Approximate Gaussian Elimination and Applications

Speaker **Sushant Sachdeva**

UToronto

Laplacian Paradigm 2.0

FOCS 2018

Gaussian Elimination

Gaussian Elimination

$$
\begin{array}{rrrrcr}
16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\
-4x_1 & +5x_2 & & -x_4 & = & -6 \\
-8x_1 & & +9x_3 & -x_4 & = & -19 \\
-4x_1 & -x_2 & -x_3 & +7x_4 & = & 11
\end{array}
$$

Gaussian Elimination

Add $^1/_4$ x Eq. 1 to Eq. 2

$$
\begin{array}{rrrrcr}
16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\
-4x_1 & +5x_2 & & -x_4 & = & -6 \\
-8x_1 & & +9x_3 & -x_4 & = & -19 \\
-4x_1 & -x_2 & -x_3 & +7x_4 & = & 11
\end{array}
$$

Gaussian Elimination

$$
\begin{array}{rrrrcr}
16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\
 & +4x_2 & -2x_3 & -2x_4 & = & -2 \\
-8x_1 & & +9x_3 & -x_4 & = & -19 \\
-4x_1 & -x_2 & -x_3 & +7x_4 & = & 11
\end{array}
$$

Gaussian Elimination

$$
\begin{array}{rrrrcr}
16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\
 & +4x_2 & -2x_3 & -2x_4 & = & -2 \\
 & -2x_2 & +5x_3 & -3x_4 & = & -11 \\
-4x_1 & -x_2 & -x_3 & +7x_4 & = & 11
\end{array}
$$

Gaussian Elimination

Eliminated the first variable

$$
\begin{array}{rrrrr}
16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\
 & +4x_2 & -2x_3 & -2x_4 & = & -2 \\
 & -2x_2 & +5x_3 & -3x_4 & = & -11 \\
 & -2x_2 & -3x_3 & +6x_4 & = & 15
\end{array}
$$

Gaussian Elimination

$$
\begin{array}{rrrrcr}
16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\
-4x_1 & +5x_2 & & -x_4 & = & -6 \\
-8x_1 & & +9x_3 & -x_4 & = & -19 \\
-4x_1 & -x_2 & -x_3 & +7x_4 & = & 11
\end{array}
$$

$$
\begin{array}{rrrrcr}
16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\
 & +4x_2 & -2x_3 & -2x_4 & = & -2 \\
 & -2x_2 & +5x_3 & -3x_4 & = & -11 \\
 & -2x_2 & -3x_3 & +6x_4 & = & 15
\end{array}
$$

Gaussian Elimination

Consider the coefficient matrix

$$\begin{array}{rrrrcr} 16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\ -4x_1 & +5x_2 & & -x_4 & = & -6 \\ -8x_1 & & +9x_3 & -x_4 & = & -19 \\ -4x_1 & -x_2 & -x_3 & +7x_4 & = & 11 \end{array}$$

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix}$$

$$\begin{array}{rrrrcr} 16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\ & +4x_2 & -2x_3 & -2x_4 & = & -2 \\ & -2x_2 & +5x_3 & -3x_4 & = & -11 \\ & -2x_2 & -3x_3 & +6x_4 & = & 15 \end{array}$$

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 5 \end{bmatrix}$$

Gaussian Elimination

Consider the coefficient matrix

$$\begin{array}{rrrrcr} 16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\ -4x_1 & +5x_2 & & -x_4 & = & -6 \\ -8x_1 & & +9x_3 & -x_4 & = & -19 \\ -4x_1 & -x_2 & -x_3 & +7x_4 & = & 11 \end{array} \qquad \begin{array}{rrrrcr} 16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\ & +4x_2 & -2x_3 & -2x_4 & = & -2 \\ & -2x_2 & +5x_3 & -3x_4 & = & -11 \\ & -2x_2 & -3x_3 & +6x_4 & = & 15 \end{array}$$

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{4} \\ -\frac{1}{2} \\ -\frac{1}{4} \end{bmatrix} \underbrace{\begin{bmatrix} 16 & -4 & -8 & -4 \end{bmatrix}}_{\text{Copies of row 1}} + \begin{bmatrix} 16 & -4 & -8 & -4 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 5 \end{bmatrix}$$

Gaussian Elimination

Consider the coefficient matrix

$$\begin{array}{rrrrcr} 16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\ -4x_1 & +5x_2 & & -x_4 & = & -6 \\ -8x_1 & & +9x_3 & -x_4 & = & -19 \\ -4x_1 & -x_2 & -x_3 & +7x_4 & = & 11 \end{array} \qquad \begin{array}{rrrrcr} 16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\ & +4x_2 & -2x_3 & -2x_4 & = & -2 \\ & -2x_2 & +5x_3 & -3x_4 & = & -11 \\ & -2x_2 & -3x_3 & +6x_4 & = & 15 \end{array}$$

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{1}{4} \\ -\frac{1}{2} \\ -\frac{1}{4} \end{bmatrix} \underbrace{\begin{bmatrix} 16 & -4 & -8 & -4 \end{bmatrix}}_{\text{Copies of row 1}} + \begin{bmatrix} 16 & -4 & -8 & -4 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 6 \end{bmatrix}$$

Gaussian Elimination

Consider the coefficient matrix

$$\begin{array}{rrrrcr} 16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\ -4x_1 & +5x_2 & & -x_4 & = & -6 \\ -8x_1 & & +9x_3 & -x_4 & = & -19 \\ -4x_1 & -x_2 & -x_3 & +7x_4 & = & 11 \end{array} \qquad \begin{array}{rrrrcr} 16x_1 & -4x_2 & -8x_3 & -4x_4 & = & 16 \\ & +4x_2 & -2x_3 & -2x_4 & = & -2 \\ & -2x_2 & +5x_3 & -3x_4 & = & -11 \\ & -2x_2 & -3x_3 & +6x_4 & = & 15 \end{array}$$

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{1}{4} \\ -\frac{1}{2} \\ -\frac{1}{4} \end{bmatrix} \underbrace{\begin{bmatrix} 16 & -4 & -8 & -4 \end{bmatrix}}_{\text{Copies of row 1}} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 6 \end{bmatrix}$$

Gaussian Elimination

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{1}{4} \\ -\frac{1}{2} \\ -\frac{1}{4} \end{bmatrix} \begin{bmatrix} 16 & -4 & -8 & -4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 6 \end{bmatrix}$$

Symmetric

Gaussian Elimination

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{1}{4} \\ -\frac{1}{2} \\ -\frac{1}{4} \end{bmatrix} \begin{bmatrix} 16 & -4 & -8 & -4 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 6 \end{bmatrix}$$

Symmetric

Symmetric rank 1

Cholesky Factorization

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 6 \end{bmatrix}$$

Symmetric

Symmetric rank 1

Cholesky Factorization

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 6 \end{bmatrix}$$

Symmetric

Symmetric rank 1

Schur Complement

Cholesky Factorization

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}}$$

PSD

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 6 \end{bmatrix}$$

The Schur complement is PSD.

Cholesky Factorization

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}}$$

$$+ \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 6 \end{bmatrix}$$

Cholesky Factorization

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 \\ 2 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ -1 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 4 & -4 \\ 0 & 0 & -4 & 5 \end{bmatrix}$$

Cholesky Factorization

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 \\ 2 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ -1 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 \\ 0 \\ 2 \\ -2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -2 \\ -2 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Cholesky Factorization

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 \\ 2 \\ -1 \\ -1 \end{bmatrix} \begin{bmatrix} 0 \\ 2 \\ -1 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 \\ 0 \\ 2 \\ -2 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ -2 \\ -2 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}}$$

Cholesky Factorization

$$\begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 7 \end{bmatrix} = \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}\begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 \\ 2 \\ -1 \\ -1 \end{bmatrix}\begin{bmatrix} 0 \\ 2 \\ -1 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 \\ 0 \\ 2 \\ -2 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ -2 \\ -2 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}^{\mathsf{T}}$$

$$= \begin{bmatrix} 4 & -1 & -2 & -1 \\ 0 & 2 & -1 & -1 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}^{\mathsf{T}} \begin{bmatrix} 4 & -1 & -2 & -1 \\ 0 & 2 & -1 & -1 \\ 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$= U^{\mathsf{T}}U$ $\quad$ U is upper triangular

Solving $Lx = b$

Cholesky Factorization

Find U, upper triangular matrix, s.t.

$$U^\top U = L$$

Then solve

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

Both inverses easy to apply!

Cholesky Factorization on Laplacians

$$L = \begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 5 \end{bmatrix}$$

Cholesky Factorization on Laplacians

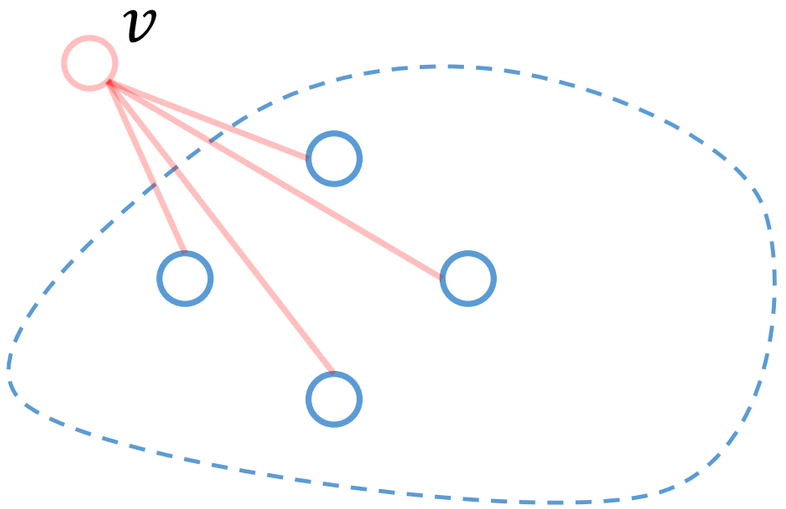
$$L = \begin{bmatrix} 16 & -4 & -8 & -4 \\ -4 & 5 & 0 & -1 \\ -8 & 0 & 9 & -1 \\ -4 & -1 & -1 & 6 \end{bmatrix}$$

$$= \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix} \begin{bmatrix} 4 \\ -1 \\ -2 \\ -1 \end{bmatrix}^{\mathsf{T}} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 4 & -2 & -2 \\ 0 & -2 & 5 & -3 \\ 0 & -2 & -3 & 5 \end{bmatrix}$$

Another Laplacian!

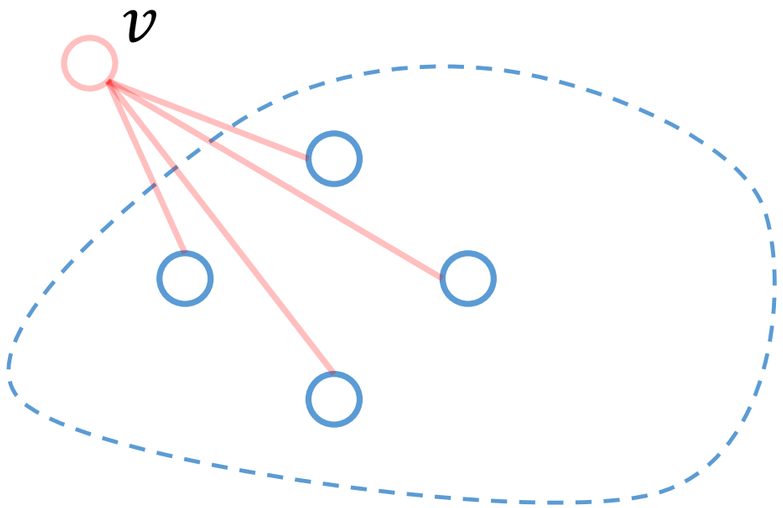
Cholesky Factorization on Laplacians

$L =$

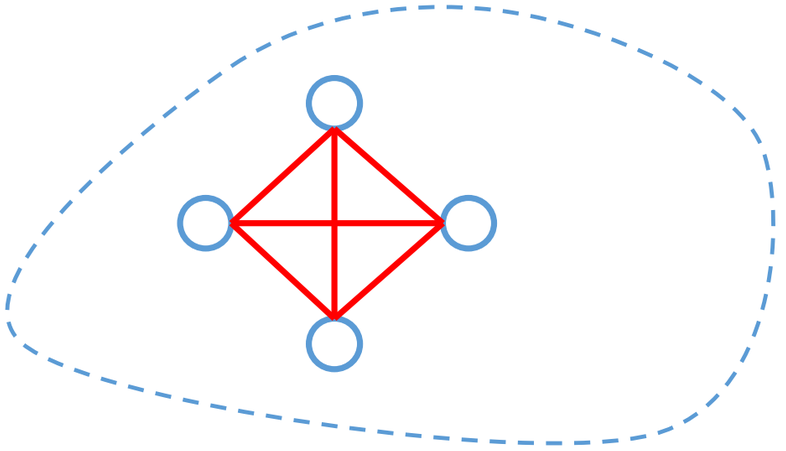


Cholesky Factorization on Laplacians

$L =$

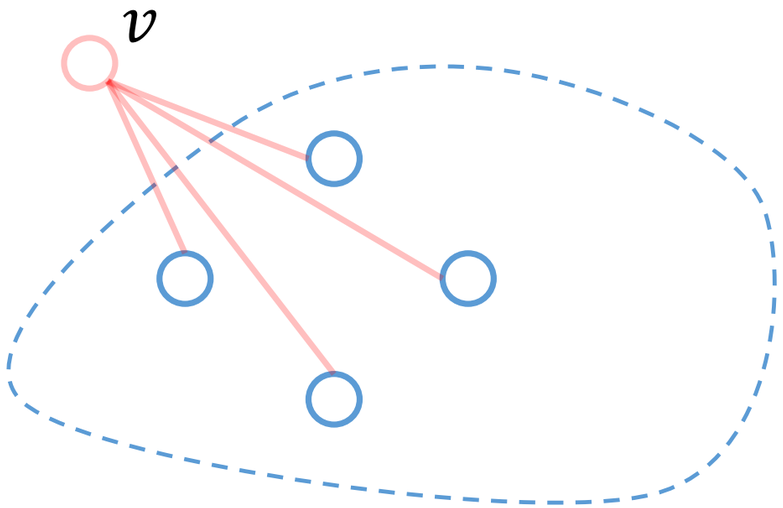


Schur complement $=$

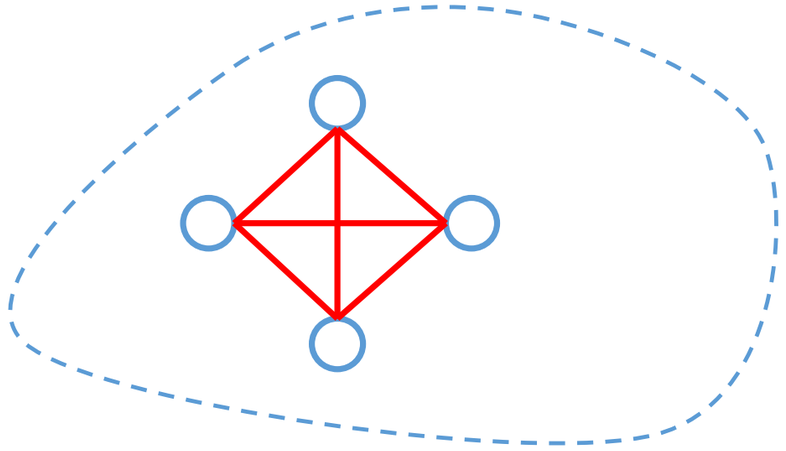


Cholesky Factorization on Laplacians

$L =$



Schur complement $=$

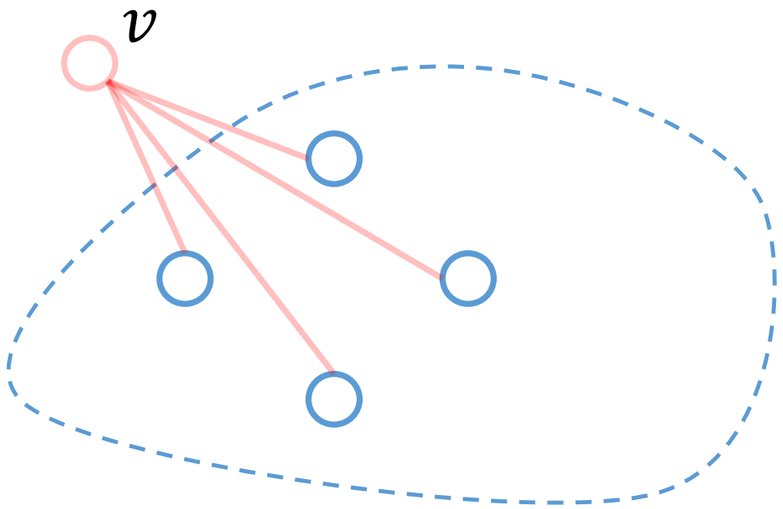


Elimination creates a clique on the neighbors of v

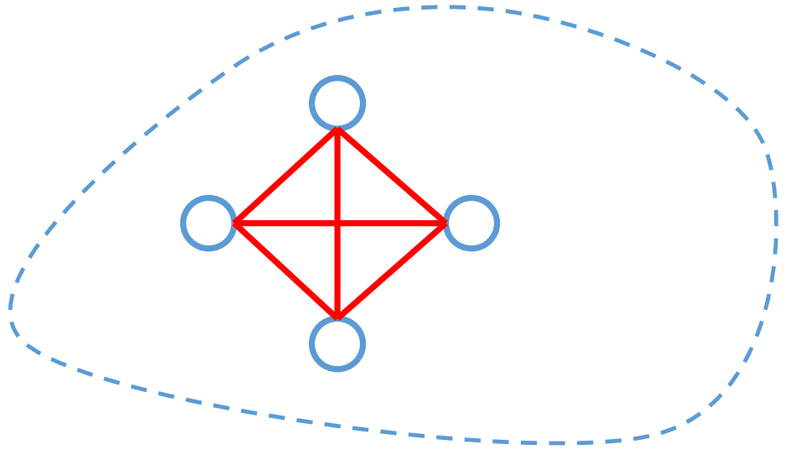
Cholesky Factorization on Laplacians

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

$L =$



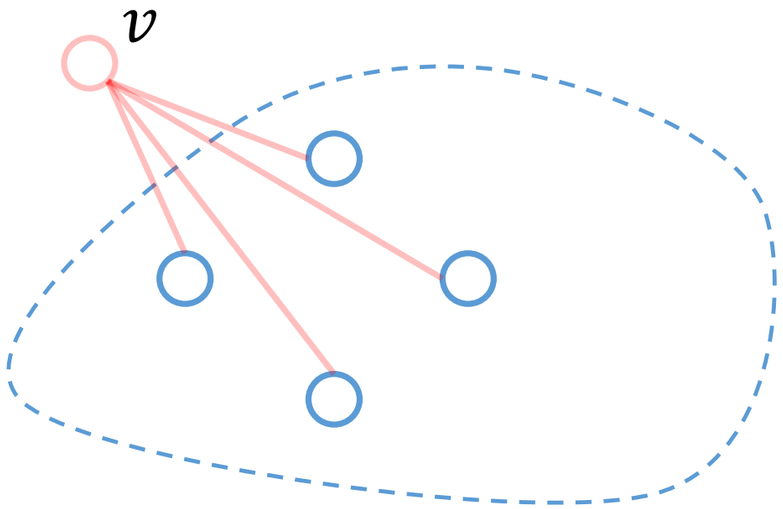
Schur complement $=$



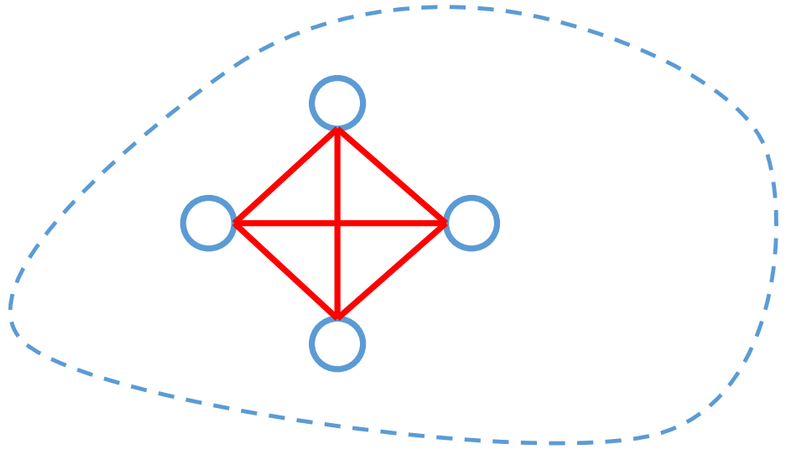
Elimination creates a clique on the neighbors of v

Why is Cholesky Factorization Slow?

$L =$



Schur complement $=$

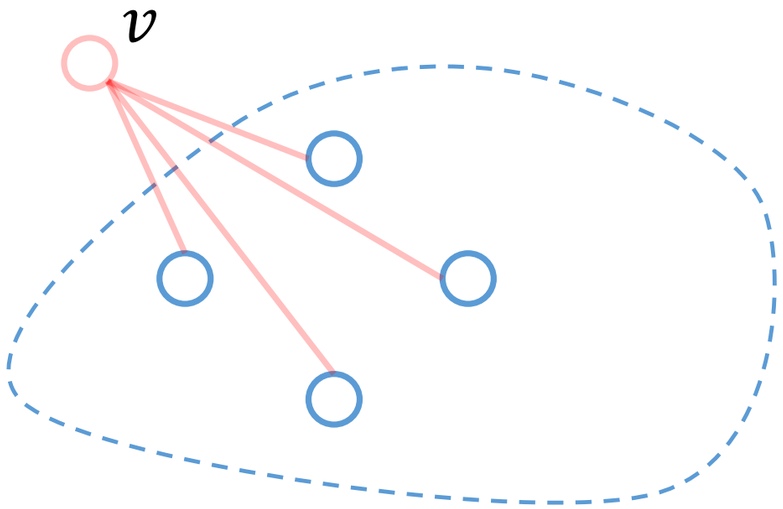


Elimination creates a clique on the neighbors of v

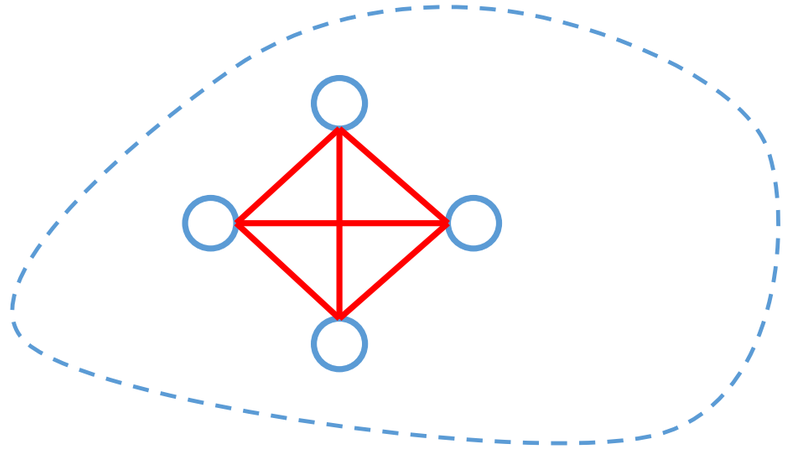
Why is Cholesky Factorization Slow?

The main issue is **fill.** Remove $\deg(v)$ edges, insert $\sim\deg(v)^2$ edges.

$L =$



Schur complement $=$



Elimination creates a clique on the neighbors of v

Why is Cholesky Factorization Slow?

Eliminate n vertices $\times$ add clique:

$$O\left(\sum_{v} \deg(v)^2\right) = O(n^3)$$

Why is Cholesky Factorization Slow?

Eliminate n vertices $\times$ add clique:

$$O\left(\sum_{v} \deg(v)^2\right) = O(n^3)$$

Cholesky Factorization on expanders takes $\Omega(n^3)$ time.

Why is Cholesky Factorization Slow?

Eliminate n vertices $\times$ add clique:

$$O\left(\sum_{v} \deg(v)^2\right) = O(n^3)$$

Cholesky Factorization on expanders takes $\Omega(n^3)$ time.

Central challenge in scientific computing. Numerous heuristic approaches to sparse Cholesky factorization, back to [Varga '60]

Why is Cholesky Factorization Slow?

Eliminate n vertices $\times$ add clique:

$$O\left(\sum_v \deg(v)^2\right) = O(n^3)$$

Cholesky Factorization on expanders takes $\Omega(n^3)$ time.

Central challenge in scientific computing. Numerous heuristic approaches to sparse Cholesky factorization, back to [Varga '60].

None provide guarantees for general Laplacians.

One-by-one Laplacian solver

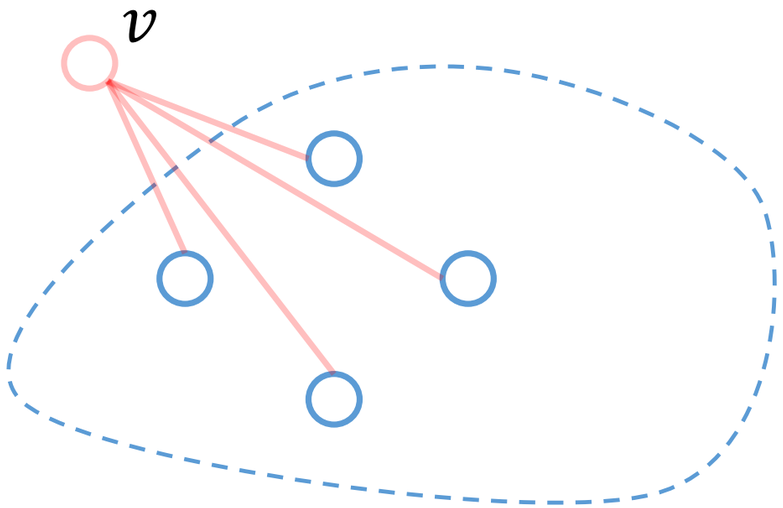
Fast Approximate Cholesky Factorization [Kyng-S ‘16]

Building on [Lee-Peng-Spielman ‘15]

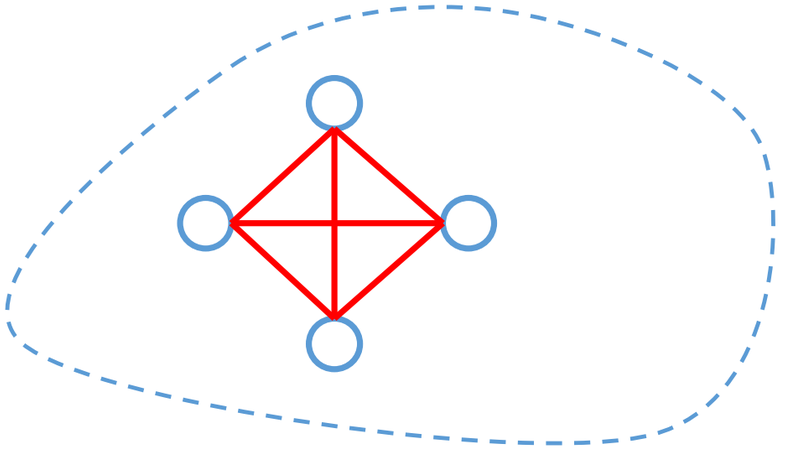
Inspiration for the Paper

Elimination creates a clique on the neighbors of v

$L =$



Schur complement $=$

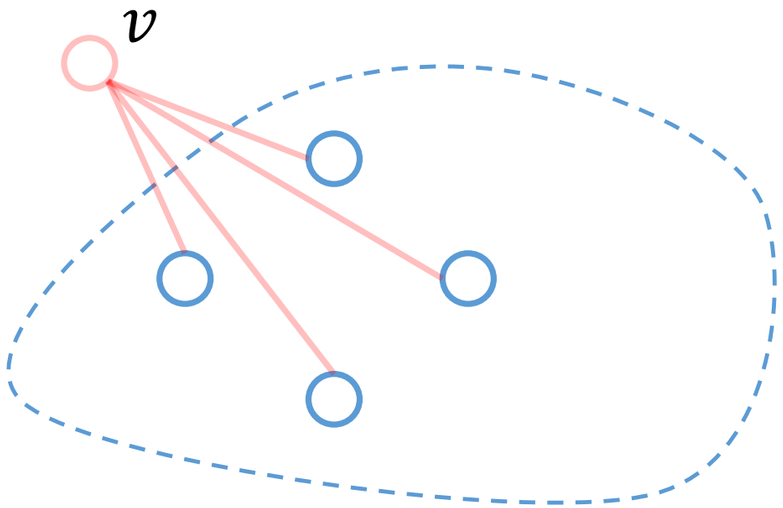


Inspiration for the Paper

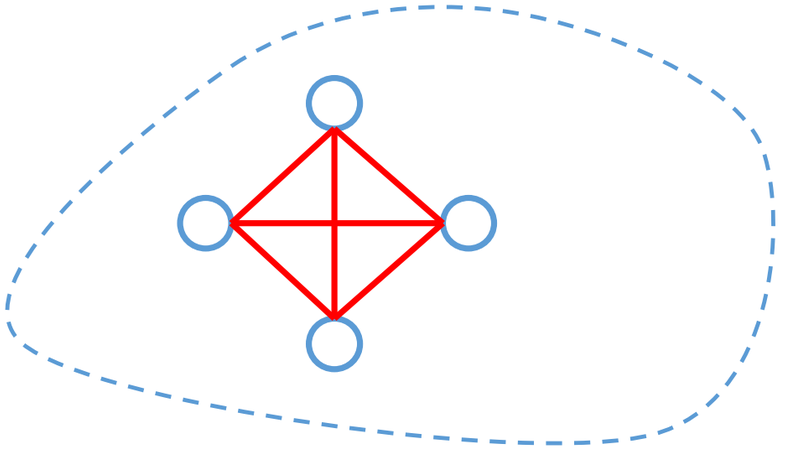
Elimination creates a clique on the neighbors of v

Cliques can be sparsified!

$L =$



Schur complement $=$

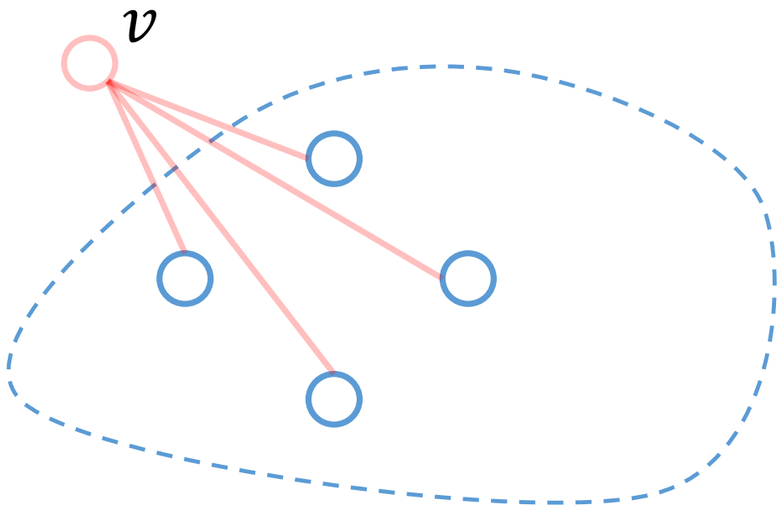


Inspiration for the Paper

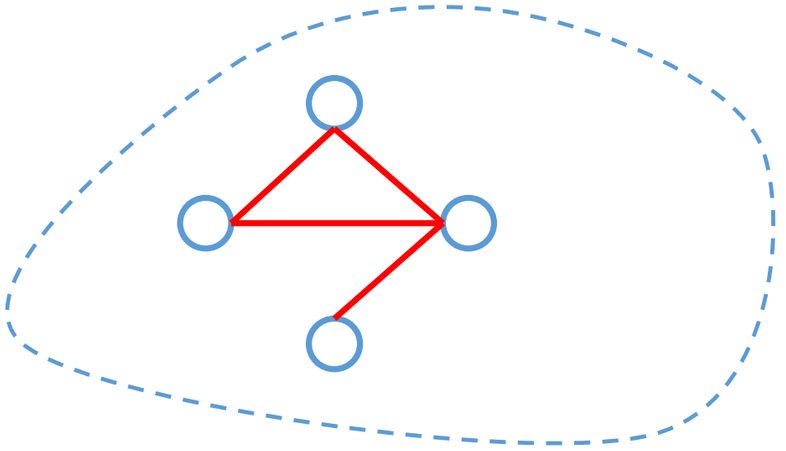
Elimination creates a clique on the neighbors of v

Cliques can be sparsified!

$L =$



Schur complement $\approx_\delta$

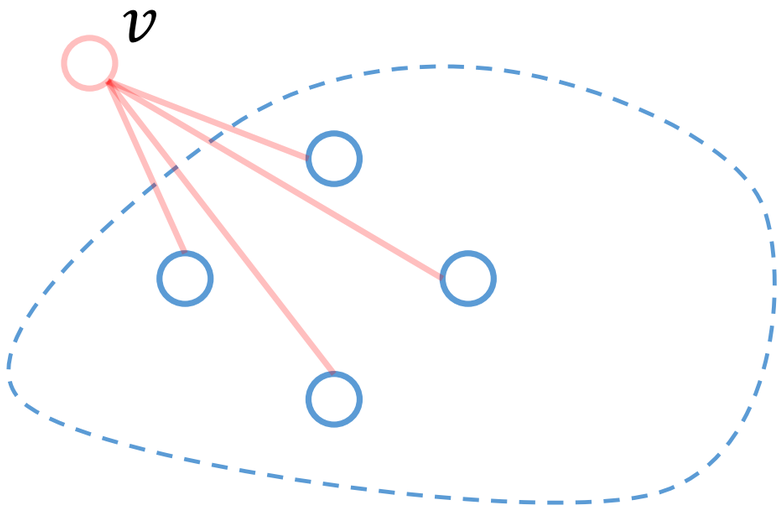


Inspiration for the Paper

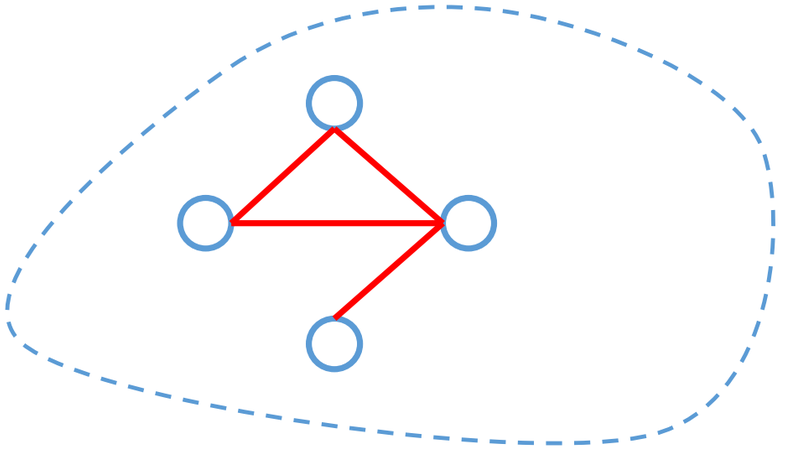
Elimination creates a clique on the neighbors of v

Cliques can be sparsified! Replacing with an expander not enough.

$L =$



Schur complement $\approx_\delta$



Sparse Cholesky Factorization

[Kyng-S'17]

1. Pick a random vertex v to eliminate
2. Sample $\deg(v)$ edges from the clique created by eliminating v
3. Repeat

Sparse Cholesky Factorization

[Kyng-S'17]

1. Pick a random vertex v to eliminate
2. Sample $\deg(v)$ edges from the clique created by eliminating v
3. Repeat

Most heuristic algorithms have a similar flavor.

e.g. Incomplete Cholesky Factorization (ICC)

Key difference: which edges to keep

What makes [Kyng-S'17] work?

First provably approximate sparse Cholesky factorization we're aware of

What makes [Kyng-S'17] work?

First provably approximate sparse Cholesky factorization we're aware of

1. Maintains a PSD matrix as approximation with same kernel as L

What makes [Kyng-S'17] work?

First provably approximate sparse Cholesky factorization we're aware of

1. Maintains a PSD matrix as approximation with same kernel as L
2. At each step, correct expectation: $\mathbb{E}\, U^\top U = L$. Analysis using matrix martingales

What makes [Kyng-S'17] work?

First provably approximate sparse Cholesky factorization we're aware of

1. Maintains a PSD matrix as approximation with same kernel as L
2. At each step, correct expectation: $\mathbb{E}\, U^\top U = L$. Analysis using matrix martingales
3. Number of edges in remaining graph does not increase

Running Time

[Kyng-S'17] **Sparsified Cholesky Factorization**

Eliminate n vertices $\times$ add sparsified clique:

$$\tilde{O}\left(\sum_{v} \deg(v)\right) = \tilde{O}(m)$$

Sparse, Approximate Cholesky Factorization

[Kyng-S ‘16] For any Laplacian L, in time $O(m \log^3 n)$, can compute an upper triangular U s.t.

$$U^\top U \approx_{0.1} L$$

and U has $O(m \log^3 n)$ entries.

Sparse, Approximate Cholesky Factorization

[Kyng-S '16] For any Laplacian L, in time $O(m \log^3 n)$, can compute an upper triangular U s.t.

$$U^\top U \approx_{0.1} L$$

and U has $O(m \log^3 n)$ entries.

Can find an approximate solution to $Lx = b$ in $O\left(m \log^3 n \log\left(\frac{1}{\delta}\right)\right)$ time.

Sparse, Approximate Cholesky Factorization

[Kyng-S '16] For any Laplacian L, in time $O(m \log^3 n)$, can compute an upper triangular U s.t.

$$U^\top U \approx_{0.1} L$$

and U has $O(m \log^3 n)$ entries.

Can find an approximate solution to $Lx = b$ in $O\left(m \log^3 n \log\left(\frac{1}{\delta}\right)\right)$ time.

[Lee-Peng-Spielman '15] had shown an existential result.

Sparse, Approximate Cholesky Factorization

[Kyng-S ‘16] For any Laplacian L, in time $O(m \log^3 n)$, can compute an upper triangular U s.t.

$$U^\top U \approx_{0.1} L$$

and U has $O(m \log^3 n)$ entries.

Can find an approximate solution to $Lx = b$ in $O\left(m$

[Lee-Peng-Spielman ‘15] had shown an existential re

Approximate Schur Complements

[Kyng-S ’16, Durfee-Kyng-Peebles-Rao-S ’17, Kyng ‘17]

Cholesky Factorization : Another view

$$L = \left[\begin{array}{c|c} a & b^{\mathsf{T}} \\ \hline b & C \end{array}\right] = \frac{1}{a}\begin{bmatrix} a \\ b \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix}^{\mathsf{T}} + \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & C - \frac{1}{a}bb^{\mathsf{T}} \end{array}\right]$$

Rank 1

Cholesky Factorization : Another view

$$L = \left[\begin{array}{c|c} a & b^{\mathsf{T}} \\ \hline b & C \end{array}\right] = \frac{1}{a}\begin{bmatrix} a \\ b \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix}^{\mathsf{T}} + \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & C - \frac{1}{a}bb^{\mathsf{T}} \end{array}\right]$$

Schur complement

Cholesky Factorization : Another view

$$L = \left[\begin{array}{c|c} a & b^{\mathsf{T}} \\ \hline b & C \end{array}\right] = \frac{1}{a}\begin{bmatrix} a \\ \hline b \end{bmatrix}\begin{bmatrix} a \\ \hline b \end{bmatrix}^{\mathsf{T}} + \begin{array}{c} \bar{F} \\ \\ \bar{F} \end{array}\left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & C - \frac{1}{a}bb^{\mathsf{T}} \end{array}\right]\begin{array}{c} F \\ \\ F \end{array}$$

Schur complement

Cholesky Factorization : Another view

$$L = \left[\begin{array}{c|c} a & b^{\top} \\ \hline b & C \end{array}\right] = \frac{1}{a}\begin{bmatrix} a \\ \hline b \end{bmatrix}\begin{bmatrix} a \\ \hline b \end{bmatrix}^{\top} + \begin{array}{c} \begin{matrix} \bar{F} & F \end{matrix} \\ \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & C - \frac{1}{a}bb^{\top} \end{array}\right] \end{array} \begin{matrix} \\ \bar{F} \\ F \end{matrix}$$

Schur complement $Sc(L, F)$

Cholesky Factorization : Another view

$$L = \left[\begin{array}{c|c} a & b^{\mathsf{T}} \\ \hline b & C \end{array}\right] = \frac{1}{a}\begin{bmatrix} a \\ b \end{bmatrix}\begin{bmatrix} a \\ b \end{bmatrix}^{\mathsf{T}} + \left[\begin{array}{c|c} 0 & 0 \\ \hline 0 & C - \frac{1}{a}bb^{\mathsf{T}} \end{array}\right]$$

Schur complement $Sc(L, F)$

$$= \left[\begin{array}{c|c} 1 & 0 \\ \hline \frac{1}{a}b & I \end{array}\right]\left[\begin{array}{c|c} a & 0 \\ \hline 0 & C - \frac{1}{a}bb^{\mathsf{T}} \end{array}\right]\left[\begin{array}{c|c} 1 & \frac{1}{a}b^{\mathsf{T}} \\ \hline 0 & I \end{array}\right]$$

Partial Cholesky Factorization

Can eliminate several vertices at once. The order is irrelevant.

$$L = \begin{array}{c} \\ \bar{F} \\ F \end{array} \overset{\begin{array}{cc} \bar{F} & F \end{array}}{\left[\begin{array}{c|c} A & B^{\mathsf{T}} \\ \hline B & C \end{array}\right]}$$

$$= \begin{bmatrix} I & 0 \\ \hline BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ \hline 0 & C - BA^{-1}B^{\mathsf{T}} \end{bmatrix} \begin{bmatrix} I & A^{-1}B^{\mathsf{T}} \\ \hline 0 & I \end{bmatrix}$$

Approximate Schur Complement

[Kyng-S ’16]

For any Laplacian L, any set F, in time $O(m\varepsilon^{-2}\log^3 n)$, can compute a graph H on F such that

$$L_H \approx_\varepsilon Sc(L,F)$$

and H has $O(m\varepsilon^{-2}\log^3 n)$ edges.

Approximate Schur Complement

[Kyng-S ’16]

For any Laplacian L, any set F, in time $O(m\varepsilon^{-2}\log^3 n)$, can compute a graph H on F such that

$$L_H \approx_\varepsilon Sc(L,F)$$

and H has $O(m\varepsilon^{-2}\log^3 n)$ edges.

[Durfee-Kyng-Peebles-Rao-S ’17, Kyng ‘17]

Combining with sparsification improves sparsity of H to $O(|F|\varepsilon^{-2}\log n)$, and running time to $O((m+n\varepsilon^{-2})\log^3 n)$

Approximate Schur Complement

[Kyng-S ’16]

For any Laplacian L, any set F, in time $O(m\varepsilon^{-2}\log^3 n)$, can compute a graph H on F such that

$$L_H \approx_\varepsilon Sc(L,F)$$

and H has $O(m\varepsilon^{-2}\log^3 n)$ edges.

[Durfee-Kyng-Peebles-Rao-S ’17, Kyng ‘17]

Combining with sparsification improves sparsity of H to $O(|F|\varepsilon^{-2}\log n)$, and running time to $O((m+n\varepsilon^{-2})\log^3 n)$

Applications of Approximate Schur Complement

1. Faster estimation of effective resistances + Sample Random Spanning trees
   [Durfee-Kyng-Peebles-Rao-S ’17, Durfee-Peebles-Peng-Rao ’17]

   Schur complements preserve effective resistances

Applications of Approximate Schur Complement

1. Faster estimation of effective resistances + Sample Random Spanning trees
   [Durfee-Kyng-Peebles-Rao-S ’17, Durfee-Peebles-Peng-Rao ’17]

2. Estimate Determinants of Laplacians / number of spanning trees
   [Durfee-Peebles-Peng-Rao ’17]

Spectral Sketches for Schur Complement

[Chu-Gao-Peng-S-Sawlani-Wang ’18 (this FOCS)]
Spectral sketch of Schur complement (weaker than spectral approximation)
Better ε dependence, worse $n^{o(1)}$ factors, in size / running time

Builds on [Andoni-Chen-Krauthgamer-Qin-Woodruff-Zhang ‘16, Jambulapati-Sidford ‘18].

Spectral Sketches for Schur Complement

[Chu-Gao-Peng-S-Sawlani-Wang ’18 (this FOCS)]
Spectral sketch of Schur complement (weaker than spectral approximation)
Better ε dependence, worse $n^{o(1)}$ factors, in size / running time

Builds on [Andoni-Chen-Krauthgamer-Qin-Woodruff-Zhang ‘16, Jambulapati-Sidford ‘18].

Faster estimation of Laplacian determinant.

Recursive solver for Connection Laplacians

[Kyng-Lee-Peng-S-Spielman ‘16]

Exact solvers via Partial Cholesky Factorization

$$L = \left[\begin{array}{c|c} I & 0 \\ \hline BA^{-1} & I \end{array}\right] \left[\begin{array}{c|c} A & 0 \\ \hline 0 & S \end{array}\right] \left[\begin{array}{c|c} I & A^{-1}B^{\top} \\ \hline 0 & I \end{array}\right]$$

$$S = C - BA^{-1}B^{\top}$$

Schur complement $Sc(L, F)$

Exact solvers via Partial Cholesky Factorization

$$L = \left[\begin{array}{c|c} I & 0 \\ \hline BA^{-1} & I \end{array}\right] \left[\begin{array}{c|c} A & 0 \\ \hline 0 & S \end{array}\right] \left[\begin{array}{c|c} I & A^{-1}B^{\top} \\ \hline 0 & I \end{array}\right]$$

Easy to invert $\quad S = C - BA^{-1}B^{\top}$ Schur complement $Sc(L, F)$ $\quad$ Easy to invert

Exact solvers via Partial Cholesky Factorization

$$L = \begin{bmatrix} I & 0 \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & 0 \\ 0 & S \end{bmatrix} \begin{bmatrix} I & A^{-1}B^{\mathsf{T}} \\ 0 & I \end{bmatrix}$$

Easy to invert $\quad S = C - BA^{-1}B^{\mathsf{T}}$ $\quad$ Easy to invert

Schur complement $Sc(L, F)$

$$L^{-1} = \begin{bmatrix} I & -A^{-1}B^{\mathsf{T}} \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA^{-1} & I \end{bmatrix}$$

Exact solvers via Partial Cholesky Factorization

$$S = C - BA^{-1}B^{\top}$$

Schur complement

$$L^{-1} = \begin{bmatrix} I & -A^{-1}B^{\top} \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA^{-1} & I \end{bmatrix}$$

Exact solvers via Partial Cholesky Factorization

Compute L^{-1} = Compute A^{-1} + Compute S^{-1} + $O(1)$ multiplications

$$S = C - BA^{-1}B^{\top}$$

Schur complement

$$L^{-1} = \begin{bmatrix} I & -A^{-1}B^{\top} \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA^{-1} & I \end{bmatrix}$$

Exact solvers via Partial Cholesky Factorization

Compute L^{-1} = Compute A^{-1} + Compute S^{-1} + $O(1)$ multiplications

$$T(n) = 2T(n/2) + O(n^{\omega})$$
$$= O(n^{\omega})$$

$$S = C - BA^{-1}B^{\top}$$
Schur complement

$$L^{-1} = \begin{bmatrix} I & -A^{-1}B^{\top} \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA^{-1} & I \end{bmatrix}$$

Fast solvers via Partial Cholesky Factorization?

Find a submatrix A such that S can be efficiently approximated,

And recurse on A, S?

$$S = C - BA^{-1}B^{\top}$$

Schur complement

$$L^{-1} = \left[\begin{array}{c|c} I & -A^{-1}B^{\top} \\ \hline 0 & I \end{array}\right] \left[\begin{array}{c|c} A^{-1} & 0 \\ \hline 0 & S^{-1} \end{array}\right] \left[\begin{array}{c|c} I & 0 \\ \hline -BA^{-1} & I \end{array}\right]$$

Fast solvers via Partial Cholesky Factorization?

Find a submatrix A such that S can be efficiently approximated,

And recurse on A, S?

Need a family of matrices closed under Schur complements

$$S = C - BA^{-1}B^{\top}$$

Schur complement

$$L^{-1} = \left[\begin{array}{c|c} I & -A^{-1}B^{\top} \\ \hline 0 & I \end{array}\right] \left[\begin{array}{c|c} A^{-1} & 0 \\ \hline 0 & S^{-1} \end{array}\right] \left[\begin{array}{c|c} I & 0 \\ \hline -BA^{-1} & I \end{array}\right]$$

Fast solvers via Partial Cholesky Factorization?

Find a submatrix A such that S can be efficiently approximated,

And recurse on A, S?

Need a family of matrices closed under Schur complements

Laplacians

$$S = C - BA^{-1}B^{\top}$$

Schur complement

$$L^{-1} = \begin{bmatrix} I & -A^{-1}B^{\top} \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA^{-1} & I \end{bmatrix}$$

Fast solvers via Partial Cholesky Factorization?

Find a submatrix A such that S can be efficiently approximated,

And recurse on A, S?

Need a family of matrices closed under Schur complements

Laplacians, Connection Laplacians

$$S = C - BA^{-1}B^{\top}$$

Schur complement

$$L^{-1} = \begin{bmatrix} I & -A^{-1}B^{\top} \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA^{-1} & I \end{bmatrix}$$

Laplacians

Connection Laplacians

Laplacians

$n \times n$ matrix

Every vertex $\mathrm{x_u} \in \mathbb{R}$

Connection Laplacians

$kn \times kn$ matrix (k fixed, say 3)

Every vertex $\mathrm{y_v} \in \mathbb{C}^k$

Laplacians

$n \times n$ matrix

Every vertex $\mathrm{x_u} \in \mathbb{R}$

$$x^\top L x = \sum_{(u,v)\in E} (x_u - x_v)^2$$

Connection Laplacians

$kn \times kn$ matrix (k fixed, say 3)

Every vertex $\mathrm{y_v} \in \mathbb{C}^k$

$$y^\top M y = \sum_{(u,v)\in E} \left\| y_u - U_{u,v} y_v \right\|^2$$

$U_{u,v}$ is unitary

Laplacians

$n \times n$ matrix

Every vertex $x_u \in \mathbb{R}$

$$x^\top L x = \sum_{(u,v)\in E} (x_u - x_v)^2$$

Connection Laplacians

$kn \times kn$ matrix (k fixed, say 3)

Every vertex $y_v \in \mathbb{C}^k$

$$y^\top M y = \sum_{(u,v)\in E} \left\| y_u - U_{u,v} y_v \right\|^2$$

$U_{u,v}$ is unitary

Applications in cryo-em, image alignment

Fast solvers for Connection Laplacians

Find a submatrix A such that S can be efficiently approximated,

$$S = C - BA^{-1}B^{\top}$$

Schur complement

$$L^{-1} = \begin{bmatrix} I & -A^{-1}B^{\top} \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA^{-1} & I \end{bmatrix}$$

Fast solvers for Connection Laplacians

Find a submatrix A such that S can be efficiently approximated,

"strongly diagonally dominant sets"

$$S = C - BA^{-1}B^{\top}$$

Schur complement

$$L^{-1} = \begin{bmatrix} I & -A^{-1}B^{\top} \\ 0 & I \end{bmatrix} \begin{bmatrix} A^{-1} & 0 \\ 0 & S^{-1} \end{bmatrix} \begin{bmatrix} I & 0 \\ -BA^{-1} & I \end{bmatrix}$$

Fast solvers for Connection Laplacians

[Kyng-Lee-Peng-S-Spielman ’16]

For any connection Laplacian M, subset A that is strongly diagonally dominant, in time $\widetilde{O_\varepsilon}(m)$, can compute a connection laplacian $\widetilde{M}$ on V\A s.t. $\widetilde{M}$ has $\widetilde{O_\varepsilon}(m)$ edges, and

$$\widetilde{M} \approx_\varepsilon Sc(M, V\backslash \mathrm{A})$$

Fast solvers for Connection Laplacians

Combined with sparsification, this gives the only known nearly-linear time solver for Connection Laplacians

[Kyng-Lee-Peng-S-Spielman ’16]
For any connection Laplacian M, with m edges
can solve $Mx = b$ ε–approximately in $\tilde{O}(m \log {}^{1}/_{\varepsilon})$

Where do we go from here?

Approximate Schur complements for other problems? Max-flow?

Faster solvers for more general families of systems? Truss matrices?

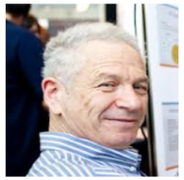
[Kyng-Zhang ‘17] Fast solvers for matrices with structure similar to connection Laplacians, imply fast solvers for all matrices

Theory @ UToronto

Large and diverse theory group

Vibrant Metropolis

Faculty



Allan Borodin
Complexity Theory,
Online Algorithms,
Game Theory

Stephen Cook
Complexity Theory,
Logic

Derek Corneil
Graph Theory

Faith Ellen
Distributed Complexity,
Data Structures,
Lower Bounds

Vassos Hadzilacos
Distributed Computing

Michael Molloy
Graph Theory,
Combinatorics

Aleksandar Nikolov
Algorithms,
Discrepancy Theory,
Privacy

Toniann Pitassi
Complexity Theory,
Proof Complexity

Charles Rackoff
Cryptography,
Complexity Theory

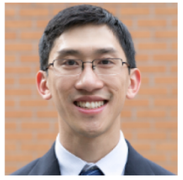
Benjamin Rossman
Circuit Complexity,
Logic

Sushant Sachdeva
Algorithms,
Optimization,
Learning

Nisarg Shah
Mechanism Design,
Game Theory

Sam Toueg
Distributed Computing

Alasdair Urquhart
Complexity of Proofs,
Logic,
History of Logic



Henry Yuen
Quantum Computing,
Quantum Cryptography,
Complexity Theory

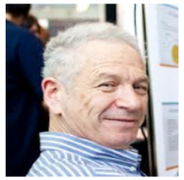
Theory @ UToronto

Large and diverse theory group

Vibrant Metropolis

Postdoc positions for all areas of theory

Faculty



Allan Borodin
Complexity Theory,
Online Algorithms,
Game Theory

Stephen Cook
Complexity Theory,
Logic

Derek Corneil
Graph Theory

Faith Ellen
Distributed Complexity,
Data Structures,
Lower Bounds

Vassos Hadzilacos
Distributed Computing

Michael Molloy
Graph Theory,
Combinatorics

Aleksandar Nikolov
Algorithms,
Discrepancy Theory,
Privacy

Toniann Pitassi
Complexity Theory,
Proof Complexity

Charles Rackoff
Cryptography,
Complexity Theory

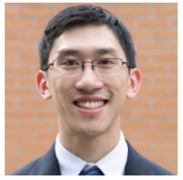
Benjamin Rossman
Circuit Complexity,
Logic

Sushant Sachdeva
Algorithms,
Optimization,
Learning

Nisarg Shah
Mechanism Design,
Game Theory

Sam Toueg
Distributed Computing

Alasdair Urquhart
Complexity of Proofs,
Logic,
History of Logic



Henry Yuen
Quantum Computing,
Quantum Cryptography,
Complexity Theory

Theory @ UToronto

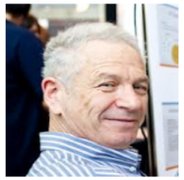
Large and diverse theory group

Vibrant Metropolis

Postdoc positions for all areas of theory

3+ expected open positions for TCS-related areas next year
(2 joint CS+Math)

Faculty



Allan Borodin
Complexity Theory,
Online Algorithms,
Game Theory

Stephen Cook
Complexity Theory,
Logic

Derek Corneil
Graph Theory

Faith Ellen
Distributed Complexity,
Data Structures,
Lower Bounds

Vassos Hadzilacos
Distributed Computing

Michael Molloy
Graph Theory,
Combinatorics

Aleksandar Nikolov
Algorithms,
Discrepancy Theory,
Privacy

Toniann Pitassi
Complexity Theory,
Proof Complexity

Charles Rackoff
Cryptography,
Complexity Theory

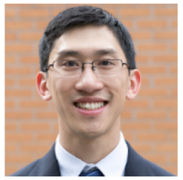
Benjamin Rossman
Circuit Complexity,
Logic

Sushant Sachdeva
Algorithms,
Optimization,
Learning

Nisarg Shah
Mechanism Design,
Game Theory

Sam Toueg
Distributed Computing

Alasdair Urquhart
Complexity of Proofs,
Logic,
History of Logic



Henry Yuen
Quantum Computing,
Quantum Cryptography,
Complexity Theory

?

Theory @ UToronto

Large and diverse theory group

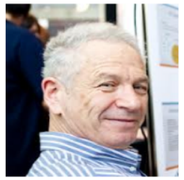
Vibrant Metropolis

Postdoc positions for all areas of theory

3+ expected open positions for TCS-related areas next year
(2 joint CS+Math)

Thanks!

Faculty



Allan Borodin
Complexity Theory,
Online Algorithms,
Game Theory

Stephen Cook
Complexity Theory,
Logic

Derek Corneil
Graph Theory

Faith Ellen
Distributed Complexity,
Data Structures,
Lower Bounds

Vassos Hadzilacos
Distributed Computing

Michael Molloy
Graph Theory,
Combinatorics

Aleksandar Nikolov
Algorithms,
Discrepancy Theory,
Privacy

Toniann Pitassi
Complexity Theory,
Proof Complexity

Charles Rackoff
Cryptography,
Complexity Theory

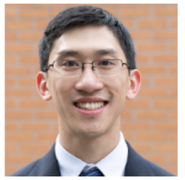
Benjamin Rossman
Circuit Complexity,
Logic

Sushant Sachdeva
Algorithms,
Optimization,
Learning

Nisarg Shah
Mechanism Design,
Game Theory

Sam Toueg
Distributed Computing

Alasdair Urquhart
Complexity of Proofs,
Logic,
History of Logic



Henry Yuen
Quantum Computing,
Quantum Cryptography,
Complexity Theory

?