1 Overview

In the last class, we defined the terms cut sparsifier and spectral sparsifier and introduced Roman Vershynin (RV) Lemma which will help us analyse Laplacians of random graphs. In this lecture, we show how to produce spectral sparsifiers with $O(n \log n/\epsilon^2)$ edges where $n$ is the number of nodes in the graph and $\epsilon$ is a measure of the quality of the sparsifier.

2 Background

Definition 1. For $A$ and $B$ symmetric matrices, $A \preceq B$ if $\forall x$, $x^\top A x \leq x^\top B x$

Definition 2. Graph Laplacian
The Laplacian matrix of a weighted graph $G = (V, E, w)$, where $w(u,v)$ is the weight of edge $(u,v)$ is defined by

$$L_G(u,v) = \begin{cases} 
-w(u,v) & \text{if } u \neq v \\
\sum_z w(u,z) & \text{if } u = v
\end{cases}$$

The Laplacian can be expressed in terms of differences of standard basis vectors.

$$L_G = \sum_{e=(u,v)} w_e (e_u - e_v)(e_u - e_v)^\top$$

$$= \sum_{e=(u,v)} w_e u_e u_e^\top$$

where $e_i$ is the standard basis vector such that $(e_i)_j = \delta_{ij}$ and for edge $e = (u,v)$, $u_e = e_u - e_v$.

Definition 3. Spectral Sparsifier
A graph $H = (V, E', w')$ is an $\epsilon$-spectral approximation of a graph $G = (V, E, w)$ if

$$(1 - \epsilon)L_G \preceq L_H \preceq (1 + \epsilon)L_G$$

where $L_G$, $L_H$ are the Laplacians of graphs $G, H$ respectively.

Note that $x^\top L_G x = \sum_{e=(u,v)} w_e (x_u - x_v)^2$ is shift invariant. So in the analysis below, we restrict ourselves to $x$ such that $x^\top 1 = 0$.

Lemma 4. Roman Vershynin Lemma
Let $\{X_i\}_{i=1}^m$ be i.i.d random vectors in $\mathbb{R}^n$, such that each $X_i$ is uniformly bounded

$$\|X_i\|_2 \leq \kappa, \quad \|\mathbb{E}[X_iX_i^\top]\| \leq 1 \quad \forall i \in [m]$$
Then
\[ \mathbb{E} \left[ \left\| \frac{1}{m} \sum_{i=1}^{m} X_i X_i^\top - \mathbb{E}[XX^\top] \right\| \right] \leq \kappa \sqrt{\frac{\log n}{m}} \]

Last class, we proposed the following randomized algorithm for computing a spectral sparsifier.

**Algorithm 1** Generates spectral sparsifier

**Input:** G = (V, E, w). **Output:** H = (V, E', w'), a spectral sparsifier of G

1. for m times do
2. Choose each edge \( e \in E \) with some probability \( p_e \)
3. Add edge \( e \) to \( E' \) with \( w'(e) = \frac{w_e}{mp_e} \)

In expectation, the Laplacian of the graph \( H \) output by the above algorithm is equal to the Laplacian of \( G \). Let \( Y_e = \sqrt{\frac{w_e}{mp_e}} u_e \) and let \( \{Z_i\}_{i=1}^{m} \) be independent random variables where \( Z_i = Y_e \) with probability \( p_e \). Note that \( L_H = \frac{1}{m} \sum_{i=1}^{m} Z_i Z_i^\top \).

\[
\mathbb{E}[L_H] = \mathbb{E} \left[ \frac{1}{m} \sum_{i=1}^{m} Z_i Z_i^\top \right] \\
= \mathbb{E}[Z_i Z_i^\top] \\
= \sum_{e \in E} p_e Y_e Y_e^\top \\
= \sum_{e \in E} w_e u_e u_e^\top \\
= L_G
\]

In the next section we discuss how to choose \( p_e \), a probability distribution over edges in \( G \), that gives us a good spectral sparsifier.

### 3 Spectral Sparsifiers

We start with the simple case of complete graphs, which have a spherical Laplacian, and move to non-complete graphs in Section 3.2. To keep the analysis simple we only consider unweighted graphs.

#### 3.1 Complete Graphs

When \( G \) is a complete graph, the Laplacian \( L_G \) is given by:

\[
L_G = \begin{pmatrix}
    n - 1 & -1 & \cdots & -1 \\
    -1 & n - 1 & \cdots & -1 \\
    \vdots & \vdots & \ddots & \vdots \\
    -1 & -1 & \cdots & n - 1
\end{pmatrix} = nI - 11^\top
\]
where \( \mathbf{1} \) is a vector of all 1’s.

From Definition 3, for \( H \) to be a spectral sparsifier, we need that

\[
(1 - \epsilon)L_G \preceq L_H \preceq (1 + \epsilon)L_G
\]

\( \Leftrightarrow \)

\[
(1 - \epsilon)x^\top L_G x \leq x^\top L_H x \leq (1 + \epsilon)x^\top L_G x \quad \forall x \text{ s.t. } x^\top \mathbf{1} = 0
\]

\( \Leftrightarrow \)

\[
|x^\top (L_H - L_G)x| \leq \epsilon x^\top L_G x \quad \forall x \text{ s.t. } x^\top \mathbf{1} = 0
\]

\( \Leftrightarrow \)

\[
||L_H - L_G||_2 \leq \epsilon n
\]

where the last step follows from the assumption that \( \mathbf{1}^\top x = 0 \) and \( x^\top L_G x = x^\top (nI - \mathbf{1}\mathbf{1}^\top) x = n||x||^2 \).

We now show that when \( m \geq \left( \frac{n \log n}{\epsilon^2} \right) \) and \( p_e \) is uniform over edges, Algorithm 2 outputs an \( \epsilon \)-spectral approximator of \( G \). We have:

\[
p_e = \frac{1}{\binom{n}{2}} = \Theta\left( \frac{1}{n^2} \right)
\]

It is easy to see that the random variables \( \{Z_i\}_{i=1}^m \) are uniformly bounded:

\[
||Z_i||_2 = ||Y_e|| \quad \text{(for some edge } e)\]

\[
= \sqrt{\frac{w_e}{p_e}} \frac{\sqrt{u_e^\top u_e}}{\sqrt{u_e^\top u_e}}
\]

\[
= \sqrt{\frac{1}{p_e}} \sqrt{2}
\]

\[
= \Theta(n)
\]

Also, from before we have that

\[
||E[Z_iZ_i^\top]|| = ||L_G|| = n
\]

Applying RV Lemma on random variables \( X_i = \frac{Z_i}{\sqrt{n}} \), we get

\[
E\left( \frac{1}{n} ||L_H - L_G|| \right) \leq \sqrt{\frac{n \log n}{m}}
\]

Thus

\[
E(||L_H - L_G||) \leq n \sqrt{\frac{n \log n}{m}}
\]

\[
\leq \epsilon n \quad \text{if } m \geq \left( \frac{n \log n}{\epsilon^2} \right)
\]

So for \( m \geq \left( \frac{n \log n}{\epsilon^2} \right) \), we get an \( \epsilon \)-approximate spectral sparsifier of \( G \).

### 3.2 Non-complete Graphs

For non-complete graphs, there are two issues that we need to deal with.
1. The Laplacian of a non-complete graph need not be spherical. So in order to apply RV lemma, rather than looking at random variables \( \{Z_i\}_{i=1}^m \), we look at transformed random variables \( \{AZ_i\}_{i=1}^m \) for some matrix \( A \).

2. Need to find a better sampling distribution, \( p_e \). For example, in the case of a barbell graph, we need to return the middle edge to get a good sparsifier.

Consider an unweighted graph \( G \). Let \( U \in \mathbb{R}^{|E| \times n} \) be matrix representing \( |E| \) edges where

\[
U = \begin{pmatrix}
- & u_1^\top & - \\
- & u_2^\top & - \\
& \vdots & \\
- & u_{|E|}^\top & -
\end{pmatrix}
\]

Then the Laplacian for \( G \) can be represented as

\[
L_G = U^\top U
\]

\( L_G \) is symmetric since \( L_G^\top = L_G \). Also we have that \( \forall x, \ x^\top L_G x = x^\top U^\top U x = \|U x\| \geq 0 \) and thus \( L_G \) is positive semi-definite. This implies that all eigenvalues of \( L_G \) are non-negative. Using the eigenvalue decomposition of \( L_G \), we can express \( L_G \) as

\[
L_G = \sum_{i=1}^n \lambda_i b_i b_i^\top
\]

where \( \{b_i\}_{i=1}^n \) are orthonormal eigenvectors and \( \lambda_i \geq 0 \).

Positive powers of \( L_G \) can be calculated by

\[
L_G^p = \sum_{i=1}^n \lambda_i^p b_i b_i^\top
\]

The Moore-Penrose pseudo-inverse of \( L_G \) is given by

\[
L_G^\dagger = \sum_{\lambda_i \neq 0} \lambda_i^{-1} b_i b_i^\top
\]

and thus

\[
(L_G^\dagger)^\frac{1}{2} = \sum_{\lambda_i \neq 0} \lambda_i^{-1/2} b_i b_i^\top
\]

Using these, we can express the projector onto the span of \( L_G \) as

\[
\Pi_{L_G} = \sum_{\lambda_i \neq 0} b_i b_i^\top = (L_G^\dagger)^\frac{1}{2} L_G^\frac{1}{2} = L_G^\frac{1}{2} (L_G^\dagger)^\frac{1}{2}
\]
Note that $\Pi^\top L G \Pi_L G = \Pi L_G$.

For $H$ to be a spectral sparsifier of $G$, we need that

$$x^\top L_H x = (1 \pm \epsilon) x^\top L_G x, \quad \forall x$$

$$\iff x^\top L_H x = (1 \pm \epsilon) x^\top L_G x, \quad \forall x \text{ s.t. } x^\top 1 = 0$$

$$\iff x^\top L_H x = (1 \pm \epsilon) x^\top L_G x, \quad \forall x \in \text{span}(L_G)$$

where the last statement holds when $G$ is a connected graph (because $L_G$ has rank $n - 1$ for a connected graph and $L_G 1 = 0$). Thus for any $x \in \text{span}(L_G)$:

$$x = \Pi_L G x = (L_G^\dagger)^{\frac{1}{2}} L_G^\dagger x = (L_G^\dagger)^{\frac{1}{2}} y$$

where $y = L_G^\frac{1}{2} x$. Then the condition becomes:

$$x^\top L_H x = (1 \pm \epsilon) x^\top L_G x, \quad \forall x \in \text{span}(L_G)$$

$$\iff y^\top L_G^\dagger L_H L_G^\dagger y = (1 \pm \epsilon) y^\top L_G^\dagger L_G^\dagger y$$

$$= (1 \pm \epsilon) y^\top L_G^\dagger L_G^\dagger = (1 \pm \epsilon) y^\top L_G^\dagger \Pi_L G y$$

$$= (1 \pm \epsilon) y^\top \Pi_L G y, \quad \forall y$$

Subtracting $y^\top \Pi_L G y$ from both sides, we get

$$|y^\top (L_G^\dagger L_H L_G^\dagger - \Pi_L G) y| \leq \epsilon y^\top \Pi_L G y$$

$$\iff \|L_G^\dagger L_H L_G^\dagger - \Pi_L G\| \leq \epsilon$$

We now apply RV Lemma on random variables $A_i = L_G^\dagger Z_i$. Let $\kappa = \max \|A_i\|$ and we have:

$$\|E[A_i A_i^\top]\| = \|L_G^\frac{1}{2} E[Z_i Z_i^\top] L_G^\frac{1}{2}\|$$

$$= \|L_G^\frac{1}{2} L_G L_G^\frac{1}{2}\|$$

$$= \|\Pi_L G\|$$

$$\leq 1$$

Applying RV Lemma we get:

$$\|\frac{1}{m} \sum A_i A_i^\top - E[A_i A_i^\top]\| = \|L_G^\frac{1}{2} L_H L_G^\frac{1}{2} - \Pi_L G\| \leq \kappa \sqrt{\frac{\log n}{m}}$$

So if $m \geq (\kappa^2 \log n/\epsilon^2)$, we get an $\epsilon$-approximate sparsifier. Note that we haven’t yet defined the probability distribution $p_e$. $\kappa$ will depend on the choice of $p_e$.

To pick a good probability distribution $p_e$ and to compute $\kappa$, we appeal to physical intuition. Consider the graph to represent nodes on a circuit and let $x \in \mathbb{R}^n$ denote the voltages on each
node. The current flow along edge $e$, denoted by $I_e$, from $u$ to $v$ is related to the voltage drop. Thus $I_e = x_u - x_v = u_e x$. The flow along all edges is given by $I = Ux$ where $I \in \mathbb{R}^{|E|}$.

Given a battery on the circuit, with $x_s$ and $x_t$ fixed at some voltages, we can calculate the rest of the internal voltages $x_v$ using Kirchoff’s Laws:

$$\text{current into vertex} - \text{current out of vertex} = \text{external flow}$$

We know that the external flow is: $(\Delta \text{ at } s, -\Delta \text{ at } t, 0 \text{ elsewhere})$

For node $v$,

$$
(I_{ext})_v = \sum_{e=(u,v)} I_e - \sum_{e=(v,u)} I_e \\
= \sum_e I_e(U_e)_v \\
= (U^\top I)_v \\
= (U^\top Ux)_v \\
= (L_G x)_v
$$

We know the external flow and want voltages so we compute $x = L_G^\dagger I_{ext}$ where $L_G^\dagger$ is the pseudoinverse.

If we set $I_{ext}$ to $u_e$ for some edge $(u, v)$ to indicate that 1 unit of current is pushed from $u$ to $v$, then $L_G^\dagger u_e$ is a vector of all voltages in the circuit and thus $u_e^\top L_G^\dagger u_e$ is the voltage drop from $u$ to $v$.

From Ohm’s Law, we know that $V = IR_{eff}$ where $R_{eff}$ is the effective resistance. Since we have 1 unit of current, we conclude that $R_{eff} = u_e^\top L_G^\dagger u_e$.

We use this fact in our calculation of $\|A_i\|$.

$$
\|A_i\|^2 = A_i^\top A_i \\
= Z_i^\top L_G^{1\frac{1}{2}} L_G^{1\frac{1}{2}} Z_i \\
= Z_i^\top L_G^{1\frac{1}{2}} Z_i \\
= \frac{1}{p_e} u_e^\top L_G^\dagger u_e \\
= \frac{R_{eff}}{p_e}
$$

This suggests to set $p_e \propto R_{eff}$ and after normalizing

$$
p_e = \frac{R_{eff}(e)}{\sum R_{eff}(e)}
$$

Thus $\kappa^2 = \sum R_{eff}(e)$ and we need $m \geq \left((\sum R_{eff}(e)) \log n/\epsilon^2 \right)$ to get a $\epsilon$--approximate sparsifier.

And finally to compute $\sum R_{eff}(e)$, we use Foster’s Theorem.
Theorem 5. Foster’s Theorem
Let $R_{\text{eff}}(e)$ denote the effective resistance along edge $e$ on a connected graph of $n$ nodes. Then

$$\sum_{e \in E} R_{\text{eff}}(e) = n - 1$$

Proof. Define $P = UL_G^\dagger U^\top$. Then

$$P^2 = UL_G^\dagger L_G L_G^\dagger U^\top = P$$

Thus $P$ is a projection matrix, and all its eigenvalues $\lambda_i \in \{0, 1\}$. Since $L_G$ has rank $n - 1$ and $P$ has the same rank as $L_G$, $n - 1$ eigenvalues of $P$ are equal to 1 and the rest are 0. From the definition of effective resistance we have:

$$R_{\text{eff}}(e) = u_e^\top L_G^\dagger u_e = P_{e,e}$$

$$\Rightarrow \sum R_{\text{eff}}(e) = \text{tr}(P) = \sum \lambda_i = n - 1$$

Finally, we conclude that we need $m \geq (n \log n/\epsilon^2)$ and complete the proof for non-complete graphs.

References


