1 Graph Laplacians

High-level motivation: We saw dimension reduction for \( \ell_2 \) (the JL-lemma). What is the analogue for graphs (and combinatorial objects in general)? The idea is to find a sparse graph \( G' \) that is “similar” to \( G \), either (1) in the sense of cuts in the graph, or (2) viewing a graph as a real matrix (i.e., a linear operator).

Graph Laplacians: Let \( G = (V, E, w) \) be an undirected graph with edge weights \( w_e \geq 0 \), where \( w_{ij} = 0 \) effectively means that \( ij \notin E \). As usual, it is illustrative to think of the unit-weight case.

Notation: Assume \( V = \{1, \ldots, n\} \) and let \( e_i \in \mathbb{R}^n \) be the \( i \)-th standard basis vector. For an edge \( uv \in E \), define

\[
\begin{align*}
z_{uv} &:= e_u - e_v \in \mathbb{R}^n \\
Z_{uv} &:= z_{uv} z_{uv}^\top \in \mathbb{R}^{n \times n}.
\end{align*}
\]

Remark: \( z_{uv} = -z_{vu} \) but \( Z_{uv} = Z_{vu} \).

Definition: The Laplacian matrix of \( G \) is the matrix

\[
L_G := \sum_{uv \in E} w_{uv} Z_{uv} \in \mathbb{R}^{n \times n}.
\]

Alternative definition: Then \( L_G \) is the matrix with diagonal entries \( (L_G)_{ii} = d_i \), and off-diagonal entries \( (L_G)_{ij} = -w_{ij} \).

Fact 1: The matrix \( L = L_G \) is symmetric, non-diagonals entries are \( L_{ij} = -w_{ij} \), and its diagonal entries are \( L_{ii} = d_i \), where \( d_i = \sum_{j:ij \in E} w_{ij} \) is the degree of vertex \( i \).

It is useful to put these values in a diagonal matrix \( D = \text{diag}(d) \). If \( G \) is unweighted, then \( L = D - A \) where \( A \) is the adjacency matrix.
2 Basics of Symmetric Matrices

The Spectral Theorem: Every symmetric matrix $M \in \mathbb{R}^{n \times n}$ can be written as

$$M = U \Lambda U^T,$$

where $\Lambda$ is a diagonal matrix and $U$ is an orthogonal matrix (i.e., $UU^T = I$). This is called the spectral decomposition of $M$. Denoting the $i$-th column of $U$ by $u_i \in \mathbb{R}^n$, we get that $\{u_1, \ldots, u_n\}$ is an orthonormal basis consisting of the eigenvectors of $M$, each associated with the eigenvalue $\lambda_i = \Lambda_{ii}$, and we can rewrite the above as

$$M = \sum_{i=1}^n \lambda_i u_i u_i^T.$$

PSD matrices: A symmetric matrix $M \in \mathbb{R}^{n \times n}$ is called positive semidefinite (PSD) if it can be written as $M = BB^T$. This is equivalent to requiring that all eigenvalues of $M$ are non-negative, and also equivalent to requiring that

$$\forall x \in \mathbb{R}^n, \quad x^T M x \geq 0.$$

Exer: Show that every Symmetric Diagonally Dominant (SDD) matrix $M$ (defined as $M_{ii} \geq \sum_{j \neq i} |M_{ij}|$ for all $i$) is PSD.

Fact 2: For every graph $G$, the Laplacian matrix $L_G$ is PSD. Moreover, the number of nonzero eigenvalues of $L_G$ (equivalently, rank($L_G$) = $n - 1$), is exactly $n$ minus the number of connected components in $G$. Thus, $G$ is connected if and only if $L_G$ has $n - 1$ nonzero eigenvalues.

Proof: For every $x \in \mathbb{R}^n$,

$$x^T L_G x = \sum_{uv \in E} w_{uv} (x^T Z_{uv} x) = \sum_{uv \in E} w_{uv} (z_{uv}^T x)^2 = \sum_{uv \in E} w_{uv} (x_u - x_v)^2 \geq 0.$$

We leave the second part as an exercise, and just observe that for $x = \vec{1}$, the above expression is 0, and thus we always have an eigenvalue $\lambda = 0$, i.e., rank($L_G$) $\leq n - 1$.

3 Spectral Sparsifiers

Definition: A $(1 \pm \varepsilon)$-spectral sparsifier of a graph $G = (V, E, w)$ is a graph $G' = (V, E', w')$ (on the same vertex set) such that

$$\forall x \in \mathbb{R}^n, \quad x^T L_{G'} x \in (1 \pm \varepsilon) x^T L_G x.$$  \hspace{1cm} (2)

Theorem 3 [Spielman-Srivastava, 2008]: For every $\varepsilon \in (0, 1/2)$, every $n$-vertex graph $G = (V, E, w)$ has a $(1 \pm \varepsilon)$-spectral sparsifier $G'$ with $|E'| = O(\varepsilon^{-2} n \log n)$ edges. Moreover, $G'$ is a reweighted subgraph of $G$, and it can be computed in randomized polynomial time (given $G$ and $\varepsilon$ as input).
Remarks:

(1) This theorem improves [Spielman-Teng, 2004] and [Benczur-Karger, 1996]. It was later improved by removing the log \(n\) factor in sparsity, which is the optimal bound [Batson-Spielman-Srivastava].

(2) We will focus on the existence of \(G'\); a randomized polynomial-time algorithms is quite straightforward, and with more effort the runtime can be further improved to near-linear.

(3) We assume WLOG that \(G\) is connected.

**Proposition 4:** Suppose \(G'\) is a \((1 \pm \varepsilon)\)-spectral sparsifier of \(G\), and denote the weight of a cut \((S, \bar{S})\) by \(w(S, \bar{S}) := \sum_{uv \in E: u \in S, v \in \bar{S}} w_{uv}\) (and similarly for \(G'\)). Then

\[
\forall S \subset V, \quad w'(S, \bar{S}) \in (1 \pm \varepsilon) \cdot w(S, \bar{S}).
\]

(Such a graph \(G'\) is usually called a cut sparsifier.)

**Proof:** Was seen in class by considering 0-1 vectors \(x\).

**Exer:** Suppose \(G'\) is a \((1 \pm \varepsilon)\)-spectral sparsifier of \(G\), and denote the eigenvalues of \(L_G\) by \(\lambda_1 \geq \cdots \geq \lambda_n\), and those of \(L'_{G'}\) by \(\lambda'_1 \geq \cdots \geq \lambda'_n\). Show that

\[
\forall i \in [n], \quad \lambda'_i \in (1 \pm \varepsilon) \lambda_i.
\]

Hint: use the Courant-Fischer (min-max) characterization of eigenvalues.

## 4 Matrix Chernoff

**Löwner ordering:** We write \(A \preceq B\) to denote that \(A\) is PSD. We extend it to a partial ordering between symmetric matrices, defining \(A \preceq B\) if \(A - B \succeq 0\).

Observe that (2) can be written as

\[
(1 - \varepsilon)L_G \preceq L_{G'} \preceq (1 + \varepsilon)L_G.
\]

**Matrix Chernoff bound [Tropp, 2012]:** Let \(X_1, \ldots, X_k\) be independent random \(n \times n\) symmetric matrices. Suppose that

\[
\forall i \in [k], \quad 0 \preceq X_i \preceq I \quad \text{and} \quad \mu \cdot I \preceq \sum_{i=1}^k \mathbb{E}[X_i] \preceq \overline{\mu} \cdot I.
\]

Then for all \(\varepsilon \in [0, 1]\),

\[
\Pr\left[\lambda_{\max}(\sum_{i=1}^k X_i) \geq (1 + \varepsilon)\overline{\mu}\right] \leq n \cdot e^{-\varepsilon^2 \pi / 3},
\]

\[
\Pr\left[\lambda_{\min}(\sum_{i=1}^k X_i) \leq (1 + \varepsilon)\underline{\mu}\right] \leq n \cdot e^{-\varepsilon^2 \pi / 2}.
\]
5 Construction of Spectral Sparsifiers

We prove Theorem 3 using the following algorithm.

**Algorithm SS:**
1. Init $w' = 0$ and $k := 6 \varepsilon^{-2} n \ln n$
2. Viewing $G$ as an electrical network where each edge $e \in E$ has resistance $r_e = 1/w_e$, compute for every edge $e \in E$ its effective resistance $R_{\text{eff}}(e)$
3. For $i = 1, \ldots, k$
4. Pick an edge $e$ at random with probability $p_e := \frac{w_e R_{\text{eff}}(e)}{n}$
5. Increase $w_e'$ by $\frac{1}{k} \frac{1}{p_e} w_e = \frac{n-1}{k R_{\text{eff}}(e)}$
6. Output the graph defined by $w'$, i.e., the Laplacian $L_G' = \sum_{e \in E} w_e' Z_e$, similarly to (2).

Observe that $G'$ is sparse, because $|E'| = k$.

The next lemma shows that this algorithm (step 4) is well-defined. It requires expressing effective resistances explicitly using the Laplacian.

**Lemma 5:** The edge probabilities $p_e$ sum up to 1.

**Expressing effective resistances via Laplacians:** Consider the electrical network corresponding to $G$, i.e., each edge $e \in E$ is resistor with resistance $r_e = 1/w_e$. If we fix the potentials according to some vector $\phi \in \mathbb{R}^n$, then some electrical flow (current) $f$ will go through the resistors, and some will flow in/out of the vertices. Denote by a vector $x \in \mathbb{R}^n$ the flow injected to the vertices (opposite of the excess flow at each vertex). Then for every $u \in V$ (recall $d_u := \sum_{v \in N(u)} w_{uv}$),

$$x_u = \sum_{v \in N(u)} f_{uv} \quad \text{(KCL)}$$

$$= \sum_{v \in N(u)} \frac{\phi_u - \phi_v}{r_{uv}} \quad \text{(Ohm)}$$

$$= d_v \cdot \phi_v - \sum_{v \in N(u)} w_{vu} \phi_u.$$  

In matrix notation, this is just

$$x = L_G \phi.$$  

It also works in the opposite direction, i.e., if we inject flow $x \in \mathbb{R}^n$ to the vertices, then the vertex potentials will be fixed to $\phi = L_G^{-1} x$ (formally, this should be the pseudo-inverse because $L_G$ is singular, see more below, but we will generally gloss over this issue).

Recall that the effective resistance $R_{\text{eff}}(uv)$ is defined as the potential difference between $u,v \in V$ when shipping one unit of flow from $u$ to $v$, i.e., injecting flow $z_{uv} = e_u - e_v$ (as the vector $x$). Then the vertex potentials are given by $\phi = L_G^{-1} z_{uv}$, and

$$R_{\text{eff}}(uv) = \phi_u - \phi_v = (e_u - e_v)^T \phi = z_{uv}^T L_G^{-1} z_{uv}. \quad (3)$$
Matrix powering and pseudo-inverse: Let $M$ be a symmetric matrix, and recall we can always write it as $M = U\Lambda U^\top$, where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n)$. Given $\alpha \in \mathbb{R}$, we can define the matrix power by essentially powering each eigenvalue separately, i.e.,

$$M^\alpha := U \text{diag}(\lambda_1^\alpha, \ldots, \lambda_n^\alpha) U^\top.$$ 

It clearly generalizes the usual matrix powers (for natural $\alpha$), e.g., $M \cdot M = (U\Lambda U^\top)(U\Lambda U^\top) = U\Lambda^2 U^\top = M^2$.

For us, the really important values of $\alpha$ are $\{-1, 1/2, -1/2\}$. For $\alpha = -1$, the only problem is with zero eigenvalues $\lambda_i = 0$, in which case just we leave them intact (not inverting these eigenvalues). This is called the Moore-Penrose pseudo-inverse, denote $M^\dagger$. Observe that $M$ and $M^\dagger$ have the same kernel.

For $\alpha = 1/2$, we basically restrict attention to PSD matrices, i.e., all $\lambda_i \geq 0$, and then there is no problem. For $\alpha = -1/2$, we combine both, i.e., restrict attention to PSD matrices (e.g., a Laplacian $L_G$), and power only the positive eigenvalues.

Observe that using these definitions, $(L_G^{1/2})^2 = L_G$ and that $L_G^{-1} L_G$ operates like the identity on every $x \perp \overline{1}$.

**Proof of Lemma 5:** Was seen in class using the cyclic property of trace.

**Proof of Theorem 3:** Was seen in class. The basic idea is to use the Matrix Chernoff bound, but since it is “built” for scenarios where the expectation is $\mu I$, we need to rotate/change the basis, achieved by multiplying by $L_G^{-1/2}$. More precisely, we define

$$y_{uv} := L_G^{-1/2} z_{uv},$$

and now claim (as an exercise) that

$$(1 - \varepsilon)L_G \preceq L_G' \preceq (1 + \varepsilon)L_G$$

if and only if (modulo the pseudo-inverse/kernel issue)

$$(1 - \varepsilon)I \preceq L_G^{-1/2} \left( \sum_{e \in E} w'_e z_e z_e^\top \right) L_G^{-1/2} = \sum_{e \in E} w'_e y_e y_e^\top \preceq (1 + \varepsilon)I$$

(we just multiplied from left and right by $L_G^{-1/2}$). We denote the random edge chosen at iteration $i \in [k]$ by $e_i$, and then the matrix of interest can be written as

$$M' = \sum_{e \in E} w'_e y_e y_e^\top = \sum_{i=1}^k \frac{n - 1}{k \cdot R_{\text{eff}}(e_i)} y_{e_i} y_{e_i}^\top.$$  

(5)

To complete the proof of Theorem 3, we bound $M'$ using the matrix Chernoff bound (after checking the conditions).

**Exer:** Explain how to modify the analysis when the sampling loop in steps 3-5 of Algorithm SS is changed to the following: for each edge $e \in E$, repeat $k' = O(\varepsilon^{-2} \log n)$ times, where each repetition increases the weight $w'_e$ (as in step 6) independently with probability $p_e$. 

5
Exer: Show how to modify the algorithm and its analysis to use estimates $\tilde{p}_e$ instead of $p_e$ (e.g., maybe these estimates can be computed very quickly), under the assumption that every $\tilde{p}_e \geq p_e$, and that $\sum_{e \in E} \tilde{p}_e \leq C$.

Hint: you may use the preceding exercise.