Matrix Martingales in Randomized Numerical Linear Algebra

Speaker **Rasmus Kyng** Harvard University

Sep 27, 2018

Concentration of Scalar Random Variables

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

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. $|X_i| \le r$
4. $\sum_i \mathbb{E}X_i^2 \le \sigma^2$

gives

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

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

Concentration of Scalar **Martingales**

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

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

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

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

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

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

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

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

gives

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

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

$$\leq \tau + \delta$$

Concentration of Matrix Martingales

Matrix Freedman's Inequality (Tropp 11)

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

Predictable quadratic variation

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

Laplacian Matrices

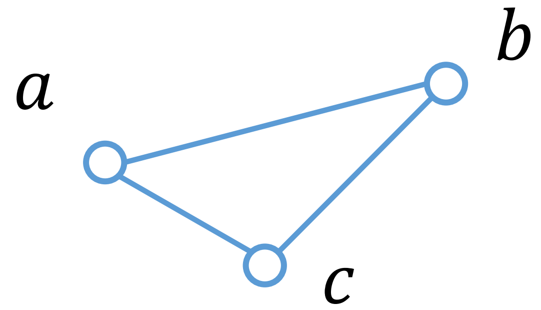
Graph $G = (V, E)$

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

$n = |V|$

$m = |E|$

$n \times n$ matrix



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

Laplacian Matrices

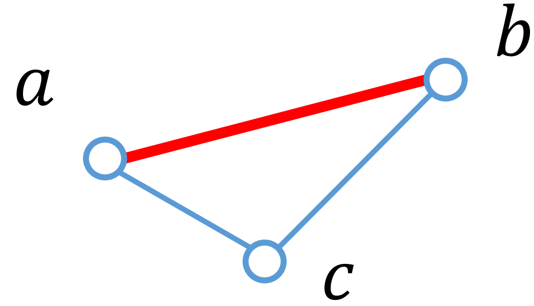
Graph $G = (V, E)$

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

$n = |V|$

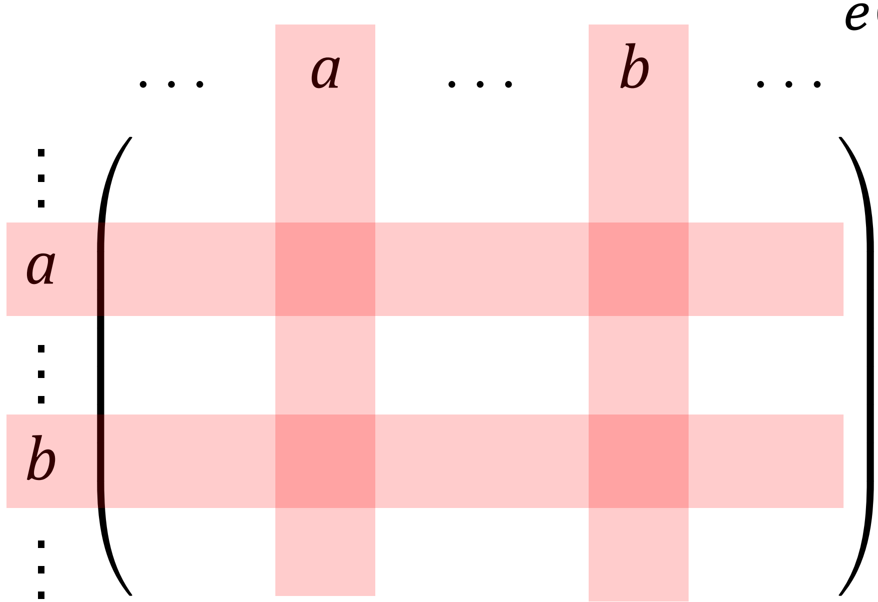
$m = |E|$

$n \times n$ 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

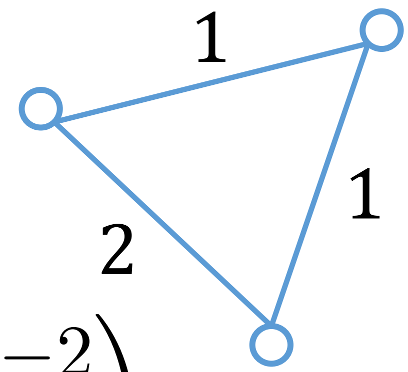
Symmetric matrix L

All off-diagonals are non-positive and

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

Laplacian of a Graph

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



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

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

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

Laplacian Matrices

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

$\vdots$

[KS16]: simple algorithm

Solving a Laplacian Linear Equation

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

Gaussian Elimination

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

$$\boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{L}$$

Then

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

Easy to apply $\boldsymbol{U}^{-1}$ and $\boldsymbol{U}^{-\top}$

Solving a Laplacian Linear Equation

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

Approximate Gaussian Elimination

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

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

$\boldsymbol{U}$ is sparse.

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

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 $\boldsymbol{U}$ with $O(m \log^3 n)$ non-zeros,
s.t. w.h.p.

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

Additive View of Gaussian Elimination

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

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

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

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

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.

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

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

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.

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

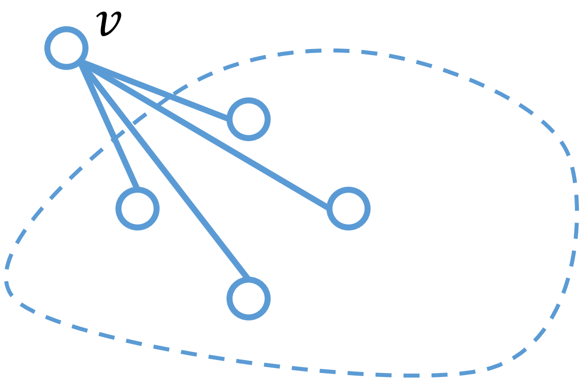
A new Laplacian!

Why is Gaussian Elimination Slow?

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

The main issue is **fill**

$\boldsymbol{L} =$

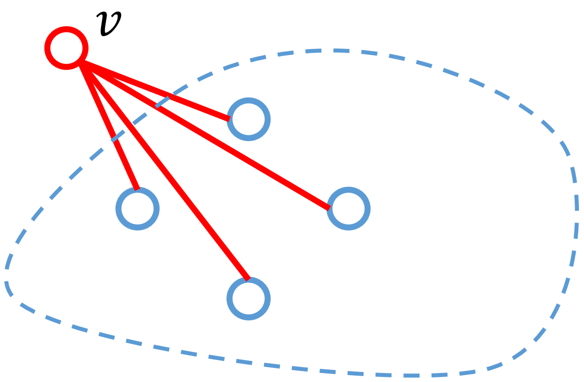


Why is Gaussian Elimination Slow?

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

The main issue is **fill**

$\boldsymbol{L} =$

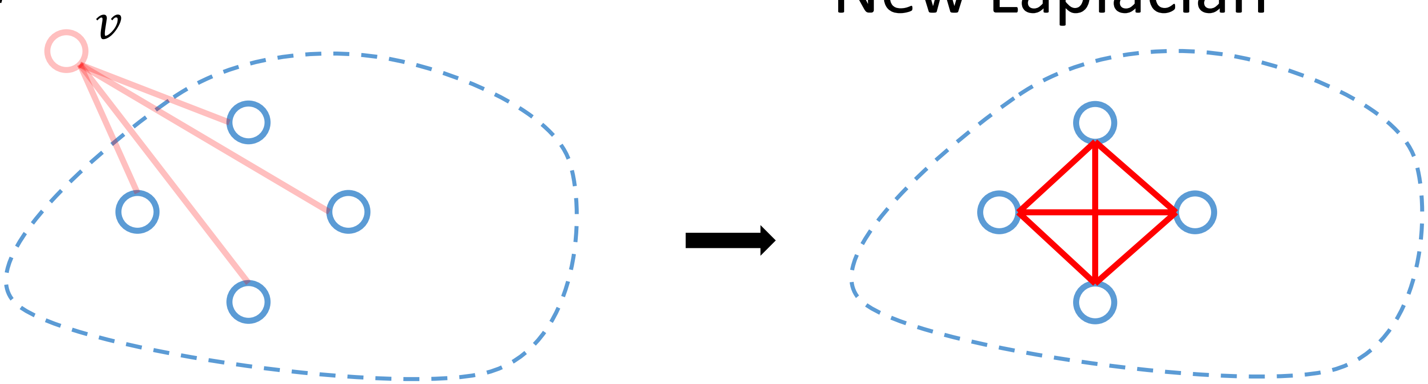


Why is Gaussian Elimination Slow?

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

The main issue is **fill**

$\boldsymbol{L} =$ New Laplacian



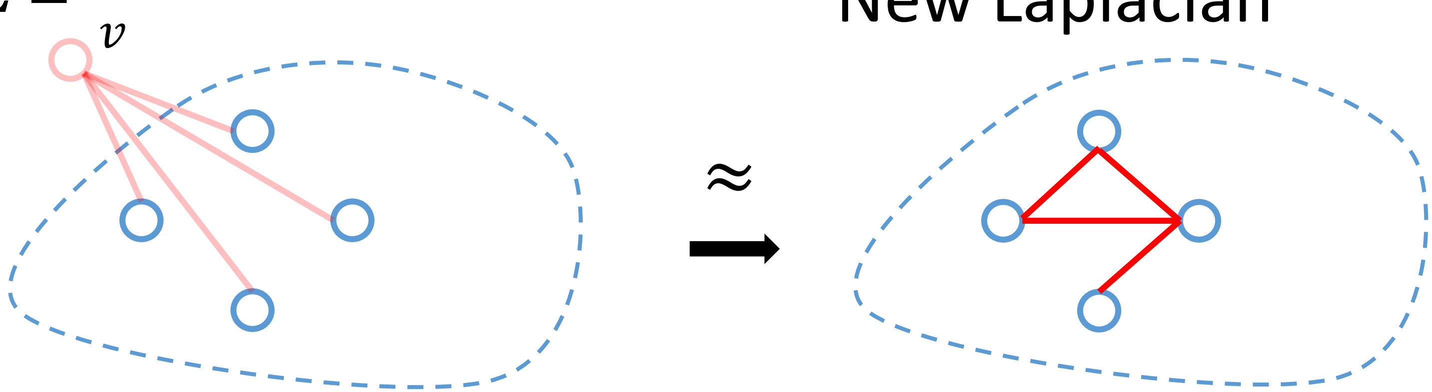
Elimination creates a clique
on the neighbors of v

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

Approximating Matrices by Sampling

Goal

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

Approach

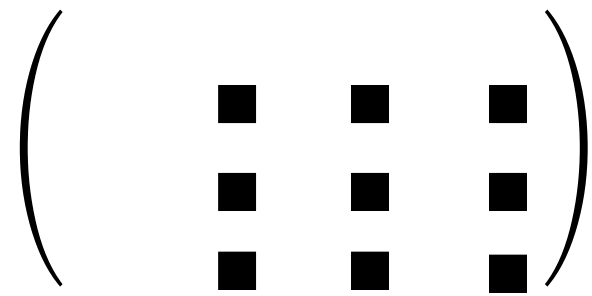
1. $\mathbb{E}\, \boldsymbol{U}^\top \boldsymbol{U} = \boldsymbol{L}$
2. Show $\boldsymbol{U}^\top \boldsymbol{U}$ concentrated around expectation

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

Approximating Matrices in Expectation

Consider eliminating the first variable

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



Original Laplacian

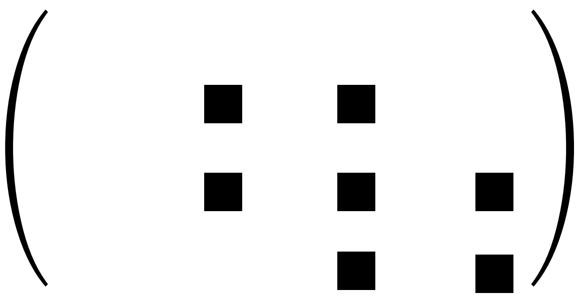
Rank 1 term

Remaining graph + clique

Approximating Matrices in Expectation

Consider eliminating the first variable

$$= \left(\boldsymbol{c}_1\right)\left(\boldsymbol{c}_1\right)^{\top} +$$



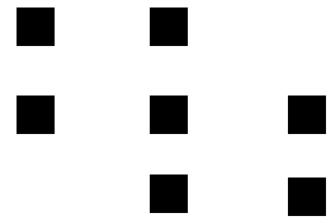
Rank 1 term

Remaining graph + sparsified clique

Approximating Matrices in Expectation

Consider eliminating the first variable

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



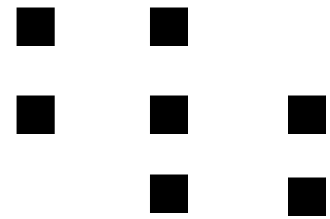
Rank 1 term

Remaining graph + sparsified clique

Approximating Matrices in Expectation

Consider eliminating the first variable

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



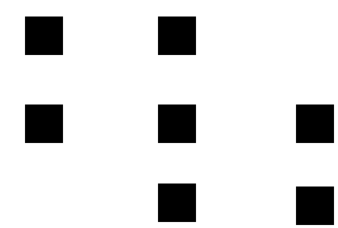
Rank 1 term

Remaining graph + sparsified clique

Approximating Matrices in Expectation

Consider eliminating the first variable

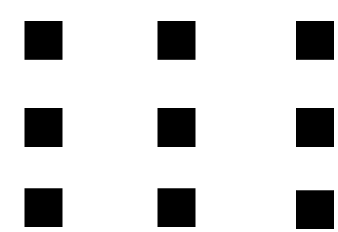
$$\mathbb{E}\left(\boldsymbol{L}^{(1)} \right) = \left(\boldsymbol{c}_1\right)\left(\boldsymbol{c}_1\right)^{\top} + \mathbb{E}\left(\quad \right)$$



Rank 1 term

Remaining graph + sparsified clique

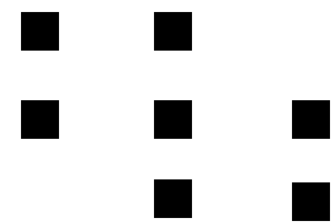
Suppose $= \left(\boldsymbol{c}_1\right)\left(\boldsymbol{c}_1\right)^{\top} + \left(\quad \right)$



Approximating Matrices in Expectation

Consider eliminating the first variable

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



Rank 1 term

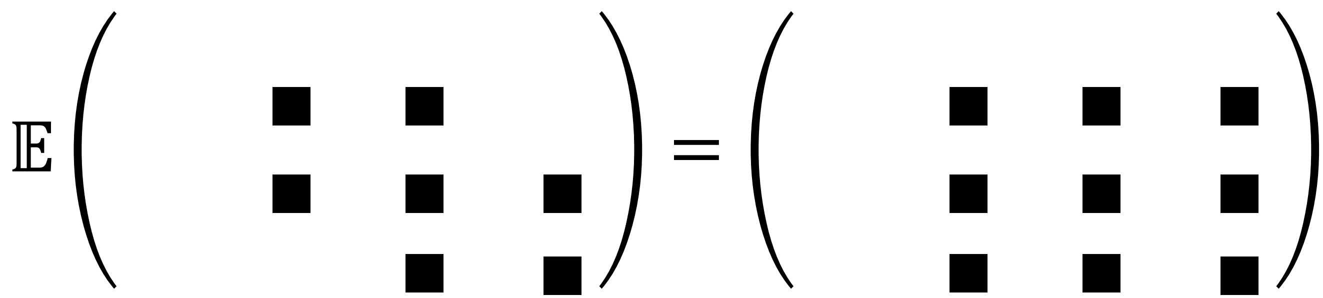
Remaining graph + sparsified clique

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

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

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

Approximation?

Approximate Gaussian Elimination

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

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

$$\sout{\left\| \boldsymbol{U}^\top \boldsymbol{U} - \boldsymbol{L} \right\| \le 0.5}$$

$$\left\| \boldsymbol{L}^{-1/2} \boldsymbol{U}^\top \boldsymbol{U} \boldsymbol{L}^{-1/2} - \boldsymbol{I} \right\| \le 0.5$$

Essential Tools

Isotropic position

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

Goal is now

$$\overline{\boldsymbol{U}^{\top}\boldsymbol{U}} - \boldsymbol{I} \approx \boldsymbol{0}$$

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 Concentration: Edge Variables

$$\boldsymbol{Y}_e = \begin{cases} \frac{1}{p_e}\boldsymbol{L_e} & \text{w. probability } p_e \\ \boldsymbol{0} & \text{o.w.} \end{cases}$$

Zero-mean variables $\boldsymbol{X}_e = \boldsymbol{Y}_e - \boldsymbol{L}_e$

Isotropic position variables $\overline{\boldsymbol{X}}_e$

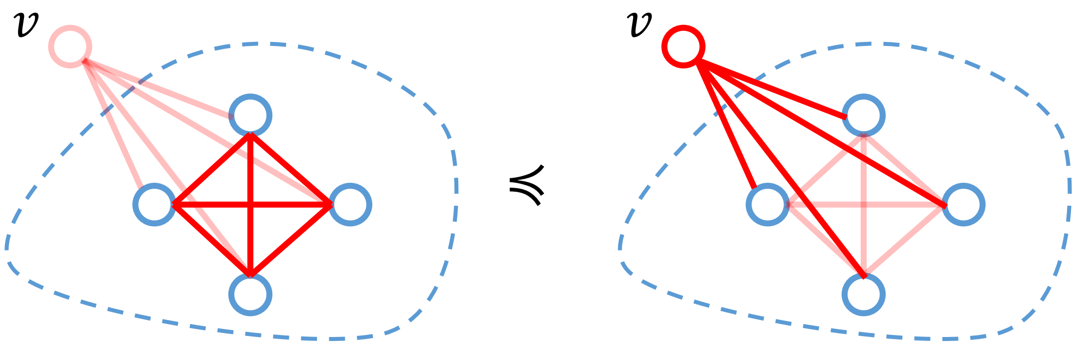
Predictable Quadratic Variation

Predictable quadratic variation $= \sum_i \mathbb{E}[\boldsymbol{X}_i^2 \mid \text{prev. steps}]$

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

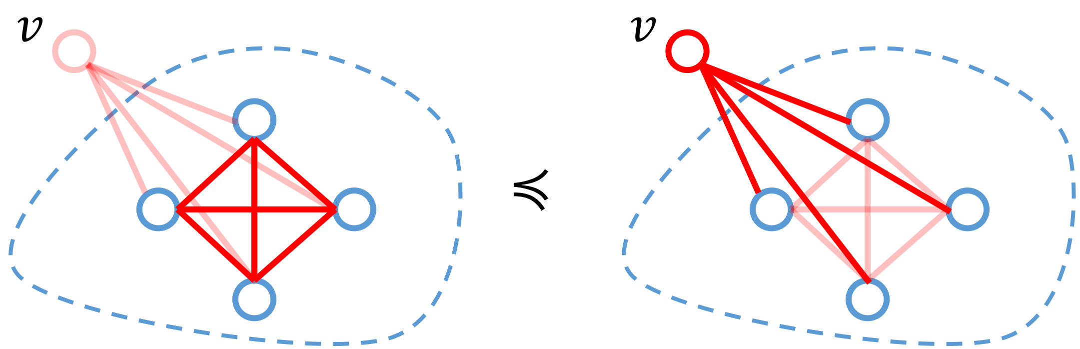
Promise: $\mathbb{E}\overline{\boldsymbol{X}}_e^2 \preccurlyeq r\overline{\boldsymbol{L}}_e$

Sample Variance



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

Sample Variance



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

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

WARNING: only true w.h.p.

Sample Variance

Recall $\mathbb{E}\overline{\boldsymbol{X}}_e^2 \preccurlyeq r\overline{\boldsymbol{L}}_e$

Putting it together

$$\mathbb{E}_v \sum_{e \in \text{ elim. clique of } v} \overline{\boldsymbol{L}}_e \preccurlyeq \frac{4\boldsymbol{I}}{\#\text{ vertices}}$$

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

variance
in one round of elimination

Sample Variance

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

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

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

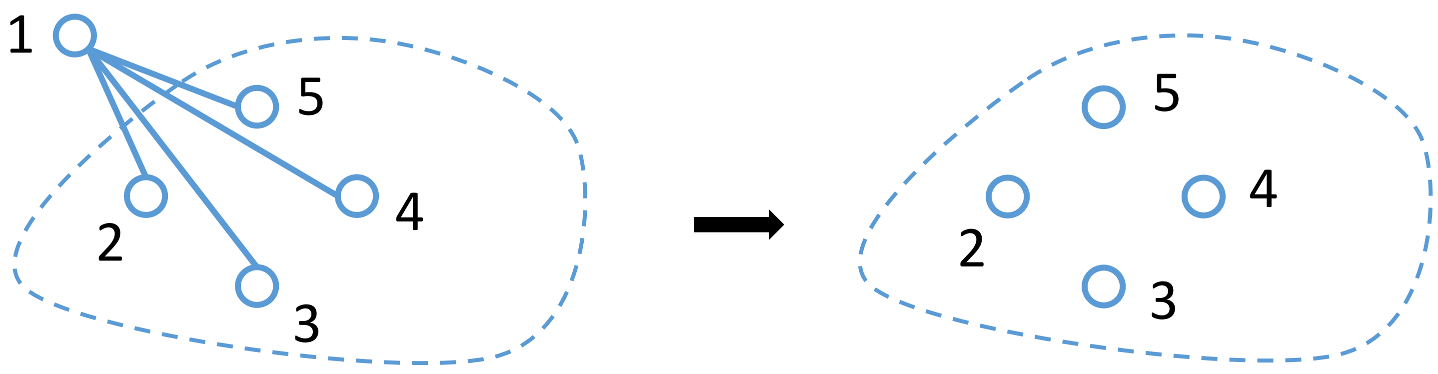
Thanks!

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 $\frac{w_u w_v}{w_u + w_v}$

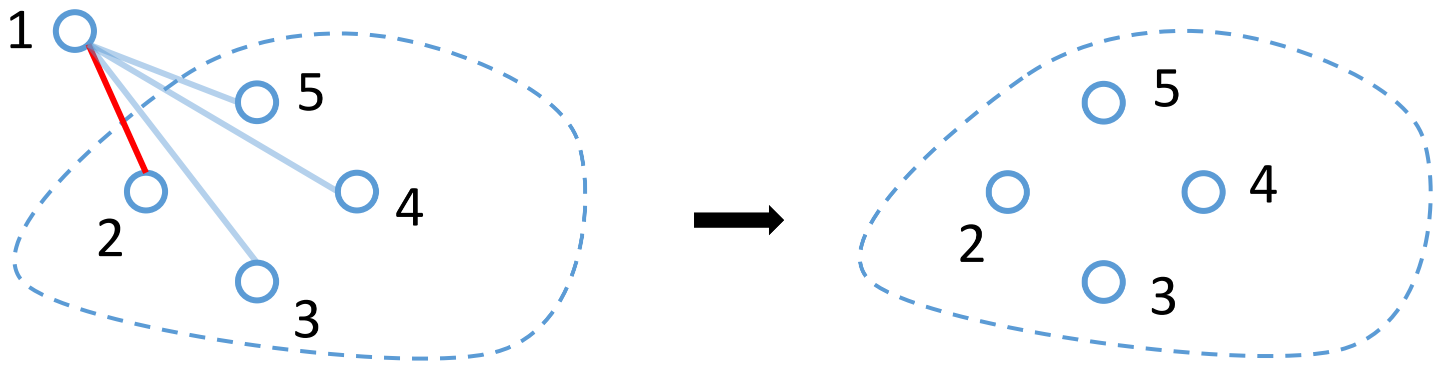


How to Sample a Clique

For each edge $(1, v)$

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

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

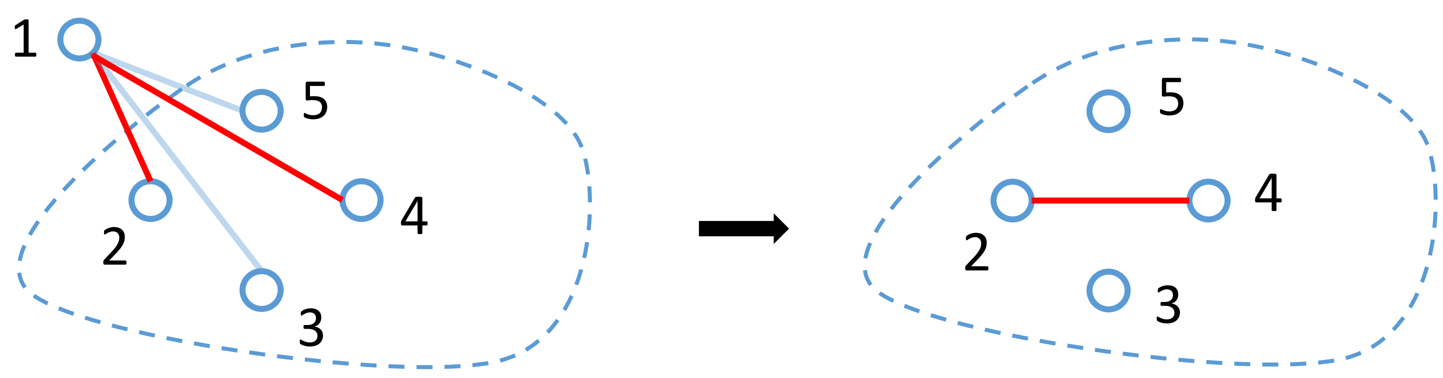


How to Sample a Clique

For each edge $(1, v)$

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

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

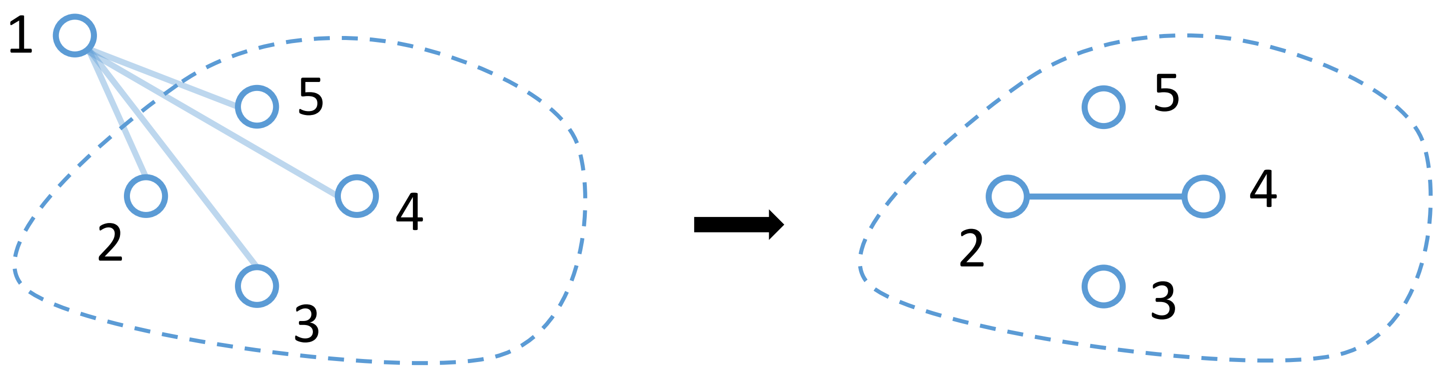


How to Sample a Clique

For each edge $(1, v)$

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

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

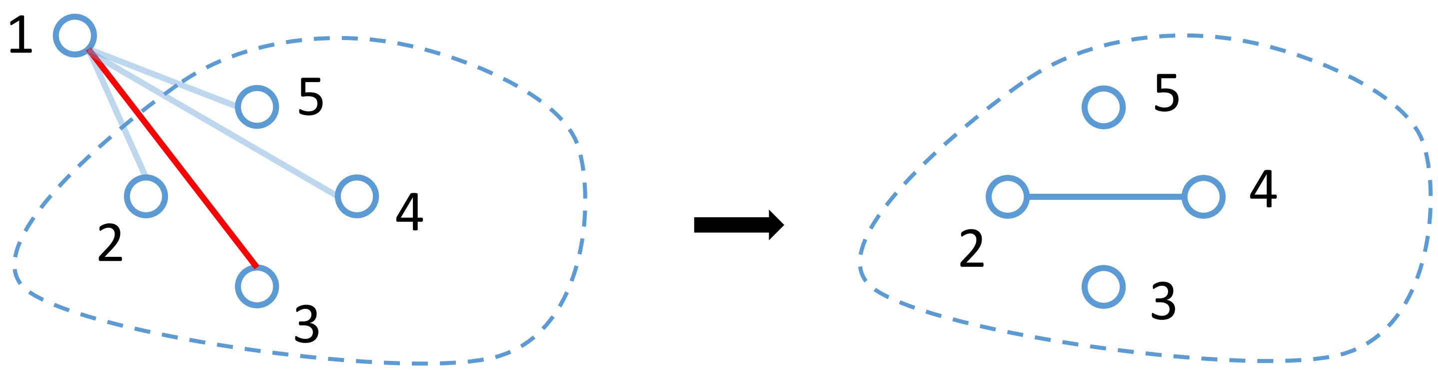


How to Sample a Clique

For each edge $(1, v)$

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

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

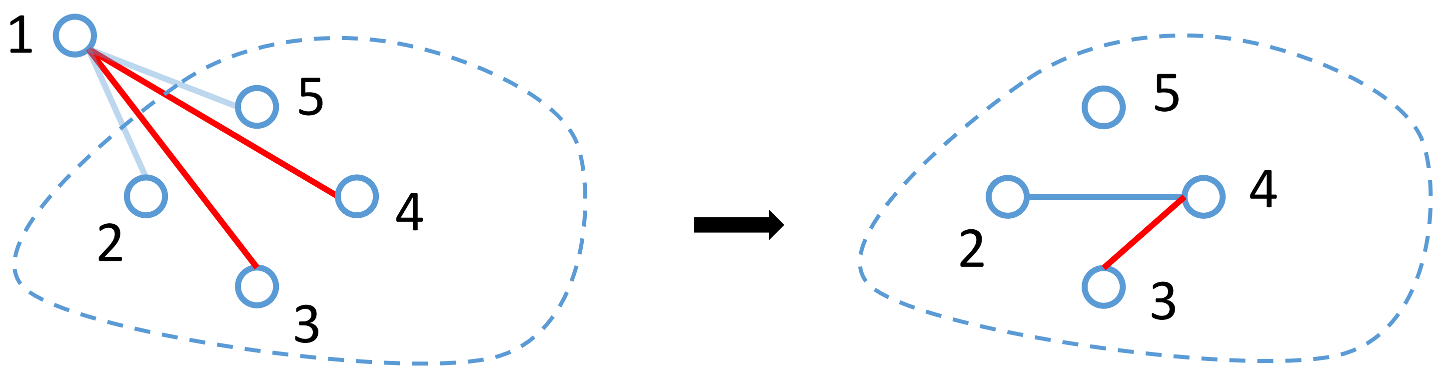


How to Sample a Clique

For each edge $(1, v)$

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

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

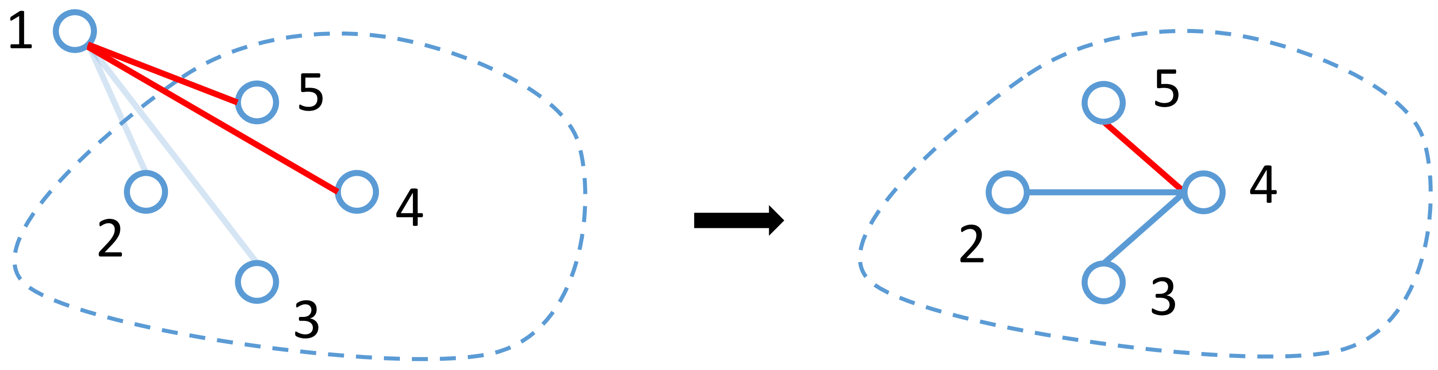


How to Sample a Clique

For each edge $(1, v)$

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

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

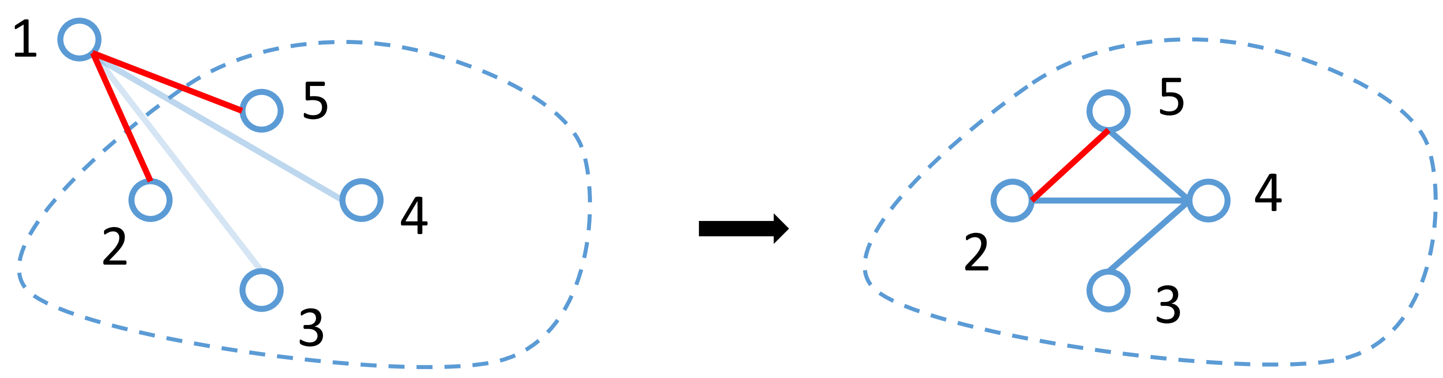


How to Sample a Clique

For each edge $(1, v)$

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

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

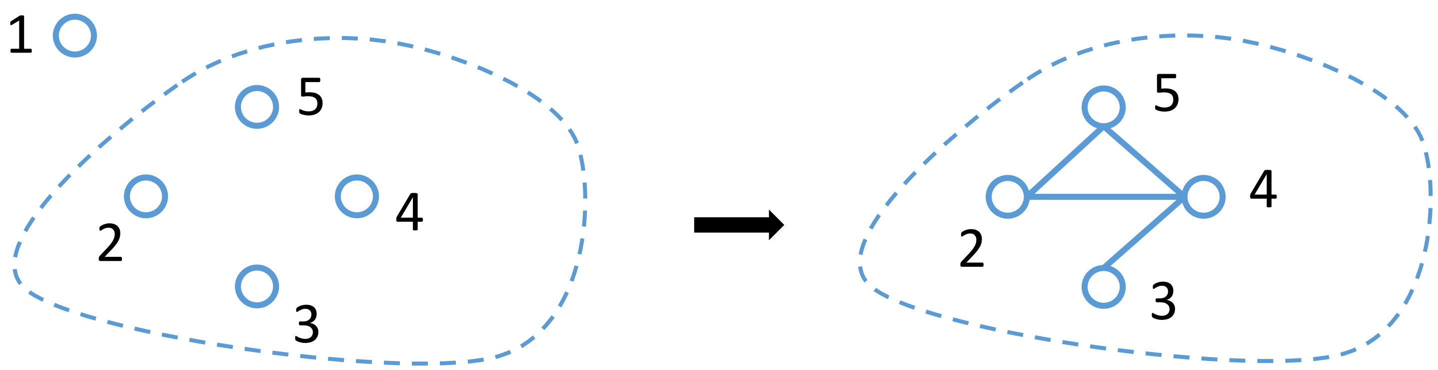


How to Sample a Clique

For each edge $(1, v)$

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

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



Practice

Julia Package: Laplacians.jl

tiny.cc/spielman-solver-code

Theory

Nearly-linear time Directed Laplacian solvers using approximate Gaussian elimination [CK**K**PPSR17]

This is really the end