Advanced Graph Algorithms and Optimization Spring 2020

Solving Laplacian Linear Equations

Rasmus Kyng, Scribe: Hongjie Chen *Lecture 9 — Wednesday, April 22nd*

1 Solving Linear Equations Approximately

Given a Laplacian $\boldsymbol{L}$ of a connected graph and a demand vector $\boldsymbol{d} \perp \mathbf{1}$, we want to find $\boldsymbol{x}^*$ solving the linear equation $\boldsymbol{L}\boldsymbol{x}^* = \boldsymbol{d}$. We are going to focus on fast algorithms for finding approximate (but highly accurate) solutions.

This means we need a notion of an approximate solution. Since our definition is not special to Laplacians, we state it more generally for positive semi-definite matrices.

Definition 1.1. Given PSD matrix $\boldsymbol{M}$ and $\boldsymbol{d} \in \ker(\boldsymbol{M})^\perp$, let $\boldsymbol{M}\boldsymbol{x}^* = \boldsymbol{d}$. We say that $\tilde{\boldsymbol{x}}$ is an ϵ-approximate solution to the linear equation $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{d}$ if

$$\|\tilde{\boldsymbol{x}} - \boldsymbol{x}^*\|_{\boldsymbol{M}}^2 \leq \epsilon \|\boldsymbol{x}^*\|_{\boldsymbol{M}}^2 .$$

Remark 1.2. The requirement $\boldsymbol{d} \in \ker(\boldsymbol{M})^\perp$ can be removed, but this is not important for us.

Theorem 1.3 (Spielman and Teng (2004) [ST04])**.** *Given a Laplacian $\boldsymbol{L}$ of a weighted undirected graph $G = (V, E, \boldsymbol{w})$ with $|E| = m$ and $|V| = n$ and a demand vector $\boldsymbol{d} \in \mathbb{R}^V$, we can find $\tilde{\boldsymbol{x}}$ that is an ϵ-approximate solution to $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$, using an algorithm that takes time $O(m \log^c n \log(1/\epsilon))$ for some fixed constant c and succeeds with probability $1 - 1/n^{10}$.*

In the original algorithm of Spielman and Teng, the exponent on the log in the running time was $c \approx 70$.

Today, we are going to see a simpler algorithm. But first, we'll look at one of the key tools behind all algorithms for solving Laplacian linear equations quickly.

2 Preconditioning and Approximate Gaussian Elimination

Recall our definition of two positive semi-definite matrices being approximately equal.

Definition 2.1 (Spectral approximation)**.** Given $\boldsymbol{A}, \boldsymbol{B} \in S_+^n$, we say that

$$\boldsymbol{A} \approx_K \boldsymbol{B} \text{ if and only if } \frac{1}{1+K}\boldsymbol{A} \preceq \boldsymbol{B} \preceq (1+K)\boldsymbol{A}.$$

Suppose we have a positive definite matrix $\boldsymbol{M} \in S_{++}^n$ and want to solve a linear equation $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{d}$. We can do this using gradient descent or accelerated gradient descent, as we covered in Graded Homework 1. But if we have access to an easy-to-invert matrix that happens to also be a good spectral approximation of $\boldsymbol{M}$, then we can use this to speed up the (accelerated) gradient descent

algorithm. An example of this would be that we have a factorization $\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top \approx_K \boldsymbol{M}$, where $\boldsymbol{\mathcal{L}}$ is lower triangular and sparse, which means we can invert it quickly.

The following lemma, which you will prove in Problem Set 6, makes this preconditioning precise.

Lemma 2.2. *Given a matrix $\boldsymbol{M} \in S^n_{++}$, a vector $\boldsymbol{d}$ and a decomposition $\boldsymbol{M} \approx_K \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top$, we can find $\tilde{\boldsymbol{x}}$ that ϵ-approximately solves $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{d}$, using $O((1+K)\log(K/\epsilon)(T_{matvec} + T_{sol} + n))$ time.*

- *T_{matvec} denotes the time required to compute $\boldsymbol{M}\boldsymbol{z}$ given a vector $\boldsymbol{z}$, i.e. a "matrix-vector multiplication".*
- *T_{sol} denotes the time required to compute $\boldsymbol{\mathcal{L}}^{-1}\boldsymbol{z}$ or $(\boldsymbol{\mathcal{L}}^\top)^{-1}\boldsymbol{z}$ given a vector $\boldsymbol{z}$.*

Dealing with pseudo-inverses. When our matrices have a null space, preconditioning becomes slightly more complicated, but as long as it is easy to project to the complement of the null space, there's no real issue. The following describes precisely what we need (but you can ignore the null-space issue when first reading these notes without losing anything significant).

Lemma 2.3. *Given a matrix $\boldsymbol{M} \in S^n_+$, a vector $\boldsymbol{d} \in \ker(\boldsymbol{M})^\perp$ and a decomposition $\boldsymbol{M} \approx_K \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{D}}\boldsymbol{\mathcal{L}}^\top$, where $\boldsymbol{\mathcal{L}}$ is invertible, we can find $\tilde{\boldsymbol{x}}$ that ϵ-approximately solves $\boldsymbol{M}\boldsymbol{x} = \boldsymbol{d}$, using $O((1+K)\log(K/\epsilon)(T_{matvec} + T_{sol} + T_{proj} + n))$ time.*

- *T_{matvec} denotes the time required to compute $\boldsymbol{M}\boldsymbol{z}$ given a vector $\boldsymbol{z}$, i.e. a "matrix-vector multiplication".*
- *T_{sol} denotes the time required to compute $\boldsymbol{\mathcal{L}}^{-1}\boldsymbol{z}$ and $(\boldsymbol{\mathcal{L}}^\top)^{-1}\boldsymbol{z}$ and $\boldsymbol{\mathcal{D}}^+\boldsymbol{z}$ given a vector $\boldsymbol{z}$.*
- *T_{proj} denotes the time required to compute $\boldsymbol{\Pi}_{\boldsymbol{M}}\boldsymbol{z}$ given a vector $\boldsymbol{z}$.*

Theorem 2.4 (Kyng and Sachdeva (2015) [KS16])**.** *Given a Laplacian $\boldsymbol{L}$ of a weighted undirected graph $G = (V, E, \boldsymbol{w})$ with $|E| = M$ and $|V| = n$, we can find a decomposition $\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top \approx_{0.5} \boldsymbol{L}$, such that $\boldsymbol{\mathcal{L}}$ has number of non-zeroes $\mathrm{nnz}(\boldsymbol{\mathcal{L}}) = O(m\log^3 n)$, with probability at least $1 - 3/n^5$. in time $O(m\log^3 n)$.*

We can combine Theorem 2.4 with Lemma 2.3 to get a fast algorithm for solving Laplacian linear equations.

Corollary 2.5. *Given a Laplacian $\boldsymbol{L}$ of a weighted undirected graph $G = (V, E, \boldsymbol{w})$ with $|E| = m$ and $|V| = n$ and a demand vector $\boldsymbol{d} \in \mathbb{R}^V$, we can find $\tilde{\boldsymbol{x}}$ that is an ϵ-approximate solution to $\boldsymbol{L}\boldsymbol{x} = \boldsymbol{d}$, using an algorithm that takes time $O(m\log^3 n\log(1/\epsilon))$ and succeeds with probability $1 - 1/n^{10}$.*

Proof sketch. First we need to get a factorization that confirms to Lemma 2.3. The decomposition $\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top$ provided by Theorem 2.4 can be rewritten as $\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top = \widetilde{\boldsymbol{\mathcal{L}}}\boldsymbol{\mathcal{D}}(\widetilde{\boldsymbol{\mathcal{L}}})^\top$ where $\widetilde{\boldsymbol{\mathcal{L}}}$ is equal to $\boldsymbol{\mathcal{L}}$ except $\boldsymbol{\mathcal{L}}(n,n) = 1$ and we let $\boldsymbol{\mathcal{D}}$ be the identity matrix, except $\boldsymbol{\mathcal{D}}(n,n) = 0$. This ensures $\boldsymbol{\mathcal{D}}^+ = \boldsymbol{\mathcal{D}}$ and that $\widetilde{\boldsymbol{\mathcal{L}}}$ is invertible and lower triagular with $O(m\log^3 n\log(1/\epsilon))$ non-zeros. We note that the inverse of an invertible lower or upper triangular matrix with N non-zeros can be applied in time $O(N)$ given an adjacency list representation of the matrix. Finally, as $\ker(\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top) = \mathrm{span}\{\mathbf{1}\}$, we have $\boldsymbol{\Pi}_{\widetilde{\boldsymbol{\mathcal{L}}}\boldsymbol{\mathcal{D}}(\widetilde{\boldsymbol{\mathcal{L}}})^\top} = \boldsymbol{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, and this projection matrix can be applied in $O(n)$ time. Altogether, this means that $T_{\mathrm{matvec}} + T_{\mathrm{sol}} + T_{\mathrm{proj}} = O(n)$, which suffices to complete the proof. □

3 Approximate Gaussian Elimination Algorithm

Recall *Gaussian Elimination / Cholesky decomposition* of a graph Laplacian $\boldsymbol{L}$. We will use $\boldsymbol{A}(:, i)$ to denote the the ith column of a matrix $\boldsymbol{A}$. We can write the algorithm as

Algorithm 1: Gaussian Elimination / Cholesky Decomposition

Input: Graph Laplacian $\boldsymbol{L}$
Output: Lower triangular $\boldsymbol{\mathcal{L}}$ s.t. $\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top = \boldsymbol{L}$
Let $\boldsymbol{S}_0 = \boldsymbol{L}$;
for $i = 1$ *to* $i = n - 1$ **do**
  $\boldsymbol{l}_i = \frac{1}{\sqrt{\boldsymbol{S}_{i-1}(i,i)}} \boldsymbol{S}_{i-1}(:, i)$;
  $\boldsymbol{S}_i = \boldsymbol{S}_{i-1} - \boldsymbol{l}_i \boldsymbol{l}_i^\top$.
$\boldsymbol{l}_n = \boldsymbol{0}_{n \times 1}$;
return $\boldsymbol{\mathcal{L}} = \begin{bmatrix} \boldsymbol{l}_1 \cdots \boldsymbol{l}_n \end{bmatrix}$;

We want to introduce some notation that will help us describe and analyze a faster version of Gaussian elimination – one that uses sampling to create a sparse approximation of the decomposition.

Consider a Laplacian $\boldsymbol{S}$ of a graph H and a vertex v of H. We define $\textsc{Star}(v, \boldsymbol{S})$ to be the Laplacian of the subgraph of H consisting of edges incident on v. We define

$$\textsc{Clique}(v, \boldsymbol{S}) = \textsc{Star}(v, \boldsymbol{S}) - \frac{1}{\boldsymbol{S}(v, v)} \boldsymbol{S}(:, v) \boldsymbol{S}(:, v)^\top$$

For example, suppose

$$\boldsymbol{L} = \begin{pmatrix} W & -\boldsymbol{a}^\top \\ -\boldsymbol{a} & \operatorname{diag}(\boldsymbol{a}) + \boldsymbol{L}_{-1} \end{pmatrix}$$

Then

$$\textsc{Star}(1, \boldsymbol{L}) = \begin{pmatrix} W & -\boldsymbol{a}^\top \\ -\boldsymbol{a} & \operatorname{diag}(\boldsymbol{a}) \end{pmatrix} \text{ and } \textsc{Clique}(1, \boldsymbol{L}) = \begin{pmatrix} 0 & \boldsymbol{0} \\ \boldsymbol{0} & \operatorname{diag}(\boldsymbol{a}) - \frac{1}{W} \boldsymbol{a}\boldsymbol{a}^\top \end{pmatrix}$$

which is illustrated in Figure 1.

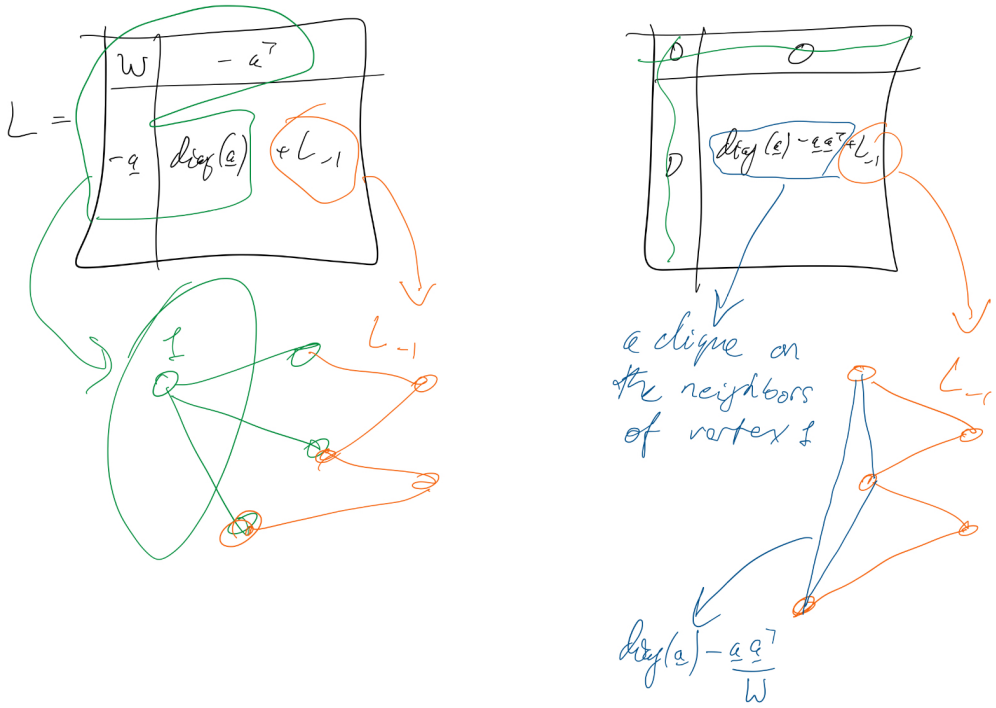

Figure 1: Gaussian Elimination: $\textsc{Clique}(1, \boldsymbol{L}) = \textsc{Star}(1, \boldsymbol{L}) - \frac{1}{\boldsymbol{L}(1,1)} \boldsymbol{L}(:,1)\boldsymbol{L}(:,1)^\top$.

In Lecture 7, we proved that $\textsc{Clique}(v, \boldsymbol{S})$ is a graph Laplacian – it follows from the proof of Claim 1.1 in that lecture. Thus we have that following.

Claim 3.1. *If $\boldsymbol{S}$ is the Laplacian of a connected graph, then $\textsc{Clique}(v, \boldsymbol{S})$ is a graph Laplacian.*

Note that in Algorithm 1, we have $\boldsymbol{l}_i \boldsymbol{l}_i^\top = \textsc{Star}(v_i, \boldsymbol{S}_{i-1}) - \textsc{Clique}(v_i, \boldsymbol{S}_{i-1})$. The update rule can be rewritten as

$$\boldsymbol{S}_i = \boldsymbol{S}_{i-1} - \textsc{Star}(v_i, \boldsymbol{S}_{i-1}) + \textsc{Clique}(v_i, \boldsymbol{S}_{i-1}),$$

This also provides way to understand why Gaussian Elimination is slow in some cases. At each step, one vertex is eliminated, but a clique is added to the subgraph on the remaining vertices, making the graph denser. And at the ith step, computing $\textsc{Star}(v_i, \boldsymbol{S}_{i-1})$ takes around $\deg(v_i)$ time, but computing $\textsc{Clique}(v_i, \boldsymbol{S}_{i-1})$ requires around $\deg(v_i)^2$ time. In order to speed up Gaussian Elimination, the algorithmic idea of [KS16] is to plug in a sparser appproximate of the intended clique instead of the entire one.

The following procedure $\textsc{CliqueSample}(v, \boldsymbol{S})$ produces a sparse approximation of $\textsc{clique}(v, \boldsymbol{S})$. Let V be the vertex set of the graph associated with $\boldsymbol{S}$ and E the edge set. We define $\boldsymbol{b}_{i,j} \in \mathbb{R}^V$ to be the vector with

$$\boldsymbol{b}_{i,j}(i) = 1 \text{ and } \boldsymbol{b}_{i,j}(j) = -1 \text{ and } \boldsymbol{b}_{i,j}(k) = 0 \text{ for } k \neq i, j.$$

Given weights $\boldsymbol{w} \in \mathbb{R}^E$ and a vertex $v \in V$, we let

$$\boldsymbol{w}_v = \sum_{(u,v) \in E} \boldsymbol{w}(u, v).$$

Algorithm 2: CliqueSample$(v, \boldsymbol{S})$

Input: Graph Laplacian $\boldsymbol{S} \in \mathbb{R}^{V \times V}$, of a graph with edge weights $\boldsymbol{w}$, and vertex $v \in V$
Output: $\boldsymbol{Y}_v \in \mathbb{R}^{V \times V}$ sparse approximation of clique$(v, \boldsymbol{S})$
$\boldsymbol{Y}_v \leftarrow \mathbf{0}_{n \times n}$;
foreach *Multiedge* $e = (v, i)$ *from* v *to a neighbor* i **do**
  Randomly pick a neighbor j of v with probability $\frac{\boldsymbol{w}(j,v)}{\boldsymbol{w}_v}$;
  If $i \neq j$, let $\boldsymbol{Y}_v \leftarrow \boldsymbol{Y}_v + \frac{\boldsymbol{w}(i,v)\boldsymbol{w}(j,v)}{\boldsymbol{w}(i,v)+\boldsymbol{w}(j,v)} \boldsymbol{b}_{i,j} \boldsymbol{b}_{i,j}^\top$;
return $\boldsymbol{Y}_v$;

Remark 3.2. We can implement each sampling of a neighbor j in $O(1)$ time using a classical algorithm known as Walker's method (also known as the Alias method or Vose's method). This algorithm requires an additional $O(\deg_{\boldsymbol{S}}(v))$ time to initialize a data structure used for sampling. Overall, this means the total time for $O(\deg_{\boldsymbol{S}}(v))$ samples is still $O(\deg_{\boldsymbol{S}}(v))$.

Lemma 3.3. $\mathbb{E}[\boldsymbol{Y}_v] = \text{Clique}(v, \boldsymbol{S})$.

Proof. Let $\boldsymbol{C} = \text{Clique}(v, \boldsymbol{S})$. Observe that both $\mathbb{E}[\boldsymbol{Y}_v]$ and $\boldsymbol{C}$ are Laplacians. Thus it suffices to verify $\mathbb{E}\, \boldsymbol{Y}_v(i,j) = \boldsymbol{C}(i,j)$ for $i \neq j$.

$$\boldsymbol{C}(i,j) = -\frac{\boldsymbol{w}(i,v)\boldsymbol{w}(j,v)}{\boldsymbol{w}_v},$$

$$\mathbb{E}\, \boldsymbol{Y}_v(i,j) = -\frac{\boldsymbol{w}(i,v)\boldsymbol{w}(j,v)}{\boldsymbol{w}(i,v)+\boldsymbol{w}(j,v)} \left(\frac{\boldsymbol{w}(j,v)}{\boldsymbol{w}_v} + \frac{\boldsymbol{w}(i,v)}{\boldsymbol{w}_v} \right) = -\frac{\boldsymbol{w}(i,v)\boldsymbol{w}(j,v)}{\boldsymbol{w}_v} = -\boldsymbol{C}(i,j).$$

□

Remark 3.4. Lemma 3.3 shows that CliqueSample$(v, \boldsymbol{L})$ produces the original Clique$(v, \boldsymbol{L})$ in expectation.

Now, we define *Approximate Gaussian Elimination*.

Algorithm 3: Approximate Gaussian Elimination / Cholesky Decomposition

Input: Graph Laplacian $\boldsymbol{L}$
Output: Lower triangular[a] $\boldsymbol{\mathcal{L}}$ as given in Theorem 2.4
Let $\boldsymbol{S}_0 = \boldsymbol{L}$;
Generate a random permutation π on $[n]$;
for $i = 1$ *to* $i = n - 1$ **do**
  $\boldsymbol{l}_i = \frac{1}{\sqrt{\boldsymbol{S}_{i-1}(\pi(i),\pi(i))}} \boldsymbol{S}_{i-1}(:, \pi(i))$;
  $\boldsymbol{S}_i = \boldsymbol{S}_{i-1} - \text{Star}(\pi(i), \boldsymbol{S}_{i-1}) + \text{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1})$
$\boldsymbol{l}_n = \mathbf{0}_{n \times 1}$;
return $\boldsymbol{\mathcal{L}} = \begin{bmatrix} \boldsymbol{l}_1 \cdots \boldsymbol{l}_n \end{bmatrix}$ *and* π;

[a] $\boldsymbol{\mathcal{L}}$ is not actually lower triangular. However, if we let $\boldsymbol{P}_\pi$ be the permutation matrix corresponding to π, then $\boldsymbol{P}_\pi \boldsymbol{\mathcal{L}}$ is lower triangular. Knowing the ordering that achieves this is enough to let us implement forward and backward substitution for solving linear equations in $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{L}}^\top$.

Note that if we replace CliqueSample$(\pi(i), \boldsymbol{S}_{i-1})$ by Clique$(\pi(i), \boldsymbol{S}_{i-1})$ at each step, then we can recover Gaussian Elimination, but with a random elimination order.

4 Analyzing Approximate Gaussian Elimination

In this Section, we're going to analyze Approximate Gaussian Elimination, and see why it works.

Ultimately, the main challenge in proving Theorem 2.4 will be to prove for the output $\boldsymbol{\mathcal{L}}$ of Algorithm 3 that with high probability

$$0.5\boldsymbol{L} \preceq \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top \preceq 1.5\boldsymbol{L}. \tag{1}$$

We can reduce this to proving that with high probability

$$\left\|\boldsymbol{L}^{+/2}(\boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top - \boldsymbol{L})\boldsymbol{L}^{+/2}\right\| \leq 0.5 \tag{2}$$

Ultimately, the proof is going to have a lot in common with our proof of Matrix Bernstein in Lecture 8. Overall, the lesson there was that when we have a sum of independent, zero-mean random matrices, we can show that the sum is likely to have small spectral norm if the spectral norm of each random matrix is small, and the matrix-valued variance is also small.

Thus, to replicate the proof, we need control over

1. The *sample norms*.
2. The *sample variance*.

But, there is seemlingly another major obstacle: We are trying to analyze a process where the samples are far from independent. Each time we sample edges, we add new edges to the remaining graph, which we will the later sample again. This creates a lot of dependencies between the samples, which we have to handle.

However, it turns out that independence is more than what is needed to prove concentration. Instead, it suffices to have a sequence of random variables such that each is mean-zero in expectation, conditional on the previous ones. This is called a martingale difference sequence. We'll now learn about those.

4.1 Normalization, a.k.a. Isotropic Position

Since our analysis requires frequently measuring matrices after right and left-multiplication by $\boldsymbol{L}^{+/2}$, we reintroduce the "normalizing map" $\Phi : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ defined by

$$\Phi(\boldsymbol{A}) = \boldsymbol{L}^{+/2}\boldsymbol{A}\boldsymbol{L}^{+/2}.$$

We previously saw this in Lectures 7 and 8.

4.2 Martingales

A scalar martingale is a sequence of random variables $Z_0, \ldots, Z_k$, such that

$$\mathbb{E}\left[Z_i \mid Z_0, \ldots, Z_{i-1}\right] = Z_{i-1}. \tag{3}$$

That is, conditional on the outcome of all the previous random variables, the expectation of Z_i equals Z_{i-1}. If we unravel the sequence of conditional expectations, we get that *without conditioning* $\mathbb{E}[Z_k] = \mathbb{E}[Z_0]$.

Typically, we use martingales to show a statement along like "Z_k is concentrated around $\mathbb{E}[Z_k]$".

We can also think of a martingale in terms of the sequence of changes in the Z_i variables. Let $X_i = Z_i - Z_{i-1}$. The sequence of X_is is called a martingale difference sequence. We can now state the martingale condition as

$$\mathbb{E}[X_i \mid Z_0, \ldots, Z_{i-1}] = 0.$$

And because Z_0 and $X_1, \ldots, X_{i-1}$ completely determine $Z_1, \ldots, Z_{i-1}$, we could also write the martingale condition equivalently as

$$\mathbb{E}[X_i \mid Z_0, X_1, \ldots, X_{i-1}] = 0.$$

Crucially, we can write

$$Z_k = Z_0 + \sum_{i=1}^{k} Z_i - Z_{i-1} = Z_0 + \sum_{i=1}^{k} X_i$$

and when we are trying to prove concentration, the martingale difference property of the X_i's is often "as good as" independence, meaning that $\sum_{i=1}^{k} X_i$ concentrates similarly to a sum of independent random variables.

Matrix-valued martingales. We can also define matrix-valued martingales. In this case, we replace the martingalue condition of Equation (3), with the condition that the whole matrix stays the same in expectation. For example, we could have a sequence of random matrices $\boldsymbol{Z}_0, \ldots, \boldsymbol{Z}_k \in \mathbb{R}^{n \times n}$, such that

$$\mathbb{E}[\boldsymbol{Z}_i \mid \boldsymbol{Z}_0, \ldots, \boldsymbol{Z}_{i-1}] = \boldsymbol{Z}_{i-1}. \tag{4}$$

Lemma 4.1. *Let* $\boldsymbol{L}_i = \boldsymbol{S}_i + \sum_{j=1}^{i} \boldsymbol{l}_j \boldsymbol{l}_j^\top$ *for* $i = 1, ..., n$ *and* $\boldsymbol{L}_0 = \boldsymbol{S}_0 = \boldsymbol{L}$. *Then*

$$\mathbb{E}[\boldsymbol{L}_i | \textit{all random variables before } \textsc{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1})] = \boldsymbol{L}_{i-1}.$$

Proof. Let's only consider $i = 1$ here as other cases are similar.

$$\boldsymbol{L}_0 = \boldsymbol{L} = \boldsymbol{l}_1 \boldsymbol{l}_1^\top + \textsc{Clique}(v, \boldsymbol{L}) + \boldsymbol{L}_{-1}$$

$$\begin{aligned}
\boldsymbol{L}_1 &= \boldsymbol{l}_1 \boldsymbol{l}_1^\top + \textsc{CliqueSample}(v, \boldsymbol{L}) + \boldsymbol{L}_{-1} \\
\mathbb{E}[\boldsymbol{L}_1 | \pi(1)] &= \boldsymbol{l}_1 \boldsymbol{l}_1^\top + \mathbb{E}[\textsc{CliqueSample}(v, \boldsymbol{L}) | \pi(1)] + \boldsymbol{L}_{-1} \\
&= \boldsymbol{l}_1 \boldsymbol{l}_1^\top + \textsc{Clique}(v, \boldsymbol{L}) + \boldsymbol{L}_{-1} \\
&= \boldsymbol{L}_0
\end{aligned}$$

where we used Lemma 3.3 to get $\mathbb{E}[\textsc{CliqueSample}(v, \boldsymbol{L}) | \pi(1)] = \textsc{Clique}(v, \boldsymbol{L})$. □

Remark 4.2. $\sum_{j=1}^{i} \boldsymbol{l}_j \boldsymbol{l}_j^\top$ can be treated as what has already been eliminated by (Approximate) Gaussian Elimination, while $\boldsymbol{S}_i$ is what still left or going to be eliminated. In Approximate Gaussian Elimination, $\boldsymbol{L}_n = \sum_{i=1}^{n} \boldsymbol{l}_i \boldsymbol{l}_i^\top$ and our goal is to show that $\boldsymbol{L}_n \approx_K \boldsymbol{L}$. Note that $\boldsymbol{L}_i$ is always equal to the original Laplacian $\boldsymbol{L}$ for all i in Gaussian Elimination. Lemma 4.1 demonstrates that $\boldsymbol{L}_0, \boldsymbol{L}_1, ..., \boldsymbol{L}_n$ forms a matrix martingale.

Ultimately, our plan is to use this matrix martingale structure to show that "$\boldsymbol{L}_n$ is concentrated around $\boldsymbol{L}$" in some appropriate sense. More precisely, the spectral approximation we would like to show can be established by showing that "$\Phi(\boldsymbol{L}_n)$ is concentrated around $\Phi(\boldsymbol{L})$"

4.3 Martingale Difference Sequence as Edge-Samples

We start by taking a slightly different view of the observations we used to prove Lemma 4.1. Recall that $\boldsymbol{L}_i = \boldsymbol{S}_i + \sum_{j=1}^{i} \boldsymbol{l}_j \boldsymbol{l}_j^\top$, and $\boldsymbol{L}_{i-1} = \boldsymbol{S}_{i-1} + \sum_{j=1}^{i-1} \boldsymbol{l}_j \boldsymbol{l}_j^\top$ and

$$\boldsymbol{S}_i = \boldsymbol{S}_{i-1} - \textsc{Star}(\pi(i), \boldsymbol{S}_{i-1}) + \textsc{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1})\,.$$

Putting these together, we get

$$\begin{aligned}
\boldsymbol{L}_i - \boldsymbol{L}_{i-1} &= \boldsymbol{l}_i \boldsymbol{l}_i^\top + \textsc{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1}) - \textsc{Star}(\pi(i), \boldsymbol{S}_{i-1}) \\
&= \textsc{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1}) - \textsc{Clique}(\pi(i), \boldsymbol{S}_{i-1}) \qquad (5) \\
&= \textsc{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1}) - \mathbb{E}\left[\textsc{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1}) \mid \text{preceding samples}\right] \\
&\qquad\qquad \text{by Lemma 3.3.}
\end{aligned}$$

In particular, recall that by Lemma 3.3, conditional on the randomness before the call to $\textsc{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1})$, we have

$$\mathbb{E}\left[\textsc{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1}) \mid \text{preceding samples}\right] = \textsc{Clique}(\pi(i), \boldsymbol{S}_{i-1})$$

Adopting the notation of Lemma 3.3 we write

$$\boldsymbol{Y}_{\pi(i)} = \textsc{CliqueSample}(\pi(i), \boldsymbol{S}_{i-1})$$

and we further introduce notation each multi-edge sample for $e \in \textsc{Star}(\pi(i), \boldsymbol{S}_{i-1})$, as $\boldsymbol{Y}_{\pi(i),e}$, denoting the random edge Laplacian sampled when the algorithm is processing multi-edge e. Thus, conditional on preceding samples, we have

$$\boldsymbol{Y}_{\pi(i)} = \sum_{e \in \textsc{Star}(\pi(i), \boldsymbol{S}_{i-1})} \boldsymbol{Y}_{\pi(i),e} \qquad (6)$$

Note that even the number of multi-edges in $\textsc{Star}(\pi(i), \boldsymbol{S}_{i-1})$ depends on the preceding samples. We also want to associate zero-mean variables with each edge. Conditional on preceding samples, we also define

$$\boldsymbol{X}_{i,e} = \Phi\left(\boldsymbol{Y}_{\pi(i),e} - \mathbb{E}\left[\boldsymbol{Y}_{\pi(i),e}\right]\right) \text{ and } \boldsymbol{X}_i = \sum_{e \in \textsc{Star}(\pi(i), \boldsymbol{S}_{i-1})} \boldsymbol{X}_{i,e}$$

and combining this with Equations (5) and (6)

$$\boldsymbol{X}_i = \Phi(\boldsymbol{Y}_{\pi(i)} - \mathbb{E}\left[\boldsymbol{Y}_{\pi(i)}\right]) = \Phi(\boldsymbol{L}_i - \boldsymbol{L}_{i-1})$$

Altogether, we can write

$$\Phi\left(\boldsymbol{L}_n - \boldsymbol{L}\right) = \sum_{i=1}^{n} \Phi(\boldsymbol{L}_i - \boldsymbol{L}_{i-1}) = \sum_{i=1}^{n} \boldsymbol{X}_i = \sum_{i=1}^{n} \sum_{e \in \textsc{Star}(\pi(i), \boldsymbol{S}_{i-1})} \boldsymbol{X}_{i,e}$$

Note that the $\boldsymbol{X}_{i,e}$ variables form a martingale difference sequence, because the linearity of Φ ensures they are zero-mean conditional on preceding randomness.

4.4 Stopped Martingales

Unfortunately, directly analyzing the concentration properties of the $\boldsymbol{L}_i$ martingale that we just introduced turns out to be difficult. The reason is that we're trying to prove some very delicate multiplicative error guarantees. And, if we analyze $\boldsymbol{L}_i$, we find that the multiplicative error is not easy to control, *after it's already gotten big*. But that's not really what we care about anyway: We want to say it never gets big in the first place, with high probability. So we need to introduce another martingale, that lets us ignore the bad case when the error has already gotten too big. At the same time, we also need to make sure that statements about our new martingale can help us prove guarantees about $\boldsymbol{L}_i$. Fortunately, we can achieve both at once. The technique we use is related to the much broader topic of martingale *stopping times*, which we only scratch the surface of here. We're also going to be quite informal about it, in the interest of brevity. Lecture notes by Tropp [Tro19] give a more formal introduction for those who are interested.

We define the stopped martingale sequence $\tilde{\boldsymbol{L}}_i$ by

$$\tilde{\boldsymbol{L}}_i = \begin{cases} \boldsymbol{L}_i & \text{if for all } j < i \text{ we have } \boldsymbol{L}_i \preceq 1.5\boldsymbol{L} \\ \boldsymbol{L}_{j^*} & \text{for } j^* \text{ being the least } j \text{ such that } \boldsymbol{L}_j \not\preceq 1.5\boldsymbol{L} \end{cases} \tag{7}$$

Figure 2 shows the $\tilde{\boldsymbol{L}}_i$ martingale getting stuck at the first time $\boldsymbol{L}_{j^*} \not\preceq 1.5\boldsymbol{L}$.

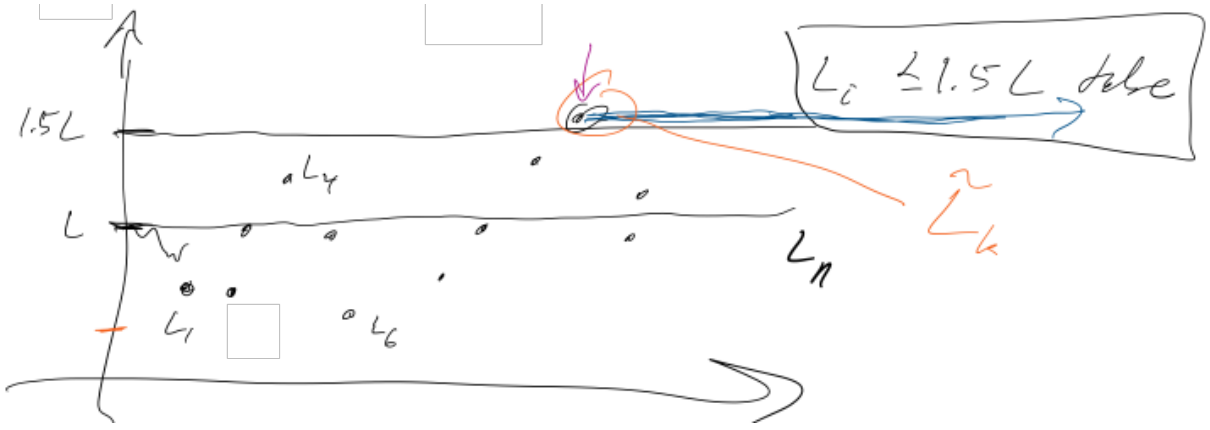

Figure 2: Gaussian Elimination : $\textsc{Clique}(1, \boldsymbol{L}) = \textsc{Star}(1, \boldsymbol{L}) - \frac{1}{\boldsymbol{L}(1,1)}\boldsymbol{L}(:,1)\boldsymbol{L}(:,1)^\top$.

We state the following without proof:

Claim 4.3.

1. *The sequence* $\left\{\tilde{\boldsymbol{L}}_i\right\}$ *for* $i = 0, \ldots, n$ *is a martingale.*
2. $\left\|\boldsymbol{L}^{+/2}(\tilde{\boldsymbol{L}}_i - \boldsymbol{L})\boldsymbol{L}^{+/2}\right\| \leq 0.5$ *implies* $\left\|\boldsymbol{L}^{+/2}(\boldsymbol{L}_i - \boldsymbol{L})\boldsymbol{L}^{+/2}\right\| \leq 0.5$

The martingale property also implies that the unconditional expectation satisfies $\mathbb{E}\left[\tilde{\boldsymbol{L}}_n\right] = \boldsymbol{L}$. The proof of the claim is easy to sketch: For Part 1, each difference is zero-mean if the condition has not been violated, and is identically zero (and hence zero-mean) if it has been violated. For Part 2,

if the martingale $\left\{\tilde{\boldsymbol{L}}_i\right\}$ has stopped, then $\left\|\boldsymbol{L}^{+/2}(\tilde{\boldsymbol{L}}_i - \boldsymbol{L})\boldsymbol{L}^{+/2}\right\| \leq 0.5$ is false, and the implication is vacuosly true. If the, on the other hand, if the martingale has not stopped, the quantities are equal, because $\tilde{\boldsymbol{L}}_i = \boldsymbol{L}_i$, and again it's easy to see the implication holds.

Thus, ultimately, our strategy is goin to be to show that $\left\|\boldsymbol{L}^{+/2}(\tilde{\boldsymbol{L}}_i - \boldsymbol{L})\boldsymbol{L}^{+/2}\right\| \leq 0.5$ with high probability. Expressed using the normalizing map $\Phi(\cdot)$, our goal is to show that with high probability

$$\left\|\Phi(\tilde{\boldsymbol{L}}_n - \boldsymbol{L})\right\| \leq 0.5.$$

Stopped martingale difference sequence. In order to prove the spectral norm bound, we want to express the $\left\{\tilde{\boldsymbol{L}}_i\right\}$ martingale in terms of a sequence of martingale differences. To this end, we define $\tilde{\boldsymbol{X}}_i = \Phi(\tilde{\boldsymbol{L}}_i - \tilde{\boldsymbol{L}}_{i-1})$. This ensures that

$$\tilde{\boldsymbol{X}}_i = \begin{cases} \boldsymbol{X}_i & \text{if for all } j < i \text{ we have } \boldsymbol{L}_i \preceq 1.5\boldsymbol{L} \\ \boldsymbol{0} & \text{otherwise} \end{cases} \tag{8}$$

Whenever the modified martingale $\tilde{\boldsymbol{X}}_i$ has not yet stopped, we also introduce individual modified edge samples $\tilde{\boldsymbol{X}}_{i,e} = \boldsymbol{X}_{i,e}$. If the martingale *has* stopped, i.e. $\tilde{\boldsymbol{X}}_i = \boldsymbol{0}$, then we can take these edge samples $\tilde{\boldsymbol{X}}_{i,e}$ to be zero. We can now write

$$\Phi\left(\tilde{\boldsymbol{L}}_n - \boldsymbol{L}\right) = \sum_{i=1}^{n} \Phi(\tilde{\boldsymbol{L}}_i - \tilde{\boldsymbol{L}}_{i-1}) = \sum_{i=1}^{n} \tilde{\boldsymbol{X}}_i = \sum_{i=1}^{n} \sum_{e \in \textsc{Star}(\pi(i), \boldsymbol{S}_{i-1})} \tilde{\boldsymbol{X}}_{i,e}.$$

Thus, we can see that Equation (2) is implied by

$$\left\|\sum_{i=1}^{n} \tilde{\boldsymbol{X}}_i\right\| \leq 0.5. \tag{9}$$

4.5 Sample Norm Control

In this Subsection, we're going to see that the norms of each multi-edge sample is controlled throughout the algorithm.

Lemma 4.4. *Given two Laplacians $\boldsymbol{L}$ and $\boldsymbol{S}$ on the same vertex set.*[1] *If each multiedge e of* $\textsc{Star}(v, \boldsymbol{S})$ *has bounded norm in the following sense,*

$$\left\|\boldsymbol{L}^{+/2}\boldsymbol{w}_{\boldsymbol{S}}(e)\boldsymbol{b}_e\boldsymbol{b}_e^\top\boldsymbol{L}^{+/2}\right\| \leq R,$$

then each possible sampled multiedge e' of $\textsc{CliqueSample}(v, \boldsymbol{S})$ *also satisfies*

$$\left\|\boldsymbol{L}^{+/2}\boldsymbol{w}_{\text{new}}(e')\boldsymbol{b}_{e'}\boldsymbol{b}_{e'}^\top\boldsymbol{L}^{+/2}\right\| \leq R.$$

[1] $\boldsymbol{L}$ can be regarded as the original Laplacian we care about, while $\boldsymbol{S}$ can be regarded as some intermediate Laplacian appearing during Approximate Gaussian Elimination.

Proof. Let $\boldsymbol{w} = \boldsymbol{w}_{\boldsymbol{S}}$ for simplicity. Consider a sampled edge between i and j with weight $\boldsymbol{w}_{\text{new}}(i,j) = \boldsymbol{w}(i,v)\boldsymbol{w}(j,v)/(\boldsymbol{w}(i,v)+\boldsymbol{w}(j,v))$.

$$\begin{aligned}
\left\|\boldsymbol{L}^{+/2}\boldsymbol{w}_{\text{new}}(i,j)\boldsymbol{b}_{ij}\boldsymbol{b}_{ij}^{\top}\boldsymbol{L}^{+/2}\right\| &= \boldsymbol{w}_{\text{new}}(i,j)\left\|\boldsymbol{L}^{+/2}\boldsymbol{b}_{ij}\boldsymbol{b}_{ij}^{\top}\boldsymbol{L}^{+/2}\right\| \\
&= \boldsymbol{w}_{\text{new}}(i,j)\left\|\boldsymbol{L}^{+/2}\boldsymbol{b}_{ij}\right\|^2 \\
&\leq \boldsymbol{w}_{\text{new}}(i,j)\left(\left\|\boldsymbol{L}^{+/2}\boldsymbol{b}_{iv}\right\|^2 + \left\|\boldsymbol{L}^{+/2}\boldsymbol{b}_{jv}\right\|^2\right) \\
&= \frac{\boldsymbol{w}(j,v)}{\boldsymbol{w}(i,v)+\boldsymbol{w}(j,v)}\left\|\boldsymbol{L}^{+/2}\boldsymbol{w}(i,v)\boldsymbol{b}_{iv}\boldsymbol{b}_{iv}^{\top}\boldsymbol{L}^{+/2}\right\| + \\
&\quad \frac{\boldsymbol{w}(i,v)}{\boldsymbol{w}(i,v)+\boldsymbol{w}(j,v)}\left\|\boldsymbol{L}^{+/2}\boldsymbol{w}(j,v)\boldsymbol{b}_{jv}\boldsymbol{b}_{jv}^{\top}\boldsymbol{L}^{+/2}\right\| \\
&\leq \frac{\boldsymbol{w}(j,v)}{\boldsymbol{w}(i,v)+\boldsymbol{w}(j,v)}R + \frac{\boldsymbol{w}(i,v)}{\boldsymbol{w}(i,v)+\boldsymbol{w}(j,v)}R \\
&= R
\end{aligned}$$

The first inequality uses the triangle inequality of effective resistance in $\boldsymbol{L}$, in that effective resistance is a distance as proved in lecture 6. The second inequality just uses the conditions of this lemma. □

Remark 4.5. Lemma 4.4 only requires that each single multiedge has small norm instead of that the sum of all edges between a pair of vertices have small norm. And this lemma tells us, after sampling, each multiedge in the new graph still satisfies the bounded norm condition.

From the Lemma, we can conclude that each edge sample $\boldsymbol{Y}_{\pi(i),e}$ satisfies $\left\|\Phi(\boldsymbol{Y}_{\pi(i),e})\right\| \leq R$ provided the assumptions of the Lemma hold. Let's record this observation as a Lemma.

Lemma 4.6. *If for all $e \in \textsc{Star}(v, \boldsymbol{S}_i)$,*

$$\left\|\Phi(\boldsymbol{w}_{\boldsymbol{S}_i}(e)\boldsymbol{b}_e\boldsymbol{b}_e^{\top})\right\| \leq R.$$

then all $e \in \textsc{Star}(\pi(i), \boldsymbol{S}_i)$,

$$\left\|\Phi(\boldsymbol{Y}_{\pi(i),e})\right\| \leq R.$$

Preprocessing by multi-edge splitting. In the original graph of Laplacian $\boldsymbol{L}$ of graph $G = (V, E, \boldsymbol{w})$, we have for each edge $\hat{e}$ that

$$\boldsymbol{w}(\hat{e})\boldsymbol{b}_{\hat{e}}\boldsymbol{b}_{\hat{e}}^{\top} \preceq \sum_e \boldsymbol{w}(e)\boldsymbol{b}_e\boldsymbol{b}_e^{\top} = \boldsymbol{L}$$

This also implies that

$$\left\|\boldsymbol{L}^{+/2}\boldsymbol{w}(\hat{e})\boldsymbol{b}_{\hat{e}}\boldsymbol{b}_{\hat{e}}^{\top}\boldsymbol{L}^{+/2}\right\| \leq 1.$$

Now, that means that if we split every original edge e of the graph into K multi-edges $e_1, \ldots e_K$, with a fraction $1/K$ of the weight, we get a new graph $G' = (V, E', \boldsymbol{w}')$ such that

Claim 4.7.

1. *G' and G have the same graph Laplacian.*

2. $|E'| = K\,|E|$

3. *For every multi-edge in* G'

$$\left\| \boldsymbol{L}^{+/2} \boldsymbol{w}'(e) \boldsymbol{b}_e \boldsymbol{b}_e^\top \boldsymbol{L}^{+/2} \right\| \leq 1/K.$$

Before we run Approximate Gaussian Elimination, we are going to do this multi-edge splitting to ensure we have control over multi-edge sample norms. Combined with Lemma 4.4 immediately establishes the next lemma, because we start off with all multi-edges having bounded norm and only produce multi-edges with bounded norm.

Lemma 4.8. *When Algorithm 3 is run on the (multi-edge) Laplacian of* G'*, arising from splitting edges of* G *into* K *multi-edges, the every edge sample* $\boldsymbol{Y}_{\pi(i),e}$ *satisfies*

$$\left\| \Phi(\boldsymbol{Y}_{\pi(i),e}) \right\| \leq 1/K.$$

As we will see later $K = 200 \log^2 n$ suffices.

4.6 Random Matrix Concentration from Trace Exponentials

Let us recall how matrix-valued variances come into the picture when proving concentration following the strategy from Matrix Bernstein in Lecture 8.

For some matrix-valued random variable $\boldsymbol{X} \in S^n$, we'd like to show $\Pr[\|\boldsymbol{X}\| \leq 0.5]$. Using Markov's inequality, and some observations about matrix exponentials and traces, we saw that for all $\theta > 0$,

$$\Pr[\|\boldsymbol{X}\| \geq 0.5] \leq \exp(-0.5\theta) \left(\mathbb{E}\left[\operatorname{Tr}\left(\exp\left(\theta \boldsymbol{X} \right) \right) \right] + \mathbb{E}\left[\operatorname{Tr}\left(\exp\left(-\theta \boldsymbol{X} \right) \right) \right] \right). \tag{10}$$

We then want to bound $\mathbb{E}\left[\operatorname{Tr}\left(\exp\left(\theta \boldsymbol{X} \right) \right) \right]$ using Lieb's theorem. We can handle $\mathbb{E}\left[\operatorname{Tr}\left(\exp\left(-\theta \boldsymbol{X} \right) \right) \right]$ similarly.

Theorem 4.9 (Lieb)**.** *Let* $f : S^n_{++} \to \mathbb{R}$ *be a matrix function given by*

$$f(\boldsymbol{A}) = \operatorname{Tr}\left(\exp\left(\boldsymbol{H} + \log(\boldsymbol{A}) \right) \right)$$

for some $\boldsymbol{H} \in S^n$*. Then* $-f$ *is convex (i.e.* f *is concave).*

As observed by Tropp, this is useful for proving matrix concentration statements. Combined with Jensen's inequality, it gives that for a random matrix $\boldsymbol{X} \in S^n$ and a fixed $\boldsymbol{H} \in S^n$

$$\mathbb{E}\left[\operatorname{Tr}\left(\exp\left(\boldsymbol{H} + \boldsymbol{X} \right) \right) \right] \leq \operatorname{Tr}\left(\exp\left(\boldsymbol{H} + \log(\mathbb{E}\left[\exp(\boldsymbol{X}) \right]) \right) \right).$$

The next crucial step was to show that it suffices to obtain an upper bound on the matrix $\mathbb{E}\left[\exp(\boldsymbol{X}) \right]$ w.r.t the Loewner order. Using the following three lemmas, this conclusion is an immediate corollary.

Lemma 4.10. *If* $\boldsymbol{A} \preceq \boldsymbol{B}$*, then* $\operatorname{Tr}\left(\exp(\boldsymbol{A}) \right) \leq \operatorname{Tr}\left(\exp(\boldsymbol{B}) \right)$.

Lemma 4.11. *If* $0 \prec \boldsymbol{A} \preceq \boldsymbol{B}$*, then* $\log(\boldsymbol{A}) \preceq \log(\boldsymbol{B})$.

Lemma 4.12. $\log(\boldsymbol{I} + \boldsymbol{A}) \preceq \boldsymbol{A}$ *for* $\boldsymbol{A} \succ -\boldsymbol{I}$.

Corollary 4.13. *For a random matrix* $\boldsymbol{X} \in S^n$ *and a fixed* $\boldsymbol{H} \in S^n$*, if* $\mathbb{E}\left[\exp(\boldsymbol{X}) \right] \preceq \boldsymbol{I} + \boldsymbol{U}$ *where* $\boldsymbol{U} \succ -\boldsymbol{I}$*, then*

$$\mathbb{E}\left[\operatorname{Tr}\left(\exp\left(\boldsymbol{H} + \boldsymbol{X} \right) \right) \right] \leq \operatorname{Tr}\left(\exp\left(\boldsymbol{H} + \boldsymbol{U} \right) \right).$$

4.7 Mean-Exponential Bounds from Variance Bounds

To use Corollary 4.13, we need to construct useful upper bounds on $\mathbb{E}[\exp(\boldsymbol{X})]$. This can be done, starting from the following lemma.

Lemma 4.14. $\exp(\boldsymbol{A}) \preceq \boldsymbol{I} + \boldsymbol{A} + \boldsymbol{A}^2$ *for* $\|\boldsymbol{A}\| \leq 1$.

If $\boldsymbol{X}$ is zero-mean and $\|\boldsymbol{X}\| \leq 1$, this means that $\mathbb{E}[\exp(\boldsymbol{X})] \preceq \boldsymbol{I} + \mathbb{E}\left[\boldsymbol{X}^2\right]$, which is how we end up wanting to bound the matrix-valued variance $\mathbb{E}\left[\boldsymbol{X}^2\right]$. In the rest of this Subsection, we're going to see the matrix-valued variance of the stopped martingale is bounded throughout the algorithm.

Firstly, we note that for a single edge sample $\tilde{\boldsymbol{X}}_{i,e}$, by Lemma 4.8, we have that

$$\left\|\tilde{\boldsymbol{X}}_{i,e}\right\| \leq \left\|\Phi\left(\boldsymbol{Y}_{\pi(i),e} - \mathbb{E}\left[\boldsymbol{Y}_{\pi(i),e}\right]\right)\right\| \leq 1/K,$$

using that $\|\boldsymbol{A} - \boldsymbol{B}\| \leq \max(\|\boldsymbol{A}\|, \|\boldsymbol{B}\|)$, for $\boldsymbol{A}, \boldsymbol{B} \succeq \boldsymbol{0}$, and $\|\mathbb{E}[\boldsymbol{A}]\| \leq \mathbb{E}[\|\boldsymbol{A}\|]$ by Jensen's inequality.

Thus, if $0 < \theta \leq K$, we have that

$$\begin{aligned}\mathbb{E}\left[\exp(\theta\tilde{\boldsymbol{X}}_{i,e}) \mid \text{preceding samples}\right] &\preceq \boldsymbol{I} + \mathbb{E}\left[(\theta\tilde{\boldsymbol{X}}_{i,e})^2 \mid \text{preceding samples}\right] \\ &\preceq \boldsymbol{I} + \frac{1}{K}\theta^2 \cdot \mathbb{E}\left[\Phi(\boldsymbol{Y}_{\pi(i),e}) \mid \text{preceding samples}\right]\end{aligned} \tag{11}$$

4.8 The Overall Mean-Trace-Exponential Bound

We will use $\mathbb{E}_{(<i)}$ to denote expectation over variables preceding the ith elimination step. We are going to refrain from explicitly writing out conditioning in our expectations, but any *inner* expectation that appears inside another *outer* expectation should be taken as conditional on the outer expectation. We are going to use d_i to denote the multi-edge degree of vertex $\pi(i)$ in $\boldsymbol{S}_{i-1}$. This is exactly the number of edge samples in the ith elimination. Note that there is no elimination at step n (the algorithm is already finished). As a notational convenience, let's write $\hat{n} = n - 1$. With all that in mind, we bound the mean-trace-exponential for some parameter $0 < \theta \leq 0.5/\sqrt{K}$

$$\mathbb{E}\,\mathrm{Tr}\left(\exp(\theta\sum_{i=1}^{\hat{n}}\tilde{\boldsymbol{X}}_i)\right) \tag{12}$$

$$= \mathop{\mathbb{E}}_{(<\hat{n})}\mathop{\mathbb{E}}_{\pi(\hat{n})}\mathop{\mathbb{E}}_{\tilde{\boldsymbol{X}}_{\hat{n},1}}\cdots\mathop{\mathbb{E}}_{\tilde{\boldsymbol{X}}_{\hat{n},d_{\hat{n}}-1}}\mathop{\mathbb{E}}_{\tilde{\boldsymbol{X}}_{\hat{n},d_{\hat{n}}}}\mathrm{Tr}\exp\left(\underbrace{\sum_{i=1}^{\hat{n}-1}\theta\tilde{\boldsymbol{X}}_i + \sum_{e=1}^{d_{\hat{n}}-1}\theta\tilde{\boldsymbol{X}}_{\hat{n},e}}_{\boldsymbol{H}} + \theta\tilde{\boldsymbol{X}}_{\hat{n},d_{\hat{n}}}\right)$$

$\tilde{\boldsymbol{X}}_{\hat{n},1}\ldots,\tilde{\boldsymbol{X}}_{\hat{n},d_{\hat{n}}}$ are independent conditional on $(<\hat{n}), \pi(\hat{n})$

$$\leq \mathop{\mathbb{E}}_{(<\hat{n})}\mathop{\mathbb{E}}_{\pi(\hat{n})}\mathop{\mathbb{E}}_{\tilde{\boldsymbol{X}}_{\hat{n},1}}\cdots\mathop{\mathbb{E}}_{\tilde{\boldsymbol{X}}_{\hat{n},d_{\hat{n}}-1}}\mathrm{Tr}\exp\left(\sum_{i=1}^{\hat{n}-1}\theta\tilde{\boldsymbol{X}}_i + \sum_{e=1}^{d_{\hat{n}}-1}\theta\tilde{\boldsymbol{X}}_{\hat{n},e} + \frac{1}{K}\theta^2 \cdot \mathop{\mathbb{E}}_{\tilde{\boldsymbol{X}}_{\hat{n},d_{\hat{n}}}}\Phi(\boldsymbol{Y}_{\pi(\hat{n}),d_{\hat{n}}})\right)$$

By Equation (11) and Corollary 4.13 .

$\vdots$ Repeat for each multi-edge sample $\tilde{\boldsymbol{X}}_{\hat{n},1}\ldots,\tilde{\boldsymbol{X}}_{\hat{n},d_{\hat{n}}-1}$

$$\leq \mathop{\mathbb{E}}_{(<\hat{n})} \mathop{\mathbb{E}}_{\pi(\hat{n})} \operatorname{Tr}\exp\left(\sum_{i=1}^{\hat{n}-1}\theta\tilde{\boldsymbol{X}}_i + \sum_{e=1}^{d_{\hat{n}}}\frac{1}{K}\theta^2 \cdot \mathop{\mathbb{E}}_{\tilde{\boldsymbol{X}}_{\hat{n},e}} \Phi(\boldsymbol{Y}_{\pi(\hat{n}),e})\right)$$

$$= \mathop{\mathbb{E}}_{(<\hat{n})} \mathop{\mathbb{E}}_{\pi(\hat{n})} \operatorname{Tr}\exp\left(\sum_{i=1}^{\hat{n}-1}\theta\tilde{\boldsymbol{X}}_i + \frac{1}{K}\theta^2 \textsc{Clique}(\pi(\hat{n}), \boldsymbol{S}_{\hat{n}-1})\right)$$

To further bound the this quantity, we now need to deal with the random choice of $\pi(\hat{n})$. We'll be able to use this to bound the trace-exponential in a very strong way. From a random matrix perspective, it's the following few steps that give the analysis it's surprising strength.

We can treat $\frac{1}{K}\theta^2\textsc{Clique}(\pi(\hat{n}), \boldsymbol{S}_{\hat{n}-1})$ as a random matrix. It is not zero-mean, but we can still bound the trace-exponential using Corollary 4.13.

We can also bound the expected matrix exponential in that case, using a simple corollary of Lemma 4.14.

Corollary 4.15. $\exp(\boldsymbol{A}) \preceq \boldsymbol{I} + (1+R)\boldsymbol{A}$ *for* $\boldsymbol{0} \preceq \boldsymbol{A}$ *with* $\|\boldsymbol{A}\| \leq R$.

Proof. The conclusion follows after observing that for $\boldsymbol{0} \preceq \boldsymbol{A}$ with $\|\boldsymbol{A}\| \leq R$, we have $\boldsymbol{A}^2 \preceq R\boldsymbol{A}$. We can see this by considering the spectral decomposition of $\boldsymbol{A}$ and dealing with each eigenvalue separately. □

Next, we need a simple structural observation about the cliques created by elimination:

Claim 4.16.

$$\textsc{Clique}(\pi(i), \boldsymbol{S}_i) \preceq \textsc{Star}(\pi(i), \boldsymbol{S}_i) \preceq \boldsymbol{S}_i$$

Proof. The first inequality is immediate from $\textsc{Clique}(\pi(i), \boldsymbol{S}_i) \preceq \textsc{Clique}(\pi(i), \boldsymbol{S}_i) + \boldsymbol{l}_i\boldsymbol{l}_i^\top = \textsc{Star}(\pi(i), \boldsymbol{S}_i)$. The latter inequality $\textsc{Star}(\pi(i), \boldsymbol{S}_i) \preceq \boldsymbol{S}_i$ follows from the star being a subgraph of the whole Laplacian $\boldsymbol{S}_i$. □

Next we make use of the fact that $\tilde{\boldsymbol{X}}_i$ is from the difference sequence of the *stopped* martingale. This means we can assume

$$\boldsymbol{S}_i \preceq 1.5\boldsymbol{L},$$

since otherwise $\tilde{\boldsymbol{X}}_i = \boldsymbol{0}$ and we get an even better bound on the trace-exponential. To make this formal, in Equation (12), we ought to do a case analysis that also includes the case $\tilde{\boldsymbol{X}}_i = \boldsymbol{0}$ when the martingale has stopped, but we omit this.

Thus we can conclude by Claim 4.16 that

$$\|\Phi(\textsc{Clique}(\pi(i), \boldsymbol{S}_i))\| \leq 1.5.$$

By our assumption $0 < \theta \leq 0.5/\sqrt{K}$, we have $\left\|\frac{1}{K}\theta^2\Phi(\textsc{Clique}(\pi(i), \boldsymbol{S}_{i-1}))\right\| \leq 1$, so that by Corollary 4.15,

$$\mathop{\mathbb{E}}_{\pi(i)} \exp\left(\frac{1}{K}\theta^2\Phi(\textsc{Clique}(\pi(i), \boldsymbol{S}_{i-1}))\right) \preceq \boldsymbol{I} + \frac{2}{K}\theta^2 \mathop{\mathbb{E}}_{\pi(i)} \Phi(\textsc{Clique}(\pi(i), \boldsymbol{S}_{i-1})) \tag{13}$$

$$\preceq \boldsymbol{I} + \frac{2}{K}\theta^2 \mathop{\mathbb{E}}_{\pi(i)} \Phi(\textsc{Star}(\pi(i), \boldsymbol{S}_{i-1})) \qquad \text{by Claim 4.16.}$$

Next we observe that, because every multi-edge appears in exactly two stars, and $\pi(i)$ is chosen uniformly at random among the $n+1-i$ vertices that $\boldsymbol{S}_{i-1}$ is supported on, we have

$$\mathop{\mathbb{E}}_{\pi(i)} \textsc{Star}(\pi(i), \boldsymbol{S}_{i-1}) = 2\frac{1}{n+1-i}\boldsymbol{S}_{i-1}.$$

And, since we assume $\boldsymbol{S}_i \preceq 1.5\boldsymbol{L}$, we further get

$$\mathop{\mathbb{E}}_{\pi(i)} \exp\left(\frac{1}{K}\theta^2\Phi(\textsc{Clique}(\pi(i), \boldsymbol{S}_{i-1}))\right) \preceq \boldsymbol{I} + \frac{6\theta^2}{K(n+1-i)}\boldsymbol{I}.$$

We can combine this with Equation (12) and Corollary 4.13 to get

$$\begin{aligned}
&\mathbb{E}\,\mathrm{Tr}\left(\exp(\theta\sum_{i=1}^{\hat{n}}\tilde{\boldsymbol{X}}_i)\right)\\
&\leq \mathop{\mathbb{E}}_{(<\hat{n})}\mathop{\mathbb{E}}_{\pi(\hat{n})}\mathrm{Tr}\exp\left(\sum_{i=1}^{\hat{n}-1}\theta\tilde{\boldsymbol{X}}_i + \frac{1}{K}\theta^2\textsc{Clique}(\pi(\hat{n}), \boldsymbol{S}_{\hat{n}-1})\right)\\
&\leq \mathop{\mathbb{E}}_{(<\hat{n})}\mathrm{Tr}\exp\left(\sum_{i=1}^{\hat{n}-1}\theta\tilde{\boldsymbol{X}}_i + \frac{6\theta^2}{K(n+1-i)}\boldsymbol{I}\right)
\end{aligned}$$

And by repeating this analysis for each term $\tilde{\boldsymbol{X}}_i$, we get

$$\begin{aligned}
\mathbb{E}\,\mathrm{Tr}\left(\exp(\theta\sum_{i=1}^{\hat{n}}\tilde{\boldsymbol{X}}_i)\right) &\leq \mathrm{Tr}\exp\left(\sum_{i=1}^{\hat{n}}\frac{6\theta^2}{K(n+1-i)}\boldsymbol{I}\right)\\
&\leq \mathrm{Tr}\exp\left(\frac{7\theta^2\log(n)}{K}\boldsymbol{I}\right)\\
&= n\exp\left(\frac{7\theta^2\log(n)}{K}\right)
\end{aligned}$$

Then, by choosing $K = 200\log^2 n$ and $\theta = 0.5\sqrt{K}$, we get

$$\exp(-0.5\theta)\,\mathbb{E}\,\mathrm{Tr}\left(\exp(\theta\sum_{i=1}^{\hat{n}}\tilde{\boldsymbol{X}}_i)\right) \leq \exp(-0.5\theta)n\exp\left(\frac{7\theta^2\log(n)}{K}\right) \leq 1/n^5.$$

$\mathbb{E}\,\mathrm{Tr}\left(\exp(-\theta\sum_{i=1}^{\hat{n}}\tilde{\boldsymbol{X}}_i)\right)$ can be bounded by an identical argument, so that Equation (10) gives

$$\Pr\left[\left\|\sum_{i=1}^{\hat{n}}\tilde{\boldsymbol{X}}_i\right\| \geq 0.5\right] \leq 2/n^5.$$

Thus we have established $\left\|\sum_{i=1}^{\hat{n}}\tilde{\boldsymbol{X}}_i\right\| \leq 0.5$ with high probability (Equation (9)), and this in turn implies Equation (2), and finally Equation (1):

$$0.5\boldsymbol{L} \preceq \boldsymbol{\mathcal{L}}\boldsymbol{\mathcal{L}}^\top \preceq 1.5\boldsymbol{L}.$$

Now, all that's left to note is that the running time is linear in the multi-edge degree of the vertex being eliminated in each iteration (and this also bounds the number of non-zero entries being

created in $\boldsymbol{\mathcal{L}}$). The total number of multi-edges left in the remaining graph stays constant at $Km = O(m \log^2 n)$. Thus the expected degree in the ith elimination is $Km/(n+i-1)$, because the remaining number of vertices is $n+i-1$. Hence the total running time and total number of non-zero entries created can both be bounded as

$$Km \sum_i 1/(n+i-1) = O(m \log^3 n).$$

We can further prove that the bound $O(m \log^3 n)$ on running time and number of non-zeros in $\boldsymbol{L}$ holds with high probability (e.g. $1 - 1/n^5$). To show this, we essentially need a scalar Chernoff bound, in except the degrees are in fact not independent, and so we need a scalar martingale concentration result, e.g. Azuma's Inequality. This way, we complete the proof of Theorem 2.4.

References

[KS16] R. Kyng and S. Sachdeva. Approximate gaussian elimination for laplacians - fast, sparse, and simple. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 573–582, 2016.

[ST04] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, STOC '04, page 81–90, New York, NY, USA, 2004. Association for Computing Machinery.

[Tro19] Joel A Tropp. Matrix concentration & computational linear algebra. 2019.