Lectures on Algorithmic Spectral Graph Theory

(revised version)

He Sun

August 2017

Contents

1 Basics on Linear Algebra **1**
- 1.1 Vectors . . . 1
- 1.2 Matrices . . . 2
- 1.3 Eigenvalues, and Eigenvectors . . . 3
- 1.4 Courant-Fischer Characterisation of Eigenvalues . . . 5
- 1.5 Positive Definite Matrices . . . 5

2 Graph Spectra **7**
- 2.1 Graph Laplacians . . . 7
- 2.2 Bounding λ_2 . . . 8
- 2.3 Bounding λ_n . . . 12
- 2.4 Examples of Graph Spectra . . . 13
- 2.5 Laplacians for Weighted Graphs . . . 13

3 The Cheeger Inequality **15**
- 3.1 The Cheeger Inequality . . . 16
- 3.2 Proof of Lemma 3.3 . . . 17
- 3.3 Further Discussions . . . 19
- 3.4 Higher-Order Cheeger Inequality . . . 20

4 Quasi-randomness **22**
- 4.1 Vertex Expansion . . . 22
- 4.2 Expander Graphs . . . 23
- 4.3 Expander Mixing Lemma . . . 24

5 Random Walks **27**
- 5.1 Mixing Time of a Random Walk . . . 28
- 5.2 Hitting Time and Cover Time . . . 29

6 Construction of Expander Graphs **33**
6.1 Replacement Product . . . 34
6.2 Zig-Zag Product . . . 35
6.3 Construction of Expanders . . . 38

7 The Power Method for Computing the Eigenvalues **39**
7.1 Computing the Largest Eigenvalue . . . 39
7.2 Computing the Second Largest Eigenvalue . . . 42

8 Graph Clustering **43**
8.1 The Structure Theorem . . . 44
8.2 Spectral Clustering . . . 46
8.3 Linear Time Spectral Clustering Algorithm . . . 48

9 Spectral Sparsification **52**
9.1 Electrical Networks . . . 53
9.2 Spielman-Srivastava Algorithm . . . 54
9.3 Analysis of the Algorithm . . . 55
9.4 Useful Facts . . . 57

10 Linear-Sized Spectral Sparsifiers **58**
10.1 Overview of the Algorithm . . . 59
10.2 The Changes of the Potential Functions . . . 59
10.3 Implementation of the Algorithm . . . 62
10.4 Recent Progress . . . 63
10.5 Useful Facts . . . 63

Lecture 1

Basics on Linear Algebra

In this initial lecture we summarise many useful concepts and facts, some of which will provide the foundation for the discussions in the rest of the lectures. Most of the materials will be familiar to a reader who has had a standard Linear Algebra course, so the materials will be presented quickly without formal proofs.

1.1 Vectors

Throughout the lectures we will consider finite-dimensional vector spaces over the field $\mathbb{C}$ of real numbers. Vectors will be usually represented by symbols u, v, w, x, etc., and scalers by a, b, s, t, etc. The symbol n will mean the dimension of the vector space under consideration.

For any two vectors u, v, the inner product between u and v will be denoted by $\langle u, v\rangle = \sum_{i=1}^{n} u_i \cdot v_i$ and we say that u, v are orthogonal, and we write $u \perp v$, if $\langle u, v\rangle = 0$. We will focus on the space $\mathbb{R}^n$, in which every element is a column vector with n coordinates. In this case, we can write $\langle u, v\rangle = u^\intercal v$, where $\intercal$ denotes the transpose of matrices of any size. Notice that the distinction between column vectors and row vectors is important: as we define $u, v \in \mathbb{R}^n$ as columns vectors, the dimension of $u^\intercal v$ is 1, while the dimension of $uv^\intercal$ is $n \times n$.

For any vector $v \in \mathbb{R}^n$, the ℓ_p-norm of v is defined by $\|v\|_p = \left(\sum_{i=1}^{n} |v_i|^p\right)^{1/p}$. In particular, the ∞-norm of v is defined by $\|v\|_\infty = \max_{1\le i\le n} |v_i|$, and for simplicity we write $\|v\| = \|v\|_2$.

Theorem 1.1 (Hölder inequality)**.** *For any $p, q \in \mathbb{R}$ with $1/p + 1/q = 1$ and $u, v \in \mathbb{R}^n$, it holds that $|\langle u, v\rangle| \le \|u\|_p \cdot \|v\|_q$.*

By setting $p = q = 1/2$, the Hölder inequality gives $|\langle u, v\rangle| \le \|u\| \cdot \|v\|$, i.e., the **Cauchy-Schwarz inequality**.

Theorem 1.2. *For any $v \in \mathbb{R}^n$ and $1 \leq p \leq q < \infty$, it holds that $\|v\|_q \leq \|v\|_p \leq n^{1/p-1/q}\|v\|_q$.*

A **subspace** $S \subseteq \mathbb{R}^n$ is a set of vectors closed under scalar multiplication and addition. A linear combination of vectors is an expression of the form $c_1v_1 + \cdots + c_kv_k$, where $v_1, \ldots, v_k \in \mathbb{R}^n$ and $c_1, \ldots, c_k \in \mathbb{R}$. The set of all linear combinations of $v_1, \ldots, v_k \in \mathbb{R}^n$ forms a subspace, and is denoted by

$$\operatorname{span}\{v_1, \ldots, v_k\} = \{c_1v_1 + \cdots + c_kv_k : c_1, \ldots, c_k \in \mathbb{R}\}.$$

Vectors $v_1, \ldots, v_k \in \mathbb{R}^n$ are said to be **linearly dependent** if there exist scalars $c_1, \ldots, c_k \in \mathbb{R}$, not all zero, such that $c_1v_1 + \cdots + c_kv_k = 0$; otherwise they are called **linearly independent**. If $S = \operatorname{span}\{v_1, \ldots, v_k\}$ and $v_1, \ldots, v_k \in \mathbb{R}^n$ are linearly independent, we say that $\{v_1, \ldots, v_k\}$ is a **basis** for S. Moreover, if the vectors are pairwise orthogonal and have all norm one, they form an **orthonormal basis**. The **dimension** of a subspace S, $\dim S$, is the number of vectors in a basis for S. Finally, we say that two subspaces $S, T \subseteq \mathbb{R}^n$ are **orthogonal**, and write $S \perp T$, if for any $x \in S$ and $y \in T$ we have that $x \perp y$.

1.2 Matrices

The fundamental object used in the course is the notion of matrices. For any $m, n \in \mathbb{N}$, a matrix $A \in \mathbb{R}^{m\times n}$ is an m-by-n array of real numbers $(A_{ij})_{1\leq i\leq m, 1\leq j\leq n}$. For any $A \in \mathbb{R}^{m\times n}$, $A^\intercal \in \mathbb{R}^{n\times m}$ where $A^\intercal_{i,j} = A_{i,j}$.

The **image** of A is

$$\operatorname{im} A = \{y \in \mathbb{R}^m : \exists x \in \mathbb{R}^n \setminus \{0\} \text{ s.t. } Ax = y\},$$

and its **kernel** is $\ker A = \{x \in \mathbb{R}^n : Ax = 0\}$. Notice that $\operatorname{im} A$ is a subspace of $\mathbb{R}^m$, while $\ker A$ is a subspace of $\mathbb{R}^n$. The dimension of the image is called the **rank** of A, and is equal the number of linearly independent columns (or rows) of A, while the dimension of the kernel is called **nullity**. They are related by the **rank-nullity theorem**:

$$\dim \operatorname{im} A + \dim \ker A = n.$$

If A is a square matrix, i.e., $m = n$, then $\ker A = \mathbb{R}^n \setminus \operatorname{im} A$, and kernel and image are orthogonal to each other. If $A = A^\intercal$, we say that A is *symmetric*.

The **trace** of a matrix $A \in \mathbb{R}^{m\times n}$ is defined as $\operatorname{tr}(A) = A_{1,1} + \cdots + A_{p,p}$, where $p = \min\{m, n\}$. The trace is invariant under cyclic permutations, i.e.,

$$\operatorname{tr}(ABCD) = \operatorname{tr}(BCDA) = \operatorname{tr}(CDAB).$$

The **norm** of a matrix A is defined as

$$\|A\| = \sup_{\|x\|=1} \|Ax\|.$$

A matrix $A \in \mathbb{R}^{n\times n}$ is called **invertible** if there is a matrix $B \in \mathbb{R}^{n\times n}$ such that $AB = BA = I_n$, where I_n denotes the $n \times n$ identity matrix. If it is the case, then B is uniquely determined by A, and is called the **inverse** of matrix A, denoted by A^{-1}.

Lemma 1.3. *If $\|A\| < 1$, then $I - A$ is invertible and*

$$(I - A)^{-1} = I + A + A^2 + \cdots,$$

*a convergent power series. This is called the **Neumann Series**.*

Lemma 1.4. *For any matrix A, the series*

$$\exp A = I + A + \frac{A^2}{2!} + \cdots + \frac{A^n}{n!} + \cdots$$

*converges. This is called the **exponential** of A. The matrix $\exp A$ is always invertibvle and*

$$(\exp A)^{-1} = \exp(-A).$$

Conversely, every invertible matrix can be expressed as the exponential of some matrix.

Exercise 1.5. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric, and $x, y \in \mathbb{R}^n$. Then, it holds that $\langle Ax, y\rangle = \langle x, Ay\rangle$.*

1.3 Eigenvalues, and Eigenvectors

A fundamental concept in matrix analysis is the set of eigenvalues of a square matrix.

Definition 1.6. *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix. If $\lambda \in \mathbb{R}$ and $v \in \mathbb{R}^n \setminus \{0\}$ satisfy*

$$Av = \lambda v,$$

*then λ is called an **eigenvalue** of A and v is called an **eigenvector** of A associated with λ. The pair (λ, v) is called an eigenpair of A. The set of A's eigenvalues is called the **spectrum** of A, and is denoted by $\sigma(A)$. The **condition number** of A, denoted by $\tau(A)$, is the ratio between the maximum eigenvalue of A and the minimum eigenvalue of A.*

When A is not symmetric, the eigenvalues of A can be complex-valued, and it is sometimes more useful to consider *singular values* instead of eigenvalues. We say that s is a **singular value** of A if and only if s is the non-negative square root of an eigenvalue of $A^\intercal A$. It can be shown that A and $A^\intercal$ (respectively $A^\intercal A$ and $AA^\intercal$) have the same nonzero singular values (eigenvalues).

The main focus of the lectures is the symmetric matrices, and the **Spectral Theorem** states that matrix A has n eigenvalues, and there exists an orthonormal basis of $\mathbb{R}^n$ that consists of eigenvectors of A.

Theorem 1.7. *Assume that $(\lambda_1, f_1), \ldots, (\lambda_n, f_n)$ are the eigenpairs of matrix $A \in \mathbb{R}^{n\times n}$, and $f_1, \ldots, f_n$ are orthonormal. Then, matrix A can be written as*

$$A = \sum_{i=1}^{n} \lambda_i f_i f_i^\intercal.$$

Based on this theorem, one can generalise a real-valued function to a **matrix-valued function**: let $g : \mathcal{D} \to \mathbb{R}$ be a function with domain $\mathcal{D} \subseteq \mathbb{R}$, and $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigendecomposition $A = \sum_{i=1}^{n} \lambda_i f_i f_i^\intercal$ and $\sigma(A) \subseteq \mathcal{D}$. Then, $g(A)$ is defined as

$$g(A) = \sum_{i=1}^{n} g(\lambda_i) f_i f_i^\intercal.$$

Exercise 1.8. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric, and A^k be the k-th power of A for some $k \in \mathbb{N}$. Then, it holds that*

$$A^k = \sum_{i=1}^{n} \lambda_i^k f_i f_i^\intercal.$$

Exercise 1.9. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric. Then, it holds that*

$$\exp A = \sum_{i=1}^{n} \mathrm{e}^{\lambda_i} f_i f_i^\intercal.$$

It is not difficult to prove that, for any invertible matrix A, one can write A^{-1} as $A^{-1} = \sum_{i=1}^{n} \lambda_i^{-1} f_i f_i^\intercal$, and from this equation it is obvious to see that A^{-1} exists if and only if A has no eigenvalue equal to 0.

Remark 1.1. *A matrix-valued function does not necessarily inherit all the properties from its real-valued analogue. For instance, for any $x, y \in \mathbb{R}$, $\exp(xy) = \exp(x)\exp(y)$, but $\exp(AB) = \exp(A)\exp(B)$ holds if and only if $AB = BA$.*

In later lectures we will study a generalisation of the inverse of matrices with zeroes in their spectrum. For this reason we define the **pseudoinverse** $A^\dagger$, the matrix that has the same kernel of A and acts as its inverse on the image of A. It can be decomposed as

$$A^\dagger = \sum_{\lambda_i \neq 0} \frac{1}{\lambda_i} f_i f_i^\intercal.$$

By definition, the matrix $AA^\dagger = A^\dagger A = \sum_{\lambda_i \neq 0} f_i f_i^\intercal$ can be seen as the projection on the subspace spanned by the eigenvectors corresponding to the nonzero eigenvalues of A, i.e., the image of A. In general, given a subspace S with orthonormal basis $\{v_1, \ldots, v_k\}$, the projection on S is $P_S = \sum_{i=1}^{k} v_i v_i^\intercal$. Observe that P_S acts as the identity matrix on S, and as the all-zeroes matrix on its orthogonal subspace.

Theorem 1.10. *Let $A \in \mathbb{R}^{n\times n}$ be symmetric, and $\lambda_1, \ldots, \lambda_n$ be the eigenvalues of A. Then, it holds that $\operatorname{tr}(A) = \sum_{i=1}^{n} \lambda_i$.*

1.4 Courant-Fischer Characterisation of Eigenvalues

For any matrix $A \in \mathbb{C}^{n\times n}$ and a nonzero vector $v \in \mathbb{C}^n$, the **Rayleigh quotient** of A at v is defined as

$$\frac{v^\intercal Av}{v^\intercal v} \in \mathbb{C}.$$

When v is the eigenvector of A associated with an eigenvalue λ, it is straightforward to check that

$$\frac{v^\intercal Av}{v^\intercal v} = \frac{v^\intercal(\lambda v)}{v^\intercal v} = \lambda. \tag{1.1}$$

Exercise 1.11. *For any symmetric $A \in \mathbb{R}^{n\times n}$ with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$, it holds for any $v \in \mathbb{R}^n$ that*

$$\frac{v^\intercal Av}{v^\intercal v} \in [\lambda_1, \lambda_n].$$

Combining (1.1) with the fact above, the smallest eigenvalue $\lambda_1 = \lambda_{\min}(A)$ and the largest eigenvalue $\lambda_n = \lambda_{\max}(A)$ of a symmetric matrix $A \in \mathbb{R}^{n\times n}$ can be written as

$$\lambda_{\min}(A) = \min_{v\in\mathbb{R}^n, v\neq 0} \frac{v^\intercal Av}{v^\intercal v}, \qquad \lambda_{\max}(A) = \max_{v\in\mathbb{R}^n, v\neq 0} \frac{v^\intercal Av}{v^\intercal v}.$$

Generalising this fact, any eigenvalue of A can be written as a solution of an optimisation problem as follows:

Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$ and corresponding eigenvectors $f_1, \ldots, f_n$. Then, it holds that

$$\lambda_i = \min_{v\perp f_1,\ldots,f_{i-1}, v\neq 0} \frac{v^\intercal Av}{v^\intercal v} = \max_{v\perp f_{i+1},\ldots,f_n, v\neq 0} \frac{v^\intercal Av}{v^\intercal v}. \tag{1.2}$$

Moreover, the corresponding $v \in \mathbb{R}^n$ that minimises or maximises (1.2) above is the eigenvector f_i.

Theorem 1.12 (Courant-Fisher Min-Max Characterisation of Eigenvalues)**.** *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues $\lambda_1 \le \cdots \le \lambda_n$ and corresponding eigenvectors $f_1, \ldots, f_n$. Then, it holds for any $1 \le i \le n$ that*

$$\lambda_i = \min_{S:\dim S=i} \max_{x\in S, x\neq 0} \frac{x^\intercal Ax}{x^\intercal x} = \max_{S:\dim S=n-i+1} \min_{x\in S, x\neq 0} \frac{x^\intercal Ax}{x^\intercal x}, \tag{1.3}$$

where S is a subspace of $\mathbb{R}^n$.

1.5 Positive Definite Matrices

Positive (semi)-definite matrices are the matrices that always have non-negative (or positive) Rayleigh quotients. Formally, any symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive definite if $v^\intercal Av > 0$ for any nonzero $v \in \mathbb{R}^n$, and we write $A \succ 0$ if A is positive

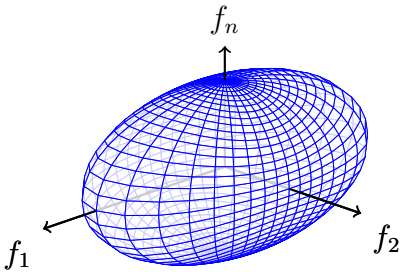

Figure 1.1: Example of an ellipsoid in $\mathbb{R}^n$ defined by a positive definite matrix A.

definitive. Similarly, a symmetric matrix $A \in \mathbb{R}^{n\times n}$ is positive semidefinite if $v^\intercal Av \geq 0$ for any nonzero $v \in \mathbb{R}^n$, and we write $A \succeq 0$ to represent that A is positive semidefinite. The notions $\prec$ and $\preceq$ can be defined in the same way.

The relations $\succ$ and $\succeq$ are generalisations of partial orders defined over $\mathbb{R}$, and it is straightforward to prove the following facts.

Exercise 1.13. *Let $A, B, C \in \mathbb{R}^{n\times n}$ be symmetric matrices. Then the following holds:*

- *If $A \succeq B$, then $A - B \succeq 0$;*
- *If $A \succeq B$, then $A + C \succeq B + C$;*
- *If $A \succeq B$ and $B \succeq C$, then $A \succeq C$.*

For any positive definite matrix $A \in \mathbb{R}^{n\times n}$, one can view A as an ellipsoid in n-dimensional Euclidean space:

$$\mathsf{ellip}(A) = \left\{x \in \mathbb{R}^n : x^\intercal A^{-1} x \leq 1\right\}.$$

That is, the eigenvectors $f_1, \ldots, f_n$ of A are the semi-principal axes of $\mathsf{ellip}(A)$, and the length of the i-th semiaxis f_i is $\lambda_i^{-1/2}$, see Figure 1.1 for illustration. From this perspective, the smaller the value of $\tau(A)$ is, the closer $\mathsf{ellip}(A)$ is to be a sphere in $\mathbb{R}^n$.

Lecture 2

Graph Spectra

Throughout the course let $G = (V, E)$ be an undirected graph with n vertices and m edges. The set of neighbours of vertex u is represented by $N(u)$, and its degree is $d_u = |N(u)|$. For simplicity, we write $u \sim v$ if $\{u, v\}$ is an edge of G. For any set $S \subseteq V$, let $\mathrm{vol}(S) = \sum_{u \in S} d_u$. In particular, let $\mathrm{vol}(G) = \sum_{u \in V[G]} d_u$.

2.1 Graph Laplacians

We first define the graph matrices used in the course. Let $D \in \mathbb{R}^{n \times n}$ be the diagonal matrix where $D_{u,u} = d_u$ for any vertex u. The **adjacency matrix** of graph G is the matrix A defined by $A_{u,v} = 1$ if $u \sim v$, and $A_{u,v} = 0$ otherwise. In particular, we write $A_{u,u} = 1$ if there is a self-loop of vertex u.

Exercise 2.1. *It holds that* $\mathrm{tr}(A^2) = \mathrm{vol}(G)$.

The **Laplacian matrix** of G is defined by $L = D - A$, where A is the **adjacency matrix** of G. The **normalised Laplacian matrix** of G is defined by

$$\mathcal{L} = D^{-1/2} L D^{-1/2} = I - D^{-1/2} A D^{-1/2}.$$

By definition, we know that $\mathcal{L} = I - (1/d) \cdot A$ if G is d-regular. Moreover, one can view $\mathcal{L}$ as an operator on the space of function $g : V(G) \to \mathbb{R}$ such that

$$\mathcal{L}g(u) = \frac{1}{\sqrt{d_u}} \sum_{v:\ u \sim v} \left(\frac{g(u)}{\sqrt{d_u}} - \frac{g(v)}{\sqrt{d_v}} \right).$$

For matrix $\mathcal{L}$, we denote its n eigenvalues with $\lambda_1 \leq \cdots \leq \lambda_n$, with their corresponding orthonormal eigenvectors $f_1, \ldots, f_n$. The set of n eigenvalues $\{\lambda_i\}_{i=1}^n$ together with their multiplicities is called the **spectrum of a graph** G.

Lemma 2.2. *All the eigenvalues of $\mathcal{L}$ are non-negative.*

Proof. Let $g \in \mathbb{R}^n$ be any vector. Then, we have that

$$\begin{aligned}\frac{\langle g, \mathcal{L}g\rangle}{\langle g, g\rangle} &= \frac{\langle g, D^{-1/2}LD^{-1/2}g\rangle}{\langle g, g\rangle} \\ &= \frac{\langle D^{-1/2}g, LD^{-1/2}g\rangle}{\langle g, g\rangle}.\end{aligned}$$

Let $f = D^{-1/2}g$, and we have that

$$\frac{\langle g, \mathcal{L}g\rangle}{\langle g, g\rangle} = \frac{\langle f, Lf\rangle}{\langle D^{1/2}f, D^{1/2}f\rangle}.$$

Notice that

$$\langle f, Lf\rangle = \sum_{u,v} L_{u,v} \cdot f_u f_v = \sum_u d_u \cdot f_u^2 - \sum_{u\sim v} 2 \cdot f_u f_v = \sum_{u\sim v} (f_u - f_v)^2,$$

and therefore

$$\frac{\langle g, \mathcal{L}g\rangle}{\langle g, g\rangle} = \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u \cdot f_u^2} \geq 0.$$

By (1.2), all the eigenvalues are non-negative. Moreover,

$$\frac{\langle g, \mathcal{L}g\rangle}{\langle g, g\rangle} = 0$$

if $f_u = 1$ for every vertex u, i.e., $D^{1/2} \cdot \mathbf{1}$ is an eigenvector of $\mathcal{L}$ with the corresponding eigenvalue 0, where $\mathbf{1} \in \mathbb{R}^n$ is the vector with $\mathbf{1}_u = 1$ for every u. □

Lemma 2.3. *Let G be a connected graph. Then, it holds that $\sum_{i=1}^n \lambda_i = n$.*

Proof. By the definition of $\mathcal{L}$, it holds that $\sum_{i=1}^n \lambda_i = \mathrm{tr}(\mathcal{L}) = n$. □

2.2 Bounding λ_2

Among the n eigenvalues, λ_2 plays a central role in studying various aspects of a graph. Here, we present some simple bounds, and more detailed analysis about λ_2 will be discussed in later lectures.

Theorem 2.4. *It holds that*

$$\lambda_2 = \inf_{f \perp D\mathbf{1}} \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u \cdot f_u^2}.$$

Proof. The statement follows from the proof of Lemma 2.2 and the fact that

$$\langle g, D^{1/2}\mathbf{1}\rangle = 0 \Leftrightarrow \langle D^{1/2}f, D^{1/2}\mathbf{1}\rangle = 0 \Leftrightarrow \langle f, D\mathbf{1}\rangle = 0.$$ □

The following equivalent definitions of λ_2 will be used in our lectures.

Lemma 2.5. *It holds that*

$$\begin{aligned}\lambda_2 &= \inf_f \sup_t \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u \cdot (f_u - t)^2} \\ &= \inf_f \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u \cdot \left(f_u - \bar{f}\right)^2},\end{aligned}$$

where

$$\bar{f} = \frac{\sum_u d_u \cdot f_u}{\mathrm{vol}(G)}.$$

Proof. By direct calculation, we have that

$$\inf_f \sup_t \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u \cdot (f_u - t)^2} = \inf_f \sup_t \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u f_u^2 - 2t\sum_u d_u f_u + t^2 \sum_u d_u}. \tag{2.1}$$

Let us define

$$h(t) = t^2 \sum_u d_u - 2t \cdot \sum_u d_u f_u.$$

Since

$$\sup_t \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u f_u^2 - 2t\sum_u d_u f_u + t^2 \sum_u d_u}$$

is achieved if and only if $h(t)$ achieves its minimum. Notice that

$$h'(t) = 2t \cdot \mathrm{vol}(G) - 2 \cdot \sum_u d_u f_u,$$

and $h'(t) = 0$ iff

$$t = \bar{f} = \frac{\sum_u d_u f_u}{\mathrm{vol}(G)}.$$

Also, since $h''(t) = 2\,\mathrm{vol}(G) > 0$, we know that $h(t)$ achieves its minimum at the point $t = \bar{f}$. Given this, we can rewrite (2.1) as

$$\begin{aligned}\inf_f \sup_t \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u \cdot (f_u - t)^2} &= \inf_f \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u \cdot (f_u - \bar{f})^2} \\ &= \inf_f \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u f_u^2 - 2\bar{f}\sum_u d_u f_u + \sum_u d_u \bar{f}^2} \\ &= \inf_f \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u f_u^2 - (\sum_u d_u f_u)^2/\,\mathrm{vol}(G)}.\end{aligned}$$

Now for any $f \in \mathbb{R}^n$, we assume $\sum_u f_u d_u = \alpha$ for some α, and define vector $g \in \mathbb{R}^n$ such that

$$g_u = f_u - \frac{\alpha}{\mathrm{vol}\, G},$$

which implies that

$$\sum_u g_u d_u = \sum_u f_u d_u - \frac{\alpha}{\mathrm{vol}(G)} \sum_u d_u = 0,$$

i.e., $g \perp D\mathbf{1}$. Since any vector f can be determined by a vector g satisfying $g \perp D\mathbf{1}$ and α, we have that

$$\begin{aligned}
\inf_f \sup_t \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u \cdot (f_u - t)^2} &= \inf_f \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u f_u^2 - (\sum_u d_u f_u)^2/\mathrm{vol}(G)} \\
&= \inf_{g,\alpha} \frac{\sum_{u\sim v}(g_u - g_v)^2}{\sum_u d_u (g_u + \alpha/\mathrm{vol}(G))^2 - \alpha^2/\mathrm{vol}(G)} \\
&= \inf_{g,\alpha} \frac{\sum_{u\sim v}(g_u - g_v)^2}{\sum_u d_u g_u^2} \\
&= \inf_g \frac{\sum_{u\sim v}(g_u - g_v)^2}{\sum_u d_u g_u^2} \\
&= \inf_{f\perp D\mathbf{1}} \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u f_u^2}
\end{aligned}$$

Combining this with Theorem 2.4 proves the statement. □

Exercise 2.6. *It holds that*

$$\lambda_2 = \mathrm{vol}(G) \cdot \inf_f \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_{u,v} d_u d_v \cdot (f_u - f_v)^2},$$

where $\sum_{u,v}$ *denotes the sum of all unordered pairs of vertices* u, v *in* G.

Exercise 2.7. $\lambda_2 \le n/(n-1)$, *and* $\lambda_2 = n/(n-1)$ *iff* G *is a complete graph.*

Lemma 2.8. *If* G *is not a complete graph, then* $\lambda_2 \le 1$.

Proof. Since G is not a complete graph, there are vertices u, v which are not connected by an edge. Define a vector $f \in \mathbb{R}^n$ such that

$$f_w = \begin{cases} d_u & \text{if } w = v, \\ -d_v & \text{if } w = u, \\ 0 & \text{otherwise.} \end{cases}$$

By Theorem 2.4, it holds that

$$\lambda_2 \le \frac{\sum_{u\sim v}(f_u - f_v)^2}{\sum_u d_u \cdot f_u^2} = 1.$$

□

Exercise 2.9. *G has k connected components iff $\lambda_k = 0$ and $\lambda_{k+1} > 0$. In particular, G is connected iff $\lambda_2 > 0$.*

Lemma 2.10. *For any connected graph with diameter α, it holds that*

$$\lambda_2 \geq \frac{1}{\alpha \cdot \mathrm{vol}(G)}.$$

Proof. Let $f \in \mathbb{R}^n$ with $f \perp D\mathbf{1}$ be the vector such that

$$\lambda_2 = \frac{\sum_{u \sim v}(f_u - f_v)^2}{\sum_u d_u \cdot f_u^2}.$$

Let v_0 be the vertex such that $|f_{v_0}| = \max_v |f_v|$. Since $\sum_v f_v \cdot d_v = 0$, there is a vertex u_0 such that $f_{v_0} \cdot f_{u_0} < 0$. Let $\mathcal{P}$ be a shortest path between u_0 and v_0. Hence, it holds that

$$\begin{aligned}\lambda_2 = \frac{\sum_{u \sim v}(f_u - f_v)^2}{\sum_v d_v f_v^2} &\geq \frac{\sum_{\{u,v\} \in \mathcal{P}}(f_u - f_v)^2}{\mathrm{vol}(G) \cdot f_{v_0}^2} \geq \frac{(1/\alpha) \cdot (f_{u_0} - f_{v_0})^2}{\mathrm{vol}(G) \cdot f_{v_0}^2} \\ &\geq \frac{1}{\alpha \cdot \mathrm{vol}(G)}.\end{aligned}$$

□

Corollary 2.11. *For any connected graph G with n vertices, it holds that $\lambda_2 = O(1/n^3)$.*

Proof. The result follow from Lemma 2.10, $\mathrm{vol}(G) = O(n^2)$, and $\alpha = O(n)$ for any connected graph. □

Based on Lemma 2.10, we know that a smaller value of α implies a larger value of λ_2 and, informally, the value of λ_2 is closely related to the expansion property of a graph. While this connection between λ_2 and graph's expansion will become more clear in later lectures, the following result shows that a refined bound of λ_2 can be obtained once we have more information about the graph's expansion properties.

Theorem 2.12. *Let G be a d-regular graph of n vertices. Assume that there is a set P of $\binom{n}{2}$ paths connecting all pairs of vertices such that each path in P has length at most ℓ and each edge of G is contained in at most β paths in P. Then,*

$$\lambda_2 \geq \frac{n}{d\beta\ell}.$$

Proof. Let $f \in \mathbb{R}^n$ be the eigenvector associated with λ_2. Then we have that

$$\lambda_2 = \frac{n}{d} \cdot \frac{\sum_{u \sim v}(f_u - f_v)^2}{\sum_{u,v}(f_u - f_v)^2}. \tag{2.2}$$

For any vertices u, v, let $P(u,v)$ be the path connecting u and v. Then, it holds that

$$(f_u - f_v)^2 \leq |P(u,v)| \cdot \sum_{\substack{e \in P(u,v) \\ e=\{u' \sim v'\}}} (f_{u'} - f_{v'})^2 \leq \ell \cdot \sum_{\substack{e \in P(u,v) \\ e=\{u' \sim v'\}}} (f_{u'} - f_{v'})^2 ,$$

and therefore

$$\frac{1}{\ell} \cdot \sum_{u,v} (f_u - f_v)^2 \leq \sum_{u,v} \sum_{\substack{e \in P(u,v) \\ e=\{u' \sim v'\}}} (f_{u'} - f_{v'})^2 \leq \beta \cdot \sum_{u \sim v} (f_u - f_v)^2,$$

which is equivalent to

$$\sum_{u \sim v} (f_u - f_v)^2 \geq \frac{1}{\beta \cdot \ell} \sum_{u,v} (f_u - f_v)^2.$$

Combining this with (2.2) proves the theorem. □

2.3 Bounding λ_n

Lemma 2.13. $\lambda_n \leq 2$.

Proof. Let $g \in \mathbb{R}^n$ be any vector. Based on the calculations in the proof of Lemma 2.2, it holds that

$$\frac{\langle g, \mathcal{L} g \rangle}{\langle g, g \rangle} = \frac{\sum_{u \sim v} (f_u - f_v)^2}{\sum_u d_u \cdot f_u^2} \leq \frac{\sum_{u \sim v} 2(f_u^2 + f_v^2)}{\sum_u d_u \cdot f_u^2} = \frac{2 \cdot \sum_u d_u \cdot f_u^2}{\sum_u d_u \cdot f_u^2} = 2. \tag{2.3}$$

□

Lemma 2.14. *$\lambda_n = 2$ iff a connected component of G is a non-empty bipartite graph.*

Proof. From (2.3), we know that $\lambda_n = 2$ iff the eigenvector associated with λ_n satisfies

$$\sum_{u \sim v} (f_u - f_v)^2 = \sum_{u \sim v} 2(f_u^2 + f_v^2),$$

i.e., $\sum_{u \sim v} (f_u + f_v)^2 = 0$, which is equivalent to say that

$$f_u = -f_v, \quad \text{for every } u \sim v. \tag{2.4}$$

Since f is an eigenvector, there is at least one coordinate $f_u \neq 0$. Based on this and (2.4), the sign of each $f_v \neq 0$ shows the partition of a connected component of G. □

2.4 Examples of Graph Spectra

We have seen that some of a graph's properties can be obtained by its graph spectrum. To summarise our discussion, let us look at several examples.

Example 2.15. *Let the spectrum of a graph G be*

$$0, \quad 2/3, \quad 2/3, \quad 2/3, \quad 2/3, \quad 2/3, \quad 5/3, \quad 5/3, \quad 5/3, \quad 5/3.$$

From this sequence, we know that (i) G has 10 vertices; (2) G is connected; (3) G is not bipartite.

Example 2.16. *Let the spectrum of a graph G be*

$$0, \quad 0, \quad 0.69, \quad 0.69, \quad 1.5, \quad 1.5, \quad 1.8, \quad 1.8.$$

From this sequence, we know that (i) G has 8 vertices; (2) G is disconnected, and has 2 connected components; (3) none of G's connected component is bipartite.

Finally, we show some examples of the spectra for specific graphs.

Example 2.17. *For the complete graph K_n on n vertices, the eigenvalues are 0 and $n/(n-1)$ with multiplicity $n-1$.*

Example 2.18. *For the star S_n on n vertices, the eigenvalues are 0, 1 with multiplicity $n-2$, and 2.*

Example 2.19. *For the path P_n on n vertices, the eigenvalues are $1-\cos\left(\frac{\pi k}{n-1}\right)$ for $k = 0, 1, \ldots, n-1$.*

Example 2.20. *For the cycle C_n on n vertices, the eigenvalues are $1-\cos\left(\frac{2\pi k}{n}\right)$ for $k = 0, 1, \ldots, n-1$.*

2.5 Laplacians for Weighted Graphs

Finally, we define graph matrices for weighted graphs. Let $G = (V, E, w)$ be an undirected graph with weight function $w : V \times V \to \mathbb{R}$, such that $w(u, v) = w(v, u)$ for any $u \sim v$, and $w(u, v) \geq 0$ for any pair of u, v. Then, the degree d_u of vertex u is defined as $d_u = \sum_{u \sim v} w(u, v)$, and $\mathrm{vol}(G) = \sum_v d_v$. The adjacency matrix of G is defined as $A_{u,v} = w(u, v)$ for any pair of u, v. Then, the Laplacian matrix and the normalised Laplacian matrix are defined in the same way as for an unweighted graph, i.e.,

$$L = D - A,$$
$$\mathcal{L} = I - D^{-1/2} A D^{-1/2}.$$

We can still use the same characterisations for the eigenvalues of $\mathcal{L}$, by taking the weight function into account. For example, we have that

$$\lambda_2 = \inf_{g \perp D^{1/2}\mathbf{1}} \frac{\langle g, \mathcal{L} \rangle}{\langle g, g \rangle} = \inf_{f \perp D\mathbf{1}} \frac{\sum_{u,v} w(u,v)(f_u - f_v)^2}{\sum_v d_v f_v^2}.$$

Lecture 3

The Cheeger Inequality

For any undirected graph $G = (V, E)$ and a set $S \subset V$, let

$$h_G(S) = \frac{|\partial S|}{\min\{\mathrm{vol}(S), \mathrm{vol}(V \setminus S)\}},$$

where $\partial S = E(S, V \setminus S)$ denotes the set of edges with one endpoint in S and the other endpoint in $V \setminus S$. The **Cheeger constant** or the **conductance** of graph G is defined as

$$h_G = \min_S h_G(S).$$

By definition, the set S achieving h_G corresponds to the sparsest cut in G, and finding such set S has many applications in computer science. For example, when analysing a physical network, one can view the servers and the links connecting different servers as the vertices and edges in graph G. Then, a higher value of h_G shows that the underlying network is more reliable, since one have to remove many links to make the network disconnected. Moreover, as the number of edges corresponds to the construction cost, it is desired to construct a network G with higher value of h_G but keeping the number of edges in G as small as possible. For image segmentation, a common approach is to build a graph G based on the RGB values and the pairwise distances among different pixels, and the sparsest cuts in G are used to identify different objects in a picture. While we can formulate a sparse cut in different ways with respect to different settings, one of the simplest formulations is as follows:

Problem 3.1 (**The Sparsest Cut Problem**). *Given an undirected graph $G = (V, E)$ of n vertices as input, find a set $S \subset V$ such that $h_G(S) = h_G$.*

The sparest cut problem is NP-hard, and the current best approximation algorithm achieves an approximation ratio of $O(\sqrt{\log n})$, which is based spectral geometry and

semi-definite programming. Indeed, designing approximation algorithms for the sparsest cut problem is one of the most central problems in approximation algorithms.

In this lecture, we will see how h_G relates to λ_2, which is polynomial-time computable, and design an approximation algorithm for the sparsest cut problem. Then, we will discuss the high-order generalisation of the Cheeger inequality.

3.1 The Cheeger Inequality

We have seen several equivalent formulations for λ_2 from the last lecture. From these formulations, we can write λ_2 as the minimum of a function $g(x)$ over possible $x \in \mathcal{D} \subseteq \mathbb{R}^n$, and the value of $g(x)$ for any $x \in \mathcal{D}$ is an upper bound of λ_2. Now, we use the same method to show that λ_2 can be upper bounded with respect to h_G.

Lemma 3.1. $\lambda_2 \leq 2 \cdot h_G$.

Proof. Let $C = (A, B)$ be the optimal cut that achieves h_G, and let $|C|$ be the number of edges in this cut. We define a vector $x \in \mathbb{R}^n$ such that $x_u = 1/\operatorname{vol}(A)$ if $u \in A$, and $x_u = -1/\operatorname{vol}(B)$ of $u \in B$. Since

$$\langle x, D\mathbf{1} \rangle = \sum_{u \in A} \frac{d_u}{\operatorname{vol}(A)} - \sum_{u \in B} \frac{d_u}{\operatorname{vol}(B)} = 0,$$

it holds that

$$\begin{aligned} \lambda_2 &\leq \frac{\sum_{u \sim v} (x_u - x_v)^2}{\sum_u d_u \cdot x_u^2} = \frac{|C| \cdot (1/\operatorname{vol}(A) + 1/\operatorname{vol}(B))^2}{1/\operatorname{vol}(A) + 1/\operatorname{vol}(B)} \\ &= |C| \cdot \left(\frac{1}{\operatorname{vol}(A)} + \frac{1}{\operatorname{vol}(B)} \right) \leq \frac{2|C|}{\min\{\operatorname{vol}(A), \operatorname{vol}(B)\}} = h_G, \end{aligned}$$

which proves the statement. □

Next, we will show that h_G can be upper bounded with respect to λ_2 as well.

Theorem 3.2 (Cheeger Inequality)**.** *It holds that* $h_G \leq \sqrt{2 \cdot \lambda_2}$.

The core behind the proof of the Cheeger inequality is the following fact, which corresponds to an approximation algorithm for finding a sparse cut. For any vector $y \in \mathbb{R}^n$, we assume that $y_1 \leq \ldots \leq y_n$. For any $t \in \mathbb{R}$, define

$$S_t = \{u : \ y_u \leq t\}.$$

We call these $\{S_t\}_{t=1}^n$ **sweep sets**.

Lemma 3.3. *For any vector y satisfying $y^\intercal D\mathbf{1} = 0$, there is a number t such that*

$$h_G(S_t) \leq \sqrt{2 \cdot \frac{y^\intercal L y}{y^\intercal D y}}.$$

Notice that the vector $y = D^{-1/2} f_2$ satisfies

$$\frac{y^\intercal L y}{y^\intercal D y} = \frac{f_2^\intercal D^{-1/2} L D^{-1/2} f_2}{f_2^\intercal D^{-1/2} D D^{-1/2} f_2} = \frac{f_2^\intercal \mathcal{L} f_2}{f_2^\intercal f_2} = \lambda_2.$$

Hence, based on Lemma 3.3 we have the following algorithm for finding a sparse cut.

Algorithm 1 Algorithm for finding a sparse cut

$f = D^{-1/2} f_2$
Sort all the vertices such that $f(u_1) \leq \ldots \leq f(u_n)$
$t = 0$
$S = \emptyset$
$S^\star = \{u_1\}$
while $t \leq n$ **do**
    $t = t + 1$
    $S = S \cup \{u_t\}$
    if $h_G(S) \leq h_G(S^\star)$ **then** $S^\star = S$
return $S^\star$

Theorem 3.4. *Algorithm 1 returns a set S such that $h_G(S) \leq \sqrt{2\lambda_2}$.*

3.2 Proof of Lemma 3.3

Proof of Lemma 3.3. Let

$$\rho = \frac{y^\intercal L y}{y^\intercal D y} = \frac{\sum_{u \sim v} (y_u - y_v)^2}{\sum_u d_u \cdot y_u^2}.$$

Without loss of generality, we assume that $y_1 \leq \ldots \leq y_n$, and let j be the smallest number such that $\sum_{i \leq j} d_i \geq \mathrm{vol}(G)/2$.

We introduce another vector $z \in \mathbb{R}^n$ such that $z_u = y_u - y_j$, and hence $z_j = 0$. Moreover, it is easy to show that

$$\frac{z^\intercal L z}{z^\intercal D z} = \frac{y^\intercal L y}{y^\intercal D y + \mathrm{vol}(G) \cdot y_j^2} \leq \rho.$$

We further scale vector z such that $z_1^2 + z_n^2 = 1$, and define set

$$V_t = \{u : z_u \leq t\}.$$

Since

$$h_G(V_t) = \frac{|\partial V_t|}{\min\{\mathrm{vol}(V_t), \mathrm{vol}(V \setminus V_t)\}},$$

our goal is to define a distribution on t such that

$$\frac{\mathbb{E}\left[|\partial V_t|\right]}{\mathbb{E}\left[\min\{\mathrm{vol}(V_t), \mathrm{vol}(V \setminus V_t)\}\right]} \leq \sqrt{2\rho}, \tag{3.1}$$

Notice that (3.1) is equivalent to show that $\mathbb{E}\left[\sqrt{2\rho} \cdot \min\left\{\mathrm{vol}(V_t), \mathrm{vol}(V \setminus V_t)\right\} - |\partial V_t|\right] \geq 0$. Therefore, there is a set V' such that $\sqrt{2\rho} \cdot \min\left\{\mathrm{vol}(V'), \mathrm{vol}(V \setminus V')\right\} \geq |\partial V'|$, i.e.,

$$\frac{|\partial V'|}{\min\left\{\mathrm{vol}(V'), \mathrm{vol}(V \setminus V')\right\}} \leq \sqrt{2\rho}.$$

To define such a distribution, we choose t according to the probability density function $2|t|$. Hence, the probability that a value between $[a, b]$ is chosen is

$$\mathbb{P}\left[t \in [a, b]\right] = \int_a^b 2|t|\mathrm{d}t = \mathrm{sgn}(b) \cdot b^2 - \mathrm{sgn}(a) \cdot a^2.$$

Since $z_1^2 + z_n^2 = 1$, we have that

$$\mathbb{P}\left[t \in [z_1, z_n]\right] = \int_{z_1}^{z_n} 2|t|\mathrm{d}t = \mathrm{sgn}(z_n) \cdot z_n^2 - \mathrm{sgn}(z_1) \cdot z_1^2 = 1.$$

So it suffices to analyse $\mathbb{E}\left[\min\left\{\mathrm{vol}\left(V_t\right), \mathrm{vol}(V \setminus V_t)\right\}\right]$ and $\mathbb{E}\left[|\partial V_t|\right]$.

Analysis of $\mathbb{E}\left[\min\left\{\mathrm{vol}\left(V_t\right), \mathrm{vol}(V \setminus V_t)\right\}\right]$. Notice that

$$\mathbb{E}\left[\mathrm{vol}\left(V_t\right)\right] = \sum_u \mathbb{P}\left[z_u \leq t\right] \cdot d_u.$$

By the choice of j, we know that $t < 0$ implies that $\mathrm{vol}(V_t) < \mathrm{vol}(G)/2$, while $t > 0$ implies that $\mathrm{vol}(V \setminus V_t) \leq \mathrm{vol}(G)/2$. Hence, it holds that

$$\begin{aligned}
&\mathbb{E}\left[\min\left\{\mathrm{vol}\left(V_t\right), \mathrm{vol}(V \setminus V_t)\right\}\right] \\
&= \sum_u \mathbb{P}\left[z_u \leq t \text{ and } t < 0\right] \cdot d_u + \sum_u \mathbb{P}\left[z_u > t \text{ and } t \geq 0\right] \cdot d_u \\
&= \sum_{u:\ z_u \leq t} d_u \cdot z_u^2 + \sum_{u:\ z_u > t} d_u \cdot z_u^2 \\
&= z^\intercal D z.
\end{aligned}$$

Analysis of $\mathbb{E}\left[|\partial V_t|\right]$. Notice that an edge $u \sim v$ with $z_u \leq z_v$ is in ∂V_t iff $z_u \leq t$ and $z_v \geq t$. This event occurs with probability

$$\int_{z_u}^{z_v} 2|t|\mathrm{d}t = \mathrm{sgn}(z_v) \cdot z_v^2 - \mathrm{sgn}(z_u) \cdot z_u^2,$$

which equals to $|z_u^2 - z_v^2|$ if $\mathrm{sgn}(z_u) = \mathrm{sgn}(z_v)$, and $z_u^2 + z_v^2$ otherwise. We upper bound both terms by the inequality

$$|z_u^2 - z_v^2| \leq |z_u - z_v| \cdot (|z_u| + |z_v|)\,,$$

and

$$z_u^2 + z_v^2 \leq (z_u - z_v)^2 \leq |z_u - z_v| \cdot (|z_u| + |z_v|)\,.$$

Then, it holds that

$$\begin{aligned}
\mathbb{E}\left[|\partial V_t|\right] &= \sum_{\{u,v\}\in E} \mathbb{P}\left[z_u \leq t \text{ and } z_v > t\right] \\
&\leq \sum_{u\sim v} |z_u - z_v| \cdot (|z_u| + |z_v|) \\
&\leq \sqrt{\sum_{u\sim v} |z_u - z_v|^2} \cdot \sqrt{\sum_{u\sim v} (|z_u| + |z_v|)^2} \\
&\leq \sqrt{z^\intercal L z} \cdot \sqrt{2 \cdot z^\intercal D z},
\end{aligned}$$

where the second inequality follows by the Cauchy-Schwarz inequality. Therefore, we have that

$$\frac{\mathbb{E}\left[|\partial V_t|\right]}{\mathbb{E}\left[\min\{\mathrm{vol}(V_t), \mathrm{vol}(V \setminus V_t)\}\right]} \leq \sqrt{2 \cdot \frac{z^\intercal L z}{z^\intercal D z}} \leq \sqrt{2\rho}.$$

By the averaging argument, there is a set V_t, such that $h_G(V_t) \leq \sqrt{2\rho}$. □

3.3 Further Discussions

Are these inequalities tight? Combining the Cheeger inequality with Lemma 3.1, we have that

$$\lambda_2/2 \leq h_G \leq \sqrt{2 \cdot \lambda_2}. \tag{3.2}$$

The following two examples show that both sides of (3.2) are tight up to a constant factor.

- For a path graph P_n, the Cheeger constant is

  $$\frac{1}{\lceil (n-1)/2 \rceil},$$

  and

  $$\lambda_2 = 1 - \cos\left(\frac{\pi}{n-1}\right) \approx \frac{\pi^2}{2(n-1)^2}.$$

  This shows that the Cheeger inequality is tight up to a constant factor.

- For an n-cube, the Cheeger constant is $2/n$ which is equal to λ_2. Hence, the first inequality in (3.2) is tight within a constant factor as well.

Why is it called the Cheeger inequality? Theorem 3.2 was originally proven by Cheeger in the setting of manifolds. It was shown about 20 years later that the same inequality developed by Cheeger holds for graphs as well, and the proof for graphs essentially follows exactly from the proof for manifolds. However, it is worth pointing out that, the easier direction of (3.2), i.e., $\lambda_2/2 \leq h_G$ does not hold for manifolds.

Graphs in which the sweep cut failed to find a sparse cut. There have been extensive studies about the graphs in which a sweep set algorithm based on f_2 fails to find a sparse cut. As an example, we define a grid graph as follows:

- There are $\sqrt{n}$ rows and $3\sqrt{n}$ columns in the grid, and there is a vertex at the every crossing "point" between a horizontal line segment and a vertical line segment.
- The weight of every edge, except the edges sitting in the middle row, has weight 1.
- The weight of every edge sitting in the middle row has weight $1/\sqrt{n}$.

See Figure 3.1 for example. it is easy to see that the "horizontal cut" crossing the "thin" edges is the sparsest cut, while the output of a sweep set algorithm is the "vertical cut".

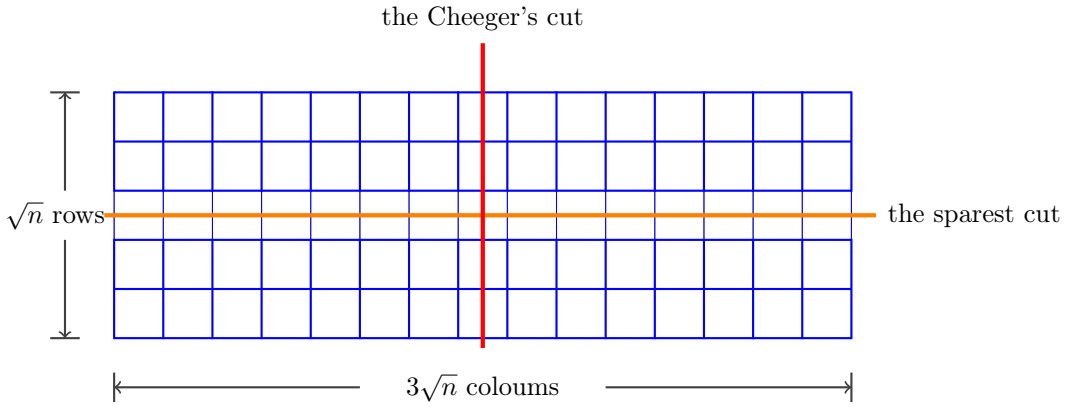

Figure 3.1: A grid graph with $\sqrt{n}$ rows, and $3\sqrt{n}$ columns.

3.4 Higher-Order Cheeger Inequality

So far we have discussed the relations between λ_2 and h_G. Based on this, one can ask if the structure of multi-clusters in a graph relates to the other eigenvalues of $\mathcal{L}$.

To build this connection, we generalise the Cheeger constant and define the k-**way expansion constant** as

$$\rho(k) \triangleq \min_{\text{partition } A_1,\ldots,A_k} \max_{1\le i\le k} h_G(A_i). \tag{3.3}$$

Here, we call subsets of vertices (i.e. **clusters**) $A_1, \ldots, A_k$ a k-**way partition** of G if $A_i \cap A_j = \emptyset$ for different i and j, and $\bigcup_{i=1}^{k} A_i = V$. Usually we say a graph G occurring in practice has k clusters if we can partition the vertex set of G into k subsets $A_1, \ldots, A_k$, such that different clusters are loosely connected, i.e., the value of $\rho(k)$ is small. Lee et al. showed that the value of $\rho(k)$ relates to λ_k, and proved the following higher-order Cheeger inequality:

$$\frac{\lambda_k}{2} \le \rho(k) \le O(k^2)\sqrt{\lambda_k}. \tag{3.4}$$

At a very high level, the proof of the higher-order Cheeger inequality is to apply the eigenvectors associated with $\lambda_2, \ldots, \lambda_k$ to embed every vertex into a point in $\mathbb{R}^k$. We will discuss more about this approach when we discuss spectral clustering algorithms in later lectures.

Lecture 4

Quasi-randomness

From previous discussions we know that high expansion property can be shown by a large value of h_G, or a large value of λ_2. In this lecture we study **vertex expansion**, and see how these different parameters relate each other. Throughout the lecture, let $\Gamma(v) = \{u | u \sim v\}$ be the set of neighbours of v. For any $S \subset V$, let

$$\Gamma(S) = \bigcup_{v \in S} \Gamma(v).$$

We further call $\lambda = \max_{i \geq 2} |1 - \lambda_i|$ the **spectral expansion** of G.

4.1 Vertex Expansion

Definition 4.1 (vertex expansion)**.** *For any graph G with n vertices, we say that G has vertex expansion (K, A) if it holds for any $S \subset V$ with $|S| \leq K$ that*

$$|\Gamma(S)| \geq A \cdot |S|.$$

When $K = n/2$, we call G an A-expander.

Theorem 4.2 (spectral expansion⇒ vertex expansion)**.** *If a d-regular G has spectral expansion λ, then for all α, G has vertex expansion $\left(\alpha n, \frac{1}{(1-\alpha)\lambda^2+\alpha}\right)$.*

Before showing the proof, we introduce some notations at first. For any $x \in \mathbb{R}^n$, we say that x is a **probability distribution** if $x_i \geq 0$ for any i, and $\sum_{i=1}^n x_i = 1$. For any probability distribution x, the support of x is defined by $\text{support}(x) = \{i : x_i > 0\}$.

Definition 4.3. *Given a probability distribution x, the collision probability of x is defined by $\mathsf{CP}(x) = \sum_{i=1}^n x_i^2$.*

Lemma 4.4. *For every probability distribution $x \in [0,1]^n$, we have*

1. $\mathsf{CP}(x) = \|x\|^2 = \|x - \pi\|^2 + \|\pi\|^2$.
2. $\mathsf{CP}(x) \geq 1/|\text{support}(x)|$.

Proof. (1) We write x as $x = \pi + (x - \pi)$ where $\pi \perp (x - \pi)$. Thus

$$\mathsf{CP}(x) = \|x\|^2 = \|x - \pi\|^2 + \|\pi\|^2.$$

(2) By Cauchy-Schwarz inequality, we get

$$1 = \sum_{u \in \text{support}(x)} x_u \leq \sqrt{|\text{support}(x)|} \cdot \sqrt{\sum_u x_u^2}$$

and $\mathsf{CP}(x) = \sum_u x_u^2 \geq 1/|\text{support}(x)|$. □

Proof of Theorem 4.2. Let $|S| \leq \alpha n$. Choose a probability distribution y that is uniform on S and 0 on the $\overline{S}$. Then, it holds that $\mathsf{CP}(y) = 1/|S|$, and

$$\mathsf{CP}(My) \geq 1/|\text{support}(My)| = 1/|\Gamma(S)|,$$

where $M = (1/d) \cdot A$. Therefore $1/|\Gamma(S)| - 1/n \leq \lambda^2(1/|S| - 1/n)$. Combining the formula above and $|S| \leq \alpha n$, we get

$$|\Gamma(S)| \geq \frac{|S|}{(1-\alpha)\lambda^2 + \alpha}.$$ □

Theorem 4.5 (vertex expansion ⇒ spectral expansion)**.** *For every $\delta > 0$ and $d > 0$, there is $\gamma > 0$ such that if G is a d-regular $(1+\delta)$-expander, then it is also $(1-\gamma)$ spectral expander. Specifically, we can take $\gamma = \Omega(\delta^2/d)$.*

4.2 Expander Graphs

From the discussions above, we know that a small value of λ shows that the underlying graph behaves more like a random graph. So a natural question is to study a lower bound of λ. Alon and Boppana showed that, for any constant $\varepsilon > 0$, every sufficiently large d-regular graph has $\lambda \geq 2\sqrt{d-1}/d - \varepsilon$. We call a d-regular graph G **Ramanujan** if $\lambda(G) \leq 2\sqrt{d-1}/d$.

For any fixed constant d, constructing an infinite family of d-regular Ramanujan graph is one of the most challenging questions in algorithmic spectral graph theory. On the other side, it is not difficult to show that a random d-regular graph has good expansion. Here, we show that a random bipartite graph has good vertex expansion.

Let $\mathcal{G}_{d,n}$ be the set of bipartite graphs with bipartite sets L, R of size n and left degree d. We have the following result:

Theorem 4.6. *For any d, there exists an $\alpha(d) > 0$, such that for all n*

$$\mathbb{P}[G \text{ is an } (\alpha n, d-2)\text{-expander }] \geq 1/2,$$

where G is chosen uniformly from $\mathcal{G}_{d,n}$.

Proof. Let p_k be the probability that there is set $S \subset L$ of size k such that $|\Gamma(S)| < (d-2)|S|$. Hence, G is not an $(\alpha n, d-2)$-expander iff $\sum_k p_k > 0$.

Assume that there is a set S of size k and $|\Gamma(S)| < (d-2)|S|$. Then there are at least $2k$ repeats among all the neighbours of vertices in S. By direct calculation it holds that the probability

$$\mathbb{P}[\text{at least } 2k \text{ repeats among all the neighbors of vertices in } S] \leq \binom{dk}{2k}\left(\frac{dk}{n}\right)^{2k}.$$

Therefore

$$\begin{aligned} p_k &\leq \binom{n}{k}\binom{dk}{2k}\left(\frac{dk}{n}\right)^{2k} \\ &\leq \left(\frac{n\mathrm{e}}{k}\right)^{k} \cdot \left(\frac{dk\mathrm{e}}{2k}\right)^{2k} \cdot \left(\frac{dk}{n}\right)^{2k} \\ &= \left(\frac{cd^4k}{n}\right)^{k} \end{aligned}$$

where $c = \mathrm{e}^3$. By setting $\alpha = 1/(cd^4)$ and $k \leq \alpha n$, we know that $p_k \leq 4^{-k}$ and

$$\mathbb{P}[G \text{ is not an } (\alpha n, d-2)\text{-expander}] \leq \sum_{k=1}^{\alpha n} p_k \leq \sum_{k=1}^{\alpha n} 4^{-k} \leq 1/2.$$

□

Theorem 4.7. *For any fixed $d \geq 3$, with high probability a random d-regular graph is an $(\Omega(n), d - 1.01)$-expander. Moreover, this probability tends to 1 as $n \to \infty$.*

4.3 Expander Mixing Lemma

Theorem 4.8 (Expander Mixing Lemma)**.** *Let G be a graph, and $X, Y \subset V$ be sets of vertices. Then, it holds that*

$$\left| |E(X,Y)| - \frac{\operatorname{vol} X \cdot \operatorname{vol} Y}{\operatorname{vol} G} \right| \leq \lambda \cdot \sqrt{\operatorname{vol} X \operatorname{vol} Y}, \tag{4.1}$$

where $\lambda = \max_{i \geq 2} |1 - \lambda_i|$, and $\operatorname{vol} \bar{X} = \operatorname{vol}(V \setminus X)$.

Notice that $\operatorname{vol} X \cdot \operatorname{vol} Y / \operatorname{vol} G$ is the expected value of $|E(X,Y)|$ in a random graph of edge density $\operatorname{vol} G/n^2$. Hence, the left side of (4.1) is the difference between $|E(X,Y)|$ and its expected value in a random graph. So, a smaller value of λ shows that the graph is more close to be a random graph. Before proving the expander mixing lemma, we look at its applications in analysing combinatorial properties of a graph.

Corollary 4.9. *The volume of any independent set in G is at most $\lambda \cdot \operatorname{vol} G$.*

Proof. Let $X = Y$ be an independent set of G. Then $|E(X,X)| = 0$. We apply Theorem 4.8 and obtain that

$$\left| \frac{(\operatorname{vol} X)^2}{\operatorname{vol} G} \right| \leq \lambda \cdot \operatorname{vol} X,$$

which implies that $\operatorname{vol} X \leq \lambda \cdot \operatorname{vol} G$. Hence, the number of any independent set in G is at most $\lambda \cdot \operatorname{vol} G$. For the case of regular graphs, the number of vertices in an independent set is at most $\lambda \cdot n$. □

Corollary 4.10. *Let G be a regular graph. Then the chromatic number of G is at least $1/\lambda$.*

Proof. Let $c : V \to \{1, \ldots, k\}$ be a colouring of G. Then, for every $1 \leq i \leq k$, $c^{-1}(i)$ is an independent set. Since the number of vertices in an independent set is at most λn, the chromatic number if at least $1/\lambda$. □

Theorem 4.11 (Expander Mixing Lemma, Stronger Version)**.** *Let G be a graph, and $X, Y \subset V$ be sets of vertices. Then, it holds that*

$$\left| |E(X,Y)| - \frac{\operatorname{vol} X \cdot \operatorname{vol} Y}{\operatorname{vol} G} \right| \leq \lambda \cdot \frac{\sqrt{\operatorname{vol} X \operatorname{vol} \bar{X} \operatorname{vol} Y \operatorname{vol} \bar{Y}}}{\operatorname{vol}(G)},$$

where $\lambda = \max_{i \geq 2} |1 - \lambda_i|$, and $\operatorname{vol} \bar{X} = \operatorname{vol}(V \setminus X)$.

Proof. For any set $S \subset V$, let χ_S be the indicator vector of set S, i.e. $\chi_S(u) = 1$ if $u \in S$, and $\chi_S(u) = 0$ otherwise. Then, we have that

$$|E(X,Y)| = \langle \chi_X, A\chi_Y \rangle = \chi_X^\intercal D^{1/2}(I - \mathcal{L})D^{1/2}\chi_Y.$$

Without loss of generality, we write

$$D^{1/2}\chi_X = \sum_{i=1}^{n} a_i f_i$$

and

$$D^{1/2}\chi_Y = \sum_{i=1}^{n} b_i f_i,$$

where $f_1, \ldots, f_n$ are orthonormal eigenvectors of $\mathcal{L}$, and $f_0 = D^{1/2}\mathbf{1}/\sqrt{\operatorname{vol} G}$. Hence, we have that $a_1 = \operatorname{vol} X/\sqrt{\operatorname{vol} G}$ and $b_1 = \operatorname{vol} Y/\sqrt{\operatorname{vol} G}$. With this, it holds that

$$\begin{aligned}
\left||E(X,Y)| - \frac{\operatorname{vol} X \cdot \operatorname{vol} Y}{\operatorname{vol} G}\right| &= \left|\chi_X^\intercal D^{1/2}(I-\mathcal{L})D^{1/2}\chi_Y - a_1 b_1\right| \\
&= \left|\left(\sum_{i=1}^n a_i f_i\right)(I-\mathcal{L})\left(\sum_{i=1}^n b_i f_i\right) - a_1 b_1\right| \\
&= \left|\sum_{i=2}^n (1-\lambda_i)\cdot a_i b_i\right| \\
&\leq \lambda \sqrt{\sum_{i=2}^n a_i^2}\sqrt{\sum_{i=2}^n b_i^2} \\
&\leq \lambda \cdot \frac{\sqrt{\operatorname{vol} X \operatorname{vol} \bar{X} \operatorname{vol} Y \operatorname{vol} \bar{Y}}}{\operatorname{vol}(G)},
\end{aligned}$$

where the second last inequality follows by the Cauchy-Schwarz inequality, and the last inequality follows by the fact that

$$\sum_{i=1}^n a_i^2 = \left(\sum_{i=1}^n a_i f_i^\intercal\right)\left(\sum_{i=1}^n a_i f_i\right) = \langle \chi_X^\intercal D^{1/2}, D^{1/2}\chi_X\rangle = \operatorname{vol} X,$$

and

$$\sum_{i=2}^n a_i^2 = \operatorname{vol} X - \frac{(\operatorname{vol} X)^2}{\operatorname{vol} G} = \operatorname{vol} X \operatorname{vol} \bar{X}/\operatorname{vol} G. \qquad \square$$

Lecture 5

Random Walks

A random walk in a graph is a sequence of vertices

$$v_0, v_1, \cdots, v_\ell, \ldots,$$

where v_0 is fixed and every v_i is chosen uniformly at random from the neighbours of v_{i-1} for any $i \geq 1$. This basic random process arises in modelling many problems in mathematics and physics, and has numerous applications in computer science. The classical theory of random walks studies the behaviours of this random process in infinite graphs. For example, Pólya proved the following result in 1921:

Theorem 5.1. *Consider a random walk on an infinite D-dimensional grid. If $D = 2$, then with probability 1, the walk returns to the starting point an infinite number of times. If $D > 2$, with probability 1, the walk returns to the starting point only a finite number of times.*

There are also a lot of studies focusing on the random walks in finite graphs, and the results developed there have many applications in studying computational complexity and designing fast randomised and distributed algorithms.

The goal of this lecture is to give a basic introduction to random walk theory. For simplicity we assume that $G = (V, E)$ is an undirected, unweighted, and d-regular graph. Recall that $M = (1/d) \cdot A$ is the normalised adjacency matrix which will be transition matrix of the random walk on G. Also, notice that the orthonormal eigenvectors $f_1, \ldots, f_n$ of $\mathcal{L}$ are the eigenvectors of M, and the eigenvalue associated with the eigenvector f_i of M is $1 - \lambda_i$.

5.1 Mixing Time of a Random Walk

Definition 5.2 (Mixing Time)**.** *The mixing time of a graph G with n vertices is the minimum t such that for any starting probability distribution x, it holds that*

$$\|M^t x - \pi\|_\infty \leq \frac{1}{2n},$$

where $\pi \in \mathbb{R}^n$ is the stationary distribution, i.e., $\pi_v = d_v / \operatorname{vol}(G)$.

Lemma 5.3. *The mixing time of a random walk in G is at most*

$$t = O\left(\frac{\log n}{\log(1/\lambda)}\right) = O\left(\frac{\log n}{1-\lambda}\right),$$

where λ is the spectral expansion of G.

Proof. By definition, we have that $\pi = \frac{1}{\sqrt{n}} \cdot f_1$, and we can write x as

$$x = \sum_{i=1}^{n} \alpha_i f_i.$$

Since x is a probability vector and all $f_i, i \geq 2$ are orthogonal to π, it holds that $\alpha_1 = 1/\sqrt{n}$, and

$$\begin{aligned}
\|Mx - \pi\|^2 &= \left\| M\left(\sum_{i=1}^{n} \alpha_i f_i\right) - \pi \right\|^2 \\
&= \left\| \pi + \sum_{i=2}^{n} \alpha_i (1-\lambda_i) f_i - \pi \right\|^2 \\
&= \left\| \sum_{i=2}^{n} \alpha_i (1-\lambda_i) f_i \right\|^2 \\
&= \sum_{i=2}^{n} \|\alpha_i (1-\lambda_i) f_i\|^2 \\
&\leq \lambda^2 \sum_{i=2}^{n} \|\alpha_i f_i\|^2 \\
&= \lambda^2 \left\| \sum_{i=2}^{n} \alpha_i f_i \right\|^2 \\
&= \lambda^2 \|x - \pi\|^2,
\end{aligned}$$

where the first inequality follows by the definition of λ. Hence, it holds that

$$\|Mx - \pi\| \leq \lambda \|x - \pi\|,$$

and by induction we have that

$$\left\|M^t x - \pi\right\| = \left\|M(M^{t-1}x) - \pi\right\| \leq \lambda \left\|M^{t-1}x - \pi\right\| \leq \lambda^t \|x - \pi\|.$$

Finally, we have that

$$\left\|M^t x - \pi\right\| \leq \lambda^t \cdot \|x - \pi\| \leq \lambda^t \cdot \|x\| \leq \lambda^t \cdot \|x\|_1 = \lambda^t,$$

where the first inequality follows by

$$\|x - \pi\|^2 + \|\pi\|^2 = \|x\|^2,$$

since $x - \pi$ and π are orthogonal, which immediately implies $\|x - \pi\| \leq \|x\|$. Then, the statement follows by the fact that

$$\left\|M^t x - \pi\right\|_\infty \leq \left\|M^t x - \pi\right\| \leq \lambda^t \leq \frac{1}{2n}$$

when $t = O\left(\frac{\log n}{1-\lambda}\right)$. □

5.2 Hitting Time and Cover Time

The mixing time of a random walk measures the minimum number of steps needed so that a random walk visits every vertex with approximately the same probability. Now we analyse the expected time of a random walk to visit a fixed vertex, or all the vertices of an underlying graph. We will show several combinatorial and spectral bounds on these parameters.

Definition 5.4 (Hitting Time)**.** *For any vertices u, v, the hitting time of u, v, denoted by $h_{u,v}$, is the expected number of steps before vertex v is visited, if the random walk starts from vertex u.*

Theorem 5.5. *Let G be an undirected graph. Then, it holds for any vertex u that $h_{u,u} = 2m/d_u$.*

Definition 5.6 (Cover Time)**.** *For any vertex u, the cover time of u, denoted by $\mathsf{Cov}(u)$ is the expected number of steps to visit all the vertices, if the random walk starts from vertex u. Moreover, let*

$$\mathsf{Cov}_G = \max_{u \in V} \mathsf{Cov}(u).$$

Lemma 5.7. *There are regular graphs for which $h_{u,v} \neq h_{v,u}$.*

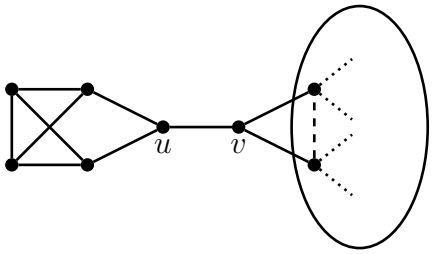

Figure 5.1: Illustration of the graph with an edge $u \sim v$ such that $\mathsf{Hitting}(u, v) = O(1)$, but $\mathsf{Hitting}(v, u) = \Omega(n)$. The ellipse on the right hand side represents a 3-regular graph with $\Omega(n)$ vertices, where the dashed edge has been removed.

Proof. Consider the graph in Figure 5.1 where an edge $u \sim v$ connects a left graph with 5 vertices to a right graph with $n - 5$ vertices. Then, $h_{u,v} = O(1)$, since from every vertex in the left graph there is a path of constant length to reach v. On the other hand,

$$n = h_{u,u} = \frac{1}{3} \cdot \sum_{w \in \partial u} (h_{w,u} + 1) = \frac{1}{3} \cdot (2 \cdot (O(1) + 1) + h_{v,u} + 1)\,,$$

from which it follows that $h_{v,u} = \Omega(n)$. □

Theorem 5.8. *It holds that*

$$\mathsf{Cov}_G \leq 2 \cdot (n-1) \cdot 2|E| \leq 2n \cdot (n-1)^2.$$

Proof. We first observe that for any edge $u \sim v$ that $h_{u,v} < 2|E|$, since $h_{u,u} = n$ implies

$$n = h_{u,u} = \frac{1}{d_u} \sum_{w \in \partial u} (h_{w,u} + 1)\,.$$

Now let $\mathcal{T}$ be any spanning tree of G. Consider a traversal of $\mathcal{T} = (v_0, v_1, \ldots, v_{2n-2} = v_0)$ such that every vertex is visited at least once and every edge is traversed at most twice. Then, the cover time is bounded by the expected time needed for this traversal of $\mathcal{T}$:

$$\mathsf{Cov}_G \leq \sum_{i=0}^{2n-3} h_{v_i, v_{i+1}} \leq (2n-2) \cdot 2|E|.$$ □

An example for a graph with a cover time of $\Omega(n^3)$ is the so-called Lollipop graph which consists of a clique of $(2/3)n$ vertices and a path of length $n/3$ attached to it. In fact, it can be shown that for this graph the cover time is $(4/27 + o(1))n^3$ and moreover, the cover time is also always upper bounded by this value.

The next theorem connects the maximum hitting time to the cover time. It implies that the maximum hitting time approximates the cover time up to logarithmic factors.

Theorem 5.9. *For any graph G, it holds that*

$$\mathsf{Cov}_G = 6 \max_{u,v} h_{u,v} \cdot \log n + 2n.$$

Proof. We divide the random walk of length $6 \max_{u,v} h_{u,v} \log n$ into $3 \ln n$ epochs each of length $2 \cdot \max_{u,v} h_{u,v}$. By Markov's inequality, the probability that a fixed vertex is not visited within one epoch is at most $1/2$. Hence the probability that a vertex is not visited after $3 \ln n$ epochs is $2^{-3 \log n} = n^{-3}$. Taking the union bound, it follows that all vertices are visited with probability at least $1 - 1/n^2$. If this does not happen, then we use our upper bound of $4|V| \cdot |E| \leq 2n^3$ from Theorem 5.8. Hence, the expected total time to cover all vertices is upper bounded by

$$6 \max_{u,v} h_{u,v} \cdot \log n + \frac{1}{n^2} \cdot 2n^3 = 6 \max_{u,v} h_{u,v} \cdot \log n + 2n. \qquad \square$$

Lemma 5.10. *Let u, v be two vertices in a graph of arbitrary degree sequences. Then any shortest path $\mathcal{P} = (v_0 = u, v_1, \ldots, v_\ell = v)$ between u and v satisfies*

$$\sum_{i=0}^{\ell-1} d_{v_i} \leq 3n.$$

Proof. Let $\mathcal{P}$ be any shortest path between u and v. Every vertex w not lying on the shortest path can only be adjacent to at most three (consecutive) vertices on $\mathcal{P}$. Hence,

$$\sum_{i=0}^{\ell-1} d_{v_i} \leq \ell \cdot 2 + (n - \ell) \cdot 3 \leq 3n. \qquad \square$$

Theorem 5.11. *For any regular graph $G = (V, E)$,*

$$\max_{u,v} h_{u,v} \leq 9n^2.$$

Consequently, $\mathsf{Cov}_G = O(n^2 \log n)$.

Proof. Fix two arbitrary vertices u, v and consider a shortest path $\mathcal{P} = (v_0 = u, v_1, \ldots, v_\ell = v)$ from u and v. Consider two consecutive vertices x and y on $\mathcal{P}$. Let $h^\star_{x,x}$ be the expected time to return to x conditioned on the event that the random walk does not move to y in the first step. Then,

$$n = h_{x,x} \geq \frac{d-1}{d} \cdot \left(h^\star_{x,x} + 1\right),$$

and hence,

$$h^\star_{x,x} \leq 2n.$$

Using this and the fact that the random walk at x moves to y with probability $1/d$, we have that

$$h_{x,y} \leq \left(1 + h^{\star}_{x,x}\right) \cdot d \leq 3n \cdot d.$$

Thus by the triangle inequality,

$$h_{u,v} \leq \sum_{i=0}^{\ell-1} h_{v_i,v_{i+1}} \leq \sum_{i=0}^{\ell-1} 2n \cdot d \leq 3n \cdot 3n = 9n^2. \qquad \square$$

Theorem 5.12. *For any two vertices $u, v \in V$,*

$$h_{u,v} = n \cdot \sum_{k=2}^{n} \frac{1}{\lambda_k} \cdot \left(f_k^2(v) - f_k(u) f_k(v)\right)$$

Based on the theorem above, we will present the following well-known **Random Target Lemma**.

Theorem 5.13 (Random Target Lemma)**.** *For any vertex $u \in V$,*

$$\sum_{v \in V} h_{u,v} = \sum_{k=2}^{n} \frac{n}{\lambda_k}.$$

Notice that the right-hand side of the formula above is independent of the starting vertex u.

Proof. By Theorem 5.12, we have that

$$\begin{aligned}
\sum_{v \in V} h_{u,v} &= n \cdot \sum_{v \in V} \sum_{k=2}^{n} \frac{1}{\lambda_k} \left(f_k^2(v) - f_k(u) f_k(v)\right) \\
&= \sum_{k=2}^{n} \frac{n}{\lambda_k} \cdot \left(\sum_{v \in V} f_k^2(v) - \sum_{v \in V} f_k(u) f_k(v) \right) \\
&= \sum_{k=2}^{n} \frac{n}{\lambda_k},
\end{aligned}$$

which finishes the proof. $\square$

Lecture 6

Construction of Expander Graphs

The main focus of this lecture is **expander graphs**, a family of sparse graphs with high expansion properties. Since the sparsity corresponds to the practical construction cost and high expansion implies high connectivity, expander graphs are originally applied to construct low-cost communication networks. However, over the past two decades expander graphs have been applied in many fields of computer science, including coding theory, distributed computing, randomised algorithms, and computational complexity.

We will present explicit constructions of expander graphs in the lecture. We call a family $\{G_j\}_{j\geq 1}$ of d-regular expander graphs **explicitly constructible** if the construction satisfies the following properties:

1. Every graph G_j can be constructed in time polynomial in the number of vertices n_j in G_j.
2. For any vertex v and index $i \in \{1, \cdots, d\}$, the i-th neighbor of v can be computed in time $\text{poly}(\log n_j, \log d)$.
3. For any vertices v and u, whether u and v are adjacent can be determined in time $\text{poly}(\log n_j)$.

We first introduce some notations. Throughout the lecture, we study d-regular graphs, and say G is an (n, d, λ)-graph, if G has n vertices, and $\max_{i\geq 2}\{|1-\lambda_i|\} \leq \lambda$. We will work with the normalised adjacency matrix M. Recall that $M = I - \mathcal{L}$ if G is d-regular. It is easy to check that $\max_{i\geq 2}\{|1-\lambda_i|\} \leq \lambda$ is equivalent to say that all the non-trivial eigenvalues of M belong to $[-\lambda, \lambda]$.

We also use the **rotation map** to express a graph: for any $v \in V$, we label the edges adjacent to v, and let $v[i]$ be the i-th edge of v. Then the rotation map is the function

$$\text{Rot}_G : V \times [d] \to V \times [d],$$

where $\mathrm{Rot}_G(v,i) = (u,j)$ if v is the j-th neighbour of u and u is the i-th neighbour of v.

Exercise 6.1. *For the graph shown in Figure 6.1, we have* $\mathrm{Rot}(a,3) = (h,1)$.

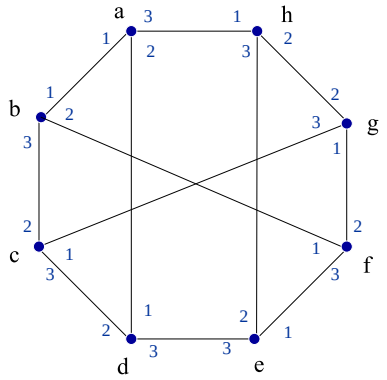

Figure 6.1: Rotation Map

Lemma 6.2. *Let G be an (n,d,λ)-graph. Then G^k, the k-th power of the adjacency matrix of G, is an (n,d^k,λ^k)-graph.*

Lemma 6.3. *Let G be a d-regular graph with rotation map Rot_G. Then G^k is a d^k-regular graph whose rotation map is given by*

$$\mathrm{Rot}_{G^k}(v_0,(a_1,\cdots,a_k)) = (v_k,(b_k,\cdots,b_1)),$$

where the values $b_1,\cdots,b_k$ and v_k are computed via the rule $(v_i,b_i) = \mathrm{Rot}_G(v_{i-1},a_i)$.

6.1 Replacement Product

Let G be a d-regular graph with n vertices, and H be a d'-regular graph with d vertices. Then, **the replacement product** between G and H, denoted by $G \textcircled{r} H$ is a $(d'+1)$-regular graph with $n \cdot d$ vertices, and each vertex in G is replaced by a graph H, called a **cloud**. Formally, we have $\mathrm{Rot}_{G\textcircled{r}H}((u,k),i) = ((v,\ell),j)$ if and only if $u = v$ and $\mathrm{Rot}_H(k,i) = (\ell,j)$, or $i = j = d+1$ and $\mathrm{Rot}_G(u,k) = (v,\ell)$. Figure 6.2 shows an example of the replacement product.

Notice that, given a large (n,d,λ)-graph G and a small (d,d',λ')-graph H, the degree of $G \textcircled{r} H$ only depends on the degree of the smaller graph H, so replacement product is used to reduce the degree of a graph without losing a graph's connectivity. Hence, the replacement product is sometimes used to show that, for some problems studied in graph theory, it is sufficient to solve the problem for a 3-regular connected graph.

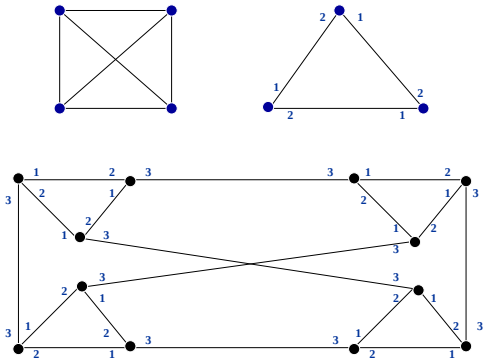

Figure 6.2: An example of the replacement produce.

6.2 Zig-Zag Product

Based on rotation maps, the zig-zag product is defined as follows:

Definition 6.4. *If G is a D-regular graph on $[N]$ with rotation map Rot_G and H is a d-regular graph on $[D]$ with rotation map Rot_H, then their zig-zag product $G \textcircled{z} H$ is defined to be the d^2-regular graph on $[N]\times[D]$ whose rotation map $\mathrm{Rot}_{G\textcircled{z}H}$ is as follows: (1) Let $(a', i') = \mathrm{Rot}_H(a, i)$; (2) Let $(w, b') = \mathrm{Rot}_G(v, a')$; (3) Let $(b, j') = \mathrm{Rot}_H(b', j)$; (4) Output $((w, b), (j', i'))$ as the value of $\mathrm{Rot}_{G\textcircled{z}H}((v, a), (i, j))$.*

Exercise 6.5. *Let G and H be two graphs shown in Figure 6.3. Then*

$$\mathrm{Rot}_{G\textcircled{z}H}((C, z), (1, 2)) = ((F, z), (1, 2)).$$

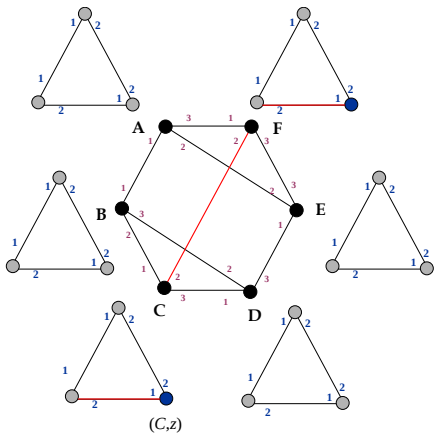

Figure 6.3: An example of the zig-zag product

Notice that, comparing with the replacement product, the label of each edge in $\mathrm{Rot}_{G\textcircled{z}H}$ is a pair of the edge labels in the small graph. Moreover, taking a zig-zig product of a large graph with a small graph, the resulting graph roughly inherits its

size from the large graph, its degree from the small graph, and its expansion properties from the both graphs. This will be the key to build an arbitrary large graph with bounded degree and good expansion properties.

Before we formally analyse the zig-zag product, let us look at the meaning behind this definition: the zig-zag product corresponds to 3-step walks on the replacement product graph, where the first and the last steps are in the inner-cloud edges and the middle step is an inter-cloud edge, and each vertex in the cloud corresponds to an edge starting from the vertex which the cloud represents, see Figure 6.4 for example.

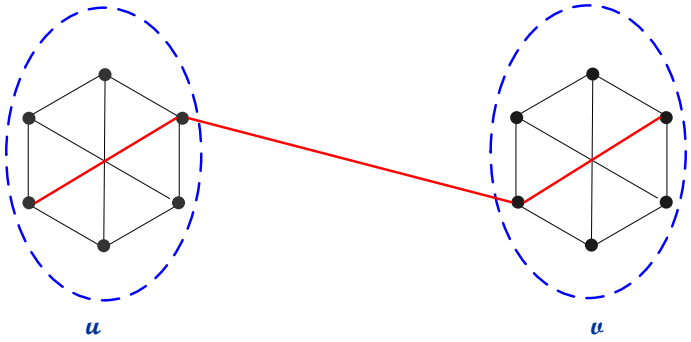

Figure 6.4: The meaning behind the zig-zag product

Theorem 6.6. *Suppose that G is an (N_1, d_1, λ_1)-expander and H is a (d_1, d_2, λ_2)-expander. Then $G \textcircled{z} H$ is an $(N_1 d_1, d_2^2, f(\lambda_1, \lambda_2))$-expander, where*

$$f(\lambda_1, \lambda_2) \leq \lambda_1 + \lambda_2 + \lambda_2^2.$$

Proof. The number of vertices and degree of $G \textcircled{z} H$ are obtained directly from the definition of the zig-zag product so we only need to analyse the spectral expansion of $G \textcircled{z} H$. Let M be the normalised adjacency matrix of $G \textcircled{z} H$. It suffices to show that for any $\alpha \perp \mathbf{1}_{N_1 d_1}, \alpha \in \mathbb{R}^{N_1 d_1}$, there holds that

$$|\langle M\alpha, \alpha \rangle| \leq f(\lambda_1, \lambda_2) \cdot |\langle \alpha, \alpha \rangle|.$$

Let $\alpha \in \mathbb{R}^{N_1 d_1}$ with the property that $\alpha \perp \mathbf{1}_{N_1 d_1}$. For any vertex $v \in [N_1]$, define $\alpha_v \in \mathbb{R}^{d_1}$ by $(\alpha_v)_k = \alpha_{vk}$. Also, let $C : \mathbb{R}^{N_1 d_1} \to \mathbb{R}^{N_1}$ be a linear mapping such that $(C\alpha)_v = \sum_{k=1}^{d_1} \alpha_{vk}$. Then we can express α as

$$\alpha = \sum_{v \in [N_1]} e_v \otimes \alpha_v.$$

Let $\alpha_v = \alpha_v^{\parallel} + \alpha_v^{\perp}$ where $\alpha_v^{\perp} \perp \mathbf{1}_{d_1}$. Then

$$\begin{aligned} \alpha &= \sum_v \left(e_v \otimes \alpha_v^{\parallel} \right) + \sum_v \left(e_v \otimes \alpha_v^{\perp} \right) \\ &:= \alpha^{\parallel} + \alpha^{\perp}. \end{aligned}$$

Let A and B be the normalised adjacency matrices of G and H, respectively. Let $\widetilde{B} = I_{N_1} \otimes B$ and $\widetilde{A}$ be the permutation matrix corresponding to Rot_G, i.e., an $N_1 d_1 \times N_1 d_1$ matrix where

$$\widetilde{A}_{(u,k)(v,\ell)} = \begin{cases} 1 & \text{if } \mathrm{Rot}_G(u,k) = \mathrm{Rot}_G(v,\ell) \\ 0 & \text{otherwise.} \end{cases}$$

Then $M = \widetilde{B}\widetilde{A}\widetilde{B}$. Since $\widetilde{B}$ is real symmetric, we have

$$\langle M\alpha, \alpha \rangle = \left\langle \widetilde{B}\widetilde{A}\widetilde{B}\alpha, \alpha \right\rangle = \left\langle \widetilde{A}\widetilde{B}\alpha, \widetilde{B}\alpha \right\rangle.$$

On the other hand, we have $\widetilde{B}\alpha = \widetilde{B}(\alpha^{\parallel} + \alpha^{\perp}) = \alpha^{\parallel} + \widetilde{B}\alpha^{\perp}$. Thus

$$\begin{aligned} \langle M\alpha, \alpha \rangle &= \left\langle \widetilde{A}\left(\alpha^{\parallel} + \widetilde{B}\alpha^{\perp}\right), \left(\alpha^{\parallel} + \widetilde{B}\alpha^{\perp}\right) \right\rangle \\ &= \left\langle \widetilde{A}\alpha^{\parallel}, \alpha^{\parallel} \right\rangle + \left\langle \widetilde{A}\alpha^{\parallel}, \widetilde{B}\alpha^{\perp} \right\rangle + \left\langle \widetilde{A}\widetilde{B}\alpha^{\perp}, \alpha^{\parallel} \right\rangle + \left\langle \widetilde{A}\widetilde{B}\alpha^{\perp}, \widetilde{B}\alpha^{\perp} \right\rangle \end{aligned}$$

and

$$\begin{aligned} &|\langle M\alpha, \alpha \rangle| \\ &\leq \left|\left\langle \widetilde{A}\alpha^{\parallel}, \alpha^{\parallel} \right\rangle\right| + \left\|\widetilde{A}\alpha^{\parallel}\right\| \cdot \left\|\widetilde{B}\alpha^{\perp}\right\| + \left\|\widetilde{A}\widetilde{B}\alpha^{\perp}\right\| \cdot \left\|\alpha^{\parallel}\right\| + \left\|\widetilde{A}\widetilde{B}\alpha^{\perp}\right\| \cdot \left\|\widetilde{B}\alpha^{\perp}\right\| \\ &= \left|\left\langle \widetilde{A}\alpha^{\parallel}, \alpha^{\parallel} \right\rangle\right| + 2\left\|\alpha^{\parallel}\right\| \cdot \left\|\widetilde{B}\alpha^{\perp}\right\| + \|\widetilde{B}\alpha^{\perp}\|^2, \end{aligned} \tag{6.1}$$

where the last equality holds as $\widetilde{A}$ is a permutation and $\|\widetilde{A}x\| = \|x\|$ for any $x \in \mathbb{N}^{N_1 d_1}$. Notice that

$$\begin{aligned} \|\widetilde{B}\alpha^{\perp}\|^2 &= \left\|\widetilde{B}\left(\sum_v e_v \otimes \alpha_v^{\perp}\right)\right\|^2 \\ &= \left\|\sum_v e_v \otimes B\alpha_v^{\perp}\right\|^2 \\ &= \sum_v \left\|B\alpha_v^{\perp}\right\|^2 \\ &\leq \sum_v \lambda_2^2 \left\|\alpha_v^{\perp}\right\|^2 \\ &\leq \lambda_2^2 \left\|\alpha^{\perp}\right\|^2. \end{aligned} \tag{6.2}$$

So we only need to bound $\left|\left\langle \widetilde{A}\alpha^{\parallel}, \alpha^{\parallel} \right\rangle\right|$. Direct calculation gives us that

$$\left\langle \widetilde{A}\alpha^{\parallel}, \alpha^{\parallel} \right\rangle = \left\langle \widetilde{A}\alpha^{\parallel}, C\alpha \otimes \mathbf{1}_{d_1} \right\rangle / d_1 = \left\langle C\widetilde{A}\alpha^{\parallel}, C\alpha \right\rangle / d_1 = \langle AC\alpha, C\alpha \rangle / d_1$$

and

$$\begin{aligned}\left|\left\langle \widetilde{A}\alpha^{\|}, \alpha^{\|}\right\rangle\right| &\leq \lambda_1 \langle C\alpha, C\alpha\rangle / d_1 \\ &= \lambda_1 \langle C\alpha \otimes \mathbf{1}_{d_1}, C\alpha \otimes \mathbf{1}_{d_1}\rangle / d_1^2 \\ &= \lambda_1 \cdot \langle \alpha^{\|}, \alpha^{\|}\rangle \\ &= \lambda_1 \left\|\alpha^{\|}\right\|^2.\end{aligned} \tag{6.3}$$

Combining (6.2) and (6.3), we have

$$|\langle M\alpha, \alpha\rangle| \leq \lambda_1 \left\|\alpha^{\|}\right\|^2 + 2\lambda_2 \left\|\alpha^{\|}\right\| \cdot \left\|\alpha^{\perp}\right\| + \lambda_2^2 \left\|\alpha^{\perp}\right\|^2.$$

By taking $p = \frac{\|\alpha^{\|}\|}{\|\alpha\|}$ and $q = \frac{\|\alpha^{\perp}\|}{\|\alpha\|}$, we have $p^2 + q^2 = 1$. Therefore

$$\begin{aligned}\frac{|\langle M\alpha, \alpha\rangle|}{|\langle \alpha, \alpha\rangle|} &\leq \lambda_1 p^2 + 2\lambda_2 pq + \lambda_2^2 q^2 \\ &\leq \lambda_1 + \lambda_2 + \lambda_2^2,\end{aligned}$$

which completed the proof. □

6.3 Construction of Expanders

Finally, we use the the zig-zag product to construct expander graphs. Our contribution is based on iterations, and in each iteration we construct a graph based on a combination of graph powering and the zig-zag product.

Theorem 6.7. *Let H be a (d^4, d, λ_0) graph for some $\lambda_0 \leq 1/5$. Define $G_1 = H^2$ and $G_{t+1} = G_t^2 \textcircled{z} H$ for $t \geq 1$. Then for all t, G_t is a (d^{4t}, d^2, λ)-expander with $\lambda \leq 2/5$.*

Proof. We prove the theorem by induction. When $t = 1$, it is straightforward to see that G_1 is a (d^4, d^2, λ_0^2)-expander. Assume that G_{t-1} is a $(d^{4(t-1)}, d^2, \lambda)$-expander for $\lambda \leq 2/5$. By Definition 6.4, the number of vertices in G_t is d^{4t}. So it suffices to show the spectral expansion of G_t. By Theorem 6.6, the spectral expansion of G_t is

$$\lambda(G_t) \leq \lambda(G_{t-1}^2) + \lambda(H) + \lambda(H^2) = \left(\frac{2}{5}\right)^2 + \frac{1}{5} + \frac{1}{25} = \frac{2}{5},$$

which finishes the proof. □

Lecture 7

The Power Method for Computing the Eigenvalues

In this lecture we study efficient algorithms for computing the eigenvalues. Throughout the lecture, we assume that the matrix $M \in \mathbb{R}^{n \times n}$ is PSD. The eigenvalues of M are $\lambda_1 \geq \ldots \geq \lambda_n$, with the associated eigenvectors $f_1, \ldots, f_n$.

7.1 Computing the Largest Eigenvalue

Algorithm 2 Power Method for Approximating the Largest Eigenvalue

1: **Input:** a PSD symmetric matrix $M \in \mathbb{R}^{n \times n}$, and positive integer t
2: Choose uniformly at random $x_0 \sim \{-1, 1\}^n$.
3: **for** $i = 1$ to t **do**
4: $\quad x_i = M x_{i-1}$
5: **return** x_t

Notice that the algorithm performs $O(t \cdot (n + m))$ arithmetic operations, where m is the number of non-zero entries of the matrix M.

Theorem 7.1. *For every positive and semi-definite matrix M, positive integer t and parameter $\varepsilon > 0$, with probability $4/5$ over the initial choices of x_0, Algorithm 2 outputs a vector x_t such that*

$$\frac{x_t^\intercal M x_t}{x_t^\intercal x_t} \geq (1 - \varepsilon) \cdot \lambda_1 \cdot \frac{1}{1 + 4n(1 - \varepsilon)^{2t}}.$$

In particular, when setting $t = O(\log n / \varepsilon)$, we have that

$$\frac{x_t^\intercal M x_t}{x_t^\intercal x_t} \geq (1 - O(\varepsilon))\lambda_1.$$

The proof is based on the following two lemmas.

Lemma 7.2. *Let $v \in \mathbb{R}^n$ such that $\|v\| = 1$. Sample uniformly $x \in \{-1, 1\}^n$. Then it holds that*

$$\mathbb{P}\left[|\langle x, v\rangle| \geq \frac{1}{2}\right] \geq \frac{3}{16}.$$

Lemma 7.3. *Let $x \in \mathbb{R}^n$ be a vector such that $|\langle x, f_1\rangle| \geq 1/2$. Then, for every positive integer t and positive $\varepsilon > 0$, if we define $y = M^t x$, then we have that*

$$\frac{y^\intercal M y}{y^\intercal y} \geq (1-\varepsilon) \cdot \lambda_1 \cdot \frac{1}{1 + 4\|x\|^2(1-\varepsilon)^{2t}}.$$

Proof of Theorem 7.1. By Lemma 7.2, with constant probability, a randomly sampled $x \in \{-1, 1\}^n$ satisfies $|\langle x, v\rangle| \geq 1/2$ for any $\|v\| = 1$. Conditioning on this event, Lemma 7.3 states that

$$\frac{y^\intercal M y}{y^\intercal y} \geq (1-\varepsilon) \cdot \lambda_1 \cdot \frac{1}{1 + 4\|x\|^2(1-\varepsilon)^{2t}}.$$

Then, the theorem holds by the fact that $\|x\|^2 = n$. □

Proof of Lemma 7.2. Define a random variable $S = \langle x, v\rangle$. Then, it holds that $\mathbb{E}[S] = 0$, $\mathbb{E}\left[S^2\right] = 1$, and

$$\mathbb{E}\left[S^4\right] = 3\|v\|^2 - 2\|v\|_4^2 \leq 3.$$

Recall that the Paley-Zygmund inequality states that if Z is a non-negative random variable with finite variance, then it holds for every $0 \leq \delta \leq 1$ that

$$\mathbb{P}\left[Z \geq \delta \cdot \mathbb{E}Z\right] \geq (1-\delta)^2 \cdot \frac{(\mathbb{E}Z)^2}{\mathbb{E}[Z^2]},$$

which follows by noticing that

$$\begin{aligned}
\mathbb{E}[Z] &= \mathbb{E}\left[Z \cdot \mathbf{1}_{Z < \delta\mathbb{E}Z}\right] + \mathbb{E}\left[Z \cdot \mathbf{1}_{Z \geq \delta\mathbb{E}Z}\right] \\
&\leq \delta\mathbb{E}Z + \sqrt{\mathbb{E}Z^2} \cdot \sqrt{\mathbb{E}\mathbf{1}_{Z \geq \delta\mathbb{E}Z}} \\
&= \delta\mathbb{E}Z + \sqrt{\mathbb{E}Z^2} \cdot \sqrt{\mathbb{P}\left[Z \geq \delta\mathbb{E}Z\right]}.
\end{aligned}$$

We apply the Paley-Zygmund inequality to the case $Z = S^2$ and $\delta = 1/4$ and have that

$$\mathbb{P}\left[S^2 \geq \frac{1}{4}\right] \geq \left(\frac{3}{4}\right)^2 \cdot \frac{1}{3} = \frac{3}{16}.$$

□

Notice that the proof above works even if the seeding vector x is selected according to a 4-wise independent distribution. This means that the algorithm can be derandomised in polynomial time.

Proof of Lemma 7.3. We write x as a linear combination of the eigenvectors

$$x = a_1 f_1 + \cdots + a_n f_n$$

where the coefficients can be computed as $a_i = \langle x, f_i \rangle$. Then, we have that

$$y = a_1 \lambda_1^t f_1 + \cdots + a_n \lambda_n^t f_n,$$

and therefore

$$y^\intercal M y = \sum_{i=1}^{n} a_i^2 \lambda_i^{2t+1},$$

as well as

$$y^\intercal y = \sum_{i=1}^{n} a_i^2 \lambda_i^{2t}.$$

Without loss of generality let k be the number of eigenvalues larger than $\lambda_1 \cdot (1-\varepsilon)$. Then, it holds that

$$y^\intercal M y \geq \sum_{i=1}^{k} a_i^2 \lambda_i^{2t+1} \geq \lambda_1 (1-\varepsilon) \sum_{i=1}^{k} a_i^2 \lambda_i^{2t}.$$

We also see that

$$\begin{aligned}
\sum_{i=k+1}^{n} a_i^2 \lambda_i^{2t} &\leq \lambda_1^{2t} \cdot (1-\varepsilon)^{2t} \sum_{i=k+1}^{n} a_i^2 \\
&\leq \lambda_1^{2t} \cdot (1-\varepsilon)^{2t} \|x\|^2 \\
&\leq 4 a_1^2 \lambda_1^{2t} \cdot (1-\varepsilon)^{2t} \|x\|^2 \\
&\leq 4 \|x\|^2 (1-\varepsilon)^{2t} \sum_{i=1}^{k} a_i^2 \lambda_i^{2t}.
\end{aligned}$$

Hence, we have that

$$y^\intercal y \leq \left(1 + 4\|x\|^2 (1-\varepsilon)^{2t}\right) \cdot \sum_{i=1}^{k} a_i^2 \lambda_i^{2t},$$

which implies

$$\frac{y^\intercal M y}{y^\intercal y} \geq \lambda_1 \cdot (1-\varepsilon) \cdot \frac{1}{1 + 4\|x\|^2 (1-\varepsilon)^{2t}}.$$

□

7.2 Computing the Second Largest Eigenvalue

If M is a PSD matrix and if we know a unit-length eigenvector f_1 of the largest eigenvalue of M, we can approximately find the second eigenvalue with the following adaption of the algorithm from the previous section.

Algorithm 3 Power Method for Approximating the Second Largest Eigenvalue

1: **Input:** a PSD symmetric matrix $M \in \mathbb{R}^{n \times n}$, and positive integer t
2: Choose uniformly at random $x \sim \{-1, 1\}^n$.
3: Let $x_0 = x - \langle f_1, x \rangle \cdot f_1$
4: **for** $i = 1$ to t **do**
5: $\quad x_i = M x_{i-1}$
6: **return** x_t

If $f_1, \ldots, f_n$ is an orthonormal basis of the eigenvectors for the eigenvalues $\lambda_1 \geq \ldots \geq \lambda_n$ of M, then we can write the picked random vector initially as

$$x = a_1 f_1 + \cdots + a_n f_n,$$

and with constant probability it holds that $|a_2| \geq 1/2$. Then, x_0 is the projection of x on the subspace orthogonal to f_1, i.e.,

$$x_0 = a_2 f_2 + \cdots + a_n f_n.$$

Notice that $\|x_0\| \leq n$. Moreover, the output vector x_t can be written as

$$x_t = a_2 \lambda_2^t f_2 + \cdots + a_n \lambda_n^t f_n.$$

Then, we can apply the same analysis as before, and obtain the following result:

Theorem 7.4. *For every PSD matrix M, positive integer t and parameter $\varepsilon > 0$, with probability $4/5$ over the choices of x, Algorithm 3 outputs a vector $y \perp f_1$ such that*

$$\frac{y^\intercal M y}{y^\intercal y} \geq \lambda_2 \cdot (1 - \varepsilon) \cdot \frac{1}{1 + 4n(1-\varepsilon)^{2t}},$$

where λ_2 is the second largest eigenvalue of M, counting multiplicities.

Finally, we study the applications of the power methods in computing the second smallest eigenvalue of $\mathcal{L} = I - D^{-1/2} A D^{-1/2}$ for an undirected graph. It is easy to see that any eigenvector f_i of $\mathcal{L}$ with eigenvalue λ_i is the eigenvector of $I + D^{-1/2} A D^{-1/2}$ with eigenvalue $2 - \lambda_i$, and therefore $I + D^{-1/2} A D^{-1/2}$ is PSD as well. Since f_1 of $\mathcal{L}$ is known and only depends on the degree sequence, the second smallest eigenvalue of $\mathcal{L}$ can be computed by the power method as well.

Lecture 8

Graph Clustering

In this lecture we study graph clustering algorithms for the *k-way partitioning problem*. We call subsets of vertices (i.e. *clusters*) $A_1, \ldots, A_k$ a *k-way partition* of G if $A_i \cap A_j = \emptyset$ for different i and j, and $\bigcup_{i=1}^{k} A_i = V$. The k-way partitioning problem asks for a k-way partition of G such that the conductance of any A_i in the partition is at most the *k-way expansion constant*, defined by

$$\rho(k) \triangleq \min_{\text{partition } A_1,\ldots,A_k} \max_{1 \le i \le k} h_G(A_i). \tag{8.1}$$

Clusters of low conductance in networks appearing in practice usually capture the notion of *community*, and algorithms for finding these subsets have applications in various domains such as community detection and network analysis. In computer vision, most image segmentation procedures are based on region-based merge and split, which in turn rely on partitioning graphs into multiple subsets. On a theoretical side, decomposing vertex/edge sets into multiple disjoint subsets is used in designing approximation algorithms for Unique Games, and efficient algorithms for graph problems.

To capture a structure of clusters in graphs, we look at the following higher-order Cheeger inequality:

$$\frac{\lambda_k}{2} \le \rho(k) \le O(k^2)\sqrt{\lambda_k}. \tag{8.2}$$

Informally, the higher-order Cheeger inequality shows that graph G has a k-way partition with low $\rho(k)$ if and only if λ_k is small. Indeed, (8.2) implies that a large gap between λ_{k+1} and $\rho(k)$ *guarantees* (i) existence of a k-way partition $\{S_i\}_{i=1}^{k}$ with bounded $h_G(S_i) \le \rho(k)$, and (ii) any $(k+1)$-way partition of G contains a subset with significantly higher conductance $\rho(k+1) \ge \lambda_{k+1}/2$ compared with $\rho(k)$. Hence, a

suitable lower bound on the *gap* $\Upsilon(k)$ for some k, defined by

$$\Upsilon(k) \triangleq \frac{\lambda_{k+1}}{\rho(k)}, \tag{8.3}$$

implies the existence of a k-way partition for which every cluster has low conductance, and that G is a *well-clustered* graph.

8.1 The Structure Theorem

We first study the relations between the multiple cuts of a graph and the eigenvectors of the graph's normalized Laplacian matrix. Given clusters $S_1 \dots S_k$, define the indicator vector of cluster S_i by

$$g_i(u) = \begin{cases} 1 & \text{if } u \in S_i, \\ 0 & \text{if } u \notin S_i, \end{cases} \tag{8.4}$$

and define the corresponding normalized indicator vector by

$$\overline{g}_i = \frac{D^{1/2} g_i}{\|D^{1/2} g_i\|}. \tag{8.5}$$

We know that G has k connected components if and only if the k smallest eigenvalues are 0, implying that the spaces spanned by $f_1, \cdots, f_k$ and $\bar{g}_1, \cdots, \bar{g}_k$ are the same. Generalising this result, we expect that these two spaces would be still similar if these k components of G are loosely connected, in the sense that (i) every eigenvector f_i can be approximately expressed by a linear combination of $\{\overline{g}_i\}_{i=1}^k$, and (ii) every indicator vector $\bar{g}_i$ can be approximately expressed by a linear combination of $\{f_i\}_{i=1}^k$. This leads to our structure theorem, which is illustrated in Figure 8.1.

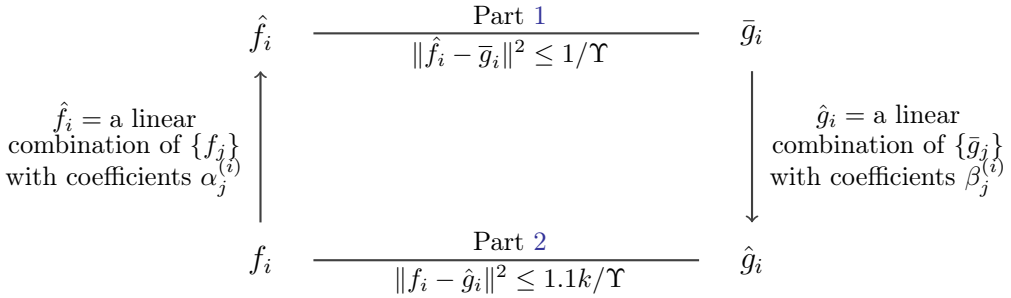

Figure 8.1: Relations among $\{\hat{f}_i\}$, $\{f_i\}$, $\{\bar{g}_i\}$, and $\{\hat{g}_i\}$ given in Theorem 8.1.

Theorem 8.1 (The Structure Theorem)**.** *Let* $\Upsilon = \Omega(k^2)$*, and* $1 \le i \le k$*. Then, the following statements hold:*

1. *There is a linear combination of the eigenvectors* $f_1, \dots, f_k$ *with coefficients* $\alpha_j^{(i)}$*:* $\hat{f}_i = \alpha_1^{(i)} f_1 + \cdots + \alpha_k^{(i)} f_k$*, such that* $\left\| \overline{g}_i - \hat{f}_i \right\|^2 \le 1/\Upsilon$.

2. *There is a linear combination of the vectors* $\bar{g}_1, \ldots, \bar{g}_k$ *with coefficients* $\beta_j^{(i)}$*:* $\hat{g}_i = \beta_1^{(i)}\bar{g}_1 + \cdots + \beta_k^{(i)}\bar{g}_k$*, such that* $\|f_i - \hat{g}_i\|^2 \leq 1.1k/\Upsilon$.

Part 1 of Theorem 8.1 shows that the normalized indicator vectors $\bar{g}_i$ of every cluster S_i can be approximated by a linear combination of *the first k eigenvectors*, with respect to the value of Υ. The proof follows from the fact that if $\bar{g}_i$ has small Rayleigh quotient, then the inner product between $\bar{g}_i$ and the eigenvectors corresponding to larger eigenvalues must be small.

Proof of Part 1 of Theorem 8.1. We write $\overline{g}_i$ as a linear combination of the eigenvectors of $\mathcal{L}$, i.e.

$$\overline{g}_i = \alpha_1^{(i)} f_1 + \cdots + \alpha_n^{(i)} f_n$$

and let the vector $\hat{f}_i$ be the projection of vector $\bar{g}_i$ on the subspace spanned by $\{f_i\}_{i=1}^k$, i.e.

$$\hat{f}_i = \alpha_1^{(i)} f_1 + \cdots + \alpha_k^{(i)} f_k.$$

By the definition of Rayleigh quotients, we have that

$$\begin{aligned}
\mathcal{R}(\overline{g}_i) &= \left(\alpha_1^{(i)} f_1 + \cdots + \alpha_n^{(i)} f_n\right)^{\mathsf{T}} \mathcal{L} \left(\alpha_1^{(i)} f_1 + \cdots + \alpha_n^{(i)} f_n\right) \\
&= \left(\alpha_1^{(i)}\right)^2 \lambda_1 + \cdots + \left(\alpha_n^{(i)}\right)^2 \lambda_n \\
&\geq \left(\alpha_2^{(i)}\right)^2 \lambda_2 + \cdots + \left(\alpha_k^{(i)}\right)^2 \lambda_k + \left(1 - \alpha' - \left(\alpha_1^{(i)}\right)^2\right) \lambda_{k+1},
\end{aligned}$$

where $\alpha' \triangleq \left(\alpha_2^{(i)}\right)^2 + \cdots + \left(\alpha_k^{(i)}\right)^2$. Therefore, we have that

$$1 - \alpha' - \left(\alpha_1^{(i)}\right)^2 \leq \mathcal{R}(\overline{g}_i)/\lambda_{k+1} \leq 1/\Upsilon,$$

and

$$\|\overline{g}_i - \hat{f}_i\|^2 = \left(\alpha_{k+1}^{(i)}\right)^2 + \cdots + \left(\alpha_n^{(i)}\right)^2 = 1 - \alpha' - \left(\alpha_1^{(i)}\right)^2 \leq 1/\Upsilon,$$

which finishes the proof. □

Part 2 of Theorem 8.1 is more interesting, and shows that the opposite direction holds as well, i.e., any f_i $(1 \leq i \leq k)$ can be approximated by a linear combination of the normalized indicator vectors $\{\overline{g}_i\}_{i=1}^k$. To sketch the proof, note that if we could write every $\overline{g}_i$ *exactly* as a linear combination of $\{f_i\}_{i=1}^k$, then we could write every f_i $(1 \leq i \leq k)$ as a linear combination of $\{\overline{g}_i\}_{i=1}^k$. This is because both of $\{f_i\}_{i=1}^k$ and $\{\overline{g}_i\}_{i=1}^k$ are sets of linearly independent vectors of the same dimension and $\operatorname{span}\{\overline{g}_1, \ldots, \overline{g}_k\} \subseteq \operatorname{span}\{f_1, \ldots, f_k\}$. However, the $\overline{g}_i$'s are only close to a linear combination of the first k eigenvectors, as shown in Part 1. We will denote this combination as $\hat{f}_i$, and use the fact that the errors of approximation are small to show that these $\{\hat{f}_i\}_{i=1}^k$ are almost orthogonal between each other. This allows us to show that $\operatorname{span}\{\hat{f}_1, \ldots, \hat{f}_k\} = \operatorname{span}\{f_1, \ldots, f_k\}$, which implies Part 2.

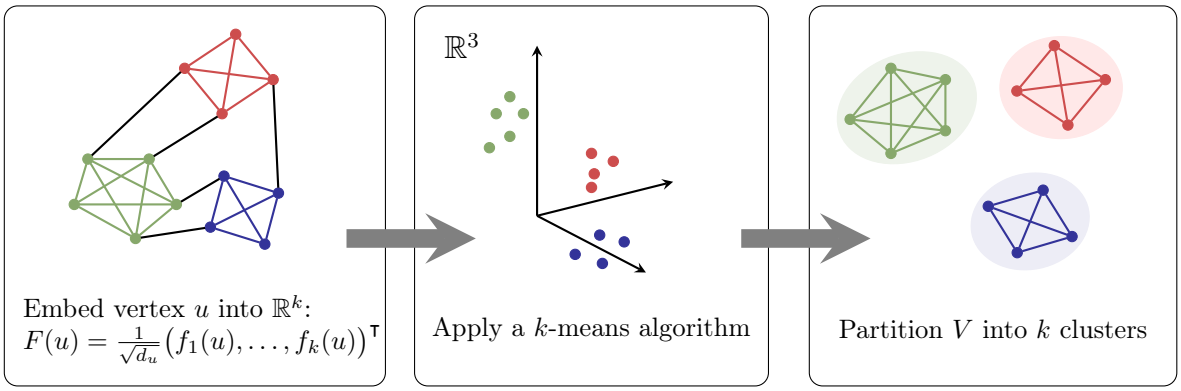

Figure 8.2: The three steps of a spectral clustering algorithm: (1) embed each vertex u to a point in $\mathbb{R}^k$ based on the top or bottom k eigenvectors of a matrix representing the graph; (2) apply a k-means algorithm to group the embedded points into k clusters; (3) return the corresponding partition of the vertex set as output.

The structure theorem shows a close connection between the first k eigenvectors and the indicator vectors of the clusters. We leverage this and the fact that the $\{\hat{g}_i\}$'s are almost orthogonal between each other to show that, for any two different clusters S_i and S_j, there exists an eigenvector having reasonably different values on the coordinates which correspond to S_i and S_j. The proof is essentially based on direct calculations.

Lemma 8.2. *Let $\Upsilon = \Omega(k^3)$. For any $1 \le i \le k$, let $\hat{g}_i = \beta_1^{(i)}\overline{g}_1 + \cdots + \beta_k^{(i)}\overline{g}_k$ be such that $\|f_i - \hat{g}_i\| \le 1.1k/\Upsilon$. Then, for any $\ell \ne j$, there exists $i \in \{1, \ldots, k\}$ such that*

$$\left|\beta_\ell^{(i)} - \beta_j^{(i)}\right| \ge \zeta \triangleq \frac{1}{10\sqrt{k}}. \tag{8.6}$$

8.2 Spectral Clustering

Spectral clustering is one of the most popular algorithms for graph clustering. The general framework of spectral clustering consists in (1) computing an embedding of the vertices in a low dimensional Euclidean space using the eigenvectors of a matrix representing the graph; (2) partition these points using a geometric clustering algorithm; (3) output a corresponding partition of the graph, see Figure 8.2 for illustration.

To formally describe a spectral clustering algorithm, let us assume that we are given a graph $G = (V, E, w)$ with normalised Laplacian $\mathcal{L}_G$, and $f_1, \ldots, f_k$ be the bottom k eigenvectors of $\mathcal{L}$. The *spectral embedding* $F : V \to \mathbb{R}^k$ maps any vertex $u \in V$ to a point

$$F(u) = \frac{1}{\sqrt{d_u}}\left(f_1(u), \ldots, f_k(u)\right)^\intercal. \tag{8.7}$$

The second step of a spectral clustering algorithm is to cluster the embedded points through a k-means algorithm. For any partition $\mathcal{X}_1, \cdots, \mathcal{X}_k$ of the set $\mathcal{X} \subset \mathbb{R}^d$, we define the cost function as

$$\mathsf{COST}(\mathcal{X}_1, \ldots, \mathcal{X}_k) \triangleq \min_{c_1,\ldots,c_k \in \mathbb{R}^d} \sum_{i=1}^{k} \sum_{x \in \mathcal{X}_i} \|x - c_i\|^2,$$

i.e., the COST function is the total ℓ_2^2-distance between the points in $\mathcal{X}$ and their individually closest centre c_i, where $c_1, \ldots, c_k \in \mathbb{R}^d$ are chosen to minimise this distance. We further define the optimal clustering cost as

$$\Delta_k^2(\mathcal{X}) \triangleq \min_{\text{partition } \mathcal{X}_1,\ldots,\mathcal{X}_k} \mathsf{COST}(\mathcal{X}_1, \ldots, \mathcal{X}_k), \tag{8.8}$$

that is, the minimum COST of a k-way partition of $\mathcal{X}$. The problem of finding an optimal k-means solution is NP-hard even when all the points belongs to $\mathbb{R}^2$. So we say that an algorithm for k-means achieves an APT-approximation ratio if, for any set of points $\mathcal{X} \subseteq \mathbb{R}^d$, it outputs a partition $\{A_1, \ldots, A_k\}$ of $\mathcal{X}$ such that $\mathsf{COST}(A_1, \ldots, A_k) \leq \mathsf{APT} \cdot \Delta_k^2(\mathcal{X})$.

Now we analyse the spectral clustering algorithm. We first define k points $p^{(i)} \in \mathbb{R}^k$ $(1 \leq i \leq k)$, where

$$p^{(i)} \triangleq \frac{1}{\sqrt{\operatorname{vol}(S_i)}} \left(\beta_i^{(1)}, \ldots, \beta_i^{(k)}\right)^{\intercal} \tag{8.9}$$

and the parameters $\{\beta_i^{(j)}\}_{j=1}^k$ are defined in Theorem 8.1. We will show in Lemma 8.3 that all embedded points $\mathcal{X}_i \triangleq \{F(u) : u \in S_i\}$ $(1 \leq i \leq k)$ are concentrated around $p^{(i)}$. Moreover, we bound the total ℓ_2^2-distance between points in $\mathcal{X}_i$ and $p^{(i)}$, which is proportional to $1/\Upsilon$: the bigger the value of Υ, the higher concentration the points within the same cluster have. Notice that we *do not* claim that $p^{(i)}$ is the actual center of $\mathcal{X}_i$. However, these approximated points $p^{(i)}$'s suffice for our analysis.

Lemma 8.3. *It holds that*

$$\sum_{i=1}^{k} \sum_{u \in S_i} d_u \left\| F(u) - p^{(i)} \right\|^2 \leq 1.1k^2/\Upsilon.$$

The next lemma shows that the ℓ_2^2-norm of $p^{(i)}$ is inversely proportional to the volume of S_i. This implies that embedded points from a big cluster are close to the origin, while embedded points from a small cluster are far from the origin. The proof can be essentially obtained by direct calculation.

Lemma 8.4. *It holds for every $1 \leq i \leq k$ that*

$$\frac{99}{100 \operatorname{vol}(S_i)} \leq \left\| p^{(i)} \right\|^2 \leq \frac{101}{100 \operatorname{vol}(S_i)}.$$

We will further show in Lemma 8.5 that these points $p^{(i)} (1 \leq i \leq k)$ exhibit another excellent property: the distance between $p^{(i)}$ and $p^{(j)}$ is inversely proportional to the volume of the *smaller* cluster between S_i and S_j. Therefore, points in S_i of smaller $\mathrm{vol}(S_i)$ are far from points in S_j of bigger $\mathrm{vol}(S_j)$. Notice that, if this were not the case, a misclassification of a small fraction of points in S_j could introduce a large error to S_i.

Lemma 8.5. *For every $i \neq j$, it holds that*

$$\left\|p^{(i)} - p^{(j)}\right\|^2 \geq \frac{\zeta^2}{10 \min\left\{\mathrm{vol}(S_i), \mathrm{vol}(S_j)\right\}},$$

where ζ is defined in (8.6).

Putting Lemma 8.3, Lemma 8.4, and Lemma 8.5 together, we can show that the volume of misclassified vertices in each cluster is small, since otherwise a significant amount of misclassified vertices in any cluster will significantly increase the value of $\sum_{i=1}^{k} \sum_{u \in S_i} d_u \|F(u) - p^{(i)}\|^2$, which contradicts Lemma 8.3. With some more combinatorial analysis for all possible correspondences between the returned k clusters and the optimal clusters, we have the following result:

Theorem 8.6. *Let G be a graph such that $\Upsilon(k) = \lambda_{k+1}/\rho(k) = \Omega(k^3)$, and $k \in \mathbb{N}$. Let $F : V[G] \to \mathbb{R}^k$ be the embedding defined in (8.7). Let $\{A_i\}_{i=1}^k$ be a k-way partition by any k-means algorithm running in $\mathbb{R}^k$ that achieves an approximation ratio APT. Then, the following statements hold: (i) $\mathrm{vol}(A_i \triangle S_i) = O\left(\mathsf{APT} \cdot k^3/\Upsilon(k)\right) \mathrm{vol}(S_i)$, and (ii) $\phi_G(A_i) = 1.1 \cdot \phi_G(S_i) + O\left(\mathsf{APT} \cdot k^3/\Upsilon(k)\right)$.*

8.3 Linear Time Spectral Clustering Algorithm

Now we present a nearly-linear time algorithm for partitioning well-clustered graphs, which achieves basically the same approximation guarantee as a spectral clustering algorithm. At a high level, our algorithm follows the general framework of k-means algorithms, and consists of two steps: the seeding step, and the grouping step. The seeding step chooses k candidate centers such that each one is close to the actual center of a different cluster. The grouping step assigns the remaining vertices to their individual closest candidate centers.

All the discussions for the seeding and grouping steps assume that we have an embedding $\{x(u)\}_{u \in V[G]}$ such that

$$\left(1 - \frac{1}{10 \log n}\right) \cdot \|F(u)\|^2 \leq \|x(u)\|^2 \leq \|F(u)\|^2 + \frac{1}{n^5}, \tag{8.10}$$

$$\left(1 - \frac{1}{10 \log n}\right) \cdot \|F(u) - F(v)\|^2 \leq \|x(u) - x(v)\|^2 \leq \|F(u) - F(v)\|^2 + \frac{1}{n^5} \tag{8.11}$$

Notice that these two conditions hold trivially if $\{x(u)\}_{u\in V[G]}$ is the spectral embedding $\{F(u)\}_{u\in V[G]}$, or any embedding produced by good approximations of the first k eigenvectors. However, obtaining such embedding becomes non-trivial for a large value of k, as directly computing the first k eigenvectors takes super-linear time. We will present a nearly-linear time algorithm that computes an embedding satisfying (8.10) and (8.11). By using standard dimensionality reduction techniques, we can always assume that the dimension of the embedding $\{x(u)\}_{u\in V[G]}$ is $d = O(\log^3 n)$.

The Seeding Step. We showed that the approximate center $p^{(i)}$ for every $1 \le i \le k$ satisfies

$$\frac{99}{100\,\mathrm{vol}(S_i)} \le \left\|p^{(i)}\right\|^2 \le \frac{101}{100\,\mathrm{vol}(S_i)},$$

and most embedded points $F(u)$ are close to their approximate centers. Together with (8.10) and (8.11), these two properties imply that, when sampling points $x(u)$ with probability proportional to $d_u \cdot \|x(u)\|^2$, vertices from different clusters will be approximately sampled with the same probability. Hence, when sampling $\Theta(k \log k)$ points in this way, with constant probability there is at least one point sampled from each cluster.

In the next step we remove the sampled points which are close to each other, and call this resulting set $C^\star$. It is easy to show that with constant probability there is exactly one point in $C^\star$ from a cluster. Algorithm 4 below gives a formal description of the seeding step.

Algorithm 4 SeedAndTrim$(k, \{x(u)\}_{u\in V[G]})$

1: **Input**: the number of clusters k, and the embedding $\{x(u)\}_{u\in V[G]}$
2: Let $K = \Theta(k \log k)$
3: **for** $i = 1, \ldots, K$ **do**
4: Set $c_i = u$ with probability proportional to $d_u \|x(u)\|^2$.
5: **for** $i = 2, \ldots, K$ **do**
6: Delete all c_j with $j < i$ such that $\|x(c_i) - x(c_j)\|^2 < \frac{\|x(c_i)\|^2}{2\cdot 10^4 k}$.
7: **return** the remaining sampled vertices.

The Grouping Step. After the seeding step, with constant probability we obtain a set of k vertices $C^\star = \{c_1, \cdots, c_k\}$, and these k vertices belong to k different clusters. Now we assign each remaining vertex u to a cluster S_i if, comparing with all other points $x(c_j)$ with $c_j \in C^\star$, $x(u)$ is closer to $x(c_i)$. To speed it up, we apply ε-approximate nearest neighbor data structures (ε-NNS), whose formal description is as follows:

Problem 8.1 (ε-approximate nearest neighbor problem)**.** *Given a set of point $P \subset \mathbb{R}^d$ and a point $q \in \mathbb{R}^d$, find a point $p \in P$ such that, for all $p' \in P$, $\|p-q\| \le (1+\varepsilon)\|p'-q\|$.*

Theorem 8.7. *Given a set P of points in $\mathbb{R}^d$, there is an algorithm that solves the ε-approximate nearest neighbor problem with $\widetilde{O}\left(|P|^{1+\frac{1}{1+\varepsilon}} + d \cdot |P|\right)$ preprocessing time and $\widetilde{O}\left(d \cdot |P|^{\frac{1}{1+\varepsilon}}\right)$ query time.*

Now we set $P = \{x(c_1), \ldots, x(c_k)\}$, and apply the above ε-approximate nearest neighbor data structures to assign the remaining vertices to k clusters $A_1, \cdots, A_k$. By Theorem 8.7 and setting $\varepsilon = \log k - 1$, this step can be finished with $\widetilde{O}(k)$ preprocessing time and $\widetilde{O}(1)$ query time for each query. Hence, the runtime of the grouping step is $\widetilde{O}(n)$.

Fast Computation of the Required Embedding. So far we assumed the existence of the embedding $\{x(u)\}_{u \in V[G]}$ satisfying (8.10) and (8.11), and analyzed the performance of the seeding and grouping steps. Now we give a sketch of the algorithm for computing all the required distances used in the seeding and grouping steps. Our algorithm is based on the so-called *heat kernal* of a graph.

Formally, the heat kernel of G with parameter $t \geq 0$ is defined by

$$H_t \triangleq \mathrm{e}^{-t\mathcal{L}} = \sum_{i=1}^{n} \mathrm{e}^{-t\lambda_i} f_i f_i^{\intercal}. \tag{8.12}$$

We view the heat kernel as a geometric embedding from $V[G]$ to $\mathbb{R}^n$ defined by

$$x_t(u) \triangleq \frac{1}{\sqrt{d_u}} \cdot \left(\mathrm{e}^{-t\cdot\lambda_1} f_1(u), \cdots, \mathrm{e}^{-t\cdot\lambda_n} f_n(u)\right), \tag{8.13}$$

and define the ℓ_2^2-distance between the points $x_t(u)$ and $x_t(v)$ by

$$\eta_t(u, v) \triangleq \|x_t(u) - x_t(v)\|^2. \tag{8.14}$$

The following lemma shows that, when $k = \Omega(\log n)$ and $\Upsilon = \Omega(k^3)$, the values of $\eta_t(u, v)$ for all edges $\{u, v\} \in E[G]$ can be approximately computed in $\widetilde{O}(m)$ time.

Lemma 8.8. *Let $k = \Omega(\log n)$ and $\Upsilon = \Omega(k^3)$. Then, there is $t = O(\mathrm{poly}(n))$ such that the embedding $\{x_t(u)\}_{u \in V[G]}$ defined in (8.13) satisfies (8.10) and (8.11). Moreover, the values of $\eta_t(u, v)$ for all $\{u, v\} \in E[G]$ can be approximately computed in $\widetilde{O}(m)$ time, such that with high probability the conditions (8.10) and (8.11) hold for all edges $u \sim v$.*

The Main Algorithm. We proved in Section 8.3 that if $k = \Omega(\log n)$ and $\Upsilon = \Omega(k^3)$, there is a

$$t \in \left(\frac{10 \log n}{\lambda_{k+1}}, \quad \frac{1}{20 \cdot \lambda_k \cdot \log n}\right) \tag{8.15}$$

such that $\{x_t(u)\}_{u \in V[G]}$ satisfies the conditions (8.10) and (8.11). Moreover, the values of $\|x_t(u) - x_t(v)\|$ for $\{u, v\} \in E[G]$ can be approximately computed in nearly-linear

time. However, it is unclear how to approximate λ_k, and without this approximation, obtaining the desired embedding $\{x(u)\}_{u\in V[G]}$ becomes highly non-trivial.

To overcome this, we run the seeding and grouping steps for all possible t of the form 2^i, where $t \in \mathbb{N}_{\geq 0}$, as it allows us to run the seeding and grouping steps with the right values of t at some point. However, by (8.14) the distance between any pair of embedded vertices decreases when we increase the value of t. Moreover, all these embedded points $\{x_t(u)\}_{u\in V[G]}$ tend to "concentrate" around a single point for an arbitrary large value of t. To avoid this situation, for every possible t we compute the value of $\sum_{v\in V[G]} d_v\|x_t(v)\|^2$, and the algorithm only moves to the next iteration if

$$\sum_{v\in V[G]} d_v\|x_t(v)\|^2 \geq k\left(1-\frac{2}{\log n}\right). \tag{8.16}$$

See Algorithm 5 for the formal description of our final algorithm.

Algorithm 5 A nearly-linear time graph clustering algorithm, $k = \Omega(\log n)$

input: the input graph G, and the number of clusters k
Let $t = 2$.
repeat
    Let $(c_1, \ldots, c_k) = \textsc{SeedAndTrim}\big(k, \{x_t(u)\}_{u\in V[G]}\big)$.
    if SeedAndTrim returns exactly k points **then**
        Compute a partition $A_1, \ldots, A_k$ of $V[G]$: for every $v \in V[G]$ assign v to its nearest center c_i using the ε-NNS algorithm with $\varepsilon = \log k - 1$.
    Let $t = 2t$
until $t > n^{10}$ **or** $\sum_{v\in V[G]} d_v\|x_t\|^2 < k\left(1-\frac{2}{\log n}\right)$.
return $(A_1, \cdots, A_k)$.

Lecture 9

Spectral Sparsification

A sparse graph is one whose number of edges is reasonably viewed as being proportional to the number of vertices. Since most algorithms run faster on sparse instances of graphs and it is more space-efficient to store sparse graphs, it is useful to obtain a sparse representation H of G so that certain properties between G and H are preserved, see Figure 9.1 for an illustration. Over the past three decades, different notions of graph sparsification have been proposed and widely used to design approximation algorithms. For instance, a *spanner* H of a graph G is a subgraph of G so that the shortest path distance between any pair of vertices is approximately preserved. Benczúr and Karger defined a *cut sparsifier* of a graph G to be a sparse subgraph H such that the value of any cut between G and H are approximately the same.

In this lecture we study efficient constructions of a **spectral sparsifer**, which is a sparse subgraph H of an undirected graph G such that many spectral properties of the Laplacian matrices between G and H are approximately preserved. Formally, for any undirected graph G with n vertices and m edges, we call a subgraph H of G, with proper reweighting of the edges, a $(1+\varepsilon)$-spectral sparsifier if

$$(1-\varepsilon)x^\intercal L_G x \leq x^\intercal L_H x \leq (1+\varepsilon)x^\intercal L_G x$$

holds for any $x \in \mathbb{R}^n$, where L_G and L_H are the respective Laplacian matrices of G and H. Our goal is to construct a spectral sparisfier with fewer edges as fast as possible. The main result we will present in this lecture is as follows:

Theorem 9.1. *There is a randomised algorithm that, given a graph G and an approximation parameter $\varepsilon > 0$, constructs a spectral sparsifier H of $O(n \log n/\varepsilon^2)$ edges in $\widetilde{O}(m \log(1/\varepsilon))$ time with constant probability. Here, the notation $\widetilde{O}$ hides the term of* $\operatorname{poly} \log n$.

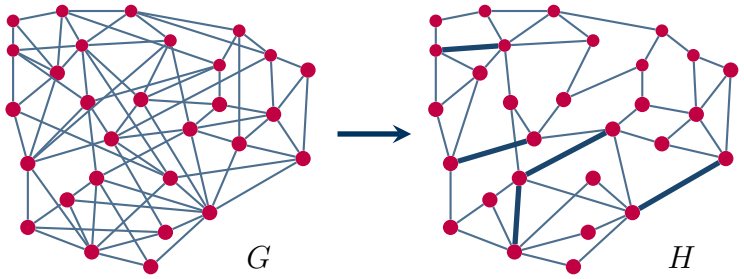

Figure 9.1: The graph sparsification is a reweighted subgraph H of an original graph G such that certain properties are preserved. These subgraphs are sparse, and are more space-efficient to be stored than the original graphs. The picture above uses the thickness of edges in H to represent their weights.

9.1 Electrical Networks

For an undirected graph G with n vertices and m edges, the incidence matrix of G is the matrix $B \in \mathbb{R}^{m \times n}$, where the rows and columns of B are indexed by the edges and vertices of graph G. Formally, for any edge $e \in E[G]$ and vertex v, we have that

$$B_{e,v} = \begin{cases} 1 & \text{if } v \text{ is } e\text{'s head} \\ -1 & \text{if } v \text{ is } e\text{'s tail} \\ 0 & \text{otherwise,} \end{cases}$$

where the orientation of edge e is chosen arbitrarily. We also define the diagonal matrix W, where $W_{e,e}$ equals to the weight of edge e.

Lemma 9.2. *Let G be a graph with (arbitrarily chosen) incidence matrix B and Laplacian L. Then, it holds that $L = B^\intercal W B$.*

Now we associate an electrical network to G. We replace edge edge $u \sim v$ with a resistor of value $1/w(u,v)$. To make this the setup intersting, we need to add power sources to its vertices. Suppose $i_{\mathsf{ext}} \in \mathbb{R}^n$ is a vector which indicates how much current is going in at each vertex. This will induce voltages at each vertex and a current across each edge. We capture these by vectors $v \in \mathbb{R}^n$ and $i \in \mathbb{R}^m$ respectively. Kirchoff's law states that, for every vertex, the difference of the outgoing current and the incoming current on the edges adjacent to it equals to the external current input at that vertex. Hence, we have that

$$B^\intercal i = i_{\mathsf{ext}}.$$

On the other hand, Ohm's law states that the current in an edge equals the potential difference across its ends times its conductance:

$$i = WBv.$$

Combining these two facts, we have that

$$i_{\mathsf{ext}} = B^\intercal W B v = Lv.$$

If $i_{\mathsf{ext}} \perp \mathrm{span}(\mathbf{1}) = \ker(L)$, i.e., the total amount of current injected is equal to the total amount extracted, then we can write

$$v = L^\dagger i_{\mathsf{ext}}.$$

The **effective resistance** between two vertices u and v is defined as the potential difference induced between them when a unit current is injected at one vertex and extracted at the other. Let us derive an algebraic expression for the effective resistance in terms of $L^\dagger$. To inject and extract a unit current across the endpoints of an edge $u \sim v$, we set $i_{\mathbf{ext}} = (\delta_u - \delta_v)$, where $\delta_u \in \{0,1\}^n$ is the indicator vector of vertex u. Then the potentials induced by i_{ext} at the vertices are given by $L^\dagger i_{\mathsf{ext}}$. To measure the potential difference across $u \sim v$, we simply multiply by $(\delta_u - \delta_v)^\intercal$ on the left, hence the effective resistance $\mathsf{Reff}(u,v)$ of edge $u \sim v$ can be written as

$$\mathsf{Reff}(u,v) = (\delta_u - \delta_v)^\intercal L^\dagger (\delta_u - \delta_v).$$

9.2 Spielman-Srivastava Algorithm

The Spielman-Srivastava algorithm constructs a spectral sparsifier of G by sampling edges with probability proportional to their effective resistances. The formal description of the algorithm is shown in Algorithm 6:

Algorithm 6 Algorithm for constructing a spectral sparsifier

for every edge $u \sim v$ **do**
  let $q_{u,v} = w(u,v) \cdot \mathsf{Reff}(u,v)$
  let $p_{u,v} = \min\{1, C \cdot (\log n) q_{u,v}/\varepsilon^2\}$ for some constant C
$t = 0$;
$H = (V, \emptyset)$
while $t \leq O(n \log n/\varepsilon^2)$ **do**
  $t = t + 1$
  Sample an edge $u \sim v$ with probability $p_{u,v}$
  Add the sampled edge $u \sim v$ into graph H with weight $w(u,v)/p_{u,v}$.
return graph H

Before analysing Algorithm 6, we first explain the reweighting scheme. We define

$$L_{u\sim v} = (\delta_u - \delta_v)(\delta_u - \delta_v)^\intercal.$$

Then, the Laplacian matrix of graph G can be written as $L_G = \sum_{u\sim v} w(u,v) \cdot L_{u\sim v}$. Assuming that every edge $u \sim v$ is sampled with probability $p_{u,v}$ and, once $u \sim v$ is

sampled, we add $u \sim v$ with new weight $w(u,v)/p_{u,v}$ into the new graph. Then we have that

$$\mathbb{E}[L_H] = \sum_{u,v} p_{u,v} \cdot \frac{w(u,v)}{p_{u,v}} \cdot L_{u,v} = L_G,$$

i.e., in expectation the sampled H equals to G. So the key is to analyse the concentration properties.

9.3 Analysis of the Algorithm

Why do we sample edges based on their effective resistances? We first discuss why we sample edges with probability proportional to their effective resistance. Notice that

$$\begin{aligned}
\sum_{u\sim v} q_{u,v} &= \sum_{u\sim v} w(u,v)\mathsf{Reff}(u,v) \\
&= \sum_{u\sim v} w(u,v)(\delta_u - \delta_v)^\intercal L^\dagger (\delta_u - \delta_v) \\
&= \sum_{u\sim v} w(u,v)\,\mathrm{tr}(L^\dagger(\delta_u - \delta_v)(\delta_u - \delta_v)^\intercal) \\
&= \mathrm{tr}\left(L^\dagger \sum_{u\sim v} w(u,v)(\delta_u - \delta_v)(\delta_u - \delta_v)^\intercal\right) \\
&= \mathrm{tr}\left(L^\dagger L\right) \\
&= n-1.
\end{aligned}$$

The fact above can be also explained in a combinatorial way. Notice that $q_{u,v}$ is the probability that edge $u \sim v$ appears in a random spanning tree of G when we sample spanning trees with probability proportional to the product of their edge weights. Since every spanning tree has $n-1$ edges, the sum of these probabilities is $n-1$.

Sparsity of graph H. It is easy to see that the expected number of edges in H can be bounded by

$$\sum_{u\sim v} p_{u,v} = \sum_{u\sim v} \min\left\{1, C\cdot(\log n) q_{u,v}/\varepsilon^2\right\} \le \sum_{u\sim v} C\cdot(\log n) q_{u,v}/\varepsilon^2 \le Cn\log n/\varepsilon^2.$$

By Chernoff bound, it is exponentially unlikely that the number of edges in H is more than a small multiplicity factor of the expected value.

Why H is a spectra sparsifier? Now we prove that the sampled graph H is indeed a spectral sparsifier of G. First we notice that, for any positive definite matrices A

and B, it holds that $A \preceq B$ iff

$$B^{-1/2} A B^{-1/2} \preceq I.$$

Similarly, we have that $L_H \preceq L_G$ iff

$$L_G^{\dagger/2} L_H L_G^{\dagger/2} \preceq L_G^{\dagger/2} L_G L_G^{\dagger/2},$$

where $L_G^{\dagger/2}$ is the square root of the pseudo-inverse of L_G. Let

$$\Pi = L_G^{\dagger/2} L_G L_G^{\dagger/2}$$

be the projection onto the range of L_G. Then, by linearity of expectation it holds that

$$\mathbb{E}\left[L_G^{\dagger/2} L_H L_G^{\dagger/2}\right] = L_G^{\dagger/2} \mathbb{E}\left[L_H\right] L_G^{\dagger/2} = L_G^{\dagger/2} L_G L_G^{\dagger/2} = \Pi.$$

Now let us define a random matrix $X_{u,v}$, where

$$X_{u,v} = \begin{cases} \dfrac{w_{u,v}}{p_{u,v}} \cdot L_G^{\dagger/2} L_{u\sim v} L_G^{\dagger/2} & \text{with probability } p_{u,v} \\ 0 & \text{otherwise.} \end{cases}$$

Hence, it holds that

$$L_G^{\dagger/2} L_H L_G^{\dagger/2} = \sum_{u\sim v} X_{u,v}.$$

Notice that proving that H is a spectral sparsifier is equivalent to show that $\sum_{u\sim v} X_{u,v}$ is close to Π with high probability. By direct calculation, we have that

$$\begin{aligned} \|X_{u,v}\| &= (w_{u,v}/p_{u,v}) \cdot \left\| L_G^{\dagger/2} L_{u\sim v} L_G^{\dagger/2} \right\| \\ &= (w_{u,v}/p_{u,v}) \cdot \left\| L_G^{\dagger/2} (\delta_u - \delta_v)(\delta_u - \delta_v)^\intercal L_G^{\dagger/2} \right\| \\ &= (w_{u,v}/p_{u,v}) \cdot \operatorname{tr}\left(L_G^{\dagger/2} (\delta_u - \delta_v)(\delta_u - \delta_v)^\intercal L_G^{\dagger/2} \right) \\ &= (w_{u,v}/p_{u,v}) \cdot \operatorname{tr}\left((\delta_u - \delta_v)^\intercal L_G^{\dagger/2} L_G^{\dagger/2} (\delta_u - \delta_v) \right) \\ &= (w_{u,v}/p_{u,v}) \cdot \operatorname{tr}\left((\delta_u - \delta_v)^\intercal L_G^{\dagger} (\delta_u - \delta_v) \right) \\ &= (w_{u,v}/p_{u,v}) \cdot \mathsf{Reff}(u,v). \end{aligned}$$

By the definition of $p_{u,v}$, it holds that

$$\|X_{u,v}\| \leq \frac{\varepsilon^2}{C \cdot \log n}.$$

Notice that the upper bound above is independent of edge $u \sim v$. By Theorem 9.3, it holds that

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_{u\sim v} X_{u,v}\right) \geq (1+\varepsilon)\lambda_{\max}(\Pi)\right] \leq n^{-(C/3)+1},$$

which is small for $C > 3$.

Fast computation of the effective resistances. So far we showed the constructed H has $O(n \log n/\varepsilon^2)$ edges, and is indeed a $(1+\varepsilon)$-spectral sparsifier. To prove Theorem 9.1, it remains to analyse the runtime of the algorithm. Spielman and Srivastava show that the values $\{\mathsf{Reff}(u,v)\}_{u\sim v}$ with good approximation can be computed in $\widetilde{O}(m/\varepsilon^2)$ time. Due to time limit, we drop their algorithm for computing effective resistances here.

9.4 Useful Facts

Theorem 9.3 (Matrix Chernoff Bound)**.** *Let $X_1, \ldots, X_m \in \mathbb{R}^{n\times n}$ be independent random PSD such that $\|X_i\| \leq R$ almost surely. Let $X = \sum_{i=1}^m X_i$, and let $\mu_{\min}$ and $\mu_{\max}$ be the minimum and maximum eigenvalues of*

$$\mathbb{E}[X] = \sum_{i=1}^m \mathbb{E}[X_i].$$

Then,

$$\mathbb{P}\left[\lambda_{\min}\left(\sum_{i=1}^m X_i\right) \leq (1-\varepsilon)\mu_{\min}\right] \leq n\left(\frac{\mathrm{e}^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\mu_{\min}/R}$$

for $0 < \varepsilon < 1$, and

$$\mathbb{P}\left[\lambda_{\max}\left(\sum_{i=1}^m X_i\right) \leq (1-\varepsilon)\mu_{\max}\right] \geq n\left(\frac{\mathrm{e}^{-\varepsilon}}{(1-\varepsilon)^{1-\varepsilon}}\right)^{\mu_{\max}/R}$$

for $\varepsilon > 0$.

Lecture 10

Linear-Sized Spectral Sparsifiers

In the last lecture we have seen a nearly-linear time algorithm for constructing a $(1+\varepsilon)$-spectral sparsifier of $O(n \log n/\varepsilon^2)$ edges. In this lecture, we will see that, for any undirected graph G, a $(1+\varepsilon)$-spectral sparsifier of G with $O(n)$ edges exists, and can be constructed in $\widetilde{O}(m)$ time as well.

Without loss of generality, we study the algorithm of sparsifying the sum of rank-1 PSD matrices. Our goal is to, for any vectors $v_1, \cdots v_m$ with $\sum_{i=1}^m v_i v_i^\intercal = I$, find scalars $\{c_i\}_{i=1}^m$ satisfying

$$|\{c_i : c_i \neq 0\}| = O\left(\frac{n}{\varepsilon^2}\right),$$

such that

$$(1-\varepsilon) \cdot I \preceq \sum_{i=1}^m c_i v_i v_i^\intercal \preceq (1+\varepsilon) \cdot I.$$

Exercise 10.1. *Show that any algorithm for solving the problem above gives an algorithm for constructing a $(1+\varepsilon)$-spectral sparsifier of graphs that consists of $O(n/\varepsilon^2)$ edges.*

For simplicity, we treat $\varepsilon = O(1)$ in the lecture, and our goal is to prove the following theorem:

Theorem 10.2. *Let $v_1, \ldots, v_m$ be vectors in $\mathbb{R}^n$ such that $\sum_{i=1}^m v_i v_i^\intercal = I$. Then, for every $\varepsilon > 0$ there exists a set S along with scaling factors c_i such that*

$$(1 - O(1)) \cdot I \preceq \sum_{i \in S} c_i v_i v_i^\intercal \preceq (1 + O(1)) \cdot I,$$

and $|S| \leq \lceil n/\varepsilon^2 \rceil$. Moreover, this set S can be found in polynomial-time.

10.1 Overview of the Algorithm

Given the correspondence between PSD matrices and ellipsoids, the problem essentially asks to use $O(n)$ vectors from S to construct an ellipsoid, whose *shape* is close to be a sphere. To construct such an ellipsoid with desired shape, our algorithm proceeds by iterations: in each iteration j the algorithm chooses a vector v, and adds $\Delta_j \triangleq vv^\intercal$ to the currently constructed matrix by setting $A_j = A_{j-1} + \Delta_j$. To control the shape of the constructed ellipsoid, two barrier values, the *upper barrier* u_j and the *lower barrier* ℓ_j, are maintained such that the constructed ellipsoid $\mathsf{Ellip}(A_j)$ is sandwiched between the *outer* sphere $u_j \cdot I$ and the *inner* sphere $\ell_j \cdot I$ for any iteration j. That is, the following invariant always maintains:

$$\ell_j \cdot I \prec A_j \prec u_j \cdot I. \tag{10.1}$$

To ensure (10.1) holds, two barrier values ℓ_j and u_j are increased properly after each iteration, i.e.,

$$u_{j+1} = u_j + \Delta u, \qquad \ell_{j+1} = \ell_j + \Delta \ell$$

for some positive values Δu and $\Delta \ell$. We will continue this process, until after T iterations $\mathsf{Ellip}(A_T)$ is close to be a sphere. This implies that A_T approximates the identity matrix, see Figure 10.1 for an illustration.

To insure that that the invariant (10.1) always holds, we introduce the following potential functions: for any symmetric matrix $A \in \mathbb{R}^{n\times n}$ with eigenvalues $\lambda_1 \le \ldots \le \lambda_n$, we call $u \in \mathbb{R}$ an upper bound of A if $u > \lambda_n$. Given an upper bound u, we define the **upper barrier function** of A by

$$\Phi^u(A) = \sum_{i=1}^{n} \frac{1}{u - \lambda_i} = \operatorname{tr}(uI - A)^{-1}.$$

Similarly, for a lower bound $\ell < \lambda_1$, we define the **lower barrier function** of A by

$$\Phi_\ell(A) = \sum_{i=1}^{n} \frac{1}{\lambda_i - \ell} = \operatorname{tr}(A - \ell I)^{-1}.$$

10.2 The Changes of the Potential Functions

Given the intuitions above, what we need is to choose a proper vector $v \in \mathbb{R}^n$, and update the barrier values $u, \ell \in \mathbb{R}$ accordingly, so that the invariant always holds. Let us first look at the change of the potential functions when we add matrix $cvv^\intercal$ for some

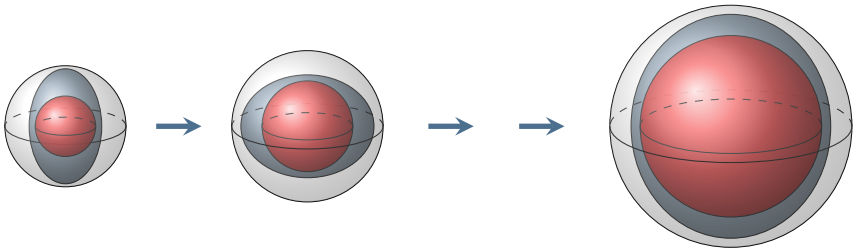

Figure 10.1: Illustration of the algorithms for constructing a linear-sized spectral sparsifier. Here, the light grey ball and the red ball in iteration j represent the spheres $u_j \cdot I$ and $\ell_j \cdot I$, and the blue ellipsoid sandwiched between the two balls corresponds to the constructed ellipsoid in iteration j. After each iteration j, the algorithm increases the value of ℓ_j and u_j by some $\Delta\ell$ and Δu so that the invariant (10.1) holds in iteration $j+1$. This process is repeated for T iterations, so that the final constructed ellipsoid is close to be a sphere.

$c \in \mathbb{R}$. By the Sherman-Morrison formula (Lemma 10.7), it holds that

$$\begin{aligned}
\Phi^u\left(A + cvv^\intercal\right) &= \operatorname{tr}\left(uI - A - cvv^\intercal\right)^{-1} \\
&= \operatorname{tr}(uI - A)^{-1} + c \cdot \frac{\operatorname{tr}\left((uI - A)^{-1}vv^\intercal(uI - A)^{-1}\right)}{1 - cv^\intercal(uI - A)^{-1}v} \\
&= \Phi^u(A) + c \cdot \frac{\operatorname{tr}\left(v^\intercal(uI - A)^{-1}(uI - A)^{-1}v\right)}{1 - cv^\intercal(uI - A)^{-1}v} \\
&= \Phi^u(A) + c \cdot \frac{v^\intercal(uI - A)^{-2}v}{1 - cv^\intercal(uI - A)^{-1}v}.
\end{aligned}$$

When we increase u by Δu, by setting $u' = u + \Delta u$ we have that

$$\begin{aligned}
\Phi^{u'}\left(A + cvv^\intercal\right) &= \Phi^{u'}(A) + c \cdot \frac{v^\intercal(u'I - A)^{-2}v}{1 - cv^\intercal(u'I - A)^{-1}v} \\
&= \Phi^u(A) - \left(\Phi^u(A) - \Phi^{u'}(A)\right) + c \cdot \frac{v^\intercal(u'I - A)^{-2}v}{1 - cv^\intercal(u'I - A)^{-1}v}.
\end{aligned}$$

Remember that our goal is to make the barrier functions bounded after each iteration. To achieve it, let us assume that

$$\Phi^{u'}\left(A + cvv^\intercal\right) \leq \Phi^u(A),$$

which is equivalent to say that

$$\left(\Phi^u(A) - \Phi^{u'}(A)\right) \geq c \cdot \frac{v^\intercal(u'I - A)^{-2}v}{1 - cv^\intercal(u'I - A)^{-1}v},$$

i.e.,

$$\frac{1}{c} \geq \frac{v^\intercal (u'I - A)^{-2} v}{\Phi^u(A) - \Phi^{u'}(A)} + v^\intercal (u'I - A)^{-1} v.$$

Let us define

$$U_A = \frac{(u'I - A)^{-2}}{\Phi^u - \Phi^{u'}(A)} + (u'I - A)^{-1}.$$

Lemma 10.3. *If $1/c \geq v^\intercal U_A v$, then it holds that*

$$\Phi^{u'}\left(A + cvv^\intercal\right) \leq \Phi^u(A).$$

By the same analysis, we can analyse the change of Φ_ℓ. We set $\ell' = \ell + \Delta$, and

$$L_A = \frac{(A - \ell' I)^{-1}}{\Phi_{\ell'}(A) - \Phi_\ell(A)} - (A - \ell' I)^{-1}.$$

We have the following result.

Lemma 10.4. *If $1/c \leq v^\intercal L_A v$, then it holds that*

$$\Phi_{\ell'}\left(A + cvv^\intercal\right) \leq \Phi_\ell(A).$$

It remains to show that there exits a vector v and a scaling factor c such that

$$\Phi^{u'}(A + cvv^\intercal) \leq \Phi^u(A),$$

and

$$\Phi_{\ell'}(A + cvv^\intercal) \leq \Phi_\ell(A).$$

That is, we need to show the existence of $c \in \mathbb{R}$ such that

$$v_i^\intercal U_A v_i \leq 1/c \leq v_i^\intercal L_A v_i$$

Lemma 10.5. *It holds that*

$$\sum_{i=1}^{m} v_i^\intercal U_A v_i \leq \frac{1}{\Delta u} + \Phi^u(A).$$

Proof. Notice that

$$\sum_{i=1}^{m} v_i^\intercal U_A v_i = \mathrm{tr}(U_A) = \mathrm{tr}\left(\frac{(u'I - A)^{-2}}{\Phi^u - \Phi^{u'}(A)}\right) + \mathrm{tr}\left((u'I - A)^{-1}\right).$$

For the second term, notice that

$$\mathrm{tr}\left((u'I - A)^{-1}\right) = \Phi^{u'}(A) \leq \Phi^u(u). \tag{10.2}$$

Now we analyse the first term. It holds that

$$\frac{\partial}{\partial u}\Phi^u(A) = \frac{\partial}{\partial u}\sum_{i=1}^{n}\frac{1}{u-\lambda_i} = -\sum_{i=1}^{n}\left(\frac{1}{u-\lambda_i}\right)^2 = -\operatorname{tr}\left((uI-A)^{-2}\right).$$

Since $\Phi^u(A)$ is convex in u, we have that

$$\Phi^u(A) - \Phi^{u+\Delta u}(A) \geq -\Delta u \cdot \frac{\partial}{\partial u}\Phi^u(A) = \Delta u \cdot \operatorname{tr}\left((uI-A)^{-2}\right). \tag{10.3}$$

Combining (10.2) with (10.3) proves the claimed statement. □

The analysis for $\sum_{i=1}^{m} v_i^\intercal L_A v_i$ is similar, and we have the following result.

Lemma 10.6. *It holds that*

$$\sum_{i=1}^{m} v_i^\intercal L_A v_i \geq \frac{1}{\Delta\ell} - \frac{1}{1/\Phi_\ell(A) - \Delta\ell}.$$

Now, if we further assume that

$$\frac{1}{\Delta u} + \Phi^u(A) \leq \frac{1}{\Delta\ell} - \frac{1}{1/\Phi_\ell(A) - \Delta\ell},$$

then by Lemma 10.5 and Lemma 10.6 we know the existence of some i such that

$$v_i^\intercal U_A v_i \leq v_i^\intercal L_A v_i,$$

and the existence of some $c \in \mathbb{R}$ such that

$$v_i^\intercal U_A v_i \leq \frac{1}{c} \leq v_i^\intercal L_A v_i.$$

Hence, in each iteration we can always find some vector $v_i \in \mathbb{R}^n$ and some scaling factor $c \in \mathbb{R}$ such that the potential functions are always bounded during the execution of the algorithm.

10.3 Implementation of the Algorithm

The choice of parameters. We set the initial value of u and ℓ by

$$\ell = -n, \qquad u = n.$$

After each iteration, the values of u and ℓ will be increased by 2 and $1/3$ respectively. Hence, after $6n$ iteration, the condition number of our constructed matrix is upper bounded by

$$\frac{n + 2\cdot 6n}{-n + 6n/3} = 13,$$

which implies that the constructed matrix approximates the identity matrix.

Runtime analysis. The algorithm takes $O(n)$ iterations. In each iteration, the algorithm needs to compute $((u + \Delta u)I - A)^{-1}$, $((u + \Delta u)I - A)^{-2}$, and the same matrices for the lower potential functions. This quantities can be computed in time $O(n^3)$. We also need to decided which edge to add in each iteration by computing $U_A v_i$ and $L_A v_i$ for each edge, which can be computed in time $O(n^2 m)$. Since we need to run for $O(n)$ iterations, the total time of the algorithm is $O(n^3 m)$.

10.4 Recent Progress

There have been several important results for fast constructions of linear-sized spectral sparsifers. Recently, Lee and Sun showed that a linear-sized spectral sparsifier can be constructed in nearly-linear time. At a very high-level, their algorithm successfully breaks the runtime barriers of the algorithm discussed above:

- Lee-Sun algorithm runs for $O(\text{poly}\log n)$ iterations, instead of $\Omega(n)$ iterations discussed above.
- In each iteration, Lee-Sun algorithm picks $O(n/\text{poly}\log n)$ vectors by solving a SDP program.
- Through a refined potential function, they showed that the constructed matrix is never close to the barrier values. Hence, many quantities can be computed in linearly-time through Taylor expansion of matrices.

10.5 Useful Facts

Lemma 10.7 (Sherman-Morrison)**.** *Let A be a nonsingular symmetric matrix, v be a vector and let $c \in \mathbb{R}$. Then, it holds that*

$$(A - cvv^\intercal)^{-1} = A^{-1} + c \cdot \frac{A^{-1}vv^\intercal A^{-1}}{1 - cv^\intercal A^{-1}v}.$$