -1 & 2 & -1 \\ 0 & -1 & -1 & 2 \end{bmatrix} \end{array}.$$

- $\boldsymbol{f}^* L \boldsymbol{f} \propto \sum_{uv \in \mathcal{E}} |f(u) - f(v)|^2$
- Let $\mathcal{G}$ be connected. We have $\text{null}(L) = \text{span}(\mathbf{1})$.
- Classical decomposition:

$$L = D - W$$

with $D = \text{Diag}(\text{deg})$ and $\text{deg}(u) = \text{nb of neighbors of } u \in \mathcal{V}$.

Sparsification: edge sampling

Recall

$$L = B_0^* B_0 \text{ with } B_0 \in \mathbb{R}^{m \times n}.$$

Sparsification: edge sampling

Recall

$$L = B_0^* B_0 \text{ with } B_0 \in \mathbb{R}^{m \times n}.$$

Graph Laplacian

$$L = \sum_{\text{edge } uv \in \mathcal{E}} \overset{\text{column}}{(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)} \overset{\text{row}}{(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^*} .$$

Sparsification: edge sampling

Recall

$$L = B_0^* B_0 \text{ with } B_0 \in \mathbb{R}^{m \times n}.$$

Graph Laplacian

$$L = \sum_{\text{edge } uv \in \mathcal{E}} \overbrace{(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)}^{\text{column}} \overbrace{(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^*}^{\text{row}} \; .$$

Sparsify: take $\mathcal{S} \in \mathcal{E}$,

$$\widetilde{L}(\mathcal{S}) = \sum_{\text{edge } uv \in \mathcal{S}} \overbrace{\widetilde{w}_{uv}}^{\text{weight} > 0} (\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^*.$$

Sparsification: edge sampling

Recall

$$L = B_0^* B_0 \text{ with } B_0 \in \mathbb{R}^{m\times n}.$$

Graph Laplacian

$$L = \sum_{\text{edge } uv \in \mathcal{E}} \overbrace{(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)}^{\text{column}} \overbrace{(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^*}^{\text{row}} .$$

Sparsify: take $\mathcal{S} \in \mathcal{E}$,

$$\widetilde{L}(\mathcal{S}) = \sum_{\text{edge } uv \in \mathcal{S}} \overbrace{\widetilde{w}_{uv}}^{\text{weight} > 0} (\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)(\boldsymbol{\delta}_u - \boldsymbol{\delta}_v)^*.$$

$\widetilde{L}(\mathcal{S})$ is obtained by sampling & reweighting **rows** of B_0.

$(1 \pm \epsilon)$ multiplicative approximation

Loewner order

Let X, Y be $m \times m$ Hermitian matrices. We have

$$X \preceq Y \text{ iff } \boldsymbol{f}^* X \boldsymbol{f} \leq \boldsymbol{f}^* Y \boldsymbol{f} \text{ for all } \boldsymbol{f} \in \mathbb{C}^m.$$

$(1 \pm \epsilon)$ multiplicative approximation

Loewner order

Let X, Y be $m \times m$ Hermitian matrices. We have

$$X \preceq Y \text{ iff } \boldsymbol{f}^* X \boldsymbol{f} \leq \boldsymbol{f}^* Y \boldsymbol{f} \text{ for all } \boldsymbol{f} \in \mathbb{C}^m.$$

Let $\epsilon > 0$. How do we sample a set of edges $\mathcal{S}$ such that

$$(1 - \epsilon)L \preceq \widetilde{L}(\mathcal{S}) \preceq (1 + \epsilon)L$$

occurs *with high probability*?

$(1 \pm \epsilon)$ multiplicative approximation

Loewner order

Let X, Y be $m \times m$ Hermitian matrices. We have

$$X \preceq Y \text{ iff } \boldsymbol{f}^* X \boldsymbol{f} \leq \boldsymbol{f}^* Y \boldsymbol{f} \text{ for all } \boldsymbol{f} \in \mathbb{C}^m.$$

Let $\epsilon > 0$. How do we sample a set of edges $\mathcal{S}$ such that

$$(1-\epsilon)L \preceq \widetilde{L}(\mathcal{S}) \preceq (1+\epsilon)L$$

occurs *with high probability*?

We wish to have as few edges as possible.

A spanning tree is a connected spanning subgraph without cycle.

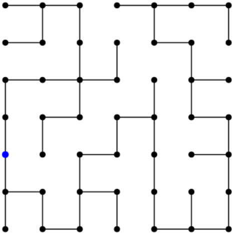


Figure: A spanning tree of a 7×7 square grid.

A spanning tree is a connected spanning subgraph without cycle.

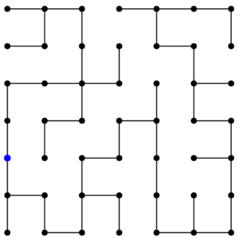


Figure: A spanning tree of a 7×7 square grid.

Uniform measure. For all spanning tree $\mathcal{S}$

$$\mathbb{P}_{\mathrm{ST}}(\mathcal{S}) = \frac{1}{\det L_{\hat{r}}}.$$

Theorem (Kaufman, Kyng, Solda (2022))

Let $\delta \in (0,1)$. There exists a sparsifier $\widetilde{L}_t$ built with a batch of t independent spanning trees $\sim \mathbb{P}_{\mathrm{ST}}$, such that if

$$t \gtrsim \frac{1}{\epsilon^2} \log\left(\frac{n}{\delta}\right),$$

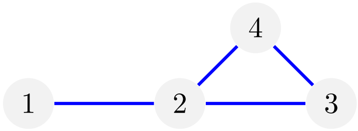
with $\epsilon \in (0,1)$ then, with probability at least $1-\delta$,

$$(1-\epsilon)L \preceq \widetilde{L}_t \preceq (1+\epsilon)L.$$

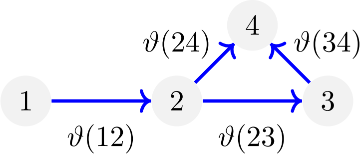
Here, $n = |\mathcal{V}|$ is the number of nodes.
See also Kyng & Song (2018).

Magnetic Laplacian and sparsification

Twisted edge-vertex incidence matrix



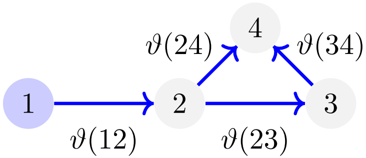
Twisted edge-vertex incidence matrix



Row uv is $(\boldsymbol{\delta}_u - e^{\mathrm{i}\,\vartheta(uv)}\boldsymbol{\delta}_v)^\top$.

$$B = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} & & & \\ & & & \\ & & & \\ & & & \end{array} \right] \end{array}.$$

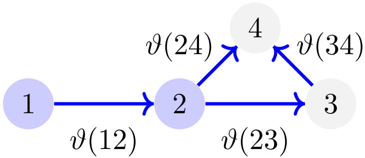
Twisted edge-vertex incidence matrix



Row uv is $(\boldsymbol{\delta}_u - e^{\mathrm{i}\,\vartheta(uv)}\boldsymbol{\delta}_v)^\top$.

$$B = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & & & \\ 0 & & & \\ 0 & & & \\ 0 & & & \end{bmatrix} \end{array}.$$

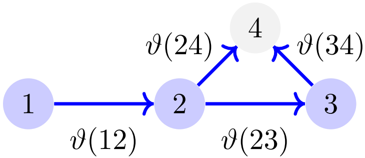
Twisted edge-vertex incidence matrix



Row uv is $(\boldsymbol{\delta}_u - e^{\mathrm{i}\,\vartheta(uv)}\boldsymbol{\delta}_v)^\top$.

$$B = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & -e^{\mathrm{i}\,\vartheta(12)} & & \\ 0 & 1 & & \\ 0 & 0 & & \\ 0 & 1 & & \end{bmatrix} \end{array}.$$

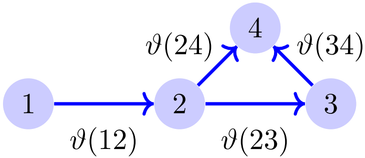
Twisted edge-vertex incidence matrix



Row uv is $(\boldsymbol{\delta}_u - e^{\mathrm{i}\,\vartheta(uv)}\boldsymbol{\delta}_v)^\top$.

$$B = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \left[\begin{array}{cccc} 1 & -e^{\mathrm{i}\,\vartheta(12)} & 0 & \\ 0 & 1 & -e^{\mathrm{i}\,\vartheta(23)} & \\ 0 & 0 & 1 & \\ 0 & 1 & 0 & \end{array}\right] \end{array}.$$

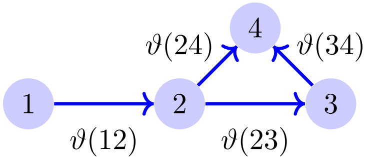
Twisted edge-vertex incidence matrix



Row uv is $(\boldsymbol{\delta}_u - e^{\mathrm{i}\,\vartheta(uv)}\boldsymbol{\delta}_v)^\top$.

$$B = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & -e^{\mathrm{i}\,\vartheta(12)} & 0 & 0 \\ 0 & 1 & -e^{\mathrm{i}\,\vartheta(23)} & 0 \\ 0 & 0 & 1 & -e^{\mathrm{i}\,\vartheta(34)} \\ 0 & 1 & 0 & -e^{\mathrm{i}\,\vartheta(24)} \end{bmatrix} \end{array}.$$

Twisted edge-vertex incidence matrix

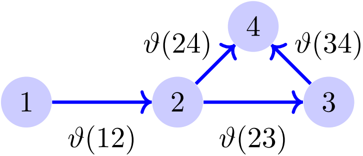


Row uv is $(\boldsymbol{\delta}_u - e^{\mathrm{i}\,\vartheta(uv)}\boldsymbol{\delta}_v)^\top$.

$$B = \begin{array}{c} \\ 12 \\ 23 \\ 34 \\ 24 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 1 & -\,e^{\mathrm{i}\,\vartheta(12)} & 0 & 0 \\ 0 & 1 & -\,e^{\mathrm{i}\,\vartheta(23)} & 0 \\ 0 & 0 & 1 & -\,e^{\mathrm{i}\,\vartheta(34)} \\ 0 & 1 & 0 & -\,e^{\mathrm{i}\,\vartheta(24)} \end{bmatrix} \end{array}.$$

Non-triviality of B depends on cycle *consistency*!

Cycle holonomy



Define $c = 234$ and $\theta(c) = \vartheta(23) + \vartheta(34) + \vartheta(42) \mod 2\pi$.

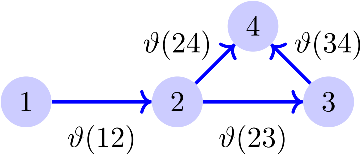
Holonomy

The holonomy of the connection along any oriented cycle c is

$$\prod_{e \in c} \phi_e \triangleq \exp(-\mathrm{i}\,\theta(c))$$

where $\phi_{uv} = e^{-\mathrm{i}\,\vartheta(uv)}$.

Cycle holonomy



Define $c = 234$ and $\theta(c) = \vartheta(23) + \vartheta(34) + \vartheta(42) \mod 2\pi$.

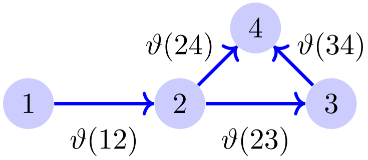
Holonomy

The holonomy of the connection along any oriented cycle c is

$$\prod_{e \in c} \phi_e \triangleq \exp(-\mathrm{i}\,\theta(c)) = \mathrm{hol}(c),$$

where $\phi_{uv} = e^{-\mathrm{i}\,\vartheta(uv)}$.

Cycle holonomy



Define $c = 234$ and $\theta(c) = \vartheta(23) + \vartheta(34) + \vartheta(42) \mod 2\pi$.

Holonomy

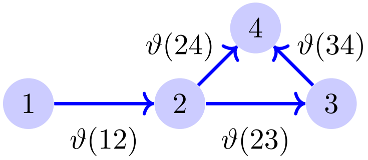
The holonomy of the connection along any oriented cycle c is

$$\prod_{e \in c} \phi_e \triangleq \exp(-\mathrm{i}\,\theta(c)) = \mathrm{hol}(c),$$

where $\phi_{uv} = e^{-\mathrm{i}\,\vartheta(uv)}$.

- If $\cos\theta(c) \geq 0$, we say that c is **weakly inconsistent**.

Cycle holonomy



Define $c = 234$ and $\theta(c) = \vartheta(23) + \vartheta(34) + \vartheta(42) \mod 2\pi$.

Holonomy

The holonomy of the connection along any oriented cycle c is

$$\prod_{e \in c} \phi_e \triangleq \exp(-\mathrm{i}\,\theta(c)) = \mathrm{hol}(c),$$

where $\phi_{uv} = e^{-\mathrm{i}\,\vartheta(uv)}$.

- If $\cos\theta(c) \geq 0$, we say that c is **weakly inconsistent**.
- We say that a U(1)-connection graph is weakly inconsistent if all its cycles are weakly inconsistent.

Magnetic Laplacian of a connected graph

Magnetic Laplacian

$$\Delta = B^*B = \begin{bmatrix} 1 & -\phi_{12}^* & 0 & 0 \\ -\phi_{12} & 3 & -\phi_{23}^* & -\phi_{24}^* \\ 0 & -\phi_{23} & 2 & -\phi_{34}^* \\ 0 & -\phi_{24} & -\phi_{34} & 2 \end{bmatrix}$$

with $\phi_{uv} = \exp(-\mathrm{i}\,\vartheta(uv))$.

Magnetic Laplacian of a connected graph

Magnetic Laplacian

$$\Delta = B^*B = \begin{bmatrix} 1 & -\phi_{12}^* & 0 & 0 \\ -\phi_{12} & 3 & -\phi_{23}^* & -\phi_{24}^* \\ 0 & -\phi_{23} & 2 & -\phi_{34}^* \\ 0 & -\phi_{24} & -\phi_{34} & 2 \end{bmatrix}$$

with $\phi_{uv} = \exp(-\mathrm{i}\,\vartheta(uv))$.

- $\boldsymbol{f}^*\Delta\boldsymbol{f} \propto \sum_{uv\in\mathcal{E}} |f(u) - \phi_{vu}f(v)|^2$

Magnetic Laplacian of a connected graph

Magnetic Laplacian

$$\Delta = B^*B = \begin{bmatrix} 1 & -\phi_{12}^* & 0 & 0 \\ -\phi_{12} & 3 & -\phi_{23}^* & -\phi_{24}^* \\ 0 & -\phi_{23} & 2 & -\phi_{34}^* \\ 0 & -\phi_{24} & -\phi_{34} & 2 \end{bmatrix}$$

with $\phi_{uv} = \exp(-\mathrm{i}\,\vartheta(uv))$.

- $\boldsymbol{f}^*\Delta\boldsymbol{f} \propto \sum_{uv\in\mathcal{E}} |f(u) - \phi_{vu}f(v)|^2$
- $\mathrm{null}(\Delta) = \{\mathbf{0}\}$ iff there exists at least one c s.t. $\cos\theta(c) \neq 1$.

Magnetic Laplacian of a connected graph

Magnetic Laplacian

$$\Delta = B^*B = \begin{bmatrix} 1 & -\phi_{12}^* & 0 & 0 \\ -\phi_{12} & 3 & -\phi_{23}^* & -\phi_{24}^* \\ 0 & -\phi_{23} & 2 & -\phi_{34}^* \\ 0 & -\phi_{24} & -\phi_{34} & 2 \end{bmatrix}$$

with $\phi_{uv} = \exp(-\mathrm{i}\,\vartheta(uv))$.

- $\boldsymbol{f}^*\Delta\boldsymbol{f} \propto \sum_{uv\in\mathcal{E}} |f(u) - \phi_{vu} f(v)|^2$
- $\mathrm{null}(\Delta) = \{\mathbf{0}\}$ iff there exists at least one c s.t. $\cos\theta(c) \neq 1$.
- In what follows, we assume $\exists\ c$ s.t. $\cos\theta(c) \neq 1$.

Magnetic Laplacian of a connected graph

Magnetic Laplacian

$$\Delta = B^* B = \begin{bmatrix} 1 & -\phi_{12}^* & 0 & 0 \\ -\phi_{12} & 3 & -\phi_{23}^* & -\phi_{24}^* \\ 0 & -\phi_{23} & 2 & -\phi_{34}^* \\ 0 & -\phi_{24} & -\phi_{34} & 2 \end{bmatrix}$$

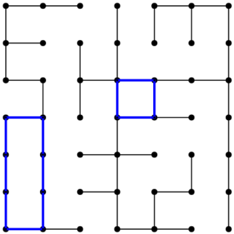
with $\phi_{uv} = \exp(-\mathrm{i}\,\vartheta(uv))$.

- $\boldsymbol{f}^* \Delta \boldsymbol{f} \propto \sum_{uv \in \mathcal{E}} |f(u) - \phi_{vu} f(v)|^2$
- $\mathrm{null}(\Delta) = \{\mathbf{0}\}$ iff there exists at least one c s.t. $\cos\theta(c) \neq 1$.
- In what follows, we assume $\exists\, c$ s.t. $\cos\theta(c) \neq 1$.
- $$\Delta = D - W_\phi$$

  with $D = \mathrm{Diag}(\deg)$ and $\deg(u) = \sharp$ neighbors of $u \in \mathcal{V}$.

Cycle-rooted spanning forest

Kenyon (2017)



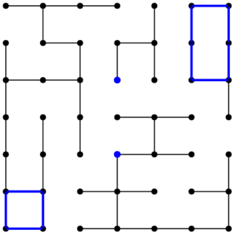
$\mathcal{S}$ is cycle-rooted spanning forest (CRSF) of $\mathcal{G}$, i.e., a spanning subgraph of $\mathcal{G}$ in which each connected component has **exactly one** cycle.

$$\mathbb{P}_{\mathrm{CRSF}}(\mathcal{S}) = \frac{1}{\det(\Delta)} \prod_{\substack{\text{non-oriented} \\ \text{cycle } c \subseteq \mathcal{S}}} 2\,(1 - \cos\theta(c))\,.$$

Multi-type spanning forest

Kenyon (2019)

Let $q > 0$.



$\mathcal{S}$ is a multi-type spanning forest (MTSF) of $\mathcal{G}$, i.e., a spanning subgraph of $\mathcal{G}$ in which each connected component has either **exactly one** cycle or **no** cycle.

$$\mathbb{P}_{\mathrm{MTSF}}(\mathcal{S}) = \frac{q^{\rho(\mathcal{S})}}{\det(\Delta + q\mathbb{I})} \prod_{\substack{\text{non-oriented} \\ \text{cycle } c \subseteq \mathcal{S}}} 2\Big(1 - \cos\theta(c)\Big),$$

where $\rho(\mathcal{S})$ is the number of components without cycle.

Sparsification guarantees

Fanuel & Bardenet, arxiv 2208.14797

Let $q \geq 0$ and let

$$d_{\text{eff}} = \operatorname{Tr}(\Delta(\Delta + q\mathbb{I}_n)^{-1}) \text{ and } \kappa = \|\Delta(\Delta + q\mathbb{I}_n)^{-1}\|_{\text{op}}.$$

Statistical guarantees

Theorem (Informal)

There exits a sparsifier $\widetilde{\Delta}_t$ built with a batch of t independent MTSFs $\sim \mathbb{P}_{\text{MTSF}}$, such that if

$$t \gtrsim \frac{\kappa}{\epsilon^2} \log\left(\frac{d_{\text{eff}}}{\kappa\delta}\right) = \epsilon^{-2} \cdot \textit{decreasing fct of } q,$$

with $\epsilon \in (0,1)$ then, with probability at least $1-\delta$,

$$(1-\epsilon)(\Delta + q\mathbb{I}) \preceq \widetilde{\Delta}_t + q\mathbb{I} \preceq (1+\epsilon)(\Delta + q\mathbb{I}).$$

Sparsifier with t i.i.d. MTSFs

The sparsifier is

$$\widetilde{\Delta}_t = \frac{1}{t}\sum_{\ell=1}^{t} \widetilde{\Delta}(\mathcal{S}_\ell)$$

with

$$\widetilde{\Delta}(\mathcal{S}) = \sum_{\text{edge } uv \in \mathcal{S}} \frac{1}{l(uv)} (\boldsymbol{\delta}_u - \phi_{uv}\boldsymbol{\delta}_v)(\boldsymbol{\delta}_u - \phi_{uv}\boldsymbol{\delta}_v)^*,$$

and where the leverage score of $e \in \mathcal{E}$ is

$$l(e) = [B(\Delta + q\mathbb{I}_n)^{-1}B^*]_{ee}.$$

Sampling edges with a loop-erased random walk

Connection-aware transition matrix and CyclePopping

Let x and y be neighboring nodes. Define

$$\Pi_{xy} = \frac{1}{\deg(x)} \cdot \exp(-\mathrm{i}\,\vartheta(xy)),$$

where $\deg(x)$ is $\sharp$ of neighbors of x. Note $\mathsf{\Pi} = \mathbb{I} - \mathsf{D}^{-1}\mathsf{\Delta}$.

Stricto sensu, Π is not a transition matrix.

- $1/\deg(x)$: transition probability from x to y
- $\vartheta(xy)$ is an angle used to define CyclePopping. Recall

$$\prod_{xy \in c} \exp(-\mathrm{i}\,\vartheta(xy)) \triangleq \exp(-\mathrm{i}\,\theta(c)).$$

Weak inconsistency: $\cos\theta(c) \geq 0$ for all cycle c.

CyclePopping considers $\cos\theta(c)$ as
the probability to pop (erase) c.

CRSF sampling $\sim \mathbb{P}_{\text{CRSF}}$ (Kassel and Kenyon, 2017)

Extension of Wilson's algorithm (1996)

CYCLEPOPPING

Fix an ordering of the nodes. Initialize $\mathcal{S} = \emptyset$.

1. Start from the first node in the ordering and not in $\mathcal{S}$.
2. Do a **nearest-neighbor random walk** until
   - either the walk intersects $\mathcal{S}$. Then, this branch is added to $\mathcal{S}$.
   - or the walk self-intersects, i.e., makes a **cycle** c.
     Then, draw $B \sim \text{Bern}(1 - \cos\theta(c))$.
     - If $B = 0$, the cycle c is **popped** (erased),
       and the walk continues from the knot (go to step 2.).
     - Else if $B = 1$, c is **accepted**,
       and the lasso is added to $\mathcal{S}$.

The sequence 1-2 is repeated until $\mathcal{S}$ covers the graph.
Finally, we forget edge orientations.

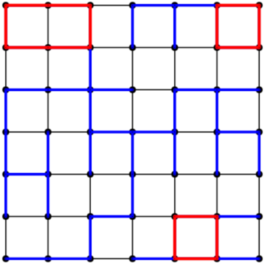
MTSF Sampling $\sim \mathbb{P}_{\mathrm{MTSF}}(\mathcal{S})$

Similar algorithm for sampling MTSFs.

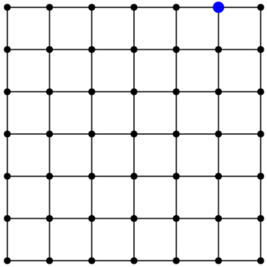
The only change is that the walker can, at node u,

- become a root with a probability $q/(\deg(u)+q)$,
- or do a step uniformly to a neighbor of u.

CyclePopping



CyclePopping



T: the number of steps to finish CyclePopping

Fanuel & Bardenet, arxiv 2404.14803

Law of T

Theorem

For a weakly inconsistent U(1)*-connection graph, we have*

$$\mathbb{E}[T] = \mathrm{Tr}\left(\mathsf{D}\Delta^{-1}\right)$$

with Δ *the magnetic Laplacian and* D *the degree matrix.*

T: the number of steps to finish CyclePopping

Fanuel & Bardenet, arxiv 2404.14803

Law of T

Theorem

For a weakly inconsistent U(1)*-connection graph, we have*

$$\mathbb{E}[T] = \mathrm{Tr}\left(\mathsf{D}\Delta^{-1}\right)$$

with Δ *the magnetic Laplacian and* D *the degree matrix. Furthermore,*

$$T \overset{(law)}{=} n + \sum_{[\gamma]\in\mathcal{X}} |\gamma| \text{ with } \mathcal{X} \sim \mathrm{Poisson}(m, \mathit{Loops}),$$

T: the number of steps to finish CyclePopping

Fanuel & Bardenet, arxiv 2404.14803

Law of T

Theorem

For a weakly inconsistent U(1)*-connection graph, we have*

$$\mathbb{E}[T] = \mathrm{Tr}\left(\mathsf{D}\mathsf{\Delta}^{-1}\right)$$

with $\mathsf{\Delta}$ *the magnetic Laplacian and* D *the degree matrix. Furthermore,*

$$T \overset{(law)}{=} n + \sum_{[\gamma]\in\mathcal{X}} |\gamma| \text{ with } \mathcal{X} \sim \mathrm{Poisson}(m, \mathit{Loops}),$$

where $m([\gamma]) = \frac{1}{\mathit{mult}(\gamma)} \prod_{xy\in[\gamma]} \frac{1}{\deg(x)} \prod_{c\in \mathit{cycles}(\gamma)} \cos\theta(c)$.

To better understand CyclePopping

A *based loop* γ is an oriented walk $\gamma = (x_0, \ldots, x_k)$ in the graph G, with $x_k = x_0$ for some integer $k \geq 2$.

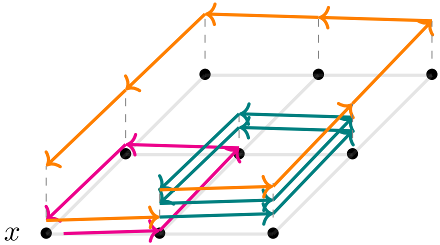


Figure: Based loop γ based at x.

Numerical simulations

Condition number after preconditioning

Magnetic Laplacian case ($q = 0$)

- We draw random connection graphs.
- We compute $\text{cond}(\widetilde{\Delta}^{-1}\Delta)$ where $\widetilde{\Delta}$ is obtained with several methods.

Baselines

- i.i.d. leverage score sampling.
- uniform spanning tree sampling.

Edge weights

- sketched leverage scores with Johnson-Lindenstrauss lemma.
- uniform heuristics

$$l(e) = |\mathcal{S}|/m.$$

Simulation settings: random connection graphs

- **Multiplicative Uniform Noise** (MUN).
  With probability p, and independently, there is an edge $e = uv$ for $1 \leq u < v \leq n$ with

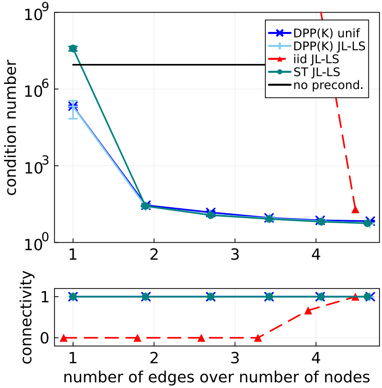
$$\vartheta(uv) = (h_u - h_v)(1 + \eta\epsilon_{uv})/(\pi(n-1))$$

where $\epsilon_{uv} \sim \mathcal{U}([0,1])$ are independent noise variables.

Uniform noise (MUN)

$n = 2000$, $p = 0.01$, $\eta = 10^{-3}$.

We display $\mathrm{cond}(\widetilde{\Delta}^{-1}\Delta)$.



Random MUN connection on top of a real graph

$n = 255,265$ nodes and $m = 1,941,926$ edges.

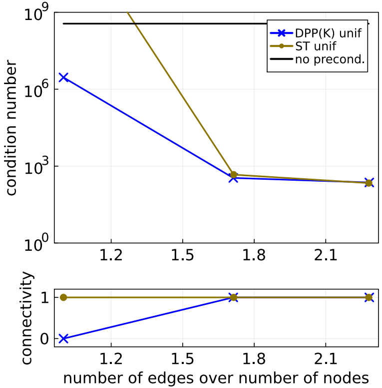


Figure: $\mathrm{cond}(\widetilde{\Delta}^{-1}\Delta)$ Stanford-MUN: $\eta = 10^{-2}$.

Research perspectives

- Go beyond the case of weakly inconsistent connection graphs with CyclePopping.
- Fast numerical implementation of CyclePopping.
- Generalization to diagonally dominant Hermitian matrices.
- Approximate leverage scores.
- Use more general connection graphs (e.g. $SO(3)$).

Thanks for your attention!

https://github.com/For-a-few-DPPs-more/MagneticLaplacianSparsifier.jl

We acknowledge support from ERC grant BLACKJACK (ERC-2019-STG-851866) and ANR AI chair BACCARAT (ANR-20-CHIA-0002). PI: R. Bardenet.

Importance sampling with capped cycle weights

Define the importance sampling distribution

$$p_{\mathrm{IS}}(\mathcal{C}) \propto q^{|\rho(\mathcal{C})|} \prod_{\text{cycles } \eta \in \mathcal{C}} 2\{1 \wedge (1 - \cos\theta(\eta))\},$$

and the corresponding importance weights

$$w(\mathcal{C}) \propto \prod_{\text{cycles } \eta \in \mathcal{C}} \left\{1 \vee \left(1 - \cos\theta(\eta)\right)\right\},$$

We define a sparsifier with importance weights:

$$\widetilde{\Delta}_t^{(\mathrm{IS})} = \frac{1}{\sum_{s=1}^{t} w(\mathcal{C}'_s)} \sum_{\ell=1}^{t} w(\mathcal{C}'_\ell)\widetilde{\Delta}(\mathcal{C}'_\ell), \text{ with } \mathcal{C}'_\ell \overset{\text{i.i.d.}}{\sim} p_{\mathrm{IS}} \text{ for } 1 \leq \ell \leq t.$$

Proposition

Let $p \in (0,1)$. *Let* $\mathcal{C}'_1, \mathcal{C}'_2, \ldots,$ *be i.i.d. random MTSFs with the capped distribution* p_{IS}, *and consider the sequence of matrices*

$$(\widetilde{\Delta}_t^{(\mathrm{IS})})_{t \geq 1}.$$

Finally, let $z > 0$ *be such that*

$$\Pr(\|\boldsymbol{u}\|_2 \leq z) = p \textit{ for } \boldsymbol{u} \sim \mathcal{N}(0, \mathbb{I}_{n^2}).$$

Then, as $t \to \infty$,

$$\Pr\left[-z(\Delta + q\mathbb{I}_n) \preceq \widetilde{\Delta}_t^{(\mathrm{IS})} - \Delta \preceq z(\Delta + q\mathbb{I}_n)\right] \to 1 - p\,.$$

References

- A. Kassel and R. Kenyon, Random curves on surfaces induced from the Laplacian determinant, Ann. Probab. 2017.
- A. Kassel, Learning about critical phenomena from scribbles and sandpiles, ESAIM: Proc., 2015.
- R. Kenyon. Spanning Forests and the Vector Bundle Laplacian. Ann. Probab., 2011.
- R. Kenyon. Determinantal Spanning Forests on Planar Graphs. Ann. Probab., 2019.
- D. A. Spielman and N. Srivastava. Graph Sparsification by Effective Resistances. SIAM Journal on Computing, 2011.
- R. Kyng and Z. Song. A Matrix Chernoff Bound for Strongly Rayleigh Distributions and Spectral Sparsifiers from a few Random Spanning Trees, FOCS, 2018.

References (continued)

- T. Kaufman, R. Kyng, and F. Solda. Scalar and Matrix Chernoff Bounds from ℓ_∞-Independence, SODA, 2022.
- M. Cucuringu. Sync-Rank: Robust Ranking, Constrained Ranking and Rank Aggregation via Eigenvector and SDP Synchronization. IEEE Transactions on Network Science and Engineering, 2016.
- A. Singer. Angular Synchronization by Eigenvectors and Semidefinite Programming. Applied and Computational Harmonic Analysis, 2011.