Matrix Martingales in Randomized Numerical Linear Algebra

Speaker **Rasmus Kyng** Harvard University

Sep 27, 2018

Concentration of Scalar Random Variables

Random $X = \sum_i X_i$, $\quad X_i \in \mathbb{R}$

1. X_i are independent
2. $\mathbb{E}X_i = 0$

Is $X \approx 0$ with high probability?

Concentration of Scalar Random Variables

Bernstein's Inequality

Random $X = \sum_i X_i, \qquad X_i \in \mathbb{R}$

1. X_i are independent
2. $\mathbb{E}X_i = 0$
3. $|X_i| \leq r$
4. $\sum_i \mathbb{E}X_i^2 \leq \sigma^2$

gives

E.g. if $\varepsilon = 0.5$, and $r, \sigma^2 = 0.1/\log(1/\tau)$

$$\mathbb{P}[|X| > \varepsilon] \leq 2\exp\left(-\frac{\varepsilon^2/2}{r\varepsilon + \sigma^2}\right)$$

Concentration of Scalar **Martingales**

Random $X = \sum_i X_i$, $\quad X_i \in \mathbb{R}$

1. ~~X_i are independent~~
2. ~~$\mathbb{E}X_i = 0$~~ $\quad \mathbb{E}[X_i | X_1, \ldots, X_{i-1}] = 0$

Concentration of Scalar **Martingales**

Random $X = \sum_i X_i$, $\quad X_i \in \mathbb{R}$

1. ~~X_i are independent~~
2. ~~$\mathbb{E}X_i = 0$~~ $\quad \mathbb{E}[X_i | \text{ previous steps}] = 0$

Concentration of Scalar **Martingales**

Freedman's Inequality

~~**Bernstein's Inequality**~~

Random $X = \sum_i X_i, \qquad X_i \in \mathbb{R}$

1. ~~X_i are independent~~
2. ~~$\mathbb{E}X_i = 0$~~ $\qquad \mathbb{E}[X_i | \text{ previous steps}] = 0$
3. $|X_i| \leq r$
4. ~~$\sum_i \mathbb{E}X_i^2 \leq \sigma^2$~~ $\qquad$ ~~$\sum_i \mathbb{E}[X_i^2 | \text{ previous steps}] \leq \sigma^2$~~

$$\mathbb{P}\left[\sum_i \mathbb{E}\left[X_i^2 | \text{ previous steps}\right] > \sigma^2\right] \leq \delta$$

gives

$$\mathbb{P}[|X| > \varepsilon] \leq 2\exp\left(-\frac{\varepsilon^2/2}{r\varepsilon + \sigma^2}\right) + \delta$$

Concentration of Scalar Martingales

Freedman's Inequality

Random $X = \sum_i X_i, \qquad X_i \in \mathbb{R}$

1. X_i are independent
2. $\mathbb{E}[X_i | \text{ previous steps}] = 0$
3. $|X_i| \leq r$
4. $\mathbb{P}[\sum_i \mathbb{E}\left[X_i^2 | \text{ previous steps}\right] > \sigma^2] \leq \delta$

gives

$$\mathbb{P}[|X| > \varepsilon] \leq 2\exp\left(-\frac{\varepsilon^2/2}{r\varepsilon + \sigma^2}\right) + \delta$$

Concentration of **Matrix** Random Variables

Matrix Bernstein's Inequality (Tropp 11)

Random $\boldsymbol{X} = \sum_i \boldsymbol{X}_i$ $\quad\quad$ $\boldsymbol{X}_i \in \mathbb{R}^{d\times d}$, symmetric

1. $\boldsymbol{X}_i$ are independent
2. $\mathbb{E}\boldsymbol{X}_i = \mathbf{0}$
3. $\|\boldsymbol{X}_i\| \leq r$
4. $\left\|\sum_i \mathbb{E}\boldsymbol{X}_i^2\right\| \leq \sigma^2$

gives

$$\mathbb{P}[\|\boldsymbol{X}\| > \epsilon] \leq d\, 2\exp\left(-\frac{\varepsilon^2/2}{r\varepsilon+\sigma^2}\right)$$

Concentration of **Matrix Martingales**

Matrix Freedman's Inequality (Tropp 11)

Random $\boldsymbol{X} = \sum_i \boldsymbol{X}_i$ $\quad\quad$ $\boldsymbol{X}_i \in \mathbb{R}^{d\times d}$, symmetric

1. ~~$\boldsymbol{X}_i$ are independent~~
2. ~~$\mathbb{E}\boldsymbol{X}_i = 0$~~ $\quad\quad$ $\mathbb{E}[\boldsymbol{X}_i |$ previous steps$] = 0$
3. $\|\boldsymbol{X}_i\| \le r$
4. ~~$\left\|\sum_i \mathbb{E}\boldsymbol{X}_i^2\right\| \le \sigma^2$~~ $\mathbb{P}[\left\|\sum_i \mathbb{E}\left[\boldsymbol{X}_i^2 |\text{ prev. steps}\right]\right\| > \sigma^2] \le \delta$

gives

E.g. if $\varepsilon = 0.5$, and $r, \sigma^2 = 0.1/\log(d/\tau)$

$$\mathbb{P}[\|\boldsymbol{X}\| > \varepsilon] \le d\, 2\exp\left(-\frac{\varepsilon^2/2}{r\varepsilon+\sigma^2}\right) + \delta$$

$$\le \tau + \delta$$

Concentration of Matrix Martingales

Matrix Freedman's Inequality (Tropp 11)

$$\mathbb{P}\left[\left\|\sum_i \mathbb{E}\left[\boldsymbol{X}_i^2 \,|\, \text{prev. steps}\right]\right\| > \sigma^2\right] \leq \delta$$

Predictable quadratic variation

$$= \sum_i \mathbb{E}\left[\boldsymbol{X}_i^2 \,|\, \text{prev. steps}\right]$$

Laplacian Matrices

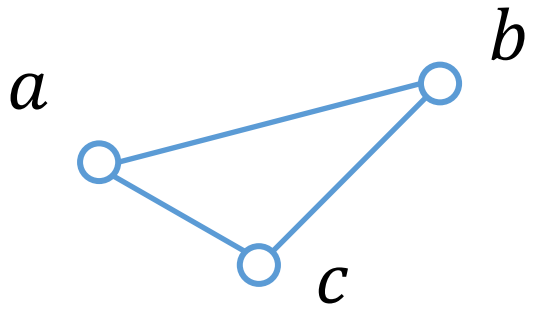
Graph $G = (V, E)$

Edge weights $w: E \to \mathbb{R}_+$

$n = |V|$

$m = |E|$

$n \times n$ matrix



$$L = \sum_{e \in E} L_e$$

Laplacian Matrices

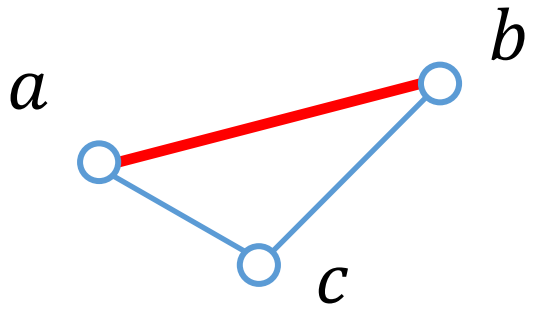
Graph $G = (V, E)$

Edge weights $w: E \to \mathbb{R}_+$

$n = |V|$

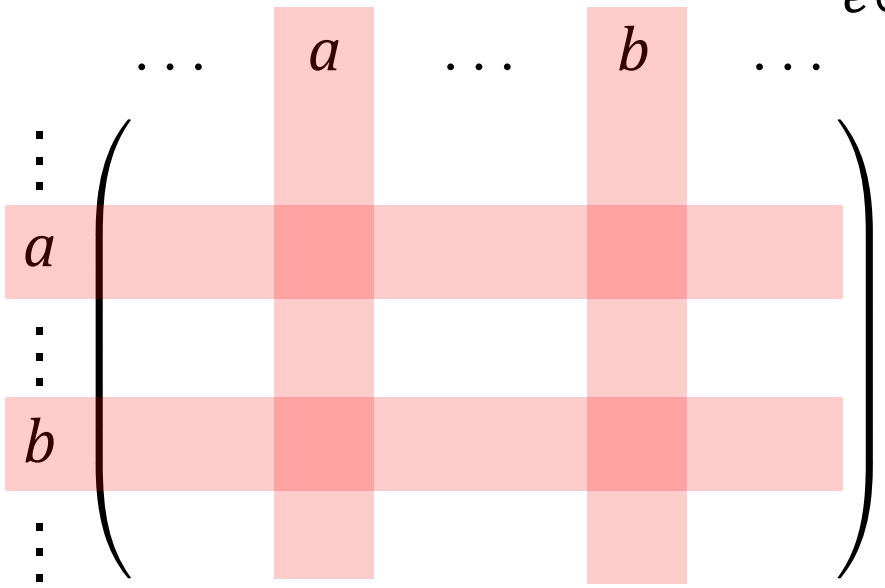
$m = |E|$

$n \times n$ matrix



$$L = \sum_{e \in E} L_e$$

$$L_{(a,b)} = w_{(a,b)}$$



Laplacian Matrices

Graph $G = (V, E)$

Edge weights $w: E \to \mathbb{R}_+$

$n = |V|$

$m = |E|$

$\boldsymbol{A}$: weighted adjacency matrix of the graph

$\boldsymbol{A}_{ij} = w_{ij}$

$\boldsymbol{D}$: diagonal matrix of weighted degrees

$\boldsymbol{D}_{ii} = \sum_j w_{ij}$

$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$

Laplacian Matrices

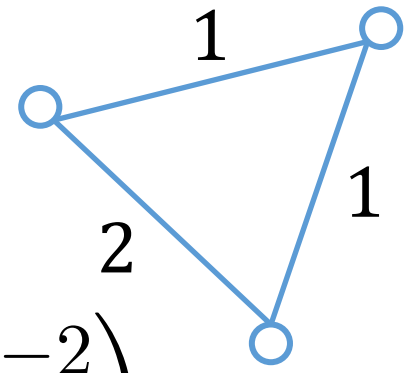
Symmetric matrix L

All off-diagonals are non-positive and

$$L_{ii} = \sum_{j \neq i} |L_{ij}|$$

Laplacian of a Graph

$$\begin{pmatrix} 1 & -1 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & -1 \\ 0 & -1 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & -2 \\ 0 & 0 & 0 \\ -2 & 0 & 2 \end{pmatrix}$$

$$\begin{pmatrix} 3 & -1 & -2 \\ -1 & 2 & -1 \\ -2 & -1 & 3 \end{pmatrix}$$

Approximation between graphs

PSD Order

$\boldsymbol{A} \preccurlyeq \boldsymbol{B}$ iff for all $\boldsymbol{x}$ $\boldsymbol{x}^\top \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{x}^\top \boldsymbol{B}\boldsymbol{x}$

Matrix Approximation

$\boldsymbol{A} \approx_\varepsilon \boldsymbol{B}$ iff $\boldsymbol{A} \preccurlyeq (1+\varepsilon)\boldsymbol{B}$ and $\boldsymbol{B} \preccurlyeq (1+\varepsilon)\boldsymbol{A}$

Graphs

$G \approx_\varepsilon H$ iff $\boldsymbol{L}_G \approx_\varepsilon \boldsymbol{L}_H$

Implies multiplicative approximation to cuts

Sparsification of Laplacians

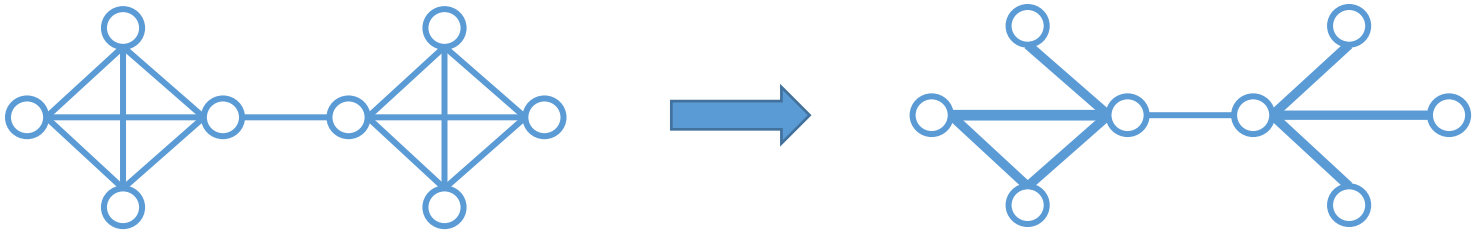
Every graph $G^{(0)}$ on n vertices
for any $\varepsilon \in (0,1)$,
has a sparsifier $G^{(1)} \approx_{\varepsilon} G^{(0)}$
with $O(n\,\varepsilon^{-2})$ edges

Many applications!

[BSS'09,LS'17]

Sampling edges independently
by leverage scores w.h.p. gives sparsifier
with $O(n\,\varepsilon^{-2}\log n)$ edges

[Spielman-Srivastava'08]



Graph Semi-Streaming Sparsification

m edges of a graph arriving as stream
$(\{a_1, b_1\}, w_1), (\{a_2, b_2\}, w_2), \ldots$

GOAL: maintain ε-sparsifier,
minimize working space (& time)

[Ahn-Guha '09]
$n\varepsilon^{-2}\log(n)\log\left(\frac{m}{n}\right)$ space
$n\varepsilon^{-2}\log(n)\log\left(\frac{m}{n}\right)$ edge **cut** ε-sparsifier
Used scalar martingales

Independent Edge Samples

$$\boldsymbol{L}_e^{(1)} = \begin{cases} \dfrac{1}{p_e}\boldsymbol{L}_e^{(0)} & \text{w. probability } p_e \\ \mathbf{0} & \text{o.w.} \end{cases}$$

Expectation $\quad \mathbb{E}\boldsymbol{L}_e^{(1)} = \boldsymbol{L}_e^{(0)}$

Zero-mean $\quad \boldsymbol{X}_e^{(1)} = \boldsymbol{L}_e^{(1)} - \boldsymbol{L}_e^{(0)}$

$$\boldsymbol{X}^{(1)} = \boldsymbol{L}^{(1)} - \boldsymbol{L}^{(0)} = \sum_e \boldsymbol{X}_e^{(1)}$$

Matrix Martingale?

Zero mean?

So what if we just sparsify again?

It's a martingale!

Every time we accumulate $20\, n\varepsilon^{-2}\log n$,
sparsify down to $10\, n\varepsilon^{-2}\log n$,
and continue

Independent Edge Samples

$$\boldsymbol{L}_e^{(1)} = \begin{cases} \frac{1}{p_e}\boldsymbol{L}_e^{(0)} & \text{w. probability } p_e \\ \mathbf{0} & \text{o.w.} \end{cases}$$

Expectation $\quad \mathbb{E}\boldsymbol{L}_e^{(1)} = \boldsymbol{L}_e^{(0)}$

Zero-mean $\quad \boldsymbol{X}_e^{(1)} = \boldsymbol{L}_e^{(1)} - \boldsymbol{L}_e^{(0)}$

$$\boldsymbol{X}^{(1)} = \sum_e \boldsymbol{X}_e^{(1)}$$

Repeated Edge Sampling

$$\boldsymbol{L}_e^{(i)} = \begin{cases} \dfrac{1}{p_e^{(i-1)}} \boldsymbol{L}_e^{(i-1)} & \text{w. prob. } p_e^{(i-1)} \\ \mathbf{0} & \text{o.w.} \end{cases}$$

Expectation $\quad \mathbb{E}\boldsymbol{L}_e^{(i)} = \boldsymbol{L}_e^{(i-1)}$

Zero-mean $\quad \boldsymbol{X}_e^{(i)} = \boldsymbol{L}_e^{(i)} - \boldsymbol{L}_e^{(i-1)}$

$$\boldsymbol{X}^{(i)} = \boldsymbol{L}^{(i)} - \boldsymbol{L}^{(i-1)} = \sum_{\boldsymbol{e}} \boldsymbol{X}_e^{(i)}$$

Repeated Edge Sampling

$$\boldsymbol{L}_e^{(i)} = \begin{cases} \dfrac{1}{p_e^{(i-1)}} \boldsymbol{L}_e^{(i-1)} & \text{w. prob. } p_e^{(i-1)} \\ \mathbf{0} & \text{o.w.} \end{cases}$$

Expectation $\quad \mathbb{E}\boldsymbol{L}_e^{(i)} = \boldsymbol{L}_e^{(i-1)}$

Zero-mean $\quad \boldsymbol{X}_e^{(i)} = \boldsymbol{L}_e^{(i)} - \boldsymbol{L}_e^{(i-1)}$

$$\boldsymbol{X}^{(i)} = \boldsymbol{L}^{(i)} - \boldsymbol{L}^{(i-1)} = \sum_{\boldsymbol{e}} \boldsymbol{X}_e^{(i)}$$

Repeated Edge Sampling

$$\boldsymbol{L}_e^{(i)} = \begin{cases} \dfrac{1}{p_e^{(i-1)}} \boldsymbol{L}_e^{(i-1)} & \text{w. prob. } p_e^{(i-1)} \\ \mathbf{0} & \text{o.w.} \end{cases}$$

Expectation $\quad \mathbb{E}\boldsymbol{L}_e^{(i)} = \boldsymbol{L}_e^{(i-1)}$

Zero-mean $\quad \boldsymbol{X}_e^{(i)} = \boldsymbol{L}_e^{(i)} - \boldsymbol{L}_e^{(i-1)}$

$$\boldsymbol{X}^{(i)} = \boldsymbol{L}^{(i)} - \boldsymbol{L}^{(i-1)} = \textstyle\sum_{\boldsymbol{e}} \boldsymbol{X}_e^{(i)}$$

Martingale $\quad \mathbb{E}\left[\boldsymbol{X}^{(i)} \middle| \text{prev. steps}\right] = \mathbf{0}$

Repeated Edge Sampling

It works – and it couldn't be simpler

[K.PPS'17]

$n\varepsilon^{-2}\log(n)$ space

$n\varepsilon^{-2}\log(n)$ edge ε-sparsifier

Shave a log off many algorithms

that do repeated matrix sampling…

Approximation?

$\boldsymbol{L}_H \approx_{0.5} \boldsymbol{L}_G$

$\boldsymbol{L}_H \preccurlyeq 1.5\boldsymbol{L}_G$ and $\boldsymbol{L}_G \preccurlyeq 1.5\boldsymbol{L}_H$

~~$\|\boldsymbol{L}_H - \boldsymbol{L}_G\| \leq 0.5$?~~

$\left\|\boldsymbol{L}_G^{-1/2}(\boldsymbol{L}_H - \boldsymbol{L}_G)\boldsymbol{L}_G^{-1/2}\right\| \leq 0.1$

Isotropic position

$$\overline{\boldsymbol{Z}} \stackrel{\text{def}}{=} \boldsymbol{L}_G^{-1/2}\, \boldsymbol{Z}\, \boldsymbol{L}_G^{-1/2}$$

Goal is now

$$\|\overline{\boldsymbol{L}}_H - \overline{\boldsymbol{L}}_G\| \leq 0.1$$

$$\|\overline{\boldsymbol{L}}_H - \boldsymbol{I}\| \leq 0.1$$

Concentration of **Matrix Martingales**

Matrix Freedman's Inequality

Random $\boldsymbol{X} = \sum_i \sum_e \boldsymbol{X}_e^{(i)}$ $\quad\quad$ $\boldsymbol{X}_e^{(i)} \in \mathbb{R}^{d\times d}$, symmetric

1. $\mathbb{E}\left[\boldsymbol{X}_e^{(i)} \mid \text{previous steps}\right] = 0$
2. $\left\|\boldsymbol{X}_e^{(i)}\right\| \leq r$
3. $\mathbb{P}\left[\left\|\sum_i \sum_e \mathbb{E}\left[\left(\boldsymbol{X}_e^{(i)}\right)^2 \mid \text{prev. steps}\right]\right\| > \sigma^2\right] \leq \delta$

gives

$$\mathbb{P}[\|\boldsymbol{X}\| > \varepsilon] \leq d\, 2\exp\left(-\frac{\varepsilon^2/2}{r\varepsilon+\sigma^2}\right) + \delta$$

Concentration of **Matrix Martingales**

Matrix Freedman's Inequality

Random $\overline{\boldsymbol{X}} = \sum_i \sum_e \overline{\boldsymbol{X}}_e^{(i)}$ $\overline{\boldsymbol{X}}_e^{(i)} \in \mathbb{R}^{d\times d}$, symmetric

1. $\mathbb{E}\left[\overline{\boldsymbol{X}}_e^{(i)} \mid \text{previous steps}\right] = 0$
2. $\left\|\overline{\boldsymbol{X}}_e^{(i)}\right\| \leq r$
3. $\mathbb{P}\left[\left\|\sum_i \sum_e \mathbb{E}\left[\left(\overline{\boldsymbol{X}}_e^{(i)}\right)^2 \mid \text{prev. steps}\right]\right\| > \sigma^2\right] \leq \delta$

gives

$$\mathbb{P}[\|\overline{\boldsymbol{X}}\| > \varepsilon] \leq d\, 2\exp\left(-\frac{\varepsilon^2/2}{r\varepsilon+\sigma^2}\right) + \delta$$

Norm Control

Can there be many edges with large norm?

$$\bar{\boldsymbol{L}}_e^{(1)} = \begin{cases} \frac{1}{p_e} \bar{\boldsymbol{L}}_e^{(0)} & \text{w. probability } p_e \\ \mathbf{0} & \text{o.w.} \end{cases}$$

$$\sum_e \left\| \bar{\boldsymbol{L}}_e^{(0)} \right\| = \sum_e \mathrm{Tr}\left(\bar{\boldsymbol{L}}_e^{(0)} \right) = \mathrm{Tr}\left(\sum_e \bar{\boldsymbol{L}}_e^{(0)} \right) = \mathrm{Tr}\left(\bar{\boldsymbol{L}}^{(0)} \right)$$
$$= \mathrm{Tr}(\boldsymbol{I}) = n - 1$$

Can afford $r = \frac{1}{\log n}$, $\quad p_e = \frac{\left\| \bar{\boldsymbol{L}}_e^{(0)} \right\|}{\log n}$, $\quad \sum_e p_e = n \log n$

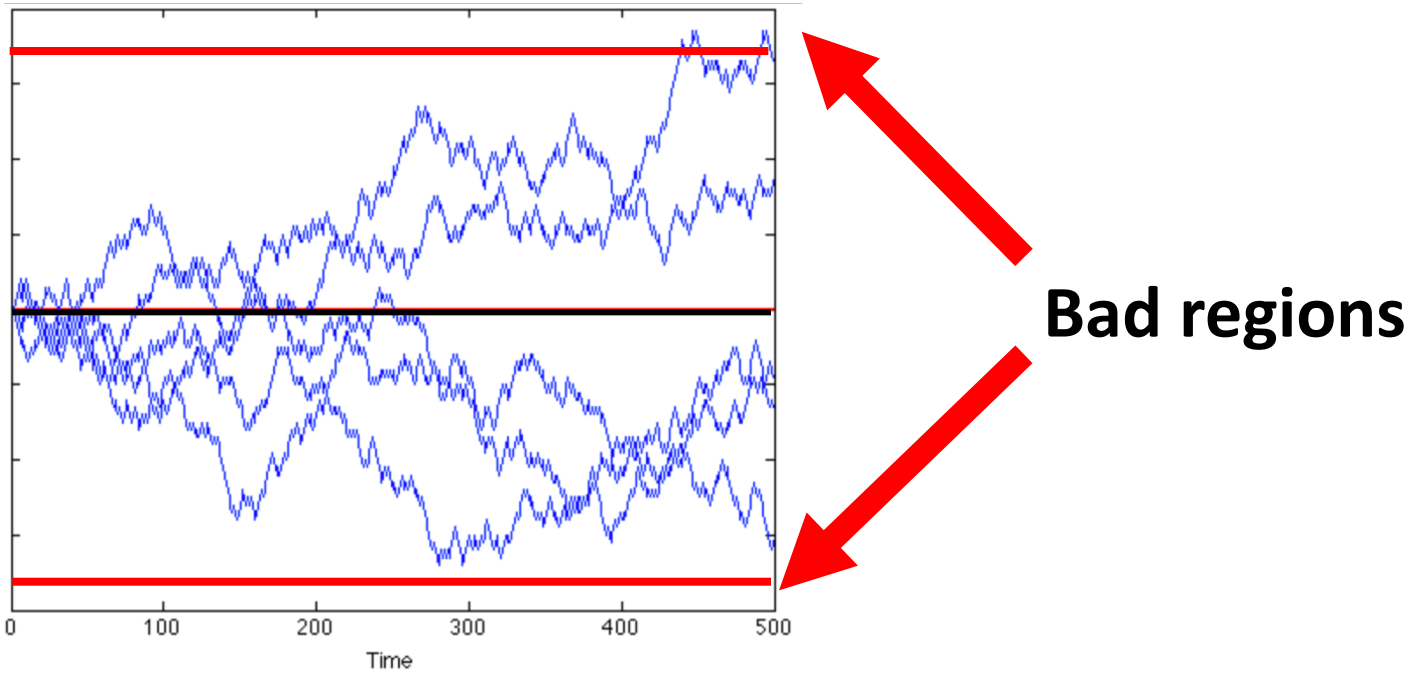
Norm Control

$$\left\| \overline{\boldsymbol{X}}_e^{(i)} \right\| \leq r$$

If we lose track of the original graph,

we can't estimate the norm

Stopping time argument



Variance Control

$$\mathbb{E}\left(\bar{\boldsymbol{X}}_e^{(1)}\right)^2 \preccurlyeq r\bar{\boldsymbol{L}}_e^{(0)}$$

$$\sum_e \mathbb{E}\left(\bar{\boldsymbol{X}}_e^{(1)}\right)^2 \preccurlyeq \sum_e r\bar{\boldsymbol{L}}_e^{(0)} = r\boldsymbol{I}$$

But then edges get resampled.

Geometric sequence for variance, roughly.

Resparsification

Every time we accumulate $20\, n\varepsilon^{-2}\log n$,
sparsify down to $10\, n\varepsilon^{-2}\log n$,
and continue

[K.PPS'17]
$n\varepsilon^{-2}\log(n)$ space
$n\varepsilon^{-2}\log(n)$ edge ε-sparsifier

Shave a log off many algorithms
that do repeated matrix sampling…

Laplacian Linear Equations

[ST04]: solving Laplacian linear equations in $\tilde{O}(m)$ time

⋮

[KS16]: simple algorithm

Solving a Laplacian Linear Equation

$$\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}$$

Gaussian Elimination

Find $\mathfrak{U}$, upper triangular matrix, s.t.

$$\mathfrak{U}^{\mathsf{T}}\mathfrak{U} = \boldsymbol{L}$$

Then

$$\boldsymbol{x} = \mathfrak{U}^{-1}\mathfrak{U}^{-\mathsf{T}}\boldsymbol{b}$$

Easy to apply $\mathfrak{U}^{-1}$ and $\mathfrak{U}^{-\mathsf{T}}$

Solving a Laplacian Linear Equation

$\boldsymbol{Lx} = \boldsymbol{b}$

Approximate Gaussian Elimination

Find $\mathfrak{U}$, upper triangular matrix, s.t.

$$\mathfrak{U}^{\top}\mathfrak{U} \approx_{0.5} \boldsymbol{L}$$

$\mathfrak{U}$ is sparse.

$O\left(\log\frac{1}{\varepsilon}\right)$ iterations to get

ε-approximate solution $\tilde{\boldsymbol{x}}$.

Approximate Gaussian Elimination

Theorem [**K**S]
When $\boldsymbol{L}$ is an **Laplacian** matrix with m non-zeros,
we can find in $O(m \log^3 n)$ time an upper triangular matrix $\mathfrak{U}$ with $O(m \log^3 n)$ non-zeros,
s.t. w.h.p.

$$\mathfrak{U}^\top \mathfrak{U} \approx_{0.5} \boldsymbol{L}$$

Additive View of Gaussian Elimination

Find $\mathfrak{U}$, upper triangular matrix, s.t $\mathfrak{U}^{\mathsf{T}}\mathfrak{U} = \boldsymbol{M}$

$$\boldsymbol{M} = \begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix}$$

Additive View of Gaussian Elimination

$$\begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix}$$

Find the rank-1 matrix that agrees with $\boldsymbol{M}$ on the first row and column.

$$\begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 1 & 2 & 1 \\ -8 & 2 & 4 & 2 \\ -4 & 1 & 2 & 1 \end{pmatrix} = \begin{pmatrix} 4 \\ -1 \\ -2 \\ -1 \end{pmatrix} \begin{pmatrix} 4 \\ -1 \\ -2 \\ -1 \end{pmatrix}^{\top}$$

Additive View of Gaussian Elimination

$$\begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix} -$$

Subtract the rank 1 matrix.

We have **eliminated the first variable.**

$$\begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 1 & 2 & 1 \\ -8 & 2 & 4 & 2 \\ -4 & 1 & 2 & 1 \end{pmatrix}$$

Additive View of Gaussian Elimination

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 10 & -2 \\ 0 & -2 & -2 & 6 \end{pmatrix}$$

The remaining matrix is PSD.

Additive View of Gaussian Elimination

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 10 & -2 \\ 0 & -2 & -2 & 6 \end{pmatrix}$$

Find rank-1 matrix that agrees with
our matrix on the **next** row and column.

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 1 & 1 \\ 0 & -2 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 2 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ -1 \\ -1 \end{pmatrix}^\top$$

Additive View of Gaussian Elimination

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 10 & -2 \\ 0 & -2 & -2 & 6 \end{pmatrix} - \qquad = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 9 & -3 \\ 0 & 0 & -3 & 5 \end{pmatrix}$$

Subtract the rank 1 matrix.

We have **eliminated the second variable.**

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 1 & 1 \\ 0 & -2 & 1 & 1 \end{pmatrix}$$

Additive View of Gaussian Elimination

Repeat until all parts written as rank 1 terms.

$$\boldsymbol{M} = \begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix}$$

$$= \begin{pmatrix} 4 \\ -1 \\ -2 \\ -1 \end{pmatrix} \begin{pmatrix} 4 \\ -1 \\ -2 \\ -1 \end{pmatrix}^{\top} + \begin{pmatrix} 0 \\ 2 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 2 \\ -1 \\ -1 \end{pmatrix}^{\top} + \begin{pmatrix} 0 \\ 0 \\ 3 \\ -1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 3 \\ -1 \end{pmatrix}^{\top} + \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \end{pmatrix}^{\top}$$

Additive View of Gaussian Elimination

Repeat until all parts written as rank 1 terms.

$$\boldsymbol{M} = \begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix}$$

$$= \begin{pmatrix} 4 & 0 & 0 & 0 \\ -1 & 2 & 0 & 0 \\ -2 & -1 & 3 & 0 \\ -1 & -1 & -1 & 2 \end{pmatrix} \begin{pmatrix} 4 & -1 & -2 & -1 \\ 0 & 2 & -1 & -1 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & 2 \end{pmatrix}$$

Additive View of Gaussian Elimination

Repeat until all parts written as rank 1 terms.

$$\boldsymbol{M} = \begin{pmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 14 & 0 \\ -4 & -1 & 0 & 7 \end{pmatrix}$$

$$= \begin{pmatrix} 4 & -1 & -2 & -1 \\ 0 & 2 & -1 & -1 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & 2 \end{pmatrix}^{\top} \begin{pmatrix} 4 & -1 & -2 & -1 \\ 0 & 2 & -1 & -1 \\ 0 & 0 & 3 & -1 \\ 0 & 0 & 0 & 2 \end{pmatrix} = \mathfrak{U}^{\top} \mathfrak{U}$$

Additive View of Gaussian Elimination

What is special about Gaussian Elimination on Laplacians?

The **remaining matrix** is always Laplacian.

$$\boldsymbol{L} = \begin{pmatrix} 16 & -8 & -4 & -4 \\ -8 & 8 & 0 & 0 \\ -4 & 0 & 4 & 0 \\ -4 & 0 & 0 & 4 \end{pmatrix}$$

Additive View of Gaussian Elimination

What is special about Gaussian Elimination on Laplacians?

The **remaining matrix** is always Laplacian.

$$\boldsymbol{L} = \begin{pmatrix} 16 & -8 & -4 & -4 \\ -8 & 4 & 2 & 2 \\ -4 & 2 & 1 & 1 \\ -4 & 2 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 3 & -1 \\ 0 & -2 & -1 & 3 \end{pmatrix}$$

Additive View of Gaussian Elimination

What is special about Gaussian Elimination on Laplacians?

The **remaining matrix** is always Laplacian.

$$\boldsymbol{L} = \begin{pmatrix} 4 \\ -2 \\ -1 \\ -1 \end{pmatrix} \begin{pmatrix} 4 \\ -2 \\ -1 \\ -1 \end{pmatrix}^{\top} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 3 & -1 \\ 0 & -2 & -1 & 3 \end{pmatrix}$$

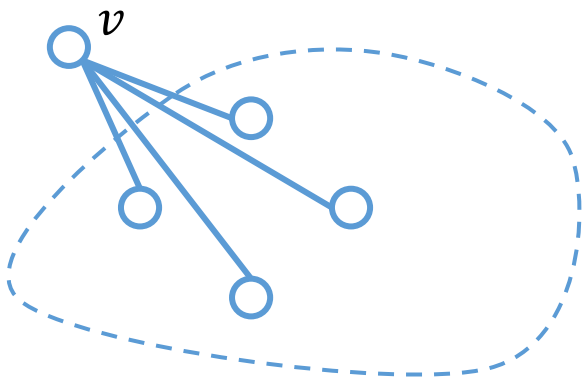
A new Laplacian!

Why is Gaussian Elimination Slow?

Solving $\boldsymbol{Lx} = \boldsymbol{b}$ by Gaussian Elimination can take $\Omega(n^3)$ time.

The main issue is **fill**

$\boldsymbol{L} =$

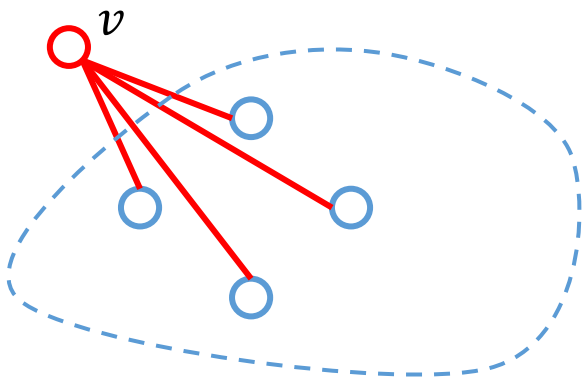


Why is Gaussian Elimination Slow?

Solving $\boldsymbol{Lx} = \boldsymbol{b}$ by Gaussian Elimination can take $\Omega(n^3)$ time.

The main issue is **fill**

$\boldsymbol{L} =$

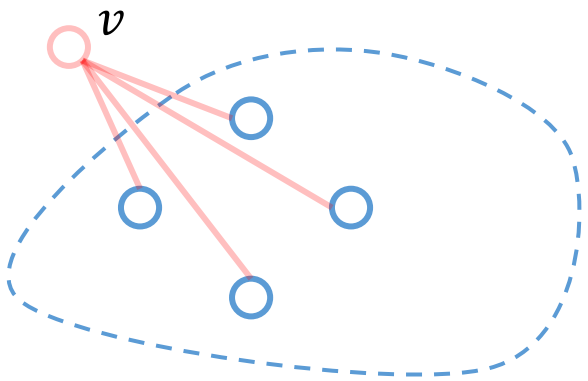


Why is Gaussian Elimination Slow?

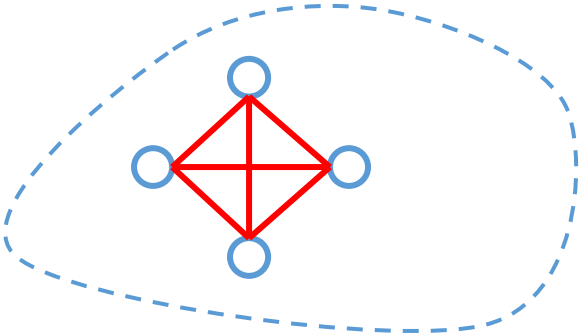
Solving $\boldsymbol{Lx} = \boldsymbol{b}$ by Gaussian Elimination can take $\Omega(n^3)$ time.

The main issue is **fill**

$\boldsymbol{L} =$



New Laplacian



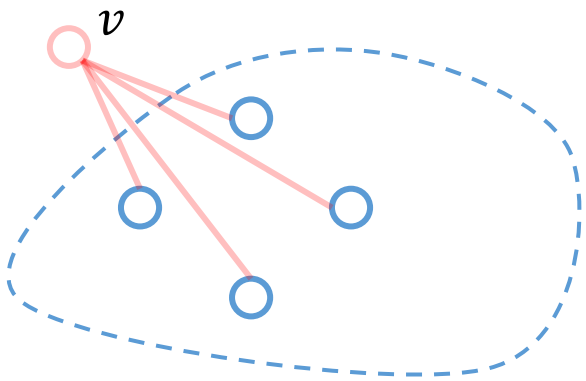
Elimination creates a clique on the neighbors of v

Why is Gaussian Elimination Slow?

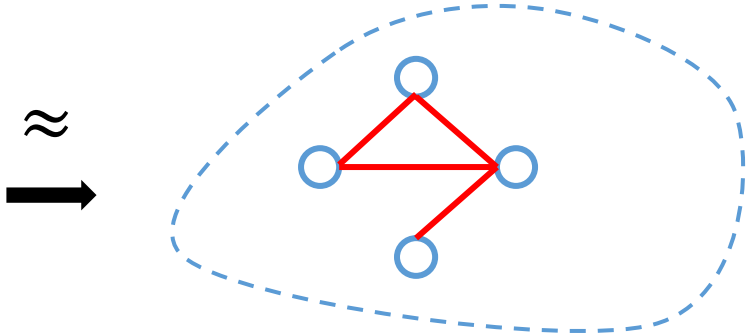
Solving $\boldsymbol{Lx} = \boldsymbol{b}$ by Gaussian Elimination
can take $\Omega(n^3)$ time.

The main issue is **fill**

$\boldsymbol{L} =$



New Laplacian



Laplacian cliques can be sparsified!

Gaussian Elimination

1. Pick a vertex v to eliminate
2. Add the clique created by eliminating v
3. Repeat until done

Approximate Gaussian Elimination

1. Pick a vertex v to eliminate
2. Add the clique created by eliminating v
3. Repeat until done

Approximate Gaussian Elimination

1. Pick a random vertex v to eliminate
2. Add the clique created by eliminating v
3. Repeat until done

Approximate Gaussian Elimination

1. Pick a random vertex v to eliminate
2. Sample the clique created by eliminating v
3. Repeat until done

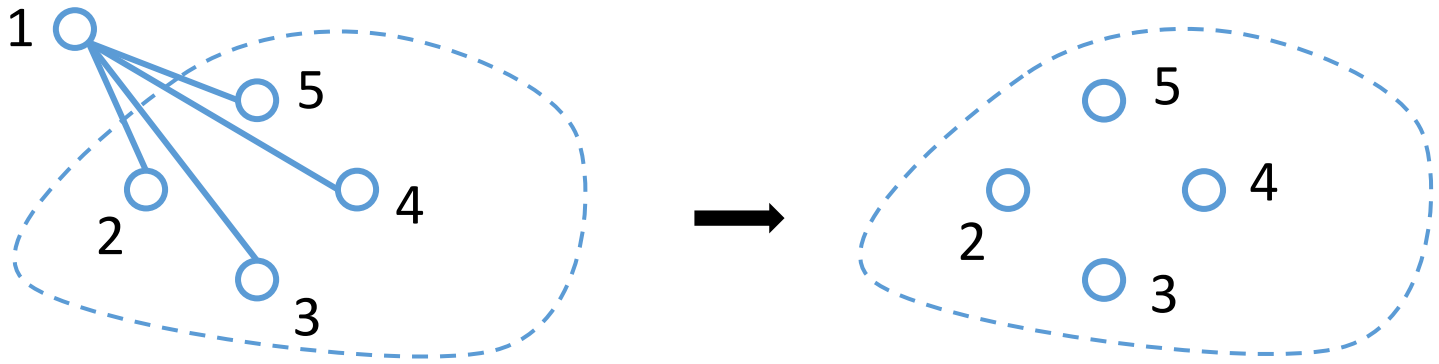
Resembles **randomized** Incomplete Cholesky

How to Sample a Clique

For each edge $(1, v)$

pick an edge $(1, u)$ with probability $\sim w_u$

insert edge (v, u) with weight $\frac{w_u w_v}{w_u + w_v}$

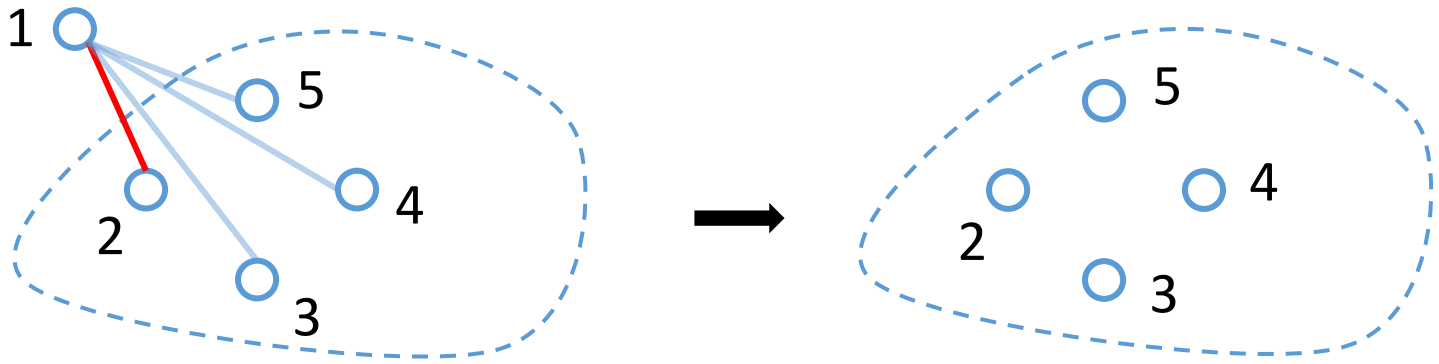


How to Sample a Clique

For each edge $(1, v)$

pick an edge $(1, u)$ with probability $\sim w_u$

insert edge (v, u) with weight $\frac{w_u w_v}{w_u + w_v}$

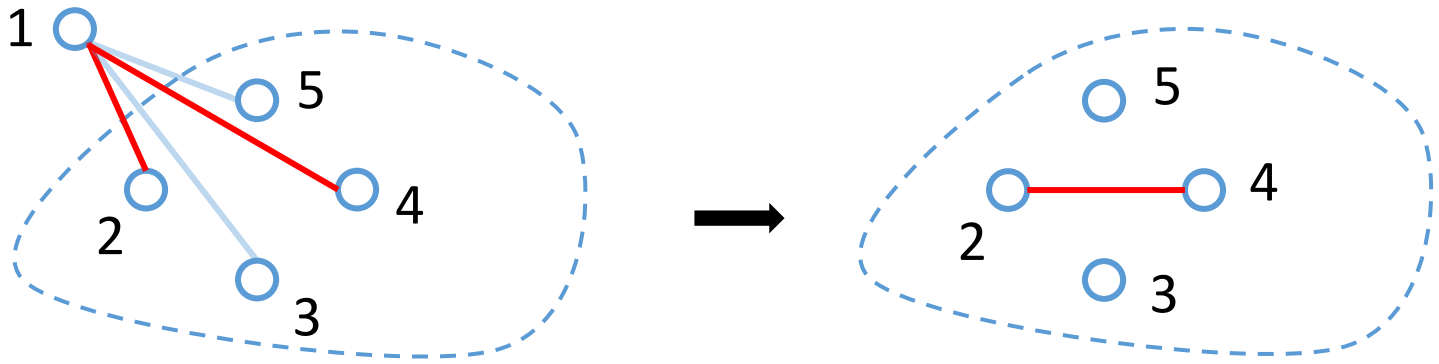


How to Sample a Clique

For each edge $(1, v)$

pick an edge $(1, u)$ with probability $\sim w_u$

insert edge (v, u) with weight $\frac{w_u w_v}{w_u + w_v}$

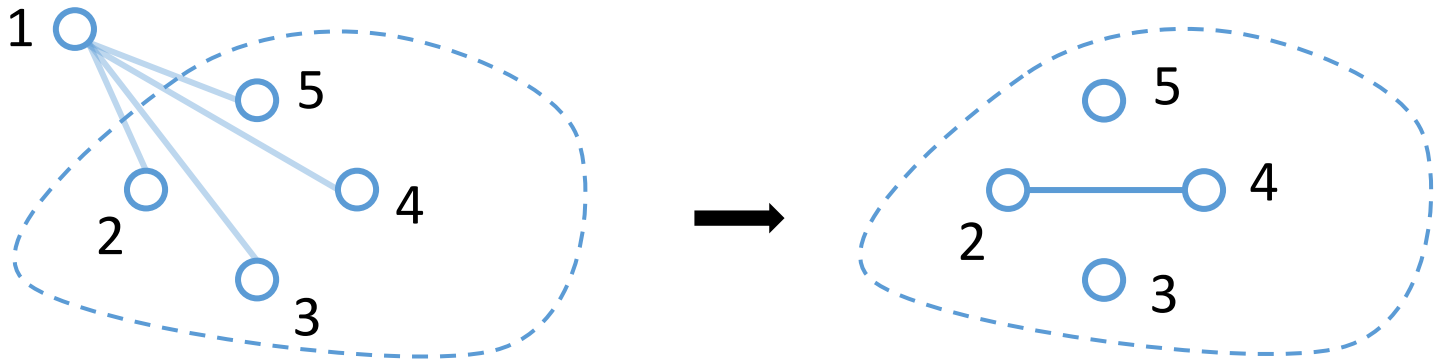


How to Sample a Clique

For each edge $(1, v)$

pick an edge $(1, u)$ with probability $\sim w_u$

insert edge (v, u) with weight $\frac{w_u w_v}{w_u + w_v}$

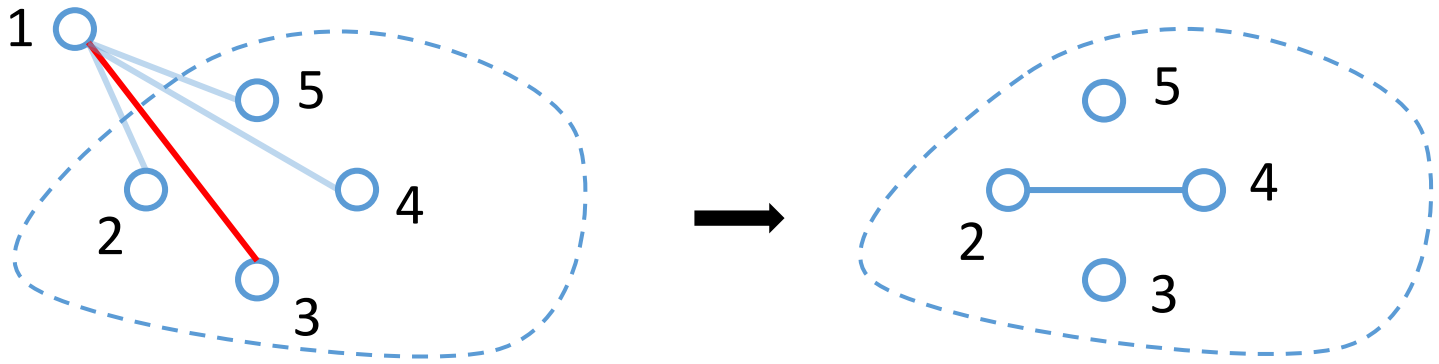


How to Sample a Clique

For each edge $(1, v)$

pick an edge $(1, u)$ with probability $\sim w_u$

insert edge (v, u) with weight $\frac{w_u w_v}{w_u + w_v}$

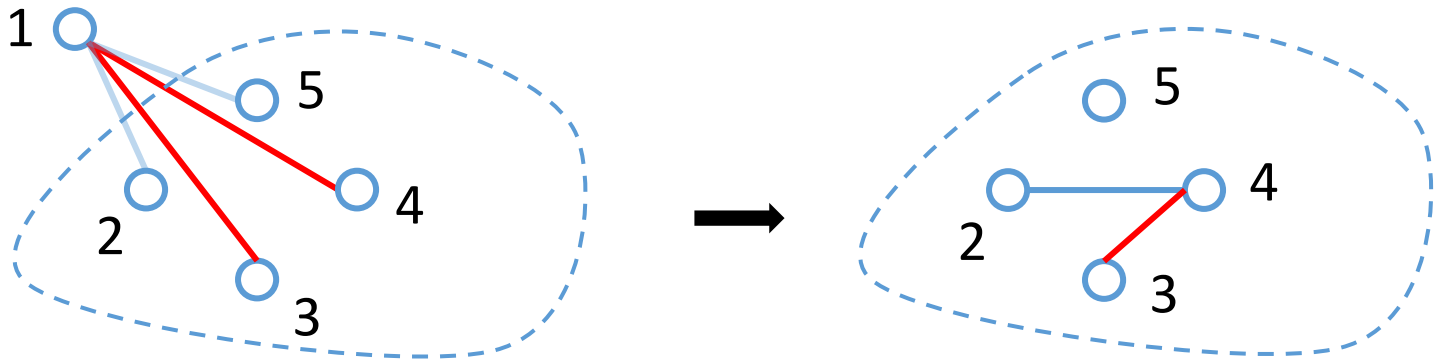


How to Sample a Clique

For each edge $(1, v)$

pick an edge $(1, u)$ with probability $\sim w_u$

insert edge (v, u) with weight $\frac{w_u w_v}{w_u + w_v}$

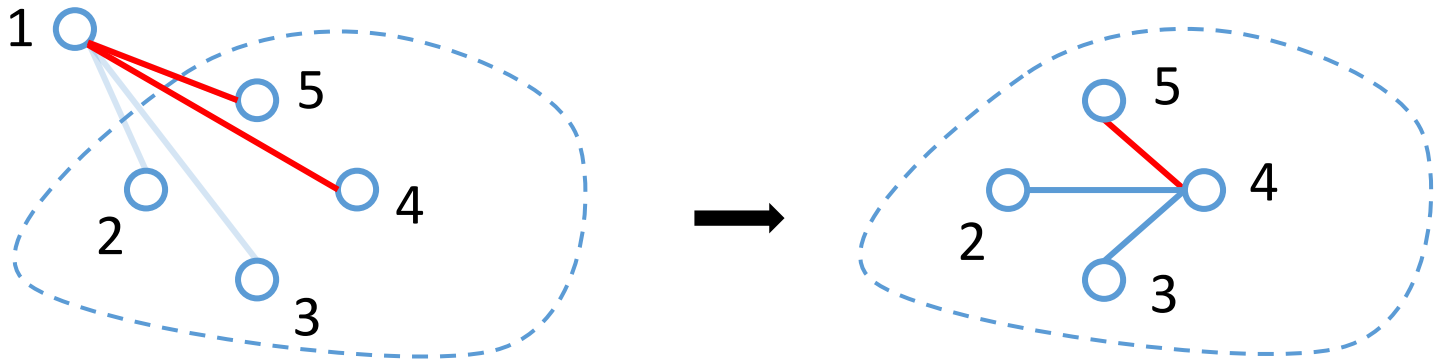


How to Sample a Clique

For each edge $(1, v)$

pick an edge $(1, u)$ with probability $\sim w_u$

insert edge (v, u) with weight $\frac{w_u w_v}{w_u + w_v}$

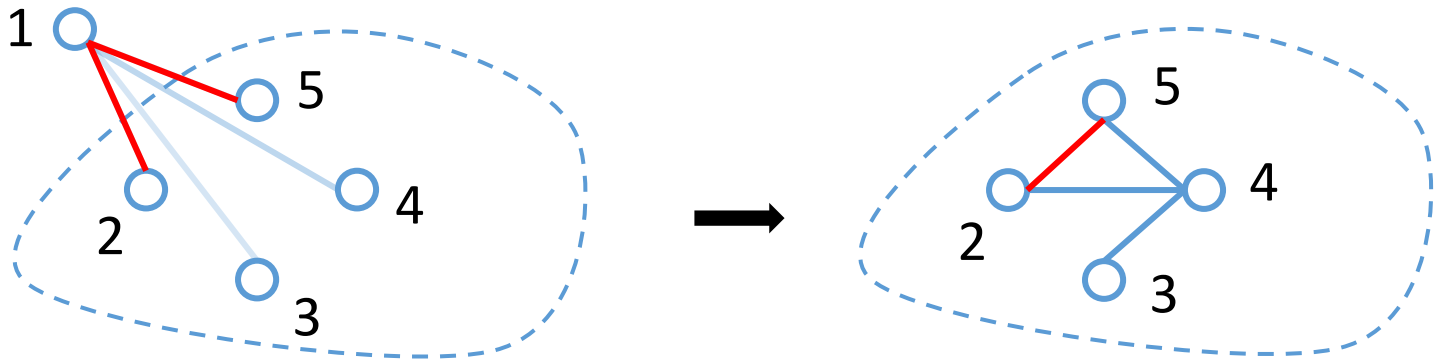


How to Sample a Clique

For each edge $(1, v)$

pick an edge $(1, u)$ with probability $\sim w_u$

insert edge (v, u) with weight $\frac{w_u w_v}{w_u + w_v}$

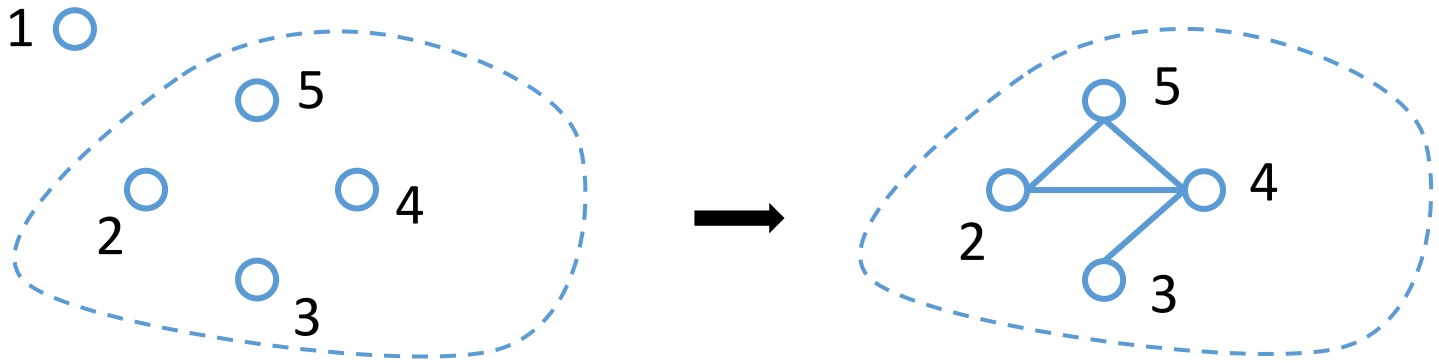


How to Sample a Clique

For each edge $(1, v)$

pick an edge $(1, u)$ with probability $\sim w_u$

insert edge (v, u) with weight $\frac{w_u w_v}{w_u + w_v}$



Approximating Matrices by Sampling

Goal

$$\mathfrak{U}^\top \mathfrak{U} \approx \boldsymbol{L}$$

Approach

1. $\mathbb{E}\, \mathfrak{U}^\top \mathfrak{U} = \boldsymbol{L}$
2. Show $\mathfrak{U}^\top \mathfrak{U}$ concentrated around expectation

Gives $\mathfrak{U}^\top \mathfrak{U} \approx \boldsymbol{L}$ w. high probability

Approximating Matrices in Expectation

Consider eliminating the first variable

$$\left(\quad \boldsymbol{L}^{(0)} \quad \right) = \left(\boldsymbol{c}_1 \right)\left(\boldsymbol{c}_1 \right)^{\mathsf{T}} + \left(\quad \blacksquare \quad \blacksquare \quad \blacksquare \quad \right)$$

Original Laplacian

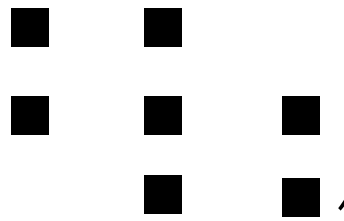
Rank 1 term

Remaining graph + clique

Approximating Matrices in Expectation

Consider eliminating the first variable

$$= \left(\boldsymbol{c}_1\right)\left(\boldsymbol{c}_1\right)^{\mathsf{T}} + \left(\quad \right)$$



Rank 1 term

Remaining graph + sparsified clique

Approximating Matrices in Expectation

Consider eliminating the first variable

$$\left(\quad \boldsymbol{L}^{(1)} \quad \right) = \left(\boldsymbol{c}_1 \right)\left(\boldsymbol{c}_1 \right)^{\mathsf{T}} + \left(\quad \blacksquare \quad \blacksquare \quad \blacksquare \quad \blacksquare \quad \blacksquare \quad \blacksquare \quad \right)$$

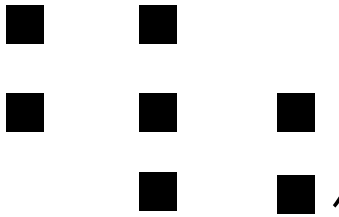
Rank 1 term

Remaining graph + sparsified clique

Approximating Matrices in Expectation

Consider eliminating the first variable

$$\mathbb{E}\left(\boldsymbol{L}^{(1)} \right) = \left(\boldsymbol{c}_1 \right)\left(\boldsymbol{c}_1 \right)^{\mathsf{T}} + \mathbb{E}\left(\right.$$



$\left.\right)$

Rank 1 term

Remaining graph + sparsified clique

Approximating Matrices in Expectation

Consider eliminating the first variable

$$\mathbb{E}\begin{pmatrix} & \boldsymbol{L}^{(1)} & \end{pmatrix} = \begin{pmatrix} \boldsymbol{c}_1 \end{pmatrix}\begin{pmatrix} \boldsymbol{c}_1 \end{pmatrix}^{\mathsf{T}} + \mathbb{E}\begin{pmatrix} \blacksquare & \blacksquare & \\ \blacksquare & \blacksquare & \blacksquare \\ & \blacksquare & \blacksquare \end{pmatrix}$$

Rank 1 term

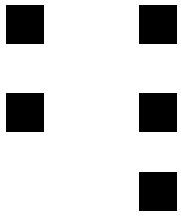
Remaining graph + sparsified clique

$$\text{Suppose} \quad = \begin{pmatrix} \boldsymbol{c}_1 \end{pmatrix}\begin{pmatrix} \boldsymbol{c}_1 \end{pmatrix}^{\mathsf{T}} + \begin{pmatrix} \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \\ \blacksquare & \blacksquare & \blacksquare \end{pmatrix}$$

Approximating Matrices in Expectation

Consider eliminating the first variable

$$\mathbb{E}\left(\quad \boldsymbol{L}^{(1)} \quad \right) = \left(\boldsymbol{c}_1 \right)\left(\boldsymbol{c}_1 \right)^{\mathsf{T}} + \mathbb{E}\left(\quad \right)$$



Rank 1 term

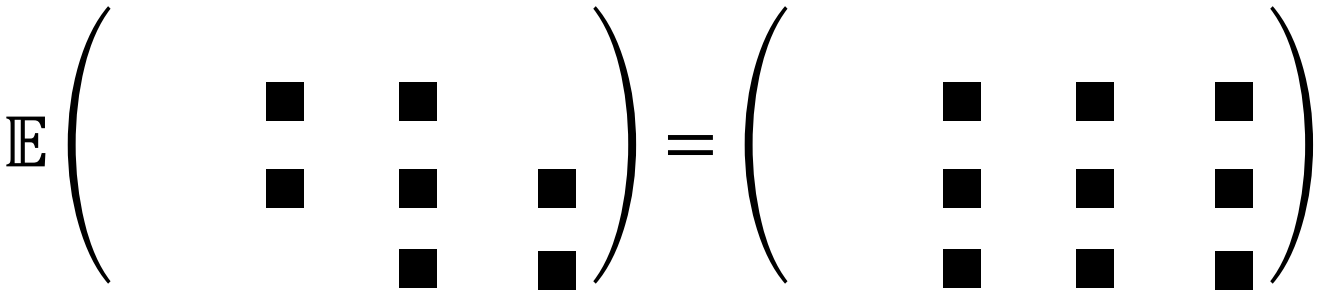
Remaining graph + sparsified clique

Then $= \boldsymbol{L}^{(0)}$

Approximating Matrices in Expectation

Let $\boldsymbol{L}^{(i)}$ be our approximation after i eliminations

If we ensure at each step



Sparsified clique Clique

Then

$$\mathbb{E}\, \boldsymbol{L}^{(i)} = \boldsymbol{L}^{(i-1)}$$

$$\mathbb{E}\left[\boldsymbol{L}^{(i)} - \boldsymbol{L}^{(i-1)} \middle| \text{ previous steps} \right] = \mathbf{0}$$

Predictable Quadratic Variation

Predictable quadratic variation

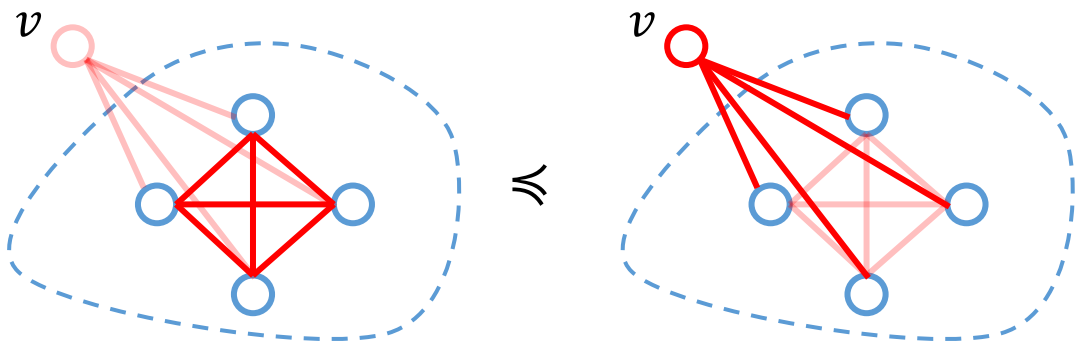
$$\boldsymbol{W} = \sum_i \sum_e \mathbb{E}\left[\left(\bar{\boldsymbol{X}}_e^{(i)}\right)^2 \mid \text{prev. steps}\right]$$

Want to show

$$\mathbb{P}[\|\boldsymbol{W}\| > \sigma^2] \leq \delta$$

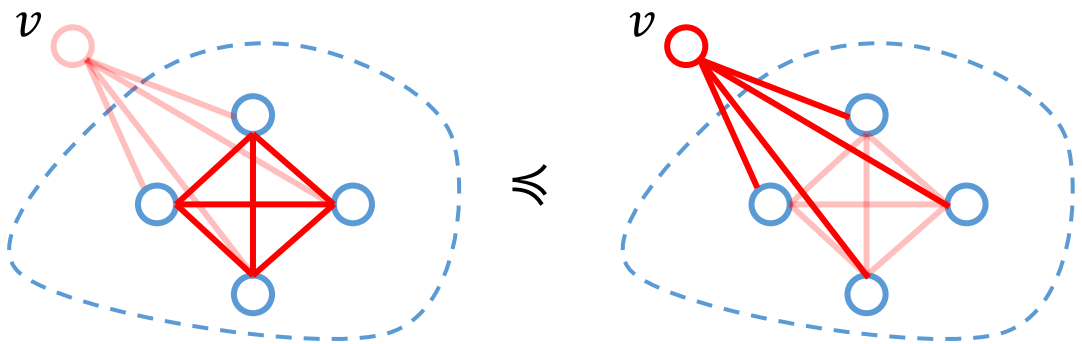
Promise: $\mathbb{E}\bar{\boldsymbol{X}}_e^2 \preccurlyeq r\bar{\boldsymbol{L}}_e$

Sample Variance



$$\sum_{e \in \text{ elim. clique of } v} \bar{L}_e \preccurlyeq \sum_{e \ni v} \bar{L}_e$$

Sample Variance



$$\mathbb{E}_v \sum_{e \in \text{ elim. clique of } v} \bar{L}_e \preccurlyeq \mathbb{E}_v \sum_{e \ni v} \bar{L}_e$$

$$= \frac{2\overline{\text{current lap.}}}{\#\text{ vertices}} \preccurlyeq \frac{4\bar{L}}{\#\text{ vertices}} = \frac{4I}{\#\text{ vertices}}$$

WARNING: only true w.h.p.

Sample Variance

Recall $\mathbb{E}\bar{X}_e^2 \lesssim r\bar{L}_e$

Putting it together

$$\mathbb{E}_v \sum_{e \in \text{ elim. clique of } v} \bar{L}_e \lesssim \mathbb{E}_v \sum_{e \in \text{ elim. clique of } v} \mathbb{E}\bar{X}_e^2 \lesssim r \frac{4I}{\#\text{ vertices}} \frac{4I}{\#\text{ vertices}}$$

variance in one round of elimination

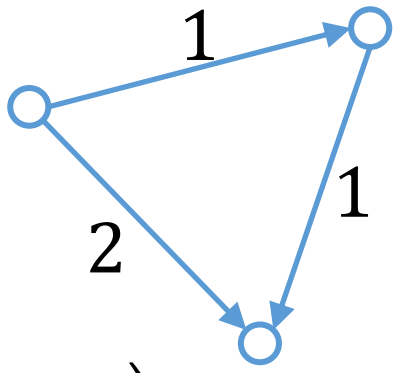
Sample Variance

$$\sum_{\text{rounds of elimination}} \text{variance} \preccurlyeq \sum_{\text{rounds of elimination}} r \cdot \frac{4\boldsymbol{I}}{\#\text{ vertices}}$$

$$\preccurlyeq 4\, r \log n \cdot \boldsymbol{I}$$

Directed Laplacian of a Graph

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

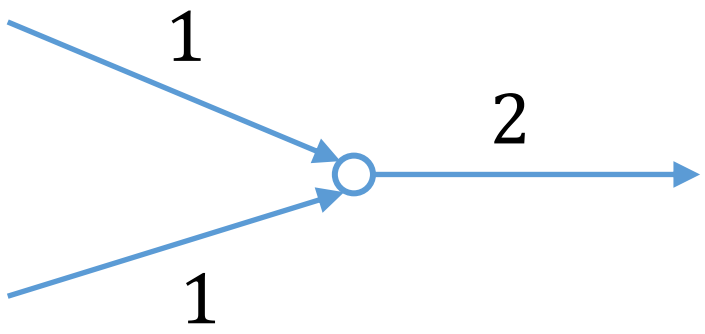


$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ -2 & 0 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & -1 & 0 \end{pmatrix}$$

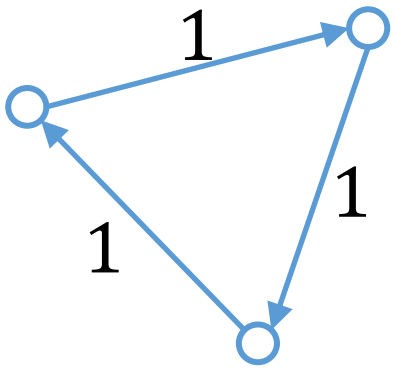
(Weighted) Eulerian Graph



weighted in-degree = weighted out-degree

Eulerian Laplacian of a Graph

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$



$$\begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 0 \end{pmatrix}$$

$$\begin{pmatrix} 0 & 0 & -1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} 1 & 0 & -1 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

Solving an Eul. Laplacian Linear Equation

$$\boldsymbol{L}\boldsymbol{x} = \boldsymbol{b}$$

Gaussian Elimination

Find $\mathfrak{L}, \mathfrak{U}$, lower and upper triangular matrices, s.t.

$$\mathfrak{L}\mathfrak{U} = \boldsymbol{L}$$

Then

$$\boldsymbol{x} = \mathfrak{U}^{-1}\mathfrak{L}^{-1}\boldsymbol{b}$$

Easy to apply $\mathfrak{U}^{-1}$ and $\mathfrak{L}^{-1}$

Solving an Eul. Laplacian Linear Equation

$$\boldsymbol{Lx} = \boldsymbol{b}$$

Approximate Gaussian Elimination [CKKPPRS FOCS'18]
Find $\mathfrak{L}, \mathfrak{U}$, lower and upper triangular matrices, s.t.

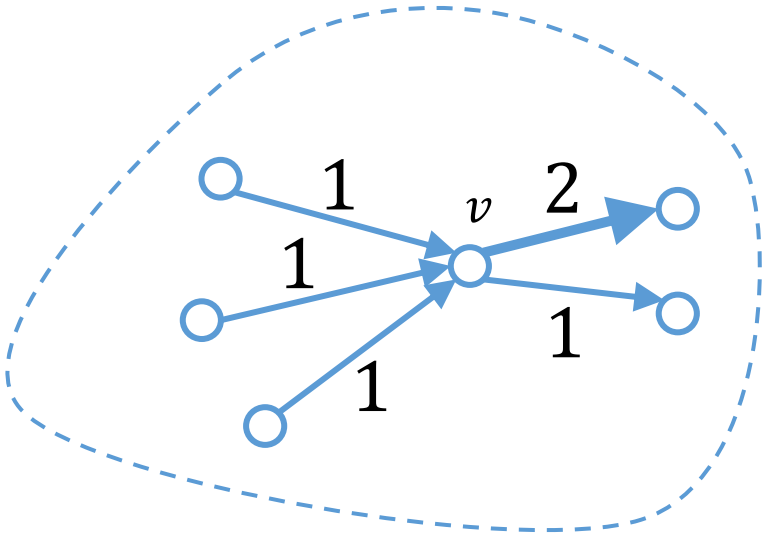
$$\mathfrak{L}\mathfrak{U} \approx \boldsymbol{L}$$

$\mathfrak{L}, \mathfrak{U}$ are sparse.

$O\left(\log \frac{1}{\varepsilon}\right)$ iterations to get
ε-approximate solution $\tilde{\boldsymbol{x}}$.

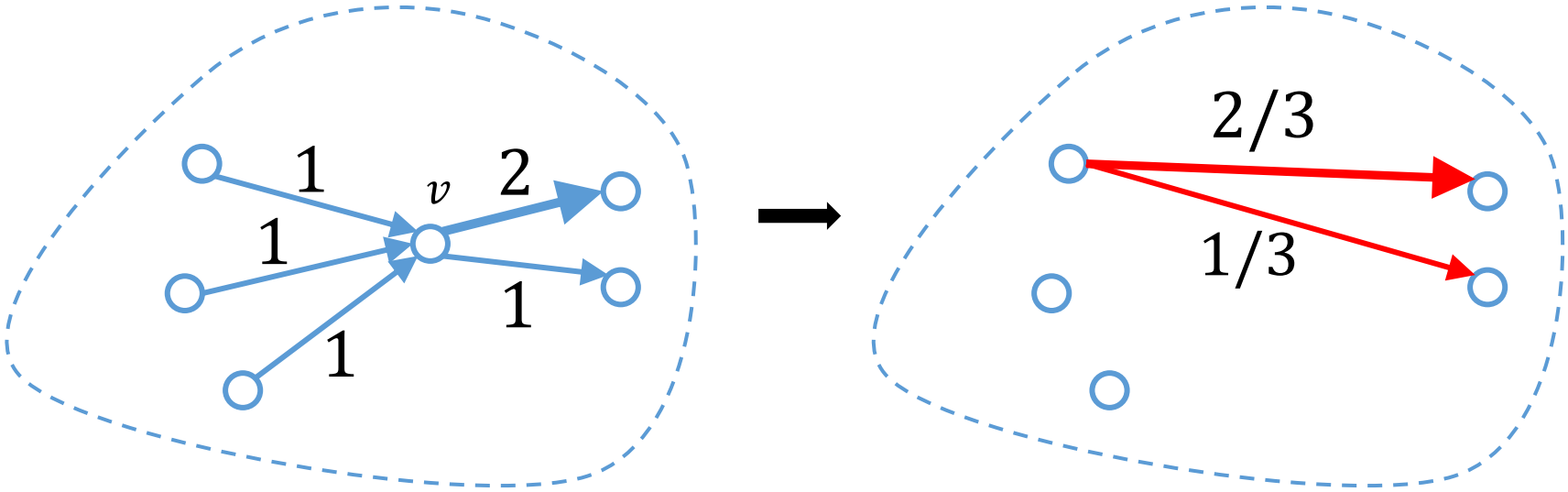
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



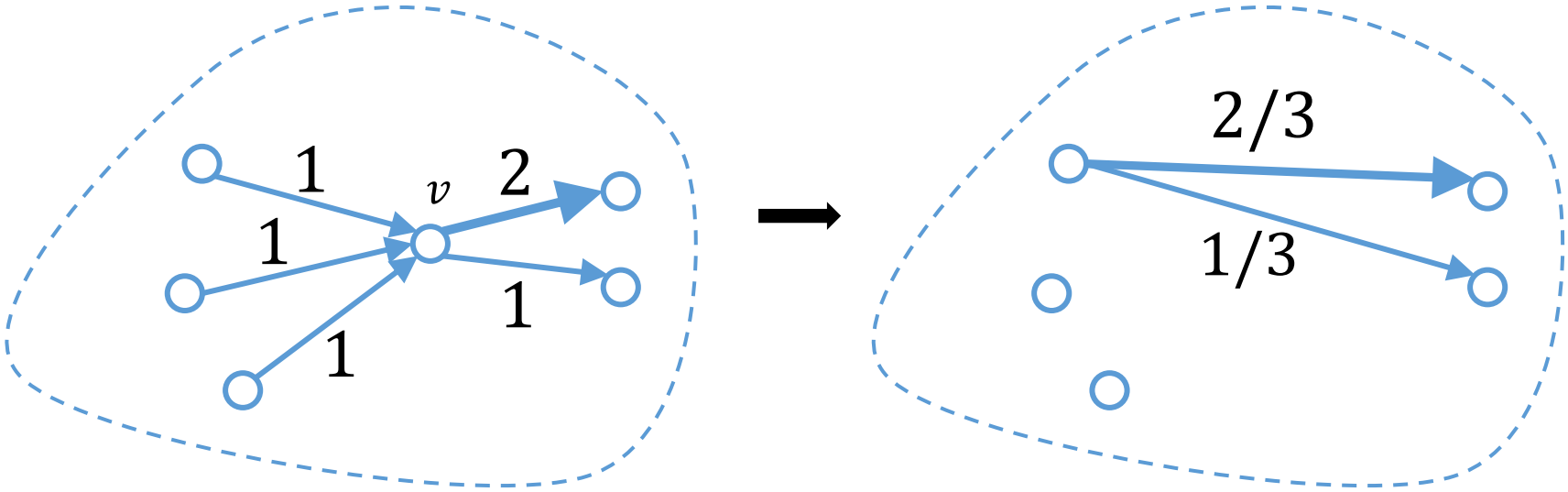
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



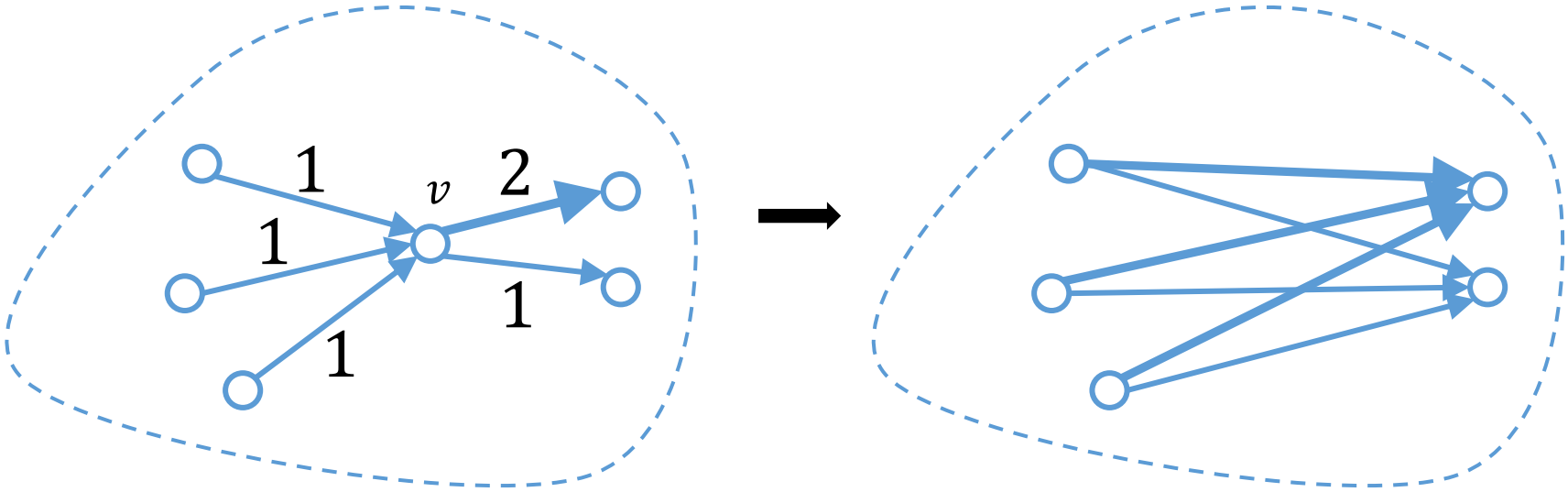
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



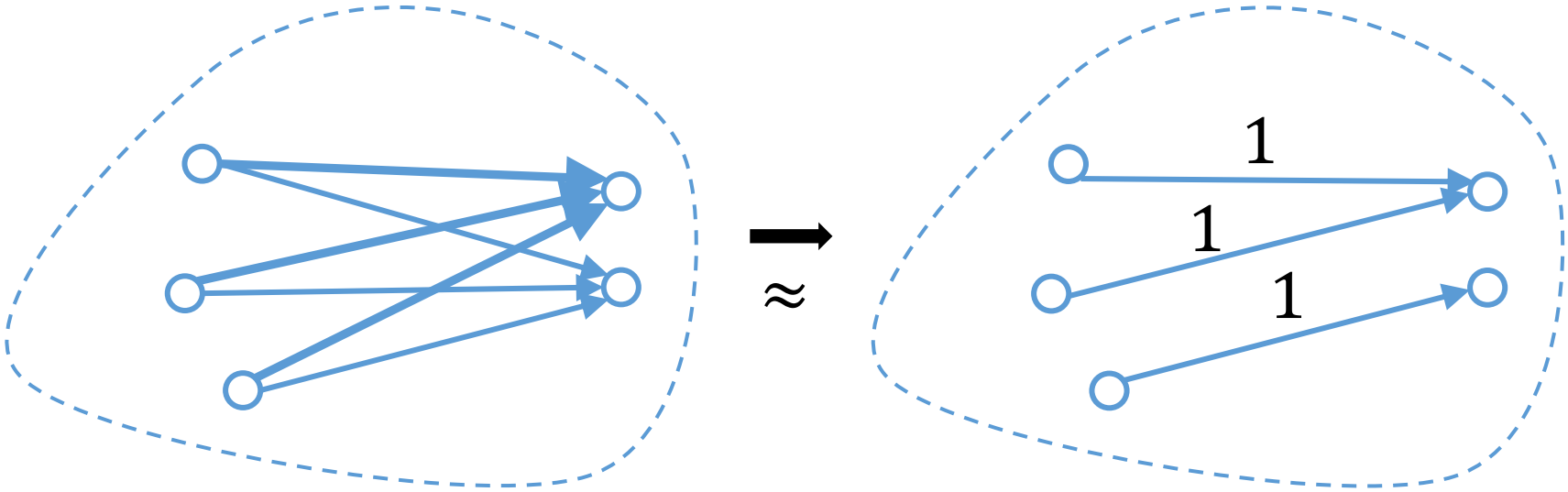
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



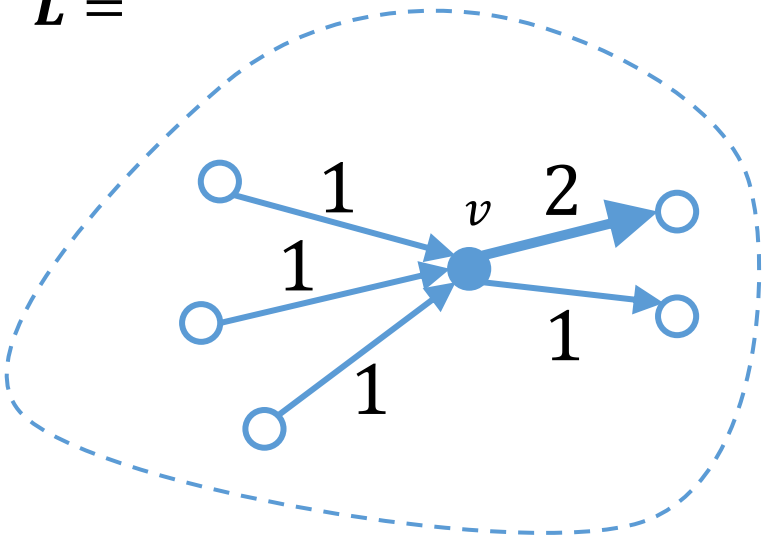
Sparsify

correct in expectation

output **always** Eulerian

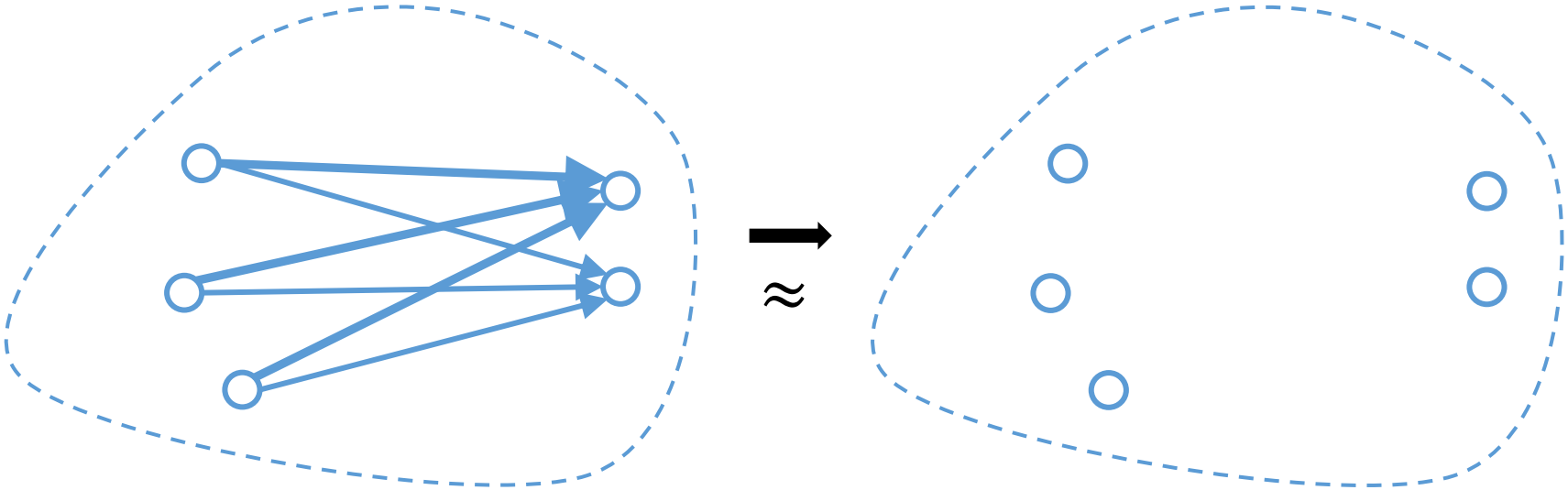
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



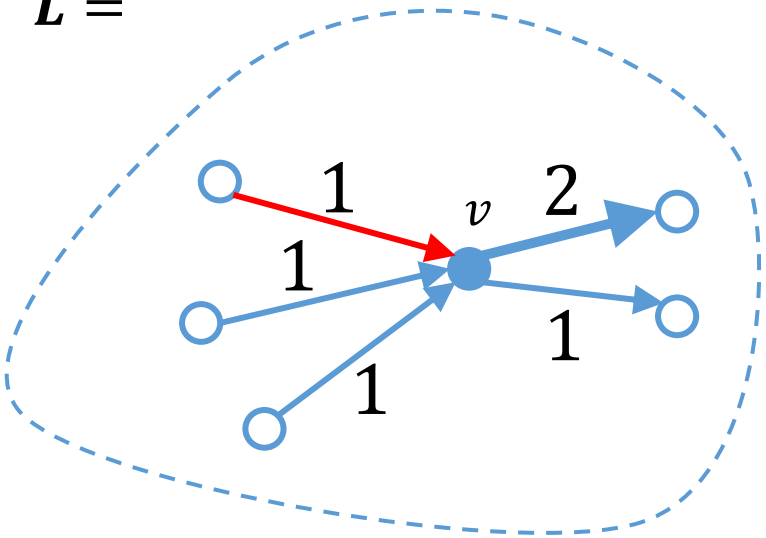
While edges remain:

1. Pick a minimum weight edge
2. Pair with random neighbor
3. Reduce mass on both by $w_{\min}$



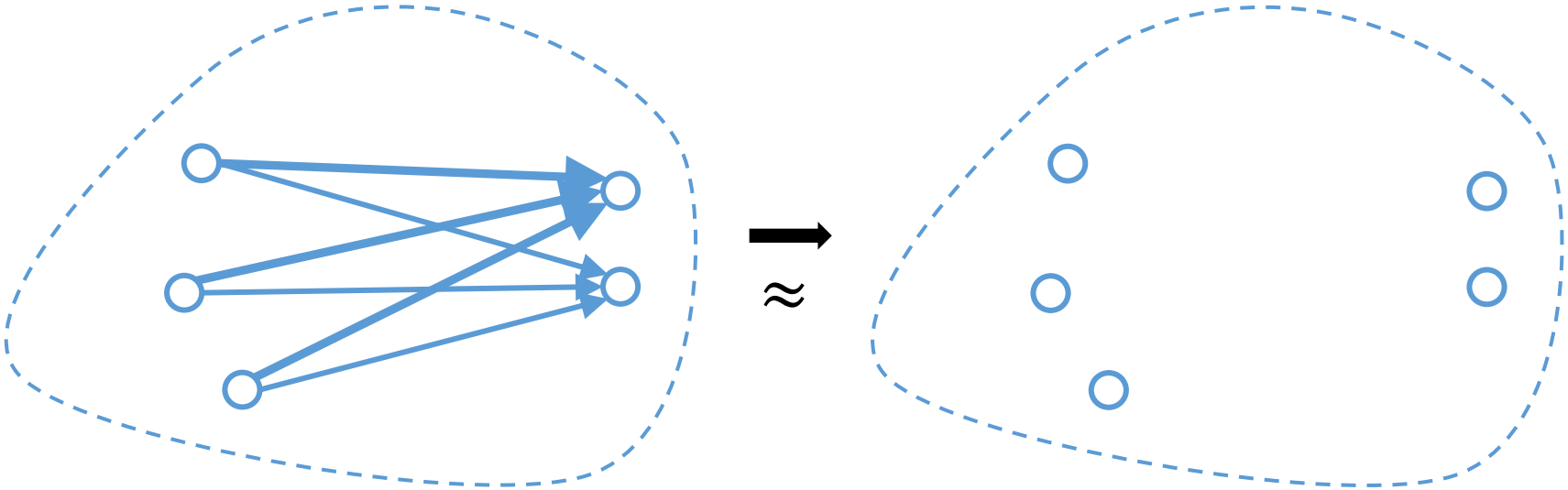
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



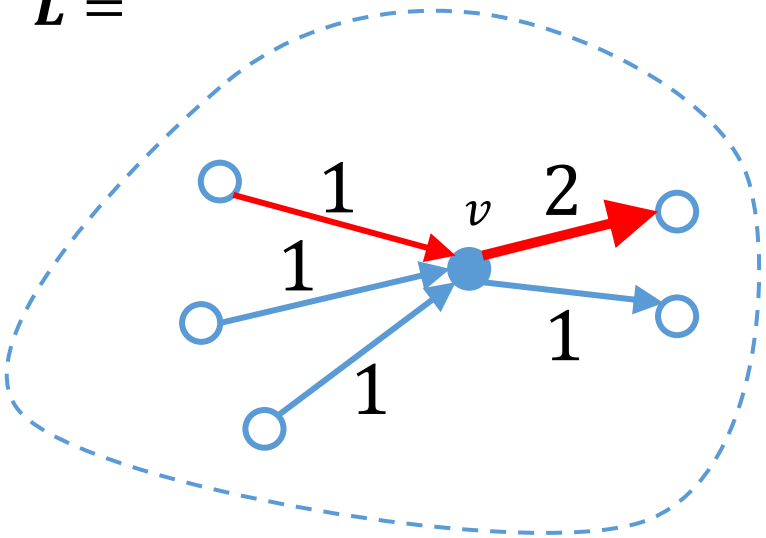
While edges remain:

1. Pick a minimum weight edge
2. Pair with random neighbor
3. Reduce mass on both by $w_{\min}$



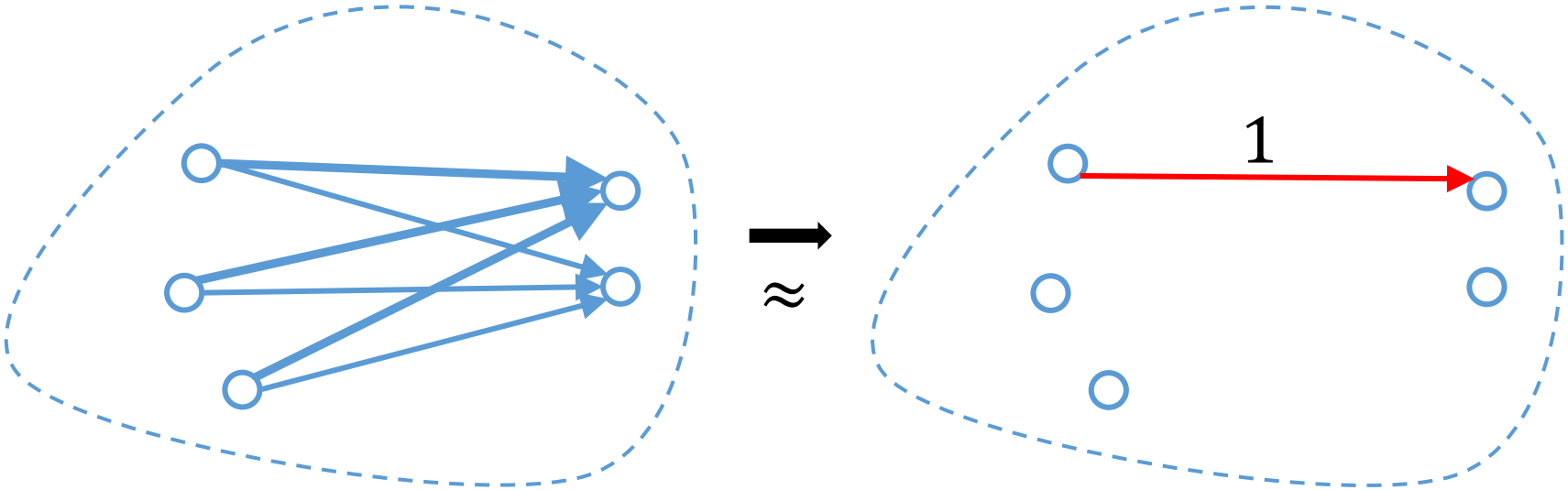
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



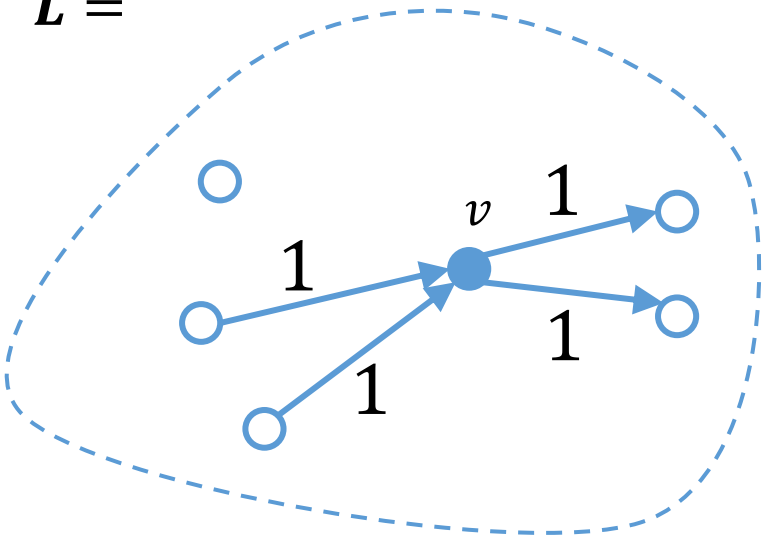
While edges remain:

1. Pick a minimum weight edge
2. Pair with random neighbor
3. Reduce mass on both by $w_{\min}$



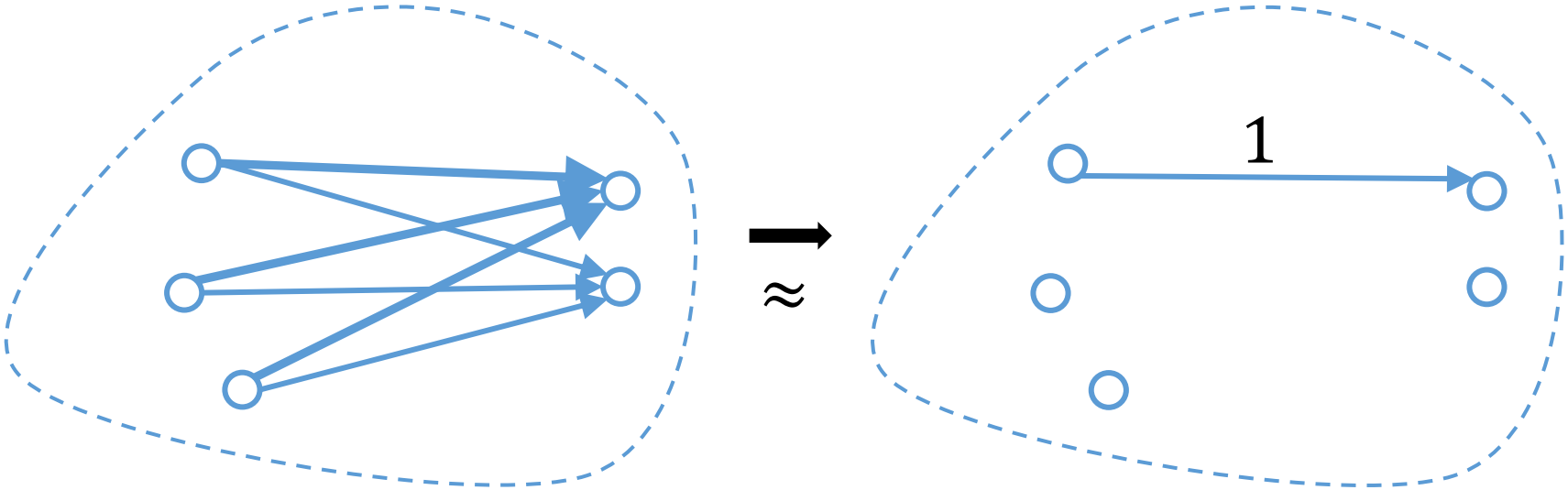
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



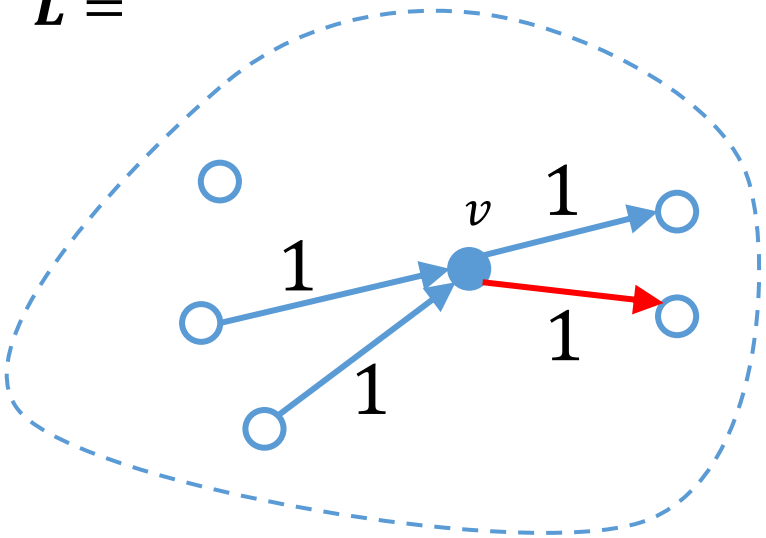
While edges remain:

1. Pick a minimum weight edge
2. Pair with random neighbor
3. Reduce mass on both by $w_{\min}$



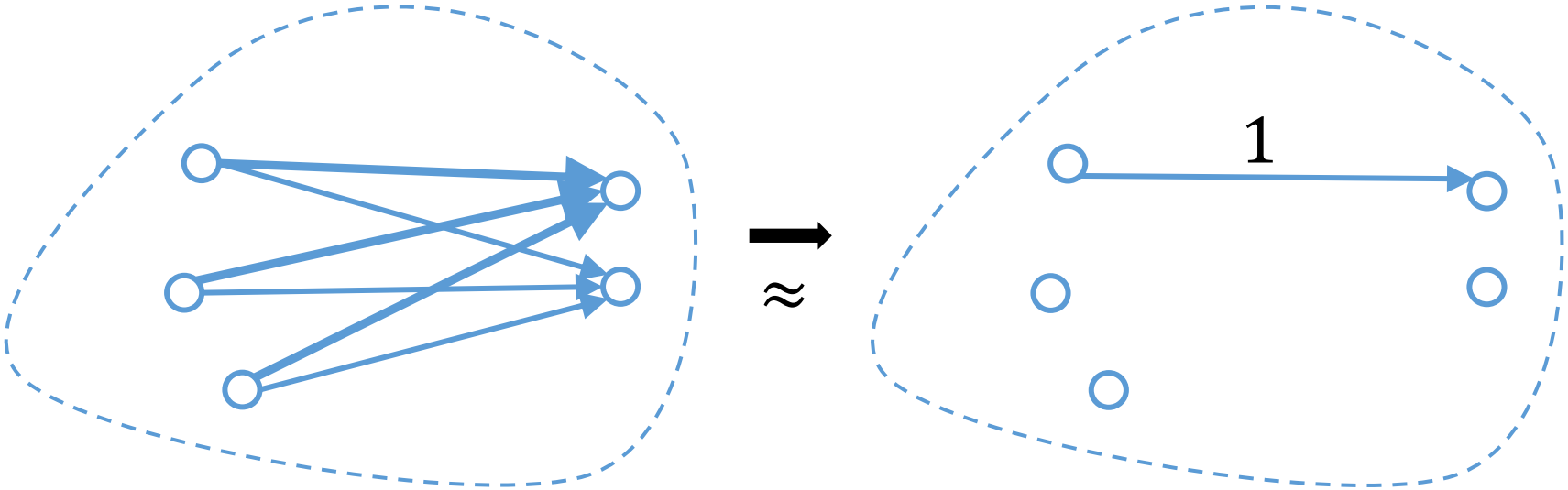
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



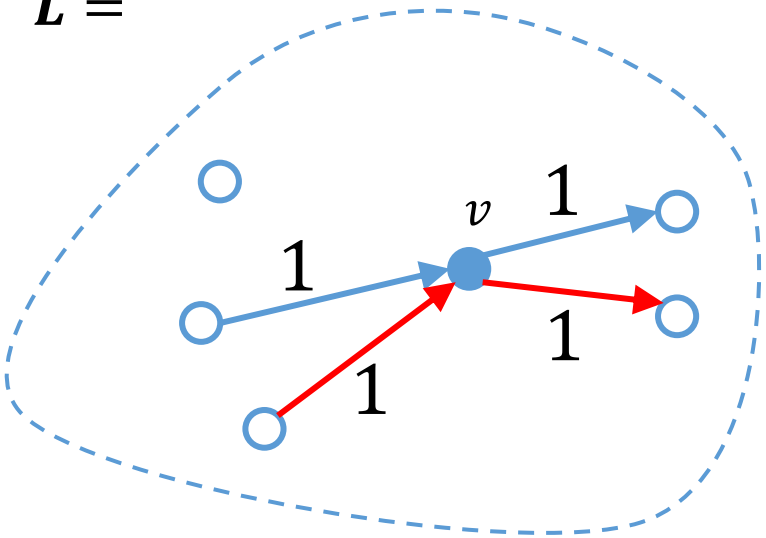
While edges remain:

1. Pick a minimum weight edge
2. Pair with random neighbor
3. Reduce mass on both by $w_{\min}$



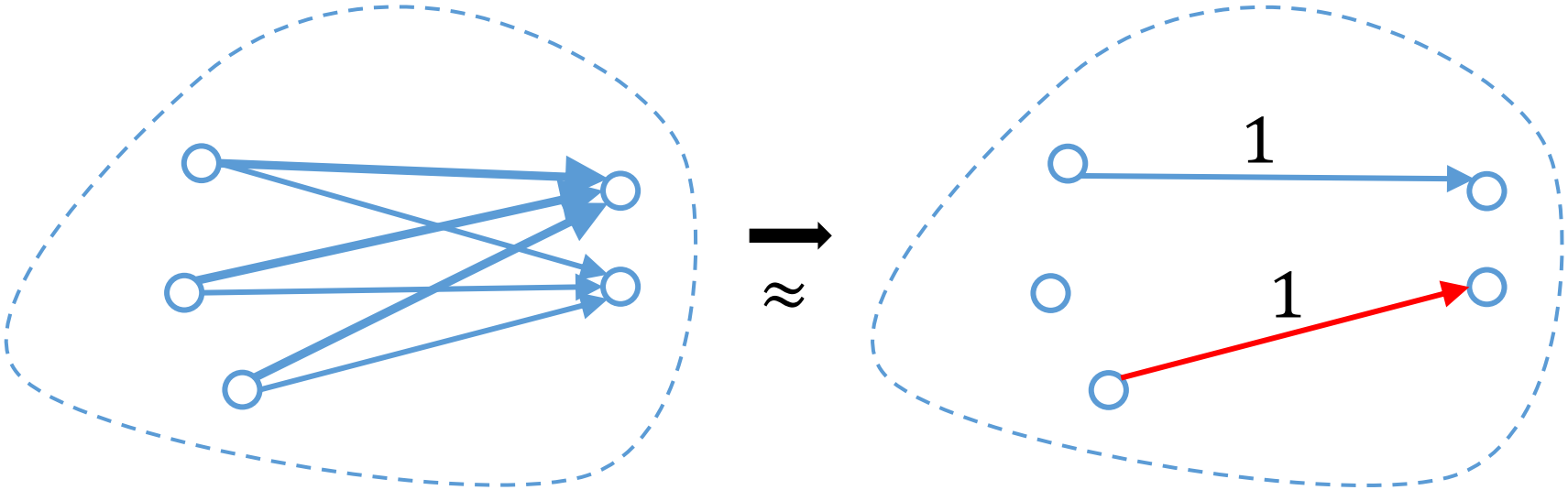
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



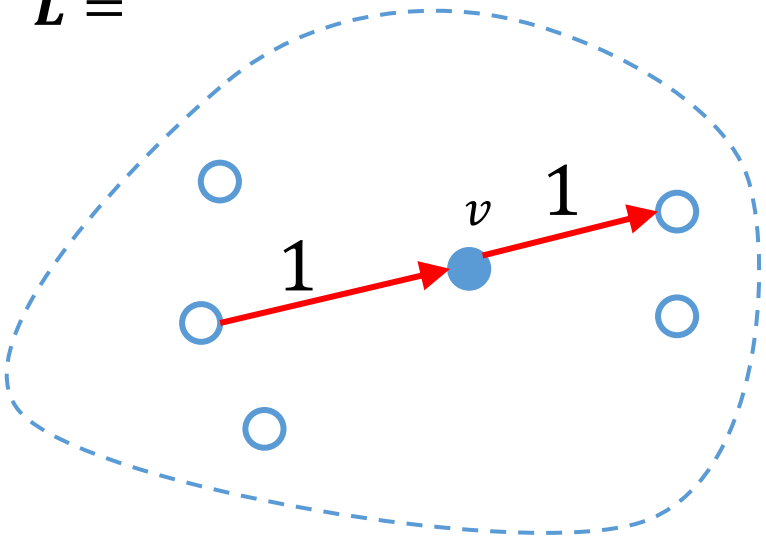
While edges remain:

1. Pick a minimum weight edge
2. Pair with random neighbor
3. Reduce mass on both by $w_{\min}$



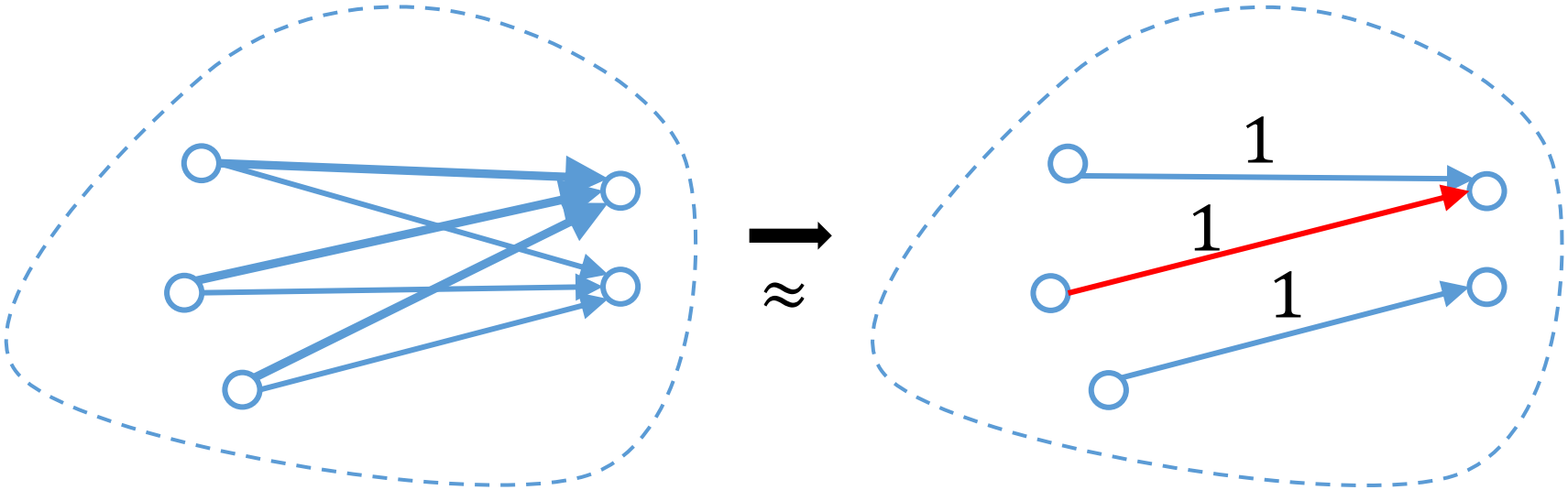
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$



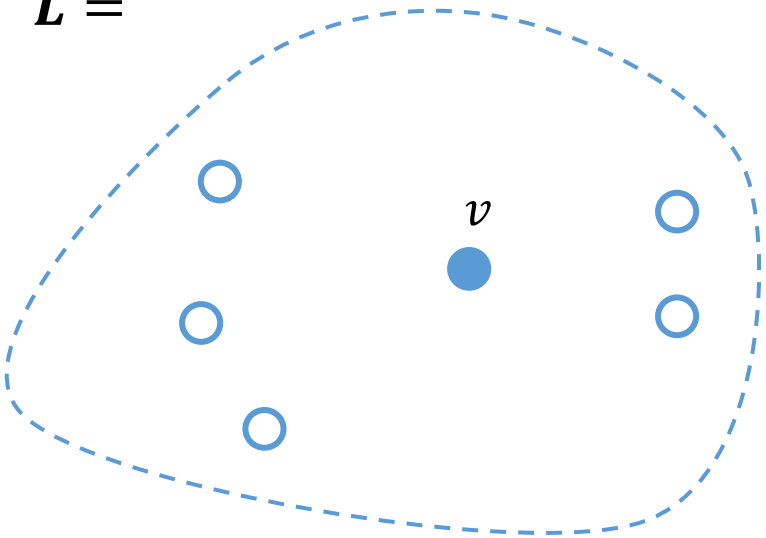
While edges remain:

1. Pick a minimum weight edge
2. Pair with random neighbor
3. Reduce mass on both by $w_{\min}$



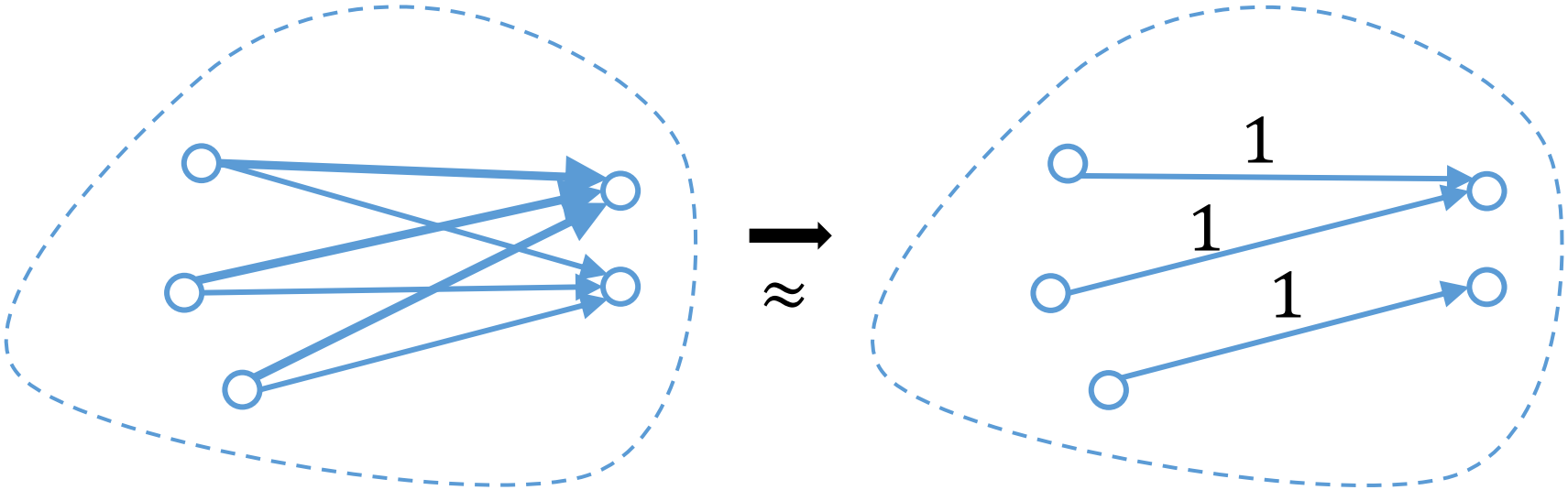
Gaussian Elimination on Eulerian Laplacians

$\boldsymbol{L} =$

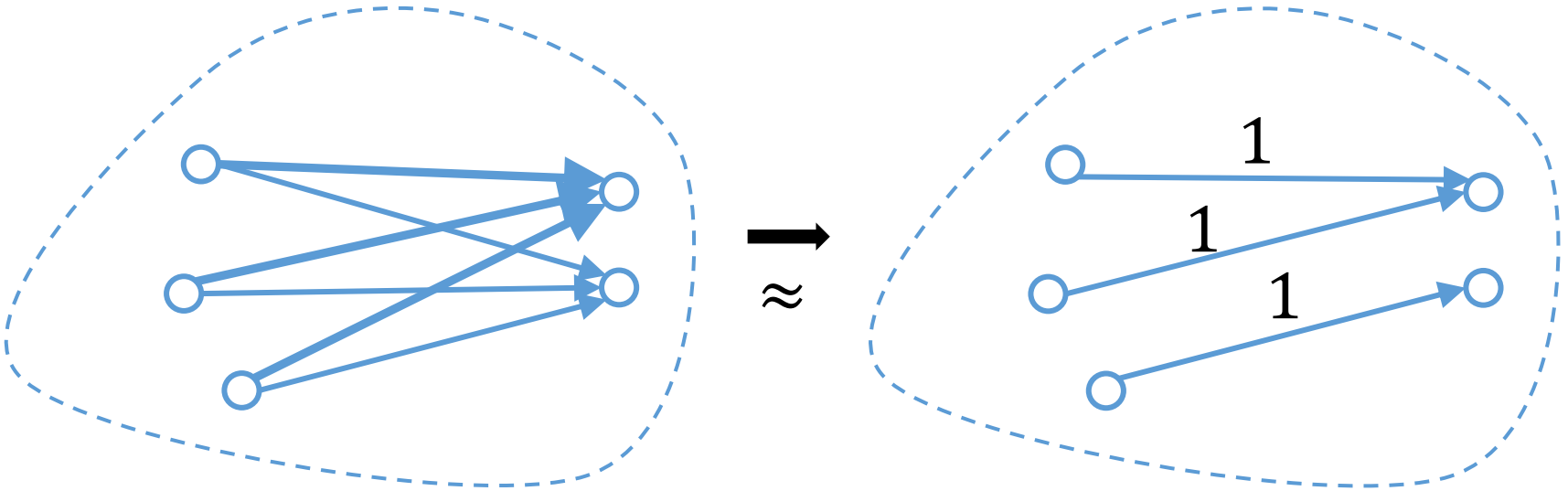


While edges remain:

1. Pick a minimum weight edge
2. Pair with random neighbor
3. Reduce mass on both by $w_{\min}$



Gaussian Elimination on Eulerian Laplacians



Our sparsification

correct in expectation

output **always** Eulerian

Average over $\log^3(n)$ runs to reduce variance

Approximating Matrices by Sampling

Goal

$$\mathfrak{LU} \approx L$$

Approach

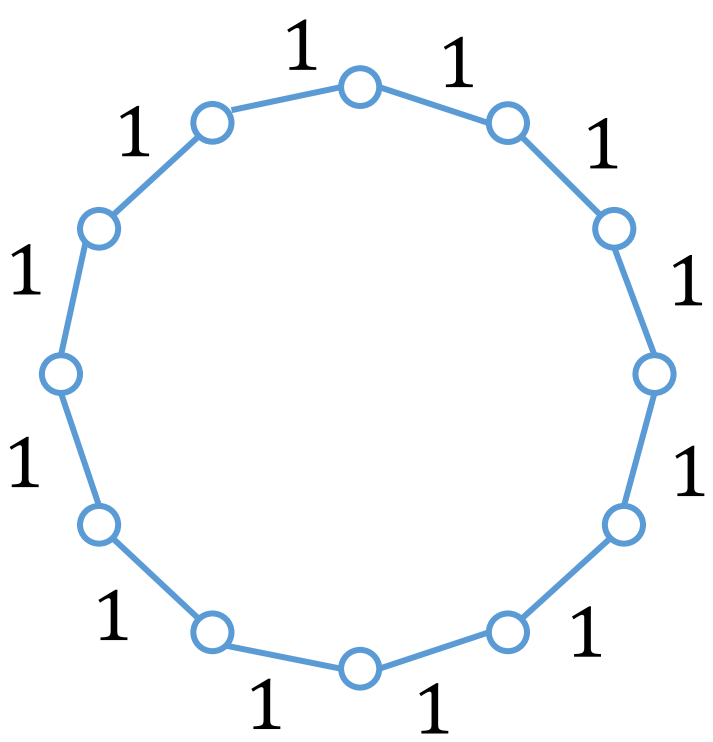
1. $\mathbb{E}\,\mathfrak{LU} = L$
2. Show $\mathfrak{LU}$ concentrated around expectation

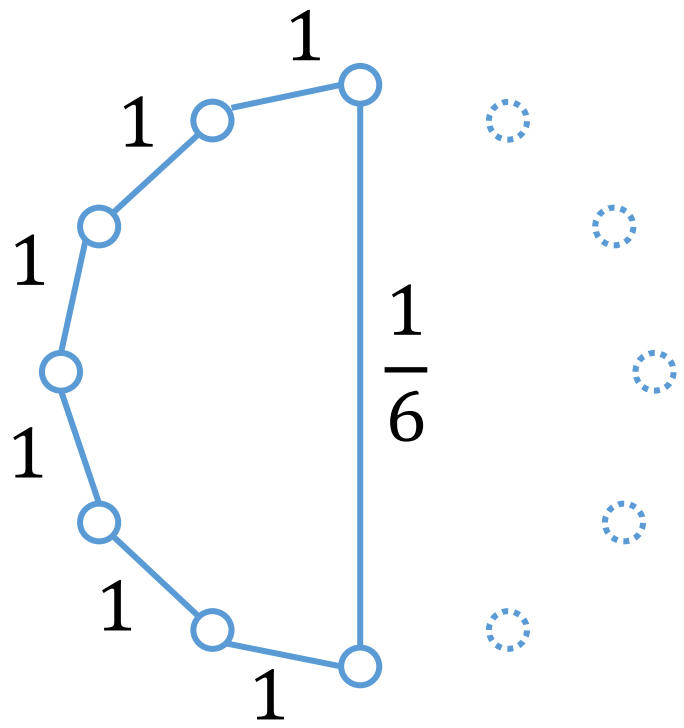
Gives $\mathfrak{LU} \approx L$ w. high probability (?)

But what does $\approx$ mean? L is not PSD?!?

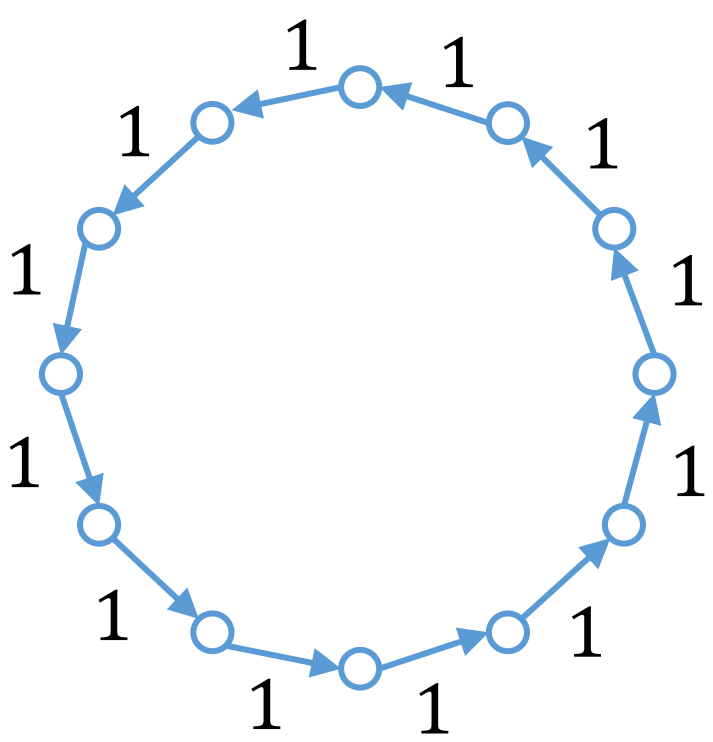
Come to my talk and I'll tell you!

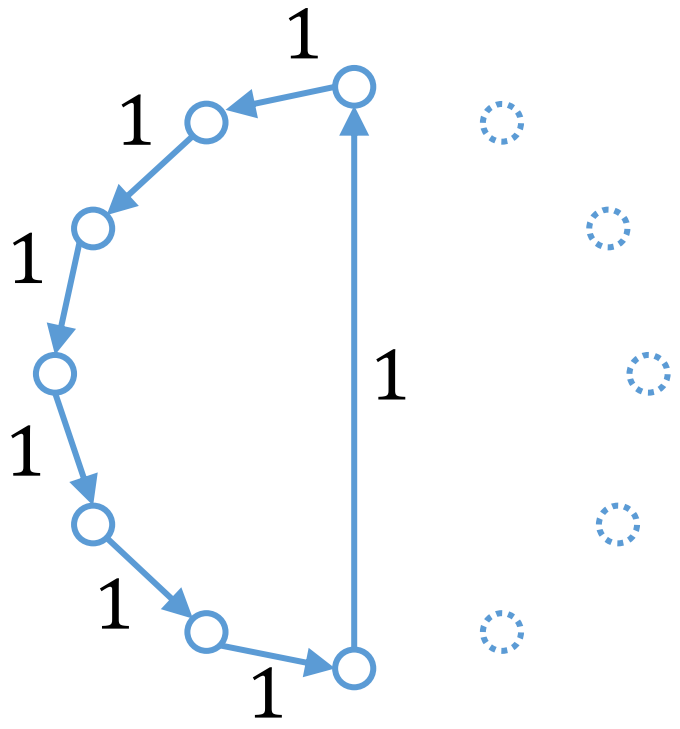
Difficulties with Approximation



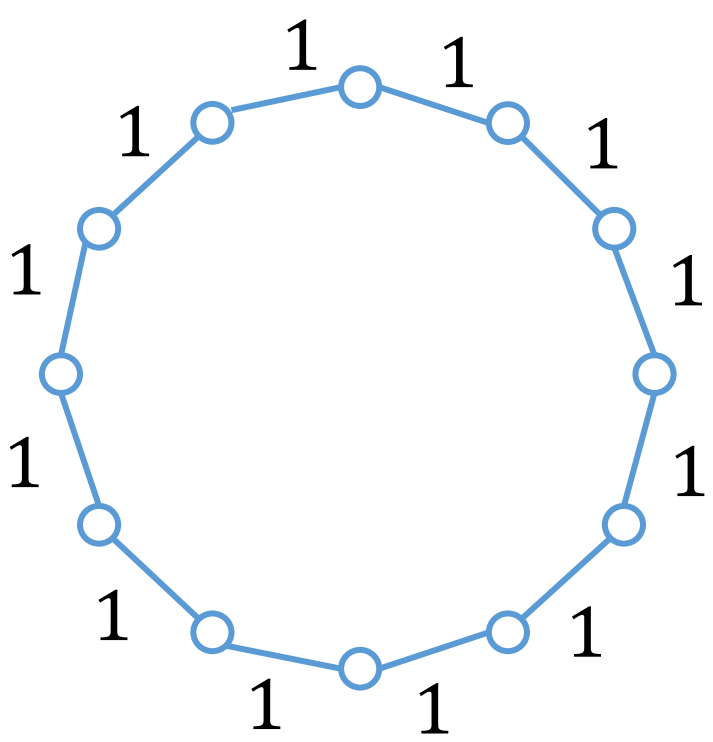


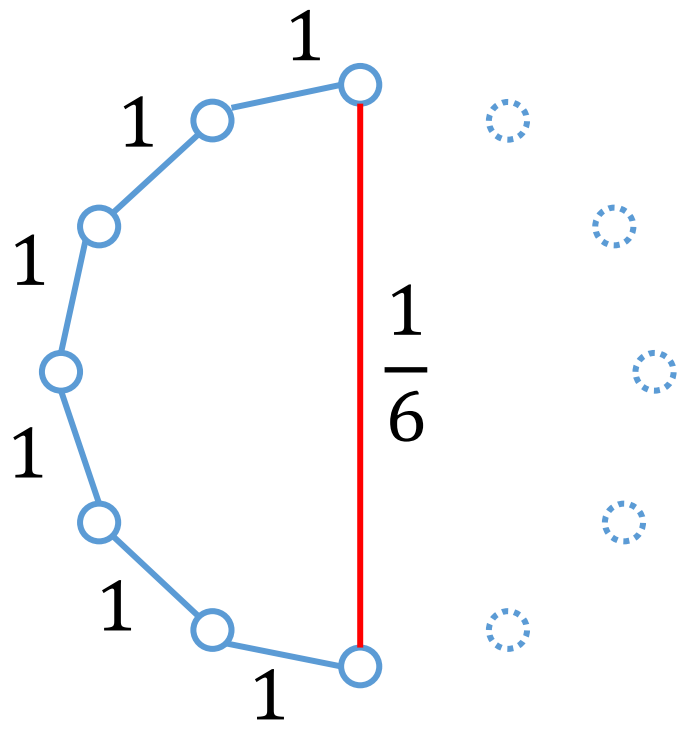
Difficulties with Approximation





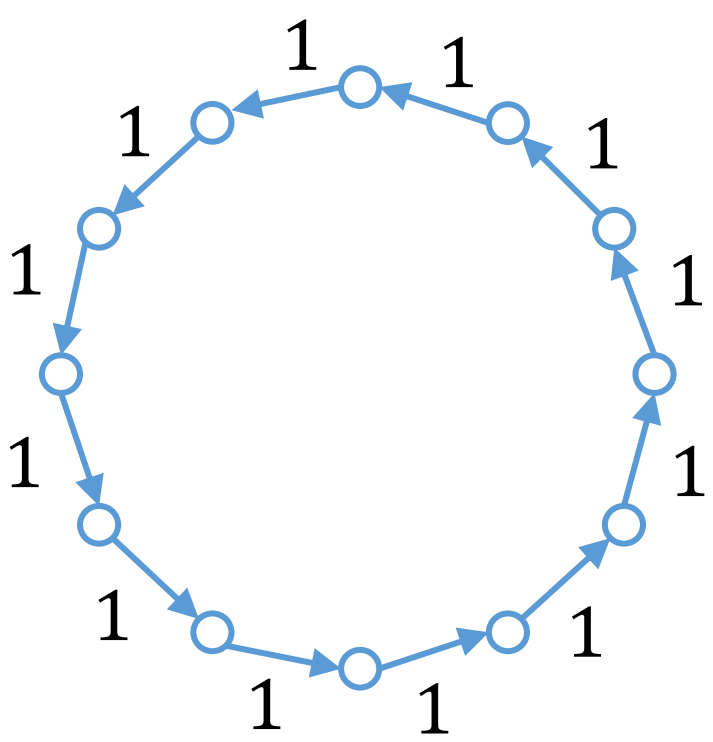
Difficulties with Approximation

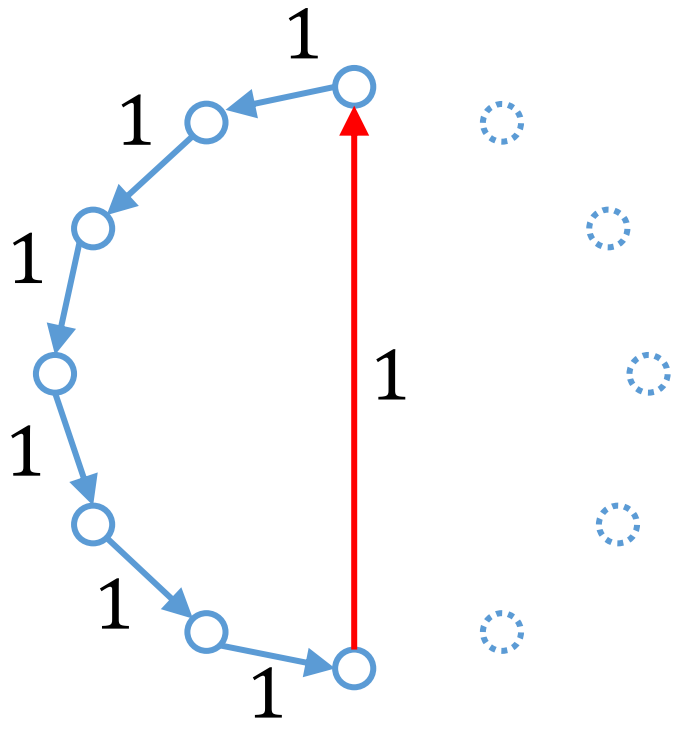




Variance manageable

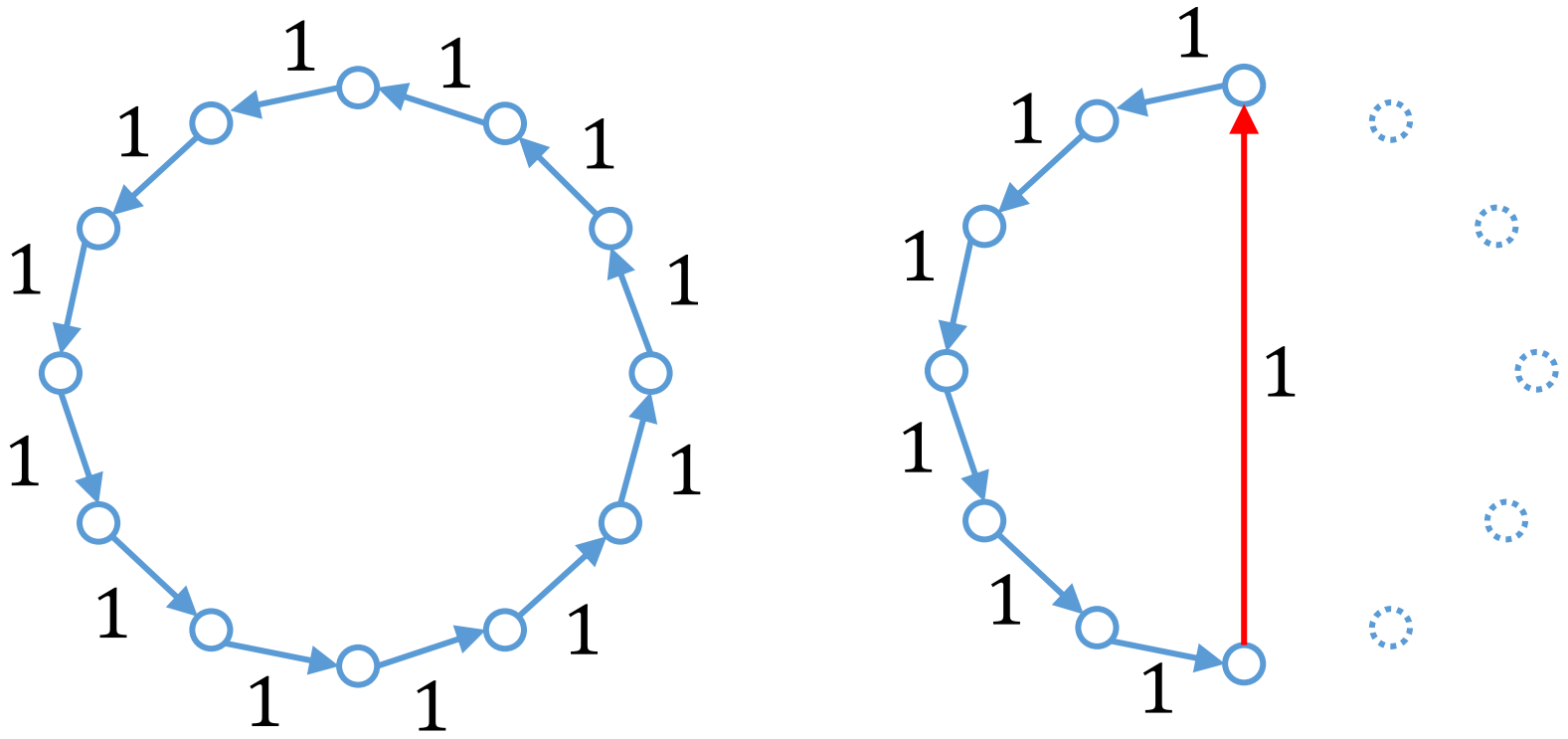
Difficulties with Approximation





Variance problematic!

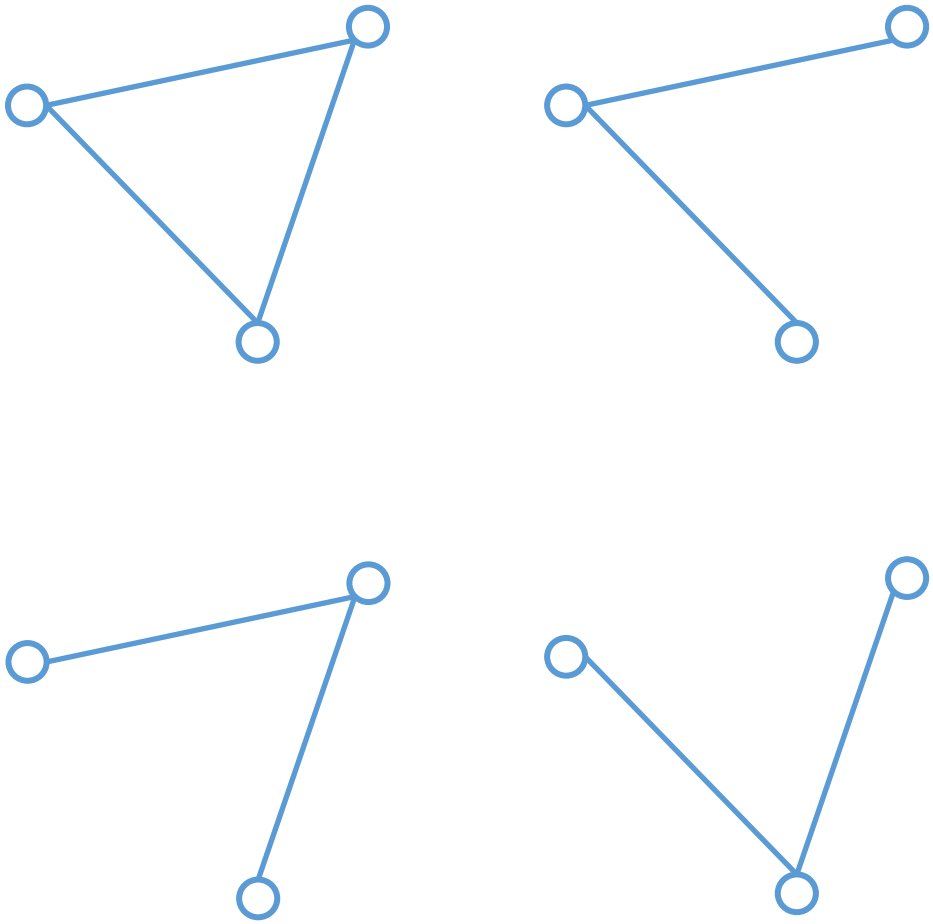
Difficulties with Approximation



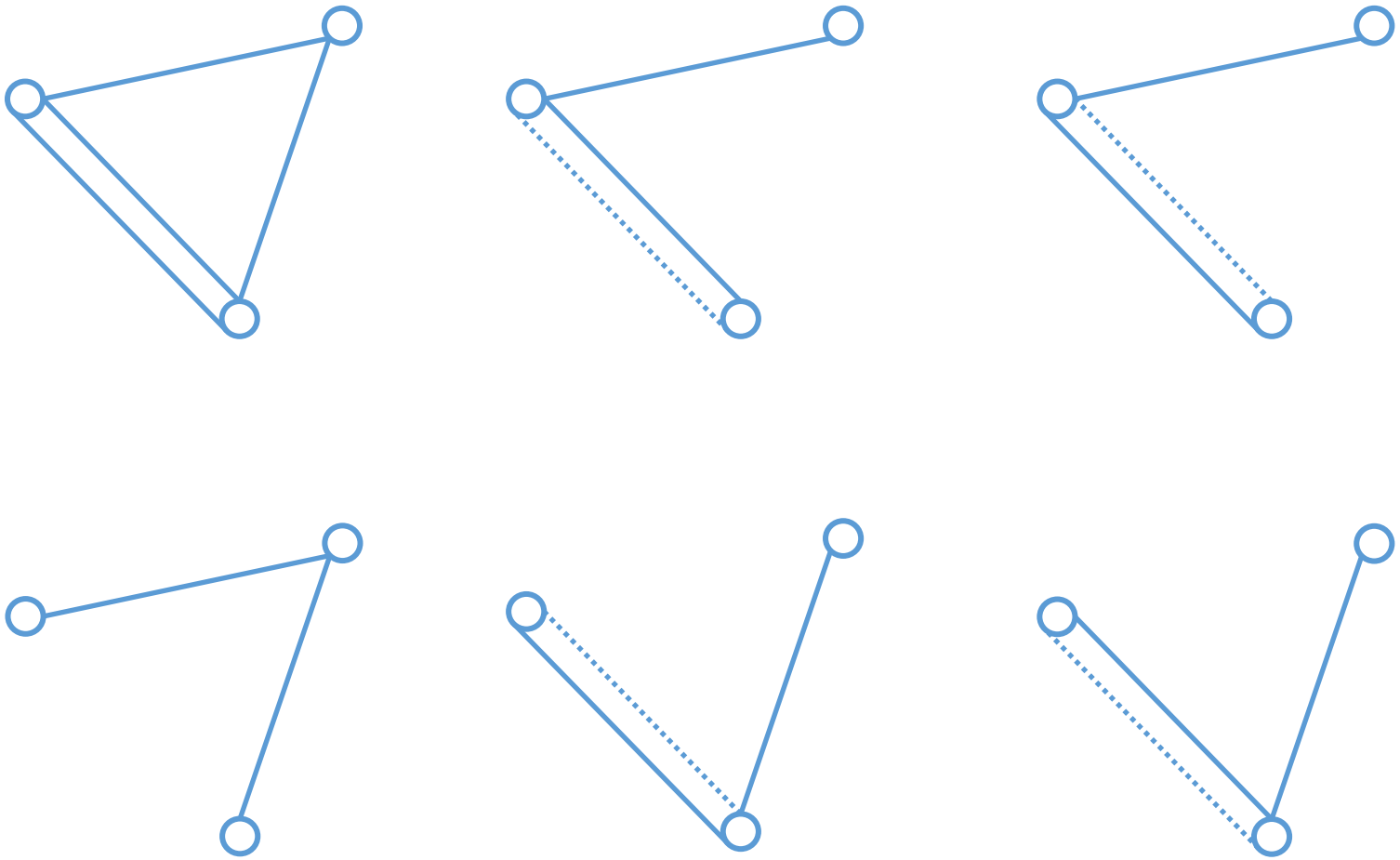
Come to my talk and hear the rest of the story!

[Cohen-Kelner-Kyng-Peebles-Peng-Rao-Sidford FOCS'18]

Spanning Trees of a Graph



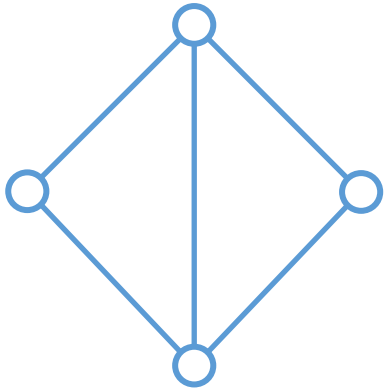
Spanning Trees of a (Multi)-Graph



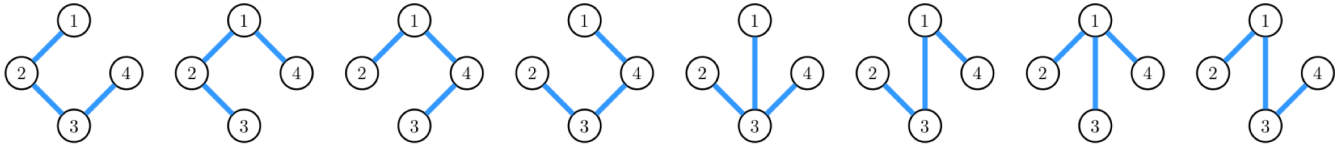
Pick a random tree?

Random Spanning Trees

Graph G



Tree distribution

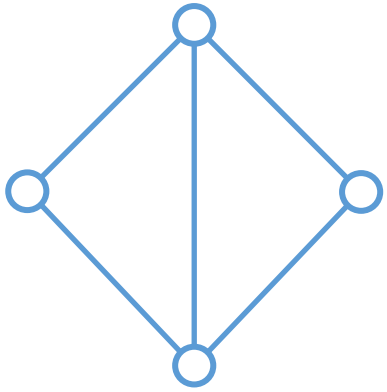


Marginal probability of edge: $p_e = \|\bar{\boldsymbol{L}}_e\|$

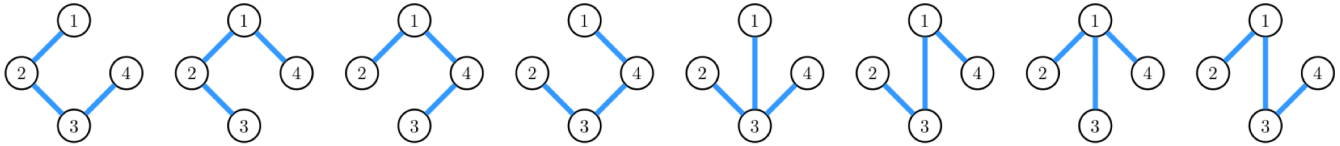
Similar to sparsification?!

Random Spanning Trees

Graph G



Tree distribution



Marginal probability of edge: $p_e = \|\bar{\boldsymbol{L}}_e\|$

$\boldsymbol{L}_G = \sum_{e \in G} \boldsymbol{L}_e$

Consider $\boldsymbol{L}_T = \sum_{e \in T} \frac{1}{p_e} \boldsymbol{L}_e$ $\qquad\qquad$ $\mathbb{E}\boldsymbol{L}_T = \boldsymbol{L}_G$

Random Spanning Trees

Concentration?

$\boldsymbol{L}_T \approx_{0.5} \boldsymbol{L}_G$? NO

$\boldsymbol{L}_T \preccurlyeq \log n \, \boldsymbol{L}_G$? YES, w.h.p

$\frac{1}{t}\sum_{i=1}^{t} \boldsymbol{L}_{T_i} \approx_{\varepsilon} \boldsymbol{L}_G \quad , \quad t = \varepsilon^{-2}\log^2 n$

[Kyng-Song FOCS'18]

Come see my talk!

Sometimes You Get a Martingale for Free

Random variables $Z_1, \ldots, Z_k$

Prove concentration for $f(Z_1, \ldots, Z_k)$

where f is ``stable" under small changes to $Z_1, \ldots, Z_k$

$f(Z_1, \ldots, Z_k) \approx \mathbb{E} f(Z_1, \ldots, Z_k)$?

Doob Martingales

Random variables $Z_1, \ldots, Z_k$ NOT indep.

Prove concentration for $f(Z_1, \ldots, Z_k)$

where f is ``stable" under small changes to $Z_1, \ldots, Z_k$

Pick outcome $z_1, \ldots, z_k$

$\mathbb{E}[f(Z_1, \ldots, Z_k)]$

$$\rightarrow \mathbb{E}[f(Z_1, \ldots, Z_k) | Z_1 = z_1, Z_2 = z_2]$$

$$\rightarrow f(z_1, \ldots, z_k)$$

Doob Martingales

$$\mathbb{E}[f(Z_1, \dots, Z_k)] \quad \to \mathbb{E}[f(Z_1, \dots, k)|Z_1 = z_1]$$
$$\to \mathbb{E}[f(Z_1, \dots, Z_k)|Z_1 = z_1, Z_2 = z_2]$$
$$\to f(z_1, \dots, z_k)$$

$$\mathbb{E}[f(Z_1, \dots, Z_k)] = \mathbb{E}_{z_1}\left[\mathbb{E}[f(Z_1, \dots, Z_k)|Z_1 = z_1]\right]$$

$$Y_0 = \mathbb{E}[f(Z_1, \dots, Z_k)]$$
$$Y_1 = \mathbb{E}[f(Z_1, \dots, Z_k)|Z_1 = z_1]$$
$$Y_2 = \mathbb{E}[f(Z_1, \dots, Z_k)|Z_1 = z_1, Z_2 = z_2]$$

$$\mathbb{E}[Y_{i+1} - Y_i|\text{prev. steps}] = 0$$

Martingale! Despite $Z_1, \dots, Z_k$ NOT indep.

Concentration of Random Spanning Trees

Random variables $Z_1, \dots, Z_{n-1}$

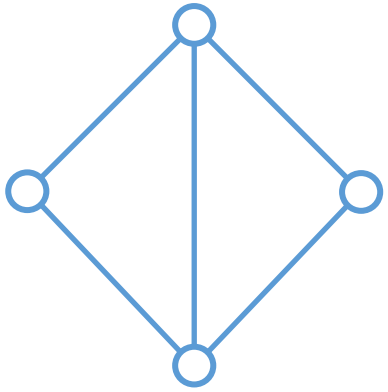
Positions of $n-1$ edges of random spanning tree

$f(Z_1, \dots, Z_{n-1}) = \boldsymbol{L}_T$

Similar to [Peres-Pemantle '14]

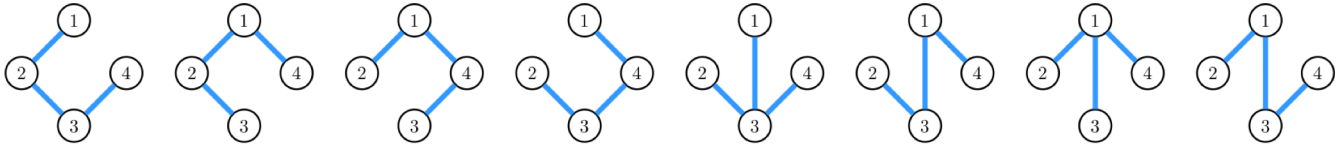
Conditioning Changes Distributions?

Graph



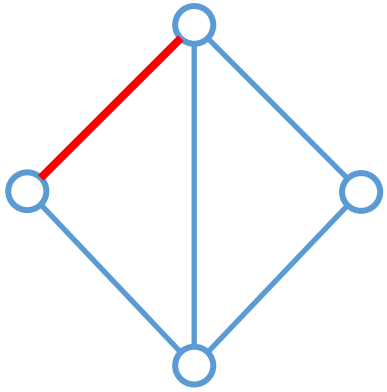
Tree distribution

All



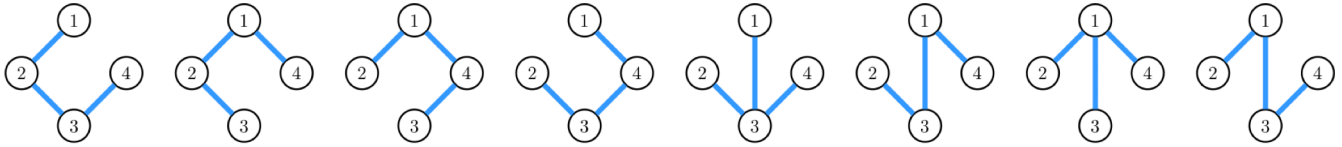
Conditioning Changes Distributions?

Graph



Tree distribution

All



Conditional

Summary

Matrix martingales: a natural fit for algorithmic analysis

Understanding the Predictable Quadratic Variation is key

Some results using matrix martingales

Cohen, Musco, Pachocki ’16 – online row sampling

Kyng, Sachdeva ’17 – approximate Gaussian elimination

Kyng, Peng, Pachocki, Sachdeva ’17 – semi-streaming graph sparsification

Cohen, Kelner, Kyng, Peebles, Peng, Rao, Sidford ’18 – solving Directed Laplacian eqs.

Kyng, Song ’18 – Matrix Chernoff bound for negatively dependent random variables

Thanks!

This is really the end

Older slides