Spectral and Algebraic Graph Theory
Incomplete Draft, dated December 4, 2019

Current version available at http://cs-www.cs.yale.edu/homes/spielman/sagt.

Daniel A. Spielman
Yale University

Chapter List

Preface

Please note that this is a rapidly evolving draft.

This book is about how combinatorial properties of graphs are related to algebraic properties of associated matrices, as well as applications of those connections. One's initial excitement over this material usually stems from its counter-intuitive nature. I hope to convey this initial amazement, but then make the connections seem intuitive. After gaining intuition, I hope the reader will appreciate the material for its beauty.

This book is mostly based on lecture notes from the "Spectral Graph Theory" course that I have taught at Yale, with notes from "Graphs and Networks" and "Spectral Graph Theory and its Applications" mixed in. I love the material in these courses, and find that I can never teach everything I want to cover within one semester. This is why I am have written this book. As this book is based on lecture notes, it does not contain the tightest or most recent results. Rather, my goal is to introduce the main ideas and to provide intuition.

There are three tasks that one must accomplish in the beginning of a course on Spectral Graph Theory:

- One must convey how the coordinates of eigenvectors correspond to vertices in a graph. This is obvious to those who understand it, but it can take a while for students to grasp.
- One must introduce necessary linear algebra and show some interesting interpretations of graph eigenvalues.
- One must derive the eigenvalues of some example graphs to ground the theory.

I find that one has to do all these at once. For this reason my first few lectures jump between developing theory and examining particular graphs. For this book I have decided to organize the material differently, mostly separating examinations of particular graphs from the development of the theory. To help the reader reconstruct the flow of my courses, I give three orders that I have used for the material:

put orders here

There are many terrific books on Spectral Graph Theory. The four that influenced me the most are

"Algebraic Graph Theory" by Norman Biggs, "Spectral Graph Theory" by Fan Chung,

"Algebraic Combinatorics" by Chris Godsil, and

"Algebraic Graph Theory" by Chris Godsil and Gordon Royle.

Other books that I find very helpful and that contain related material include

"Modern Graph Theory" by Bela Bollobas,

"Probability on Trees and Networks" by Russell Llyons and Yuval Peres,

"Spectra of Graphs" by Dragos Cvetkovic, Michael Doob, and Horst Sachs, and

"Eigenspaces of Graphs" By Dragos Cvetkovic, Peter Rowlinson, and Slobodan Simic"

"Non-negative Matrices and Markov Chains" by Eugene Seneta

"Nonnegative Matrices and Applications" by R. B. Bapat and T. E. S. Raghavan

"Numerical Linear Algebra" by Lloyd N. Trefethen and David Bau, III

"Applied Numerical Linear Algebra" by James W. Demmel

For those needing an introduction to linear algebra, a perspective that is compatible with this book is contained in Gil Strang's "Introduction to Linear Algebra." For more advanced topics in linear algebra, I recommend "Matrix Analysis" by Roger Horn and Charles Johnson, as well as their "Topics in Matrix Analysis" For treatments of physical systems related to graphs, the topic of Part III, I recommend Gil Strang's "Introduction to Applied Mathematics", Sydney H. Gould's "Variational Methods for Eigenvalue Problems", and "Markov Chains and Mixing Times" by Levin, Peres and Wilmer.

I include some example in these notes. All of these have been generated inside Jupyter notebooks using the Julia language. Some of them require use of the package Laplacians.jl. A simple search will produce good instructions for installing Julia and packages for it. The notebooks used in this book may be found at http://cs-www.cs.yale.edu/homes/spielman/sagt.

Contents

Notation

This section lists the notation that I try to use throughout the book. I sometimes fall into different notations when the conventions surrounding a result are so strong that failing to follow them would make it difficult for experts to understand this book, or would cause cognitive stress.

I almost always treat vectors as functions, and thus write $\boldsymbol{x}(i)$ for the ith component of the vector $\boldsymbol{x}$. I place subscripts on vectors, like $\boldsymbol{x}_i$, to indicate the ith vector in a set of vectors.

$\boldsymbol{I}$	An identity matrix
$\boldsymbol{J}$	An all-1s matrix
$\boldsymbol{D}$	The diagonal matrix of weighted degrees of a graph
$\boldsymbol{L}$	Laplacian Matrix
$\boldsymbol{M}$	Adjacency Matrix or Generic Matrix
$\boldsymbol{N}$	Normalized Laplacian Matrix
$\boldsymbol{W}$	The diagonal matrix of edge weights, or
$\boldsymbol{W}$	The Walk Matrix, $\boldsymbol{M}\boldsymbol{D}^{-1}$
$\widetilde{\boldsymbol{W}}$	Lazy Walk Matrix, $\boldsymbol{I}/2 + \boldsymbol{W}/2$
$\boldsymbol{A}^{+}$	The Moore-Penrose pseudoinverse of $\boldsymbol{A}$.
$\boldsymbol{A}^{+/2}$	The square root of of $\boldsymbol{A}^{+}$.

V	The set of vertices
E	The set of edges
n	usually the number of vertices in a graph
a, b	vertices
(a, b)	an edge
$\boldsymbol{x}(a)$	the component of the vector $\boldsymbol{x}$ corresponding to vertex a.
$\boldsymbol{d}$	the vector of weighted degrees of the vertices in a graph.
$\boldsymbol{d}(a)$	the weighted degree of vertex a
$w(F)$	the sum of the weights of edges in $F \subset E$
λ	an eigenvalue, usually of a graph Laplacian
μ	an eigenvalue, usually of an adjacency matrix
ν	an eigenvalue of a normalized Laplacian matrix
ω	an eigenvalue of a walk matrix
$\boldsymbol{\psi}, \boldsymbol{\phi}$	eigenvectors
$\boldsymbol{\psi}_i$	the ith eigenvector, associated with eigenvalue λ_i
$\boldsymbol{\delta}_a$	the elementary unit vector in direction a; $\boldsymbol{\delta}_a(a) = 1$
$\boldsymbol{\Psi}$	the orthogonal matrix with columns $\boldsymbol{\psi}_i$
$\boldsymbol{\pi}$	the stable distribution of a random walk
$\lambda_{max}(\boldsymbol{M})$	the largest eigenvalue of $\boldsymbol{M}$
$\lambda_{min}(\boldsymbol{M})$	the smallest eigenvalue of $\boldsymbol{M}$
$\mu_k(\boldsymbol{M})$	the kth largest eigenvalue of $\boldsymbol{M}$
$\lambda_k(\boldsymbol{L})$	the kth smallest eigenvalue of $\boldsymbol{L}$
$\\|\boldsymbol{M}\\|$	the operator norm of the matrix $\boldsymbol{M}$
$\\|\boldsymbol{x}\\|$	the Hamming weight of the vector $\boldsymbol{x}$

$a \sim b$	a is a neighbor of b
$G(S)$	the subgraph induced on the vertices in S
$\chi(G)$	the chromatic number of the graph G
$\alpha(G)$	the independence number of the graph G
$\partial(S)$	the boundary of a set of vertices
$\theta(S)$	the isoperimetric ratio of S
$\phi(S)$	the conductance of S
θ_G	the isoperimetric ratio of G
ϕ_G	the conductance of G
$\boldsymbol{A} \succcurlyeq \boldsymbol{B}$	$\boldsymbol{A} - \boldsymbol{B}$ is positive semidefinite
$G \succcurlyeq H$	$\boldsymbol{L}_G \succcurlyeq \boldsymbol{L}_H$
$[n]$	the set $\{1, 2, \ldots, n\}$
$\mathbb{F}_p$	the field with p elements, aka the integers modulo the prime p

Part I

Introduction and Background

Chapter 1

Introduction

In this chapter we present essential background on graphs and spectral theory. We also provide a brief introduction to some of the ideas of spectral graph theory, describe some of the topics covered in this book, and try to give some useful intuition about graph spectra.

1.1 Graphs

First, we recall that a graph $G = (V, E)$ is specified by its vertex[1] set, V, and edge set E. In an undirected graph, the edge set is a set of unordered pairs of vertices. Unless otherwise specified, all graphs will be undirected, simple (having no loops or multiple edges) and finite. We will sometimes assign weights to edges. These will usually be positive real numbers. If no weights have been specified, we view all edges as having weight 1. This is an arbitrary choice, and we should remember that it has an impact.

Graphs (also called "networks") are typically used to model connections or relations between things, where "things" are vertices. When the edges in a graph are more important than the vertices, we may just specify an edge set E and ignore the ambient vertex set.

Common "natural" examples of graphs are:

- Friendship graphs: people are vertices, edges exist between pairs of people who are friends (assuming the relation is symmetric).
- Network graphs: devices, routers and computers are vertices, edges exist between pairs that are connected.
- Circuit graphs: electronic components, such as transistors, are vertices: edges exist between pairs connected by wires.
- Protein-Protein Interaction graphs: proteins are vertices. Edges exist between pairs that

[1] I will use the words "vertex" and "node" interchangeably. Sorry about that.

interact. These should really have weights indicating the strength and nature of interaction. So should most other graphs.

It is much easier to study abstract, mathematically defined graphs. For example,

- The path on n vertices. The vertices are $\{1, \ldots n\}$. The edges are $(i, i+1)$ for $1 \leq i < n$.
- The ring on n vertices. The vertices are $\{1, \ldots n\}$. The edges are all those in the path, plus the edge $(1, n)$.
- The hypercube on 2^k vertices. The vertices are elements of $\{0, 1\}^k$. Edges exist between vertices that differ in only one coordinate.

1.2 Matrices for Graphs

The naive view of a matrix is that it is essentially a spreadsheet—a table we use to organize numbers. This is like saying that a car is an enclosed metal chair with wheels. It says nothing about what it does!

We will use matrices to do two things. First, we will view a matrix $\boldsymbol{M}$ as providing a function that maps a vector $\boldsymbol{x}$ to the vector $\boldsymbol{M}\boldsymbol{x}$. That is, we view $\boldsymbol{M}$ as an operator. Second, we use the matrix $\boldsymbol{M}$ to define a quadratic form: a function that maps a vector $\boldsymbol{x}$ to a number $\boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x}$.

1.2.1 A spreadsheet

We will usually write V for the set of vertices of a graph, and let n denote the number of vertices. There are times that we will need to order the vertices and assign numbers to them. In this case, they will usually be $\{1, \ldots, n\}$. For example, if we wish to draw a matrix as a table, then we need to decide which vertex corresponds to which row and column.

The most natural matrix to associate with a graph G is its adjacency matrix[2], $\boldsymbol{M}_G$, whose entries $\boldsymbol{M}_G(a, b)$ are given by

$$\boldsymbol{M}_G(a, b) = \begin{cases} 1 & \text{if } (a, b) \in E \\ 0 & \text{otherwise.} \end{cases}$$

It is important to realize that we index the rows and columns of the matrix by vertices, rather than by numbers. Almost every statement that we make will remain true under renaming of vertices. The first row of a matrix has no special importance. To understand this better see the exercises at the end of this section.

While the adjacency matrix is the most natural matrix to associate with a graph, I find it the least useful. Eigenvalues and eigenvectors are most meaningful when used to understand a natural operator or a natural quadratic form. The adjacency matrix provides neither.

[2] I am going to try to always use the letter $\boldsymbol{M}$ for the adjacency matrix, in contrast with my past practice which was to use $\boldsymbol{A}$. I will use letters like a and b to denote vertices.

1.2.2 An operator

The most natural operator associated with a graph G is probably its diffusion operator. This operator describes the diffusion of stuff among the vertices of a graph. Imagine a process in which each vertex can contain some amount of stuff (such as a gas). At each time step, the stuff at a vertex will be uniformly distributed to its neighbors. None of the stuff that was at a vertex remains at the vertex, but stuff can enter from other vertices. This is a discrete-time and slightly unnatural notion of diffusion, but it provides a nice matrix.

To construct the diffusion matrix, let $\boldsymbol{D}_G$ be the diagonal matrix in which $\boldsymbol{D}_G(a,a)$ is the degree of vertex a. We will usually write $\boldsymbol{d}(a)$ for the degree of vertex a. In an unweighted graph, the degree of a vertex is the number of edges attached to it. In the case of a weighted graph, we use the *weighted degree*: the sum of the weights of the edges attached to the vertex a. Algebraically, we can obtain the vector of degrees from the expression

$$\boldsymbol{d} \stackrel{\text{def}}{=} \boldsymbol{M}_G \mathbf{1},$$

where $\mathbf{1}$ is the all-ones vector.

We then set

$$\boldsymbol{W}_G = \boldsymbol{M}_G \boldsymbol{D}_G^{-1}.$$

Of course, when the graph is *regular*, that is when every vertex has the same degree, $\boldsymbol{W}_G$ is merely a rescaling of $\boldsymbol{M}_G$[3].

Formally[4], we use a vector $\boldsymbol{p} \in \mathbb{R}^V$ to indicate how much stuff is at each vertex, with $\boldsymbol{p}(a)$ being the amount of stuff at vertex a. After one time step, the distribution of stuff at each vertex will be $\boldsymbol{W}_G\boldsymbol{p}$. To see this, first consider the case when $\boldsymbol{p}$ is an elementary unit vector, $\boldsymbol{\delta}_a$, where we define $\boldsymbol{\delta}_a$ to be the vector for which $\boldsymbol{\delta}_a(a) = 1$, and for every other vertex b, $\boldsymbol{\delta}_a(b) = 0$. The vector $\boldsymbol{D}_G^{-1}\boldsymbol{\delta}_a$ has the value $1/\boldsymbol{d}(a)$ at vertex a, and is zero everywhere else. So, the vector $\boldsymbol{M}_G\boldsymbol{D}_G^{-1}\boldsymbol{\delta}_a$ has value $1/\boldsymbol{d}(a)$ at every vertex b that is a neighbor of a, and is zero everywhere else. If this is not immediately obvious, think about it until it is.

It is sometimes more convenient to consider a *lazy random walk* . These are usually defined to be walks that stay put with probability one half and take a step with probability one half. The matrix corresponding to this operator is given by

$$\widetilde{\boldsymbol{W}}_G \stackrel{\text{def}}{=} \boldsymbol{I}/2 + \boldsymbol{W}_G/2.$$

One of the purposes of spectral theory is to provide an understanding of what happens when one repeatedly applies a linear operator like $\boldsymbol{W}_G$.

[3] I think this is why researchers got away with studying the adjacency matrix for so long.

[4] We write $\mathbb{R}^V$ instead of $\mathbb{R}^n$ to emphasize that each coordinate of the vector corresponds to a vertex of the graph.

1.2.3 A quadratic form

The most natural quadratic form associated with a graph is defined in terms of its Laplacian matrix,

$$\boldsymbol{L}_G \stackrel{\text{def}}{=} \boldsymbol{D}_G - \boldsymbol{M}_G.$$

Given a function on the vertices, $\boldsymbol{x} \in \mathbb{R}^V$, the Laplacian quadratic form of a weighted graph in which edge (a, b) has weight $w_{a,b} > 0$ is

$$\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x} = \sum_{(a,b)\in E} w_{a,b} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2. \tag{1.1}$$

This form measures the smoothness of the function $\boldsymbol{x}$. It will be small if the function $\boldsymbol{x}$ does not jump too much over any edge.

We use the notation $\boldsymbol{x}(a)$ to denote the coordinate of vector $\boldsymbol{x}$ corresponding to vertex a. Other people often use subscripts for this, like $\boldsymbol{x}_a$. We usually use subscripts to name vectors.

There are many possible definitions of Laplacians with negative edge weights. So, we will only define them when we need them.

1.3 Spectral Theory

We now review the highlights of the spectral theory for symmetric matrices. Almost all of the matrices we consider will be symmetric or will be similar[5] to symmetric matrices.

We recall that a vector $\boldsymbol{\psi}$ is an eigenvector of a matrix $\boldsymbol{M}$ with eigenvalue λ if

$$\boldsymbol{M}\boldsymbol{\psi} = \lambda\boldsymbol{\psi}. \tag{1.2}$$

That is, λ is an eigenvalue if and only if $\lambda\boldsymbol{I} - \boldsymbol{M}$ is a singular matrix. Thus, the eigenvalues are the roots of the characteristic polynomial of $\boldsymbol{M}$:

$$\det(x\boldsymbol{I} - \boldsymbol{M}).$$

Theorem 1.3.1. *[The Spectral Theorem] If $\boldsymbol{M}$ is an n-by-n, real, symmetric matrix, then there exist real numbers $\lambda_1, \ldots, \lambda_n$ and n mutually orthogonal unit vectors $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$ and such that $\boldsymbol{\psi}_i$ is an eigenvector of $\boldsymbol{M}$ of eigenvalue λ_i, for each i.*

This is the great fact about symmetric matrices. If the matrix is not symmetric, it might not have n eigenvalues. And, even if it has n eigenvalues, their eigenvectors will not be orthogonal[6]. If $\boldsymbol{M}$ is not symmetric, its eigenvalues and eigenvalues might be the wrong thing to study.

[5] A matrix $\boldsymbol{M}$ is similar to a matrix $\boldsymbol{B}$ if there is a non-singular matrix $\boldsymbol{X}$ such that $\boldsymbol{X}^{-1}\boldsymbol{M}\boldsymbol{X} = \boldsymbol{B}$. In this case, $\boldsymbol{M}$ and $\boldsymbol{B}$ have the same eigenvalues. See the exercises at the end of this section.

[6] You can prove that if the eigenvectors are orthogonal, then the matrix is symmetric.

Recall that the eigenvectors are not uniquely determined, although the eigenvalues are. If $\boldsymbol{\psi}$ is an eigenvector, then $-\boldsymbol{\psi}$ is as well. Some eigenvalues can be repeated. If $\lambda_i = \lambda_{i+1}$, then $\boldsymbol{\psi}_i + \boldsymbol{\psi}_{i+1}$ will also be an eigenvector of eigenvalue λ_i. The eigenvectors of a given eigenvalue are only determined up to an orthogonal transformation.

Definition 1.3.2. *A matrix is positive definite if it is symmetric and all of its eigenvalues are positive. It is positive semidefinite if it is symmetric and all of its eigenvalues are nonnegative.*

Fact 1.3.3. *The Laplacian matrix of a graph is positive semidefinite.*

Proof. Let $\boldsymbol{\psi}$ be a unit eigenvector of $\boldsymbol{L}$ of eigenvalue λ. Then,

$$\boldsymbol{\psi}^T \boldsymbol{L} \boldsymbol{\psi} = \boldsymbol{\psi}^T \lambda \boldsymbol{\psi} = \lambda = \sum_{(a,b) \in E} w_{a,b} (\boldsymbol{\psi}(a) - \boldsymbol{\psi}(b))^2 \geq 0.$$

□

We always number the eigenvalues of the Laplacian from smallest to largest. Thus, $\lambda_1 = 0$. We will refer to λ_2, and in general λ_k for small k, as *low-frequency* eigenvalues. λ_n is a *high-frequency* eigenvalue. We will see why soon.

1.4 Some examples

Before we get to any theorems, we will examine evidence that the eigenvalues and eigenvectors of graphs are meaningful by looking at some examples. These were produced in Julia using a Jupyter notebook. You may find the notebook on the book homepage.

1.4.1 Paths

A path graph has vertices $\{1, \ldots, n\}$ and edges $(i, i+1)$ for $1 \leq i < n$. Here is the adjacency matrix of a path graph on 4 vertices.

```
M = path_graph(4)
Matrix(M)
 0.0  1.0  0.0  0.0
 1.0  0.0  1.0  0.0
 0.0  1.0  0.0  1.0
 0.0  0.0  1.0  0.0
```

And, here is its Laplacian matrix

```
Matrix(lap(M))
  1.0  -1.0   0.0   0.0
```

```
 -1.0   2.0  -1.0   0.0
  0.0  -1.0   2.0  -1.0
  0.0   0.0  -1.0   1.0
```

Here are the eigenvalues of a longer path.

```
L = lap(path_graph(10))
E = eigen(Matrix(L))
println(E.values)
```

```
[0.0, 0.097887, 0.381966, 0.824429, 1.38197, 2.0, 2.61803, 3.17557, 3.61803, 3.90211]
```

The eigenvector of the zero-eigenvalue is a constant vector (up to numerical issues):

```
E.vectors[:,1]
```

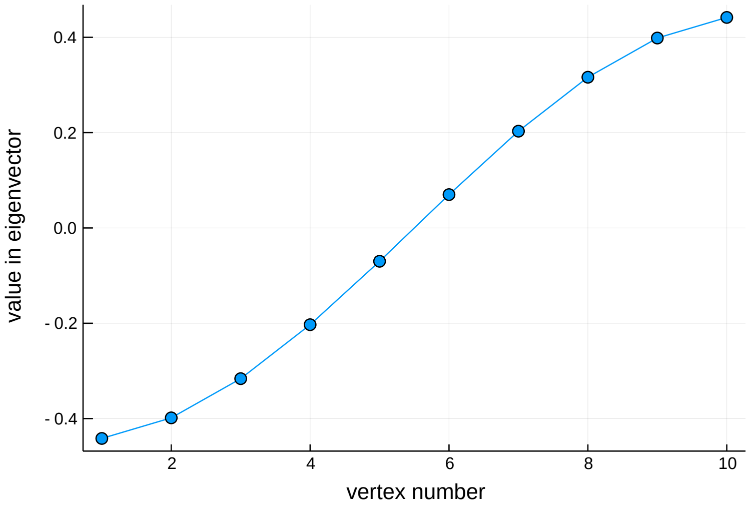
```
 0.31622776601683755
 0.31622776601683716
 0.31622776601683766
 0.3162277660168381
 0.31622776601683855
 0.3162277660168381
 0.3162277660168385
 0.31622776601683805
 0.3162277660168378
 0.3162277660168378
```

The eigenvector of λ_2 is the lowest frequency eigenvector, as we can see that it increases monotonically along the path:

```
v2 = E.vectors[:,2]
```

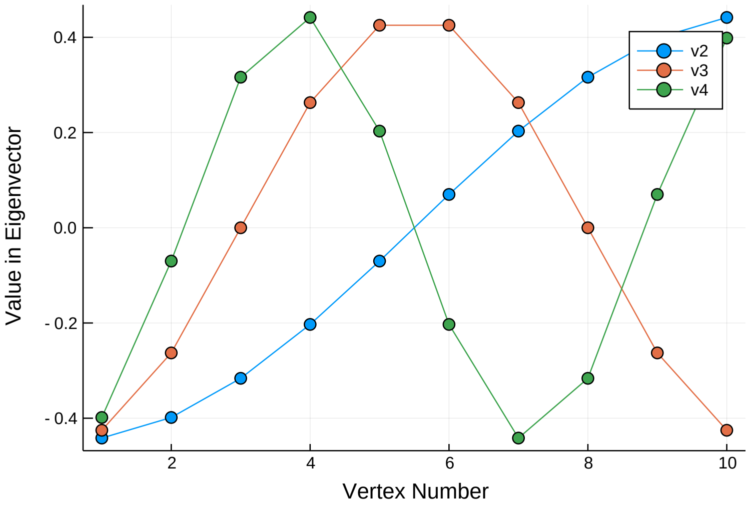
```
 -0.44170765403093937
 -0.39847023129620024
 -0.316227766016838
 -0.20303072371134553
 -0.06995961957075425
  0.06995961957075386
  0.2030307237113457
  0.31622776601683766
  0.3984702312961997
  0.4417076540309382
```

Let’s plot that.



```
plot(v2,marker=5,legend=false)
xlabel!("vertex number")
ylabel!("value in eigenvector")
```

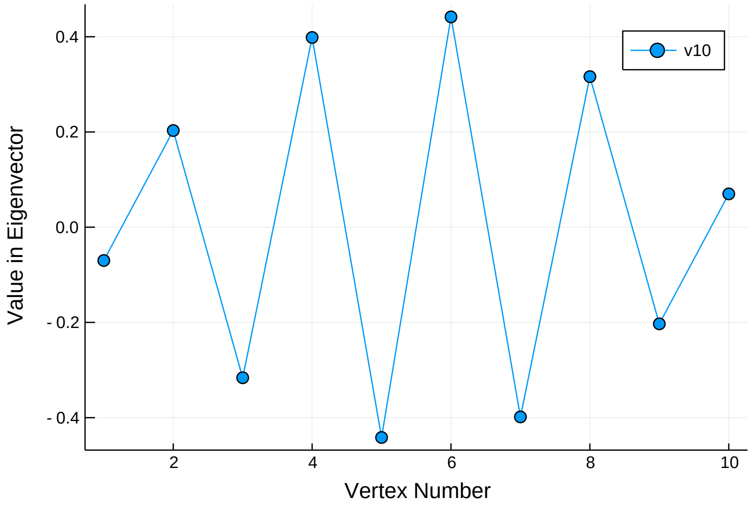
The x-axis is the name/number of the vertex, and the y-axis is the value of the eigenvector at that vertex. Now, let's look at the next few eigenvectors.



```
Plots.plot(E.vectors[:,2],label="v2",marker = 5)
Plots.plot!(E.vectors[:,3],label="v3",marker = 5)
Plots.plot!(E.vectors[:,4],label="v4",marker = 5)
xlabel!("Vertex Number")
ylabel!("Value in Eigenvector")
```

You may now understand why we refer to these as the low-frequency eigenvectors. The curves they trace out resemble the low-frequency modes of vibration of a string. The reason for this is that the path graph can be viewed as a discretization of the string, and its Laplacian matrix is a discretization of the Laplace operator. We will relate the low-frequency eigenvalues to connectivity.

In contrast, the highest frequency eigenvalue alternates positive and negative with every vertex. We will see that the high-frequency eigenvectors may be related to problems of graph coloring and finding independent sets.



```
Plots.plot(E.vectors[:,10],label="v10",marker=5)
xlabel!("Vertex Number")
ylabel!("Value in Eigenvector")
```

1.5 Highlights

We now attempt to motivate this book, and the course on which it is based, by surveying some of its highlights.

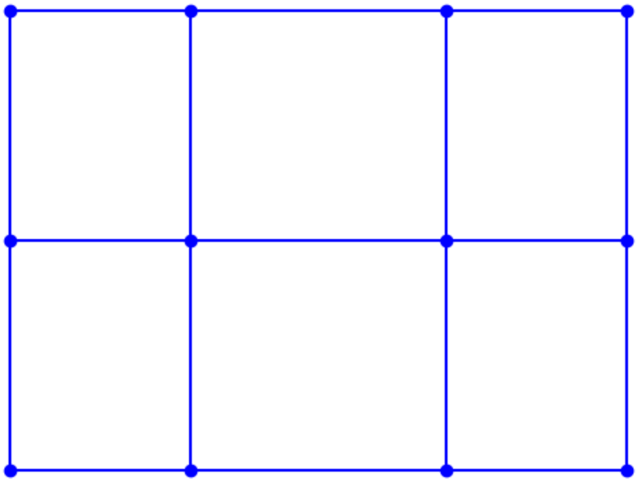
1.5.1 Spectral Graph Drawing

We can often use the low-frequency eigenvalues to obtain a nice drawing of a graph. For example, here is 3-by-4 grid graph, and its first two non-trivial eigenvectors. Looking at them suggests that they might provide nice coordinates for the vertices.

```
M = grid2(3,4)
L = lap(M)
E = eigen(Matrix(L))
V = E.vectors[:,2:3]
```

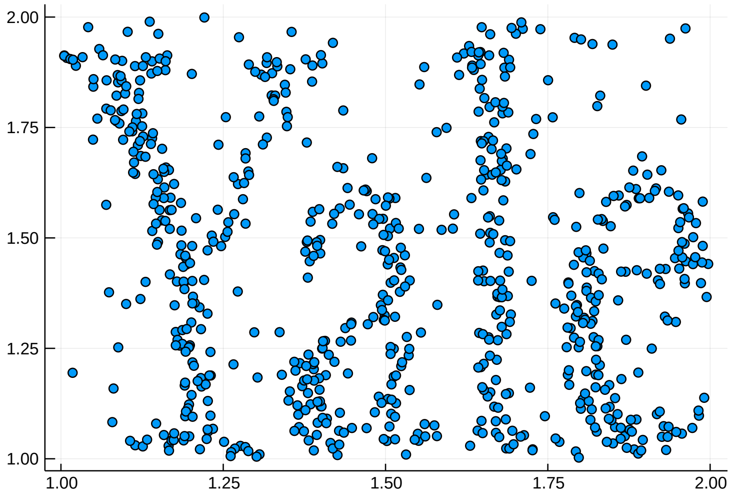
```
-0.377172   0.353553
-0.15623    0.353553
 0.15623    0.353553
 0.377172   0.353553
-0.377172  -1.66533e-16
-0.15623   -4.16334e-16
 0.15623   -5.82867e-16
 0.377172   2.77556e-16
-0.377172  -0.353553
-0.15623   -0.353553
 0.15623   -0.353553
 0.377172  -0.353553
```

In the figure below, we use these eigenvectors to draw the graph. Vertex a be been plotted at coordinates $\boldsymbol{\psi}_2(a), \boldsymbol{\psi}_3(a)$. That is, we use $\boldsymbol{\psi}_2$ to provide a horizontal coordinate for every vertex, and $\boldsymbol{\psi}_3$ to obtain a vertical coordinate. We then draw the edges as straight lines.



```
plot_graph(M,V[:,1],V[:,2])
```

Let's do a fancier example that should convince you something interesting is going on. We begin by generating points by sampling them from the Yale logo.

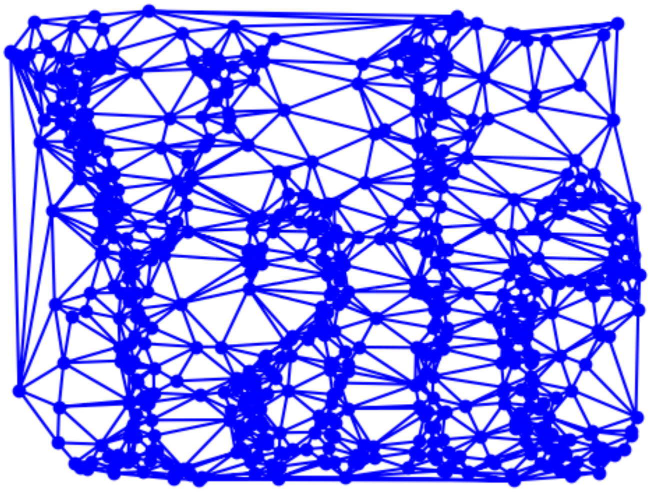


```
@load "yale.jld2"
scatter(xy[:,1],xy[:,2],legend=false)
```

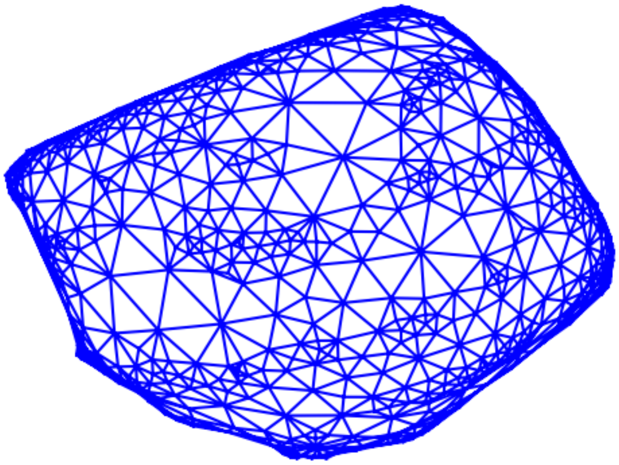
We then construct a graph on the points by forming their Delaunay triangulation[7], and use the edges of the triangles to define a graph on the points.

Since the vertices came with coordinates, it was easy to draw a nice picture of the graph. But, what if we just knew the graph, and not the coordinates? We could generate coordinates by computing two eigenvectors, and using each as a coordinate. Below, we plot vertex a at position $\boldsymbol{\psi}_2(a), \boldsymbol{\psi}_3(a)$, and again draw the edges as straight lines.

[7] While it does not make sense to cover Delaunay triangulations in this book, they are fascinating and I recommend that you look them up.



```
plot_graph(a,xy[:,1],xy[:,2])
```



```
plot_graph(a, v2,v3, dots=false)
```

That's a great way to draw a graph if you start out knowing nothing about it[8]. Note that the middle of the picture is almost planar, although edges do cross near the boundaries.

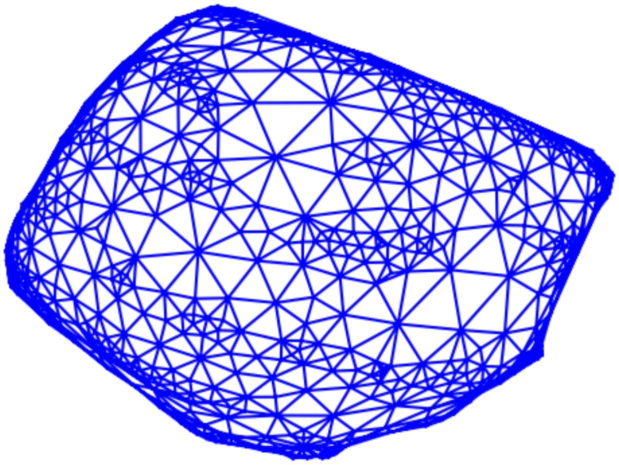
[8] It's the first thing I do whenever I meet a strange graph.

1.5.2 Graph Isomorphism

It is important to note that the eigenvalues do not change if we relabel the vertices. Moreover, if we permute the vertices then the eigenvectors are similarly permuted. That is, if $\boldsymbol{P}$ is a permutation matrix, then

$$\boldsymbol{L}\boldsymbol{\psi} = \lambda\boldsymbol{\psi} \quad \text{if and only if} \quad (\boldsymbol{P}\boldsymbol{L}\boldsymbol{P}^T)(\boldsymbol{P}\boldsymbol{\psi}) = \lambda(\boldsymbol{P}\boldsymbol{\psi}),$$

because $\boldsymbol{P}^T\boldsymbol{P} = \boldsymbol{I}$. To prove it by experiment, let's randomly permute the vertices, and plot the permuted graph.



```
Random.seed!(1)
p = randperm(size(a,1))
M = a[p,p]
E = eigen(Matrix(lap(M)))
V = E.vectors[:,2:3]
plot_graph(M,V[:,1],V[:,2], dots=false)
```

Note that this picture is slightly different from the previous one: it has flipped vertically. That's because eigenvectors are only determined up to signs, and that's only if they have multiplicity 1. This gives us a very powerful heuristic for testing if one graph is a permutation of another (this is the famous "Graph Isomorphism Testing Problem"). First, check if the two graphs have the same sets of eigenvalues. If they don't, then they are not isomorphic. If they do, and the eigenvalues have multiplicity one, then draw the pictures above. If the pictures are the same, up to horizontal or vertical flips, and no vertex is mapped to the same location as another, then by lining up the pictures we can recover the permutation.

As some vertices can map to the same location, this heuristic doesn't always work. We will learn about it to the extent to which it does. In particular, we will see in Chapter 39 that if every eigenvalue of two graphs G and H have multiplicity 1, then we can efficiently test whether or not they are isomorphic.

These algorithms have been extended to handle graph in which the multiplicity of every eigenvalue is bounded by a constant [BGM82]. But, there are graphs in which every non-trivial eigenvalue has large multiplicity. We will learn how to construct and analyze these, as they constitute fundamental examples and counter-examples to many natural conjectures. For example, here are the eigenvalues of a Latin Square Graph on 25 vertices. These are a type of Strongly Regular Graph.

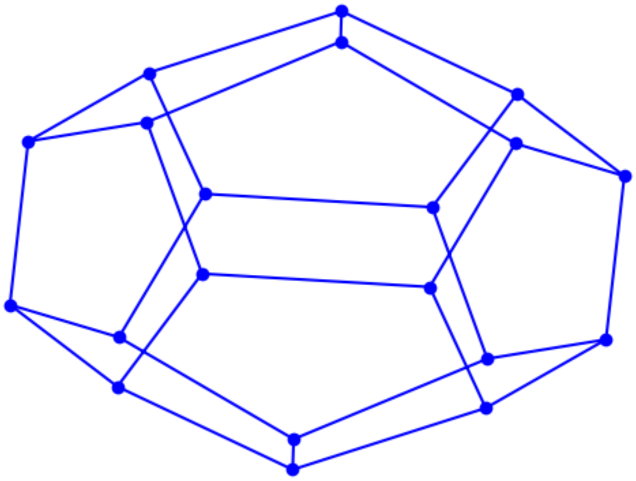
```
M = latin_square_graph(5);
println(eigvals(Matrix(lap(M))))
```

```
[0.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0, 10.0,
15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0, 15.0]
```

All Latin Square Graphs of the same size have the same eigenvalues, whether or not they are isomorphic. We will learn some surprisingly fast (but still not polynomial time) algorithms for checking whether or not Strongly Regular Graphs are isomorphic.

1.5.3 Platonic Solids

Of course, somme graphs are not meant to be drawn in 3 dimensions. For example let's try this with the dodecahedron.

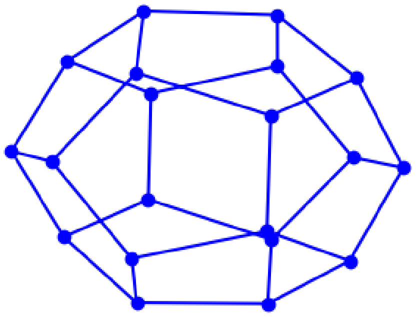


```
M = read_graph("dodec.txt")
spectral_drawing(M)
```

You will notice that this looks like what you would get if you squashed the dodecahedron down to the plane. The reason is that we really shouldn't be drawing this picture in two dimensions: the smallest non-zero eigenvalue of the Laplacian has multiplicity three.

```
E = eigen(Matrix(lap(M)))
println(E.values)
```

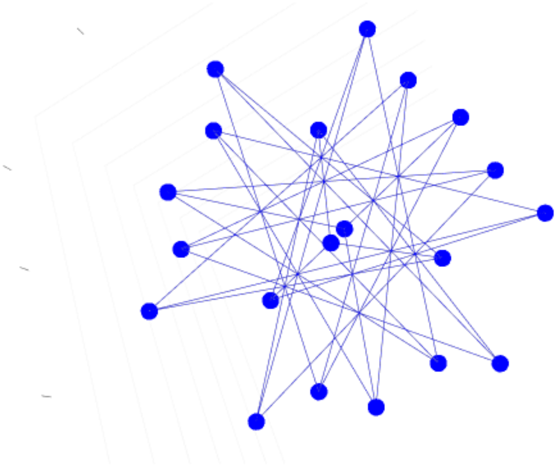
So, we can't reasonably choose just two eigenvectors. We should be choosing three that span the eigenspace. If we do, we would get the canonical representation of the dodecahedron in three dimensions.



```
x = E.vectors[:,2]
y = E.vectors[:,3]
z = E.vectors[:,4]
plot_graph(M, x, y, z; setaxis=false)
```

As you would guess, this happens for all Platonic solids. In fact, if you properly re-weight the edges, it happens for every graph that is the one-skeleton of a convex polytope [Lov01]. Let me state that more concretely. Given a convex polytope in $\mathbb{R}^d$, we can treat its 1-skeleton as a graph on its vertices. There is always a way of assigning positive weights to edges so that the second-smallest Laplacian eigenvalue has multiplicity d, and so that the corresponding eigenspace is spanned by the coordinate vectors of the vertices of the polytope.

We finish this section by contemplating an image of the high-frequency eigenvectors of the dodecahedron. This code plots them in three dimensions, although we can only print them in two. Observe that vertices are approximately opposite their neighbors.



```
x = E.vectors[:,20]
y = E.vectors[:,19]
z = E.vectors[:,18]
plot_graph(M, x, y, z; setaxis=false);
```

1.5.4 The Fiedler Value

The second-smallest eigenvalue of the Laplacian matrix of a graph is zero if and only if the graph is disconnected. If G is disconnected, then we can partition it into two graphs G_1 and G_2 with no edges between them, and then write

$$\boldsymbol{L}_G = \begin{pmatrix} \boldsymbol{L}_{G_1} & 0 \\ 0 & \boldsymbol{L}_{G_2} \end{pmatrix}.$$

As the eigenvalues of $\boldsymbol{L}_G$ are the union, with multiplicity, of the eigenvalues of $\boldsymbol{L}_{G_1}$ and $\boldsymbol{L}_{G_2}$ we see that $\boldsymbol{L}_G$ inherits a zero eigenvalue from each. Conversely, if G is connected then we can show that the only vectors $\boldsymbol{x}$ for which $\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x} = 0$ are the constant vectors. If $\boldsymbol{x}$ is not constant and G is connected then there must be an edge (a, b) for which $\boldsymbol{x}(a) \neq \boldsymbol{x}(b)$. And, this edge will contribute a positive term to the sum (1.1).

Fiedler suggested that we make this qualitative observation quantitative and think of λ_2 as a measure of how well connected the graph is. For this reason, he called it the "Algebraic Connectivity" of a graph, and we call it the "Fiedler value".

Fiedler [Fie73] that the further λ_2 is from 0, the better connected the graph is. In Chapter 21 we will prove ultimate extension of this result: Cheeger's inequality.

In short, we say that a graph is poorly connected if one can cut off many vertices by removing only a few edges. We measure how poorly connected it is by the ratio of these quantities (almost). Cheeger's inequality gives a tight connection between this ratio and λ_2. If λ_2 is small, then for some t, the set of vertices

$$S_i \overset{\text{def}}{=} \{i : \boldsymbol{\psi}_2(i) < t\}$$

may be removed by cutting much less than $|S_i|$ edges. This spectral graph partitioning heuristic has proved very successful in practice.

In general, it will be interesting to turn qualitative statements like this into quantitative ones. For example, the smallest eigenvalue of the diffusion matrix is zero if and only if the graph is bipartite. One can relate the magnitude of this eigenvalue to how far a graph is from being bipartite [Tre09].

1.5.5 Bounding Eigenvalues

We will often be interested in the magnitudes of certain eigenvalues. For this reason, we will learn multiple techniques for proving bounds on eigenvalues. The most prominent of these will be proofs by test vectors and proofs by comparison with simpler graphs.

1.5.6 Planar Graphs

We will prove that graphs that can be drawn nicely must have small Fiedler value, and we will prove very tight results for planar graphs.

We will also see how to use the graph Laplacian to draw planar graphs: Tutte [Tut63] that if one reasonably fixes the locations of the vertices on a face of a planar graph and then lets the others settle into the positions obtained by treating the edges as springs, then one obtains a planar drawing of the graph!

1.5.7 Random Walks on Graphs

Spectral graph theory is one of the main tools we use for analyzing random walks on graphs. We will devote a few chapters to this theory, connect it to Cheeger's inequality, and use tools developed to study random walks to derive a fascinating proof of Cheeger's inequality.

1.5.8 Expanders

We will be particularly interested in graphs that are very well connected. These are called *expanders*. Roughly speaking, expanders are sparse graphs (say a number of edges linear in the number of vertices), in which λ_2 is bounded away from zero by a constant. They are among the most important examples of graphs, and play a prominent role in Theoretical Computer Science.

Expander graphs have numerous applications. We will see how to use random walks on expander graphs to construct pseudo-random generators *about which one can actually prove something.* We will also use them to construct good error-correcting codes.

Error-correcting codes and expander graphs are both fundamental objects of study in the field of Extremal Combinatorics and are extremely useful. We will also use error-correcting codes to construct crude expander graphs. In Chapter 30 we will see a simple construction of good expanders. The best expanders are the Ramanujan graphs. These were first constructed by

Margulis [Mar88] Lubotzky, Phillips and Sarnak [LPS88]. In Chapters **??** and **??** we will prove that there exist infinite families of bipartite Ramanujan graphs.

1.5.9 Approximations of Graphs

We will ask what it means for one graph to approximate another. Given graphs G and H, we will measure how well G approximates H by the closeness of their Laplacian quadratic forms. We will see that expanders are precisely the sparse graphs that provide good approximations of the complete graph, and we will use this perspective for most of our analysis of expanders. We will show that every graph can be well-approximated by a sparse graph through a process called *sparsification*.

1.5.10 Solving equations in and computing eigenvalues of Laplacians

We will also ask how well a graph can be approximated by a tree, and see that low-stretch spanning-trees provide good approximations under this measure.

Our motivation for this material is the need to design fast algorithms for solving systems of linear equations in Laplacian matrices and for computing their eigenvectors. This first problem arises in numerous contexts, including the solution of elliptic PDEs by the finite element method, the solution of network flow problems by interior point algorithms, and in classification problems in Machine Learning.

In fact, our definition of graph approximation is designed to suit the needs of the Preconditioned Conjugate Gradient algorithm.

1.6 Exercises

The following exercises are intended to help you get back in practice at doing linear algebra. You should solve all of them.

1. Orthogonal eigenvectors. Let $\boldsymbol{M}$ be a symmetric matrix, and let $\boldsymbol{\psi}$ and $\boldsymbol{\phi}$ be vectors so that

$$\boldsymbol{M}\boldsymbol{\psi} = \mu\boldsymbol{\psi} \quad \text{and} \quad \boldsymbol{M}\boldsymbol{\phi} = \nu\boldsymbol{\phi}.$$

Prove that if $\mu \neq \nu$ then $\boldsymbol{\psi}$ must be orthogonal to $\boldsymbol{\phi}$. Note that your proof should exploit the symmetry of $\boldsymbol{M}$, as this statement is false otherwise.

2. Invariance under permutations.

Let $\boldsymbol{\Pi}$ be a permutation matrix. That is, there is a permutation $\pi : V \to V$ so that

$$\boldsymbol{\Pi}(u,v) = \begin{cases} 1 & \text{if } u = \pi(v), \text{ and} \\ 0 & \text{otherwise.} \end{cases}$$

Prove that if

$$\boldsymbol{M}\boldsymbol{\psi} = \lambda\boldsymbol{\psi},$$

then

$$\left(\boldsymbol{\Pi}\boldsymbol{M}\boldsymbol{\Pi}^T\right)(\boldsymbol{\Pi}\boldsymbol{\psi}) = \lambda(\boldsymbol{\Pi}\boldsymbol{\psi}).$$

That is, permuting the coordinates of the matrix merely permutes the coordinates of the eigenvectors, and does not change the eigenvalues.

3. Invariance under rotations.

Let $\boldsymbol{Q}$ be an orthogonal matrix. That is, a matrix such that $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}$. Prove that if

$$\boldsymbol{M}\boldsymbol{\psi} = \lambda\boldsymbol{\psi},$$

then

$$\left(\boldsymbol{Q}\boldsymbol{M}\boldsymbol{Q}^T\right)(\boldsymbol{Q}\boldsymbol{\psi}) = \lambda(\boldsymbol{Q}\boldsymbol{\psi}).$$

4. Similar Matrices.

A matrix $\boldsymbol{M}$ is similar to a matrix $\boldsymbol{B}$ if there is a non-singular matrix $\boldsymbol{X}$ such that $\boldsymbol{X}^{-1}\boldsymbol{M}\boldsymbol{X} = \boldsymbol{B}$. Prove that similar matrices have the same eigenvalues.

5. Spectral decomposition.

Let $\boldsymbol{M}$ be a symmetric matrix with eigenvalues $\lambda_1, \dots, \lambda_n$ and let $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$ be a corresponding set of orthonormal column eigenvectors. Let $\boldsymbol{\Psi}$ be the orthogonal matrix whose ith column is $\boldsymbol{\psi}_i$. Prove that

$$\boldsymbol{\Psi}^T\boldsymbol{M}\boldsymbol{\Psi} = \boldsymbol{\Lambda},$$

where $\boldsymbol{\Lambda}$ is the diagonal matrix with $\lambda_1, \dots, \lambda_n$ on its diagonal. Conclude that

$$\boldsymbol{M} = \boldsymbol{\Psi}\boldsymbol{\Lambda}\boldsymbol{\Psi}^T = \sum_{i \in V} \lambda_i \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T.$$

6. Traces.

Recall that the trace of a matrix $\boldsymbol{A}$, written $\mathrm{Tr}\left(\boldsymbol{A}\right)$, is the sum of the diagonal entries of $\boldsymbol{A}$. Prove that for two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$,

$$\mathrm{Tr}\left(\boldsymbol{A}\boldsymbol{B}\right) = \mathrm{Tr}\left(\boldsymbol{B}\boldsymbol{A}\right).$$

Note that the matrices **do not** need to be square for this to be true. They can be rectangular matrices of dimensions $n \times m$ and $m \times n$.

Use this fact and the previous exercise to prove that

$$\mathrm{Tr}\left(\boldsymbol{A}\right) = \sum_{i=1}^{n} \lambda_i,$$

where $\lambda_1, \dots, \lambda_n$ are the eigenvalues of $\boldsymbol{A}$. You are probably familiar with this fact about the trace, or it may have been the definition you were given. This is why I want you to remember how to prove it.

7. The Characteristic Polynomial

Let $\boldsymbol{M}$ be a symmetric matrix. Recall that the eigenvalues of $\boldsymbol{M}$ are the roots of the characteristic polynomial of $\boldsymbol{M}$:

$$p(x) \overset{\text{def}}{=} \det(x\boldsymbol{I} - \boldsymbol{M}) = \prod_{i=1}^{n}(x - \mu_i).$$

Write

$$p(x) = \sum_{k=0}^{n} x^{n-k} c_k (-1)^k.$$

Prove that

$$c_k = \sum_{S \subseteq [n], |S| = k} \det(\boldsymbol{M}(S, S)).$$

Here, we write $[n]$ to denote the set $\{1, \ldots, n\}$, and $\boldsymbol{M}(S, S)$ to denote the submatrix of $\boldsymbol{M}$ with rows and columns indexed by S.

8. Reversing products.

Let $\boldsymbol{M}$ be a d-by-n matrix. Prove that the multiset of nonzero eigenvalues of $\boldsymbol{M}\boldsymbol{M}^T$ is the same as the multiset of nonzero eigenvalues of $\boldsymbol{M}^T\boldsymbol{M}$.

Chapter 2

Eigenvalues and Optimization: The Courant-Fischer Theorem

One of the reasons that the eigenvalues of matrices have meaning is that they arise as the solution to natural optimization problems. The formal statement of this is given by the Courant-Fischer Theorem. We begin by using the Spectral Theorem to prove the Courant-Fischer Theorem. We then prove the Spectral Theorem in a form that is almost identical to Courant-Fischer.

The Rayleigh quotient of a vector $\boldsymbol{x}$ with respect to a matrix $\boldsymbol{M}$ is defined to be

$$\frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}. \tag{2.1}$$

The Rayleigh quotient of an eigenvector is its eigenvalue: if $\boldsymbol{M}\boldsymbol{\psi} = \mu\boldsymbol{\psi}$, then

$$\frac{\boldsymbol{\psi}^T \boldsymbol{M} \boldsymbol{\psi}}{\boldsymbol{\psi}^T \boldsymbol{\psi}} = \frac{\boldsymbol{\psi}^T \mu \boldsymbol{\psi}}{\boldsymbol{\psi}^T \boldsymbol{\psi}} = \mu.$$

The Courant-Fischer Theorem tells us that the vectors $\boldsymbol{x}$ that maximize the Rayleigh quotient are exactly the eigenvectors of the largest eigenvalue of $\boldsymbol{M}$. In fact it supplies a similar characterization of all the eigenvalues of a symmetric matrix.

Theorem 2.0.1 (Courant-Fischer Theorem)**.** *Let $\boldsymbol{M}$ be a symmetric matrix with eigenvalues $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$. Then,*

$$\mu_k = \max_{\substack{S \subseteq \mathbb{R}^n \\ \dim(S)=k}} \min_{\substack{\boldsymbol{x} \in S \\ \boldsymbol{x} \neq \boldsymbol{0}}} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \min_{\substack{T \subseteq \mathbb{R}^n \\ \dim(T)=n-k+1}} \max_{\substack{\boldsymbol{x} \in T \\ \boldsymbol{x} \neq \boldsymbol{0}}} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}},$$

where the maximization and minimization are over subspaces S and T of $\mathbb{R}^n$.

Be warned that we will often neglect to include the condition $\boldsymbol{x} \neq \boldsymbol{0}$, but we always intend it.

2.1 The First Proof

As with many proofs in Spectral Graph Theory, we begin by expanding a vector $\boldsymbol{x}$ in the basis of eigenvectors of $\boldsymbol{M}$. Let's recall how this is done.

Let $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$ be an orthonormal basis of eigenvectors of $\boldsymbol{M}$ corresponding to $\mu_1, \dots, \mu_n$. As these are an orthonormal basis, we may write

$$\boldsymbol{x} = \sum_i c_i \boldsymbol{\psi}_i, \qquad \text{where } c_i = \boldsymbol{\psi}_i^T \boldsymbol{x}.$$

There are many ways to verify this. We let $\boldsymbol{\Psi}$ be the matrix whose columns are $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$, and recall that the matrix $\boldsymbol{\Psi}$ is said to be *orthogonal* if its columns are orthonormal vectors. Also recall that the orthogonal matrices are exactly those matrices $\boldsymbol{\Psi}$ for which $\boldsymbol{\Psi}\boldsymbol{\Psi}^T = \boldsymbol{I}$, and that this implies that $\boldsymbol{\Psi}^T\boldsymbol{\Psi} = \boldsymbol{I}$. We now verify that

$$\sum_i c_i \boldsymbol{\psi}_i = \sum_i \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \boldsymbol{x} = \left(\sum_i \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \right) \boldsymbol{x} = \left(\boldsymbol{\Psi}\boldsymbol{\Psi}^T \right) \boldsymbol{x} = \boldsymbol{I}\boldsymbol{x} = \boldsymbol{x}.$$

When confused by orthonormal bases, just pretend that they are the basis of elementary unit vectors. For example, you know that

$$\boldsymbol{x} = \sum_i \boldsymbol{x}(i) \boldsymbol{\delta}_i, \qquad \text{and that } \boldsymbol{x}(i) = \boldsymbol{\delta}_i^T \boldsymbol{x}.$$

The first step in the proof is to express the Laplacian quadratic form of $\boldsymbol{x}$ in terms of the expansion of $\boldsymbol{x}$ in the eigenbasis.

Lemma 2.1.1. *Let $\boldsymbol{M}$ be a symmetric matrix with eigenvalues $\mu_1, \dots, \mu_n$ and a corresponding orthonormal basis of eigenvectors $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$. Let $\boldsymbol{x}$ be a vector whose expansion in the eigenbasis is*

$$\boldsymbol{x} = \sum_{i=1}^n c_i \boldsymbol{\psi}_i.$$

Then,

$$\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} = \sum_{i=1}^n c_i^2 \mu_i.$$

Proof. Compute:

$$\begin{aligned}
\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} &= \left(\sum_i c_i \boldsymbol{\psi}_i\right)^T \boldsymbol{M} \left(\sum_j c_j \boldsymbol{\psi}_j\right) \\
&= \left(\sum_i c_i \boldsymbol{\psi}_i\right)^T \left(\sum_j c_j \mu_j \boldsymbol{\psi}_j\right) \\
&= \sum_{i,j} c_i c_j \mu_j \boldsymbol{\psi}_i^T \boldsymbol{\psi}_j \\
&= \sum_i c_i^2 \mu_i,
\end{aligned}$$

as

$$\boldsymbol{\psi}_i^T \boldsymbol{\psi}_j = \begin{cases} 0 & \text{for } i \neq j \\ 1 & \text{for } i = j. \end{cases}$$

□

Proof of 2.0.1. Let $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$ be an orthonormal set of eigenvectors of $\boldsymbol{M}$ corresponding to $\mu_1, \dots, \mu_n$. We will just verify the first characterization of μ_k. The other is similar.

First, let's verify that μ_k is achievable. Let S be the span of $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_k$. We can expand every $\boldsymbol{x} \in S$ as

$$\boldsymbol{x} = \sum_{i=1}^{k} c_i \boldsymbol{\psi}_i.$$

Applying Lemma 2.1.1 we obtain

$$\frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \frac{\sum_{i=1}^k \mu_i c_i^2}{\sum_{i=1}^k c_i^2} \geq \frac{\sum_{i=1}^k \mu_k c_i^2}{\sum_{i=1}^k c_i^2} = \mu_k.$$

So,

$$\min_{\boldsymbol{x} \in S} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \geq \mu_k.$$

To show that this is in fact the maximum, we will prove that for all subspaces S of dimension k,

$$\min_{\boldsymbol{x} \in S} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \leq \mu_k.$$

Let T be the span of $\boldsymbol{\psi}_k, \dots, \boldsymbol{\psi}_n$. As T has dimension $n - k + 1$, every S of dimension k has an intersection with T of dimension at least 1. So,

$$\min_{\boldsymbol{x} \in S} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \leq \min_{\boldsymbol{x} \in S \cap T} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \leq \max_{\boldsymbol{x} \in T} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

Any $\boldsymbol{x}$ in T may be expressed as

$$\boldsymbol{x} = \sum_{i=k}^{n} c_i \boldsymbol{\psi}_i,$$

and so for $\boldsymbol{x}$ in T

$$\frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \frac{\sum_{i=k}^n \mu_i c_i^2}{\sum_{i=k}^n c_i^2} \leq \frac{\sum_{i=k}^n \mu_k c_i^2}{\sum_{i=k}^n c_i^2} = \mu_k.$$

□

2.2 Proof of the Spectral Theorem

We begin the second proof by showing that the Rayleigh quotient is maximized at an eigenvector of μ_1.

Theorem 2.2.1. *Let $\boldsymbol{M}$ be a symmetric matrix and let $\boldsymbol{x}$ be a non-zero vector that maximizes the Rayleigh quotient with respect to $\boldsymbol{M}$:*

$$\frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

Then, $\boldsymbol{M}\boldsymbol{x} = \mu_1 \boldsymbol{x}$, where μ_1 is the largest eigenvalue of $\boldsymbol{M}$. Conversely, the minimum is achieved by eigenvectors of the smallest eigenvalue of $\boldsymbol{M}$.

Proof. We first observe that the maximum is achieved: as the Rayleigh quotient is homogeneous, it suffices to consider unit vectors $\boldsymbol{x}$. As the set of unit vectors is a closed and compact set, the maximum is achieved on this set.

Now, let $\boldsymbol{x}$ be a non-zero vector that maximizes the Rayleigh quotient. We recall that the gradient of a function at its maximum must be the zero vector. Let's compute that gradient.

We have[1]

$$\nabla \boldsymbol{x}^T \boldsymbol{x} = 2\boldsymbol{x},$$

and

$$\nabla \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} = 2\boldsymbol{M}\boldsymbol{x}.$$

So,

$$\nabla \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \frac{(\boldsymbol{x}^T \boldsymbol{x})(2\boldsymbol{M}\boldsymbol{x}) - (\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x})(2\boldsymbol{x})}{(\boldsymbol{x}^T \boldsymbol{x})^2}.$$

In order for this to be zero, we must have

$$(\boldsymbol{x}^T \boldsymbol{x})\boldsymbol{M}\boldsymbol{x} = (\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x})\boldsymbol{x},$$

which implies

$$\boldsymbol{M}\boldsymbol{x} = \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \boldsymbol{x}.$$

That is, if and only if $\boldsymbol{x}$ is an eigenvector of $\boldsymbol{M}$ with eigenvalue equal to its Rayleigh quotient. As $\boldsymbol{x}$ maximizes the Rayleigh quotient, this eigenvalue must be μ_1. □

[1] In case you are not used to computing gradients of functions of vectors, you can derive these directly by reasoning like

$$\frac{\partial}{\partial \boldsymbol{x}(a)} \boldsymbol{x}^T \boldsymbol{x} = \frac{\partial}{\partial \boldsymbol{x}(a)} \sum_b \boldsymbol{x}(b)^2 = 2\boldsymbol{x}(a).$$

We now prove the Spectral Theorem by generalizing this characterization to all of the eigenvalues of $\boldsymbol{M}$. The idea is to always use Theorem 2.2.1 to show that a vector is an eigenvector. To do this, we must modify the matrix for each vector.

Theorem 2.2.2. *Let $\boldsymbol{M}$ be an n-dimensional real symmetric matrix. There exist numbers $\mu_1, \dots, \mu_n$ and orthonormal vectors $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$ such that $\boldsymbol{M}\boldsymbol{\psi}_i = \mu_i \boldsymbol{\psi}_i$. Moreover,*

$$\boldsymbol{\psi}_1 \in \arg \max_{\|\boldsymbol{x}\|=1} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x},$$

and for $2 \leq i \leq n$

$$\boldsymbol{\psi}_i \in \arg \max_{\substack{\|\boldsymbol{x}\|=1 \\ \boldsymbol{x}^T \boldsymbol{\psi}_j = 0, \textit{for } j<i}} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}. \tag{2.2}$$

Similarly,

$$\boldsymbol{\psi}_i \in \arg \min_{\substack{\|\boldsymbol{x}\|=1 \\ \boldsymbol{x}^T \boldsymbol{\psi}_j = 0, \textit{for } j>i}} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}.$$

Proof. We use Theorem 2.2.1 to obtain $\boldsymbol{\psi}_1$ and μ_1, and would like to proceed by induction. But first, we reduce to the case of positive definite matrices.

By Theorem 2.2.1, we also know that there is a μ_n such that

$$\mu_n = \min_{\boldsymbol{x}} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}^T}{\boldsymbol{x}^T \boldsymbol{x}}.$$

Now consider the matrix $\widetilde{\boldsymbol{M}} = \boldsymbol{M} + (1 - \mu_n)\boldsymbol{I}$. For all $\boldsymbol{x}$ such that $\|\boldsymbol{x}\| = 1$,

$$\boldsymbol{x}^T \widetilde{\boldsymbol{M}} \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} + 1 - \mu_n \geq 1.$$

So, $\widetilde{\boldsymbol{M}}$ is positive definite. As $\widetilde{\boldsymbol{M}}\boldsymbol{x} = \boldsymbol{M}\boldsymbol{x} + (1 - \mu_n)\boldsymbol{x}$, the eigenvectors of $\widetilde{\boldsymbol{M}}$ and $\boldsymbol{M}$ are the same. Thus it suffices to prove the theorem for positive definite matrices.

We henceforth assume without loss of generality that $\boldsymbol{M}$ is positive definite, and proceed by induction on k. Assuming that we have eigenvectors $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_k$ satisfying (2.2), we construct $\boldsymbol{\psi}_{k+1}$. Define

$$\boldsymbol{M}_k = \boldsymbol{M} - \sum_{i=1}^{k} \mu_i \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T.$$

For $j \leq k$ we have

$$\boldsymbol{M}_k \boldsymbol{\psi}_j = \boldsymbol{M} \boldsymbol{\psi}_j - \sum_{i=1}^{k} \mu_i \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \boldsymbol{\psi}_j = \mu_j \boldsymbol{\psi}_j - \mu_j \boldsymbol{\psi}_j = \boldsymbol{0}.$$

So, for vectors $\boldsymbol{x}$ that are orthogonal to $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_k$,

$$\boldsymbol{M}_k \boldsymbol{x} = \boldsymbol{M} \boldsymbol{x}, \quad \boldsymbol{x}^T \boldsymbol{M}_k \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}, \quad \text{and}$$

$$\arg \max_{\substack{\|\boldsymbol{x}\|=1 \\ \boldsymbol{x}^T \boldsymbol{\psi}_j = 0, \text{for } j \leq k}} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} \leq \arg \max_{\|\boldsymbol{x}\|=1} \boldsymbol{x}^T \boldsymbol{M}_k \boldsymbol{x}. \tag{2.3}$$

Now, let $\boldsymbol{y}$ be a unit vector that maximizes $\boldsymbol{y}^T \boldsymbol{M}_k \boldsymbol{y}$. We know from Theorem 2.2.1 that $\boldsymbol{y}$ is an eigenvector of $\boldsymbol{M}_k$. Call its eigenvalue μ. We now show that $\boldsymbol{y}$ must be orthogonal to each of $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_k$. Let

$$\widetilde{\boldsymbol{y}} = \boldsymbol{y} - \sum_{i=1}^{k} \boldsymbol{\psi}_i (\boldsymbol{\psi}_i^T \boldsymbol{y})$$

be the projection of $\boldsymbol{y}$ orthogonal to $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_k$, and let $\widehat{\boldsymbol{y}} = \widetilde{\boldsymbol{y}} / \left\| \widetilde{\boldsymbol{y}} \right\|$. As $\boldsymbol{M}_k \boldsymbol{\psi}_i = \boldsymbol{0}$ for $i \leq k$, we know that $\widetilde{\boldsymbol{y}}^T \boldsymbol{M}_k \widetilde{\boldsymbol{y}} = \boldsymbol{y}^T \boldsymbol{M} \boldsymbol{y}$. If $\boldsymbol{y}$ is not orthogonal to these vectors, that is if some $\boldsymbol{\psi}_i^T \boldsymbol{x}$ is nonzero, then $\left\| \widetilde{\boldsymbol{y}} \right\| < \left\| \boldsymbol{y} \right\|$. As $\widetilde{\boldsymbol{y}}^T \boldsymbol{M} \widetilde{\boldsymbol{y}} > 0$, this would imply that for the unit vector $\widehat{\boldsymbol{y}}$, $\widehat{\boldsymbol{y}}^T \boldsymbol{M}_k \widehat{\boldsymbol{y}} > \boldsymbol{y}^T \boldsymbol{M}_k \boldsymbol{y}$, a contradiction. As $\boldsymbol{y}$ is orthogonal to $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_k$ and it is an eigenvector of $\boldsymbol{M}_k$, it is also an eigenvector of $\boldsymbol{M}$:

$$\boldsymbol{M} \boldsymbol{y} = \boldsymbol{M}_k \boldsymbol{y} = \mu \boldsymbol{y},$$

and by (2.3)

$$\boldsymbol{y} \in \arg \max_{\substack{\|\boldsymbol{x}\|=1 \\ \boldsymbol{x}^T \boldsymbol{\psi}_j = 0, \text{for } j \leq k}} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}.$$

We now set $\boldsymbol{\psi}_{k+1} = \boldsymbol{y}$ and $\mu_{k+1} = \mu$. □

2.3 Notes

The characterization of eigenvalues by maximizing or minimizing the Rayleigh quotient only works for symmetric matrices. The analogous quantities for non-symmetric matrices $\boldsymbol{A}$ are the *singular vectors* and *singular values* of $\boldsymbol{A}$, which are the eigenvectors of $\boldsymbol{A}\boldsymbol{A}^T$ and $\boldsymbol{A}^T\boldsymbol{A}$, and the square roots of the eigenvalues of those matrices.

2.4 Exercise

1. A tighter characterization.

Tighten Theorem 2.2.2 by proving that for every sequence of vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ such that

$$\boldsymbol{x}_i \in \arg \max_{\substack{\|\boldsymbol{x}\|=1 \\ \boldsymbol{x}^T \boldsymbol{x}_j = 0, \text{for } j < i}} \boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x},$$

each $\boldsymbol{x}_i$ is an eigenvector of $\boldsymbol{M}$.

Chapter 3

The Laplacian and Graph Drawing

3.1 The Laplacian Matrix

We being this section by establishing the equivalence of multiple expressions for the Laplacian.

The Laplacian Matrix of a weighted graph $G = (V, E, w)$, $w : E \to \mathbb{R}^+$, is designed to capture the Laplacian quadratic form:

$$\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x} = \sum_{(a,b) \in E} w_{a,b} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2. \tag{3.1}$$

We will now use this quadratic form to derive the structure of the matrix. To begin, consider a graph with just two vertices and one edge of weight 1. Let's call it $G_{1,2}$. We have

$$\boldsymbol{x}^T \boldsymbol{L}_{G_{1,2}} \boldsymbol{x} = (\boldsymbol{x}(1) - \boldsymbol{x}(2))^2. \tag{3.2}$$

Consider the vector $\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2$, where $\boldsymbol{\delta}_a$ is the elementary unit vector with a 1 in coordinate a. We have

$$\boldsymbol{x}(1) - \boldsymbol{x}(2) = \boldsymbol{\delta}_1^T \boldsymbol{x} - \boldsymbol{\delta}_2^T \boldsymbol{x} = (\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2)^T \boldsymbol{x},$$

so

$$(\boldsymbol{x}(1) - \boldsymbol{x}(2))^2 = \left((\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2)^T \boldsymbol{x}\right)^2 = \boldsymbol{x}^T (\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2)(\boldsymbol{\delta}_1 - \boldsymbol{\delta}_2)^T \boldsymbol{x} = \boldsymbol{x}^T \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \boldsymbol{x}.$$

Thus,

$$\boldsymbol{L}_{G_{1,2}} = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Now, let $G_{a,b}$ be the graph with just one edge between a and b. It can have as many other vertices as you like. The Laplacian of $G_{a,b}$ can be written in the same way:

$$\boldsymbol{L}_{G_{a,b}} = (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T.$$

This is the matrix that is zero except at the intersection of rows and columns indexed by a and b, where it looks looks like

$$\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}.$$

Summing the matrices for every edge, we obtain

$$\boldsymbol{L}_G = \sum_{(a,b)\in E} w_{a,b}(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T = \sum_{(a,b)\in E} w_{a,b}\boldsymbol{L}_{G_{a,b}}.$$

You can check that this agrees with the definition of the Laplacian from Section 1.2.3:

$$\boldsymbol{L}_G = \boldsymbol{D}_G - \boldsymbol{A}_G,$$

where

$$\boldsymbol{D}_G(a,a) = \sum_b w_{a,b}.$$

This formula turns out to be useful when we view the Laplacian as an operator. For every vector $\boldsymbol{x}$ we have

$$(\boldsymbol{L}_G\boldsymbol{x})(a) = d(a)\boldsymbol{x}(a) - \sum_{(a,b)\in E} w_{a,b}\boldsymbol{x}(b) = \sum_{(a,b)\in E} w_{a,b}(\boldsymbol{x}(a) - \boldsymbol{x}(b)). \tag{3.3}$$

From (3.1), we see that if all entries of $\boldsymbol{x}$ are the same, then $\boldsymbol{x}^T\boldsymbol{L}\boldsymbol{x}$ equals zero. From (3.3), we can immediately see that $\boldsymbol{L}\boldsymbol{1} = \boldsymbol{0}$, so the constant vectors are eigenvectors of eigenvalue zero. If the graph is connected, these are the only eigenvectors of eigenvalue zero.

Lemma 3.1.1. *Let $G = (V,E)$ be a graph, and let $0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of its Laplacian matrix, $\boldsymbol{L}$. Then, $\lambda_2 > 0$ if and only if G is connected.*

Proof. We first show that $\lambda_2 = 0$ if G is disconnected. If G is disconnected, then it can be described as the union of two graphs, G_1 and G_2. After suitably reordering the vertices, we can write

$$\boldsymbol{L} = \begin{bmatrix} L_{G_1} & 0 \\ 0 & L_{G_2} \end{bmatrix}.$$

So, $\boldsymbol{L}$ has at least two orthogonal eigenvectors of eigenvalue zero:

$$\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{1} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \boldsymbol{1} \\ \boldsymbol{0} \end{bmatrix}.$$

where we have partitioned the vectors as we did the matrix $\boldsymbol{L}$.

On the other hand, assume that G is connected and that $\boldsymbol{\psi}$ is an eigenvector of $\boldsymbol{L}$ of eigenvalue 0. As

$$\boldsymbol{L}\boldsymbol{\psi} = \boldsymbol{0},$$

we have

$$x^T\boldsymbol{L}x = \sum_{(a,b)\in E} (\boldsymbol{\psi}(a) - \boldsymbol{\psi}(b))^2 = 0.$$

Thus, for every pair of vertices (a,b) connected by an edge, we have $\boldsymbol{\psi}(a) = \boldsymbol{\psi}(b)$. As every pair of vertices a and b are connected by a path, we may inductively apply this fact to show that $\boldsymbol{\psi}(a) = \boldsymbol{\psi}(b)$ for all vertices a and b. Thus, $\boldsymbol{\psi}$ must be a constant vector. We conclude that the eigenspace of eigenvalue 0 has dimension 1. □

Of course, the same holds for weighted graphs.

3.2 Drawing with Laplacian Eigenvalues

The idea of drawing graphs using eigenvectors demonstrated in Section 1.5.1 was suggested by Hall [Hal70] in 1970.

To explain Hall's approach, we first consider the problem of drawing a graph on a line. That is, mapping each vertex to a real number. It isn't easy to see what a graph looks like when you do this, as all of the edges sit on top of one another. One can fix this either by drawing the edges of the graph as curves, or by wrapping the line around a circle.

Let $\boldsymbol{x} \in \mathbb{R}^V$ be the vector that describes the assignment of a real number to each vertex. We would like vertices that are neighbors to be close to one another. So, Hall suggested that we choose an $\boldsymbol{x}$ minimizing

$$\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} = \sum_{(a,b)\in E} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2. \tag{3.4}$$

Unless we place restrictions on $\boldsymbol{x}$, the solution will be degenerate. For example, all of the vertices could map to 0. To avoid this, and to fix the scale of the embedding overall, we require

$$\sum_{a\in V} \boldsymbol{x}(a)^2 = \|\boldsymbol{x}\|^2 = 1. \tag{3.5}$$

Even with this restriction, another degenerate solution is possible: it could be that every vertex maps to $1/\sqrt{n}$. To prevent this from happening, we impose the additional restriction that

$$\sum_{a} \boldsymbol{x}(a) = \boldsymbol{1}^T \boldsymbol{x} = 0. \tag{3.6}$$

On its own, this restriction fixes the shift of the embedding along the line. When combined with (3.5), it guarantees that we get something interesting.

As $\boldsymbol{1}$ is the eigenvector of smallest eigenvalue of the Laplacian, Theorem 2.2.2 implies that a unit eigenvector of λ_2 minimizes $\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x}$ subject to (3.5) and (3.6).

Of course, we really want to draw a graph in two dimensions. So, we will assign two coordinates to each vertex given by $\boldsymbol{x}$ and $\boldsymbol{y}$. As opposed to minimizing (3.4), we will minimize the sum of the squares of the lengths of the edges in the embedding:

$$\sum_{(a,b)\in E} \left\| \begin{pmatrix} \boldsymbol{x}(a) \\ \boldsymbol{y}(a) \end{pmatrix} - \begin{pmatrix} \boldsymbol{x}(b) \\ \boldsymbol{y}(b) \end{pmatrix} \right\|^2 .$$

This turns out not to be so different from minimizing (3.4), as it equals

$$\sum_{(a,b)\in E} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2 + (\boldsymbol{y}(a) - \boldsymbol{y}(b))^2 = \boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} + \boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}.$$

As before, we impose the scale conditions

$$\|\boldsymbol{x}\|^2 = 1 \quad \text{and} \quad \|\boldsymbol{y}\|^2 = 1,$$

and the centering constraints

$$\mathbf{1}^T \boldsymbol{x} = 0 \quad \text{and} \quad \mathbf{1}^T \boldsymbol{y} = 0.$$

However, this still leaves us with the degenerate solution $\boldsymbol{x} = \boldsymbol{y} = \boldsymbol{\psi}_2$. To ensure that the two coordinates are different, Hall introduced the restriction that $\boldsymbol{x}$ be orthogonal to $\boldsymbol{y}$. To embed a graph in k dimensions, we find k orthonormal vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ that are orthogonal to $\mathbf{1}$ and minimize $\sum_i \boldsymbol{x}_i^T \boldsymbol{L} \boldsymbol{x}_i$. A natural choice for these is $\boldsymbol{\psi}_2$ through $\boldsymbol{\psi}_{k+1}$, and this choice achieves objective function value $\sum_{i=2}^{k+1} \lambda_i$.

The following theorem says that this choice is optimal. It is a variant of [Fan49, Theorem 1].

Theorem 3.2.1. *Let $\boldsymbol{L}$ be a Laplacian matrix and let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$ be orthonormal vectors that are all orthogonal to $\mathbf{1}$. Then*

$$\sum_{i=1}^{k} \boldsymbol{x}_i^T \boldsymbol{L} \boldsymbol{x}_i \geq \sum_{i=2}^{k+1} \lambda_i,$$

and this inequality is tight only when $\boldsymbol{x}_i^T \boldsymbol{\psi}_j = 0$ for all j such that $\lambda_j > \lambda_{k+1}$.

Proof. Without loss of generality, let $\boldsymbol{\psi}_1$ be a constant vector.

Let $\boldsymbol{x}_{k+1}, \dots, \boldsymbol{x}_n$ be vectors such that $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ is an orthonormal basis. We can find these by choosing $\boldsymbol{x}_{k+1}, \dots, \boldsymbol{x}_n$ to be an orthonormal basis of the space orthogonal to $\boldsymbol{x}_1, \dots, \boldsymbol{x}_k$. We now know that for all $1 \leq i \leq n$

$$\sum_{j=1}^{n} (\boldsymbol{\psi}_j^T \boldsymbol{x}_i)^2 = 1 \qquad \text{and} \qquad \sum_{j=1}^{n} (\boldsymbol{x}_j^T \boldsymbol{\psi}_i)^2 = 1.$$

That is, the matrix with i, j entry $(\boldsymbol{\psi}_j^T \boldsymbol{x}_i)^2$ is *doubly-stochastic*[1].

Since $\boldsymbol{\psi}_1^T \boldsymbol{x}_i = 0$, Lemma 2.1.1 implies

$$\begin{aligned}
\boldsymbol{x}_i^T \boldsymbol{L} \boldsymbol{x}_i &= \sum_{j=2}^{n} \lambda_j (\boldsymbol{\psi}_j^T \boldsymbol{x}_i)^2 \\
&= \lambda_{k+1} + \sum_{j=2}^{n} (\lambda_j - \lambda_{k+1}) (\boldsymbol{\psi}_j^T \boldsymbol{x}_i)^2, && \text{by } \sum_{j=2}^{n} (\boldsymbol{\psi}_j^T \boldsymbol{x}_i)^2 = 1 \\
&\geq \lambda_{k+1} + \sum_{j=2}^{k+1} (\lambda_j - \lambda_{k+1}) (\boldsymbol{\psi}_j^T \boldsymbol{x}_i)^2,
\end{aligned}$$

as $\lambda_j \geq \lambda_{k+1}$ for $j > k+1$. This inequality is only tight when $(\boldsymbol{\psi}_j^T \boldsymbol{x}_i)^2 = 0$ for j such that $\lambda_j > \lambda_{k+1}$.

[1] This theorem is really about *majorization*, which is easily established through multiplication by a doubly-stochastic matrix.

Summing over i we obtain

$$\begin{aligned}\sum_{i=1}^{k} \boldsymbol{x}_i^T \boldsymbol{L} \boldsymbol{x}_i &\geq k\lambda_{k+1} + \sum_{j=2}^{k+1} (\lambda_j - \lambda_{k+1}) \sum_{i=1}^{k} (\boldsymbol{\psi}_j^T \boldsymbol{x}_i)^2 \\ &\geq k\lambda_{k+1} + \sum_{j=2}^{k+1} (\lambda_j - \lambda_{k+1}) \\ &= \sum_{j=2}^{k+1} \lambda_j,\end{aligned}$$

where the inequality follows from the facts that $\lambda_j - \lambda_{k+1} \leq 0$ and $\sum_{i=1}^{k} (\boldsymbol{\psi}_j^T \boldsymbol{x}_i)^2 \leq 1$. This inequality is tight under the same conditions as the previous one.

□

The beautiful pictures that we sometimes obtain from Hall's graph drawing should convince you that eigenvectors of the Laplacian should reveal a lot about the structure of graphs. But, it is worth pointing out that there are many graphs for which this approach does not produce nice images, and there are in fact graphs that can not be nicely drawn. Expander graphs are good examples of these.

Many other approaches to graph drawing borrow ideas from Hall's work: they try to minimize some function of the distances of the edges subject to some constraints that keep the vertices well separated. However, very few of these have compactly describable solutions, or even solutions that can provably be computed in polynomial time. The algorithms that implement them typically use a gradient based method to attempt to minimize the function of the distances subject to constraints. This means that relabeling the vertices could produce very different drawings! Thus, one must be careful before using these images to infer some truth about a graph.

Chapter 4

Adjacency matrices, Eigenvalue Interlacing, and the Perron-Frobenius Theorem

In this chapter, we examine the meaning of the smallest and largest eigenvalues of the adjacency matrix of a graph. Note that the largest eigenvalue of the adjacency matrix corresponds to the smallest eigenvalue of the Laplacian. Our focus in this chapter will be on the features that adjacency matrices possess but which Laplacians do not. Where the smallest eigenvector of the Laplacian is a constant vector, the largest eigenvector of an adjacency matrix, called the *Perron vector*, need not be. The Perron-Frobenius theory tells us that the largest eigenvector of an adjacency matrix is non-negative, and that its value is an upper bound on the absolute value of the smallest eigenvalue. These are equal precisely when the graph is bipartite.

We will examine the relation between the largest adjacency eigenvalue and the degrees of vertices in the graph. This is made more meaningful by the fact that we can apply Cauchy's Interlacing Theorem to adjacency matrices. We will use it to prove a theorem of Wilf [Wil67] which says that a graph can be colored using at most $1 + \lfloor \mu_1 \rfloor$ colors. We will learn more about eigenvalues and graph coloring in Chapter 19.

4.1 The Adjacency Matrix

Let $\boldsymbol{M}$ be the adjacency matrix of a (possibly weighted) graph G. As an operator, $\boldsymbol{M}$ acts on a vector $\boldsymbol{x} \in \mathbb{R}^V$ by

$$(\boldsymbol{M}\boldsymbol{x})(a) = \sum_{(a,b)\in E} w_{a,b}\boldsymbol{x}(b). \tag{4.1}$$

We will denote the eigenvalues of $\boldsymbol{M}$ by $\mu_1, \ldots, \mu_n$. But, we order them in the opposite direction than we did for the Laplacian: we assume

$$\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n.$$

The reason for this convention is so that μ_i corresponds to the ith Laplacian eigenvalue, λ_i. If G is a d-regular graph, then $\boldsymbol{D} = \boldsymbol{I}d$,

$$\boldsymbol{L} = \boldsymbol{I}d - \boldsymbol{M},$$

and so

$$\lambda_i = d - \mu_i.$$

Thus the largest adjacency eigenvalue of a d-regular graph is d, and its corresponding eigenvector is the constant vector. We could also prove that the constant vector is an eigenvector of eigenvalue d by considering the action of $\boldsymbol{M}$ as an operator (4.1): if $\boldsymbol{x}(a) = 1$ for all a, then $(\boldsymbol{M}\boldsymbol{x})(b) = d$ for all b.

4.2 The Largest Eigenvalue, μ_1

We now examine μ_1 for graphs which are not necessarily regular. Let G be a graph, let d_{max} be the maximum degree of a vertex in G, and let d_{ave} be the average degree of a vertex in G.

Lemma 4.2.1.

$$d_{ave} \leq \mu_1 \leq d_{max}.$$

Proof. The lower bound follows by considering the Rayleigh quotient with the all-1s vector:

$$\mu_1 = \max_{\boldsymbol{x}} \frac{\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \geq \frac{\boldsymbol{1}^T \boldsymbol{M} \boldsymbol{1}}{\boldsymbol{1}^T \boldsymbol{1}} = \frac{\sum_{a,b} \boldsymbol{M}(a,b)}{n} = \frac{\sum_a \boldsymbol{d}(a)}{n} = d_{ave}.$$

To prove the upper bound, Let $\boldsymbol{\phi}_1$ be an eigenvector of eigenvalue μ_1. Let a be the vertex on which $\boldsymbol{\phi}_1$ takes its maximum value, so $\boldsymbol{\phi}_1(a) \geq \boldsymbol{\phi}_1(b)$ for all b, and we may assume without loss of generality that $\boldsymbol{\phi}_1(a) > 0$ (use $-\boldsymbol{\phi}_1$ if $\boldsymbol{\phi}_1$ is strictly negative). We have

$$\mu_1 = \frac{(\boldsymbol{M}\boldsymbol{\phi}_1)(a)}{\boldsymbol{\phi}_1(a)} = \frac{\sum_{b:b\sim a} \boldsymbol{\phi}_1(b)}{\boldsymbol{\phi}_1(a)} = \sum_{b:b\sim a} \frac{\boldsymbol{\phi}_1(b)}{\boldsymbol{\phi}_1(a)} \leq \sum_{b:b\sim a} 1 = d(a) \leq d_{max}. \tag{4.2}$$

□

Lemma 4.2.2. *If G is connected and $\mu_1 = d_{max}$, then G is d_{max}-regular.*

Proof. If we have equality in (4.2), then it must be the case that $d(a) = d_{max}$ and $\boldsymbol{\phi}_1(b) = \boldsymbol{\phi}_1(a)$ for all $(a,b) \in E$. Thus, we may apply the same argument to every neighbor of a. As the graph is connected, we may keep applying this argument to neighbors of vertices to which it has already been applied to show that $\boldsymbol{\phi}_1(c) = \boldsymbol{\phi}_1(a)$ and $d(c) = d_{max}$ for all $c \in V$. □

The technique used in these last two proofs will appear many times in this Chapter.

4.3 Eigenvalue Interlacing

We can strengthen the lower bound in Lemma 4.2.1 by proving that μ_1 is at least the average degree of every subgraph of G. We will prove this by applying Cauchy's Interlacing Theorem.

For a graph $G = (V, E)$ and $S \subset V$, we define the subgraph induced by S, written $G(S)$, to be the graph with vertex set S and all edges in E connecting vertices in S:

$$\{(a, b) \in E : a \in S \text{ and } b \in S\}.$$

For a symmetric matrix $\boldsymbol{M}$ whose rows and columns are indexed by a set V, and a $S \subset V$, we write $\boldsymbol{M}(S)$ for the symmetric submatrix with rows and columns in S.

Theorem 4.3.1 (Cauchy's Interlacing Theorem)**.** *Let $\boldsymbol{A}$ be an n-by-n symmetric matrix and let $\boldsymbol{B}$ be a principal submatrix of $\boldsymbol{A}$ of dimension $n-1$ (that is, $\boldsymbol{B}$ is obtained by deleting the same row and column from $\boldsymbol{A}$). Then,*

$$\alpha_1 \geq \beta_1 \geq \alpha_2 \geq \beta_2 \geq \cdots \geq \alpha_{n-1} \geq \beta_{n-1} \geq \alpha_n,$$

where $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$ and $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_{n-1}$ are the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively.

Proof. Without loss of generality we will assume that $\boldsymbol{B}$ is obtained from $\boldsymbol{A}$ by removing its first row and column. We now apply the Courant-Fischer Theorem, which tells us that

$$\alpha_k = \max_{\substack{S \subseteq \mathbb{R}^n \\ \dim(S)=k}} \min_{\boldsymbol{x} \in S} \frac{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

Applying this to $\boldsymbol{B}$ gives

$$\beta_k = \max_{\substack{S \subseteq \mathbb{R}^{n-1} \\ \dim(S)=k}} \min_{\boldsymbol{x} \in S} \frac{\boldsymbol{x}^T \boldsymbol{B} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \max_{\substack{S \subseteq \mathbb{R}^{n-1} \\ \dim(S)=k}} \min_{\boldsymbol{x} \in S} \frac{\begin{pmatrix} 0 \\ \boldsymbol{x} \end{pmatrix}^T \boldsymbol{A} \begin{pmatrix} 0 \\ \boldsymbol{x} \end{pmatrix}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

We see that the right-hand expression is taking a maximum over a special family of subspaces of dimension k: all the vectors in the family must have first coordinate 0. As the maximum over all subspaces of dimension k can only be larger, we immediately have

$$\alpha_k \geq \beta_k.$$

We may prove the inequalities in the other direction, such as $\beta_k \geq \alpha_{k+1}$, by replacing $\boldsymbol{A}$ and $\boldsymbol{B}$ with $-\boldsymbol{A}$ and $-\boldsymbol{B}$. □

Lemma 4.3.2. *For every $S \subseteq V$, let $d_{ave}(S)$ be the average degree of $G(S)$. Then,*

$$d_{ave}(S) \leq \mu_1.$$

Proof. If $\boldsymbol{M}$ is the adjacency matrix of G, then $\boldsymbol{M}(S)$ is the adjacency matrix of $G(S)$. Lemma 4.2.1 says that $d_{ave}(S)$ is at most the largest eigenvalue of the adjacency matrix of $G(S)$, and Theorem 4.3.1 says that this is at most μ_1. □

If we remove the vertex of smallest degree from a graph, the average degree can increase. On the other hand, Cauchy's Interlacing Theorem says that μ_1 can only decrease when we remove a vertex.

Lemma 4.3.2 is a good demonstration of Cauchy's Theorem. But, using Cauchy's Theorem to prove it was overkill. An more direct way to prove it is to emulate the proof of Lemma 4.2.1, but computing the quadratic form in the characteristic vector of S instead of $\mathbf{1}$.

4.4 Wilf's Theorem

We now apply Lemma 4.3.2 to obtain an upper bound on the chromatic number of a graph. Recall that a coloring of a graph is an assignment of colors to vertices in which adjacent vertices have distinct colors. A graph is said to be k-colorable if it can be colored with only k colors[1]. The *chromatic number* of a graph, written $\chi(G)$, is the least k for which G is k-colorable. The bipartite graphs are exactly the graph of chromatic number 2.

It is easy to show that every graph is $(d_{max}+1)$-colorable. Assign colors to the vertices one-by-one. As each vertex has at most d_{max} neighbors, there is always some color one can assign that vertex that is different than those assigned to its neighbors. The following theorem of Wilf [Wil67] improves upon this bound.

Theorem 4.4.1.

$$\chi(G) \leq \lfloor \mu_1 \rfloor + 1.$$

Proof. We prove this by induction on the number of vertices in the graph. To ground the induction, consider the graph with one vertex and no edges. It has chromatic number 1 and largest eigenvalue zero[2]. Now, assume the theorem is true for all graphs on $n-1$ vertices, and let G be a graph on n vertices. By Lemma 4.2.1, G has a vertex of degree at most $\lfloor \mu_1 \rfloor$. Let a be such a vertex and let $S = V \setminus \{a\}$. By Theorem 4.3.1, the largest eigenvalue of $G(S)$ is at most μ_1, and so our induction hypothesis implies that $G(S)$ has a coloring with at most $\lfloor \mu_1 \rfloor + 1$ colors. Let c be any such coloring. We just need to show that we can extend c to a. As a has at most $\lfloor \mu_1 \rfloor$ neighbors, there is some color in $\{1, \ldots, \lfloor \mu_1 \rfloor + 1\}$ that does not appear among its neighbors, and which it may be assigned. Thus, G has a coloring with $\lfloor \mu_1 \rfloor + 1$ colors. □

The simplest example in which this theorem improves over the naive bound of $d_{max}+1$ is the path graph on 3 vertices: it has $d_{\max} = 2$ but $\mu_1 < 2$. Thus, Wilf's theorem tells us that it can be colored with 2 colors. Star graphs provide more extreme examples. A star graph with n vertices has $d_{max} = n-1$ but $\mu_1 = \sqrt{n-1}$.

[1] To be precise, we often identify these k colors with the integers 1 through k. A k-coloring is then a function $c : \{1, \ldots, k\} \rightarrow V$ such that $c(a) \neq c(b)$ for all $(a, b) \in E$.

[2] If this makes you uncomfortable, you could use both graphs on two vertices

4.5 Perron-Frobenius Theory for symmetric matrices

The eigenvector corresponding to the largest eigenvalue of the adjacency matrix of a graph is usually not a constant vector. However, it is always a positive vector if the graph is connected. This follows from the Perron-Frobenius theory (discovered independently by Perron [Per07] and Frobenius [Fro12]). In fact, the Perron-Frobenius theory says much more, and it can be applied to adjacency matrices of strongly connected directed graphs. Note that these need not even be diagonalizable!

In the symmetric case, the theory is made much easier by both the spectral theory and the characterization of eigenvalues as extreme values of Rayleigh quotients. For a treatment of the general Perron-Frobenius theory, we recommend Seneta [Sen06] or Bapat and Raghavan [BR97].

Theorem 4.5.1. *[Perron-Frobenius, Symmetric Case] Let G be a connected weighted graph, let $\boldsymbol{M}$ be its adjacency matrix, and let $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ be its eigenvalues. Then*

a. The eigenvalue μ_1 has a strictly positive eigenvector,

b. $\mu_1 \geq -\mu_n$, and

c. $\mu_1 > \mu_2$.

Before proving Theorem 4.5.1, we will prove a lemma that will be used in the proof. It says that non-negative eigenvectors of non-negative adjacency matrices of connected graphs must be strictly positive.

Lemma 4.5.2. *Let G be a connected weighted graph (with non-negative edge weights), let $\boldsymbol{M}$ be its adjacency matrix, and assume that some non-negative vector $\boldsymbol{\phi}$ is an eigenvector of $\boldsymbol{M}$. Then, $\boldsymbol{\phi}$ is strictly positive.*

Proof. If $\boldsymbol{\phi}$ is not strictly positive, there is some vertex a for which $\boldsymbol{\phi}(a) = 0$. As G is connected, there must be some edge (b, c) for which $\boldsymbol{\phi}(b) = 0$ but $\boldsymbol{\phi}(c) > 0$. Let μ be the eigenvalue of $\boldsymbol{\phi}$. As $\boldsymbol{\phi}(b) = 0$, we obtain a contradiction from

$$\mu\boldsymbol{\phi}(b) = (\boldsymbol{M}\boldsymbol{\phi})(b) = \sum_{(b,z)\in E} w_{b,z}\boldsymbol{\phi}(z) \geq w_{b,c}\boldsymbol{\phi}(c) > 0,$$

where the inequalities follow from the fact that the terms $w_{b,z}$ and $\boldsymbol{\phi}(z)$ are non-negative.

So, we conclude that $\boldsymbol{\phi}$ must be strictly positive. □

Proof of Theorem 4.5.1. Let $\boldsymbol{\phi}_1$ be an eigenvector of μ_1 of norm 1, and construct the vector $\boldsymbol{x}$ such that

$$\boldsymbol{x}(u) = \left|\boldsymbol{\phi}_1(u)\right|, \text{for all } u.$$

We will show that $\boldsymbol{x}$ is an eigenvector of eigenvalue μ_1.

We have $\boldsymbol{x}^T\boldsymbol{x} = \boldsymbol{\phi}_1^T\boldsymbol{\phi}_1$. Moreover,

$$\mu_1 = \boldsymbol{\phi}_1^T \boldsymbol{M}\boldsymbol{\phi}_1 = \sum_{a,b} \boldsymbol{M}(a,b)\boldsymbol{\phi}_1(a)\boldsymbol{\phi}_1(b) \leq \sum_{a,b} \boldsymbol{M}(a,b)\left|\boldsymbol{\phi}_1(a)\right|\left|\boldsymbol{\phi}_1(b)\right| = \boldsymbol{x}^T\boldsymbol{M}\boldsymbol{x}.$$

So, the Rayleigh quotient of $\boldsymbol{x}$ is at least μ_1. As μ_1 is the maximum possible Rayleigh quotient for a unit vector, the Rayleigh quotient of $\boldsymbol{x}$ must be μ_1 and Theorem 2.2.1 implies that $\boldsymbol{x}$ must be an eigenvector of μ_1. As $\boldsymbol{x}$ is non-negative, Lemma 4.5.2 implies that it is strictly positive.

To prove part b, let $\boldsymbol{\phi}_n$ be the eigenvector of μ_n and let $\boldsymbol{y}$ be the vector for which $\boldsymbol{y}(u) = |\boldsymbol{\phi}_n(u)|$. In the spirit of the previous argument, we can again show that

$$|\mu_n| = |\boldsymbol{\phi}_n\boldsymbol{M}\boldsymbol{\phi}_n| \leq \sum_{a,b} \boldsymbol{M}(a,b)\boldsymbol{y}(a)\boldsymbol{y}(b) \leq \mu_1\boldsymbol{y}^T\boldsymbol{y} = \mu_1. \tag{4.3}$$

To show that the multiplicity of μ_1 is 1 (that is, $\mu_2 < \mu_1$), consider an eigenvector $\boldsymbol{\phi}_2$. As $\boldsymbol{\phi}_2$ is orthogonal to $\boldsymbol{\phi}_1$, it must contain both positive and negative values. We now construct the vector $\boldsymbol{y}$ such that $\boldsymbol{y}(u) = |\boldsymbol{\phi}_2(u)|$ and repeat the argument that we used for $\boldsymbol{x}$. We find that

$$\mu_2 = \boldsymbol{\phi}_2^T\boldsymbol{M}\boldsymbol{\phi}_2 \leq \boldsymbol{y}^T\boldsymbol{M}\boldsymbol{y} \leq \mu_1.$$

If $\mu_2 = \mu_1$, then $\boldsymbol{y}$ is a nonnegative eigenvector of eigenvalue μ_1, and so Lemma 4.5.2 says that it is strictly positive. Thus, $\boldsymbol{\phi}_2$ does not have any zero entries. As it has both positive and negative entries and the graph is connected, there must be some edge (a,b) for which $\boldsymbol{\phi}_2(a) < 0 < \boldsymbol{\phi}_2(b)$. Then the above inequality must be strict because the edge (a,b) will make a negative contribution to $\boldsymbol{\phi}_2^T\boldsymbol{M}\boldsymbol{\phi}_2$ and a positive contribution to $\boldsymbol{y}^T\boldsymbol{M}\boldsymbol{y}$. This contradicts our assumption that $\mu_2 = \mu_1$. □

Finally, we show that for a connected graph G, $\mu_n = -\mu_1$ if and only if G is bipartite. In fact, if $\mu_n = -\mu_1$, then $\mu_{n-i} = -\mu_{i+1}$ for every i.

Proposition 4.5.3. *If G is a connected graph and $\mu_n = -\mu_1$, then G is bipartite.*

Proof. Consider the conditions necessary to achieve equality in (4.3). First, $\boldsymbol{y}$ must be an eigenvector of eigenvalue μ_1. Thus, $\boldsymbol{y}$ must be strictly positive, $\boldsymbol{\phi}_n$ can not have any zero values, and there must be an edge (a,b) for which $\boldsymbol{\phi}_n(a) < 0 < \boldsymbol{\phi}_n(b)$. It must also be the case that all of the terms in

$$\sum_{(a,b)\in E} \boldsymbol{M}(a,b)\boldsymbol{\phi}_n(a)\boldsymbol{\phi}_n(b)$$

have the same sign, and we have established that this sign must be negative. Thus, for every edge (a,b), $\boldsymbol{\phi}_n(a)$ and $\boldsymbol{\phi}_n(b)$ must have different signs. That is, the signs provide the bipartition of the vertices. □

Proposition 4.5.4. *If G is bipartite then the eigenvalues of its adjacency matrix are symmetric about zero.*

Proof. As G is bipartite, we may divide its vertices into sets S and T so that all edges go between S and T. Let $\boldsymbol{\phi}$ be an eigenvector of $\boldsymbol{M}$ with eigenvalue μ. Define the vector $\boldsymbol{x}$ by

$$\boldsymbol{x}(a) = \begin{cases} \boldsymbol{\phi}(a) & \text{if } a \in S, and \\ -\boldsymbol{\phi}(a) & \text{if } a \in T. \end{cases}$$

To see that $\boldsymbol{x}$ is an eigenvector with eigenvalue $-\mu$, note that for $a \in S$,

$$(\boldsymbol{M}\boldsymbol{x})(a) = \sum_{(a,b) \in E} \boldsymbol{M}(a,b)\boldsymbol{x}(b) = \sum_{(a,b) \in E} \boldsymbol{M}(a,b)(-\boldsymbol{\phi}(b)) = -\mu\boldsymbol{\phi}(a) = -\mu\boldsymbol{x}(a).$$

We may similarly show that $(\boldsymbol{M}\boldsymbol{x})(a) = -\mu\boldsymbol{x}(a)$ for $a \in T$. □

Chapter 5

Comparing Graphs

5.1 Overview

It is rare than one can analytically determine the eigenvalues of an abstractly defined graph. Usually one is only able to prove loose bounds on some eigenvalues.

In this lecture we will see a powerful technique that allows one to compare one graph with another, and prove things like lower bounds on the smallest eigenvalue of a Laplacians. It often goes by the name "Poincaré Inequalities" (see [DS91, SJ89, GLM99]), or "Graphic inequalities".

5.2 The Loewner order

I begin by recalling an extremely useful piece of notation that is used in the Optimization community. For a symmetric matrix $\boldsymbol{A}$, we write

$$\boldsymbol{A} \succcurlyeq 0$$

if $\boldsymbol{A}$ is *positive semidefinite*. That is, if all of the eigenvalues of $\boldsymbol{A}$ are nonnegative, which is equivalent to

$$\boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} \geq \boldsymbol{0},$$

for all $\boldsymbol{v}$. We similarly write

$$\boldsymbol{A} \succcurlyeq \boldsymbol{B}$$

if

$$\boldsymbol{A} - \boldsymbol{B} \succcurlyeq \boldsymbol{0}$$

which is equivalent to

$$\boldsymbol{v}^T \boldsymbol{A} \boldsymbol{v} \geq \boldsymbol{v}^T \boldsymbol{B} \boldsymbol{v}$$

for all $\boldsymbol{v}$.

The relation $\succcurlyeq$ is called the Loewner *partial order*. It applies to some pairs of symmetric matrices, while others are incomparable. But, for all pairs to which it does apply, it acts like an order. For example, we have

$$\boldsymbol{A} \succcurlyeq \boldsymbol{B} \text{ and } \boldsymbol{B} \succcurlyeq \boldsymbol{C} \text{ implies } \boldsymbol{A} \succcurlyeq \boldsymbol{C},$$

and

$$\boldsymbol{A} \succcurlyeq \boldsymbol{B} \text{ implies } \boldsymbol{A} + \boldsymbol{C} \succcurlyeq \boldsymbol{B} + \boldsymbol{C},$$

for symmetric matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$.

We will overload this notation by defining it for graphs as well. Thus, we write

$$G \succcurlyeq H$$

if $\boldsymbol{L}_G \succcurlyeq \boldsymbol{L}_H$. When we write this, we are always describing an inequality on Laplacian matrices.

For example, if $G = (V, E)$ is a graph and $H = (V, F)$ is a subgraph of G, then

$$\boldsymbol{L}_G \succcurlyeq \boldsymbol{L}_H.$$

To see this, recall the Laplacian quadratic form:

$$\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x} = \sum_{(u,v) \in E} w_{u,v} (\boldsymbol{x}(u) - \boldsymbol{x}(v))^2.$$

It is clear that dropping edges can only decrease the value of the quadratic form. The same holds for decreasing the weights of edges.

This notation is particularly useful when we consider some multiple of a graph, such as when we write

$$G \succcurlyeq c \cdot H,$$

for some $c > 0$. What is $c \cdot H$? It is the same graph as H, but the weight of every edge is multiplied by c.

We usually use this notation for the inequalities it implies on the eigenvalues of $\boldsymbol{L}_G$ and $\boldsymbol{L}_H$.

Lemma 5.2.1. *If G and H are graphs such that*

$$G \succcurlyeq c \cdot H,$$

then

$$\lambda_k(G) \geq c\lambda_k(H),$$

for all k.

Proof. The Courant-Fischer Theorem tells us that

$$\lambda_k(G) = \min_{\substack{S \subseteq \mathbb{R}^n \\ \dim(S)=k}} \max_{\boldsymbol{x} \in S} \frac{\boldsymbol{x}^T L_G \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \geq c \min_{\substack{S \subseteq \mathbb{R}^n \\ \dim(S)=k}} \max_{\boldsymbol{x} \in S} \frac{\boldsymbol{x}^T L_H \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = c\lambda_k(H).$$

□

Corollary 5.2.2. *Let G be a graph and let H be obtained by either adding an edge to G or increasing the weight of an edge in G. Then, for all a*

$$\lambda_i(G) \leq \lambda_i(H).$$

5.3 Approximations of Graphs

We consider one graph to be a good approximation of another if their Laplacian quadratic forms are similar. For example, we will say that H is a c-approximation of G if

$$cH \succcurlyeq G \succcurlyeq H/c.$$

Surprising approximations exist. For example, random regular and random Erdös-Rényi graphs are good approximations of complete graphs. We will encounter infinite families of expander graphs that for every $\epsilon > 0$ provide a $d > 0$ such that for all $n > 0$ there is a d-regular graph G_n that is a $(1+\epsilon)$-approximation of K_n. As d is fixed, such a graph has many fewer edges than a complete graph!

In Chapters **??** and **??** we will also prove that every graph can be well-approximated by a sparse graph.

5.4 The Path Inequality

By now you should be wondering, "how do we prove that $G \succcurlyeq c \cdot H$ for some graph G and H?" Not too many ways are known. We'll do it by proving some inequalities of this form for some of the simplest graphs, and then extending them to more general graphs. For example, we will prove

$$(n-1) \cdot P_n \succcurlyeq G_{1,n}, \tag{5.1}$$

where P_n is the path from vertex 1 to vertex n, and $G_{1,n}$ is the graph with just the edge $(1,n)$. All of these edges are unweighted.

The following very simple proof of this inequality was discovered by Sam Daitch.

Lemma 5.4.1.

$$(n-1) \cdot P_n \succcurlyeq G_{1,n}.$$

Proof. We need to show that for every $\boldsymbol{x} \in \mathbb{R}^n$,

$$(n-1)\sum_{a=1}^{n-1}(\boldsymbol{x}(a+1)-\boldsymbol{x}(a))^2 \geq (\boldsymbol{x}(n)-\boldsymbol{x}(1))^2.$$

For $1 \leq a \leq n-1$, set

$$\boldsymbol{\Delta}(a) = \boldsymbol{x}(a+1) - \boldsymbol{x}(a).$$

The inequality we need to prove then becomes

$$(n-1)\sum_{a=1}^{n-1}\boldsymbol{\Delta}(a)^2 \geq \left(\sum_{a=1}^{n-1}\boldsymbol{\Delta}(a)\right)^2.$$

But, this is just the Cauchy-Schwartz inequality. I'll remind you that Cauchy-Schwartz follows from the fact that the inner product of two vectors is at most the product of their norms. In this case, those vectors are $\boldsymbol{\Delta}$ and the all-ones vector of length $n-1$:

$$\left(\sum_{a=1}^{n-1} \boldsymbol{\Delta}(a)\right)^2 = \left(\mathbf{1}_{n-1}^T \boldsymbol{\Delta}\right)^2 \leq \left(\left\|\mathbf{1}_{n-1}\right\| \left\|\boldsymbol{\Delta}\right\|\right)^2 = \left\|\mathbf{1}_{n-1}\right\|^2 \left\|\boldsymbol{\Delta}\right\|^2 = (n-1)\sum_{i=1}^{n-1} \boldsymbol{\Delta}(i)^2.$$

□

5.4.1 Bounding λ_2 of a Path Graph

In Lemma 6.6.1 we will prove that $\lambda_2(P_n) \approx \pi^2/n^2$. For now, we demonstrate the power of Lemma 5.4.1 by using it to prove a lower bound on $\lambda_2(P_n)$ that is very close to this.

To prove a lower bound on $\lambda_2(P_n)$, we will prove that some multiple of the path is at least the complete graph. To this end, write

$$L_{K_n} = \sum_{a<b} L_{G_{a,b}},$$

and recall that

$$\lambda_2(K_n) = n.$$

For every $a < b$, let $P_{a,b}$ be the subgraph of the path graph induced on vertices with indices between a and b. Note that this is itself a path graph of length $b-a$.

For every edge (a,b) in the complete graph, we apply the only inequality available in the path:

$$G_{a,b} \preccurlyeq (b-a)P_{a,b} \preccurlyeq (b-a)P_n. \tag{5.2}$$

This inequality says that $G_{a,b}$ is at most $(b-a)$ times the part of the path connecting a to b, and that this part of the path is less than the whole.

Summing inequality (5.2) over all edges $(a,b) \in K_n$ gives

$$K_n = \sum_{a,b} G_{a,b} \preccurlyeq \sum_{a,b} (b-a)P_n.$$

To finish the proof, we compute

$$\sum_{1 \leq a < b \leq n} (b-a) = \sum_{c=1}^{n-1} c(n-c) = n(n+1)(n-1)/6.$$

So,

$$L_{K_n} \preccurlyeq \frac{n(n+1)(n-1)}{6} L_{P_n}.$$

Applying Lemma 5.2.1, we obtain

$$\frac{6}{(n+1)(n-1)} \leq \lambda_2(P_n).$$

5.5 The Complete Binary Tree

Let's do the same analysis with the complete binary tree.

One way of understanding the complete binary tree of depth $d+1$ is to identify the vertices of the tree with strings over $\{0,1\}$ of length at most d. The root of the tree is the empty string. Every other node has one ancestor, which is obtained by removing the last character of its string, and two children, which are obtained by appending one character to its label.

Alternatively, you can describe it as the graph on $n = 2^{d+1} - 1$ nodes with edges of the form $(i, 2i)$ and $(i, 2i+1)$ for $i < n$. We will name this graph T_d. See figure 5.1 for pictures of these.

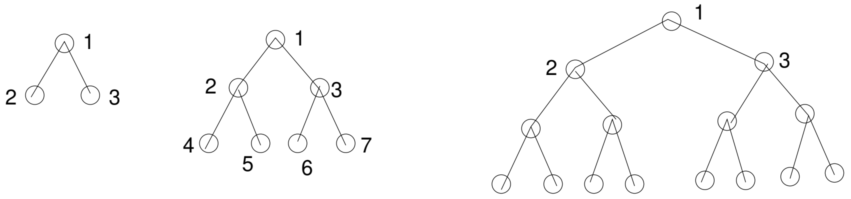


Figure 5.1: T_1, T_2 and T_3. Node 1 is at the top, 2 and 3 are its children. Some other nodes have been labeled as well.

Let's first upper bound $\lambda_2(T_d)$ by constructing a test vector $\boldsymbol{x}$. Set $\boldsymbol{x}(1) = 0$, $\boldsymbol{x}(2) = 1$, and $\boldsymbol{x}(3) = -1$. Then, for every vertex u that we can reach from node 2 without going through node 1, we set $\boldsymbol{x}(a) = 1$. For all the other nodes, we set $\boldsymbol{x}(a) = -1$.

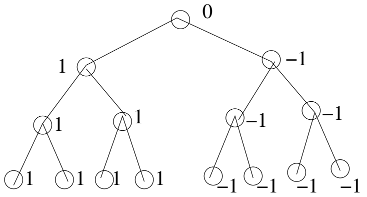


Figure 5.2: The test vector we use to upper bound $\lambda_2(T_3)$.

We have constructed $\boldsymbol{x}$ symmetrically, so that $\mathbf{1}^T \boldsymbol{x} = 0$. Thus,

$$\lambda_2 \leq \frac{\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \frac{\sum_{a \sim b} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2}{\sum_i \boldsymbol{x}(a)^2} = \frac{(\boldsymbol{x}(1) - \boldsymbol{x}(2))^2 + (\boldsymbol{x}(1) - \boldsymbol{x}(3))^2}{n-1} = 2/(n-1).$$

We will again prove a lower bound by comparing T_d to the complete graph. For each edge $a < b$, let $T_d^{a,b}$ denote the unique path in T from a to b. This path will have length at most $2d \leq 2 \log_2 n$. So, we have

$$K_n = \sum_{a<b} G_{a,b} \preccurlyeq \sum_{a<b} (2d) T_d^{a,b} \preccurlyeq \sum_{a<b} (2 \log_2 n) T_d = \binom{n}{2} (2 \log_2 n) T_d.$$

So, we obtain the bound

$$\binom{n}{2} (2 \log_2 n) \lambda_2(T_d) \geq n,$$

which implies

$$\lambda_2(T_d) \geq \frac{1}{(n-1)\log_2 n}.$$

Using the generalization of Lemma 5.4.1 presented in the next section, one can improve this lower bound to $1/cn$ for some constant c.

5.6 The weighted path

We now generalize the the inequality in Lemma 5.4.1 to weighted path graphs. Allowing for weights on the edges of the path greatly extends it applicability.

Lemma 5.6.1. *Let $w_1, \dots, w_{n-1}$ be positive. Then*

$$G_{1,n} \preccurlyeq \left(\sum_{i=1}^{n-1} \frac{1}{w_a} \right) \sum_{a=1}^{n-1} w_a G_{a,a+1}.$$

Proof. Let $\boldsymbol{x} \in \mathbb{R}^n$ and set $\boldsymbol{\Delta}(a)$ as in the proof of Lemma 5.4.1 Now, set

$$\gamma(a) = \boldsymbol{\Delta}(a)\sqrt{w_a}.$$

Let $\boldsymbol{w}^{-1/2}$ denote the vector for which

$$\boldsymbol{w}^{-1/2}(a) = \frac{1}{\sqrt{w_a}}.$$

Then,

$$\sum_a \boldsymbol{\Delta}(a) = \boldsymbol{\gamma}^T \boldsymbol{w}^{-1/2},$$

$$\left\| \boldsymbol{w}^{-1/2} \right\|^2 = \sum_a \frac{1}{w_a},$$

and

$$\|\boldsymbol{\gamma}\|^2 = \sum_a \boldsymbol{\Delta}(a)^2 w_a.$$

So,

$$\begin{aligned}
\boldsymbol{x}^T L_{G_{1,n}} \boldsymbol{x} &= \left(\sum_a \boldsymbol{\Delta}(a) \right)^2 = \left(\boldsymbol{\gamma}^T \boldsymbol{w}^{-1/2} \right)^2 \\
&\leq \left(\|\boldsymbol{\gamma}\| \left\| \boldsymbol{w}^{-1/2} \right\| \right)^2 = \left(\sum_a \frac{1}{w_a} \right) \sum_a \boldsymbol{\Delta}(a)^2 w_a = \left(\sum_a \frac{1}{w_a} \right) \boldsymbol{x}^T \left(\sum_{a=1}^{n-1} w_a L_{G_{a,a+1}} \right) \boldsymbol{x}.
\end{aligned}$$

□

5.7 Exercises

1. Let $\boldsymbol{v}$ be a vector so that $\boldsymbol{v}^T\mathbf{1} = 0$. Prove that

$$\|\boldsymbol{v}\|^2 \leq \|\boldsymbol{v} + t\mathbf{1}\|^2,$$

for every real number t.

Part II

The Zoo of Graphs

Chapter 6

Fundamental Graphs

We will bound and derive the eigenvalues of the Laplacian matrices of some fundamental graphs, including complete graphs, star graphs, ring graphs, path graphs, and products of these that yield grids and hypercubes. As all these graphs are connected, they all have eigenvalue zero with multiplicity one. We will have to do some work to compute the other eigenvalues.

We will see in Part IV that the Laplacian eigenvalues that reveal the most about a graph are the smallest and largest ones. To interpret the smallest eigenvalues, we will exploit a relation between λ_2 and the isoperimetric ratio of a graph that is derived in Chapter 20, and which we state here for convenience:

For every $S \subset V$,

$$\theta(S) \geq \lambda_2(1 - s),$$

where $s = |S| \,/\, |V|$ and

$$\theta(S) \stackrel{\text{def}}{=} \frac{|\partial(S)|}{|S|}$$

is the *isoperimetric ratio* of S.

6.1 The complete graph

The complete graph on n vertices, K_n, has edge set $\{(a,b) : a \neq b\}$.

Lemma 6.1.1. *The Laplacian of K_n has eigenvalue 0 with multiplicity 1 and n with multiplicity $n-1$.*

Proof. To compute the non-zero eigenvalues, let $\boldsymbol{\psi}$ be any non-zero vector orthogonal to the all-1s vector, so

$$\sum_{a} \boldsymbol{\psi}(a) = 0. \tag{6.1}$$

We now compute the first coordinate of $\boldsymbol{L}_{K_n}\boldsymbol{\psi}$. Using (3.3), the expression for the action of the Laplacian as an operator, we find

$$\left(\boldsymbol{L}_{K_n}\boldsymbol{\psi}\right)(1) = \sum_{v\geq 2}(\boldsymbol{\psi}(1) - \boldsymbol{\psi}(b)) = (n-1)\boldsymbol{\psi}(1) - \sum_{v=2}^{n}\boldsymbol{\psi}(b) = n\boldsymbol{\psi}(1), \quad \text{by (6.1)}.$$

As the choice of coordinate was arbitrary, we have $\boldsymbol{L}\boldsymbol{\psi} = n\boldsymbol{\psi}$. So, every vector orthogonal to the all-1s vector is an eigenvector of eigenvalue n. □

Alternative approach. Observe that $\boldsymbol{L}_{K_n} = n\boldsymbol{I} - \mathbf{1}\mathbf{1}^T$. □

We often think of the Laplacian of the complete graph as being a scaling of the identity. For every $\boldsymbol{x}$ orthogonal to the all-1s vector, $\boldsymbol{L}\boldsymbol{x} = n\boldsymbol{x}$.

Now, let's see how our bound on the isoperimetric ratio works out. Let $S \subset [n]$. Every vertex in S has $n - |S|$ edges connecting it to vertices not in S. So,

$$\theta(S) = \frac{|S|\,(n-|S|}{|S|} = n - |S| = \lambda_2(\boldsymbol{L}_{K_n})(1-s),$$

where $s = |S|/n$. Thus, Theorem 20.1.1 is sharp for the complete graph.

6.2 The star graphs

The star graph on n vertices S_n has edge set $\{(1,a) : 2 \leq a \leq n\}$.

To determine the eigenvalues of S_n, we first observe that each vertex $a \geq 2$ has degree 1, and that each of these degree-one vertices has the same neighbor. Whenever two degree-one vertices share the same neighbor, they provide an eigenvector of eigenvalue 1.

Lemma 6.2.1. *Let $G = (V,E)$ be a graph, and let a and b be vertices of degree one that are both connected to another vertex c. Then, the vector $\boldsymbol{\psi} = \boldsymbol{\delta}_a - \boldsymbol{\delta}_b$ is an eigenvector of $\boldsymbol{L}_G$ of eigenvalue 1.*

Proof. Just multiply $\boldsymbol{L}_G$ by $\boldsymbol{\psi}$, and check (using (3.3)) vertex-by-vertex that it equals $\boldsymbol{\psi}$. □

As eigenvectors of different eigenvalues are orthogonal, this implies that $\boldsymbol{\psi}(a) = \boldsymbol{\psi}(b)$ for every eigenvector with eigenvalue different from 1.

Lemma 6.2.2. *The graph S_n has eigenvalue 0 with multiplicity 1, eigenvalue 1 with multiplicity $n-2$, and eigenvalue n with multiplicity 1.*

Proof. Applying Lemma 6.2.1 to vertices i and $i+1$ for $2 \leq i < n$, we find $n-2$ linearly independent eigenvectors of the form $\boldsymbol{\delta}_i - \boldsymbol{\delta}_{i+1}$, all with eigenvalue 1. As 0 is also an eigenvalue, only one eigenvalue remains to be determined.

Recall that the trace of a matrix equals both the sum of its diagonal entries and the sum of its eigenvalues. We know that the trace of $\boldsymbol{L}_{S_n}$ is $2n-2$, and we have identified $n-1$ eigenvalues that sum to $n-2$. So, the remaining eigenvalue must be n.

To determine the corresponding eigenvector, recall that it must be orthogonal to the other eigenvectors we have identified. This tells us that it must have the same value at each of the points of the star. Let this value be 1, and let x be the value at vertex 1. As the eigenvector is orthogonal to the constant vectors, it must be that

$$(n-1)+x=0,$$

so $x=-(n-1)$. □

6.3 Products of graphs

We now define a product on graphs. If we apply this product to two paths, we obtain a grid. If we apply it repeatedly to one edge, we obtain a hypercube.

Definition 6.3.1. *Let $G=(V,E,v)$ and $H=(W,F,w)$ be weighted graphs. Then $G \times H$ is the graph with vertex set $V \times W$ and edge set*

$$\Big((a,b),(\widehat{a},b)\Big) \text{ with weight } v_{a,\widehat{a}}, \text{ where } (a,\widehat{a}) \in E \text{ and}$$
$$\Big((a,b),(a,\widehat{b})\Big) \text{ with weight } w_{b,\widehat{b}}, \text{ where } (b,\widehat{b}) \in F.$$

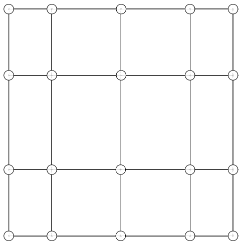


Figure 6.1: An m-by-n grid graph is the product of a path on m vertices with a path on n vertices. This is a drawing of a 5-by-4 grid made using Hall's algorithm.

Theorem 6.3.2. *Let $G=(V,E,v)$ and $H=(W,F,w)$ be weighted graphs with Laplacian eigenvalues $\lambda_1,\ldots,\lambda_n$ and $\mu_1,\ldots,\mu_m$, and eigenvectors $\boldsymbol{\alpha}_1,\ldots,\boldsymbol{\alpha}_n$ and $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_m$, respectively. Then, for each $1 \leq i \leq n$ and $1 \leq j \leq m$, $G \times H$ has an eigenvector $\boldsymbol{\gamma}_{i,j}$ of eigenvalue $\lambda_i+\mu_j$ such that*

$$\boldsymbol{\gamma}_{i,j}(a,b)=\boldsymbol{\alpha}_i(a)\boldsymbol{\beta}_j(b).$$

Proof. Let $\boldsymbol{\alpha}$ be an eigenvector of L_G of eigenvalue λ, let $\boldsymbol{\beta}$ be an eigenvector of L_H of eigenvalue μ, and let $\boldsymbol{\gamma}$ be defined as above.

To see that $\boldsymbol{\gamma}$ is an eigenvector of eigenvalue $\lambda + \mu$, we compute

$$\begin{aligned}(\boldsymbol{L}\boldsymbol{\gamma})(a,b) &= \sum_{(a,\widehat{a})\in E} v_{a,\widehat{a}}\left(\boldsymbol{\gamma}(a,b) - \boldsymbol{\gamma}(\widehat{a},b)\right) + w_{b,\widehat{b}} \sum_{(b,\widehat{b})\in F} \left(\boldsymbol{\gamma}(a,b) - \boldsymbol{\gamma}(a,\widehat{b})\right)\\
&= \sum_{(a,\widehat{a})\in E} v_{a,\widehat{a}}\left(\boldsymbol{\alpha}(a)\boldsymbol{\beta}(b) - \boldsymbol{\alpha}(\widehat{a})\boldsymbol{\beta}(b)\right) + \sum_{(b,\widehat{b})\in F} w_{b,\widehat{b}}\left(\boldsymbol{\alpha}(a)\boldsymbol{\beta}(b) - \boldsymbol{\alpha}(a)\boldsymbol{\beta}(\widehat{b})\right)\\
&= \sum_{(a,\widehat{a})\in E} v_{a,\widehat{a}}\boldsymbol{\beta}(b)\left(\boldsymbol{\alpha}(a) - \boldsymbol{\alpha}(\widehat{a})\right) + \sum_{(b,\widehat{b})\in F} w_{b,\widehat{b}}\boldsymbol{\alpha}(a)\left(\boldsymbol{\beta}(b) - \boldsymbol{\beta}(\widehat{b})\right)\\
&= \sum_{(a,\widehat{a})\in E} \boldsymbol{\beta}(b)\lambda\boldsymbol{\alpha}(a) + \sum_{(b,\widehat{b})\in F} \boldsymbol{\alpha}(a)\mu\boldsymbol{\beta}(b)\\
&= (\lambda+\mu)(\boldsymbol{\alpha}(a)\boldsymbol{\beta}(b)).\end{aligned}$$

□

An alternative approach to defining the graph product and proving Theorem 6.3.2 is via Kronecker products. $G \times H$ is the graph with Laplacian matrix

$$(\boldsymbol{L}_G \otimes \boldsymbol{I}_W) + (\boldsymbol{I}_V \otimes \boldsymbol{L}_H).$$

6.3.1 The Hypercube

The d-dimensional hypercube graph, H_d, is the graph with vertex set $\{0,1\}^d$, with edges between vertices whose names differ in exactly one bit. The hypercube may also be expressed as the product of the one-edge graph with itself $d-1$ times.

Let H_1 be the graph with vertex set $\{0,1\}$ and one edge between those vertices. It's Laplacian matrix has eigenvalues 0 and 2. As $H_d = H_{d-1} \times H_1$, we may use this to calculate the eigenvalues and eigenvectors of H_d for every d.

The eigenvectors of H_1 are

$$\begin{pmatrix}1\\1\end{pmatrix} \quad \text{and} \quad \begin{pmatrix}1\\-1\end{pmatrix},$$

with eigenvalues 0 and 2, respectively. Thus, if $\boldsymbol{\psi}$ is an eigenvector of H_{d-1} with eigenvalue λ, then

$$\begin{pmatrix}\boldsymbol{\psi}\\\boldsymbol{\psi}\end{pmatrix} \quad \text{and} \quad \begin{pmatrix}\boldsymbol{\psi}\\-\boldsymbol{\psi}\end{pmatrix},$$

are eigenvectors of H_d with eigenvalues λ and $\lambda + 2$, respectively. This means that H_d has eigenvalue $2i$ for each $0 \leq i \leq d$ with multiplicity $\binom{d}{i}$. Moreover, each eigenvector of H_d can be identified with a vector $\boldsymbol{y} \in \{0,1\}^d$:

$$\boldsymbol{\psi}_{\boldsymbol{y}}(\boldsymbol{x}) = (-1)^{\boldsymbol{y}^T\boldsymbol{x}},$$

where $\boldsymbol{x} \in \{0,1\}^d$ ranges over the vertices of H_d. Each $\boldsymbol{y} \in \{0,1\}^{d-1}$ indexing an eigenvector of H_{d-1} leads to the eigenvectors of H_d indexed by $(\boldsymbol{y}, 0)$ and $(\boldsymbol{y}, 1)$.

Using Theorem 20.1.1 and the fact that $\lambda_2(H_d) = 2$, we can immediately prove the following isoperimetric theorem for the hypercube.

Corollary 6.3.3.

$$\theta_{H_d} \geq 1.$$

In particular, for every set of at most half the vertices of the hypercube, the number of edges on the boundary of that set is at least the number of vertices in that set.

This result is tight, as you can see by considering one face of the hypercube, such as all the vertices whose labels begin with 0. It is possible to prove this by more concrete combinatorial means. In fact, very precise analyses of the isoperimetry of sets of vertices in the hypercube can be obtained. See [Har76] or [Bol86].

6.4 Bounds on λ_2 by test vectors

If we can guess an approximation of $\boldsymbol{\psi}_2$, we can often plug it in to the Laplacian quadratic form to obtain a good upper bound on λ_2. The Courant-Fischer Theorem tells us that every vector $\boldsymbol{v}$ orthogonal to $\mathbf{1}$ provides an upper bound on λ_2:

$$\lambda_2 \leq \frac{\boldsymbol{v}^T \boldsymbol{L} \boldsymbol{v}}{\boldsymbol{v}^T \boldsymbol{v}}.$$

When we use a vector $\boldsymbol{v}$ in this way, we call it a *test vector.*

Let's see what a test vector can tell us about λ_2 of a path graph on n vertices. I would like to use the vector that assigns i to vertex a as a test vector, but it is not orthogonal to $\mathbf{1}$. So, we will use the next best thing. Let $\boldsymbol{x}$ be the vector such that $\boldsymbol{x}(a) = (n+1) - 2a$, for $1 \leq a \leq n$. This vector satisfies $\boldsymbol{x} \perp \mathbf{1}$, so

$$\begin{aligned}
\lambda_2(P_n) &\leq \frac{\sum_{1 \leq a < n} (x(a) - x(a+1))^2}{\sum_a x(a)^2} \\
&= \frac{\sum_{1 \leq a < n} 2^2}{\sum_a (n+1-2a)^2} \\
&= \frac{4(n-1)}{(n+1)n(n-1)/3} && \text{(clearly, the denominator is } n^3/c \text{ for some } c) \\
&= \frac{12}{n(n+1)}. && (6.2)
\end{aligned}$$

We will soon see that this bound is of the right order of magnitude. Thus, Theorem 20.1.1 does not provide a good bound on the isoperimetric ratio of the path graph. The isoperimetric ratio is minimized by the set $S = \{1, \ldots, n/2\}$, which has $\theta(S) = 2/n$. However, the upper bound provided by Theorem 20.1.1 is of the form c/n. Cheeger's inequality, which appears in Chapter 21, will tell us that the error of this approximation can not be worse than quadratic.

The Courant-Fischer theorem is not as helpful when we want to prove lower bounds on λ_2. To prove lower bounds, we need the form with a maximum on the outside, which gives

$$\lambda_2 \geq \max_{S:\dim(S)=n-1} \min_{\boldsymbol{v}\in S} \frac{\boldsymbol{v}^T \boldsymbol{L}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{v}}.$$

This is not too helpful, as it is difficult to prove lower bounds on

$$\min_{\boldsymbol{v}\in S} \frac{\boldsymbol{v}^T \boldsymbol{L}\boldsymbol{v}}{\boldsymbol{v}^T\boldsymbol{v}}$$

over a space S of large dimension. We will see a technique that lets us prove such lower bounds next lecture.

But, first we compute the eigenvalues and eigenvectors of the path graph exactly.

6.5 The Ring Graph

The ring graph on n vertices, R_n, may be viewed as having a vertex set corresponding to the integers modulo n. In this case, we view the vertices as the numbers 0 through $n-1$, with edges $(a, a+1)$, computed modulo n.

Lemma 6.5.1. *The Laplacian of R_n has eigenvectors*

$$\begin{aligned} \boldsymbol{x}_k(a) &= \cos(2\pi k a/n), \text{ and} \\ \boldsymbol{y}_k(a) &= \sin(2\pi k a/n), \end{aligned}$$

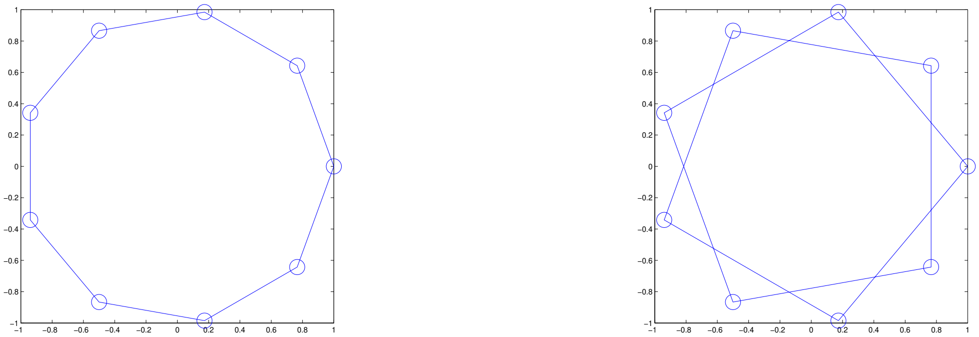
for $0 \leq k \leq n/2$, ignoring $\boldsymbol{y}_0$ which is the all-zero vector, and for even n ignoring $\boldsymbol{y}_{n/2}$ for the same reason. Eigenvectors $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ have eigenvalue $2 - 2\cos(2\pi k/n)$.

Note that $\boldsymbol{x}_0$ is the all-ones vector. When n is even, we only have $\boldsymbol{x}_{n/2}$, which alternates ±1.

Proof. We will first see that $\boldsymbol{x}_1$ and $\boldsymbol{y}_1$ are eigenvectors by drawing the ring graph on the unit circle in the natural way: plot vertex a at point $(\cos(2\pi a/n), \sin(2\pi a/n))$.

You can see that the average of the neighbors of a vertex is a vector pointing in the same direction as the vector associated with that vertex. This should make it obvious that both the x and y coordinates in this figure are eigenvectors of the same eigenvalue. The same holds for all k.

Alternatively, we can verify that these are eigenvectors by a simple computation.



(a) The ring graph on 9 vertices.

(b) The eigenvectors for $k = 2$.

Figure 6.2:

$$\begin{aligned}
\left(L_{R_n} \boldsymbol{x}_k\right)(a) &= 2\boldsymbol{x}_k(a) - \boldsymbol{x}_k(a+1) - \boldsymbol{x}_k(a-1) \\
&= 2\cos(2\pi ka/n) - \cos(2\pi k(a+1)/n) - \cos(2\pi k(a-1)/n) \\
&= 2\cos(2\pi ka/n) - \cos(2\pi ka/n)\cos(2\pi k/n) + \sin(2\pi ka/n)\sin(2\pi k/n) \\
&\qquad - \cos(2\pi ka/n)\cos(2\pi k/n) - \sin(2\pi ka/n)\sin(2\pi k/n) \\
&= 2\cos(2\pi ka/n) - \cos(2\pi ka/n)\cos(2\pi k/n) - \cos(2\pi ka/n)\cos(2\pi k/n) \\
&= (2 - 2\cos(2\pi k/n))\cos(2\pi ka/n) \\
&= (2 - \cos(2\pi k/n))\boldsymbol{x}_k(a).
\end{aligned}$$

The computation for $\boldsymbol{y}_k$ follows similarly. □

6.6 The Path Graph

We will derive the eigenvalues and eigenvectors of the path graph from those of the ring graph. To begin, I will number the vertices of the ring a little differently, as in Figure 6.3.

Lemma 6.6.1. *Let $P_n = (V, E)$ where $V = \{1, \dots, n\}$ and $E = \{(a, a+1) : 1 \leq a < n\}$. The Laplacian of P_n has the same eigenvalues as R_{2n}, excluding 2. That is, P_n has eigenvalues namely $2(1 - \cos(\pi k/n))$, and eigenvectors*

$$\boldsymbol{v}_k(a) = \cos(\pi ka/n - \pi k/2n).$$

for $0 \leq k < n$

Proof. We derive the eigenvectors and eigenvalues by treating P_n as a quotient of R_{2n}: we will identify vertex a of P_n with vertices a and $a+n$ of R_{2n} (under the new numbering of R_{2n}). These are pairs of vertices that are above each other in the figure that I drew.

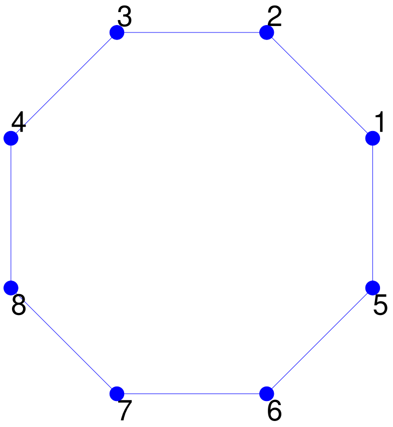


Figure 6.3: The ring on 8 vertices, numbered differently

Let $\boldsymbol{I}_n$ be the n-dimensional identity matrix. You should check that

$$\begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{I}_n \end{pmatrix} \boldsymbol{L}_{R_{2n}} \begin{pmatrix} \boldsymbol{I}_n \\ \boldsymbol{I}_n \end{pmatrix} = 2\boldsymbol{L}_{P_n}.$$

If there is an eigenvector $\boldsymbol{\psi}$ of R_{2n} with eigenvalue λ for which $\boldsymbol{\psi}(a) = \boldsymbol{\psi}(a+n)$ for $1 \leq a \leq n$, then the above equation gives us a way to turn this into an eigenvector of P_n: Let $\boldsymbol{\phi} \in \mathbb{R}^n$ be the vector for which

$$\boldsymbol{\phi}(a) = \boldsymbol{\psi}(a), \text{for } 1 \leq a \leq n.$$

Then,

$$\begin{pmatrix} \boldsymbol{I}_n \\ \boldsymbol{I}_n \end{pmatrix} \boldsymbol{\phi} = \boldsymbol{\psi}, \quad \boldsymbol{L}_{R_{2n}} \begin{pmatrix} \boldsymbol{I}_n \\ \boldsymbol{I}_n \end{pmatrix} \boldsymbol{\phi} = \lambda \boldsymbol{\psi}, \quad \text{and} \quad \begin{pmatrix} \boldsymbol{I}_n & \boldsymbol{I}_n \end{pmatrix} \boldsymbol{L}_{R_{2n}} \begin{pmatrix} \boldsymbol{I}_n \\ \boldsymbol{I}_n \end{pmatrix} \boldsymbol{\psi} = 2\lambda \boldsymbol{\phi}.$$

So, if we can find such a vector $\boldsymbol{\psi}$, then the corresponding $\boldsymbol{\phi}$ is an eigenvector of P_n of eigenvalue λ.

As you've probably guessed, we can find such vectors $\boldsymbol{\psi}$. I've drawn one in Figure 6.3. For each of the two-dimensional eigenspaces of R_{2n}, we get one such a vector. These provide eigenvectors of eigenvalue

$$2(1 - \cos(\pi k/n)),$$

for $1 \leq k < n$. Thus, we now know $n-1$ distinct eigenvalues. The last, of course, is zero. □

The type of quotient used in the above argument is known as an *equitable partition*. You can find a extensive exposition of these in Godsil's book [God93].

Chapter 7

Cayley Graphs

7.1 Cayley Graphs

Ring graphs and hypercubes are types of Cayley graph. In general, the vertices of a Cayley graph are the elements of some group Γ. In the case of the ring, the group is the set of integers modulo n. The edges of a Cayley graph are specified by a set $S \subset \Gamma$, which are called the *generators* of the Cayley graph. The set of generators must be closed under inverse. That is, if $s \in S$, then $s^{-1} \in S$. Vertices $u, v \in \Gamma$ are connected by an edge if there is an $s \in S$ such that

$$u \circ s = v,$$

where $\circ$ is the group operation. In the case of Abelian groups, like the integers modulo n, this would usually be written $u + s = v$. The generators of the ring graph are $\{1, -1\}$.

The d-dimensional hypercube, H_d, is a Cayley graph over the additive group $(\mathbf{Z}/2\mathbf{Z})^d$: that is the set of vectors in $\{0,1\}^d$ under addition modulo 2. The generators are given by the vectors in $\{0,1\}^d$ that have a 1 in exactly one position. This set is closed under inverse, because every element of this group is its own inverse.

We require S to be closed under inverse so that the graph is undirected:

$$u + s = v \qquad \Longleftrightarrow \qquad v + (-s) = u.$$

Cayley graphs over Abeliean groups are particularly convenient because we can find an orthonormal basis of eigenvectors without knowing the set of generators. They just depend on the group[1]. Knowing the eigenvectors makes it much easier to compute the eigenvalues. We give the computations of the eigenvectors in sections 7.4 and 7.8.

We will now examine two exciting types of Cayley graphs: Paley graphs and generalized hypercubes.

[1] More precisely, the characters always form an orthonormal set of eigenvectors, and the characters just depend upon the group. When two different characters have the same eigenvalue, we obtain an eigenspace of dimension greater than 1. These eigenspaces do depend upon the choice of generators.

7.2 Paley Graphs

The Paley graph are Cayley graphs over the group of integer modulo a prime, p, where p is equivalent to 1 modulo 4. Such a group is often written $\mathbb{Z}/p$.

I should begin by reminding you a little about the integers modulo p. The first thing to remember is that the integers modulo p are actually a field, written $\mathbb{F}_p$. That is, they are closed under both addition and multiplication (completely obvious), have identity elements under addition and multiplication (0 and 1), and have inverses under addition and multiplication. It is obvious that the integers have inverses under addition: $-x$ modulo p plus x modulo p equals 0. It is a little less obvious that the integers modulo p have inverses under multiplication (except that 0 does not have a multiplicative inverse). That is, for every $x \neq 0$, there is a y such that $xy = 1$ modulo p. When we write $1/x$, we mean this element y.

The generators of the Paley graphs are the squares modulo p (usually called the *quadratic residues*). That is, the set of numbers s such that there exits an x for which $x^2 \equiv_p s$. Thus, the vertex set is $\{0, \dots, p-1\}$, and there is an edge between vertices u and v if $u - v$ is a square modulo p. I should now prove that $-s$ is a quadratic residue if and only if s is. This will hold provided that p is equivalent to 1 modulo 4. To prove that, I need to tell you one more thing about the integers modulo p: their multiplicative group is cyclic.

Fact 7.2.1. *For every prime p, there exists a number g such that for every number x between 1 and $p-1$, there is a unique i between 1 and $p-1$ such that*

$$x \equiv g^i \mod p.$$

In particular, $g^{p-1} \equiv 1$.

Corollary 7.2.2. *If p is a prime equivalent to 1 modulo 4, then -1 is a square modulo p.*

Proof. We know that 4 divides $p-1$. Let $s = g^{(p-1)/4}$. I claim that $s^2 = -1$. This will follow from $s^4 = 1$.

To see this, consider the equation

$$x^2 - 1 \equiv 0 \mod p.$$

As the numbers modulo p are a field, it can have at most 2 solutions. Moreover, we already know two solutions, $x = 1$ and $x = -1$. As $s^4 = 1$, we know that s^2 must be one of 1 or -1. However, it cannot be the case that $s^2 = 1$, because then the powers of g would begin repeating after the $(p-1)/2$ power, and thus could not represent every number modulo p. □

We now understand a lot about the squares modulo p (formally called *quadratic residues*). The squares are exactly the elements g^i where i is even. As $g^i g^j = g^{i+j}$, the fact that -1 is a square implies that s is a square if and only if $-s$ is a square. So, S is closed under negation, and the Cayley graph of $\mathbb{Z}/p$ with generator set S is in fact a graph. As $|S| = (p-1)/2$, it is regular of degree

$$d = \frac{p-1}{2}.$$

7.3 Eigenvalues of the Paley Graphs

It will prove simpler to compute the eigenvalues of the adjacency matrix of the Paley Graphs. Since these graphs are regular, this will immediately tell us the eigenvalues of the Laplacian. Let $\boldsymbol{L}$ be the Laplacians matrix of the Paley graph on p vertices. A remarkable feature of Paley graph is that $\boldsymbol{L}^2$ can be written as a linear combination of $\boldsymbol{L}$, $\boldsymbol{J}$ and $\boldsymbol{I}$, where $\boldsymbol{J}$ is the all-1's matrix. We will prove that

$$\boldsymbol{L}^2 = p\boldsymbol{L} + \frac{p-1}{4}\boldsymbol{J} - \frac{p(p-1)}{4}\boldsymbol{I}. \tag{7.1}$$

The proof will be easiest if we express $\boldsymbol{L}$ in terms of a matrix $\boldsymbol{X}$ defined by the *quadratic character*:

$$\chi(x) = \begin{cases} 1 & \text{if } x \text{ is a quadratic residue modulo } p \\ 0 & \text{if } x = 0, \text{ and} \\ -1 & \text{otherwise.} \end{cases}$$

This is called a character because it satisfies $\chi(xy) = \chi(x)\chi(y)$. We will use this to define a matrix $\boldsymbol{X}$ by

$$\boldsymbol{X}(u,v) = \chi(u-v).$$

An elementary calculation, which I skip, reveals that

$$\boldsymbol{X} = p\boldsymbol{I} - 2\boldsymbol{L} - \boldsymbol{J}. \tag{7.2}$$

Lemma 7.3.1.

$$\boldsymbol{X}^2 = p\boldsymbol{I} - \boldsymbol{J}.$$

When combined with (7.2), this lemma immediately implies (7.1).

Proof. The diagonal entries of $\boldsymbol{X}^2$ are the squares of the norms of the columns of $\boldsymbol{X}$. As each contains $(p-1)/2$ entries that are 1, $(p-1)/2$ entries that are -1, and one entry that is 0, its squared norm is $p-1$.

To handle the off-diagonal entries, we observe that $\boldsymbol{X}$ is symmetric, so the off-diagonal entries are the inner products of columns of $\boldsymbol{X}$. That is,

$$\boldsymbol{X}(u,v) = \sum_x \chi(u-x)\chi(v-x) = \sum_y \chi(y)\chi((v-u)+y),$$

where we have set $y = u - x$. For convenience, set $w = v - u$, so we can write this more simply. As we are considering a non-diagonal entry, $w \neq 0$. The term in the sum for $y = 0$ is zero. When $y \neq 0$, $\chi(y) \in \pm 1$, so

$$\chi(y)\chi(w+y) = \chi(w+y)/\chi(y) = \chi(w/y+1).$$

Now, as y varies over $\{1,\ldots,p-1\}$, w/y varies over all of $\{1,\ldots,p-1\}$. So, $w/y+1$ varies over all elements other than 1. This means that

$$\sum_y \chi(y)\chi((v-u)+y) = \left(\sum_{z=0}^{p-1}\chi(z)\right) - \chi(1) = 0 - 1 = -1.$$

So, every off-diagonal entry in $\boldsymbol{X}^2$ is -1. □

This gives us a quadratic equation that every eigenvalue other than d must obey. Let $\boldsymbol{\phi}$ be an eigenvector of $\boldsymbol{L}$ of eigenvalue $\lambda \neq 0$. As $\boldsymbol{\phi}$ is orthogonal to the all-1s vector, $\boldsymbol{J}\boldsymbol{\phi} = \boldsymbol{0}$. So,

$$\lambda^2 \boldsymbol{\phi} = \boldsymbol{L}^2 \boldsymbol{\phi} = p\boldsymbol{L}\boldsymbol{\phi} - \frac{p(p-1)}{4}\boldsymbol{I}\boldsymbol{\phi} == (p\lambda - p(p-1)/4)\boldsymbol{\phi}.$$

So, we find

$$\lambda^2 + p\lambda - \frac{p(p-1)}{4} = 0.$$

This gives

$$\lambda = \frac{1}{2}\left(p \pm \sqrt{p}\right).$$

This tells us at least two interesting things:

1. The Paley graph is (up to a very small order term) a $1 + \sqrt{1/p}$ approximation of the complete graph.
2. Payley graphs have only two nonzero eigenvalues. This places them within the special family of Strongly Regular Graphs, that we will study later in the semester.

7.4 Generalizing Hypercubes

To generalize the hypercube, we will consider Cayley graphs over the same group, but with more generators. Recall that we view the vertex set as the vectors in $\{0,1\}^d$, modulo 2. Each generator, $\boldsymbol{g}_1, \dots, \boldsymbol{g}_k$, is in the same group.

Let G be the Cayley graph with these generators. To be concrete, set $V = \{0,1\}^d$, and note that G has edge set

$$\left\{(\boldsymbol{x}, \boldsymbol{x} + \boldsymbol{g}_j) : \boldsymbol{x} \in V, 1 \leq j \leq k\right\}.$$

Using the analysis of products of graphs, we derived a set of eigenvectors of H_d. We will now verify that these are eigenvectors for all generalized hypercubes. Knowing these will make it easy to describe the eigenvalues.

For each $\boldsymbol{b} \in \{0,1\}^d$, define the function $\boldsymbol{\psi}_{\boldsymbol{b}}$ from V to the reals given by

$$\boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{x}) = (-1)^{\boldsymbol{b}^T \boldsymbol{x}}.$$

When we write $\boldsymbol{b}^T\boldsymbol{x}$, you might wonder if we mean to take the sum over the reals or modulo 2. As both $\boldsymbol{b}$ and $\boldsymbol{x}$ are $\{0,1\}$-vectors, you get the same answer either way you do it.

While it is natural to think of $\boldsymbol{b}$ as being a vertex, that is the wrong perspective. Instead, you should think of $\boldsymbol{b}$ as indexing a Fourier coefficient (if you don't know what a Fourier coefficient is, just don't think of it as a vertex).

The eigenvectors and eigenvalues of the graph are determined by the following theorem. As this graph is k-regular, the eigenvectors of the adjacency and Laplacian matrices will be the same.

Lemma 7.4.1. *For each $\boldsymbol{b} \in \{0,1\}^d$ the vector $\boldsymbol{\psi}_{\boldsymbol{b}}$ is a Laplacian matrix eigenvector with eigenvalue*

$$k - \sum_{i=1}^{k} (-1)^{\boldsymbol{b}^T \boldsymbol{g}_i}.$$

Proof. We begin by observing that

$$\boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{x}+\boldsymbol{y}) = (-1)^{\boldsymbol{b}^T(\boldsymbol{x}+\boldsymbol{y})} = (-1)^{\boldsymbol{b}^T\boldsymbol{x}}(-1)^{\boldsymbol{b}^T\boldsymbol{y}} = \boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{x})\boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{y}).$$

Let $\boldsymbol{L}$ be the Laplacian matrix of the graph. For any vector $\boldsymbol{\psi}_{\boldsymbol{b}}$ for $\boldsymbol{b} \in \{0,1\}^d$ and any vertex $\boldsymbol{x} \in V$, we compute

$$\begin{aligned}(\boldsymbol{L}\boldsymbol{\psi}_{\boldsymbol{b}})(\boldsymbol{x}) &= k\boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{x}) - \sum_{i=1}^{k} \boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{x}+\boldsymbol{g}_i)\\ &= k\boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{x}) - \sum_{i=1}^{k} \boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{x})\boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{g}_i)\\ &= \boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{x})\left(k - \sum_{i=1}^{k} \boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{g}_i)\right).\end{aligned}$$

So, $\boldsymbol{\psi}_{\boldsymbol{b}}$ is an eigenvector of eigenvalue

$$k - \sum_{i=1}^{k} \boldsymbol{\psi}_{\boldsymbol{b}}(\boldsymbol{g}_i) = k - \sum_{i=1}^{k} (-1)^{\boldsymbol{b}^T \boldsymbol{g}_i}.$$

□

7.5 A random set of generators

We will now show that if we choose the set of generators uniformly at random, for k some constant multiple of the dimension, then we obtain a graph that is a good approximation of the complete graph. That is, all the eigenvalues of the Laplacian will be close to k. This construction comes from the work of Alon and Roichman [AR94]. We will set $k = cd$, for some $c > 1$. Think of $c = 2$, $c = 10$, or $c = 1 + \epsilon$.

For $\boldsymbol{b} \in \{0,1\}^d$ but not all zero, and for $\boldsymbol{g}$ chosen uniformly at random from $\{0,1\}^d$, $\boldsymbol{b}^T\boldsymbol{g}$ modulo 2 is uniformly distributed in $\{0,1\}$, and so

$$(-1)^{\boldsymbol{b}^T\boldsymbol{g}}$$

is uniformly distributed in ±1. So, if we pick $\boldsymbol{g}_1, \dots, \boldsymbol{g}_k$ independently and uniformly from $\{0,1\}^d$, the eigenvalue corresponding to the eigenvector $\boldsymbol{\psi}_{\boldsymbol{b}}$ is

$$\lambda_{\boldsymbol{b}} \stackrel{\text{def}}{=} k - \sum_{i=1}^{k} (-1)^{\boldsymbol{b}^T \boldsymbol{g}_i}.$$

The right-hand part is a sum of independent, uniformly chosen ±1 random variables. So, we know it is concentrated around 0, and thus $\lambda_{\boldsymbol{b}}$ will be concentrated around k. To determine how concentrated the sum actually is, we use a Chernoff bound. There are many forms of Chernoff bounds. We will not use the strongest, but settle for one which is simple and which gives results that are qualitatively correct.

Theorem 7.5.1. *Let $x_1, \dots, x_k$ be independent ± 1 random variables. Then, for all $t > 0$,*

$$Pr\left[\left|\sum_i x_i\right| \geq t\right] \leq 2e^{-t^2/2k}.$$

This becomes very small when t is a constant fraction of k. In fact, it becomes so small that it is unlikely that any eigenvalue deviates from k by more than t.

Theorem 7.5.2. *With high probability, all of the nonzero eigenvalues of the generalized hypercube differ from k by at most*

$$k\sqrt{\frac{2}{c}},$$

where $k = cd$.

Proof. Let $t = k\sqrt{2/c}$. Then, for every nonzero $\boldsymbol{b}$,

$$\Pr\left[|k - \lambda_{\boldsymbol{b}}| \geq t\right] \leq 2e^{-t^2/2k} \leq 2e^{-k/c} = 2e^{-d}.$$

Now, the probability that there is some $\boldsymbol{b}$ for which $\lambda_{\boldsymbol{b}}$ violates these bounds is at most the sum of these terms:

$$\Pr\left[\exists \boldsymbol{b} : |k - \lambda_{\boldsymbol{b}}| \geq t\right] \leq \sum_{\boldsymbol{b} \in \{0,1\}^d, \boldsymbol{b} \neq 0^d} \Pr\left[|k - \lambda_{\boldsymbol{b}}| \geq t\right] \leq (2^d - 1)2e^{-d},$$

which is always less than 1 and goes to zero exponentially quickly as d grows. □

We initially suggested thinking of $c = 2$ or $c = 10$. The above bound works for $c = 10$. To get a useful bound for $c = 2$, we need to sharpen the analysis. A naive sharpening will work down to $c = 2\ln 2$. To go lower than that, you need a stronger Chernoff bound.

7.6 Conclusion

We have now seen that a random generalized hypercube of degree k probably has all non-zero Laplacian eigenvalues between

$$k(1 - \sqrt{2/c}) \quad \text{and} \quad k(1 + \sqrt{2/c}).$$

If we let n be the number of vertices, and we now multiply the weight of every edge by n/k, we obtain a graph with all nonzero Laplacian eigenvalues between

$$n(1 - \sqrt{2/c}) \quad \text{and} \quad n(1 + \sqrt{2/c}).$$

Thus, this is essentially a $1 + \sqrt{2/c}$ approximation of the complete graph on n vertices. But, the degree of every vertex is only $c \log_2 n$. Expanders are infinite families of graphs that are constant-factor approximations of complete graphs, but with constant degrees.

We know that random regular graphs are probably expanders. If we want explicit constructions, we need to go to non-Abelian groups.

Explicit constructions that achieve bounds approaching those of random generalized hypercubes come from error-correcting codes.

Explicit constructions allow us to use these graphs in applications that require us to implicitly deal with a very large graph. In Chapter 31, we will see how to use such graphs to construct pseudo-random generators.

7.7 Non-Abelian Groups

In the homework, you will show that it is impossible to make constant-degree expander graphs from Cayley graphs of Abelian groups. The best expanders are constructed from Cayley graphs of 2-by-2 matrix groups. In particular, the Ramanujan expanders of Margulis [Mar88] and Lubotzky, Phillips and Sarnak [LPS88] are Cayley graphs over the Projective Special Linear Groups $\mathrm{PSL}(2, p)$, where p is a prime. These are the 2-by-2 matrices modulo p with determinant 1, in which we identify A with $-A$.

They provided a very concrete set of generators. For a prime q modulo to 1 modulo 4, it is known that there are $p + 1$ solutions to the equation

$$a_1^2 + a_2^2 + a_3^2 + a_4^2 = p,$$

where a_1 is odd and a_2, a_3 and a_4 are even. We obtain a generator for each such solution of the form:

$$\frac{1}{\sqrt{p}} \begin{bmatrix} a_0 + ia_1 & a_2 + ia_3 \\ -a_2 + ia_3 & a_0 - ia_1 \end{bmatrix},$$

where i is an integer that satisfies $i^2 = -1$ modulo p.

Even more explicit constructions, which do not require solving equations, may be found in [ABN+92].

7.8 Eigenvectors of Cayley Graphs of Abelian Groups

The wonderful thing about Cayley graphs of Abelian groups is that we can construct an orthornormal basis of eigenvectors for these graphs without even knowing the set of generators S. That is, the eigenvectors only depend upon the group. Related results also hold for Cayley graphs of arbitrary groups, and are related to representations of the groups. See [Bab79] for details.

As Cayley graphs are regular, it won't matter which matrix we consider. For simplicity, we will consider adjacency matrices.

Let n be an integer and let G be a Cayley graph on $\mathbf{Z}/n$ with generator set S. When $S = \{\pm 1\}$, we get the ring graphs. For general S, I think of these as generalized Ring graphs. Let's first see that they have the same eigenvectors as the Ring graphs.

Recall that we proved that the vectors $\boldsymbol{x}_k$ and $\boldsymbol{y}_k$ were eigenvectors of the ring graphs, where

$$\begin{aligned}\boldsymbol{x}_k(u) &= \sin(2\pi k u/n), \text{ and}\\ \boldsymbol{y}_k(u) &= \cos(2\pi k u/n),\end{aligned}$$

for $1 \leq k \leq n/2$.

Let's just do the computation for the $\boldsymbol{x}_k$, as the $\boldsymbol{y}_k$ are similar. For every u modulo n, we have

$$\begin{aligned}(A\boldsymbol{x}_k)(u) &= \sum_{g \in S} \boldsymbol{x}_k(u+g)\\ &= \frac{1}{2}\left(\sum_{g \in S} \boldsymbol{x}_k(u+g) + \boldsymbol{x}_k(u-g)\right)\\ &= \frac{1}{2}\left(\sum_{g \in S} \sin(2\pi k(u+g)/n) + \sin(2\pi k(u-g)/n)\right)\\ &= \frac{1}{2}\left(\sum_{g \in S} 2\sin(2\pi k u/n)\cos(2\pi k g/n)\right)\\ &= \sin(2\pi k u/n)\sum_{g \in S} \cos(2\pi k g/n)\\ &= \boldsymbol{x}_k(u)\sum_{g \in S} \cos(2\pi k g/n).\end{aligned}$$

So, the corresponding eigenvalue is

$$\sum_{g \in S} \cos(2\pi k g/n).$$

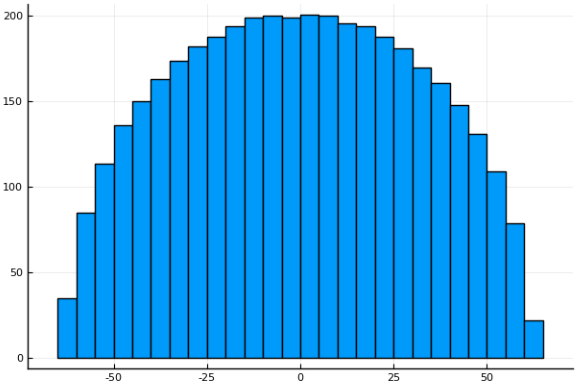
Chapter 8

Eigenvalues of Random Graphs

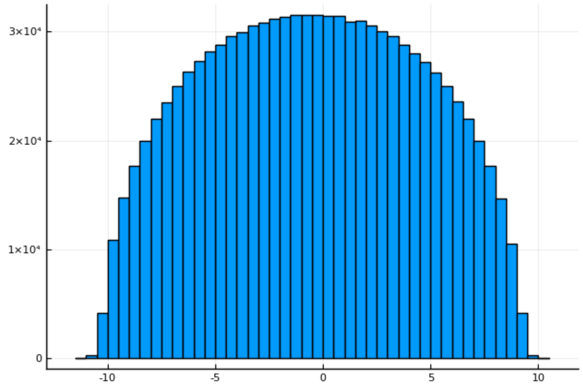
Notation: at many points during this chapter, we will write $[n]$ to indicate the set $\{1, 2, \ldots, n\}$.

In this chapter we examine the adjacency matrix eigenvalues of Erdös-Rényi random graphs. These are graphs in which each edge is chosen to appear with probability p, and the choices are made independently for each edge. We will find that these graphs typically have one large eigenvalue around pn, and that all of the others probably have absolute value at most $(1 + o(1))2\sqrt{p(1-p)n}$. In fact, their distribution within this region follows Wigner's [Wig58] semicircle law: their histogram looks like a semicircle.

For example, let's consider $p = 1/2$. Here is the histogram of all but the largest eigenvalue of a random graph on 4,000 vertices.

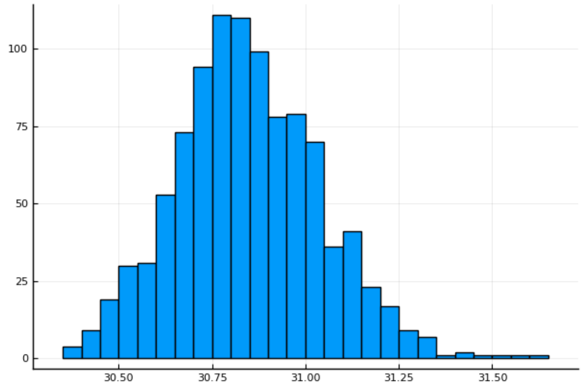


The following image is the histogram of the 99 smallest adjacency eigenvalues of 10,000 random graphs on 100 vertices.



Note that the eigenvalues are almost never outside $-10, 10$. We are going to prove something like that today.

Here is a histogram of the second-largest eigenvalues of 10,000 matrices on 1,000 vertices.



8.1 Transformation and Moments

Let $\boldsymbol{M}$ be the adjacency matrix of an Erdös-Rényi random graph. We understand $\boldsymbol{M}$ by writing it in the form

$$\boldsymbol{M} = p(\boldsymbol{J} - \boldsymbol{I}) + \boldsymbol{R},$$

where $\boldsymbol{J}$ is the all-1s matrix, and $\boldsymbol{R}$ is a random symmetric whose diagonal entries are zero and whose off-diagonal entries have distribution

$$\boldsymbol{R}(a, b) = \begin{cases} 1 - p & \text{with probability } p \text{, and} \\ -p & \text{with probability } 1 - p. \end{cases}$$

The reason we write it this way is that the expectation of every entry of $\boldsymbol{R}$, and thus $\boldsymbol{R}$ is zero. We will show that with high probability all of the eigenvalues of $\boldsymbol{R}$ are probably small, and thus we can view $\boldsymbol{M}$ as being approximately $p(\boldsymbol{J} - \boldsymbol{I})$.

As $p(\boldsymbol{J} - \boldsymbol{I})$ has one eigenvalue of pn and $n-1$ eigenvalues of $-p$, the bulk of the distribution of eigenvalues of $\boldsymbol{M}$ is very close to the distribution of eigenvalues of $\boldsymbol{R}$, minus p. To see this, first subtract $p\boldsymbol{I}$ from $\boldsymbol{R}$. This shifts all the eigenvalues down by p. We must now add $p\boldsymbol{J}$. As $\boldsymbol{J}$ is a rank-1 matrix, we can show that the eigenvalues of $p\boldsymbol{J} + (\boldsymbol{R} - p\boldsymbol{I})$ interlace the eigenvalues of $\boldsymbol{R} - p\boldsymbol{I}$ (see the exercise at the end of the chapter). So, the largest eigenvalue moves up a lot, and all the other $n-1$ move up to at most the next eigenvalue.

A good way to characterize the general shape of a distribution is through its moments. Let $\rho_1, \dots, \rho_n$ be the eigenvalues of $\boldsymbol{R}$. Their first moment is simply their sum, which is the trace of $\boldsymbol{R}$ and thus zero. Their kth moment is

$$\sum_{i=1}^{n} \rho_i^k = \operatorname{Tr}\left(\boldsymbol{R}^k\right).$$

One of the easiest ways to reason about the distribution of the eigenvalues is to estimate the expectations of the traces of powers of $\boldsymbol{R}$. This is called Wigner's trace method.

Before proceeding with our analysis, we recall a formula for the entries of a power of matrices. For matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ whose rows and columns are indexed by V,

$$(\boldsymbol{A}\boldsymbol{B})(a,b) = \sum_{c \in V} \boldsymbol{A}(a,c)\boldsymbol{B}(c,b).$$

Applying this formula inductively, we find that

$$(\boldsymbol{A}^l)(a,b) = \sum_{c_1,\dots,c_{l-1} \in V} \boldsymbol{A}(a,c_1)\boldsymbol{A}(c_1,c_2)\cdots\boldsymbol{A}(c_{l-1},b).$$

8.2 The extreme eigenvalues

Recall that the *norm* of a matrix $\boldsymbol{R}$, written $\|\boldsymbol{R}\|$, is the maximum of the absolute value of its eigenvalues. It is called a norm because for all $\boldsymbol{x}$,

$$\|\boldsymbol{R}\boldsymbol{x}\| \leq \|\boldsymbol{R}\| \, \|\boldsymbol{x}\|.$$

We will focus on proving an upper bound on the norm of $\boldsymbol{R}$ that holds with high probability. We will do this by estimating the trace of a high power of $\boldsymbol{R}$. For an even power l, this should be relatively close to $\|\boldsymbol{R}\|$. In particular, we use the fact that for every even l

$$\|\boldsymbol{R}\|^l \leq \operatorname{Tr}\left(\boldsymbol{R}^l\right)$$

and thus

$$\|\boldsymbol{R}\| \leq \left(\operatorname{Tr}\left(\boldsymbol{R}^l\right)\right)^{1/l}.$$

We will prove in Theorem 8.3.2 that for every even l such that $np(1-p) \geq 2l^8$,

$$\mathbb{E}\operatorname{Tr}\left(\boldsymbol{R}^l\right) \leq 2n(4np(1-p))^{l/2}.$$

This will allow us to show that the norm of $\boldsymbol{R}$ is usually less than u where

$$u \overset{\text{def}}{=} \left(2n(4np(1-p))^{l/2}\right)^{1/l} = (2n)^{1/l} 2\sqrt{np(1-p)}.$$

We establish this by an application of Markov's inequality. For all $\epsilon > 0$,

$$\begin{aligned}\Pr\left[\|\boldsymbol{R}\| > (1+\epsilon)u\right] &\leq \Pr\left[\operatorname{Tr}\left(\boldsymbol{R}^l\right) > (1+\epsilon)^l u^l\right] \\ &\leq \Pr\left[\operatorname{Tr}\left(\boldsymbol{R}^l\right) > (1+\epsilon)^l \mathbb{E}\operatorname{Tr}\left(\boldsymbol{R}^l\right)\right] \\ &\leq (1+\epsilon)^{-l},\end{aligned}$$

by Markov's inequality.

To understand this probability, remember that for small ϵ $(1+\epsilon)$ is approximately $\exp(\epsilon)$. So, $(1+\epsilon)^{-l}$ is approximately $\exp(-\epsilon l)$. This probability becomes small when $l > 1/\epsilon$. Concretely, for $\epsilon < 1/2$, $1+\epsilon < \exp(4\epsilon/5)$. Thus, we can take ϵ approximately $(n/2)^{-1/8}$. While this bound is not very useful for n that we encounter in practice, it is nice asymptotically. The bound can be substantially improved by more careful arguments, as we explain at the end of the Chapter.

We should also examine the term $(2n)^{1/l}$:

$$(2n)^{1/l} = \exp(\ln(2n)/l) \leq 1 + 1.1\ln(2n)/l,$$

for $\ln(2n)/l < 1/2$. Thus, for $l\boldsymbol{g}\ln(2n)$ this term is close to 1.

8.3 Expectation of the trace of a power

Recall that the trace is the sum of the diagonal entries in a matrix. By expanding the formula for matrix multiplication, one can also show

$$\boldsymbol{R}^l(a_0, a_0) = \sum_{a_1,\dots,a_{l-1}\in V} \boldsymbol{R}(a_0, a_{l-1}) \prod_{i=1}^{l-1} \boldsymbol{R}(a_{i-1}, a_i),$$

and so

$$\mathbb{E}\boldsymbol{R}^l(a_0, a_0) = \sum_{a_1,\dots,a_{l-1}\in V} \mathbb{E}\boldsymbol{R}(a_0, a_{l-1}) \prod_{i=1}^{l-1} \boldsymbol{R}(a_{i-1}, a_i).$$

To simplify this expression, we will recall that if X and Y are independent random variables, then $E(XY) = E(X)E(Y)$. So, to the extent that the terms in this product are independent, we can distribute this expectation across this product. As the entries of $\boldsymbol{R}$ are independent, up to the symmetry condition, the only terms that are dependent are those that are identical. So, if $\{b_j, c_j\}_j$ is the set of pairs that occur in

$$\{a_0, a_1\}, \{a_1, a_2\}, \dots, \{a_{l-2}, a_{l-1}\}, \{a_{l-1}, a_0\}, \tag{8.1}$$

and pair $\{b_j, c_j\}$ appears d_j times, then

$$\mathbb{E}\boldsymbol{R}(a_0, a_{l-1}) \prod_{i=1}^{l-1} \boldsymbol{R}(a_{i-1}, a_i) = \prod_j \mathbb{E}\boldsymbol{R}(b_j, c_j)^{d_j}.$$

As each entry of $\boldsymbol{R}$ has expectation 0,

$$\mathbb{E}\boldsymbol{R}(b_j, c_j)^{d_j}$$

is zero if d_j is 1. In general

$$\mathbb{E}\boldsymbol{R}^d_{(b_j,c_j)} = p(1-p)\left[(1-p)^{d-1} - (-p)^{d-1}\right] \leq p(1-p), \tag{8.2}$$

for $d \geq 2$.

So, $\mathbb{E}\boldsymbol{R}^l(a_0, a_0)$ is at most the sum over sequences $a_1, \dots, a_{l-1}$ such that each pair in (8.1) appears at least twice, times $p(1-p)$ for each pair that appears in the sequence.

To describe this more carefully, we say that a sequence $a_0, a_1, \dots, a_l$ is a *closed walk of length l on n vertices* if each $a_i \in \{1, \dots, n\}$ and $a_l = a_0$. In addition, we say that it is *significant* if for every tuple $\{b, c\}$ there are at least two indices i for which $\{a_i, a_{i+1}\} = \{b, c\}$. Let $W_{n,l,k}$ denote the number of significant closed walks of length l on n vertices such that $a_1, \dots, a_{l-1}$ contains exactly k distinct elements. As a sequence with k distinct elements must contain at least k distinct pairs, we obtain the following upper bound on the trace.

Lemma 8.3.1.

$$\mathbb{E}\mathrm{Tr}\left(\boldsymbol{R}^l\right) \leq \sum_{k=1}^{l/2} W_{n,l,k}(p(1-p))^k.$$

In the next section, we prove that

$$W_{n,l,k} \leq n^{k+1} 2^l l^{4(l-2k)}.$$

Theorem 8.3.2. *If l is even and $np(1-p) \geq 2l^8$, then*

$$\mathbb{E}\mathrm{Tr}\left(\boldsymbol{R}^l\right) \leq 2n(4np(1-p))^{l/2}.$$

Proof. Let

$$t_k = n^{k+1} 2^l l^{4(l-2k)} (p(1-p))^k.$$

We will show that the sequence t_k is geometrically increasing, and thus it is dominated by its largest term. We compute

$$\begin{aligned} \frac{t_k}{t_{k-1}} &= \frac{n^{k+1} 2^l l^{4(l-2k)} (p(1-p))^k}{n^k 2^l l^{4(l-2k+2)} (p(1-p))^{k-1}} \\ &= \frac{n(p(1-p))}{l^8} \\ &\geq 2. \end{aligned}$$

Thus,

$$\mathbb{E}\mathrm{Tr}\left(\boldsymbol{R}^l\right) \leq \sum_{k=1}^{l/2} W(n,l,k)(p(1-p))^k \leq \sum_{k=1}^{l/2} t_k \leq t_{l/2} \sum_{k=1}^{l/2} 2^{1-k} \leq 2t_{l/2} \leq 2n2^l(np(1-p))^{l/2}.$$

□

Of course, better bounds on $W_{n,l,k}$ provide better bounds on the trace. Vu [Vu07] proves that

$$W_{n,l,k} \leq n^{k+1}\binom{l}{2k}(k+1)^{2(l-2k)}2^{2k}.$$

This bound allows us to apply much higher powers of the matrix.

8.4 The number of walks

Our goal is to prove an upper bound on $W_{n,l,k}$. We will begin by proving a crude upper bound, and then refine it.

As it is tricky to obtain a clean formula for the number of such walks, we will instead derive ways of describing such walks, and then count how many such descriptions there can be.

Let $S \subset \{1,\dots,l-1\}$ be the set of i for which a_i does not appear earlier in the walk:

$$a_i \not\in \{a_j : j < i\}.$$

There are at most $\binom{l-1}{k}$ choices for the set S. Given S, we wish to record the identities of the elements a_i for $i \in S$. Each is an element of $[n]$, and we record them in the order in which they appear in the walk. You may wish to think of this data as a map

$$\sigma : S \rightarrow [n].$$

There are at most n^k such maps. We also record the identity of a_0.

For each $i \not\in S$ there is a $j \in S \cup \{0\}$ for which $a_j = a_i$. We record which element of S this is, or we record 0 if it is a_0. As S has k elements, we only need one of $k+1$ numbers to record this j. Again, you may wish to think of this data as a map

$$\tau : [l-1] \setminus S \rightarrow S \cup \{0\}.$$

There are at most $(k+1)^{l-1-k}$ choices for τ. See figure 8.1 for an example.

While not every choice of a_0, S, σ and τ corresponds to a significant walk, every significant walk with k distinct elements corresponds to some a_0, S, σ and τ. Thus,

$$W_{n,l,k} \leq \binom{l-1}{k} n^{k+1}(k+1)^{l-1-k}. \tag{8.3}$$

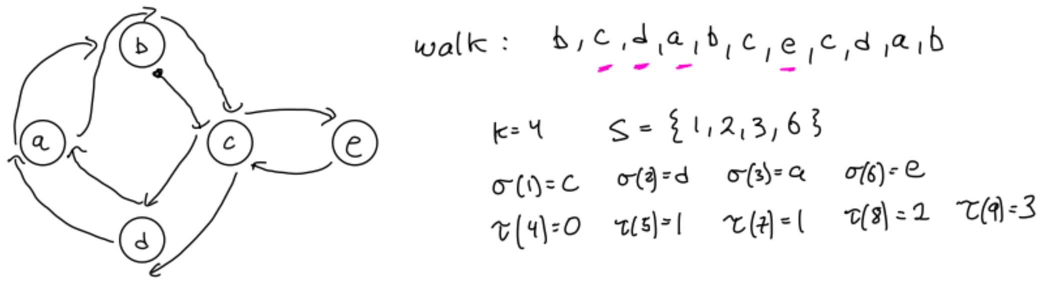


Figure 8.1: An example walk along with S, σ, and τ.

This bound is too loose to obtain the result we desire. It merely allows us to prove that $\|\boldsymbol{R}\| \leq c\sqrt{np(1-p)}\log n$ for some constant c. This bound is loosest when $k = l/2$. This is fortunate both because this is the case in which it is easiest to tighten the bound, and because the computation in Theorem 8.3.2 is dominated by this term.

Consider the graph with edges (a_{i-1}, a_i) for $i \in S$. This graph must be a tree because it contains exactly the edges from which the walk first hits each vertex. Formally, this is because the graph contains k edges, touches $k+1$ vertices, and we can prove by induction on the elements in S that it is connected, starting with a_0. See Figure 8.2.

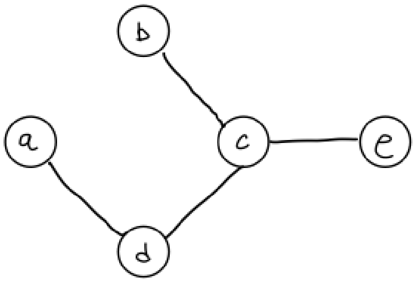


Figure 8.2: The tree edges for this walk.

We can use this tree to show that, when $k = l/2$, every pair $\{a_{i-1}, a_i\}$ that appears in the walk must appear exactly twice: the walk only takes l steps, and each pair of the $k = l/2$ in the tree must be covered at least twice.

We now argue that when $l = 2k$ the map τ is completely unnecessary: the walk is determined by S and σ alone. That is, for every $i \notin S$ there is only one edge that the walk can follow. For $i \notin S$, the tuple $\{a_{i-1}, a_i\}$ must have appeared exactly once before in the walk. We will show that at step i the vertex a_{i-1} is adjacent to exactly one edge with this property.

To this end, we keep track of the graph of edges that have appeared exactly once before in the walk. We could show by induction that at step i this graph is precisely a path from a_0 to a_{i-1}. But, we take an alternate approach. Consider the subgraph of the tree edges that have been used exactly once up to step i. We will count both its number of vertices, v, and its number of edges,

f. At step i we include a_i in this subgraph regardless of whether it is adjacent to any of the edges, so initially $v = 1$ and $e = 0$. The walk ends in the same state.

For steps in which $i \in S$, both v and f increase by 1: the edge $\{a_{i-1}, a_i\}$ and the vertex a_i is added to the subgraph. When $i \notin S$ and $\{a_{i-1}, a_i\}$ is the only tree edge adjacent to a_{i-1} that has been used exactly once, both e and v decrease: e because we use the pair $\{a_{i-1}, a_i\}$ a second time and v because a_{i-1} is no longer adjacent to any tree edge that has been used exactly once, and the walk moves to a_i.

If for some $i \notin S$ it were the case that a_{i-1} was adjacent to two tree edges that had been used exactly once, then e would decrease but v would not. As the process starts and ends with $v - e = 1$, this is not possible.

Thus

$$W_{n,l,l/2} \leq n^{l/2+1}\binom{l}{l/2}.$$

Now that we know $W_{n,l,l/2}$ is much less than the bound suggested by (8.3), we should suspect that $W_{n,l,k}$ is also much lower when k is close to $l/2$. To show this, we extend the idea used in the previous argument to show that with very little information we can determine where the walk goes for almost every step not in S.

We say that the ith step in the walk is *extra* if the pair $\{a_{i-1}, a_i\}$ is not a tree edge or if it appears at least twice in the walk before step i. Let x denote the number of extra steps. As each of the tree edges appears in at least two steps, the number of extra steps is at most $l - 2k$. We will use τ to record the destination vertex a_i of each extra step, again by indicating its position in S.

During the walk, we keep track of the set of tree edges that have been used exactly once. Let T be the set of steps in which in which a_{i-1} is adjacent to exactly one tree edge that has been used exactly once and the walk follows that edge. That is, the edge is $\{a_{i-1}, a_i\}$ and we can infer a_i given $i \in T$. If a_{i-1} is adjacent to exactly one tree edge that has been used exactly once but the walk does not follow that edge, then step i is extra: it either follows an edge that is not in the tree or it follows a tree edge that has been used at least twice.

This leaves us to account for the steps in which a_{i-1} is adjacent to more than one tree edge that has been used exactly once and follows such an edge. In this case, we call step i *ambiguous*, because we need some way to indicate which of those edges it used. In ambiguous steps we also use τ to record a_i. Every step not in S or T is extra or ambiguous. So,

$$\tau : ([l-1] \setminus (S \cup T)) \to S \cup \{0\}.$$

The data a_0, S, T, σ, and τ determine the walk. It remains to determine how many ways we can choose them.

We will show that the number of ambiguous steps is at most the number of extra steps. This implies that $|V \setminus (S \cup T)| \leq 2x$. Thus, the number of possible maps τ is at most

$$(k+1)^{2x}.$$

The number of choices for S and T may be upper bounded by

$$\binom{l-1}{k}\binom{l-1-k}{2x} \leq 2^{l-1}(l-1-k)^{2x}.$$

Thus

$$W_{n,l,k} \leq n^{k+1}2^{l-1}(l-k-1)^{2x}(k+1)^{2x} \leq n^{k+1}2^{l-1}(lk)^{2(l-2k)} \leq n^{k+1}2^{l}l^{4(l-2k)}. \tag{8.4}$$

We now finish by arguing that the number of ambiguous edges is at most the number of extra edges. As before, keep track of the subgraph of the tree edges that have been used exactly once up to step i. We will count both its number of vertices, v, and its number of edges, e. At step i we include a_i in this subgraph regardless of whether it is adjacent to any of the graphs edges, so initially $v = 1$ and $e = 0$. The walk ends in the same state.

For steps in which $i \in S$, both v and e increase by 1. For steps in which $i \in T$, the vertex a_{i-1} has degree one in this graph. When we follow the edge (a_{i-1}, a_i), we remove it from this graph. As a_{i-1} is no longer adjacent to any edge of the graph, both v and e decrease by 1.

At ambiguous steps i, we decrement e. But, because a_{i-1} was adjacent to at least two tree edges that had been used exactly once, it is not removed from the graph and v does not decrease. The ambiguous steps may be compensated by extra steps. An extra step does not change f, but it can decrease v. This happens when a_{i-1} is not adjacent to any tree edges that have been used exactly once, but a_i is. Thus, a_{i-1} contributes 1 to v during step $i-1$, but is removed from the count as soon as the walk moves to a_i. As the walk starts and ends with $v - f = 1$, neither the steps in S nor T change this difference, ambiguous steps increase it, and extra steps can decrease it, the number of extra steps must be at least the number of ambiguous steps.

8.5 Notes

The proof in this chapter is a slight simplification and weakening of result due to Vu [Vu07]. The result was first claimed by Füredi and Komlos [FK81]. However, there were a few mistakes in their paper. Vu's paper also provides concentration results that lower bound μ_2, whereas the argument in this chapter merely provides an upper bound.

8.6 Exercise

1. Interlacing.

Let $\boldsymbol{A}$ be a symmetric matrix with eigenvalues $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$. Let $\boldsymbol{B} = \boldsymbol{A} + \boldsymbol{x}\boldsymbol{x}^T$ for some vector $\boldsymbol{x}$ and let $\beta_1 \geq \beta_2 \geq \ldots \geq \beta_n$ be the eigenvalues of $\boldsymbol{B}$. Prove that for all i

$$\beta_i \geq \alpha_i \geq \beta_{i+1}.$$

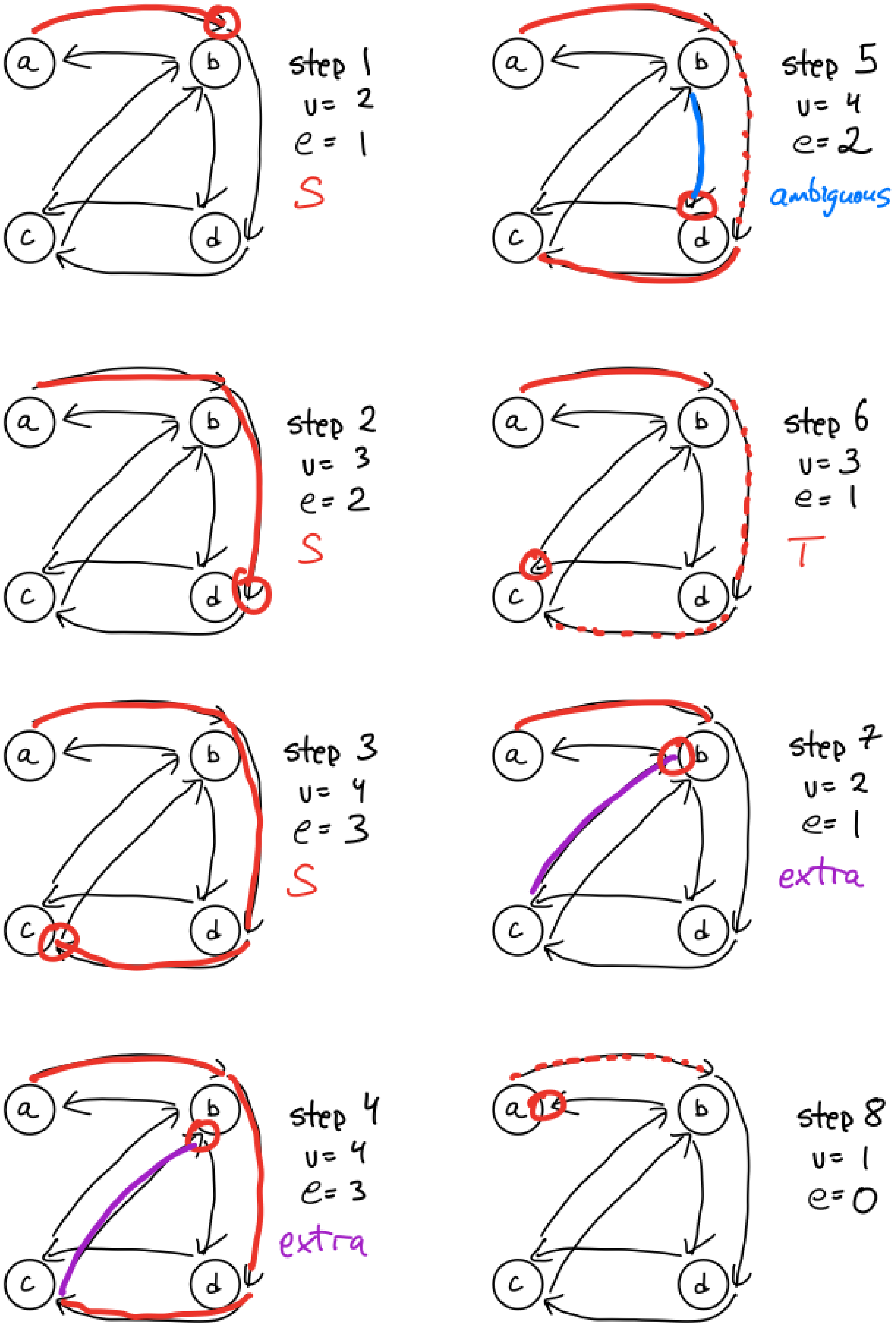


Figure 8.3: An example walk along with v and e

Chapter 9

Strongly Regular Graphs

Lecture 23 from November 18, 2009

9.1 Introduction

In this and the next lecture, I will discuss strongly regular graphs. Strongly regular graphs are extremal in many ways. For example, their adjacency matrices have only three distinct eigenvalues. If you are going to understand spectral graph theory, you must have these in mind.

In many ways, strongly-regular graphs can be thought of as the high-degree analogs of expander graphs. However, they are much easier to construct.

The Paley graphs we encountered in Chapter **??** are Strongly Regular Graphs.

Many times someone has asked me for a matrix of 0s and 1s that "looked random", and strongly regular graphs provided a resonable answer.

Warning: I will use the letters that are standard when discussing strongly regular graphs. So λ and μ will not be eigenvalues in this lecture.

9.2 Definitions

Formally, a graph G is strongly regular if

1. it is k-regular, for some integer k;
2. there exists an integer λ such that for every pair of vertices x and y that are neighbors in G, there are λ vertices z that are neighbors of *both* x and y;
3. there exists an integer μ such that for every pair of vertices x and y that are *not* neighbors in G, there are μ vertices z that are neighbors of *both* x and y.

These conditions are very strong, and it might not be obvious that there are any non-trivial graphs that satisfy these conditions. Of course, the complete graph and disjoint unions of complete graphs satisfy these conditions.

For the rest of this lecture, we will only consider strongly regular graphs that are connected and that are not the complete graph. I will now give you some examples.

9.3 The Pentagon

The simplest strongly-regular graph is the pentagon. It has parameters

$$n = 5, \quad k = 2, \quad \lambda = 0, \quad \mu = 1.$$

9.4 Lattice Graphs

For a positive integer n, the *lattice graph* L_n is the graph with vertex set $\{1, \dots n\}^2$ in which vertex (a, b) is connected to vertex (c, d) if $a = c$ or $b = d$. Thus, the vertices may be arranged at the points in an n-by-n grid, with vertices being connected if they lie in the same row or column. Alternatively, you can understand this graph as the line graph of a bipartite complete graph between two sets of n vertices.

It is routine to see that the parameters of this graph are:

$$k = 2(n - 1), \quad \lambda = n - 2, \quad \mu = 2.$$

9.5 Latin Square Graphs

A Latin Square is an n-by-n grid, each entry of which is a number between 1 and n, such that no number appears twice in any row or column. For example,

$$\begin{bmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{bmatrix}$$

Let me remark that the number of different latin squares of size n grows very quickly, at least as fast as $n!(n-1)!(n-2)! \dots 2!$.

From such a latin square, we construct a Latin Square Graph. It will have n^2 nodes, one for each cell in the square. Two nodes are joined by an edge if

1. they are in the same row,
2. they are in the same column, or

3. they hold the same number.

So, such a graph has degree $k = 3(n-1)$. Any two nodes in the same row will both be neighbors with every other pair of nodes in their row. They will have two more common neighors: the nodes in their columns holding the other's number. So, they have n common neighbors. The same obviously holds for columns, and is easy to see for nodes that have the same number. So, every pair of nodes that are neighbors have exactly $\lambda = n$ common neighbors.

On the other hand, consider two vertices that are not neighbors, say $(1,1)$ and $(2,2)$. They lie in different rows, lie in different columns, and hold different numbers. The vertex $(1,1)$ has two common neighbors of $(2,2)$ in its row: the vertex $(1,2)$ and the vertex holding the same number as $(2,2)$. Similarly, it has two common neighbors of $(2,2)$ in its column. Finally, we can find two more common neighbors of $(2,2)$ that are in different rows and columns by looking at the nodes that hold the same number as $(1,1)$, but which are in the same row or column as $(2,2)$. So, $\mu = 6$.

9.6 The Eigenvalues of Strongly Regular Graphs

We will consider the adjacency matrices of strongly regular graphs. Let A be the adjacency matrix of a strongly regular graph with parameters (k, λ, μ). We already know that A has an eigenvalue of k with multiplicity 1. We will now show that A has just two other eigenvalues.

To prove this, first observe that the (u,v) entry of A^2 is the number of common neighbors of vertices u and v. For $u = v$, this is just the degree of vertex u. We will use this fact to write A^2 as a linear combination of A, I and J. To this end, observe that the adjacency matrix of the complement of A (the graph with non-edges where A has edges) is $J - I - A$. So,

$$A^2 = \lambda A + \mu(J - I - A) + kI = (\lambda - \mu)A + \mu J + (k - \mu)I.$$

For every vector $\boldsymbol{v}$ orthogonal to $\mathbf{1}$,

$$A^2 \boldsymbol{v} = (\lambda - \mu)A\boldsymbol{v} + (k - \mu)\boldsymbol{v}.$$

So, every eigenvalue θ of A other than k satisfies

$$\theta^2 = (\lambda - \mu)\theta + k - \mu.$$

The eigenvalues of A other than k are those θ that satisfy this quadratic equation, and so are given by

$$\frac{\lambda - \mu \pm \sqrt{(\lambda - \mu)^2 + 4(k - \mu)}}{2}.$$

These eigenvalues are always denoted r and s, with $r > s$. By convention, the multiplicty of the eigenvalue r is always denoted f, and the multiplicty of s is always denoted g.

For example, for the pentagon we have

$$r = \frac{\sqrt{5} - 1}{2}, \quad s = -\frac{\sqrt{5} + 1}{2}.$$

For the lattice graph L_n, we have

$$r = n - 2, \quad s = -2.$$

For the latin square graphs of order n, we have

$$r = n - 3, \quad s = -3.$$

9.7 Regular graphs with three eigenvalues

We will now show that every regular connected graph with at most 3 eigenvalues must be a strongly regular graph. Let G be k-regular, and let its eigenvalues other than k be r and s. As G is connected, its adjacency eigenvalue k has multiplicty 1.

Then, for every vector orthogonal to $\mathbf{1}$, we have

$$(A - rI)(A - sI)\boldsymbol{v} = 0.$$

Thus, for some β,

$$(A - rI)(A - sI) = \beta J,$$

which gives

$$\begin{aligned} A^2 - (r+s)A + rsI = \beta J \qquad &\Longrightarrow \\ A^2 &= (r+s)A - rsI + \beta J \\ &= (r+s+\beta)A + \beta(J - A - I) + (rs + \beta)I. \end{aligned}$$

So, the number of common neighbors of two nodes just depends on whether or not they are neighbors, which implies that A is strongly regular.

9.8 Integrality of the eigenvalues

We will now see that, unless $f = g$, both r and s must be integers. We do this by observing a few identities that they both must satisfy. First, from the quadratic equation above, we know that

$$r + s = \lambda - \mu \tag{9.1}$$

and

$$rs = \mu - k. \tag{9.2}$$

As the trace of an adjacency matrix is zero, and is also the sum of the eigenvalues times their multiplicites, we know

$$k + fr + gs = 0. \tag{9.3}$$

So, it must be the case that $s < 0$. Equation 9.1 then gives $r > 0$.

If $f \neq g$, then equations (9.3) and (9.1) provide independent constraints on r and s, and so together they determine r and s. As the coefficients in both equations are integers, they tell us that both r and s are rational numbers. From this, and the fact that r and s are the roots of a quadratic equation with integral coefficients, we may conclude that r and s are in fact integers. Let me remind you as to why.

Lemma 9.8.1. *If θ is a rational number that satisfies*

$$\theta^2 + b\theta + c = 0,$$

where b and c are integers, then θ must be an integer.

Proof. Write $\theta = x/y$, where the greatest common divisor of x and y is 1. We then have

$$(x/y)^2 + b(x/y) + c = 0,$$

which implies

$$x^2 + bxy + cy^2 = 0,$$

which implies that y divides x^2. As we have assumed the greatest common divisor of x and y is 1, this implies $y = 1$. □

9.9 The Eigenspaces of Strongly Regular Graphs

It is natural to ask what the eigenspaces can tell us about a strongly regular graph. But, we will find that they don't tell us anything we don't already know.

Let $\boldsymbol{u}_1, \dots \boldsymbol{u}_f$ be an orthonormal set of eigenvectors of the eigenvalue r, and let U be the matrix containing these vectors as columns. Recall that U is only determined up to an orthnormal transformation. That is, we could equally take UQ for any f-by-f orthnormal matrix Q.

To the ith vertex, we associate the vector

$$\boldsymbol{x}_i \overset{\text{def}}{=} (\boldsymbol{u}_1(i), \dots, \boldsymbol{u}_f(i)).$$

While the vectors U are determined only up to orthogonal transformations, these transformations don't effect the geometry of these vectors. For example, for vertices i and j, the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ is

$$\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|,$$

and

$$\left\| \boldsymbol{x}_i - \boldsymbol{x}_j \right\|^2 = \left\| \boldsymbol{x}_i \right\|^2 + \left\| xx_j \right\|^2 - 2\boldsymbol{x}_i \boldsymbol{x}_j^T.$$

On the other hand,

$$\left\| \boldsymbol{x}_i Q - \boldsymbol{x}_j Q \right\|^2 = \left\| \boldsymbol{x}_i Q \right\|^2 + \left\| xx_j Q \right\|^2 - 2(\boldsymbol{x}_i Q)(\boldsymbol{x}_j Q)^T = \left\| \boldsymbol{x}_i Q \right\|^2 + \left\| xx_j Q \right\|^2 - 2\boldsymbol{x}_i Q Q^T \boldsymbol{x}_j^T = \left\| \boldsymbol{x}_i \right\|^2 + \left\| xx_j \right\|^2 - 2\boldsymbol{x}_i \boldsymbol{x}_j^T.$$

In fact, all the geometrical information about the vectors $\boldsymbol{x}_i$ is captured by their Gram matrix, whose (i,j) entry is $\boldsymbol{x}_i \boldsymbol{x}_j^T$. This matrix is also given by

$$UU^T.$$

Let W be the analogous matrix for the eigenvalue s. We then have

$$A = rUU^T + sWW^T + k\frac{1}{n}J.$$

As each of the matrices UU^T, WW^T and $\frac{1}{n}J$ are projections (having all eigenvalues 0 or 1), and are mutually orthogonal, we also have

$$A^2 = r^2UU^T + s^2WW^T + k^2\frac{1}{n}J.$$

Consider the polynomial

$$P(X) = \frac{(X-s)(X-k)}{(r-s)(r-k)}.$$

We have

$$P(X) = \begin{cases} 1 & \text{if } X = r \\ 0 & \text{if } X = s, \text{ and} \\ 0 & \text{if } X = k. \end{cases}$$

So,

$$P(A) = P(r)UU^T + P(s)WW^T + P(k)\frac{1}{n}J = UU^T.$$

That is, the Gram matrix of the point set $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ is a linear combination of the identity, A and A^2. So, the distance between any pair of points in this set just depends on whether or not the corresponding vertices are neighbors in G.

In particular, this means that the point set $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ is a *two-distance point set*: a set of points such that there are only two different distances between them. Next lecture, we will use this fact to prove a lower bound on the dimensions f and g.

9.10 Triangular Graphs

For a positive integer n, the *triangular graph* T_n may be defined to be the line graph of the complete graph on n vertices. To be more concrete, its vertices are the subsets of size 2 of $\{1, \dots, n\}$. Two of these sets are connected by an edge if their intersection has size 1.

You are probably familiar with some triangular graphs. T_3 is the triangle, T_4 is the skeleton of the octahedron, and T_5 is the complement of the Petersen graph.

Let's verify that these are strongly-regular, and compute their parameters. As the construction is competely symmetric, we may begin by considering any vertex, say the one labeled by the set $\{1, 2\}$. Every vertex labeled by a set of form $\{1, i\}$ or $\{2, i\}$, for $i \geq 3$, will be connected to this set. So, this vertex, and every vertex, has degree $2(n-2)$.

For any neighbor of $\{1, 2\}$, say $\{1, 3\}$, every other vertex of from $\{1, i\}$ for $i \geq 4$ will be a neighbor of both of these, as will the set $\{2, 3\}$. Carrying this out in general, we find that $\lambda = (n-3) + 1 = n-2$.

Finally, any non-neighbor of $\{1,2\}$, say $\{3,4\}$, will have 4 common neighbors with $\{1,2\}$:

$$\{1,3\}\,,\{1,4\}\,,\{2,3\}\,,\{2,4\}\,.$$

So, $\mu = 4$.

9.11 Paley Graphs

Probably next lecture.

9.12 Two-distance point sets

Recall from last lecture that each eigenspace of a strongly regular graph supply a set of points on the unit sphere such that the distance between a pair of points just depends on whether or not they are adjacent. If the graph is connected and not the complete graph, then we can show that these distances are greater than zero, so no two vertices map to the same unit vector. If we take the corresponding point sets for two strongly regular graphs with the same parameters, we can show that the graphs are isomorphic if and only if there is an orthogonal transformation that maps one point set to the other. In low dimensions, it is easy to find such an orthogonal transformation if one exists.

Consider the eigenspace of r, which we recall has dimension f. Fix any set of f independent vectors corresponding to f vertices. An orthogonal transformation is determined by its action on these vectors. So, if there is an orthogonal transformation that maps one vector set onto the other, we will find it by examining all orthogonal transformations determined by mapping these f vectors to f vectors in the other set. Thus, we need only examine $\binom{n}{f} f!$ transformations. This would be helpful if f were small. Unfortunately, it is not. We will now prove that both f and g must be at least $\sqrt{2n} - 2$.

Let $\boldsymbol{x}^1, \ldots, \boldsymbol{x}^n$ be a set of unit vectors in $\mathbb{R}^f$ such that there are two values $\alpha, \beta < 1$ such that

$$\left\langle \boldsymbol{x}^i, \boldsymbol{x}^j \right\rangle = \alpha \text{ or } \beta.$$

We will prove a lower bound on f in terms of n.

The key to our proof is to define an f-variate polynomial for each point. In particular, we set

$$p_i(\boldsymbol{y}) = (\boldsymbol{y}^T \boldsymbol{x}^i - \alpha)(\boldsymbol{y}^T \boldsymbol{x}^i - \beta),$$

for $\boldsymbol{y} \in \mathbb{R}^f$. We first note that each polynomial p_i is a polynomial of degree 2 in f variables (the coordinates of $\boldsymbol{y}$). As each f-variate polynomial of degree 2 can be expressed in the form

$$a + \sum_i b_i y_i + \sum_{i \leq k} c_{i,j} y_i y_j,$$

we see that the vector space of degree-2 polynomials in f variables has dimension

$$1 + 2f + \binom{f}{2}.$$

To prove a lower bound on f, we will show that these polynomials are linearly independent. Assume by way of contradiction that they are not. Then, without loss of generality, there exist coefficients $\gamma_1, \ldots, \gamma_n$ with $\gamma_1 \neq 0$ and

$$\sum_i \gamma_i p_i(y) = 0.$$

To obtain a contradiction, plug in $y = \boldsymbol{x}^1$, to find

$$\sum_i \gamma_i p_i(\boldsymbol{x}^1) = \gamma_1 p_1(\boldsymbol{x}^1) \neq 0.$$

Thus, we may conclude

$$n \leq 1 + 2f + \binom{f}{2},$$

which implies

$$f \geq \sqrt{2n} - 2.$$

Part III

Physical Metaphors

Chapter 10

Random Walks on Graphs

We will examine how the eigenvalues of a graph govern the convergence of a random walk.

10.1 Random Walks

We will consider random walks on undirected graphs. Let's begin with the definitions. Let $G = (V, E, w)$ be a weighted undirected graph. A random walk on a graph is a process that begins at some vertex, and at each time step moves to another vertex. When the graph is unweighted, the vertex the walk moves to is chosen uniformly at random among the neighbors of the present vertex. When the graph is weighted, it moves to a neighbor with probability proportional to the weight of the corresponding edge. While the transcript (the list of vertices in the order they are visited) of a particular random walk is sometimes of interest, it is often more productive to reason about the expected behavior of a random walk. To this end, we will investigate the probability distribution over vertices after a certain number of steps.

We will let the vector $\boldsymbol{p}_t \in \mathbb{R}^V$ denote the probability distribution at time t. We write $\boldsymbol{p}_t(a)$ to indicate the value of $\boldsymbol{p}_t$ at a vertex a—the probability of being at vertex a at time t. A probability vector $\boldsymbol{p}$ is a vector such that $\boldsymbol{p}(a) \geq 0$, for all $a \in V$, and

$$\sum_a \boldsymbol{p}(a) = 1.$$

Our initial probability distribution, $\boldsymbol{p}_0$, will typically be concentrated one vertex. That is, there will be some vertex a for which $\boldsymbol{p}_0(a) = 1$. In this case, we say that the walk starts at a.

To derive a $\boldsymbol{p}_{t+1}$ from $\boldsymbol{p}_t$, note that the probability of being at a vertex a at time $t+1$ is the sum over the neighbors b of a of the probability that the walk was at b at time t, times the probability it moved from b to a in time $t+1$. We can state this algebraically as

$$\boldsymbol{p}_{t+1}(a) = \sum_{b:(a,b)\in E} \frac{w(a,b)}{\boldsymbol{d}(b)} \boldsymbol{p}_t(b), \tag{10.1}$$

where $\boldsymbol{d}(b) = \sum_a w(a,b)$ is the weighted degree of vertex b.

We may write this in matrix form using the *walk matrix* of the graph, which is given by

$$\boldsymbol{W} \overset{\text{def}}{=} \boldsymbol{M}\boldsymbol{D}^{-1}.$$

We then have

$$\boldsymbol{p}_{t+1} = \boldsymbol{W}\boldsymbol{p}_t.$$

To see why this holds, consider how $\boldsymbol{W}$ acts as an operator on an elementary unit vector.

$$\boldsymbol{M}\boldsymbol{D}^{-1}\boldsymbol{\delta}_b = \boldsymbol{M}(\boldsymbol{\delta}_b/\boldsymbol{d}(b)) = \sum_{a\sim b}(w_{a,b}/\boldsymbol{d}(b))\boldsymbol{\delta}_a.$$

We will often consider lazy random walks, which are the variant of random walks that stay put with probability 1/2 at each time step, and walk to a random neighbor the other half of the time. These evolve according to the equation

$$\boldsymbol{p}_{t+1}(a) = (1/2)\boldsymbol{p}_t(a) + (1/2)\sum_{b:(a,b)\in E}\frac{w(a,b)}{\boldsymbol{d}(b)}\boldsymbol{p}_t(b), \tag{10.2}$$

and satisfy

$$\boldsymbol{p}_{t+1} = \widetilde{\boldsymbol{W}}\boldsymbol{p}_t,$$

where $\widetilde{\boldsymbol{W}}$ is the *lazy walk matrix* , given by

$$\widetilde{\boldsymbol{W}} \overset{\text{def}}{=} \frac{1}{2}\boldsymbol{I} + \frac{1}{2}\boldsymbol{W} = \frac{1}{2}\boldsymbol{I} + \frac{1}{2}\boldsymbol{M}\boldsymbol{D}^{-1}.$$

We will usually work with lazy random walks.

10.2 Spectra of Walk Matrices

While the walk matrices are not symmetric, they are similar to symmetric matrices. We will see that this implies that they have n real eigenvalues, although their eigenvectors are generally not orthogonal. Define the *normalized adjacency matrix* by

$$\boldsymbol{A} \overset{\text{def}}{=} \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{1/2} = \boldsymbol{D}^{-1/2}\boldsymbol{M}\boldsymbol{D}^{-1/2}.$$

So, $\boldsymbol{A}$ is symmetric.

Claim 10.2.1. *The vector $\boldsymbol{\psi}$ is an eigenvector of $\boldsymbol{A}$ of eigenvalue ω if and only if $\boldsymbol{D}^{1/2}\boldsymbol{\psi}$ is an eigenvector of $\boldsymbol{W}$ of eigenvalue ω.*

Proof. As $\boldsymbol{A} = D^{-1/2}\boldsymbol{W}\boldsymbol{D}^{1/2}$, $\boldsymbol{D}^{1/2}\boldsymbol{A} = \boldsymbol{W}\boldsymbol{D}^{1/2}$. Thus, $\boldsymbol{A}\boldsymbol{\psi} = \omega\boldsymbol{\psi}$ if and only if

$$\boldsymbol{D}^{1/2}\boldsymbol{A}\boldsymbol{\psi} = \boldsymbol{D}^{1/2}\omega\boldsymbol{\psi} = \omega(\boldsymbol{D}^{1/2}\boldsymbol{\psi}) = \boldsymbol{W}(\boldsymbol{D}^{1/2}\boldsymbol{\psi}).$$

□

Of course, $\widetilde{\boldsymbol{W}}$ has the same eigenvectors as $\boldsymbol{W}$.

We next observe that the degree vector, $\boldsymbol{d}$, is a Perron vector of $\boldsymbol{W}$ of eigenvalue 1:

$$\boldsymbol{M}\boldsymbol{D}^{-1}\boldsymbol{d} = \boldsymbol{M}\boldsymbol{1} = \boldsymbol{d}.$$

So, the Perron-Frobenius theorem (Theorem 4.5.1) tells us that all the eigenvalues of $\boldsymbol{W}$ lie between -1 and 1. As we did in Proposition 4.5.3, one can show that G is bipartite if and only if -1 is an eigenvalue of $\boldsymbol{A}$.

As $\widetilde{\boldsymbol{W}} = \boldsymbol{W}/2 + \boldsymbol{I}/2$, this implies that all the eigenvalues of $\widetilde{\boldsymbol{W}}$ lie between 0 and 1. We denote the eigenvalues of $\widetilde{\boldsymbol{W}}$ and $\boldsymbol{I}/2 + \boldsymbol{A}/2$ by

$$1 = \omega_1 \geq \omega_2 \geq \cdots \geq \omega_n \geq 0.$$

While the letter ω is not a greek equivalent "w", we use it because it looks like one.

From Claim 10.2.1, we now know that

$$\boldsymbol{\psi}_1 \stackrel{\text{def}}{=} \frac{\boldsymbol{d}^{1/2}}{\left\|\boldsymbol{d}^{1/2}\right\|}$$

is the unit-norm Perron vector of $\boldsymbol{A}$, where

$$\boldsymbol{d}^{1/2}(a) \stackrel{\text{def}}{=} \boldsymbol{d}(a)^{1/2}.$$

10.3 The stable distribution

Regardless of the starting distribution, the lazy random walk on a connected graph always converges to one distribution: the *stable distribution*. This is the other reason that we forced our random walk to be lazy. Without laziness[1], there can be graphs on which the random walks never converge. For example, consider a non-lazy random walk on a bipartite graph. Every-other step will bring it to the other side of the graph. So, if the walk starts on one side of the graph, its limiting distribution at time t will depend upon the parity of t.

In the stable distribution, every vertex is visited with probability proportional to its weighted degree. We denote the vector encoding this distribution by $\boldsymbol{\pi}$, where

$$\boldsymbol{\pi} \stackrel{\text{def}}{=} \boldsymbol{d}/(\boldsymbol{1}^T\boldsymbol{d}).$$

We have already seen that $\boldsymbol{\pi}$ is a right-eigenvector of eigenvalue 1. To show that the lazy random walk converges to $\boldsymbol{\pi}$, we will exploit the fact that all the eigenvalues other than 1 are in $[0,1)$. And, we expand the vectors $\boldsymbol{p}_t$ in the eigenbasis of $\boldsymbol{A}$, after first multiplying by $\boldsymbol{D}^{-1/2}$.

[1] Strictly speaking, any nonzero probability of staying put at any vertex in a connected graph will guarantee convergence. We don't really need a half probability at every vertex.

Let $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$ be the eigenvectors of $\boldsymbol{A}$ corresponding to eigenvalues $\omega_1, \ldots, \omega_n$ of $\widetilde{\boldsymbol{W}}$ (caution: the corresponding eigenvalues of $\boldsymbol{A}$ are $2\omega_i - 1$). For any initial distribution $\boldsymbol{p}_0$, write

$$\boldsymbol{D}^{-1/2}\boldsymbol{p}_0 = \sum_i c_i \boldsymbol{\psi}_i, \quad \text{where } c_i = \boldsymbol{\psi}_i^T \boldsymbol{D}^{-1/2}.$$

Note that

$$c_1 = \boldsymbol{\psi}_1^T (\boldsymbol{D}^{-1/2}\boldsymbol{p}_0) = \frac{(\boldsymbol{d}^{1/2})^T}{\|\boldsymbol{d}^{1/2}\|}(\boldsymbol{D}^{-1/2}\boldsymbol{p}_0) = \frac{\boldsymbol{1}^T\boldsymbol{p}_0}{\|\boldsymbol{d}^{1/2}\|} = \frac{1}{\|\boldsymbol{d}^{1/2}\|},$$

as $\boldsymbol{p}_0$ is a probability vector. One of the reasons we *do not* expand in a basis of eigenvectors of $\widetilde{\boldsymbol{W}}$ is that it, not being orthogonal, it does not allow such a nice expression for the coefficients. We have

$$\begin{aligned}
\boldsymbol{p}_t &= \widetilde{\boldsymbol{W}}^t \boldsymbol{p}_0 \\
&= \boldsymbol{D}^{1/2}\boldsymbol{D}^{-1/2}\widetilde{\boldsymbol{W}}^t \boldsymbol{D}^{1/2}\boldsymbol{D}^{-1/2}\boldsymbol{p}_0 \\
&= \boldsymbol{D}^{1/2}\left(\boldsymbol{D}^{-1/2}\widetilde{\boldsymbol{W}}\boldsymbol{D}^{1/2}\right)^t \boldsymbol{D}^{-1/2}\boldsymbol{p}_0 \\
&= \boldsymbol{D}^{1/2}\left(\boldsymbol{I}/2 + \boldsymbol{A}/2\right)^t \boldsymbol{D}^{-1/2}\boldsymbol{p}_0 \\
&= \boldsymbol{D}^{1/2}\left(\boldsymbol{I}/2 + \boldsymbol{A}/2\right)^t \sum_i c_i \boldsymbol{\psi}_i \\
&= \boldsymbol{D}^{1/2}\sum_i \omega_i^t c_i \boldsymbol{\psi}_i \\
&= \boldsymbol{D}^{1/2} c_1 \boldsymbol{\psi}_1 + \boldsymbol{D}^{1/2}\sum_{i \geq 2} \omega_i^t c_i \boldsymbol{\psi}_i
\end{aligned}$$

As $0 \leq \omega_i < 1$ for $i \geq 2$, the right-hand term must go to zero. On the other hand, $\boldsymbol{\psi}_1 = \boldsymbol{d}^{1/2}/\|\boldsymbol{d}^{1/2}\|$, so

$$\boldsymbol{D}^{1/2} c_1 \boldsymbol{\psi}_1 = \boldsymbol{D}^{1/2}\left(\frac{1}{\|\boldsymbol{d}^{1/2}\|}\right)\frac{\boldsymbol{d}^{1/2}}{\|\boldsymbol{d}^{1/2}\|} = \frac{\boldsymbol{d}}{\|\boldsymbol{d}^{1/2}\|^2} = \frac{\boldsymbol{d}}{\sum_a \boldsymbol{d}(a)} = \boldsymbol{\pi}.$$

This is a perfect example of one of the main uses of spectral theory: to understand what happens when we repeatedly apply an operator.

10.4 The Rate of Convergence

The rate of convergence of a lazy random walk to the stable distribution is dictated by ω_2: a small value of ω_2 implies fast convergence.

There are many ways of measuring convergence of a random walk. We will do so point-wise. Assume that the random walk starts at some vertex $a \in V$. For every vertex b, we will bound how far $\boldsymbol{p}_t(b)$ can be from $\boldsymbol{\pi}(b)$.

Theorem 10.4.1. *For all a, b and t, if $\boldsymbol{p}_0 = \boldsymbol{\delta}_a$, then*

$$|\boldsymbol{p}_t(b) - \boldsymbol{\pi}(b)| \leq \sqrt{\frac{\boldsymbol{d}(b)}{\boldsymbol{d}(a)}}\omega_2^t.$$

Proof. Observe that

$$\boldsymbol{p}_t(b) = \boldsymbol{\delta}_b^T \boldsymbol{p}_t.$$

From the analysis in the previous section, we know

$$\boldsymbol{p}_t(b) = \boldsymbol{\delta}_b^T \boldsymbol{p} = \boldsymbol{\pi}(b) + \boldsymbol{\delta}_b^T \boldsymbol{D}^{1/2} \sum_{i \geq 2} \omega_i^t c_i \boldsymbol{\psi}_i.$$

We need merely prove an upper bound on the magnitude of the right-hand term. To this end, recall that

$$c_i = \boldsymbol{\psi}_i^T \boldsymbol{D}^{-1/2} \boldsymbol{\delta}_a = \frac{1}{\sqrt{\boldsymbol{d}(a)}} \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a.$$

So,

$$\boldsymbol{\delta}_b^T \boldsymbol{D}^{1/2} \sum_{i \geq 2} \omega_i^t c_i \boldsymbol{\psi}_i = \sqrt{\frac{\boldsymbol{d}(b)}{\boldsymbol{d}(a)}} \boldsymbol{\delta}_b^T \sum_{i \geq 2} \omega_i^t \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a.$$

Analyzing the right-hand part of this last expression, we find

$$\begin{aligned}
\left| \boldsymbol{\delta}_b^T \sum_{i \geq 2} \omega_i^t \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right| &= \left| \sum_{i \geq 2} \omega_i^t \left(\boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right) \left(\boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right) \right| \\
&\leq \sum_{i \geq 2} \omega_i^t \left| \boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right| \left| \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right| \\
&\leq \omega_2^t \sum_{i \geq 2} \left| \boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right| \left| \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right| \\
&\leq \omega_2^t \sum_{i \geq 1} \left| \boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right| \left| \boldsymbol{\psi}_i^T \boldsymbol{\delta}_a \right| \\
&\leq \omega_2^t \sqrt{\sum_{i \geq 1} \left(\boldsymbol{\delta}_b^T \boldsymbol{\psi}_i \right)^2} \sqrt{\sum_{i \geq 1} \left(\boldsymbol{\delta}_a^T \boldsymbol{\psi}_i \right)^2} && \text{by Cauchy-Schwartz} \\
&= \omega_2^t \left\| \boldsymbol{\delta}_b \right\| \left\| \boldsymbol{\delta}_a \right\|, \quad \text{as the eigenvectors form an orthonormal basis,} \\
&= \omega_2^t
\end{aligned}$$

□

10.5 Relation to the Normalized Laplacian

The walk matrix is closely related to the *normalized Laplacian*, which is defined by

$$\boldsymbol{N} = \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2} \boldsymbol{M} \boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{A}.$$

We let $0 \leq \nu_1 \leq \nu_2 \leq \cdots \leq \nu_n$ denote the eigenvalues of $\boldsymbol{N}$, and note that they have the same eigenvectors as $\boldsymbol{A}$. Other useful relations include

$$\nu_i = 2 - 2\omega_i, \quad \omega_i = 1 - \nu_i/2,$$

and

$$\widetilde{\boldsymbol{W}} = \boldsymbol{I} - \frac{1}{2}\boldsymbol{D}^{1/2}\boldsymbol{N}\boldsymbol{D}^{-1/2}.$$

The normalized Laplacian is positive semidefinite and has the same rank as the ordinary (sometimes called "combinatorial") Laplacian. There are many advantages of working with the normalized Laplacian: the mean of its eigenvalues is 1, so they are always on a degree-independent scale. One can prove that $\nu_n \leq 2$, with equality if and only if the graph is bipartite.

The bound in Theorem 10.4.1 can be expressed in the eigenvalues of the normalized Laplacian as

$$|\boldsymbol{p}_t(b) - \boldsymbol{\pi}(b)| \leq \sqrt{\frac{\boldsymbol{d}(b)}{\boldsymbol{d}(a)}}(1 - \nu_2/2)^t.$$

We will say that a walk has *mixed* if

$$|\boldsymbol{p}_t(b) - \boldsymbol{\pi}(b)| \leq \boldsymbol{\pi}(b)/2,$$

for all vertices b. Using the approximation $1 - x \approx \exp(-x)$, we see that this should happen once

$$\begin{aligned}
\sqrt{\frac{\boldsymbol{d}(b)}{\boldsymbol{d}(a)}}(1 - \nu_2/2)^t &\leq \boldsymbol{d}(b)/2\boldsymbol{d}(V) && \iff \\
(1 - \nu_2/2)^t &\leq \sqrt{\boldsymbol{d}(b)\boldsymbol{d}(a)}/2\boldsymbol{d}(V) && \iff \\
\exp(-t\nu_2/2) &\leq \sqrt{\boldsymbol{d}(b)\boldsymbol{d}(a)}/2\boldsymbol{d}(V) && \iff \\
-t\nu_2/2 &\leq \ln\left(\sqrt{\boldsymbol{d}(b)\boldsymbol{d}(a)}/2\boldsymbol{d}(V)\right) && \iff \\
t &\geq 2\ln\left(2\boldsymbol{d}(V)/\sqrt{\boldsymbol{d}(b)\boldsymbol{d}(a)}\right)/\nu_2.
\end{aligned}$$

So, for graphs in which all degrees are approximately constant, this upper bound on the time to mix is approximately $\ln(n)/\nu_2$. For some graphs the $\ln n$ term does not appear. Note that multiplying all edge weights by a constant does not change any of these expressions.

While we have explicitly worked out λ_2 for many graphs, we have not done this for ν_2. The following lemma will allow us to relate bounds on λ_2 to bounds on ν_2:

Lemma 10.5.1. *Let $\boldsymbol{L}$ be the Laplacian matrix of a graph, with eigenvalues $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, and let $\boldsymbol{N}$ be its normalized Laplacian, with eigenvalues $\nu_1 \leq \nu_2 \leq \cdots \leq \nu_2$. Then, for all i*

$$\frac{\lambda_i}{d_{min}} \geq \nu_i \geq \frac{\lambda_i}{d_{max}},$$

where d_{min} and d_{max} are the minimum and maximum degrees of vertices in the graph.

Proof. The Courant-Fischer theorem tells us that

$$\nu_i = \min_{\dim(S)=i} \max_{x \in S} \frac{\boldsymbol{x}^T \boldsymbol{N} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

As the change of variables $\boldsymbol{y} = \boldsymbol{D}^{-1/2}\boldsymbol{x}$ is non-singular, this equals

$$\min_{\dim(T)=i} \max_{\boldsymbol{y} \in T} \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}}.$$

So,

$$\min_{\dim(T)=i} \max_{\boldsymbol{y} \in T} \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}} \geq \min_{\dim(T)=i} \max_{\boldsymbol{y} \in T} \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{d_{max} \boldsymbol{y}^T \boldsymbol{y}} = \frac{1}{d_{max}} \min_{\dim(T)=i} \max_{\boldsymbol{y} \in T} \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{y}} = \frac{\lambda_i}{d_{max}}.$$

The other bound may be proved similarly. □

10.6 Examples

We now do some examples. For each we think about the random walk in two ways: by reasoning directly about how a random walk should behave and by examining ν_2.

10.6.1 The Path

As every vertex in the path on n vertices has degree 1 or 2, ν_2 is approximately λ_2, which is approximately c/n^2 for some constant c.

To understand the random walk on the path, think about what happens when the walk starts in the middle. Ignoring the steps on which it stays put, it will either move to the left or the right with probability $1/2$. So, the position of the walk after t steps is distributed as the sum of t random variables taking values in $\{1, -1\}$. Recall that the standard deviation of such a sum is $\sqrt{t}$. So, we need to have $\sqrt{t}$ comparable to $n/4$ for there to be a reasonable chance that the walk is on the left or right $n/4$ vertices.

10.6.2 The Complete Binary Tree

As with the path, ν_2 for the tree is within a constant of λ_2 for the tree, and so is approximately c/n for some constant c. To understand the random walk on T_n, first note that whenever it is at a vertex, it is twice as likely to step towards a leaf as it is to step towards the root. So, if the walk starts at a leaf, there is no way the walk can mix until it reaches the root. The height of the walk is like a sum of ± 1 random variables, except that they are twice as likely to be -1 as they are to be 1, and that their sum never goes below 0. One can show that we need to wait approximately n steps before such a walk will hit the root. Once it does hit the root, the walk mixes rapidly.

10.6.3 The Dumbbell

The dumbell graph D_n consists of two complete graphs on n vertices, joined by one edge called the "bridge". So, there are $2n$ vertices in total, and all vertices have degree $n-1$ or n.

To understand the random walk on this graph, consider starting it at some vertex that is not attached to the bridge edge. After the first step the walk will be well mixed on the vertices in the side on which it starts. Because of this, the chance that it finds the edge going to the other side is only around $1/n^2$: there is only a $1/n$ chance of being at the vertex attached to the bridge edge, and only a $1/n$ chance of choosing that edge when at that vertex. So, we must wait some multiple of n^2 steps before there is a reasonable chance that the walk reaches the other side of the graph.

The isoperimetric ratio of this graph is

$$\theta_{D_n} \sim \frac{1}{n}.$$

Using the test vector that is 1 on one complete graph and -1 on the other, we can show that

$$\lambda_2(D_n) \lessapprox 1/n.$$

Lemma 10.5.1 then tells us that

$$\nu_2(D_n) \lessapprox 1/n^2.$$

To prove that this bound is almost tight, we use the following lemma.

Lemma 10.6.1. *Let G be an unweighted graph of diameter at most r. Then,*

$$\lambda_2(G) \geq \frac{2}{r(n-1)}.$$

Proof. For every pair of vertices (a,b), let $P(a,b)$ be a path in G of length at most r. We have

$$\boldsymbol{L}_{(a,b)} \preccurlyeq r \cdot \boldsymbol{L}_{P(a,b)} \preccurlyeq r\boldsymbol{L}_G.$$

So,

$$K_n \preccurlyeq r\binom{n}{2}G,$$

and

$$n \leq r\binom{n}{2}\lambda_2(G),$$

from which the lemma follows. □

The diameter of D_n is 3, so we have $\lambda_2(D_n) \geq 2/3(n-1)$. As every vertex of D_n has degree at least $n-1$, we may conclude $\nu_2(D_n) \gtrapprox 2/3(n-1)^2$.

10.6.4 The Bolas Graph

We define the bolas[2] graph B_n to be a graph containing two n-cliques connected by a path of length n. The bolas graph has a value of ν_2 that is almost as small as possible. Equivalently, random walks on a bolas graph mix almost as slowly as possible.

The analysis of the random walk on a bolas is similar to that on a dumbbell, except that when the walk is on the first vertex of the path the chance that it gets to the other end before moving back to the clique at which we started is only $1/n$. So, we must wait around n^3 steps before there is a reasonable chance of getting to the other side.

Next lecture, we will learn that we can upper bound ν_2 with a test vector using the fact that

$$\nu_2 = \min_{\boldsymbol{x} \perp \boldsymbol{d}} \frac{\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{D} \boldsymbol{x}}.$$

To prove an upper bound on ν_2, form a test vector that is $n/2$ on one clique, $-n/2$ on the other, and increases by 1 along the path. We can use the symmetry of the construction to show that this vector is orthogonal to $\boldsymbol{d}$. The numerator of the generalized Rayleigh quotient is n, and the denominator is the sum of the squares of the entries of the vectors times the degrees of the vertices, which is some constant times n^4. This tells us that ν_2 is at most some constant over n^3.

To see that ν_2 must be at least some constant over n^3, and in fact that this must hold for every graph, apply Lemmas 10.5.1 and 10.6.1.

10.7 Diffusion

There are a few types of diffusion that people study in a graph, but the most common is closely related to random walks. In a diffusion process, we imagine that we have some substance that can occupy the vertices, such as a gas or fluid. At each time step, some of the substance diffuses out of each vertex. If we say that half the substance stays at a vertex at each time step, and the other half is distributed among its neighboring vertices, then the distribution of the substance will evolve according to equation (10.2). That is, probability mass obeys this diffusion equation.

People often consider finer time steps in which smaller fractions of the mass leave the vertices. In the limit, this results in continuous random walks that are modeled by the matrix exponential: if the walk stays put with probability $1-\epsilon$ in each step, and we view each step as taking time ϵ, then the transition matrix of the walk after time t will be

$$((1-\epsilon)\boldsymbol{I} + \epsilon \boldsymbol{W})^{t/\epsilon} \rightarrow \exp(t(\boldsymbol{W} - \boldsymbol{I})).$$

These are in many ways more natural than discrete time random walks.

[2] A bolas is a hunting weapon consisting of two balls or rocks tied together with a cord.

10.8 Final Notes

The procedure we have described—repeatedly multiplying a vector by $\widetilde{\boldsymbol{W}}$ and showing that the result approximates $\boldsymbol{\pi}$—is known in Numerical Linear Algebra as the power method. It is one of the common ways of approximately computing eigenvectors.

In the proof of Theorem 10.4.1, we were a little loose with some of the terms. The slack comes from two sources. First, we upper bounded ω_i by ω_2 for all i, while many of the ω_i are probably significantly less than ω_2. This phenomenon is often called "eigenvalue decay", and it holds in many graphs. This sloppiness essentially costs us a multiplicative factor of $\log n$ in the number of steps t we need to achieve the claimed bound. You will note that in the examples above, the time to approximate convergence is typically on the order of $1/\nu_2$, not $(\log n)/\nu_2$. This is because of eigenvalue decay.

The second source of slack appeared when we upper bounded the absolute value of a sum by the sum of the absolute value.

Chapter 11

Walks, Springs, and Resistor Networks

Lecture 12 from October 8, 2018

11.1 Overview

In this lecture we will see how the analysis of random walks, spring networks, and resistor networks leads to the consideration of systems of linear equations in Laplacian matrices. The main purpose of this lecture is to introduce concepts and language that we will use extensively in the rest of the course.

11.2 Harmonic Functions

The theme of this whole lecture will be *harmonic functions* on graphs. These will be defined in terms of a weighted graph $G = (V, E, w)$ and a set of *boundary* vertices $B \subseteq V$. We let $S = V - B$ (I use "-" for set-minus). We will assume throughout this lecture that G is connected and that B is nonempty.

A function $\boldsymbol{x} : V \to \mathbf{R}$ is said to be *harmonic* at a vertex a if the value of $\boldsymbol{x}$ at a is the weighted average of its values at the neighbors of a where the weights are given by w:

$$\boldsymbol{x}(a) = \frac{1}{d_a} \sum_{b \sim a} w_{a,b} \boldsymbol{x}(b). \tag{11.1}$$

The function $\boldsymbol{x}$ is harmonic on S if it is harmonic for all $a \in S$.

11.3 Random Walks on Graphs

Consider the standard (not lazy) random walk on the graph G. Recall that when the walk is at a vertex a, the probability it moves to a neighbor b is

$$\frac{w_{a,b}}{d_a}.$$

Distinguish two special nodes in the graph that we will call s and t, and run the random walk until it hits either s or t. We view s and t as the boundary, so $B = \{s, t\}$.

Let $\boldsymbol{x}(a)$ be the probability that a walk that starts at a will stop at s, rather than at t. We have the boundary conditions $\boldsymbol{x}(s) = 1$ and $\boldsymbol{x}(t) = 0$. For every other node a the chance that the walk stops at s is the sum over the neighbors b of a of the chance that the walk moves to b, times the chance that a walk from b stops at s. That is,

$$\boldsymbol{x}(a) = \sum_{b \sim a} \frac{w_{a,b}}{d_a} \boldsymbol{x}(b).$$

So, the function $\boldsymbol{x}$ is harmonic at every vertex in $V - B$.

For example, consider the path graph P_n. Let's make $s = n$ and $t = 1$. So, the walk stops at either end. We then have $\boldsymbol{x}(n) = 1$, $\boldsymbol{x}(1) = 0$. It is easy to construction at least one solution to the harmonic equations (11.1): we can set

$$\boldsymbol{x}(a) = \frac{a-1}{n-1}.$$

It essentially follows from the definitions that there can be only one vector $\boldsymbol{x}$ that solves these equations. But, we will prove this algebraically later in lecture.

These solutions tell us that if the walk starts at node a, the chance that it ends at node n is $(a-1)/(n-1)$. This justifies some of our analysis of the Bolas graph from Lecture 10.

Of course, the exact same analysis goes through for the lazy random walks: those give

$$\boldsymbol{x}(a) = (1/2)\boldsymbol{x}(a) + (1/2)\sum_{b \sim a} \frac{w_{a,b}}{d_a} \boldsymbol{x}(b) \quad \iff \quad \boldsymbol{x}(a) = \sum_{b \sim a} \frac{w_{a,b}}{d_a} \boldsymbol{x}(b).$$

11.4 Spring Networks

We begin by imagining that every edge of a graph $G = (V, E)$ is an ideal spring or rubber band. They are joined together at the vertices. Given such a structure, we will pick a subset of the vertices $B \subseteq V$ and fix the location of every vertex in B. For example, you could nail each vertex in B onto a point in the real line, or onto a board in $\mathbb{R}^2$. We will then study where the other vertices wind up.

We can use Hooke's law to figure this out. To begin, assume that each rubber band is an ideal spring with spring constant 1. If your graph is weighted, then the spring constant of each edge should be its weight. If a rubber band connects vertices a and b, then Hooke's law tells us that the force it exerts at node a is in the direction of b and is proportional to the distance between a and b. Let $\boldsymbol{x}(a)$ be the position of each vertex a. You should begin by thinking of $\boldsymbol{x}(a)$ being in $\mathbb{R}$, but you will see that it is just as easy to make it a vector in $\mathbb{R}^2$ or $\mathbb{R}^k$ for any k.

The force the rubber band between a and b exerts on a is

$$\boldsymbol{x}(b) - \boldsymbol{x}(a).$$

In a stable configuration, all of the vertices that have not been nailed down must experience a zero net force. That is

$$\begin{aligned}\sum_{b \sim a} (\boldsymbol{x}(b) - \boldsymbol{x}(a)) = 0 &\iff \sum_{b \sim a} \boldsymbol{x}(b) = d_a \boldsymbol{x}(a) \\ &\iff \frac{1}{d_a} \sum_{b \sim a} \boldsymbol{x}(b) = \boldsymbol{x}(a).\end{aligned}$$

In a stable configuration, every vertex that is not on the boundary must be the average of its neighbors.

In the weighted case, we would have for each $a \in V - B$

$$\frac{1}{d_a} \sum_{b \sim a} w_{a,b} \boldsymbol{x}(b) = \boldsymbol{x}(a).$$

That is, $\boldsymbol{x}$ is *harmonic* on $V - B$.

We will next show that the equations (11.1) have a solution, and that it is unique[1] if the underlying graph is connected and B is nonempty But first, consider again the path graph P_n with the endpoints fixed: $B = \{1, n\}$. Let us fix them to the values $\boldsymbol{f}(1) = 1$ and $\boldsymbol{f}(n) = n$. The only solution to the equations (11.1) is the obvious one: vertex i is mapped to i: $\boldsymbol{x}(i) = i$ for all i.

11.5 Laplacian linear equations

If we rewrite equation (11.1) as

$$d_a \boldsymbol{x}(a) - \sum_{b \sim a} w_{a,b} \boldsymbol{x}(b) = 0, \tag{11.2}$$

we see that it corresponds to the row of the Laplacian matrix corresponding to vertex a. So, we may find a solution to the equations (11.1) by solving a system of equations in the submatrix of the Laplacian indexed by vertices in $V - B$.

To be more concete, I will set up those equations. For each vertex $a \in B$, let its position be fixed to $\boldsymbol{f}(a)$. Then, we can re-write equation (11.2) as

$$d_a \boldsymbol{x}(a) - \sum_{b \notin B : (a,b) \in E} w_{a,b} \boldsymbol{x}(b) = \sum_{b \in B : (a,b) \in E} w_{a,b} \boldsymbol{f}(b),$$

[1] It can only fail to be unique if there is a connected component that contains no vertices of B.

for each $a \in V - B$. So, all of the boundary terms wind up in the right-hand vector.

Let $S = V - B$. We now see that this is an equation of the form

$$\boldsymbol{L}(S,S)\boldsymbol{x}(S) = \boldsymbol{r}, \qquad \text{with } \boldsymbol{r} = \boldsymbol{M}(S,:)\boldsymbol{f}.$$

By $\boldsymbol{L}(S,S)$ I mean the submatrix of $\boldsymbol{L}$ indexed by rows and columns of S, and by $\boldsymbol{x}(S)$ I mean the sub-vector of $\boldsymbol{x}$ indexed by S.

We can then write the condition that entries of B are fixed to $\boldsymbol{f}$ by

$$\boldsymbol{x}(B) = \boldsymbol{f}(B).$$

We have reduced the problem to that of solving a system of equations in a submatrix of the Laplacian.

Submatrices of Laplacians are a lot like Laplacians, except that they are positive definite. To see this, note that all of the off-diagonals of the submatrix of $\boldsymbol{L}$ agree with all the off-diagonals of the Laplacian of the induced subgraph on the internal vertices. But, some of the diagonals are larger: the diagonals of nodes in the submatrix account for both edges in the induced subgraph and edges to the vertices in B.

Claim 11.5.1. *Let $\boldsymbol{L}$ be the Laplacian of $G = (V, E, w)$, let $B \subseteq V$, and let $S = V - B$. Then,*

$$\boldsymbol{L}(S,S) = \boldsymbol{L}_{G(S)} + \boldsymbol{X}_S,$$

where $G(S)$ is the subgraph induced on the vertices in S and $\boldsymbol{X}_S$ is the diagonal matrix with entries

$$\boldsymbol{X}_S(a,a) = \sum_{b \sim a, b \in B} w_{a,b}, \quad \textit{for } a \in S.$$

Lemma 11.5.2. *Let $\boldsymbol{L}$ be the Laplacian matrix of a connected graph and let $\boldsymbol{X}$ be a nonnegative, diagonal matrix with at least one nonzero entry. Then, $\boldsymbol{L} + \boldsymbol{X}$ is positive definite.*

Proof. We will prove that $\boldsymbol{x}^T(\boldsymbol{L} + \boldsymbol{X})\boldsymbol{x} > 0$ for every nonzero vector $\boldsymbol{x}$. As both $\boldsymbol{L}$ and $\boldsymbol{X}$ are positive semidefinite, we have

$$\boldsymbol{x}^T(\boldsymbol{L} + \boldsymbol{X})\boldsymbol{x} \geq \min\left(\boldsymbol{x}^T\boldsymbol{L}\boldsymbol{x}, \boldsymbol{x}^T\boldsymbol{X}\boldsymbol{x}\right).$$

As the graph is connected, $\boldsymbol{x}^T\boldsymbol{L}\boldsymbol{x}$ is positive unless $\boldsymbol{x}$ is a constant vector. If $\boldsymbol{x} = c\boldsymbol{1}$ for some $c \neq 0$, then we obtain

$$c^2\boldsymbol{1}^T(\boldsymbol{L} + \boldsymbol{X})\boldsymbol{1} = c^2\boldsymbol{1}^T\boldsymbol{X}\boldsymbol{1} = c^2\sum_i \boldsymbol{X}(i,i) > 0.$$

□

Lemma 11.5.3. *Let $\boldsymbol{L}$ be the Laplacian matrix of a connected graph $G = (V, E, w)$, let B be a nonempty, proper subset of V, and let $S = V - B$. Then, $\boldsymbol{L}(S,S)$ is positive definite.*

Proof. Let $S_1, \dots, S_k$ be the connected components of vertices of $G(S)$. We can use these to write $\boldsymbol{L}(S,S)$ as a block matrix with blocks equal to $\boldsymbol{L}(S_i,S_i)$. Each of these blocks can be written

$$\boldsymbol{L}(S_i,S_i) = \boldsymbol{L}_{G_{S_i}} + X_{S_i}.$$

As G is connected, there must be some vertex in S_i with an edge to a vertex not in S_i. This implies that X_{S_i} is not the zero matrix, and so we can apply Lemma 11.5.2 to prove that $\boldsymbol{L}(S_i,S_i)$ is invertible. □

As the matrix $\boldsymbol{L}(S,S)$ is invertible, the equations have a solution, and it must be unique.

11.6 Energy

Physics also tells us that the vertices will settle into the position that minimizes the potential energy. The potential energy of an ideal linear spring with constant w when stretched to length l is

$$\frac{1}{2}wl^2.$$

So, the potential energy in a configuration $\boldsymbol{x}$ is given by

$$\mathcal{E}\left(\boldsymbol{x}\right) \overset{\text{def}}{=} \frac{1}{2}\sum_{(a,b)\in E} w_{a,b}(\boldsymbol{x}(a)-\boldsymbol{x}(b))^2. \tag{11.3}$$

For any $\boldsymbol{x}$ that minimizes the energy, the partial derivative of the energy with respect to each variable must be zero. In this case, the variables are $\boldsymbol{x}(a)$ for $a \in S$. The partial derivative with respect to $\boldsymbol{x}(a)$ is

$$\frac{1}{2}\sum_{b\sim a} w_{a,b}2(\boldsymbol{x}(a)-\boldsymbol{x}(b)) = \sum_{b\sim a} w_{a,b}(\boldsymbol{x}(a)-\boldsymbol{x}(b)).$$

Setting this to zero gives the equations we previously derived: (11.1).

For future reference, we state this result as a theorem.

Theorem 11.6.1. *Let $G=(V,E,w)$ be a connected, weighted graph, let $B \subset V$, and let $S = V - B$. Given $\boldsymbol{x}(B)$, $\mathcal{E}\left(\boldsymbol{x}\right)$ is minimized by setting $\boldsymbol{x}(S)$ so that $\boldsymbol{x}$ is harmonic on S.*

11.7 Resistor Networks

We now consider a related physical model of a graph in which we treat every edge as a resistor. If the graph is unweighted, we will assume that each resistor has resistance 1. If an edge e has weight w_e, we will give the corresponding resistor resistance $r_e = 1/w_e$. The reason is that when the weight of an edge is very small, the edge is barely there, so it should correspond to very high resistance. Having no edge corresponds to having a resistor of infinite resistance.

Recall Ohm's law:

$$V = IR.$$

That is, the potential drop across a resistor (V) is equal to the current flowing over the resistor (I) times the resistance (R). To apply this in a graph, we will define for each edge (a,b) the current flowing *from a to b* to be $\boldsymbol{i}(a,b)$. As this is a directed quantity, we define

$$\boldsymbol{i}(b,a) = -\boldsymbol{i}(a,b).$$

I now let $\boldsymbol{v} \in \mathbb{R}^V$ be a vector of potentials (voltages) at vertices. Given these potentials, we can figure out how much current flows on each edge by the formula

$$\boldsymbol{i}(a,b) = \frac{1}{r_{a,b}} \left(\boldsymbol{v}(a) - \boldsymbol{v}(b) \right) = w_{a,b} \left(\boldsymbol{v}(a) - \boldsymbol{v}(b) \right).$$

That is, we adopt the convention that current flows from high voltage to low voltage. We would like to write this equation in matrix form. The one complication is that each edge comes up twice in $\boldsymbol{i}$. So, to treat $\boldsymbol{i}$ as a vector we will have each edge show up exactly once as (a,b) when $a < b$. We now define the *signed edge-vertex adjacency matrix* of the graph $\boldsymbol{U}$ to be the matrix with rows indexed by edges and columns indexed by vertices such that

$$\boldsymbol{U}((a,b),c) = \begin{cases} 1 & \text{if } a = c \\ -1 & \text{if } b = c \\ 0 & \text{otherwise.} \end{cases}$$

Thus the row of $\boldsymbol{U}$ corresponding to edge (a,b) is $U((a,b),:) = \boldsymbol{\delta}_a^T - \boldsymbol{\delta}_b^T$.

Define $\boldsymbol{W}$ to be the diagonal matrix with rows and columns indexed by edges with the weights of the edges on the diagonals. We then have

$$\boldsymbol{i} = \boldsymbol{W} \boldsymbol{U} \boldsymbol{v}.$$

Also recall that resistor networks cannot hold current. So, all the current entering a vertex a from edges in the graph must exit a to an external source. Let $\boldsymbol{i}_{ext} \in \mathbb{R}^V$ denote the external currents, where $\boldsymbol{i}_{ext}(a)$ is the amount of current entering the graph through node a. We then have

$$\boldsymbol{i}_{ext}(a) = \sum_{b \sim a} \boldsymbol{i}(a,b).$$

In matrix form, this becomes

$$\boldsymbol{i}_{ext} = \boldsymbol{U}^T \boldsymbol{i} = \boldsymbol{U}^T \boldsymbol{W} \boldsymbol{U} \boldsymbol{v}. \tag{11.4}$$

The matrix

$$\boldsymbol{L} \overset{\text{def}}{=} \boldsymbol{U}^T \boldsymbol{W} \boldsymbol{U}$$

is, of course, the Laplacian. This is another way of writing the expression that we derived in Lecture 3:

$$\boldsymbol{L} = \sum_{a \sim b} w_{a,b} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T.$$

It is often helpful to think of the nodes a for which $\boldsymbol{i}_{ext}(a) \neq 0$ as being *boundary* nodes. We will call the other nodes *internal.* Let's see what the equation

$$\boldsymbol{i}_{ext} = \boldsymbol{L}\boldsymbol{v}.$$

means for the internal nodes. If the graph is unweighted and a is an internal node, then the ath row of this equation is

$$0 = (\boldsymbol{\delta}_a^T \boldsymbol{L})\boldsymbol{v} = \sum_{a \sim b} (\boldsymbol{v}(a) - \boldsymbol{v}(b)) = d_a \boldsymbol{v}(a) - \sum_{a \sim b} \boldsymbol{v}(b).$$

That is,

$$\boldsymbol{v}(a) = \frac{1}{d_a} \sum_{a \sim b} \boldsymbol{v}(b),$$

which means that $\boldsymbol{v}$ is harmonic at a. Of course, the same holds in weighted graphs.

11.8 Solving for currents

We are often interested in applying (11.4) in the reverse: given a vector of external currents $\boldsymbol{i}_{ext}$ we solve for the induced voltages by

$$\boldsymbol{v} = \boldsymbol{L}^{-1}\boldsymbol{i}_{ext}.$$

This at first appears problematic, as the Laplacian matrix does not have an inverse. The way around this problem is to observe that we are only interested in solving these equations for vectors $\boldsymbol{i}_{ext}$ for which the system has a solution. In the case of a connected graph, this equation will have a solution if the sum of the values of $\boldsymbol{i}_{ext}$ is zero. That is, if the current going in to the circuit equals the current going out. These are precisely the vectors that are in the span of the Laplacian.

To obtain the solution to this equation, we multiply $\boldsymbol{i}_{ext}$ by the Moore-Penrose *pseudo-inverse* of $\boldsymbol{L}$.

Definition 11.8.1. *The pseudo-inverse of a symmetric matrix $\boldsymbol{L}$, written $\boldsymbol{L}^+$, is the matrix that has the same span as $\boldsymbol{L}$ and that satisfies*

$$\boldsymbol{L}\boldsymbol{L}^+ = \boldsymbol{\Pi},$$

where $\boldsymbol{\Pi}$ is the symmetric projection onto the span of $\boldsymbol{L}$.

I remind you that a matrix $\boldsymbol{\Pi}$ is a symmetric projetion if $\boldsymbol{\Pi}$ is symmetric and $\boldsymbol{\Pi}^2 = \boldsymbol{\Pi}$. This is equivalent to saying that all of its eigenvalues are 0 or 1. We also know that $\boldsymbol{\Pi} = (1/n)\boldsymbol{L}_{K_n}$.

The symmetric case is rather special. As $\boldsymbol{L}\boldsymbol{\Pi} = \boldsymbol{L}$, the other following properties of the Moore-Penrose pseudo inverse follow from this one:

$$\begin{aligned} \boldsymbol{L}^+\boldsymbol{L} &= \boldsymbol{\Pi}, \\ \boldsymbol{L}\boldsymbol{L}^+\boldsymbol{L} &= \boldsymbol{L} \\ \boldsymbol{L}^+\boldsymbol{L}\boldsymbol{L}^+ &= \boldsymbol{L}^+. \end{aligned}$$

It is easy to find a formula for the pseudo-inverse. First, let $\boldsymbol{\Psi}$ be the matrix whose ith column is $\boldsymbol{\psi}_i$ and let $\boldsymbol{\Lambda}$ be the diagonal matrix with λ_i on the ith diagonal. Recall that

$$\boldsymbol{L} = \boldsymbol{\Psi} \boldsymbol{\Lambda} \boldsymbol{\Psi}^T = \sum_i \lambda_i \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T .$$

Claim 11.8.2.

$$\boldsymbol{L}^+ = \sum_{i>1} (1/\lambda_i) \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T .$$

11.9 Exercise

Prove that for every $p > 0$

$$\boldsymbol{L}^p = \boldsymbol{\Psi} \boldsymbol{\Lambda}^p \boldsymbol{\Psi}^T = \sum_i \lambda_i^p \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T .$$

Moreover, this holds for any symmetric matrix. Not just Laplacians.

Chapter 12

Effective Resistance and Schur Complements

The effective resistance between two vertices a and b in an electrical network is the resistance of the entire network when we treat it as one complex resistor. That is, we reduce the rest of the network to a single edge. In general, we will see that if we wish to restrict our attention to a subset of the vertices, B, and if we require all other vertices to be internal, then we can construct a network just on B that factors out the contributions of the internal vertices. The process by which we do this is Gaussian elimination, and the Laplacian of the resulting network on B is called a Schur complement.

12.1 Electrical Flows and Effective Resistance

We now know that if a resistor network has external currents $\boldsymbol{i}_{ext}$, then the voltages induced at the vertices will be given by

$$\boldsymbol{v} = \boldsymbol{L}^{+} \boldsymbol{i}_{ext}.$$

Consider what this means when $\boldsymbol{i}_{ext}$ corresponds to a flow of one unit from vertex a to vertex b. The resulting voltages are

$$\boldsymbol{v} = \boldsymbol{L}^{+} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b).$$

Now, let c and d be two other vertices. The potential difference between c and d is

$$\boldsymbol{v}(c) - \boldsymbol{v}(d) = (\boldsymbol{\delta}_c - \boldsymbol{\delta}_d)^T \boldsymbol{v} = (\boldsymbol{\delta}_c - \boldsymbol{\delta}_d)^T \boldsymbol{L}^{+} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b).$$

Note the amazing reciprocity here: as $\boldsymbol{L}$ is symmetric this is equal to

$$(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}^{+} (\boldsymbol{\delta}_c - \boldsymbol{\delta}_d).$$

So, the potential difference between c and d when we flow one unit from a to b is the same as the potential difference between a and b when we flow one unit from c to d.

The *effective resistance* between vertices a and b is the resistance between a and b when we view the entire network as one complex resistor.

To figure out what this is, recall the equation

$$\boldsymbol{i}(a,b) = \frac{\boldsymbol{v}(a) - \boldsymbol{v}(b)}{r_{a,b}},$$

which holds for one resistor. We use the same equation to define the effective resistance of the whole network between a and b. That is, we consider an electrical flow that sends one unit of current into node a and removes one unit of current from node b. We then measure the potential difference between a and b that is required to realize this current, define this to be the effective resistance between a and b, and write it $\mathrm{R_{eff}}(a,b)$. As it equals the potential difference between a and b in a flow of one unit of current from a to b:

$$\mathrm{R_{eff}}(a,b) \overset{\mathrm{def}}{=} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}^{+} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b).$$

We will eventually show that effective resistance is a distance. For now, we observe that effective resistance is the square of a Euclidean distance.

To this end, let $\boldsymbol{L}^{+/2}$ denote the square root of $\boldsymbol{L}^{+}$. Recall that every positive semidefinite matrix has a square root: the square root of a symmetric matrix $\boldsymbol{M}$ is the symmetric matrix $\boldsymbol{M}^{1/2}$ such that $(\boldsymbol{M}^{1/2})^2 = \boldsymbol{M}$. If

$$\boldsymbol{M} = \sum_i \lambda_i \boldsymbol{\psi}_i \boldsymbol{\psi}^T$$

is the spectral decomposition of $\boldsymbol{M}$, then

$$\boldsymbol{M}^{1/2} = \sum_i \lambda_i^{1/2} \boldsymbol{\psi}_i \boldsymbol{\psi}^T.$$

We now have

$$\begin{aligned}(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}^{+} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b) = \left(\boldsymbol{L}^{+/2}(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)\right)^T \boldsymbol{L}^{+/2}(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b) &= \left\|\boldsymbol{L}^{+/2}(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)\right\|^2 \\ &= \left\|\boldsymbol{L}^{+/2}\boldsymbol{\delta}_a - \boldsymbol{L}^{+/2}\boldsymbol{\delta}_b\right\|^2 = \mathrm{dist}(\boldsymbol{L}^{+/2}\boldsymbol{\delta}_a, \boldsymbol{L}^{+/2}\boldsymbol{\delta}_b)^2.\end{aligned}$$

12.2 Effective Resistance through Energy Minimization

As you would imagine, we can also define the effective resistance through effective spring constants. In this case, we view the network of springs as one large compound network. If we define the effective spring constant of s, t to be the number w so that when s and t are stretched to distance l the potential energy in the spring is $wl^2/2$, then we should define the effective spring constant to be twice the minimum possible energy of the network,

$$2\mathcal{E}(\boldsymbol{x}) = \sum_{(a,b)\in E} w_{a,b}(\boldsymbol{x}(a) - \boldsymbol{x}(b))^2,$$

when $\boldsymbol{x}(s)$ is fixed to 0 and $\boldsymbol{x}(t)$ is fixed to 1. From Theorem 11.6.1, we know that this vector will be harmonic on $V - \{s,t\}$.

Fortunately, we already know how compute such a vector $\boldsymbol{x}$. Set

$$\boldsymbol{y} = \boldsymbol{L}^{+}(\boldsymbol{\delta}_t - \boldsymbol{\delta}_s)/\mathrm{R}_{\mathrm{eff}}(t,s).$$

We have

$$\boldsymbol{y}(t) - \boldsymbol{y}(s) = (\boldsymbol{\delta}_t - \boldsymbol{\delta}_s)^T \boldsymbol{L}^{+}(\boldsymbol{\delta}_t - \boldsymbol{\delta}_s)/\mathrm{R}_{\mathrm{eff}}(s,t) = 1,$$

and $\boldsymbol{y}$ is harmonic on $V - \{s,t\}$. So, we choose

$$\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{1}\boldsymbol{y}(s).$$

The vector $\boldsymbol{x}$ satisfies $\boldsymbol{x}(s) = 0$, $\boldsymbol{x}(t) = 1$, and it is harmonic on $V - \{s,t\}$. So, it is the vector that minimizes the energy subject to the boundary conditions.

To finish, we compute the energy to be

$$\begin{aligned}
\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} &= \boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y} \\
&= \frac{1}{(\mathrm{R}_{\mathrm{eff}}(s,t))^2} \left(\boldsymbol{L}^{+}(\boldsymbol{\delta}_t - \boldsymbol{\delta}_s)\right)^T \boldsymbol{L} \left(\boldsymbol{L}^{+}(\boldsymbol{\delta}_t - \boldsymbol{\delta}_s)\right) \\
&= \frac{1}{(\mathrm{R}_{\mathrm{eff}}(s,t))^2} (\boldsymbol{\delta}_t - \boldsymbol{\delta}_s)^T \boldsymbol{L}^{+} \boldsymbol{L} \boldsymbol{L}^{+} (\boldsymbol{\delta}_t - \boldsymbol{\delta}_s) \\
&= \frac{1}{(\mathrm{R}_{\mathrm{eff}}(s,t))^2} (\boldsymbol{\delta}_t - \boldsymbol{\delta}_s)^T \boldsymbol{L}^{+} (\boldsymbol{\delta}_t - \boldsymbol{\delta}_s) \\
&= \frac{1}{\mathrm{R}_{\mathrm{eff}}(s,t)}.
\end{aligned}$$

As the weights of edges are the reciprocals of their resistances, and the spring constant corresponds to the weight, this is the formula we would expect.

Resistor networks have an analogous quantity: the energy dissipation (into heat) when current flows through the network. It has the same formula. The reciprocal of the effective resistance is sometimes called the effective conductance.

12.3 Monotonicity

Rayleigh's Monotonicity Principle tells us that if we alter the spring network by decreasing some of the spring constants, then the effective spring constant between s and t will not increase. In terms of effective resistance, this says that if we increase the resistance of some resistors then the effective resistance can not decrease. This sounds obvious. But, it is in fact a very special property of linear elements like springs and resistors.

Theorem 12.3.1. *Let $G = (V, E, w)$ be a weighted graph and let $\widehat{G} = (V, E, \widehat{w})$ be another weighted graph with the same edges and such that*

$$\widehat{w}_{a,b} \leq w_{a,b}$$

for all $(a,b) \in E$. For vertices s and t, let $c_{s,t}$ be the effective spring constant between s and t in G and let $\widehat{c}_{s,t}$ be the analogous quantity in $\widehat{G}$. Then,

$$\widehat{c}_{s,t} \leq c_{s,t}.$$

Proof. Let $\boldsymbol{x}$ be the vector of minimum energy in G such that $\boldsymbol{x}(s) = 0$ and $\boldsymbol{x}(t) = 1$. Then, the energy of $\boldsymbol{x}$ in $\widehat{G}$ is no greater:

$$\frac{1}{2} \sum_{(a,b) \in E} \widehat{w}_{a,b} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2 \leq \frac{1}{2} \sum_{(a,b) \in E} w_{a,b} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2 = c_{s,t}.$$

So, the minimum energy of a vector $\boldsymbol{x}$ in $\widehat{G}$ such that $\boldsymbol{x}(s) = 0$ and $\boldsymbol{x}(t) = 1$ will be at most $c_{s,t}$, and so $\widehat{c}_{s,t} \leq c_{s,t}$. □

Similarly, if we let $\widehat{\mathrm{R}}_{\mathrm{eff}}(s,t)$ be the effective resistance in G between s and t, then $\widehat{\mathrm{R}}_{\mathrm{eff}}(s,t) \geq \mathrm{R}_{\mathrm{eff}}(s,t)$. That is, increasing the resistance of resistors in the network cannot decrease effective resistances.

While this principle seems very simple and intuitively obvious, it turns out to fail in just slightly more complicated situations.

12.4 Examples: Series and Parallel

In the case of a path graph with n vertices and edges of weight 1, the effective resistance between the extreme vertices is $n-1$.

In general, if a path consists of edges of resistance $r_{1,2}, \ldots, r_{n-1,n}$ then the effective resistance between the extreme vertices is

$$r_{1,2} + \cdots + r_{n-1,n}.$$

To see this, set the potential of vertex i to

$$\boldsymbol{v}(i) = r_{i,i+1} + \cdots + r_{n-1,n}.$$

Ohm's law then tells us that the current flow over the edge $(i, i+1)$ will be

$$\left(\boldsymbol{v}(i) - \boldsymbol{v}(i+1)\right) / r_{i,i+1} = 1.$$

If we have k parallel edges between two nodes s and t of resistances $r_1, \ldots, r_k$, then the effective resistance is

$$\mathrm{R}_{\mathrm{eff}}(s,t) = \frac{1}{1/r_1 + \cdots + 1/r_k}.$$

To see this, impose a potential difference of 1 between s and t. This will induce a flow of $1/r_i = w_i$ on edge i. So, the total flow will be

$$\sum_i 1/r_i = \sum_i w_i.$$

12.5 Equivalent Networks, Elimination, and Schur Complements

We have shown that the impact of the entire network on two vertices can be reduced to a network with one edge between them. We will now see that we can do the same for a subset of the vertices. We will do this in two ways: first by viewing $\boldsymbol{L}$ as an operator, and then by considering it as a quadratic form.

Let B be the subset of nodes that we would like to understand (B stands for *boundary*). All nodes not in B will be internal. Call them $I = V - B$.

As an operator, the Laplacian maps vectors of voltages to vectors of external currents. We want to examine what happens if we fix the voltages at vertices in B, and require the rest to be harmonic. Let $\boldsymbol{v}(B) \in \mathbb{R}^B$ be the voltages at B. We want the matrix $\boldsymbol{L}_B$ such that

$$\boldsymbol{i}_B = \boldsymbol{L}_B \boldsymbol{v}(B)$$

is the vector of external currents a vertices in B when we impose voltages $\boldsymbol{v}(B)$ at vertices of B. As the internal vertices will have their voltages set to be harmonic, they will not have any external currents.

The remarkable fact that we will discover is that $\boldsymbol{L}_B$ is in fact a Laplacian matrix, and that it is obtained by performing Gaussian elimination to remove the internal vertices. **Warning:** $\boldsymbol{L}_B$ is not a submatrix of $\boldsymbol{L}$. To prove this, we will move from V to B by removing one vertex at a time. We'll start with a graph $G = (V, E, w)$, and we will set $B = \{2, \ldots, n\}$, and we will treat vertex 1 as internal. Let N denote the set of neighbors of vertex 1.

We want to compute $\boldsymbol{L}\boldsymbol{v}$ given $\boldsymbol{v}(b)$ for $b \in B$, and that

$$\boldsymbol{v}(1) = \frac{1}{\boldsymbol{d}(1)} \sum_{a \in N} w_{1,a} \boldsymbol{v}(a). \tag{12.1}$$

That is, we want to substitute the value on the right-hand side for $\boldsymbol{v}(1)$ everywhere that it appears in the equation $\boldsymbol{i}_{ext} = \boldsymbol{L}\boldsymbol{v}$. The variable $\boldsymbol{v}(1)$ only appears in the equation for $\boldsymbol{i}_{ext}(a)$ when $a \in N$. When it does, it appears with coefficient $w_{1,a}$. Recall that the equation for $\boldsymbol{i}_{ext}(b)$ is

$$\boldsymbol{i}_{ext}(b) = \boldsymbol{d}(b)\boldsymbol{v}(b) - \sum_{c \sim b} w_{b,c} \boldsymbol{v}(c).$$

For $b \in N$ we expand this by making the substitution for $\boldsymbol{v}(1)$ given by (12.1).

$$\begin{aligned}
\boldsymbol{i}_{ext}(b) &= \boldsymbol{d}(b)\boldsymbol{v}(b) - w_{b,1}\boldsymbol{v}(1) - \sum_{c \sim b, c \neq 1} w_{b,c} \boldsymbol{v}(c) \\
&= \boldsymbol{d}(b)\boldsymbol{v}(b) - w_{b,1} \frac{1}{\boldsymbol{d}(1)} \sum_{a \in N} w_{1,a} \boldsymbol{v}(a) - \sum_{c \sim b, c \neq 1} w_{b,c} \boldsymbol{v}(c) \\
&= \boldsymbol{d}(b)\boldsymbol{v}(b) - \sum_{a \in N} \frac{w_{b,1} w_{a,1}}{\boldsymbol{d}(1)} \boldsymbol{v}(a) - \sum_{c \sim b, c \neq 1} w_{b,c} \boldsymbol{v}(c).
\end{aligned}$$

To finish, observe that $b \in N$, so we are counting b in the middle sum above. Removing the double-count gives.

$$\boldsymbol{i}_{ext}(b) = (\boldsymbol{d}(b) - w_{b,1}^2/\boldsymbol{d}(1))\boldsymbol{v}(b) - \sum_{a \in N, a \neq b} \frac{w_{b,1} w_{a,1}}{\boldsymbol{d}(1)} \boldsymbol{v}(a) - \sum_{c \sim b, c \neq 1} w_{b,c} \boldsymbol{v}(c).$$

We will show that these revised equations have two interesting properties: they are the result of applying Gaussian elimination to eliminate vertex 1, and the resulting equations are Laplacian.

Let's look at exactly how the matrix has changed. In the row for vertex b, the edge to vertex 1 was removed, and edges to every vertex $a \in N$ were added with weights $\frac{w_{b,1} w_{a,1}}{\boldsymbol{d}(1)}$. And, the diagonal was decreased by $\frac{w_{b,1} w_{b,1}}{\boldsymbol{d}(1)}$. Overall, the star of edges based at 1 were removed, and a clique on N was added in which edge (a, b) has weight

$$\frac{w_{b,1} w_{1,a}}{\boldsymbol{d}(1)}.$$

To see that this new system of equations comes from a Laplacian, we observe that

1. It is symmetric.
2. The off-diagonal entries that have been added are negative.
3. The sum of the changes in diagonal and off-diagonal entries is zero, so the row-sum is still zero. This follows from
$$\frac{w_{b,1}^2}{\boldsymbol{d}(1)} - \sum_{a \in N} \frac{w_{b,1} w_{a,1}}{\boldsymbol{d}(1)} = 0.$$

12.5.1 In matrix form by energy

We now do this in terms of the quadratic form. That is, we will compute the matrix $\boldsymbol{L}_B$ so that

$$\boldsymbol{v}(B)^T \boldsymbol{L}_B \boldsymbol{v}(B) = \boldsymbol{v}^T \boldsymbol{L} \boldsymbol{v},$$

given that $\boldsymbol{v}$ is harmonic at vertex 1 and agrees with $\boldsymbol{v}(B)$ elsewhere. The quadratic form that we want to compute is thus given by

$$\begin{pmatrix} \frac{1}{\boldsymbol{d}(1)} \sum_{b \sim 1} w_{1,b} \boldsymbol{v}(b) \\ \boldsymbol{v}(B) \end{pmatrix}^T \boldsymbol{L} \begin{pmatrix} \frac{1}{\boldsymbol{d}(1)} \sum_{b \sim 1} w_{1,b} \boldsymbol{v}(b) \\ \boldsymbol{v}(B) \end{pmatrix}.$$

So that we can write this in terms of the entries of the Laplacian matrix, note that $\boldsymbol{d}(1) = \boldsymbol{L}(1,1)$, and so

$$\boldsymbol{v}(1) = \frac{1}{\boldsymbol{d}(1)} \sum_{b \sim 1} w_{1,b} \boldsymbol{v}(b) = -(1/\boldsymbol{L}(1,1)) \boldsymbol{L}(1, B) \boldsymbol{v}(B).$$

Thus, we can write the quadratic form as

$$\begin{pmatrix} -(1/\boldsymbol{L}(1,1))\boldsymbol{L}(1,B)\boldsymbol{v}(B) \\ \boldsymbol{v}(B) \end{pmatrix}^T \boldsymbol{L} \begin{pmatrix} -(1/\boldsymbol{L}(1,1))\boldsymbol{L}(1,B)\boldsymbol{v}(B) \\ \boldsymbol{v}(B) \end{pmatrix}.$$

If we expand this out, we find that it equals

$$\begin{aligned}
&\boldsymbol{v}(B)^T\boldsymbol{L}(B,B)\boldsymbol{v}(B) + \boldsymbol{L}(1,1)\left(-(1/\boldsymbol{L}(1,1))\boldsymbol{L}(1,B)\boldsymbol{v}(B)\right)^2 + 2\boldsymbol{v}(1)\boldsymbol{L}(1,B)\left(-(1/\boldsymbol{L}(1,1))\boldsymbol{L}(1,B)\boldsymbol{v}(B)\right) \\
&= \boldsymbol{v}(B)^T\boldsymbol{L}(B,B)\boldsymbol{v}(B) + \left(\boldsymbol{L}(1,B)\boldsymbol{v}(B)\right)^2/\boldsymbol{L}(1,1) - 2\left(\boldsymbol{L}(1,B)\boldsymbol{v}(B)\right)^2/\boldsymbol{L}(1,1) \\
&= \boldsymbol{v}(B)^T\boldsymbol{L}(B,B)\boldsymbol{v}(B) - \left(\boldsymbol{L}(1,B)\boldsymbol{v}(B)\right)^2/\boldsymbol{L}(1,1).
\end{aligned}$$

Thus,

$$\boldsymbol{L}_B = \boldsymbol{L}(B,B) - \frac{\boldsymbol{L}(B,1)\boldsymbol{L}(1,B)}{\boldsymbol{L}(1,1)}.$$

To see that this is the matrix that appears in rows and columns 2 through n when we eliminate the entries in the first column of $\boldsymbol{L}$ by adding multiples of the first row, note that we eliminate entry $\boldsymbol{L}(a,1)$ by adding $-\boldsymbol{L}(a,1)/\boldsymbol{L}(1,1)$ times the first row of the matrix to $\boldsymbol{L}(a,:)$. Doing this for all rows in $B = \{2,\dots,n\}$ results in this formula.

We can again check that $\boldsymbol{L}_B$ is a Laplacian matrix. It is clear from the formula that it is symmetric and that the off-diagonal entries are negative. To check that the constant vectors are in the nullspace, we can show that the quadratic form is zero on those vectors. If $\boldsymbol{v}(B)$ is a constant vector, then $\boldsymbol{v}(1)$ must equal this constant, and so $\boldsymbol{v}$ is a constant vector and the value of the quadratic form is 0.

12.6 Eliminating Many Vertices

We can of course use the same procedure to eliminate many vertices. We begin by partitioning the vertex set into *boundary* vertices B and *internal* vertices I. We can then use Gaussian elimination to eliminate all of the internal vertices. You should recall that the submatrices produced by Gaussian elimination do not depend on the order of the eliminations. So, you may conclude that the matrix $\boldsymbol{L}_B$ is uniquely defined.

Or, observe that to eliminate the entries in row $a \in B$ and columns in S, using the rows in S, we need to add those rows, $\boldsymbol{L}(S,:)$ to row $\boldsymbol{L}(a,:)$ with coefficients $\boldsymbol{c}$ so that

$$\boldsymbol{L}(a,S) + \boldsymbol{c}\boldsymbol{L}(S,S) = 0.$$

This gives

$$\boldsymbol{c} = -\boldsymbol{L}(a,S)\boldsymbol{L}(S,S)^{-1},$$

and thus row a becomes

$$\boldsymbol{L}(a,:) - \boldsymbol{L}(a,S)\boldsymbol{L}(S,S)^{-1}\boldsymbol{L}(S,:).$$

Restricting to rows and columns in B, we are left with the matrix

$$\boldsymbol{L}(B,B) - \boldsymbol{L}(B,S)\boldsymbol{L}(S,S)^{-1}\boldsymbol{L}(S,B).$$

This is called the *Schur* complement on B (or with respect to S).

To see that this is equivalent to requiring that the variables in S be harmonic. Partition a vector $\boldsymbol{v}$ into $\boldsymbol{v}(B)$ and $\boldsymbol{v}(S)$. The harmonic equations become

$$\boldsymbol{L}(S,S)\boldsymbol{v}(S) + \boldsymbol{L}(S,B)\boldsymbol{v}(B) = 0,$$

which implies

$$\boldsymbol{v}(S) = -\boldsymbol{L}(S,S)^{-1}\boldsymbol{L}(S,B)\boldsymbol{v}(B) = \boldsymbol{L}(S,S)^{-1}\boldsymbol{M}(S,B)\boldsymbol{v}(B),$$

as $\boldsymbol{M}(S,B) = -\boldsymbol{L}(S,B)$ because off-diagonal blocks of the Laplacian equal the negative of the corresponding blocks in the adjacency matrix. This gives

$$\boldsymbol{i}_{ext}(B) = \boldsymbol{L}(B,S)\boldsymbol{v}(S) + \boldsymbol{L}(B,B)\boldsymbol{v}(B) = -\boldsymbol{L}(B,S)\boldsymbol{L}(S,S)^{-1}\boldsymbol{L}(S,B)\boldsymbol{v}(B) + \boldsymbol{L}(B,B)\boldsymbol{v}(B),$$

and so

$$\boldsymbol{i}_{ext}(B) = \boldsymbol{L}_B\boldsymbol{v}(B), \quad \text{where } \boldsymbol{L}_B = \boldsymbol{L}(B,B) - \boldsymbol{L}(B,S)\boldsymbol{L}(S,S)^{-1}\boldsymbol{L}(S,B)$$

is the Schur complement.

12.7 An interpretation of Gaussian elimination

This gives us a way of understand how Gaussian elimination solves a system of equations like $\boldsymbol{i}_{ext} = \boldsymbol{L}\boldsymbol{v}$. It constructs a sequence of graphs, $G_2, \dots, G_n$, so that G_i is the effective network on vertices $i, \dots, n$. It then solves for the entries of $\boldsymbol{v}$ backwards. Given $\boldsymbol{v}(i+1), \dots, \boldsymbol{v}(n)$ and $\boldsymbol{i}_{ext}(i)$, we can solve for $\boldsymbol{v}(i)$. If $\boldsymbol{i}_{ext}(i) = 0$, then $\boldsymbol{v}(i)$ is set to the weighted average of its neighbors. If not, then we need to take $\boldsymbol{i}_{ext}(i)$ into account here and in the elimination as well. In the case in which we fix some vertices and let the rest be harmonic, there is no such complication.

12.8 Effective Resistance is a Distance

A distance is any function on pairs of vertices such that

1. $\delta(a,a) = 0$ for every vertex a,
2. $\delta(a,b) \geq 0$ for all vertices a, b,
3. $\delta(a,b) = \delta(b,a)$, and
4. $\delta(a,c) \leq \delta(a,b) + \delta(b,c)$.

We claim that the effective resistance is a distance. The only non-trivial part to prove is the triangle inequality, (4).

From the previous section, we know that it suffices to consider graphs with only three vertices: we can reduce any graph to one on just vertices a, b and c without changing the effective resistances between them.

Lemma 12.8.1. *Let a, b and c be vertices in a graph. Then*

$$\mathrm{R_{eff}}(a,b) + \mathrm{R_{eff}}(b,c) \geq \mathrm{R_{eff}}(a,c).$$

Proof. Let

$$z = w_{a,b}, y = w_{a,c}, \quad \text{and} \quad x = w_{b,c}.$$

If we eliminate vertex c, we create an edge between vertices a and b of weight

$$\frac{xy}{x+y}.$$

Adding this to the edge that is already there produces weight $z + \frac{xy}{x+y}$, for

$$\mathrm{R_{eff}}_{a,b} = \frac{1}{z + \frac{xy}{x+y}} = \frac{1}{\frac{zx+zy+xy}{x+y}} = \frac{x+y}{zx+zy+xy}$$

Working symmetrically, we find that we need to prove that for all positive x, y, and z

$$\frac{x+y}{zx+zy+xy} + \frac{y+z}{zx+zy+xy} \geq \frac{x+z}{zx+zy+xy},$$

which is of course true.

□

Chapter 13

Random Spanning Trees

13.1 Introduction

In this chapter we present one of the most fundamental results in Spectral Graph Theory: the Matrix-Three Theorem. It relates the number of spanning trees of a connected graph to the determinants of principal minors of the Laplacian. We then extend this result to relate the fraction of spanning trees that contain a given edge to the effective resistance of the entire graph between the edge's endpoints.

13.2 Determinants

To begin, we review some facts about determinants of matrices and characteristic polynomials. We first recall the Leibniz formula for the determinant of a square matrix $\boldsymbol{A}$:

$$\det(\boldsymbol{A}) = \sum_{\pi} \left(\mathsf{sgn}(\pi) \prod_{i=1}^{n} \boldsymbol{A}(i, \pi(i)) \right), \tag{13.1}$$

where the sum is over all permutations π of $\{1, \ldots, n\}$.

Also recall that the determinant is multiplicative, so for square matrices $\boldsymbol{A}$ and $\boldsymbol{B}$

$$\det(\boldsymbol{A}\boldsymbol{B}) = \det(\boldsymbol{A}) \det(\boldsymbol{B}). \tag{13.2}$$

Elementary row operations do not change the determinant. If the columns of $\boldsymbol{A}$ are the vectors $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_n$, then for every c

$$\det \big(\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n\big) = \det \big(\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_n + c\boldsymbol{a}_1\big) .$$

This fact gives us two ways of computing the determinant. The first comes from the fact that we can apply elementary row operations to transform $\boldsymbol{A}$ into an upper triangular matrix, and (13.1) tells us that the determinant of an upper triangular matrix is the product of its diagonal entries.

The second comes from the observation that the determinant is the volume of the parallelepiped with axes $\boldsymbol{a}_1, \dots, \boldsymbol{a}_n$: the polytope whose corners are the origin and $\sum_{i \in S} \boldsymbol{a}_i$ for every $S \subseteq \{1, \dots, n\}$. Let

$$\boldsymbol{\Pi}_{\boldsymbol{a}_1}$$

be the symmetric projection orthogonal to $\boldsymbol{a}_1$. As this projection amounts to subtracting off a multiple of $\boldsymbol{a}_1$ and elementary row operations do not change the determinant,

$$\det\left(\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_n\right) = \det\left(\boldsymbol{a}_1, \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_2, \dots, \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_n\right).$$

The volume of this parallelepiped is $\|\boldsymbol{a}_1\|$ times the volume of the parallelepiped formed by the vectors $\boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_2, \dots, \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_n$. I would like to write this as a determinant, but must first deal with the fact that these are $n-1$ vectors in an n dimensional space. The way we first learn to handle this is to project them into an $n-1$ dimensional space where we can take the determinant. Instead, we will employ other elementary symmetric functions of the eigenvalues.

13.3 Characteristic Polynomials

Recall that the characteristic polynomial of a matrix $\boldsymbol{A}$ is

$$\det(x\boldsymbol{I} - \boldsymbol{A}).$$

I will write this as

$$\sum_{k=0}^{n} x^{n-k}(-1)^k \sigma_k(\boldsymbol{A}),$$

where $\sigma_k(\boldsymbol{A})$ is the kth elementary symmetric function of the eigenvalues of $\boldsymbol{A}$, counted with algebraic multiplicity:

$$\sigma_k(\boldsymbol{A}) = \sum_{|S|=k} \prod_{i \in S} \lambda_i.$$

Thus, $\sigma_1(\boldsymbol{A})$ is the trace and $\sigma_n(\boldsymbol{A})$ is the determinant. From this formula, we know that these functions are invariant under similarity transformations.

In Exercise 3 from Lecture 2, you were asked to prove that

$$\sigma_k(\boldsymbol{A}) = \sum_{|S|=k} \det(\boldsymbol{A}(S,S)). \tag{13.3}$$

This follows from applying the Leibnitz formula (13.1) to $\det(x\boldsymbol{I} - \boldsymbol{A})$.

If we return to the vectors $\boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_2, \dots, \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_n$ from the previous section, we see that the volume of their parallelepiped may be written

$$\sigma_{n-1}\left(\boldsymbol{0}_n, \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_2, \dots, \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_n\right),$$

as this will be the product of the $n-1$ nonzero eigenvalues of this matrix.

Recall that the matrices $\boldsymbol{B}\boldsymbol{B}^T$ and $\boldsymbol{B}^T\boldsymbol{B}$ have the same eigenvalues, up to some zero eigenvalues if they are rectangular. So,

$$\sigma_k(\boldsymbol{B}\boldsymbol{B}^T) = \sigma_k(\boldsymbol{B}^T\boldsymbol{B}).$$

This gives us one other way of computing the absolute value of the product of the nonzero eigenvalues of the matrix

$$\left(\boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_2, \ldots, \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_n\right).$$

We can instead compute their square by computing the determinant of the square matrix

$$\begin{pmatrix} \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_2 \\ \vdots \\ \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_n \end{pmatrix} \left(\boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_2, \ldots, \boldsymbol{\Pi}_{\boldsymbol{a}_1}\boldsymbol{a}_n\right).$$

When $\boldsymbol{B}$ is a singular matrix of rank k, $\sigma_k(\boldsymbol{B})$ acts as the determinant of $\boldsymbol{B}$ restricted to its span. Thus, there are situations in which σ_k is multiplicative. For example, if $\boldsymbol{A}$ and $\boldsymbol{B}$ both have rank k and the range of $\boldsymbol{A}$ is orthogonal to the nullspace of $\boldsymbol{B}$, then

$$\sigma_k(\boldsymbol{B}\boldsymbol{A}) = \sigma_k(\boldsymbol{B})\sigma_k(\boldsymbol{A}). \tag{13.4}$$

We will use this identity in the case that $\boldsymbol{A}$ and $\boldsymbol{B}$ are symmetric and have the same nullspace.

13.4 The Matrix Tree Theorem

We will state a slight variant of the standard Matrix-Tree Theorem. Recall that a spanning tree of a graph is a subgraph that is a tree.

Theorem 13.4.1. *Let $G = (V, E, w)$ be a connected, weighted graph. Then*

$$\sigma_{n-1}(\boldsymbol{L}_G) = n \sum_{\text{spanning trees } T} \prod_{e \in T} w_e.$$

Thus, the eigenvalues allow us to count the sum over spanning trees of the product of the weights of edges in those trees. When all the edge weights are 1, we just count the number of spanning trees in G.

We first prove this in the case that G is just a tree.

Lemma 13.4.2. *Let $G = (V, E, w)$ be a weighted tree. Then,*

$$\sigma_{n-1}(\boldsymbol{L}_G) = n \prod_{e \in E} w_e.$$

Proof. For $a \in V$, let $S_a = V - \{a\}$. We know from (13.3)

$$\sigma_{n-1}(\boldsymbol{L}_G) = \sum_{a \in V} \det(\boldsymbol{L}_G(S_a, S_a).$$

We will prove that for every $a \in V$,

$$\det(\boldsymbol{L}_G(S_a, S_a)) = \prod_{e \in E} w_e.$$

Write $\boldsymbol{L}_G = \boldsymbol{U}^T \boldsymbol{W} \boldsymbol{U}$, where $\boldsymbol{U}$ is the signed edge-vertex adjacency matrix and $\boldsymbol{W}$ is the diagonal matrix of edge weights. Write $\boldsymbol{B} = \boldsymbol{W}^{1/2} \boldsymbol{U}$, so

$$\boldsymbol{L}_G(S_a, S_a) = \boldsymbol{B}(:, S_a)^T \boldsymbol{B}(:, S_a),$$

and

$$\det(\boldsymbol{L}_G(S_a, S_a)) = \det(\boldsymbol{B}(:, S_a))^2,$$

where we note that $\boldsymbol{B}(:, S_a)$ is square because a tree has $n-1$ edges and so $\boldsymbol{B}$ has $n-1$ rows.

To see what is going on, first consider the case in which G is a weighted path and a is the first vertex. Then,

$$\boldsymbol{U} = \begin{pmatrix} 1 & -1 & 0 & \cdots & 0 \\ 0 & 1 & -1 & \cdots & 0 \\ \vdots & & & & \vdots \\ 0 & 0 & 0 & \cdots & -1 \end{pmatrix}, \quad \text{and} \quad \boldsymbol{B}(:, S_1) = \begin{pmatrix} -\sqrt{w_1} & 0 & \cdots & 0 \\ \sqrt{w_2} & -\sqrt{w_2} & \cdots & 0 \\ \vdots & & & \vdots \\ 0 & 0 & \cdots & -\sqrt{w_{n-1}} \end{pmatrix}.$$

We see that $\boldsymbol{B}(:, S_1)$ is a lower-triangular matrix, and thus its determinant is the product of its diagonal entries, $-\sqrt{w_i}$.

To see that the same happens for every tree, renumber the vertices (permute the columns) so that a comes first, and that the other vertices are ordered by increasing distance from 1, breaking ties arbitrarily. This permutations can change the sign of the determinant, but we do not care because we are going to square it. For every vertex $c \neq 1$, the tree now has exactly one edge (b, c) with $b < c$. Put such an edge in position $c-1$ in the ordering, and let w_c indicate its weight. Now, when we remove the first column to form $\boldsymbol{B}(:, S_1)$, we produce a lower triangular matrix with the entry $-\sqrt{w_c}$ on the cth diagonal. So, its determinant is the product of these terms and

$$\det(\boldsymbol{B}(:, S_a))^2 = \prod_{c=2}^{n} w_c.$$

□

Proof of Theorem 13.4.1 . As in the previous lemma, let $\boldsymbol{L}_G = \boldsymbol{U}^T \boldsymbol{W} \boldsymbol{U}$ and $\boldsymbol{B} = \boldsymbol{W}^{1/2} \boldsymbol{U}$. So,

$$\begin{aligned} \sigma_{n-1}(\boldsymbol{L}_G) &= \sigma_{n-1}(\boldsymbol{B}^T \boldsymbol{B}) \\ &= \sigma_{n-1}(\boldsymbol{B} \boldsymbol{B}^T) \\ &= \sum_{|S|=n-1, S \subseteq E} \sigma_{n-1}(\boldsymbol{B}(S,:) \boldsymbol{B}(S,:)^T) \quad \text{(by (13.3))} \\ &= \sum_{|S|=n-1, S \subseteq E} \sigma_{n-1}(\boldsymbol{B}(S,:)^T \boldsymbol{B}(S,:)) \\ &= \sum_{|S|=n-1, S \subseteq E} \sigma_{n-1}(\boldsymbol{L}_{G_S}), \end{aligned}$$

where by G_S we mean the graph containing just the edges in S. As S contains $n-1$ edges, this graph is either disconnected or a tree. If it is disconnected, then its Laplacian has at least two zero eigenvalues and $\sigma_{n-1}(\boldsymbol{L}_{G_S}) = 0$. If it is a tree, we apply the previous lemma. Thus, the sum equals

$$\sum_{\text{spanning trees } T \subseteq E} \sigma_{n-1}(\boldsymbol{L}_{G_T}) = n \sum_{\text{spanning trees } T} \prod_{e \in T} w_e.$$

□

13.5 Leverage Scores and Marginal Probabilities

The *leverage score* of an edge, written ℓ_e is defined to be $w_e \mathrm{R_{eff}}(e)$. That is, the weight of the edge times the effective resistance between its endpoints. The leverage score serves as a measure of how important the edge is. For example, if removing an edge disconnects the graph, then $\mathrm{R_{eff}}(e) = 1/w_e$, as all current flowing between its endpoints must use the edge itself, and $\ell_e = 1$.

Consider sampling a random spanning tree with probability proportional to the product of the weights of its edges. We will now show that the probability that edge e appears in the tree is exactly its leverage score.

Theorem 13.5.1. *If we choose a spanning tree T with probability proportional to the product of its edge weights, then for every edge e*

$$Pr\left[e \in T\right] = \ell_e.$$

For simplicity, you might want to begin by thinking about the case where all edges have weight 1.

Recall that the effective resistance of edge $e = (a, b)$ is

$$(\delta_a - \delta_b)^T \boldsymbol{L}_G^+ (\delta_a - \delta_b),$$

and so

$$\ell_{a,b} = w_{a,b} (\delta_a - \delta_b)^T \boldsymbol{L}_G^+ (\delta_a - \delta_b).$$

We can write a matrix $\boldsymbol{\Gamma}$ that has all these terms on its diagonal by letting $\boldsymbol{U}$ be the edge-vertex adjacency matrix, $\boldsymbol{W}$ be the diagonal edge weight matrix, $\boldsymbol{B} = \boldsymbol{W}^{1/2}\boldsymbol{U}$, and setting

$$\boldsymbol{\Gamma} = \boldsymbol{B}\boldsymbol{L}_G^+\boldsymbol{B}^T.$$

The rows and columns of $\boldsymbol{\Gamma}$ are indexed by edges, and for each edge e,

$$\boldsymbol{\Gamma}(e, e) = \ell_e.$$

For off-diagonal entries corresponding to edges (a, b) and (c, d), we have

$$\boldsymbol{\Gamma}((a, b), (c, d)) = \sqrt{w_{a,b}}\sqrt{w_{c,d}}(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}_G^+ (\boldsymbol{\delta}_c - \boldsymbol{\delta}_d).$$

Claim 13.5.2. *The matrix $\boldsymbol{\Gamma}$ is a symmetric projection matrix and has trace $n-1$.*

Proof. The matrix $\boldsymbol{\Gamma}$ is clearly symmetric. To show that it is a projection, it suffices to show that all of its eigenvalues are 0 or 1. This is true because, excluding the zero eigenvalues, $\boldsymbol{\Gamma}$ has the same eigenvalues as

$$\boldsymbol{L}_G^{+}\boldsymbol{B}^T\boldsymbol{B} = \boldsymbol{L}_G^{+}\boldsymbol{L}_G = \boldsymbol{\Pi},$$

where $\boldsymbol{\Pi}$ is the projection orthogonal to the all 1 vector. As $\boldsymbol{\Pi}$ has $n-1$ eigenvalues that are 1, so does $\boldsymbol{\Gamma}$. □

As the trace of $\boldsymbol{\Gamma}$ is $n-1$, so is the sum of the leverage scores:

$$\sum_e \ell_e = n-1.$$

This is a good sanity check on Theorem 13.5.1: every spanning tree has $n-1$ edges, and thus the probabilities that each edge is in the tree must sum to $n-1$.

We also obtain another formula for the leverage score. As a symmetric projection is its own square,

$$\boldsymbol{\Gamma}(e,e) = \boldsymbol{\Gamma}(e,:)\boldsymbol{\Gamma}(e,:)^T = \left\| \boldsymbol{\Gamma}(e,:) \right\|^2 .$$

This is the formula I introduced in Section **??**. If we flow 1 unit from a to b, the potential difference between c and d is $(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}_G^{+}(\boldsymbol{\delta}_c - \boldsymbol{\delta}_d)$. If we plug these potentials into the Laplacian quadratic form, we obtain the effective resistance. Thus this formula says

$$w_{a,b}\mathrm{R_{eff}}_{a,b} = w_{a,b} \sum_{(c,d)\in E} w_{c,d}\left((\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}_G^{+}(\boldsymbol{\delta}_c - \boldsymbol{\delta}_d)\right)^2 .$$

Proof of Theorem 13.5.1. Let Span(G) denote the set of spanning trees of G. For an edge e,

$$\begin{aligned}
\Pr_T\left[e \in T\right] &= \sum_{T \in \mathrm{Span}(G): e \in T} \frac{\sigma_{n-1}(\boldsymbol{L}_{G_T})}{\sigma_{n-1}(\boldsymbol{L}_G)} \\
&= \sum_{T \in \mathrm{Span}(G): e \in T} \sigma_{n-1}(\boldsymbol{L}_{G_T})\sigma_{n-1}(\boldsymbol{L}_G^{+}) \\
&= \sum_{T \in \mathrm{Span}(G): e \in T} \sigma_{n-1}(\boldsymbol{L}_{G_T}\boldsymbol{L}_G^{+}),
\end{aligned}$$

by (13.4). Recalling that the subsets of $n-1$ edges that are not spanning trees contribute 0 allows us to re-write this sum as

$$\sum_{|S|=n-1, e\in S} \sigma_{n-1}(\boldsymbol{L}_{G_S}\boldsymbol{L}_G^{+}).$$

To evaluate the terms in the sum, we compute

$$\begin{aligned}
\sigma_{n-1}(\boldsymbol{L}_{G_S}\boldsymbol{L}_G^{+}) &= \sigma_{n-1}(\boldsymbol{B}(:,S)\boldsymbol{B}(:,S)^T\boldsymbol{L}_G^{+}) \\
&= \sigma_{n-1}(\boldsymbol{B}(:,S)^T\boldsymbol{L}_G^{+}\boldsymbol{B}(:,S)) \\
&= \sigma_{n-1}(\boldsymbol{\Gamma}(S,S)) \\
&= \sigma_{n-1}(\boldsymbol{\Gamma}(S,:)\boldsymbol{\Gamma}(:,S)).
\end{aligned}$$

Let $\boldsymbol{\gamma}_e = \boldsymbol{\Gamma}(e,:)$ and let $\boldsymbol{\Pi}_{\boldsymbol{\gamma}_e}$ denote the projection orthogonal to $\boldsymbol{\gamma}_e$. As $e \in S$, we have

$$\sigma_{n-1}(\boldsymbol{\Gamma}(S,:)\boldsymbol{\Gamma}(:,S)) = \|\boldsymbol{\gamma}_e\|^2 \sigma_{n-2}(\boldsymbol{\Gamma}(S,:)\boldsymbol{\Pi}_{\boldsymbol{\gamma}_e}\boldsymbol{\Gamma}(:,S)) = \|\boldsymbol{\gamma}_e\|^2 \sigma_{n-2}((\boldsymbol{\Gamma}\boldsymbol{\Pi}_{\boldsymbol{\gamma}_e}\boldsymbol{\Gamma})(S,S)).$$

As $\boldsymbol{\gamma}_e$ is in the span on $\boldsymbol{\Gamma}$, the matrix $\boldsymbol{\Gamma}\boldsymbol{\Pi}_{\boldsymbol{\gamma}_e}\boldsymbol{\Gamma}$ is a symmetric projection onto an $n-2$ dimensional space, and so

$$\sigma_{n-2}(\boldsymbol{\Gamma}\boldsymbol{\Pi}_{\boldsymbol{\gamma}_e}\boldsymbol{\Gamma}) = 1.$$

To exploit this identity, we return to our summation:

$$\begin{aligned}
\sum_{|S|=n-1, e \in S} \sigma_{n-1}(\boldsymbol{L}_{G_S}\boldsymbol{L}_G^{+}) &= \sum_{|S|=n-1, e \in S} \|\boldsymbol{\gamma}_e\|^2 \sigma_{n-2}((\boldsymbol{\Gamma}\boldsymbol{\Pi}_{\boldsymbol{\gamma}_e}\boldsymbol{\Gamma})(S,S)) \\
&= \|\boldsymbol{\gamma}_e\|^2 \sum_{|S|=n-1, e \in S} \sigma_{n-2}((\boldsymbol{\Gamma}\boldsymbol{\Pi}_{\boldsymbol{\gamma}_e}\boldsymbol{\Gamma})(S,S)) \\
&= \|\boldsymbol{\gamma}_e\|^2 \sigma_{n-2}(\boldsymbol{\Gamma}\boldsymbol{\Pi}_{\boldsymbol{\gamma}_e}\boldsymbol{\Gamma}) \\
&= \|\boldsymbol{\gamma}_e\|^2 \\
&= \ell_e.
\end{aligned}$$

□

Chapter 14

Approximating Effective Resistances

In this chapter, we will see how to use the Johnson-Lindenstrauss Lemma, one of the major techniques for dimension reduction, to approximately represent and compute effective resistances.

Throughout this chapter, $G = (V, E, w)$ will be a connected, weighted graph with n vertices and m edges.

14.1 Representing Effective Resistances

We begin by considering the problem of building a data structure from which one can quickly estimate the effective resistance between every pair of vertices $a, b \in V$. To do this, we exploit the fact from Section 12.1 that effective resistances can be expressed as squares of Euclidean distances:

$$\begin{aligned}
\mathrm{R}_{\mathrm{eff}}(a, b) &= (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}^{+} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b) \\
&= \left\| \boldsymbol{L}^{+/2} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b) \right\|^2 \\
&= \left\| \boldsymbol{L}^{+/2} \boldsymbol{\delta}_a - \boldsymbol{L}^{+/2} \boldsymbol{\delta}_b \right\|^2 \\
&= \mathrm{dist}(\boldsymbol{L}^{+/2} \boldsymbol{\delta}_a, \boldsymbol{L}^{+/2} \boldsymbol{\delta}_b)^2.
\end{aligned}$$

One other way of expressing the above terms is through a *matrix norm* . For a positive semidefinite matrix $\boldsymbol{A}$, the matrix norm in $\boldsymbol{A}$ is defined by

$$\|\boldsymbol{x}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}} = \left\| \boldsymbol{A}^{1/2} \boldsymbol{x} \right\|.$$

It is worth observing that this is in fact a norm: it is zero when $\boldsymbol{x}$ is zero, it is symmetric, and it obeys the triangle inequality: for $\boldsymbol{x} + \boldsymbol{y} = \boldsymbol{z}$,

$$\|\boldsymbol{z}\|_{\boldsymbol{A}} = \left\| \boldsymbol{A}^{1/2} \boldsymbol{z} \right\| = \left\| \boldsymbol{A}^{1/2} (\boldsymbol{x} + \boldsymbol{y}) \right\| \leq \left\| \boldsymbol{A}^{1/2} \boldsymbol{x} \right\| + \left\| \boldsymbol{A}^{1/2} \boldsymbol{y} \right\| = \|\boldsymbol{x}\|_{\boldsymbol{A}} + \|\boldsymbol{y}\|_{\boldsymbol{A}}.$$

The Johnson-Lindenstrauss Lemma [JL84] tells us that every Euclidean metric on n points is well-approximated by a Euclidean metric in $O(\log n)$ dimensions, regardless of the original dimension of the points. Johnson and Lindenstrauss proved this by applying a random orthogonal projection to the points. As is now common, we will analyze the simpler operation of applying a random matrix of Gaussian random variables (also known as Normal variables). All Gaussian random variables that appear in this chapter will have mean 0.

We recall that a Gaussian random variable of variance 1 has probability density

$$p(x) = \frac{1}{\sqrt{2\pi}} \exp(-x^2/2),$$

and that a Gaussian random variable of variance σ^2 has probability density

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp(-x^2/2\sigma^2).$$

The distribution of such a variable is written $\mathcal{N}(0,\sigma^2)$, where the 0 corresponds to the mean being 0. A variable with distribution $\mathcal{N}(0,\sigma^2)$ may be obtained by sampling one with distribution $\mathcal{N}(0,1)$, and then multiplying it by σ. Gaussian random variables have many special properties, some of which we will see in this chapter. For those who are not familiar with them, we begin by mentioning that they are the limit of a binomial distribution. If X is the sum of n ±1 random variables for large n, then

$$\Pr\left[X/\sqrt{n} = t\right] \rightarrow p(t).$$

Theorem 14.1.1. *Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be vectors in $\mathbb{R}^k$. For any $\epsilon, \delta > 0$, let $d = 8(\ln(n^2/\delta)/\epsilon^2$. If $\boldsymbol{R}$ is a d-by-k matrix of independent $\mathcal{N}(0,1/d)$ variables, then with probability at least $1-\delta$, for all $a \neq b$,*

$$(1-\epsilon)\mathrm{dist}(\boldsymbol{x}_a, \boldsymbol{x}_b)^2 \leq \mathrm{dist}(\boldsymbol{R}\boldsymbol{x}_a, \boldsymbol{R}\boldsymbol{x}_b)^2 \leq (1+\epsilon)\mathrm{dist}(\boldsymbol{x}_a, \boldsymbol{x}_b)^2.$$

Thus, if we set $d = 8(\ln(n^2/\delta)/\epsilon^2$, let $\boldsymbol{R}$ be a d-by-n matrix of independent $N(0,1/d)$ variables, and set $\boldsymbol{y}_a = \boldsymbol{R}\boldsymbol{L}^{+/2}\boldsymbol{\delta}_a$ for each $a \in V$, then with probability at least $1-\delta$ we will have that for every a and b, $\mathrm{R_{eff}}(a,b)$ is within a $1 \pm \epsilon$ factor of $\mathrm{dist}(\boldsymbol{y}_a, \boldsymbol{y}_b)^2$. Whereas writing all effective resistances would require $\binom{n}{2}$ numbers, storing $\boldsymbol{y}_1, \dots, \boldsymbol{y}_n$ only requires $?nd$.

We remark that the 8 in the theorem can be replace with a constant that tends towards 4 as ϵ goes to zero.

14.2 Computing Effective Resistances

Note that the naive way of computing one effective resistance requires solving one Laplacian system: $(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}^+ (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)$. We will see that we can approximate all of them by solving a logarithmic number of such systems.

If we could quickly multiply a vector by $\boldsymbol{L}^{+/2}$, then this would give us a fast way of approximately computing all effective resistances. All we would need to do is multiply each of the d rows of $\boldsymbol{R}$ by $\boldsymbol{L}^{+/2}$. This would provide the matrix $\boldsymbol{R}\boldsymbol{L}^{+/2}$, from which we could compute $\boldsymbol{R}\boldsymbol{L}^{+/2}\boldsymbol{\delta}_a$ by selecting the ath column. This leads us to ask how quickly we can multiply a vector by $\boldsymbol{L}^{+/2}$. Cheng, Cheng, Liu, Peng and Teng [CCL+15] show that this can be done in nearly-linear time. In this section, we will present a more elementary approach that merely requires solving systems of equations in Laplacian matrices. We will see in Chapter **??** that this can be done very quickly.

The key is to realize that we do not actually need to multiply by the square root of the pseudoinverse of the Laplacian. Any matrix $\boldsymbol{M}$ such that $\boldsymbol{M}^T \boldsymbol{M} = \boldsymbol{L}^+$ will suffice.

Recall that we can write $\boldsymbol{L} = \boldsymbol{U}^T \boldsymbol{W} \boldsymbol{U}$, where $\boldsymbol{U}$ is the signed edge-vertex adjacency matrix and $\boldsymbol{W}$ is the diagonal matrix of edge weights. We then have

$$\boldsymbol{L}^+ \boldsymbol{U}^T \boldsymbol{W}^{1/2} \boldsymbol{W}^{1/2} \boldsymbol{U} \boldsymbol{L}^+ = \boldsymbol{L}^+ \boldsymbol{L} \boldsymbol{L}^+ = \boldsymbol{L}^+.$$

So,

$$\left\| \boldsymbol{W}^{1/2} \boldsymbol{U} \boldsymbol{L}^+ (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b) \right\|^2 = \mathrm{R_{eff}}(a, b).$$

Now, we let $\boldsymbol{R}$ be a d-by-m matrix of independent $\mathcal{N}(0, 1/d)$ entries, and compute

$$\boldsymbol{R} \boldsymbol{W}^{1/2} \boldsymbol{U} \boldsymbol{L}^+ = (\boldsymbol{R} \boldsymbol{W}^{1/2} \boldsymbol{U}) \boldsymbol{L}^+.$$

This requires multiplying d vectors in $\mathbb{R}^m$ by $\boldsymbol{W}^{1/2} \boldsymbol{U}$, and solving d systems of linear equations in $\boldsymbol{L}$. We then set

$$\boldsymbol{y}_a = (\boldsymbol{R} \boldsymbol{W}^{1/2} \boldsymbol{U}) \boldsymbol{L}^+ \boldsymbol{\delta}_a.$$

Each of these is a vector in d dimensions, and with high probability $\| \boldsymbol{y}_a - \boldsymbol{y}_b \|^2$ is a good approximation of $\mathrm{R_{eff}}(a, b)$.

14.3 Properties of Gaussian random variables

The sum a Gaussian random variables is another Gaussian random variable.

Claim 14.3.1. *If $r_1, \ldots, r_n$ are independent Gaussian random variables of variances $\sigma_1^2, \ldots, \sigma_n^2$, respectively, then*

$$\sum_{i=1}^{n} r_i$$

is a Gaussian random variable of variance

$$\sum_{i=1}^{n} \sigma_i^2.$$

One way to remember this is to recall that for a $\mathcal{N}(0, \sigma^2)$ random variable r, $\mathbb{E} r^2 = \sigma^2$, and the variance of the sum of independent random variables is the sum of their variances. The above claim adds the fact that the sum is also Gaussian.

In particular, if $\boldsymbol{x}$ is an arbitrary vector and $\boldsymbol{r}$ is a vector of independent $\mathcal{N}(0, 1)$ random variables, then $\boldsymbol{x}^T \boldsymbol{r}$ is a Gaussian random variable of variance $\| \boldsymbol{x} \|^2$. This follows because $\boldsymbol{x}(i) \boldsymbol{r}(i)$ has variance $\boldsymbol{x}(i)^2$, and

$$\boldsymbol{x}^T \boldsymbol{r} = \sum \boldsymbol{x}(i) \boldsymbol{r}(i).$$

If $\boldsymbol{x} \in \mathbb{R}^k$ and $\boldsymbol{R}$ is a matrix of independent $\mathcal{N}(0,\sigma^2)$ variables, then each entry of $\boldsymbol{R}\boldsymbol{x}$ is an independent $\mathcal{N}(0,\sigma^2 \left\| x \right\|^2)$ random variable. They are independent because each entry comes from a separate row of $\boldsymbol{R}$, and the variables in different rows are independent from each other.

The norm of a vector of identical independent $\mathcal{N}(0,1)$ random variables is called a χ random variable, and its square is a χ^2 random variable. A lot is known about the distribution of χ^2 random variables. If the vector has dimension d, then its expectation is d. It is very unlikely to deviate too much from this.

For example, the following bound appears as Lemma 1 of [LM00].

Lemma 14.3.2. *Let $r_1, \dots, r_d$ be independent $\mathcal{N}(0,1)$ random variables and let $X = \sum_i r_i^2$. Then, for all $t > 0$,*

$$Pr\left[X \geq d + 2\sqrt{dt} + 2t\right] \leq \exp(-t), \quad \textit{and}$$
$$Pr\left[X \leq d - 2\sqrt{dt}\right] \leq \exp(-t).$$

We use the following corollary.

Corollary 14.3.3. *For $\epsilon < 1$,*

$$Pr\left[|X - d| \geq \epsilon d\right] \leq 2\exp(-\epsilon^2 d/8).$$

Proof. Set $t = \epsilon^2 d/8$. This gives

$$2\sqrt{dt} + 2t \leq 2\frac{\epsilon d}{\sqrt{8}} + \frac{\epsilon^2 d}{4} \leq 2\frac{\epsilon d}{\sqrt{8}} + \frac{\epsilon d}{4} < \epsilon d.$$

Finally, the probability that $X - d > \epsilon d$ or $X - d < -\epsilon d$ is at most the sum of these probabilities, which is at most $2\exp(-t)$. □

We remark that for small ϵ the term $2\epsilon d/\sqrt{8}$ dominates, and the upper bound of ϵd approaches $\epsilon d/\sqrt{2}$. If one pushes this into the proof below, we see that it suffices to project into a space of dimension dimension of just a little more than $4(\ln(n^2/\delta)/\epsilon^2$, instead of $8(\ln(n^2/\delta)/\epsilon^2$.

14.4 Proof of Johnson-Lindenstrauss

Proof of Theorem 14.1.1. First consider an arbitrary a and b, and let $\Delta = \left\| \boldsymbol{x}_a - \boldsymbol{x}_b \right\|^2$. Each entry of $\boldsymbol{R}(\boldsymbol{x}_a - \boldsymbol{x}_b)$ is a d-dimensional vector of $\mathcal{N}(0,\sigma^2)$ variables, where $\sigma^2 = \Delta/d$. Thus, Corollary 14.3.3 tells us that

$$\Pr\left[\left|\mathrm{dist}(\boldsymbol{R}\boldsymbol{x}_a, \boldsymbol{R}\boldsymbol{x}_b)^2 - \mathrm{dist}(\boldsymbol{x}_a, \boldsymbol{x}_b)^2\right| > \epsilon \mathrm{dist}(\boldsymbol{x}_a, \boldsymbol{x}_b)^2\right] =$$
$$\Pr\left[\left|\left\| \boldsymbol{R}(\boldsymbol{x}_a - \boldsymbol{x}_b) \right\|^2 - \Delta\right| \geq \epsilon\Delta\right] \leq 2\exp(-\epsilon^2 d/8).$$

Thus the choice of $d = 8(\ln(n^2/\delta)/\epsilon^2$ makes this probability at most

$$2\exp(-\epsilon^2 d/8) \leq 2\exp(-\ln(n^2/\delta)) = \frac{2\delta}{n^2}.$$

As there $\binom{n}{2}$ possible choices for a and b, the probability that there is one such that

$$\left\| \boldsymbol{R}(\boldsymbol{x}_a - \boldsymbol{x}_b) \right\|^2 \not\in (1 \pm \epsilon) \left\| \boldsymbol{x}_a - \boldsymbol{x}_b \right\|^2$$

is at most

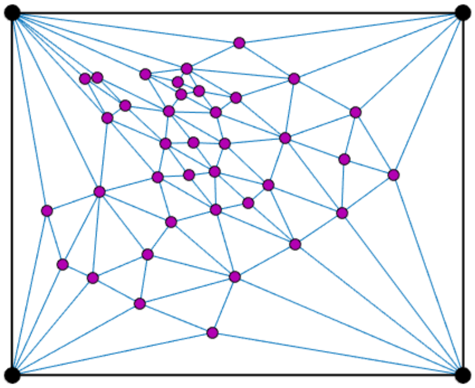
$$\binom{n}{2} \frac{2\delta}{n^2} < \delta.$$

□

Chapter 15

Tutte's Theorem: How to draw a graph

We prove Tutte's theorem [Tut63], which shows how to use spring embeddings to obtain planar drawings of 3-connected planar graphs. One begins by selecting a face, and then nailing down the positions of its vertices to the corners of a strictly convex polygon. Of course, the edges of the face should line up with the edges of the polygon. Ever other vertex goes where the springs say they should—to the center of gravity of their neighbors. Tutte proved that the result is a planar embedding of the planar graph. Here is an image of such an embedding



The presentation in this lecture is a based on notes given to me by Jim Geelen. I begin by recalling some standard results about planar graphs that we will assume.

15.1 3-Connected, Planar Graphs

A graph $G = (V, E)$ is k-connected if there is no set of $k - 1$ vertices whose removal disconnects the graph. That is, for every $S \subset V$ with $|S| < k$, $G(V - S)$ is connected. In a classical graph theory course, one usually spends a lot of time studying things like 3-connectivity.

A *planar drawing* of a graph $G = (V, E)$ consists of mapping from the vertices to the plane, $\boldsymbol{z} : V \rightarrow \mathbb{R}^2$, along with interior-disjoint curves for each edge. The curve for edge (a, b) starts at $\boldsymbol{z}(a)$, ends at $\boldsymbol{z}(b)$, never crosses itself, and its interior does not intersect the curve for any other edge. A graph is *planar* if it has a planar drawing. There can, of course, be many planar drawings of a graph.

If one removes the curves corresponding to the edges in a planar drawing, one divides the plane into connected regions called *faces*. In a 3-connected planar graph, the sets of vertices and edges that border each face are the same in every planar drawing. There are planar graphs that are not 3-connected, like those in Figures 15.1 and 15.1, in which different planar drawings result in combinatorially different faces. We will only consider 3-connected planar graphs.

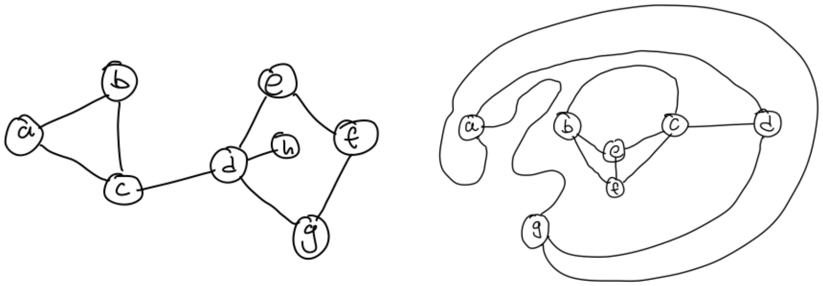


Figure 15.1: Planar graphs that are merely one-connected. Edge (c, d) appears twice on a face in each of them.

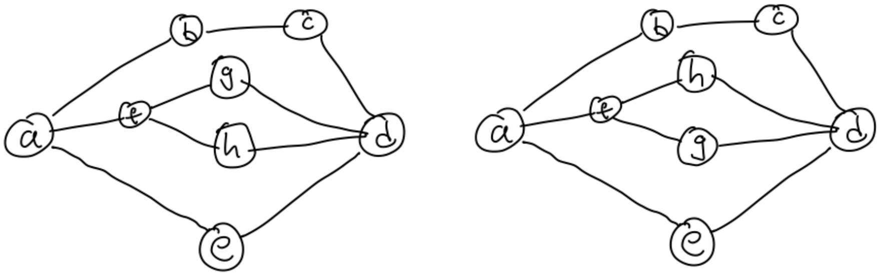


Figure 15.2: Two different planar drawings of a planar graph that is merely two-connected. Vertices g and h have switched positions, and thus appear in different faces in each drawing.

We state a few properties of 3-connected planar graphs that we will use. We will not prove these properties, as we are more concerned with algebra and these properly belong in a class on combinatorial graph theory.

Claim 15.1.1. *Let $G = (V, E)$ be a 3-connected planar graph. Then, there exists a set of faces F, each of which corresponds to a cycle in G, so that no vertex appears twice in a face, no edge appears twice in a face, and every edge appears in exactly two faces.*

We call the face on the outside of the drawing the *outside face*. The edges that lie along the outside face are the *boundary edges*.

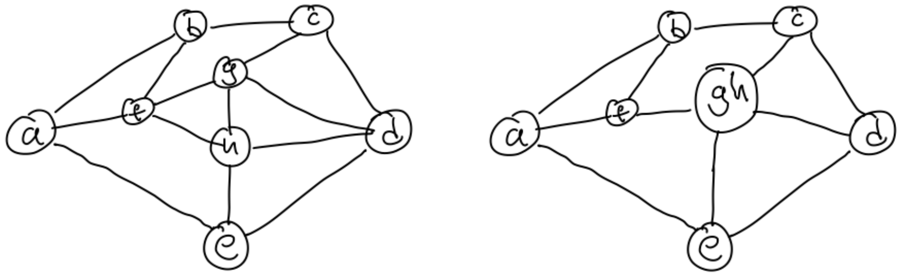


Figure 15.3: 3-connected planar graphs. Some faces of the graph on the left are abf, fgh, and $afhe$. The outer face is $abcde$. The graph on the right is obtained by contracting edge (g, h).

Another standard fact about planar graphs is that they remain planar under edge contractions. Contracting an edge (a, b) creates a new graph in which a and b become the same vertex, and all edges that went from other vertices to a or b now go to the new vertex. Contractions also preserve 3-connectivity. Figure 15.1 depicts a 3-connected planar graph and the result of contracting an edge.

A graph $H = (W, F)$ is a minor of a graph $G = (V, E)$ if H can be obtained from G by contracting some edges and possibly deleting other edges and vertices. This means that each vertex in W corresponds to a connected subset of vertices in G, and that there is an edge between two vertices in W precisely when there is some edge between the two corresponding subsets. This leads to Kuratowski's Theorem [Kur30], one of the most useful characterizations of planar graphs.

Theorem 15.1.2. *A graph G is planar if and only if it does not have a minor isomorphic to the complete graph on 5 vertices, K_5, or the bipartite complete graph between two sets of 3 vertices, $K_{3,3}$.*

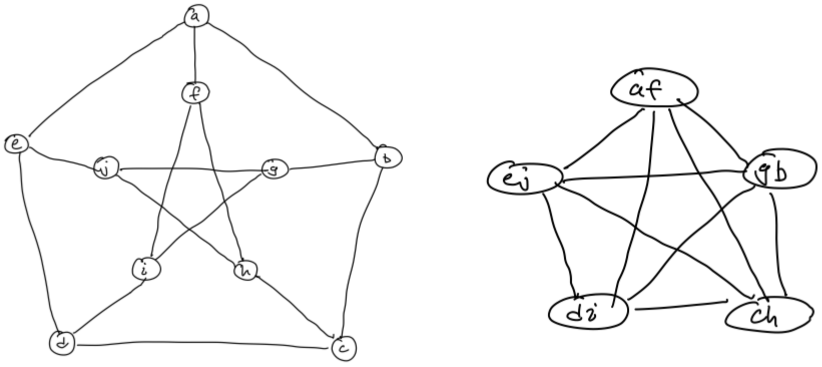


Figure 15.4: The Peterson graph appears on the left. On the right is a minor of the Peterson graph that is isomorphic to K_5, proving that the Peterson graph is not planar.

We will use one other important fact about planar graphs, whose utility in this context was observed by Jim Geelen.

Lemma 15.1.3. *Let (a, b) be an edge of a 3-connected planar graph and let S_1 and S_2 be the sets of vertices on the two faces containing (a, b). Let P be a path in G that starts at a vertex of $S_1 - \{a, b\}$, ends at a vertex of $S_2 - \{a, b\}$, and that does not intersect a or b. Then, every path in G from a to b either intersects a vertex of P or the edge (a, b).*

Proof. Let s_1 and s_2 be the vertices at the ends of the path P. Consider a planar drawing of G and ¡the closed curve in the plane that follows the path P from s_1 to s_2, and then connects s_1 to s_2 by moving inside the faces S_1 and S_2, where the path only intersects the curve for edge (a, b). This curve separates vertex a from vertex b. Thus, every path in G that connects a to b must intersect this curve. This means that it must either consist of just edge (a, b), or it must intersect a vertex of P. See Figure 15.1. □

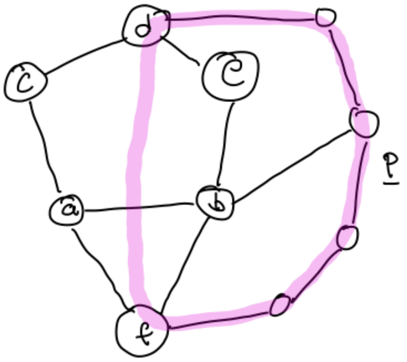
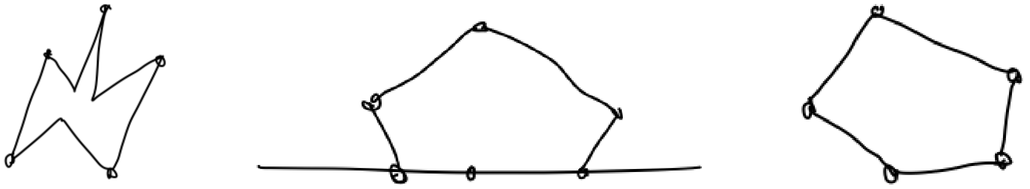


Figure 15.5: A depiction of Lemma 15.1.3. $S_1 = abcde$, $S_2 = abf$, and the path P starts at d, ends at f, and contains the other unlabeled vertices.

15.2 Strictly Convex Polygons

This is a good time to remind you what exactly a convex polygon is. A subset $C \subseteq \mathbb{R}^2$ is convex if for every two points $\boldsymbol{x}$ and $\boldsymbol{y}$ in C, the line segment between $\boldsymbol{x}$ and $\boldsymbol{y}$ is also in C. A convex polygon is a convex region of $\mathbb{R}^2$ whose boundary is comprised of a finite number of straight lines. It is *strictly* convex if in addition the angle at every corner is less than π. We will always assume that the corners of a strictly convex polygon are distinct. Two corners form an edge of the polygon if the interior of the polygon is entirely on one side of the line through those corners. This leads to another definition of a strictly convex polygon: a convex polygon is strictly convex if for every edge, all of the corners of the polygon other than those two defining the edge lie entirely on one side of the polygon. In particular, none of the other corners lie on the line.

Definition 15.2.1. *Let $G = (V, E)$ be a 3-connected planar graph. We say that $\boldsymbol{z} : V \to \mathbb{R}^2$ is a Tutte embedding if*



(a) A polygon

(b) A convex polygon

(c) A strictly convex polygon

Figure 15.6: Polygons

a. *There is a face F of G such that $\boldsymbol{z}$ maps the vertices of F to the corners of a strictly convex polygon so that every edge of the face joins consecutive corners of the polygon;*

b. *Every vertex not in F lies at the center of gravity of its neighbors.*

We will prove Tutte's theorem by proving that every face of G is embedded as a strictly convex polygon. In fact, we will not use the fact that every non-boundary vertex is exactly the average of its neighbors. We will only use the fact that every non-boundary vertex is inside the convex hull of its neighbors. This corresponds to allowing arbitrary spring constants in the embedding.

Theorem 15.2.2. *Let $G = (V, E)$ be a 3-connected planar graph, and let $\boldsymbol{z}$ be a Tutte embedding of G. If we represent every edge of G as the straight line between the embedding of its endpoints, then we obtain a planar drawing of G.*

Note that if the graph were not 3-connected, then the embedding could be rather degenerate. If there are two vertices a and b whose removal disconnects the graph into two components, then all of the vertices in one of those components will embed on the line segment from a to b.

Henceforth, G will always be a 3-connected planar graph and $\boldsymbol{z}$ will always be a Tutte embedding.

15.3 Possible Degeneracies

The proof of Theorem 15.2.2 will be easy once we rule out certain degeneracies. There are two types of degeneracies that we must show can not happen. The most obvious is that we can not have $\boldsymbol{z}(a) = \boldsymbol{z}(b)$ for any edge (a, b). The fact that this degeneracy can not happen will be a consequence of Lemma 15.4.1.

The other type of degeneracy is when there is a vertex a such that all of its neighbors lie on one line in $\mathbb{R}^2$. We will rule out such degeneracies in this section.

We first observe two simple consequences of the fact that every vertex must lie at the average of its neighbors.

Claim 15.3.1. *Let a be a vertex and let ℓ be any line in $\mathbb{R}^2$ through $\boldsymbol{z}(a)$. If a has a neighbor that lines on one side of ℓ, then it has a neighbor that lies on the other.*

Claim 15.3.2. *All vertices not in F must lie strictly inside the convex hull of the polygon of which the vertices in F are the corners.*

Proof. For every vertex a not in F, we can show that the position of a is a weighted average of the positions of vertices in F by eliminating every vertex not in $F \cup \{a\}$. As we learned in Lecture 13, this results in a graph in which all the neighbors of a are in F, and thus the position of a is some weighted average of the position of the vertices in F. As the graph is 3-connected, we can show that this average must assign nonzero weights to at least 3 of the vertices in F. □

Note that it is also possible to prove Claim 15.3.2 by showing that one could reduce the potential energy by moving vertices inside the polygon. See Claim 8.8.1 from my lecture notes from 2015.

Lemma 15.3.3. *Let H be a halfspace in $\mathbb{R}^2$ (that is, everything on one side of some line). Then the subgraph of G induced on the vertices a such that $\boldsymbol{z}(a) \in H$ is connected.*

Proof. Let $\boldsymbol{t}$ be a vector so that we can write the line ℓ in the form $\boldsymbol{t}^T \boldsymbol{x} = \mu$, with the halfspace consisting of those points $\boldsymbol{x}$ for which $\boldsymbol{t}^T \boldsymbol{x} \geq \mu$. Let a be a vertex such that $\boldsymbol{z}(a) \in H$ and let b be a vertex that maximizes $\boldsymbol{t}^T \boldsymbol{z}(b)$. So, $\boldsymbol{z}(b)$ is as far from the line defining the halfspace as possible. By Claim 15.3.2, b must be on the outside face, F.

For every vertex c, define $t(c) = \boldsymbol{t}^T \boldsymbol{z}(c)$. We will see that there is a path in G from a to b along which the function t never decreases, and thus all the vertices along the path lie in the halfspace. We first consider the case in which $t(a) = t(b)$. In this case, we also know that $a \in F$. As the vertices in F embed to a strictly convex polygon, this implies that (a, b) is an edge of that polygon, and thus the path from a to b.

If $t(a) < t(b)$, it suffices to show that there is a path from a to some other vertex c for which $t(c) > t(a)$ and along which t never decreases: we can then proceed from c to obtain a path to b. Let U be the set of all vertices u reachable from a for which $t(u) = t(a)$. As the graph is connected, there must be a vertex $u \in U$ that has a neighbor $c \notin U$. By Claim 15.3.1 u must have a neighbor c for which $t(c) > t(u)$. Thus, the a path from a through U to c suffices. □

Lemma 15.3.4. *No vertex is colinear with all of its neighbors.*

Proof. This is trivially true for vertices in F, as no three of them are colinear.

Assume by way of contradiction that there is a vertex a that is colinear with all of its neighbors. Let ℓ be that line, and let S^+ and S^- be all the vertices that lie above and below the line, respectively. Lemma 15.3.3 tells us that both sets S^+ and S^- are connected. Let U be the set of vertices u reachable from a and such that all of us neighbors lie on ℓ. The vertex a is in U. Let W be the set of nodes that lie on ℓ that are neighbors of vertices in U, but which themselves are not in U. As vertices in W are not in U, Claim 15.3.1 implies that each vertex in W has neighbors in both S^+ and S^-. As the graph is 3-connected, and removing the vertices in W would disconnect U from the rest of the graph, there are at least 3 vertices in W. Let w_1, w_2 and w_3 be three of the vertices in W.

We will now obtain a contradiction by showing that G has a minor isomorphic to $K_{3,3}$. The three vertices on one side are w_1, w_2, and w_3. The other three are obtained by contracting the vertex sets S^+, S^-, and U. □

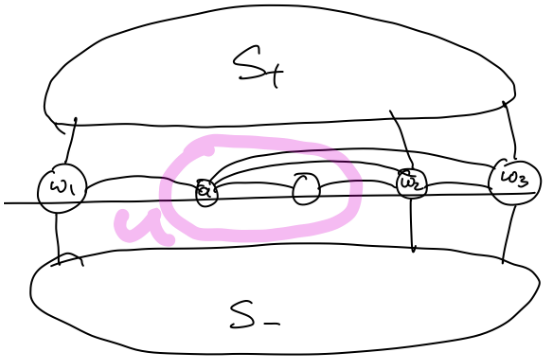


Figure 15.7: An illustration of the proof of Lemma 15.3.4.

15.4 All faces are convex

We now prove that every face of G embeds as a strictly convex polygon.

Lemma 15.4.1. *Let (a, b) be any non-boundary edge of the graph, and let ℓ be a line through $\boldsymbol{z}(a)$ and $\boldsymbol{z}(b)$ (there is probably just one). Let F_0 and F_1 be the faces that border edge (a, b) and let S_0 and S_1 be the vertices on those faces, other than a and b. Then all the vertices of S_0 and S_1 lie on opposite sides of ℓ, and none lie on ℓ.*

Note: if $\boldsymbol{z}(a) = \boldsymbol{z}(b)$, then we can find a line passing through them and one of the vertices of S_0. This leads to a contradiction, and thus rules out this type of degeneracy.

Proof. Assume by way of contradiction that the lemma is false. Without loss of generality, we may then assume that there are vertices of both S_0 and S_1 on or below the line ℓ. Let s_0 and s_1 be such vertices. By Lemma 15.3.4 and Claim 15.3.1, we know that both s_0 and s_1 have neighbors that lie strictly below the line ℓ. By Lemma 15.3.3, we know that there is a path P that connects s_0 and s_1 on which all vertices other than s_0 and s_1 lie strictly below ℓ.

On the other hand, we can similarly show that that both a and b have neighbors above the line ℓ, and that they are joined by a path that lies strictly above ℓ. Thus, this path cannot consist of the edge (a, b) and must be disjoint from P. This contradicts Lemma 15.1.3. □

So, we now know that the embedding $\boldsymbol{z}$ contains no degeneracies, that every face is embedded as a strictly convex polygon, and that the two faces bordering each edge embed on opposites sides of

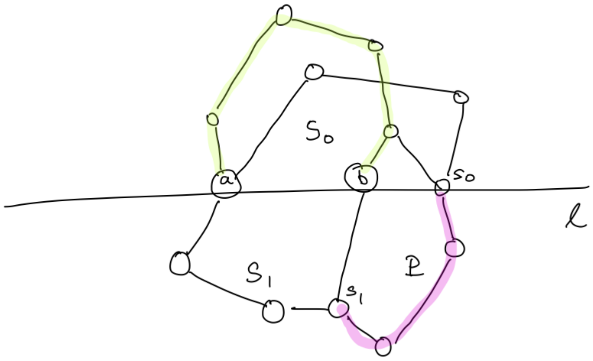


Figure 15.8: An illustration of the proof of Lemma 15.4.1.

that edge. This is all we need to know to prove Tutte's Theorem. We finish the argument in the proof below.

Proof of Theorem 15.2.2. We say that a point of the plane is *generic* if it does not lie on any $\boldsymbol{z}(a)$ for on any segment of the plane corresponding to an edge (a, b). We first prove that every generic point lies in exactly one face of G.

Begin with a point that is outside the polygon on which F is drawn. Such a point lies only in the outside face. For any other generic point we can draw a curve between these points that never intersects a $\boldsymbol{z}(a)$ and never crosses the intersection of the drawings of edges. That is, it only crosses drawings of edges in their interiors. By Lemma 15.4.1, when the curve does cross such an edge it moves from one face to another. So, at no point does it ever appear in two faces.

Now, assume by way of contradiction that the drawings of two edges cross. There must be some generic point near their intersection that lies in at least two faces. This would be a contradiction. □

15.5 Notes

This is the simplest proof of Tutte's theorem that I have seen. Over the years, I have taught many versions of Tutte's proof by building on expositions by Lovász [LV99] and Geelen [Gee12], and an alternative proof of Gortler, Gotsman and Thurston [GGT06].

Chapter 16

The Lovàsz - Simonovits Approach to Random Walks

Lecture 20 from November 7, 2018

16.1 Introduction

These notes are still very rough, and will be finished later.

For a vector $\boldsymbol{f}$ and an integer k, we define $\boldsymbol{f}\{k\}$ to be the sum of the largest k entries of $\boldsymbol{f}$. For convenience, we define $\boldsymbol{f}\{0\} = 0$. Symbolically, you can define this by setting π to be a permutation for which

$$\boldsymbol{f}(\pi(1)) \geq \boldsymbol{f}(\pi(2)) \geq ... \geq \boldsymbol{f}(\pi(n)),$$

and then setting

$$\boldsymbol{f}\{k\} = \sum_{i=1}^{k} \boldsymbol{f}(\pi(i)).$$

For real number x between 0 and n, we define $\boldsymbol{f}\{x\}$ by making it be piece-wise linear between consecutive integers. This means that for x between integers k and $k+1$, the slope of $\boldsymbol{f}\{\}$ at x is $\boldsymbol{f}(\pi(k+1))$. As these slopes are monotone nonincreasing, the function $\boldsymbol{f}\{x\}$ is concave.

We will prove the following theorem of Lovàsz and Simonovits [LS90] on the behavior of $\boldsymbol{W}\boldsymbol{f}$.

Theorem 16.1.1. *Let $\boldsymbol{W}$ be the transition matrix of the lazy random walk on a d-regular graph with conductance at least ϕ. Let $\boldsymbol{g} = \boldsymbol{W}\boldsymbol{f}$. Then for all integers $0 \leq k \leq n$*

$$\boldsymbol{g}\{k\} \leq \frac{1}{2}\left(\boldsymbol{f}\{k - \phi h\} + \boldsymbol{f}\{k + \phi h\}\right),$$

where $h = \min(k, n-k)$.

I remark that this theorem has a very clean extension to irregular, weighted graphs. I just present this version to simplify the exposition.

We can use this theorem to bound the rate of convergence of random walks in a graph. Let $\boldsymbol{p}_t$ be the probability distribution of the walk after t steps, and plot the curves $\boldsymbol{p}_t\{x\}$. The theorem tells us that these curves lie beneath each other, and that each curve lies beneath a number of chords drawn across the previous. The walk is uniformly mixed when the curve reaches a straight line from $(0,0)$ to $(n,1)$. This theorem tells us how quickly the walks approach the straight line.

Today, we will use the theorem to prove a variant of Cheeger's inequality.

16.2 Definitions and Elementary Observations

We believe that larger conductance should imply faster mixing. In the case of Theorem 16.1.1, it should imply lower curves. This is because wider chords lie beneath narrower ones.

Claim 16.2.1. *Let $h(x)$ be a convex function, and let $z > y > 0$. Then,*

$$\frac{1}{2}\left(h(x-z)+h(x+z)\right) \leq \frac{1}{2}\left(h(x-y)+h(x+y)\right).$$

Claim 16.2.2. *Let $\boldsymbol{f}$ be a vector, let $k \in [0,n]$, and let $\alpha_1,\ldots,\alpha_n$ be numbers between 0 and 1 such that*

$$\sum_i \alpha_i = k.$$

Then,

$$\sum_i \alpha_i \boldsymbol{f}(i) \leq \boldsymbol{f}\{k\}.$$

This should be obvious, and most of you proved something like this when solving problem 2 on homework 1. It is true because the way one would maximize this sum is by setting x to 1 for the largest values.

Throughout this lecture, we will only consider lazy random walks on regular graphs. For a set S and a vertex a, we define $\gamma(a,S)$ to be the probability that a walk that is at vertex a moves to S in one step. If a is not in S, this equals one half the fraction of edges from a to S. It is one half because there is a one half probability that the walk stays at a. Similarly, if a is in S, then $\gamma(a,S)$ equals one half plus one half the fraction of edges of a that end in S.

16.3 Warm up

We warm up by proving that the curves must lie under each other.

For a vector $\boldsymbol{f}$ and a set S, we define

$$\boldsymbol{f}(S) = \sum_{a\in S} \boldsymbol{f}(a).$$

For every k there is at least one set S for which

$$\boldsymbol{f}(S) = \boldsymbol{f}\{k\}.$$

If the values of $\boldsymbol{f}$ are distinct, then the set S is unique.

Lemma 16.3.1. *Let $\boldsymbol{f}$ be a vector and let $\boldsymbol{g} = \boldsymbol{W}\boldsymbol{f}$. Then for every $x \in [0, n]$,*

$$\boldsymbol{g}\{x\} \leq \boldsymbol{f}\{x\}.$$

Proof. As the function $\boldsymbol{g}\{x\}$ is piecewise linear between integers, it suffices to prove it at integers k. Let k be an integer and let S be a set of size k for which $\boldsymbol{f}(S) = \boldsymbol{f}\{k\}$. As $\boldsymbol{g} = \boldsymbol{W}\boldsymbol{f}$,

$$\boldsymbol{g}(S) = \sum_{a \in V} \gamma(a, S)\boldsymbol{f}(a).$$

As the graph is regular,

$$\sum_{a \in V} \gamma(a, S) = k.$$

Thus, Claim 16.2.2 implies

$$\sum_{a \in V} \gamma(a, S)\boldsymbol{f}(a) \leq \boldsymbol{f}\{k\}.$$

□

16.4 The proof

Recall that the conductance of a subset of vertices S in a d-regular graph is defined to be

$$\phi(S) \stackrel{\text{def}}{=} \frac{|\partial(S)|}{d \min(|S|, n - |S|)}.$$

Our proof of the main theorem improves the previous argument by exploiting the conductance through the following lemma.

Lemma 16.4.1. *Let S be any set of k vertices. Then*

$$\sum_{a \notin S} \gamma(a, S) = (\phi(S)/2) \min(k, n - k).$$

Proof. For $a \notin S$, $\gamma(a, S)$ equals half the fraction of the edges from a that land in S. And, the number of edges leaving S equals $d\phi(S) \min(k, n - k)$. □

Lemma 16.4.2. *Let $\boldsymbol{W}$ be the transition matrix of the lazy random walk on a d-regular graph, and let $\boldsymbol{g} = \boldsymbol{W}\boldsymbol{f}$. For every set S of size k with conductance at least ϕ,*

$$\boldsymbol{g}(S) \leq \frac{1}{2}\left(\boldsymbol{f}\{k - \phi h\} + \boldsymbol{f}\{k + \phi h\}\right),$$

where $h = \min(k, n - k)$.

Proof. To ease notation, define $\gamma(a) = \gamma(a, S)$. We prove the theorem by rearranging the formula

$$\boldsymbol{g}(S) = \sum_{a \in V} \gamma(a) \boldsymbol{f}(a).$$

Recall that $\sum_{a \in V} \gamma(a) = k$.

For every vertex a define

$$\alpha(a) = \begin{cases} \gamma(a) - 1/2 & \text{if } a \in S \\ 0 & \text{if } a \not\in S \end{cases} \qquad \text{and} \qquad \beta(a) = \begin{cases} 1/2 & \text{if } a \in S \\ \gamma(a) & \text{if } a \not\in S. \end{cases}$$

As $\alpha(a) + \beta(a) = \gamma(a)$,

$$\boldsymbol{g}(S) = \sum_{a \in V} \alpha(a) \boldsymbol{f}(a) + \sum_{a \in V} \beta(a) \boldsymbol{f}(a).$$

We now come to the point in the argument where we exploit the laziness of the random walk, which manifests as the fact that $\gamma(a) \geq 1/2$ for $a \in S$, and so $0 \leq \alpha(a) \leq 1/2$ for all a. Similarly, $0 \leq \beta(a) \leq 1/2$ for all a. So, we can write

$$\sum_{a \in V} \alpha(a) \boldsymbol{f}(a) = \frac{1}{2} \sum_{a \in V} (2\alpha(a)) \boldsymbol{f}(a), \qquad \text{and} \qquad \sum_{a \in V} \beta(a) \boldsymbol{f}(a) = \frac{1}{2} \sum_{a \in V} (2\beta(a)) \boldsymbol{f}(a)$$

with all coefficients $2\alpha(a)$ and $2\beta(a)$ between 0 and 1. As

$$\sum_{a \in V} \beta(a) = \frac{k}{2} + \sum_{a \not\in S} \gamma(a),$$

we can set

$$z = \sum_{a \not\in S} \gamma(a)$$

and write

$$\sum_{a \in V} (2\alpha(a)) = k - 2z \qquad \text{and} \qquad \sum_{a \in V} (2\beta(a)) = k + 2z.$$

Lemma 16.4.1 implies that

$$z \geq \phi h/2.$$

By Claim 16.2.2,

$$\boldsymbol{g}(S) \leq \frac{1}{2} \left(\boldsymbol{f}\{k - z\} + \boldsymbol{f}\{k + z\} \right).$$

So, Claim 16.2.1 implies

$$\boldsymbol{g}(S) \leq \frac{1}{2} \left(\boldsymbol{f}\{k - \phi h\} + \boldsymbol{f}\{k + \phi h\} \right).$$

□

Theorem 16.1.1 follows by applying Lemma 16.4.2 to sets S for which $\boldsymbol{f}(S) = \boldsymbol{f}\{k\}$, for each integer k between 0 and n.

16.5 Andersen's proof of Cheeger's inequality

Reid Andersen observed that the technique of Lovàsz and Simonovits can be used to give a new proof of Cheeger's inequality. I will state and prove the result for the special case of d-regular graphs that we consider in this lecture. But, one can of course generalize this to irregular, weighted graphs.

Theorem 16.5.1. *Let G be a d-regular graph with lazy random walk matrix $\boldsymbol{W}$, and let $\omega_2 = 1 - \lambda$ be the second-largest eigenvalue of $\boldsymbol{W}$. Then there is a subset of vertices S for which*

$$\phi(S) \leq \sqrt{8\lambda}.$$

Proof. Let $\boldsymbol{\psi}$ be the eigenvector corresponding to ω_2. As $\boldsymbol{\psi}$ is orthogonal to the constant vectors, $\boldsymbol{\psi}\{n\} = 0$. Define

$$k = \arg \max_{0 \leq k \leq n} \frac{\boldsymbol{\psi}\{k\}}{\sqrt{\min(k, n-k)}}.$$

Then, set γ to be the maximum value obtained:

$$\gamma = \frac{\boldsymbol{\psi}\{k\}}{\sqrt{\min(k, n-k)}}.$$

We will assume without loss of generality that $k \leq n/2$: if it is not then we replace $\boldsymbol{\psi}$ by $-\boldsymbol{\psi}$ to make it so and obtain the same γ. Now, $\boldsymbol{\psi}\{k\} = \gamma\sqrt{k}$.

We let S be a set (there is probably only one) for which

$$\boldsymbol{\psi}(S) = \boldsymbol{\psi}\{k\}.$$

As $\boldsymbol{\psi}$ is an eigenvector with positive eigenvalue, we also know that

$$(\boldsymbol{W}\boldsymbol{\psi})(S) = \boldsymbol{W}\boldsymbol{\psi}\{k\}.$$

We also know that

$$(\boldsymbol{W}\boldsymbol{\psi})(S) = (1-\lambda)\boldsymbol{\psi}(S) = (1-\lambda)\gamma\sqrt{k}.$$

Let ϕ be the conductance of S. Lemma 16.4.2 tells us that

$$(\boldsymbol{W}\boldsymbol{\psi})(S) \leq \frac{1}{2}\left(\boldsymbol{\psi}\{k - \phi k\} + \boldsymbol{\psi}\{k + \phi k\}\right).$$

By the construction of k and γ at the start of the proof, we know this quantity is at most

$$\frac{1}{2}\left(\gamma\sqrt{k - \phi k} + \gamma\sqrt{k + \phi k}\right) = \gamma\sqrt{k}\frac{1}{2}\left(\sqrt{1-\phi} + \sqrt{1+\phi}\right).$$

Combining the inequalities derived so far yields

$$(1-\lambda) \leq \frac{1}{2}\left(\sqrt{1-\phi} + \gamma\sqrt{1+\phi}\right).$$

An examination of the Taylor series for the last terms reveals that

$$\frac{1}{2}\left(\sqrt{1-\phi} + \gamma\sqrt{1+\phi}\right) \leq 1 - \phi^2/8.$$

This implies $\lambda \geq \phi^2/8$, and thus $\phi(S) \leq \sqrt{8\lambda}$. □

Chapter 17

Monotonicity and its Failures

Lecture 14 from October 15, 2013

17.1 Disclaimer

These notes are not necessarily an accurate representation of what happened in class. They are a combination of what I intended to say with what I think I said. They have not been carefully edited.

17.2 Overview

17.3 Effective Spring Constants

Consider a spring network. As in last lecture, we model it by a weighted graph $G = (V, E, w)$, where $w_{a,b}$ is the spring constant of the edge (a, b). Recall that a stronger spring constant results in a stronger connection between a and b.

Now, let s and t be arbitrary vertices in V. We can view the network as a large, complex spring connecting s to t. We then ask for the spring constant of this complex spring. We call it the *effective spring constant* between s and t.

To determine what it is, we recall the definition of the spring constant for an ordinary spring: the potential energy in a spring connecting a to b is the spring constant times times the square of the length of the spring, divided by 2. We use this definition to determine the effective spring constant between s and t.

Recall again that if we fix the positions of s and t on the real line, say to 0 and 1, then the positions $\boldsymbol{x}$ of the other vertices will minimize the total energy:

$$\mathcal{E}\left(\boldsymbol{x}\right) \overset{\text{def}}{=} \frac{1}{2} \sum_{(a,b) \in E} w_{a,b} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2. \tag{17.1}$$

As s and t are separated by a distance of 1, we may define twice this quantity to be the effective spring constant of the entire network between s and t. To verify that this definition is consistent, we should consider what happens if the displacement between s and t is something other than 1. If we fix the position of s to 0 and the position of t to y, then the homogeniety of the expression for energy (17.1) tells us that the vector $y\boldsymbol{x}$ will minimize the energy subject to the boundary conditions. Moreover, the energy in this case will be $y^2/2$ times the effective spring constant.

17.4 Monotonicity

Rayleigh's Monotonicity Principle tells us that if we alter the spring network by decreasing some of the spring constants, then the effective resistance between s and t will not increase.

Theorem 17.4.1. *Let $G = (V, E, w)$ be a weighted graph and let $\widehat{G} = (V, E, \widehat{w})$ be another weighted graph with the same edges and such that*

$$\widehat{w}_{a,b} \leq w_{a,b}$$

for all $(a, b) \in E$. For vertices s and t, let $c_{s,t}$ be the effective spring constant between s and t in G and let $\widehat{c}_{s,t}$ be the analogous quantity in $\widehat{G}$. Then,

$$\widehat{c}_{s,t} \leq c_{s,t}.$$

Proof. Let $\boldsymbol{x}$ be the vector of minimum energy in G such that $\boldsymbol{x}(s) = 0$ and $\boldsymbol{x}(t) = 1$. Then, the energy of $\boldsymbol{x}$ in $\widehat{G}$ is no greater:

$$\frac{1}{2} \sum_{(a,b) \in E} \widehat{w}_{a,b} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2 \leq \frac{1}{2} \sum_{(a,b) \in E} w_{a,b} (\boldsymbol{x}(a) - \boldsymbol{x}(b))^2 = c_{s,t}.$$

So, the minimum energy of a vector $\boldsymbol{x}$ in $\widehat{G}$ such that $\boldsymbol{x}(s) = 0$ and $\boldsymbol{x}(t) = 1$ will be at most $c_{s,t}$, and so $\widehat{c}_{s,t} \leq c_{s,t}$. □

While this principle seems very simple and intuitively obvious, it turns out to fail in just slightly more complicated situtations. Before we examine them, I will present the analogous material for electrical networks.

17.5 Effective Resistance

There are two (equivalent) ways to define the *effective resistance* between two vertices in a network of resistors. The first is to start with the formula

$$V = IR,$$

or, as I prefer to write it,

$$\boldsymbol{i}(a,b) = \frac{\boldsymbol{v}(a) - \boldsymbol{v}(b)}{r_{a,b}},$$

This formula tells us that if we have one resistor between a and b and we fix the voltage of a to 1 and the voltage of b to 0, then the amount of current that will flow from a to b is the reciprocal of the resistance. It also tells us that if we want to flow one unit of current, then we need to place a potential difference of $r_{a,b}$ between a and b. Recall that we define the weight of an edge to be the reciprocal of its resistance, as high resistance corresponds to poor connectivity. We can use this formula to define the effective resistance between two vertices s and t in an arbitrary complex network of resistors: we define the effective resistance between s and t to be the potential difference needed to flow one unit of current from s to t.

Algebraically, define $\boldsymbol{i}_{ext}$ to be the vector

$$\boldsymbol{i}_{ext}(a) = \begin{cases} 1 & \text{if } a = s \\ -1 & \text{if } a = t \\ 0 & \text{otherwise} \end{cases}.$$

This corresponds to a flow of 1 from s to t. We then solve for the voltages that realize this flow:

$$\boldsymbol{L}\boldsymbol{v} = \boldsymbol{i}_{ext},$$

by

$$\boldsymbol{v} = \boldsymbol{L}^{+}\boldsymbol{i}_{ext}.$$

We thus have

$$\boldsymbol{v}(s) - \boldsymbol{v}(t) = \boldsymbol{i}_{ext}^{T}\boldsymbol{v} = \boldsymbol{i}_{ext}^{T}\boldsymbol{L}^{+}\boldsymbol{i}_{ext} \overset{\text{def}}{=} \mathrm{R_{eff}}(s,t).$$

This agrees with the other natural approach to defining effective resistance: twice the energy dissipation when we flow one unit of current from s to t.

Theorem 17.5.1. *Let $\boldsymbol{i}$ be the electrical flow of one unit from vertex s to vertex t in a graph G. Then,*

$$\mathrm{R_{eff}}_{s,t} = \mathcal{E}(\boldsymbol{i}).$$

Proof. Recalling that $\boldsymbol{i}_{ext} = \boldsymbol{L}\boldsymbol{v}$, we have

$$\mathrm{R_{eff}}_{s,t} = \boldsymbol{i}_{ext}^{T}\boldsymbol{L}^{+}\boldsymbol{i}_{ext} = \boldsymbol{v}^{T}\boldsymbol{L}\boldsymbol{L}^{+}\boldsymbol{L}\boldsymbol{v} = \boldsymbol{v}^{T}\boldsymbol{L}\boldsymbol{v} = \mathcal{E}(\boldsymbol{v}).$$

□

Rayleigh's Monotonicity Theorem was originally stated for electrical networks.

Theorem 17.5.2 (Rayleigh's Monotonicity)**.** *The effective resistance between a pair of vertices cannot be decreased by increasing the resistance of some edges.*

17.6 Examples

In the case of a path graph with n vertices and edges of weight 1, the effective resistance between the extreme vertices is $n-1$.

In general, if a path consists of edges of resistance $r(1,2),\dots,r(n-1,n)$ then the effective resistance between the extreme vertices is

$$r(1,2)+\cdots+r(n-1,n).$$

To see this, set the potential of vertex i to

$$\boldsymbol{v}(i)=r(i,i+1)+\cdots+r(n-1,n).$$

Ohm's law then tells us that the current flow over the edge $(i,i+1)$ will be

$$\left(\boldsymbol{v}(i)-\boldsymbol{v}(i+1)\right)/r(i,i+1)=1.$$

If we have k parallel edges between two nodes s and t of resistances $r_1,\dots,r_k$, then the effective resistance is

$$\mathrm{R_{eff}}(s,t)=\frac{1}{1/r_1+\cdots+1/r_k}.$$

Again, to see this, note that the flow over the ith edge will be

$$\frac{1/r_i}{1/r_1+\cdots+1/r_k},$$

so the total flow will be 1.

17.7 Breakdown of Monotonicity

We will now exhibit a breakdown of monotonicity in networks of nonlinear elements. In this case, we will consider a network of springs and wires. For examples in electrical networks with resistors and diodes or for networks of pipes with valves, see [PP03] and [CH91].

There will be 4 important vertices in the network that I will describe, a, b, c and d. Point a is fixed in place at the top of my aparatus. Point d is attached to an object of weight 1. The network has two springs of spring constant 1: one from point a to point b and one from point c to point d. There is a very short wire connecting point b to point c.

As each spring is supporting one unit of weight, each is stretched to length 1. So, the distance from point a to point d is 2.

I now add two more wires to the network. One connects point a to point c and the other connects point b to point d. Both have lengths $1+\epsilon$, and so are slack. Thus, the addition of these wires does not change the position of the weight.

I now cut the small wire connecting point b to point c. While you would expect that removing material from the supporting structure would cause the weight to go down, it will in fact move up. To see why, let's analyze the resulting structure. It consists of two suppors in parallel. One consists of a spring from point a to point b followed by a wire of length $1 + \epsilon$ from point b to d. The other has a wire of length $1 + \epsilon$ from point a to point c followed by a spring from point c to point d. Each of these is supporting the weight, and so each carries half the weight. This means that the length of the springs will be $1/2$. So, the distance from a to d should be essentially $3/2$.

This sounds like a joke, but we will see in class that it is true. The measurements that we get will not be exactly 2 and $3/2$, but that is because it is difficult to find ideal springs at Home Depot.

In the example with resistors and diodes, one can increase electrical flow between two points by *cutting* a wire!

17.8 Traffic Networks

I will now explain some analogous behavior in traffic networks. We will examine the more formally in the next lecture.

We will use a very simple model of a road in a traffic network. It will be a directed edge between two vertices. The rate at which traffic can flow on a road will depend on how many cars are on the road: the more cars, the slower the traffic. I will assume that our roads are linear. That is, when a road has flow f, the time that it takes traffic to traverse the road is

$$af + b,$$

for some nonnegative constants a and b. I call this the characteristic function of the road.

We first consider an example of Pigou consisting of two roads between two vertices, s and t. The slow road will have characteristic function 1: think of a very wide super-highway that goes far out of the way. No matter how many cars are on it, the time from s to t will always be 1. The fast road is better: its characteristic is f. Now, assume that there is 1 unit of traffic that would like to go from s to t.

A global planner that could dictate the route that everyone takes could minimize the average time of the traffic going from s to t by assigning half of the traffic to take the fast road and half of the traffic to take the slow road. In this case, half of the traffic will take time 1 and half will take time $1/2$, for an average travel time of $3/4$. To see that this is optimal, let f be the fraction of traffic that takes the fast road. Then, the average travel time will be

$$f \cdot f + (1 - f) \cdot 1 = f^2 - f + 1.$$

Taking derivatives, we see that this is minimized when

$$2f - 1 = 0,$$

which is when $f = 1/2$.

On the other hand, this is not what people will naturally do if they have perfect information and freedom of choice. If a $f < 1$ fraction of the flow is going along the fast road, then those travelling on the fast road will get to t faster than those going on the slow road. So, anyone going on the slow road would rather take the fast road. So, all of the traffic will wind up on the fast road, and it will become not-so-fast. All of the traffic will take time 1.

We call this the Nash Optimal solution, because it is what everyone will do if they are only maximizing their own benefit. You should be concerned that this is not as well as they would do if they allowed some authority to dictate their routes. For example, the authority could dictate that half the cars go each way every-other day, or one way in the morning and another at night.

Let's see an even more disturbing example.

17.9 Braes's Paradox

We now examine Braes's Paradox, which is analogous to the troubling example we saw with springs and wires. This involves a network with 4 vertices, a, b, c, and d. All the traffic starts at $s = a$ and wants to go to $t = d$. There are slow roads from s to c and from d to t, and fast roads from s to d and from c to t. If half of the traffic goes through route sct and the other half goes through route sdt, then all the traffic will go from s to t in time $3/2$. Moroever, noone can improve their lot by taking a different route, so this is a Nash equilibrium.

We now consider what happens if some well-intentioned politician decides to build a very fast road connecting c to d. Let's say that its characteristic function is 0. This opens up a faster route: traffic can go from s to c to d to t. If no one else has changed route, then this traffic will reach t in 1 unit of time. Unfortunately, once everyone realizes this all the traffic will take this route, and everyone will now require 2 units of time to reach t.

Let's prove that formally. Let p_1, p_2 and p_3 be the fractions of traffic going over routes sct, sdt, and $scdt$, respectively. The cost of route sct is $p_1 + p_3 + 1$. The cost of route sdt is $p_2 + p_3 + 1$. And, the cost of route $scdt$ is $p_3 + p_3$. So, as long as p_3 is less than 1, the cheapest route will be $scdt$. So, all the traffic will go that way, and the cost of every route will be 2.

17.10 The Price of Anarchy

In any traffic network, we can measure the average amount of time it takes traffic to go from s to t under the optimal flow. We call this the cost of the social optimum, and denote it by $\mathrm{Opt}(G)$. When we let everyone pick the route that is best for themselves, the resulting solution is a Nash Equilibrium, and we denote it by $\mathrm{Nash}(G)$.

The "Price of Anarchy" is the cost to society of letting everyone do their own thing. That is, it is the ratio

$$\frac{\mathrm{Nash}(G)}{\mathrm{Opt}(G)}.$$

In these examples, the ratio was $4/3$. In the next lecture, we will show that the ratio is never more than 4/3 when the cost functions are linear. If there is time today, I will begin a more formal analysis of Opt(G) and Nash(G) that we will need in our proof.

17.11 Nash optimum

Let the set of s-t paths be $P_1, \ldots, P_k$, and let α_i be the fraction of the traffic that flows on path P_i. In the Nash equilibrium, no car will go along a sub-optimal path. Assuming that each car has a negligible impact on the traffic flow, this means that every path P_i that has non-zero flow must have minimal cost. That is, for all i such that $\alpha_i > 0$ and all j

$$c(P_i) \leq c(P_j).$$

17.12 Social optimum

Society in general cares more about the average time its takes to get from s to t. If we have a flow that makes this average time low, everyone could rotate through all the routes and decrease the total time that they spend in traffic. So, the social cost of the flow f is

$$\begin{aligned} c(\alpha_1, \ldots, \alpha_k) &\stackrel{\text{def}}{=} \\ \sum_i \alpha_i c(P_i) &= \sum_i \alpha_i \sum_{e \in P_i} c_e(f_e) \\ &= \sum_e c_e(f_e) \sum_{i : e \in P_i} \alpha_i \\ &= \sum_e c_e(f_e) f_e . \end{aligned}$$

Theorem 17.12.1. *All local minima of the social cost function are global minima. Moreover, the set of global minima is convex.*

Proof. This becomes easy once we re-write the cost function as

$$\sum_e c_e(f_e) f_e = \sum_e a_e f_e^2 + b_e f_e$$

and recall that we assumed that a_e and b_e are both at least zero. The cost function on each edge is convex. It is strictly convex if $a_e > 0$, but that does not matter for this theorem.

If you take two flows, say f^0 and f^1, the line segments of flows between them contains the flows of the form f^t where

$$f_e^t = t f_e^1 + (1 - t) f_e^0,$$

for $0 \leq t \leq 1$.

By the convexity of each cost function, we know that the cost of any flow f^t is at most the maximum of the costs of f^0 and f^1. So, if f^1 is the global optimum and f^0 is any other flow with higher cost, the flow f^ϵ will have a social cost lower than f^0. This means that f^0 cannot be a local optimum. Similarly, if both f^0 and f^1 are global optima, then f^t must be as well. □

Chapter 18

Dynamic and Nonlinear Networks

Lecture 16 from October 22, 2013

18.1 Disclaimer

These notes are not necessarily an accurate representation of what happened in class. They are a combination of what I intended to say with what I think I said. They have not been carefully edited.

18.2 Overview

In this lecture we will consider two generalizations of resistor networks: resistor networks with non-linear resistors and networks whose resistances change over time. While they were introduced over 50 years ago, non-linear resistor networks seem to have been recently rediscovered in the Machine Learning community. We will discuss how they can be used to improve the technique we learned in Lecture 13 for semi-supervised learning.

The material on time-varying networks that I will present comes from Cameron Musco's senior thesis from 2012.

18.3 Non-Linear Networks

A non-linear resistor network, as defined by Duffin [Duf47], is a like an ordinary resistor network but the resistances depend on the potential differences across them. In fact, it might be easier not to talk about resistances, and just say that the amount of flow across an edge increases as the potential difference across the edge does. For every resistor e, there is a function

$$\phi_e(v)$$

that gives the flow over resistor e when there is a potential difference of v between its terminals.

We will restrict our attention to functions ϕ that are

a. continuous,

b. monotone increasing,

c. symmetric, by which I mean $\phi_e(-v) = -\phi_e(v)$.

Note that condition c implies that $\phi_e(0) = 0$. For an ordinary resistor of resistance r, we have

$$\phi_e(v) = v/r.$$

However, we can and will consider more interesting functions.

If the graph is connected and we fix the voltages at some of the vertices, then there exists a setting of voltages at the other vertices that results in a flow satisfying flow-in equals flow-out at all non-boundary vertices. Moreover, this flow is unique.

We will prove this in the next section through the use of a generalization of energy dissipation.

18.4 Energy

We define the energy dissipation of an edge that has a potential difference of v to be

$$\Phi_e(v) \overset{\text{def}}{=} \int_0^v \phi_e(t)dt.$$

We will show that the setting of the voltages that minimizes the total energy provides the flow I claimed exists.

In the case of linear resistors, where $\phi_e(v) = v/r$,

$$\Phi_e(v) = \frac{1}{2}\frac{v^2}{r},$$

which is exactly the energy function we introduced in Lecture 13.

The conditions on ϕ_e imply that

d. Φ_e is strictly convex[1],

e. $\Phi_e(0) = 0$, and

f. $\Phi_e(-x) = \Phi_e(x)$.

[1] That is, for all $x \neq y$ and all $0 < \lambda < 1$, $\Phi_e(\lambda x + (1-\lambda)y) \leq \lambda\Phi_e(x) + (1-\lambda)\Phi_e(y)$.

We remark that a function that is strictly convex has a unique minimum, and that a sum of strictly convex functions is strictly convex.

Theorem 18.4.1. *Let $G = (V, E)$ be a non-linear resistor network with functions f_e satisfying conditions a, b and c for every $e \in E$. For every set $S \subseteq V$ and fixed voltages w_a for $a \in S$, there exists a setting of voltages v_a for $a \notin S$ that result in a flow of current that satisfies the flow-in equals flow-out conditions at every $a \notin S$. Moreover, these voltages are unique.*

Proof. For a vector of voltages v, define

$$\Phi(v) = \sum_{(a,b)\in E} \Phi_{(a,b)}(v_a - v_b).$$

As each of the functions $\Phi_{(a,b)}$ are strictly convex, Φ is as well. So, Φ has a minimum subject to the fixed voltages. At this minimum point, we know that for every $a \notin S$

$$\begin{aligned} 0 &= \frac{\partial \Phi(v)}{\partial v_a} \\ &= \sum_{b:(a,b)\in E} \frac{\partial \Phi_{(a,b)}(v_a - v_b)}{\partial v_a} \\ &= \sum_{b:(a,b)\in E} \phi_{(a,b)}(v_a - v_b). \end{aligned}$$

We may now set

$$f_{(a,b)} = \phi_{(a,b)}(v_a - v_b).$$

This is a valid flow because for every vertex $a \notin S$ the sum of the flows out of v_a, taken with appropriate signs, is zero.

Conversely, for any setting of voltages that results in a flow that has no loss or gain at any $a \notin S$, we can reverse the above equalities to show that the partial derivatives of $\Phi(v)$ are zero. As $\Phi(v)$ is strictly convex, this can only happen at the unique minimum of $\Phi(v)$. □

18.5 Uses in Semi-Supervised Learning

In Lecture 13, I suggested an approach to estimating a function f on the vertices of a graph given its values at a set $S \subseteq V$:

$$\min_{x: f(a)=x(a) \text{ for } a\in S} \sum_{(a,b)\in E} (x(a) - x(b))^2.$$

Moreover, we saw that we can minimize such a function by solving a system of linear equations.

Unfortunately, there are situtations in which this approach does not work very well. In general, this should not be surprising: sometimes the problem is just unsolvable. But, there are cases in which it would be reasonable to solve the learning problem in which this approach fails.

Better results are sometimes obtained by modifying the penalty function. For example, Bridle and Zhu [BZ13] (and, essentially, Herbster and Guy [HL09]) suggest

$$\min_{x: f(a)=x(a) \text{ for } a \in S} \sum_{(a,b) \in E} |x(a) - x(b)|^p ,$$

for $1 < p < 2$.

While a well-selected p will often improve accuracy, the drawback of this approach is that we cannot perform the minimization nearly as quickly as we can when $p = 2$.

18.6 Dual Energy

We can establish a corresponding, although different, energy for the flows. Let ψ be the inverse of ϕ. We then define the flow-energy of an edge that carries a flow of f to be

$$\Psi(f) \stackrel{\text{def}}{=} \int_0^f \psi(t) dt.$$

If we minimize the sum of the flow-energies over the space of flows, we again recover the unique valid flow in the network. (The function Φ is implicit in the work of Duffin. The dual Ψ comes from Millar [Mil51]).

In the classical case, Φ and Ψ are the same. While they are not the same here, *their sum is.* We will later prove that when $v = \psi(f)$,

$$\Psi(f) + \Phi(v) = fv.$$

In fact, one can show that for all f and v,

$$\Psi(f) + \Phi(v) \geq fv,$$

with equality only when $v = \psi(f)$.

Theorem 18.6.1. *Under the conditions of Theorem 18.4.1, let f_{ext} be the vector of external flows resulting from the induced voltages. Let f be the flow on the edges that is compatible with f_{ext} and that minimizes*

$$\Psi(f) \stackrel{\text{def}}{=} \sum_{(a,b) \in E} \Psi_{(a,b)}(f_{(a,b)}).$$

Then, f is the flow induced by the voltages shown to exist in Theorem 18.4.1.

Sketch. We first show that f is a potential flow. That is, that there exist voltages v so that for every edge (a, b), $f_{(a,b)} = \phi_{(a,b)}(v_a - v_b)$. The theorem then follows by the uniqueness established in Theorem 18.4.1.

To prove that f is a potential flow, we consider the potential difference that the flow "wants" to induce on each edge, $\psi(f_{(a,b)})$. There exist vertex potentials that agree with these desired potential differences if an only if for every pair of vertices and for every pair of paths between them, the sum of the desired potential differences along the edges in the paths is the same. To see this, arbitrarily fix the potential of one vertex, such as s. We may then set the potential of any other vertex a by summing the desired potential differences along the edges in any path from s.

Equivalenty, the desired potential differences are realizable if and only if the sum of these desired potential differences is zero around every cycle. To show that this is the case, we use the minimality of the flow. Because $\Psi(f)$ is strictly convex, small changes to the optimum have a negligible effect on its value (that is, the first derivative is zero). So, pushing an ϵ amount of flow around any cycle will not change the value of $\Psi(f)$. That is, the sum of the derivatives around any cycle will be zero. As

$$\frac{\partial}{\partial f} \Psi_e(f) = \psi_e(f),$$

this means that the sum of the desired potential differences around every cycle is zero. □

Theorem 18.6.2. *If $f = \phi(v)$, then*

$$\Phi(v) + \Psi(f) = vf.$$

Proof. One can prove this theorem through "integration by parts". But, I prefer a picture. In the following two figures, the curve is the plot of ϕ. In the first figure, the shaded region is the integral of ϕ between 0 and v (2 in this case). In the second figure, the shaded region is the integral of ψ between 0 and $\phi(v)$ (just turn the picture on its side). It is clear that these are complementary parts of the rectangle between the axes and the point $(v, \phi(v))$.

□

The bottom line is that almost all of the classical theory can be carried over to nonlinear networks.

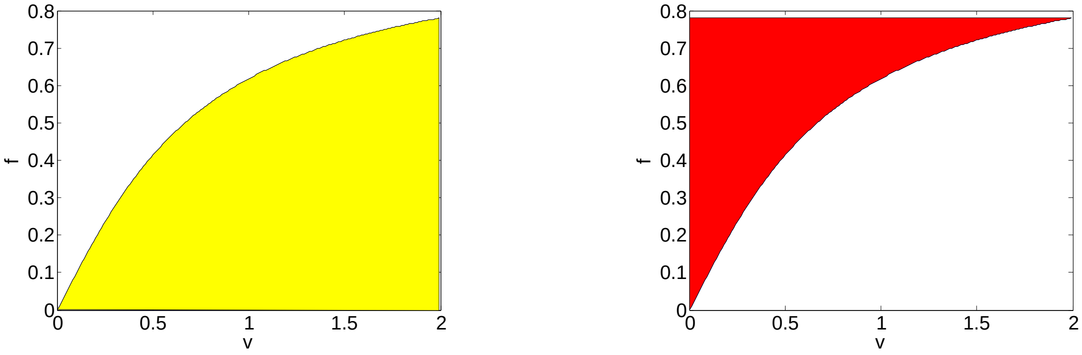
18.7 Thermistor Networks

We now turn our attention to networks of resistors whose resistance changes over time. We consider a natural model in which edges get "worn out": as they carry more flow their resistance increases. One physical model that does this is a thermistor. A thermistor is a resistor whose resistance increases with its temperature. These are used in thermostats.

Remember the "energy dissipation" of a resistor? The energy dissipates as heat. So, the temperature of resistor increases as its resistance times the square of the flow through it. To prevent the temperatures of the resistors from going to infinity, we will assume that there is an ambient temperature T_A, and that they tend to the ambient temperature. I will denote by T_e the temperature of resistor e, and I will assume that there is a constant α_e for each resistor so that its resistance

$$r_e = \alpha_e T_e. \tag{18.1}$$

We do not allow temperatures to be negative.



(a) $\Phi(v)$

(b) $\Psi(f)$ when $f = \phi(v)$

Now, assume that we would like to either flow a current between two vertices s and t, or that we have fixed the potentials of s and t. Given the temperature of every resistor at some moment, we can compute all their resistances, and then compute the resulting electrical flow as we did in Lecture 13. Let f_e be the resulting flow on resistor e. The temperature of e will increase by $r_e f_e^2$, and it will also increase in proportion to the difference between its present temperature and the ambient temperature.

This gives us the following differential equation for the change in the temperature of a resistor:

$$\frac{\partial T_e}{\partial t} = r_e f_e^2 - (T_e - T_A). \tag{18.2}$$

Ok, there should probably be some constant multiplying the $(T_e - T_A)$ term. But, since I haven't specified the units of temperature we can just assume that the constant is 1.

By substituting in (18.1) we can eliminate the references to resistance. We thus obtain

$$\frac{\partial T_e}{\partial t} = \alpha_e T_e f_e^2 - (T_e - T_A).$$

There are now two natural questions to ask: does the system converge, and if so, what does it converge to? If we choose to impose a current flow between s and t, the system does not need to converge. For example, consider just one resistor e between vertices s and t with $\alpha_e = 2$. We then find

$$\frac{\partial T_e}{\partial t} = \alpha_e T_e f_e^2 - (T_e - T_A) = 2T_e - (T_e - T_A) = T_e + T_A.$$

So, the temperature of the resistor will go to infinity.

For this reason, I prefer to just fix the voltages of certain vertices. Under these conditions, we can prove that the system will converge. While I do not have time to prove this, we can examine what it will converge to.

If the system converges, that is if the voltages at the nodes converge along with the potential drops and flows across edges, then

$$0 = \frac{\partial T_e}{\partial t} = \alpha_e T_e f_e^2 - (T_e - T_A).$$

To turn this into a relationship between f_e and v_e, we apply the identity $f_e r_e = v_e$, which becomes $f_e \alpha_e T_e = v_e$, to obtain

$$0 = v_e f_e - T_e + T_A.$$

To eliminate the last occurence of T_e, we then multiply by f_e and apply the same identity to produce

$$0 = v_e f_e^2 - v_e/\alpha_e + f_e T_A.$$

The solutions of this equation in f_e are given by

$$f_e = \pm\sqrt{\frac{1}{\alpha_e} + \left(\frac{T_A}{2v_e}\right)^2} - \frac{T_A}{2v_e}.$$

The correct choice of sign is the one that gives this the same sign as v_e:

$$f_e = \frac{1}{2v_e}\left(\sqrt{\frac{(2v_e)^2}{\alpha_e} + T_A^2} - T_A\right). \tag{18.3}$$

When v_e is small this approaches zero, so we define it to be zero when v_e is zero. As v_e becomes large this expression approaches $\alpha_e^{-1/2}$. Similarly, when v_e becomes very negative, this approaches $-\alpha_e^{-1/2}$. If we now define

$$\phi_e(v_e) = \frac{1}{2v_e}\left(\sqrt{\frac{(2v_e)^2}{\alpha_e} + T_A^2} - T_A\right),$$

we see that this function satisfies properties a, b and c. Theorem 18.4.1 then tells us that a stable solution exists.

18.8 Low Temperatures

We now observe that when the ambient temperature is low, a thermistor network produces a minimum s-t cut in a graph. The weights of the edges in the graph are related to α_e. For simplicity, we will just examine the case when all $\alpha_e = 1$. If we take the limit as T_A approaches zero, then the behavior of ϕ_e is

$$\phi_e(v_e) = \begin{cases} 0 & \text{if } v_e = 0 \\ 1 & \text{if } v_e > 0 \\ -1 & \text{if } v_e < 0. \end{cases}$$

We will obtain similar behavior for small T_A: if there is a non-negligible potential drop across an edge, then the flow on that edge will be near 1. So, every edge will either have a flow near 1 or a negligible potential drop. When an edge has a flow near 1, its energy will be near 1. On the other hand, the energy of edges with negligible potential drop will be near 0.

So, in the limit of small temperatures, the energy minimization problem becomes

$$\min_{v : v(s)=0, v(t)=1} \sum_{(a,b) \in E} \left| v(a) - v(b) \right| .$$

One can show that the minimum is achieved when all of the voltages are 0 or 1, in which case the energy is the number of edges going between voltage 0 and 1. That is, the minimum is achieved by a minimum s-t cut.

Part IV

Spectra and Graph Structure

Chapter 19

Independent Sets and Coloring

19.1 Overview

In this lecture we will see how high-frequency eigenvalues of the Laplacian and Adjacency matrix can be related to independent sets and graph coloring. Recall the we number the Laplacian matrix eigenvalues in increasing order:

$$0 = \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n.$$

We call the adjacency matrix eigenvalues $\mu_1, \dots, \mu_n$, and number them in the reverse order:

$$\mu_1 \geq \cdots \geq \mu_n.$$

The reason is that for d-regular graphs, $\mu_i = d - \lambda_i$.

19.2 Graph Coloring and Independent Sets

A coloring of a graph is an assignment of one color to every vertex in a graph so that each edge connects vertices of different colors. We are interested in coloring graphs while using as few colors as possible. Formally, a k-coloring of a graph is a function $c : V \to \{1, \dots, k\}$ so that for all $(u, v) \in V$, $c(u) \neq c(v)$. A graph is k-colorable if it has a k-coloring. The chromatic number of a graph, written $\chi(G)$, is the least k for which G is k-colorable. A graph G is 2-colorable if and only if it is bipartite. Determining whether or not a graph is 3-colorable is an NP-complete problem [] The famous 4-Color Theorem [AH77a, AH77b] says that every planar graph is 4-colorable.

A set of vertices S is *independent* if there are no edges between vertices in S. In particular, each color class in a coloring is an independent set. The size of the largest independent set in a graph, which we call its *independence number* is written $\alpha(G)$. As a k-colorable graph with n vertices must have a color class of size at least n/k,

$$\alpha(G) \geq \frac{n}{\chi(G)}.$$

The problem of finding large independent sets in a graph is NP-Complete, and it is very difficult to even approximate the size of the largest independent set in a graph [FK98]. However, for some carefully chosen graphs spectral analysis provides very good bounds on the sizes of independent sets.

19.3 Hoffman's Bound

One of the first results in spectral graph theory was Hoffman's [Hof70] proof the following upper bound on the size of an independent set in a graph G.

Theorem 19.3.1. *Let $G = (V, E)$ be a d-regular graph, and let μ_n be its smallest adjacency matrix eigenvalue. Then*

$$\alpha(G) \leq n \frac{-\mu_n}{d - \mu_n}.$$

Recall that $\mu_n < 0$. Otherwise this theorem would not make sense. We will prove a generalization of Hoffman's theorem due to Godsil and Newman [GN08]:

Theorem 19.3.2. *Let S be an independent set in G, and let $d_{ave}(S)$ be the average degree of a vertex in S. Then,*

$$|S| \leq n \left(1 - \frac{d_{ave}(S)}{\lambda_n}\right).$$

This is a generalization because in the d-regular case $d_{ave} = d$ and $\lambda_n = d - \mu_n$. So, these bounds are the same for regular graphs:

$$1 - \frac{d_{ave}(S)}{\lambda_n} = \frac{\lambda_n - d}{\lambda_n} = \frac{-\mu_n}{d - \mu_n}.$$

Proof. The Courant-Fischer Theorem tells us that

$$\lambda_n = \max_{\boldsymbol{x}} \frac{\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

Let S be an independent set of vertices, let $\mathbf{1}_S$ be the characteristic vector of S, and let $d(S)$ be the sum of the degrees of vertices in S. Consider the vector

$$\boldsymbol{x} = \mathbf{1}_S - s\mathbf{1},$$

where $s = |S| / n$. As S is independent and $\boldsymbol{L}\mathbf{1} = \mathbf{0}$, we have

$$\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} = \mathbf{1}_S \boldsymbol{L} \mathbf{1}_S = \sum_{a \sim b} (\mathbf{1}_S(a) - \mathbf{1}_S(b))^2 = d(S) = d_{ave}(S) |S|.$$

The reason that we subtracted $s\mathbf{1}$ from $\mathbf{1}_S$ is that this minimizes the norm of the result. We compute

$$\boldsymbol{x}^T \boldsymbol{x} = |S| (1 - s)^2 + (|V| - |S|) s^2 = |S| (1 - 2s + s^2) + |S| s - |S| s^2 = |S| (1 - s) = n(s - s^2).$$

Thus,

$$\lambda_n \geq \frac{d_{ave}(S)\,|S|}{n(s-s^2)} = \frac{d_{ave}(S)sn}{n(s-s^2)} = \frac{d_{ave}(S)}{1-s}.$$

Re-arranging terms, this gives

$$1 - \frac{d_{ave}(S)}{\lambda_n} \geq s,$$

which is equivalent to the claim of the theorem. □

We will use the computation of the norm of $\boldsymbol{x}$ often, so we will make it a claim.

Claim 19.3.3. *Let $S \subseteq V$ have size $s\,|V|$. Then*

$$\|\mathbf{1}_S - s\mathbf{1}\|^2 = s(1-s)\,|V|\,.$$

Claim 19.3.4. *For a vector $\boldsymbol{x}$ of length n, the value of t that minimizes the norm of $\boldsymbol{x} - t\mathbf{1}$ is $t = \mathbf{1}^T\boldsymbol{x}/n$.*

Proof. The derivative of the square of the norm is

$$\frac{d}{dt}\sum_a (\boldsymbol{x}(a) - t)^2 = 2\sum_a (\boldsymbol{x}(a) - t).$$

When the norm is minimized the derivative is zero, which implies

$$\sum_a \boldsymbol{x}(a) = nt.$$

□

19.4 Application to Paley graphs

Let's examine what Hoffman's bound on the size of the largest independent set tells us about Paley graphs.

If G is a Paley graph and S is an independent set, we have $n = p$, $d = (p-1)/2$, and $\lambda_n = (p + \sqrt{p})/2$, so Hoffman's bound tells us that

$$\begin{aligned} |S| &\leq n\left(1 - \frac{d_{ave}(S)}{\lambda_n}\right) \\ &= p\left(1 - \frac{p-1}{p+\sqrt{p}}\right) \\ &= p\left(\frac{\sqrt{p}+1}{p+\sqrt{p}}\right) \\ &= \sqrt{p}. \end{aligned}$$

One can also show that every clique in a Paley graph has size at most $\sqrt{p}$.

A graph is called a k-Ramsey graph if it contains no clique or independent set of size k. It is a challenge to find large k-Ramsey graphs. Equivalently, it is challenging to find k-Ramsey graphs on n vertices for which k is small. In one of the first papers on the Probabilistic Method in Combinatorics, Erdös proved that a random graph on n vertices in which each edge is included with probability $1/2$ is probably $2\log_2 n$ Ramsey [Erd47].

However, constructing explicit Ramsey graphs has proved much more challenging. Until recently, Paley graphs were among the best known. A recent construction of Barak, Rao, Shatltiel and Wigderson [BRSW12] constructs explicit graphs that are $2^{(\log n)^{o(1)}}$ Ramsey.

19.5 Lower Bound on the chromatic number

As a k-colorable graph must have an independent set of size at least n/k, an upper bound on the sizes of independent sets gives a lower bound on its chromatic number. However, this bound is not always a good one.

For example, consider a graph on $2n$ vertices consisting of a clique (complete graph) on n vertices and n vertices of degree 1, each of which is connected to a different vertex in the clique. The chromatic number of this graph is n, because each of the vertices in the clique must have a different color. However, the graph also has an independent set of size n, which would only give a lower bound of 2 on the chromatic number.

Hoffman proved the following lower bound on the chromatic number of a graph that does not require the graph to be regular. Numerically, it is obtained by dividing n by the bound in Theorem 19.3.1. But, the proof is very different because that theorem only applies to regular graphs.

Theorem 19.5.1.

$$\chi(G) \geq \frac{\mu_1 - \mu_n}{-\mu_n} = 1 + \frac{\mu_1}{-\mu_n}.$$

The proof of this theorem relies on the following inequality whose proof we defer to Section 19.6. To state it, we introduce the notation $\lambda_{max}(\boldsymbol{M})$ and $\lambda_{min}(\boldsymbol{M})$ to indicate the largest and smallest eigenvalues of the matrix $\boldsymbol{M}$.

Lemma 19.5.2. *Let*

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{M}_{1,1} & \boldsymbol{M}_{1,2} & \cdots & \boldsymbol{M}_{1,k} \\ \boldsymbol{M}_{1,2}^T & \boldsymbol{M}_{2,2} & \cdots & \boldsymbol{M}_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{M}_{1,k}^T & \boldsymbol{M}_{2,k}^T & \cdots & \boldsymbol{M}_{k,k} \end{bmatrix}$$

be a block-partitioned symmetric matrix with $k \geq 2$. Then

$$(k-1)\lambda_{min}(\boldsymbol{M}) + \lambda_{max}(\boldsymbol{M}) \leq \sum_i \lambda_{max}(\boldsymbol{M}_{i,i}).$$

Proof of Theorem 19.5.1. Let G be a k-colorable graph. After possibly re-ordering the vertices, the adjacency matrix of G can be written

$$\begin{bmatrix} \mathbf{0} & \boldsymbol{M}_{1,2} & \cdots & \boldsymbol{M}_{1,k} \\ \boldsymbol{M}_{1,2}^T & \mathbf{0} & \cdots & \boldsymbol{M}_{2,k} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{M}_{1,k}^T & \boldsymbol{M}_{2,k}^T & \cdots & \mathbf{0} \end{bmatrix}. \tag{19.1}$$

Each block corresponds to a color.

As each diagonal block is all-zero, Lemma 19.5.2 implies

$$(k-1)\lambda_{min}(\boldsymbol{M}) + \lambda_{max}(\boldsymbol{M}) \leq 0.$$

Recalling that $\lambda_{min}(\boldsymbol{M}) = \mu_n < 0$, and $\lambda_{max}(\boldsymbol{M}) = \mu_1$, a little algebra yields

$$1 + \frac{\mu_1}{-\mu_n} \leq k.$$

□

To return to our example of the n clique with n degree-1 vertices attached, I examined an example with $n = 6$. We find $\mu_1 = 5.19$ and $\mu_{12} = -1.62$. This gives a lower bound on the chromatic number of 4.2, which implies a lower bound of 5. We can improve the lower bound by re-weighting the edges of the graph. For example, if we give weight 2 to all the edges in the clique and weight 1 to all the others, we obtain a bound of 5.18, which agrees with the chromatic number of this graph which is 6.

19.6 Proofs for Hoffman's lower bound on chromatic number

To prove Lemma 19.5.2, we begin with the case of $k = 2$. The general case follows from this one by induction. While the lemma in the case $k = 2$ when there are zero blocks on the diagonal follows from Proposition 4.5.4, we require the general statement for induction.

Lemma 19.6.1. *Let*

$$\boldsymbol{A} = \begin{bmatrix} \boldsymbol{B} & \boldsymbol{C} \\ \boldsymbol{C}^T & \boldsymbol{D} \end{bmatrix}$$

be a symmetric matrix. Then

$$\lambda_{min}(\boldsymbol{A}) + \lambda_{max}(\boldsymbol{A}) \leq \lambda_{max}(\boldsymbol{B}) + \lambda_{max}(\boldsymbol{D}).$$

Proof. Let $\boldsymbol{x}$ be an eigenvector of $\boldsymbol{A}$ of eigenvalue $\lambda_{max}(\boldsymbol{A})$. To simplify formulae, let's also assume that $\boldsymbol{x}$ is a unit vector. Write $\boldsymbol{x} = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \end{pmatrix}$, using the same partition as we did for $\boldsymbol{A}$.

We first consider the case in which neither $\boldsymbol{x}_1$ nor $\boldsymbol{x}_2$ is an all-zero vector. In this case, we set

$$\boldsymbol{y} = \begin{pmatrix} \frac{\|\boldsymbol{x}_2\|}{\|\boldsymbol{x}_1\|}\boldsymbol{x}_1 \\ -\frac{\|\boldsymbol{x}_1\|}{\|\boldsymbol{x}_2\|}\boldsymbol{x}_2 \end{pmatrix}.$$

The reader may verify that $\boldsymbol{y}$ is also a unit vector, so

$$\boldsymbol{y}^T A \boldsymbol{y} \geq \lambda_{min}(\boldsymbol{A}).$$

We have

$$\begin{aligned}
\lambda_{max}(\boldsymbol{A}) + \lambda_{min}(\boldsymbol{A}) &\leq \boldsymbol{x}^T A \boldsymbol{x} + \boldsymbol{y}^T A \boldsymbol{y} \\
&= \boldsymbol{x}_1^T \boldsymbol{B} \boldsymbol{x}_1 + \boldsymbol{x}_1^T \boldsymbol{C} \boldsymbol{x}_2 + \boldsymbol{x}_2^T \boldsymbol{C}^T \boldsymbol{x}_1 + \boldsymbol{x}_2^T \boldsymbol{D} \boldsymbol{x}_2 + \\
&\quad + \frac{\|\boldsymbol{x}_2\|^2}{\|\boldsymbol{x}_1\|^2} \boldsymbol{x}_1^T \boldsymbol{B} \boldsymbol{x}_1 - \boldsymbol{x}_1^T \boldsymbol{C} \boldsymbol{x}_2 - \boldsymbol{x}_2^T \boldsymbol{C}^T \boldsymbol{x}_1 + \frac{\|\boldsymbol{x}_1\|^2}{\|\boldsymbol{x}_2\|^2} \boldsymbol{x}_2^T \boldsymbol{D} \boldsymbol{x}_2 \\
&= \boldsymbol{x}_1^T \boldsymbol{B} \boldsymbol{x}_1 + \boldsymbol{x}_2^T \boldsymbol{D} \boldsymbol{x}_2 + \frac{\|\boldsymbol{x}_2\|^2}{\|\boldsymbol{x}_1\|^2} \boldsymbol{x}_1^T \boldsymbol{B} \boldsymbol{x}_1 + \frac{\|\boldsymbol{x}_1\|^2}{\|\boldsymbol{x}_2\|^2} \boldsymbol{x}_2^T \boldsymbol{D} \boldsymbol{x}_2 \\
&\leq \left(1 + \frac{\|\boldsymbol{x}_2\|^2}{\|\boldsymbol{x}_1\|^2}\right) \boldsymbol{x}_1^T \boldsymbol{B} \boldsymbol{x}_1 + \left(1 + \frac{\|\boldsymbol{x}_1\|^2}{\|\boldsymbol{x}_2\|^2}\right) \boldsymbol{x}_2^T \boldsymbol{D} \boldsymbol{x}_2 \\
&\leq \lambda_{max}(\boldsymbol{B}) \left(\|\boldsymbol{x}_1\|^2 + \|\boldsymbol{x}_2\|^2\right) + \lambda_{max}(\boldsymbol{D}) \left(\|\boldsymbol{x}_1\|^2 + \|\boldsymbol{x}_2\|^2\right) \\
&= \lambda_{max}(\boldsymbol{B}) + \lambda_{max}(\boldsymbol{D}),
\end{aligned}$$

as $\boldsymbol{x}$ is a unit vector.

We now return to the case in which $\|\boldsymbol{x}_2\| = 0$ (or $\|\boldsymbol{x}_1\| = 0$, which is really the same case). Lemma 4.3.1 tells us that $\lambda_{max}(\boldsymbol{B}) \leq \lambda_{max}(\boldsymbol{A})$. So, it must be the case that $\boldsymbol{x}_1$ is an eigenvector of eigenvalue $\lambda_{max}(\boldsymbol{A})$ of $\boldsymbol{B}$, and thus $\lambda_{max}(\boldsymbol{B}) = \lambda_{max}(\boldsymbol{A})$. To finish the proof, also observe that Lemma 4.3.1 implies

$$\lambda_{max}(\boldsymbol{D}) \geq \lambda_{min}(\boldsymbol{D}) \geq \lambda_{min}(\boldsymbol{A}).$$

□

Proof of Lemma 19.5.2. For $k = 2$, this is exactly Lemma 19.6.1. For $k > 2$, we apply induction. Let

$$B = \begin{bmatrix} \boldsymbol{M}_{1,1} & \boldsymbol{M}_{1,2} & \cdots & \boldsymbol{M}_{1,k-1} \\ \boldsymbol{M}_{1,2}^T & \boldsymbol{M}_{2,2} & \cdots & \boldsymbol{M}_{2,k-1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{M}_{1,k-1}^T & \boldsymbol{M}_{2,k-1}^T & \cdots & \boldsymbol{M}_{k-1,k-1} \end{bmatrix}.$$

Lemma 4.3.1 now implies.

$$\lambda_{min}(\boldsymbol{B}) \geq \lambda_{min}(\boldsymbol{A}).$$

Applying Lemma 19.6.1 to B and the kth row and column of $\boldsymbol{A}$, we find

$$\begin{aligned}\lambda_{min}(\boldsymbol{A}) + \lambda_{max}(\boldsymbol{A}) &\leq \lambda_{max}(\boldsymbol{B}) + \lambda_{max}(\boldsymbol{M}_{k,k}) \\ &\leq -(k-2)\lambda_{min}(\boldsymbol{B}) + \sum_{i=1}^{k-1} \lambda_{max}(\boldsymbol{M}_{i,i}) + \lambda_{max}(\boldsymbol{M}_{k,k}) \qquad \text{(by induction)} \\ &\leq -(k-1)\lambda_{min}(\boldsymbol{A}) + \sum_{i=1}^{k} \lambda_{max}(\boldsymbol{M}_{i,i}),\end{aligned}$$

which proves the lemma. □

Chapter 20

Graph Partitioning

Computer Scientists are often interested in cutting, partitioning, and finding clusters of vertices in graphs. This usually means finding a set of vertices that is connected to the rest of the graph by a small number of edges. There are many ways of balancing the size of the set of vertices with the number of edges. We will examine isoperimetric ratio and conductance, and will find that they are intimately related to the second-smallest eigenvalue of the Laplacian and the normalized Laplacian. The motivations for measuring these range from algorithm design to data analysis.

20.1 Isoperimetry and λ_2

Let S be a subset of the vertices of a graph. One way of measuring how well S can be separated from the graph is to count the number of edges connecting S to the rest of the graph. These edges are called the *boundary* of S, which we formally define by

$$\partial(S) \stackrel{\text{def}}{=} \{(a, b) \in E : a \in S, b \notin S\}.$$

We are less interested in the total number of edges on the boundary than in the ratio of this number to the size of S itself. For now, we will measure this in the most natural way–by the number of vertices in S. We will call this ratio the *isoperimetric ratio* of S, and define it by

$$\theta(S) \stackrel{\text{def}}{=} \frac{|\partial(S)|}{|S|}.$$

The *isoperimetric ratio* of a graph[1] is the minimum isoperimetric ratio over all sets of at most half the vertices:

$$\theta_G \stackrel{\text{def}}{=} \min_{|S| \leq n/2} \theta(S).$$

We will now derive a lower bound on θ_G in terms of λ_2. We will present an upper bound, known as Cheeger's Inequality in Chapter 21.

[1] Other authors call this the isoperimetric number.

Theorem 20.1.1. *For every $S \subset V$*

$$\theta(S) \geq \lambda_2 (1 - s),$$

where $s = |S| / |V|$. In particular,

$$\theta_G \geq \lambda_2 / 2.$$

Proof. As

$$\lambda_2 = \min_{\boldsymbol{x} : \boldsymbol{x}^T \mathbf{1} = 0} \frac{\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}},$$

for every non-zero $\boldsymbol{x}$ orthogonal to $\mathbf{1}$ we know that

$$\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x} \geq \lambda_2 \boldsymbol{x}^T \boldsymbol{x}.$$

To exploit this inequality, we need a vector related to the set S. A natural choice is $\mathbf{1}_S$, the characteristic vector of S,

$$\mathbf{1}_S(a) = \begin{cases} 1 & \text{if } a \in S \\ 0 & \text{otherwise.} \end{cases}$$

We find

$$\mathbf{1}_S^T \boldsymbol{L}_G \mathbf{1}_S = \sum_{(a,b) \in E} (\mathbf{1}_S(a) - \mathbf{1}_S(b))^2 = |\partial(S)|\,.$$

However, $\boldsymbol{\chi}_S$ is not orthogonal to $\mathbf{1}$. To fix this, use

$$\boldsymbol{x} = \mathbf{1}_S - s\mathbf{1},$$

so

$$\boldsymbol{x}(a) = \begin{cases} 1 - s & \text{for } a \in S, \text{ and} \\ -s & \text{otherwise.} \end{cases}$$

We have $\boldsymbol{x}^T \mathbf{1} = 0$, and

$$\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x} = \sum_{(a,b) \in E} ((\mathbf{1}_S(a) - s) - (\mathbf{1}_S(b) - s))^2 = |\partial(S)|\,.$$

Claim 19.3.3 tells us that the square of the norm of $\boldsymbol{x}$ is

$$\boldsymbol{x}^T \boldsymbol{x} = n(s - s^2).$$

So,

$$\lambda_2 \leq \frac{\mathbf{1}_S^T \boldsymbol{L}_G \mathbf{1}_S}{\mathbf{1}_S^T \mathbf{1}_S} = \frac{|\partial(S)|}{|S|\,(1 - s)}.$$

□

This theorem says that if λ_2 is big, then G is very well connected: the boundary of every small set of vertices is at least λ_2 times something just slightly smaller than the number of vertices in the set.

Re-arranging terms slightly, Theorem 20.1.1 can be stated as

$$\theta(S) = |V| \frac{|\partial(S)|}{|S|\,|V - S|} \geq \lambda_2.$$

20.2 Conductance

The formula for conductance is appropriately normalized for weighted graphs. Instead of counting the edges on the boundary, we count the sum of their weights. Similarly, the denominator depends upon the sum of the weighted degrees of the vertices in S. We write $d(S)$ for the sum of the degrees of the vertices in S. Thus, $d(V)$ is twice the sum of the weights of edges in the graph. For a set of edges F, we write $w(F)$ for the sum of the weights of edges in F. We define the *conductance* of S to be

$$\phi(S) \stackrel{\text{def}}{=} \frac{w(\partial(S))}{\min(d(S), d(V-S))}.$$

Note that many similar, although sometimes slightly different, definitions appear in the literature. For example, we would instead use

$$\frac{d(V)w(\partial(S))}{d(S)d(V-S)},$$

which appears below in (20.3).

We define the *conductance* of a graph G to be

$$\phi_G \stackrel{\text{def}}{=} \min_{S \subset V} \phi(S).$$

The conductance of a graph is more useful in many applications than the isoperimetric number. I usually find that conductance is the more useful quantity when you are concerned about edges, and that isoperimetric ratio is most useful when you are concerned about vertices. Conductance is particularly useful when studying random walks in graphs.

20.3 The Normalized Laplacian

It seems natural to try to relate the conductance to the following generalized Rayleigh quotient:

$$\frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}}. \tag{20.1}$$

If we make the change of variables

$$\boldsymbol{D}^{1/2} \boldsymbol{y} = \boldsymbol{x},$$

then this ratio becomes

$$\frac{\boldsymbol{x}^T \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}}.$$

That is an ordinary Rayleigh quotient, which we understand a little better. The matrix in the middle is called the normalized Laplacian (see [Chu97]). We reserve the letter $\boldsymbol{N}$ for this matrix:

$$\boldsymbol{N} \stackrel{\text{def}}{=} \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2}.$$

This matrix often proves more useful when examining graphs in which nodes have different degrees. We will let $0 = \nu_1 \leq \nu_2 \leq \cdots \leq \nu_n$ denote the eigenvalues of $\boldsymbol{N}$.

The eigenvector of eigenvalue 0 of $\boldsymbol{N}$ is $\boldsymbol{d}^{1/2}$, by which I mean the vector whose entry for vertex u is the square root of the degree of u. Observe that

$$\boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2}\boldsymbol{d}^{1/2} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{1} = \boldsymbol{D}^{-1/2}\boldsymbol{0} = \boldsymbol{0}.$$

The eigenvector of ν_2 is given by

$$\arg\min_{\boldsymbol{x}\perp\boldsymbol{d}^{1/2}} \frac{\boldsymbol{x}^T\boldsymbol{N}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}}.$$

Transferring back into the variable $\boldsymbol{y}$, and observing that

$$\boldsymbol{x}^T\boldsymbol{d}^{1/2} = \boldsymbol{y}^T D^{1/2}\boldsymbol{d}^{1/2} = \boldsymbol{y}^T\boldsymbol{d},$$

we find

$$\nu_2 = \min_{\boldsymbol{y}\perp\boldsymbol{d}} \frac{\boldsymbol{y}^T\boldsymbol{L}\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y}}.$$

The conductance is related to ν_2 as the isoperimetric number is related to λ_2:

$$\nu_2/2 \leq \phi_G. \tag{20.2}$$

Lemma 20.3.1. *For every $S \subset V$,*

$$\frac{w(\partial(S))d(V)}{d(S)d(V-S)} \geq \nu_2.$$

Proof. We would again like to again use $\boldsymbol{1}_S$ as a test vector. But, it is not orthogonal to $\boldsymbol{d}$. To fix this, we subtract a constant. Set

$$\boldsymbol{y} = \boldsymbol{1}_S - \sigma\boldsymbol{1},$$

where

$$\sigma = d(S)/d(V).$$

You should now check that $\boldsymbol{y}^T\boldsymbol{d} = 0$:

$$\boldsymbol{y}^T\boldsymbol{d} = \boldsymbol{1}_S^T\boldsymbol{d} - \sigma\boldsymbol{1}^T\boldsymbol{d} = d(S) - (d(S)/d(V))d(V) = 0.$$

We already know that

$$\boldsymbol{y}^T\boldsymbol{L}\boldsymbol{y} = \left|\partial(S)\right|.$$

It remains to compute $\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y}$. If you remember the previous computation like this, you would guess that it is $d(S)(1-\sigma) = d(S)d(V-S)/d(V)$, and you would be right:

$$\begin{aligned}
\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y} &= \sum_{u\in S} d(u)(1-\sigma)^2 + \sum_{u\notin S} d(u)\sigma^2 \\
&= d(S)(1-\sigma)^2 + d(V-S)\sigma^2 \\
&= d(S) - 2d(S)\sigma + d(V)\sigma^2 \\
&= d(S) - d(S)\sigma \\
&= d(S)d(V-S)/d(V).
\end{aligned}$$

So,

$$\nu_2 \leq \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}} = \frac{w(\partial(S)) d(V)}{d(S) d(V - S)}. \tag{20.3}$$

□

Corollary 20.3.2. *For every $S \subset V$,*

$$\phi(S) \geq \nu_2 / 2.$$

Proof. As the larger of $d(S)$ and $d(V - S)$ is at least half of $d(V)$, we find

$$\nu_2 \leq 2 \frac{w(\partial(S))}{\min(d(S), d(V - S))}.$$

□

20.4 Notes

There are many variations on the definitions used in this chapter. For example, sometimes one wants to measure the number of *vertices* on the boundary of a set, rather than the number of edges. The ratio of the number of boundary vertices to internal vertices is often called *expansion*. But, authors are not consistent about these and related terms. *Cut ratio* is sometimes used instead of *isoperimetric ratio*. When reading anything in this area, be sure to check the formulas for the definitions.

Chapter 21

Cheeger's Inequality

In the last chapter we learned that $\phi(S) \geq \nu_2/2$ for every $S \subseteq V$. Cheeger's inequality is a partial converse. It says that there exists a set of vertices S for which

$$\phi(S) \leq \sqrt{2\nu_2},$$

and provides an algorithm for using the eigenvector of ν_2 to find such a set.

Cheeger [Che70] first proved his famous inequality for manifolds. Many discrete versions of Cheeger's inequality were proved in the late 80's [SJ89, LS88, AM85, Alo86, Dod84, Var85]. Some of these consider the walk matrix instead of the normalized Laplacian, and some consider the isoperimetic ratio instead of conductance. The proof in this Chapter follows an approach developed by Trevisan [Tre11].

21.1 Cheeger's Inequality

Cheeger's inequality proves that if we have a vector $\boldsymbol{y}$, orthogonal to $\boldsymbol{d}$, for which the generalized Rayleigh quotient (20.1) is small, then one can obtain a set of small conductance from $\boldsymbol{y}$. We obtain such a set by carefully choosing a real number τ, and setting

$$S_{tau} = \{a : \boldsymbol{y}(a) \leq \tau\}.$$

We should think of deriving $\boldsymbol{y}$ from an eigenvector of ν_2 of the normalized Laplacian. If $\boldsymbol{\psi}_2$ is an eigenvector of ν_2, then $\boldsymbol{y} = \boldsymbol{D}^{1/2}\boldsymbol{\psi}_2$ is orthogonal to $\boldsymbol{d}$ and the generalized Rayleigh quotient (20.1) of $\boldsymbol{y}$ with respect to $\boldsymbol{L}$ and $\boldsymbol{D}$ equals ν_2. But, the theorem can make use of any vector that is orthogonal to $\boldsymbol{d}$ that makes the generalized Rayleigh quotient small. In fact, we prefer vectors that are *centered* with respect to $\boldsymbol{d}$.

Definition 21.1.1. *A vector $\boldsymbol{y}$ is centered with respect to $\boldsymbol{d}$ if*

$$\sum_{a:\boldsymbol{y}(a)>0} \boldsymbol{d}(a) \leq \boldsymbol{d}(V)/2 \quad \textit{and} \quad \sum_{a:\boldsymbol{y}(a)<0} \boldsymbol{d}(a) \leq \boldsymbol{d}(V)/2.$$

By renumbering the vertices, we may assume without loss of generality that

$$\boldsymbol{y}(1) \leq \boldsymbol{y}(2) \leq \cdots \leq \boldsymbol{y}(n).$$

To center $\boldsymbol{y}$, let j be the least number for which

$$\sum_{a=1}^{j} \boldsymbol{d}(a) \geq d(V)/2.$$

We then set

$$\boldsymbol{z} = \boldsymbol{y} - \boldsymbol{y}(j)\mathbf{1}.$$

This vector $\boldsymbol{z}$ satisfies $\boldsymbol{z}(j) = 0$. And, the following lemma tells us that

$$\frac{\boldsymbol{z}^T \boldsymbol{L} \boldsymbol{z}}{\boldsymbol{z}^T \boldsymbol{D} \boldsymbol{z}} \leq \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}}.$$

Lemma 21.1.2. *Let $\boldsymbol{v}_s = \boldsymbol{y} + s\mathbf{1}$. Then, the minimum of $\boldsymbol{v}_s^T \boldsymbol{D} \boldsymbol{v}_s^T$ is achieved at the s for which $\boldsymbol{v}_s^T \boldsymbol{d} = 0$.*

Proof. The derivative with respect to s is $2\boldsymbol{d}^T \boldsymbol{v}_s$, and this is zero at the minimum. □

Theorem 21.1.3. *Let G be a weighted graph, let $\boldsymbol{L}$ be its Laplacian, and let $\boldsymbol{d}$ be its vector of weighted degrees. Let $\boldsymbol{z}$ be a vector that is centered with respect to $\boldsymbol{d}$. Then, there is a number τ for which the set $S_\tau = \{a : \boldsymbol{z}(a) < \tau\}$ satisfies*

$$\phi(S_\tau) \leq \sqrt{2 \frac{\boldsymbol{z}^T \boldsymbol{L} \boldsymbol{z}}{\boldsymbol{z}^T \boldsymbol{D} \boldsymbol{z}}}.$$

We assume without loss of generality that

$$\boldsymbol{z}(1)^2 + \boldsymbol{z}(n)^2 = 1.$$

This can be achieved by multiplying $\boldsymbol{z}$ by a constant. We begin our proof of Cheeger's inequality by defining

$$\rho = \frac{\boldsymbol{z}^T \boldsymbol{L} \boldsymbol{z}}{\boldsymbol{z}^T \boldsymbol{D} \boldsymbol{z}}.$$

So, we need to show that there is a τ for which $\phi(S_\tau) \leq \sqrt{2\rho}$.

Recall that

$$\phi(S) = \frac{w(\partial(S))}{\min(d(S), d(V - S))}.$$

We will define a distribution on τ for which we can prove that

$$\mathbb{E}\left[w(\partial(S_\tau))\right] \leq \sqrt{2\rho}\, \mathbb{E}\left[\min(d(S_\tau), d(V - S_\tau))\right].$$

This implies[1] that there is some τ for which

$$w(\partial(S_\tau)) \leq \sqrt{2\rho}\, \min(d(S_\tau), d(V - S_\tau)),$$

which means $\phi(S) \leq \sqrt{2\rho}$.

To switch from working with $\boldsymbol{y}$ to working with $\boldsymbol{z}$, define We will set $S_\tau = \{a : \boldsymbol{z}(a) \leq \tau\}$. Trevisan had the remarkable idea of choosing τ between $\boldsymbol{z}(1)$ and $\boldsymbol{z}(n)$ with probability density $2\,|t|$. That is, the probability that τ lies in the interval $[a, b]$ is

$$\int_{t=a}^{b} 2\,|t|\,.$$

To see that the total probability is 1, observe that

$$\int_{t=\boldsymbol{z}(1)}^{\boldsymbol{z}(n)} 2\,|t| = \int_{t=\boldsymbol{z}(1)}^{0} 2\,|t| + \int_{t=0}^{\boldsymbol{z}(n)} 2\,|t| = \boldsymbol{z}(n)^2 + \boldsymbol{z}(1)^2 = 1,$$

as $\boldsymbol{z}(1) \leq 0 \leq \boldsymbol{z}(n)$.

Similarly, the probability that τ lies in the interval $[a, b]$ is

$$\int_{t=a}^{b} 2\,|t| = \mathsf{sgn}(b)b^2 - \mathsf{sgn}(a)a^2,$$

where

$$\mathsf{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x = 0\text{, and} \\ -1 & \text{if } x < 0. \end{cases}$$

Lemma 21.1.4.

$$\mathbb{E}_t\left[w(\partial(S_\tau))\right] = \sum_{(a,b) \in E} w_{a,b} Pr_t\left[(a,b) \in \partial(S_\tau)\right] \leq \sum_{(a,b) \in E} w_{a,b}\,|\boldsymbol{z}(a) - \boldsymbol{z}(b)|\,(|\boldsymbol{z}(a)| + |\boldsymbol{z}(b)|). \quad (21.1)$$

Proof. An edge (a, b) with $\boldsymbol{z}(a) \leq \boldsymbol{z}(b)$ is on the boundary of S if

$$\boldsymbol{z}(a) \leq \tau < \boldsymbol{z}(b).$$

The probability that this happens is

$$\mathsf{sgn}(\boldsymbol{z}(b))\boldsymbol{z}(b)^2 - \mathsf{sgn}(\boldsymbol{z}(a))\boldsymbol{z}(a)^2 = \begin{cases} \left|\boldsymbol{z}(a)^2 - \boldsymbol{z}(b)^2\right| & \text{when } \mathsf{sgn}(a) = \mathsf{sgn}(b), \\ \boldsymbol{z}(a)^2 + \boldsymbol{z}(b)^2 & \text{when } \mathsf{sgn}(a) \neq \mathsf{sgn}(b). \end{cases}$$

We now show that both of these terms are upper bounded by

$$|\boldsymbol{z}(a) - \boldsymbol{z}(b)|\,(|\boldsymbol{z}(a)| + |\boldsymbol{z}(b)|).$$

[1] If this is not immediately clear, note that it is equivalent to assert that $\mathbb{E}\left[\sqrt{2\rho}\min(d(S), d(V - S)) - w(\partial(S))\right] \geq 0$, which means that there must be some S for which the expression is non-negative.

Regardless of the signs,

$$\left|\boldsymbol{z}(a)^2 - \boldsymbol{z}(b)^2\right| = \left|(\boldsymbol{z}(a) - \boldsymbol{z}(b))(\boldsymbol{z}(a) + \boldsymbol{z}(b))\right| \leq \left|\boldsymbol{z}(a) - \boldsymbol{z}(b)\right| (\left|\boldsymbol{z}(a)\right| + \left|\boldsymbol{z}(b)\right|).$$

When $\mathsf{sgn}(a) = -\mathsf{sgn}(b)$,

$$\boldsymbol{z}(a)^2 + \boldsymbol{z}(b)^2 \leq (\boldsymbol{z}(a) - \boldsymbol{z}(b))^2 = \left|\boldsymbol{z}(a) - \boldsymbol{z}(b)\right| (\left|\boldsymbol{z}(a)\right| + \left|\boldsymbol{z}(b)\right|).$$

□

We now derive a formula for the expected denominator of ϕ.

Lemma 21.1.5.

$$\mathbb{E}_t \left[\min(d(S_\tau), d(V - S_\tau))\right] = \boldsymbol{z}^T \boldsymbol{D} \boldsymbol{z}.$$

Proof. Observe that

$$\mathbb{E}_t \left[d(S_\tau)\right] = \sum_a \Pr_t \left[a \in S_\tau\right] d(a) = \sum_a \Pr_t \left[\boldsymbol{z}(a) \leq \tau\right] d(a).$$

The result of our centering of $\boldsymbol{z}$ at j is that

$$\begin{aligned} \tau < 0 &\implies d(S) = \min(d(S), d(V - S)), \quad \text{and} \\ \tau \geq 0 &\implies d(V - S) = \min(d(S), d(V - S)). \end{aligned}$$

That is, for $a < j$, a is in the smaller set if $\tau < 0$; and, for $a \geq j$, a is in the smaller set if $\tau \geq 0$. So,

$$\begin{aligned} \mathbb{E}_t \left[\min(d(S_\tau), d(V - S_\tau))\right] &= \sum_{a<j} \Pr\left[\boldsymbol{z}(a) < \tau \text{ and } \tau < 0\right] d(a) + \sum_{a \geq j} \Pr\left[\boldsymbol{z}(a) > \tau \text{ and } \tau \geq 0\right] d(a) \\ &= \sum_{a<j} \Pr\left[\boldsymbol{z}(a) < \tau < 0\right] d(a) + \sum_{a \geq j} \Pr\left[\boldsymbol{z}(a) > \tau \geq 0\right] d(a) \\ &= \sum_{a<j} \boldsymbol{z}(a)^2 d(a) + \sum_{a \geq j} \boldsymbol{z}(a)^2 d(a) \\ &= \sum_a \boldsymbol{z}(a)^2 d(a) \\ &= \boldsymbol{z}^T \boldsymbol{D} \boldsymbol{z}. \end{aligned}$$

□

Recall that our goal is to prove that

$$\mathbb{E}\left[w(\partial(S_\tau))\right] \leq \sqrt{2\rho}\, \mathbb{E}\left[\min(d(S_\tau), d(V - S_\tau))\right],$$

and we know that

$$\mathbb{E}_t \left[\min(d(S_\tau), d(V - S_\tau))\right] = \boldsymbol{z}^T \boldsymbol{D} \boldsymbol{z}$$

and that

$$\mathbb{E}_t\left[w(\partial(S_\tau))\right] \leq \sum_{(a,b)\in E} w_{a,b}\left|\boldsymbol{z}(a)-\boldsymbol{z}(b)\right|(\left|\boldsymbol{z}(a)\right|+\left|\boldsymbol{z}(b)\right|).$$

We may use the Cauchy-Schwartz inequality to upper bound the term above by

$$\sqrt{\sum_{(a,b)\in E} w_{a,b}(\boldsymbol{z}(a)-\boldsymbol{z}(b))^2}\sqrt{\sum_{(a,b)\in E} w_{a,b}(|\boldsymbol{z}(a)|+|\boldsymbol{z}(b)|)^2}. \tag{21.2}$$

We have defined ρ so that the term under the left-hand square root is at most

$$\boldsymbol{z}^T\boldsymbol{L}\boldsymbol{z} \leq \rho\boldsymbol{z}^T\boldsymbol{D}\boldsymbol{z}.$$

To bound the right-hand square root, we observe

$$\sum_{(a,b)\in E} w_{a,b}(|\boldsymbol{z}(a)|+|\boldsymbol{z}(b)|)^2 \leq 2\sum_{(a,b)\in E} w_{a,b}\left(\boldsymbol{z}(a)^2+\boldsymbol{z}(b)^2\right) = 2\sum_a \boldsymbol{z}(a)^2 d(a) = 2\boldsymbol{z}^T\boldsymbol{D}\boldsymbol{z}.$$

Putting all these inequalities together yields

$$\begin{aligned}\mathbb{E}\left[w(\partial(S))\right] &\leq \sqrt{\rho\boldsymbol{z}^T\boldsymbol{D}\boldsymbol{z}}\sqrt{2\boldsymbol{z}^T\boldsymbol{D}\boldsymbol{z}}\\ &= \sqrt{2\rho}\boldsymbol{z}^T\boldsymbol{D}\boldsymbol{z}\\ &= \sqrt{2\rho}\mathbb{E}\left[\min\left(d(S),d(V-S)\right)\right].\end{aligned}$$

Chapter 22

Local Graph Clustering

Local graph clustering algorithms discover small clusters of low conductance near a given input vertex. Imagine that a graph has a cluster S that is not too big–$\boldsymbol{d}(S)$ is small relative to $\boldsymbol{d}(V)$–and that has low conductance. Also imagine that we know some vertex $a \in S$. Local clustering algorithms give us a way of computing a cluster nearby S of similar size and conductance. They are not guaranteed to work for all $a \in S$. But, we can show that they work for "most" $a \in S$, where we have to measure "most" by weighted degree. In this chapter, we will see an elegant analysis due to Kwok, Lau and Lee [KLL16] of a random-walk based local graph clustering algorithm suggested by Spielman and Teng [ST04, ST13]

Most local clustering algorithms can be implemented to run on unweighted graphs in time depending on $\boldsymbol{d}(S)$, rather that on the size of the graph. This means that they can find the cluster without having to examine the entire graph! Many of the developments in these algorithms have improved the running time, the size of the set returned, and the conductance of the set returned. The end of the chapter contains pointers to major advances in these algorithms.

In this chapter, we focus on proving that we can find a cluster approximately as good as S, without optimizing parameters or run time.

22.1 The Algorithm

The input to the algorithm is a target set size, s, a conductance bound ϕ, and a seed vertex, a. We will prove that if G contains a set S with $\boldsymbol{d}(S) \leq s \leq \boldsymbol{d}(V)/32$ and $\phi(S) \leq \phi$, then there is an $a \in S$ such that when the algorithm is run with these parameters, it will return a set T with $\boldsymbol{d}(T) \leq 16s$ and $\phi(T) \leq \sqrt{8 \ln(8s)\phi}$. For the rest of this chapter we will assume that G does contain a set S that satisfies these conditions.

Here is the algorithm.

1. Set $\boldsymbol{p}_0 = \boldsymbol{\delta}_a$.

2. Set $t = 1/2\phi$ (we will assume that t is an integer).

3. Set $\boldsymbol{y} = \boldsymbol{D}^{-1}\widetilde{\boldsymbol{W}}^{t}\boldsymbol{p}_0$.
4. Return the set of the form $T_\tau = \{b : \boldsymbol{y}(b) > \tau\}$ that has least conductance among those with $\boldsymbol{d}(T_\tau) \leq 8s$.

Recall that the stable distribution of the random walk on a graph is $\boldsymbol{d}/(\boldsymbol{1}^T\boldsymbol{d})$. So, to measure how close a probability distribution $\boldsymbol{p}$ is to the stable distribution, we could ask how close $\boldsymbol{D}^{-1}\boldsymbol{p}$ is to being constant. In this chapter, we will measure this by the generalized Rayleigh quotient

$$\frac{\boldsymbol{p}^T\boldsymbol{D}^{-1}\boldsymbol{L}\boldsymbol{D}^{-1}\boldsymbol{p}}{\boldsymbol{p}^T\boldsymbol{D}^{-1}\boldsymbol{p}}.$$

When we want to apply Cheeger's inequality, we will change variables to $\boldsymbol{y} = \boldsymbol{D}^{-1}\boldsymbol{p}$. In these variables, the above quotient becomes

$$\frac{\boldsymbol{y}^T\boldsymbol{L}\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y}}.$$

We will work with the lazy random walk matrix

$$\widetilde{\boldsymbol{W}} = \frac{1}{2}(I + \boldsymbol{W}) = \frac{1}{2}(I + \boldsymbol{M}\boldsymbol{D}^{-1}).$$

22.2 Good choices for a

We will say that a vertex $a \in S$ is *good for S* if

$$\frac{\boldsymbol{d}(a)}{\boldsymbol{d}(S)} \geq \frac{1}{2\,|S|} \quad \text{and} \quad \boldsymbol{1}_S^T\widetilde{\boldsymbol{W}}^{t}\boldsymbol{\delta}_a \geq 1/2.$$

The second inequality says that after t steps the lazy walk that starts at a will be in S with probability at least 1/2. In this section we show that S contains a good vertex. We will then show that the local clustering algorithm succeeds if it begins at a good vertex.

Consider the distribution on vertices that corresponds to choosing a vertex at random from S with probability proportional to its degree:

$$\boldsymbol{p}_S \overset{\text{def}}{=} \begin{cases} \boldsymbol{d}(a)/\boldsymbol{d}(S), & \text{for } a \in S \\ 0, & \text{otherwise.} \end{cases}$$

The following lemma says that if we start a walk from a random vertex in S chosen with probability proportional to degree, then the probability it is outside S on the tth step of the lazy walk is at most $t\phi(S)/2$.

Lemma 22.2.1. *Let S be a set with $\boldsymbol{d}(S) \leq \boldsymbol{d}(V)/2$. Let $\boldsymbol{p}_t = \widetilde{\boldsymbol{W}}^{t}\boldsymbol{p}_S$. Then*

$$\boldsymbol{1}_{V-S}^T\boldsymbol{p}_t \leq t\phi(S)/2.$$

Proof. We will upper bound the probability that the lazy walk leaves S in each step by $\phi(S)/2$. In the first step, the probability that the lazy walk leaves S is exactly the sum over vertices a in S of the probability the walk begins at a times the probability it follows an edge to a vertex not in S:

$$\sum_{a \in S} \boldsymbol{p}_S(a) \frac{1}{2} \sum_{\substack{b \sim a \\ b \notin S}} \frac{w_{a,b}}{\boldsymbol{d}(a)} = \frac{1}{2} \frac{1}{\boldsymbol{d}(S)} \sum_{\substack{a \in S \\ b \notin S}} w_{a,b} = \frac{1}{2} \frac{w(\partial(S))}{\boldsymbol{d}(S)} = \frac{1}{2}\phi(S).$$

We now wish to show that in every future step the probability that the lazy walk leaves S is at most this large. To this end, let $\boldsymbol{p}_0 = \boldsymbol{p}_S$, and define

$$\boldsymbol{p}_i = \widetilde{\boldsymbol{W}} \boldsymbol{p}_{i-1}.$$

We now show by induction that for every $a \in V$, $\boldsymbol{p}_i(a) \leq \boldsymbol{d}(a)/\boldsymbol{d}(S)$. This is true for $\boldsymbol{p}_0$, and in fact the inequality is tight for $a \in S$. To establish the induction, note that all entries of $\widetilde{\boldsymbol{W}}$ and $\boldsymbol{p}_{i-1}$ are nonnegative. So, the assumption that $\boldsymbol{p}_{i-1}$ is entrywise at most $\boldsymbol{d}(a)/\boldsymbol{d}(S)$ implies that for $a \in S$

$$\boldsymbol{\delta}_a^T \boldsymbol{p}_i = \boldsymbol{\delta}_a^T \widetilde{\boldsymbol{W}} \boldsymbol{p}_{i-1} \leq \boldsymbol{\delta}_a^T \widetilde{\boldsymbol{W}} \boldsymbol{d}/\boldsymbol{d}(S) = \boldsymbol{\delta}_a^T \boldsymbol{d}/\boldsymbol{d}(S) = \boldsymbol{d}(a)/\boldsymbol{d}(S).$$

Thus, the probability that the walk transitions from a vertex in S to a vertex not in S at step i satisfies

$$\sum_{a \in S} \boldsymbol{p}_i(a) \frac{1}{2} \sum_{\substack{b \sim a \\ b \notin S}} \frac{w_{a,b}}{\boldsymbol{d}(a)} \leq \sum_{a \in S} \boldsymbol{p}_S(a) \frac{1}{2} \sum_{\substack{b \sim a \\ b \notin S}} \frac{w_{a,b}}{\boldsymbol{d}(a)} = \frac{1}{2}\phi(S).$$

□

Lemma 22.2.2. *The set S contains a good vertex a.*

Proof. After we expand $\boldsymbol{p}_S$ in an elementary unit basis as

$$\boldsymbol{p}_S = \sum_{a \in S} \frac{\boldsymbol{d}(a)}{\boldsymbol{d}(S)} \boldsymbol{\delta}_a,$$

Lemma 22.2.1 tells us that

$$\sum_{a \in S} \frac{\boldsymbol{d}(a)}{\boldsymbol{d}(S)} \boldsymbol{1}_{V-S}^T \widetilde{\boldsymbol{W}}^t \boldsymbol{\delta}_a \leq t\phi(S)/2.$$

Define f_a to be the indicator[1]for the event that

$$\boldsymbol{1}_{V-S}^T \widetilde{\boldsymbol{W}}^t \boldsymbol{\delta}_a > t\phi(S)$$

and let b_a be the indicator for the event that

$$\frac{\boldsymbol{d}(a)}{\boldsymbol{d}(S)} < \frac{1}{2\,|S|}.$$

[1] That is, $f_a = 1$ if the event holds, and $f_a = 0$ otherwise.

By an application of what is essentially Markov's inequality, we conclude

$$\sum_{a \in S} \frac{\boldsymbol{d}(a)}{\boldsymbol{d}(S)} f_a < \frac{1}{2}.$$

As

$$\sum_{a \in S} \frac{\boldsymbol{d}(a)}{\boldsymbol{d}(S)} b_a < \sum_{a \in S} \frac{1}{2\,|S|} b_a \leq \sum_{a \in S} \frac{1}{2\,|S|} = 1/2.$$

Thus, there is a vertex for which neither f_a nor b_a hold. As

$$\mathbf{1}_S^T \widetilde{\boldsymbol{W}}^t \boldsymbol{\delta}_a = 1 - \mathbf{1}_{V-S}^T \widetilde{\boldsymbol{W}}^t \boldsymbol{\delta}_a$$

and $t\phi(S) \leq 1/2$, such a vertex is good. □

By slightly loosening the constants in the definition of "good", we could prove that most vertices of S are good, where "most" is defined by sampling with probability proportional to degree.

22.3 Bounding the $\boldsymbol{D}$-norm

Claim 22.3.1. *For a probability vector* $\boldsymbol{p}$*,*

$$\boldsymbol{p}^T \boldsymbol{D}^{-1} \boldsymbol{p} \geq \frac{(\mathbf{1}_S^T \boldsymbol{p})^2}{\boldsymbol{d}(S)}.$$

Proof. Write

$$\mathbf{1}_S^T \boldsymbol{p} = \sum_{a \in S} \boldsymbol{p}(a) = \sum_{a \in S} \sqrt{\boldsymbol{d}(a)} \left(\boldsymbol{p}(a) / \sqrt{\boldsymbol{d}(a)} \right)$$

and apply the Cauchy-Schwartz inequality to conclude

$$\left(\mathbf{1}_S^T \boldsymbol{p}\right)^2 \leq \left(\sum_{a \in S} \boldsymbol{d}(a) \right) \left(\sum_{a \in S} \boldsymbol{p}(a)^2 / \boldsymbol{d}(a) \right) \leq \left(\sum_{a \in S} \boldsymbol{d}(a) \right) \left(\sum_{a} \boldsymbol{p}(a)^2 / \boldsymbol{d}(a) \right) = \boldsymbol{d}(S) \boldsymbol{p}^T \boldsymbol{D}^{-1} \boldsymbol{p}.$$

□

If a is good for S, then $\mathbf{1}_S^T \boldsymbol{p}_t \geq 1/2$, and so

$$\boldsymbol{p}_t^T \boldsymbol{D}^{-1} \boldsymbol{p}_t \geq \frac{1}{4\boldsymbol{d}(S)}.$$

22.4 Bounding the Generalized Rayleigh Quotient

The following lemma allows us to measure how close a walk is to convergence merely in terms of the quadratic form $\boldsymbol{p}_t^T \boldsymbol{D}^{-1} \boldsymbol{p}_t$ and the number of steps t.

Lemma 22.4.1. *Let $\boldsymbol{p}_t = \widetilde{\boldsymbol{W}}^t \boldsymbol{p}_0$ for some probability vector $\boldsymbol{p}_0$. Then*

$$\frac{\boldsymbol{p}_t^T \boldsymbol{D}^{-1} \boldsymbol{L} \boldsymbol{D}^{-1} \boldsymbol{p}_t}{\boldsymbol{p}_t^T \boldsymbol{D}^{-1} \boldsymbol{p}_t} \leq \frac{1}{t} \ln \left(\frac{\boldsymbol{p}_0 \boldsymbol{D}^{-1} \boldsymbol{p}_0}{\boldsymbol{p}_t \boldsymbol{D}^{-1} \boldsymbol{p}_t} \right).$$

The proof of Lemma 22.4.1 rests on the following standard inequality.

Theorem 22.4.2. *[Power Means Inequality] For $k > h > 0$, nonnegative numbers $w_1, \ldots, w_n$ that sum to 1, and nonnegative numbers $\lambda_1, \ldots, \lambda_n$,*

$$\left(\sum_{i=1}^n w_i \lambda_i^k \right)^{1/k} \geq \left(\sum_{i=1}^n w_i \lambda_i^h \right)^{1/h}$$

Proof of Lemma 22.4.1. Define

$$\boldsymbol{z}_t = \boldsymbol{D}^{-1/2} \boldsymbol{p}_t,$$

so

$$\boldsymbol{p}_t^T \boldsymbol{D}^{-1} \boldsymbol{L} \boldsymbol{D}^{-1} \boldsymbol{p}_t = \boldsymbol{z}_t^T \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2} \boldsymbol{z}_t = \boldsymbol{z}_t^T \boldsymbol{N} \boldsymbol{z}_t, \quad \text{and} \quad \boldsymbol{p}_t^T \boldsymbol{D}^{-1} \boldsymbol{p}_t = \boldsymbol{z}_t^T \boldsymbol{z}_t.$$

Write $\boldsymbol{z}_0 = \boldsymbol{D}^{-1/2} \boldsymbol{p}_0$ in the eigenbasis of $\boldsymbol{N}$ as

$$\boldsymbol{z}_0 = \sum_i c_i \boldsymbol{\psi}_i,$$

and set

$$\gamma = \frac{1}{\sum_i c_i^2} = \frac{1}{\boldsymbol{z}_0^T \boldsymbol{z}_0}$$

so that $\sum_i \gamma c_i^2 = 1$. We have

$$\boldsymbol{z}_t = \boldsymbol{D}^{-1/2} \widetilde{\boldsymbol{W}}^t \boldsymbol{p}_0 = \boldsymbol{D}^{-1/2} \widetilde{\boldsymbol{W}}^t \boldsymbol{D}^{1/2} \boldsymbol{z}_0 = \left(\boldsymbol{D}^{-1/2} \widetilde{\boldsymbol{W}} \boldsymbol{D}^{1/2} \right)^t \boldsymbol{z}_0.$$

Recall from Chapter 10 that

$$\boldsymbol{D}^{-1/2} \widetilde{\boldsymbol{W}} \boldsymbol{D}^{1/2} = \boldsymbol{I} - \frac{1}{2} \boldsymbol{N},$$

and that the eigenvalues of these matrices are related by

$$\nu_i = 2 - 2\omega_i.$$

Thus,

$$\boldsymbol{z}_t = \sum_i c_i \omega_i^t \boldsymbol{\psi}_i,$$

and

$$\boldsymbol{z}_t^T \boldsymbol{N} \boldsymbol{z}_t = 2 \sum_i c_i^2 \nu_i \omega_i^{2t} \leq 2 \sum_i c_i^2 \omega_i^{2t} - 2 \sum_i c_i^2 \omega_i^{2t+1}.$$

Thus,

$$\begin{aligned} \frac{\gamma \boldsymbol{z}_t^T \boldsymbol{N} \boldsymbol{z}_t}{\gamma \boldsymbol{z}_t^T \boldsymbol{z}_t} &= \frac{2 \sum_i \gamma c_i^2 \omega_i^{2t} - 2 \sum_i \gamma c_i^2 \omega_i^{2t+1}.}{\sum_i \gamma c_i^2 \omega_i^{2t}} \\ &= 2 - 2 \frac{\sum_i \gamma c_i^2 \omega_i^{2t+1}}{\sum_i \gamma c_i^2 \omega_i^{2t}}. \end{aligned}$$

To upper bound this last term, we recall that $\sum_i \gamma c_i^2 = 1$ and apply the Power Means Inequality to show

$$\begin{aligned} \left(\sum_i \gamma c_i^2 \omega_i^{2t+1} \right)^{1/(2t+1)} &\geq \left(\sum_i \gamma c_i^2 \omega_i^{2t} \right)^{1/(2t)} && \Longrightarrow \\ \left(\sum_i \gamma c_i^2 \omega_i^{2t+1} \right) &\geq \left(\sum_i \gamma c_i^2 \omega_i^{2t} \right)^{1+1/(2t)} && \Longrightarrow \\ \frac{\sum_i \gamma c_i^2 \omega_i^{2t+1}}{\sum_i \gamma c_i^2 \omega_i^{2t}} &\geq \left(\sum_i \gamma c_i^2 \omega_i^{2t} \right)^{1/(2t)} . \end{aligned}$$

This implies

$$\begin{aligned} 2 - 2 \frac{\sum_i \gamma c_i^2 \omega_i^{2t+1}}{\sum_i \gamma c_i^2 \omega_i^{2t}} &\leq 2 - 2 \left(\sum_i \gamma c_i^2 \omega_i^{2t} \right)^{1/2t} \\ &= 2 - 2 \left(\frac{\boldsymbol{z}_t^T \boldsymbol{z}_t}{\boldsymbol{z}_0^T \boldsymbol{z}_0} \right)^{1/2t} . \end{aligned}$$

To finish the proof, let $R = \frac{\boldsymbol{z}_t^T \boldsymbol{z}_t}{\boldsymbol{z}_0^T \boldsymbol{z}_0}$, and note that for all R

$$R^{1/2t} = \exp(-\ln(1/R)/2t) \geq 1 - \ln(1/R)/2t.$$

So,

$$2 - 2 \left(\frac{\boldsymbol{z}_t^T \boldsymbol{z}_t}{\boldsymbol{z}_0^T \boldsymbol{z}_0} \right)^{1/2t} \leq 2 - 2 \left(1 - \ln(1/R)/2t\right) = \ln(1/R)/t = \frac{1}{t} \ln \left(\frac{\boldsymbol{z}_0^T \boldsymbol{z}_0}{\boldsymbol{z}_t^T \boldsymbol{z}_t} \right).$$

□

For a vertex a that is good for S,

$$\frac{\boldsymbol{p}_t \boldsymbol{D}^{-1} \boldsymbol{p}_t}{\boldsymbol{p}_0 \boldsymbol{D}^{-1} \boldsymbol{p}_0} \geq \frac{1}{4 \boldsymbol{d}(S) \boldsymbol{p}_0 \boldsymbol{D}^{-1} \boldsymbol{p}_0} \geq \frac{\boldsymbol{d}(a)}{4 \boldsymbol{d}(S)} \geq \frac{1}{8 \left| S \right|};$$

so

$$\frac{\boldsymbol{p}_t^T \boldsymbol{D}^{-1} \boldsymbol{L} \boldsymbol{D}^{-1} \boldsymbol{p}_t}{\boldsymbol{p}_t^T \boldsymbol{D}^{-1} \boldsymbol{p}_t} \leq \frac{\ln(8 \left| S \right|)}{t} \leq 2 \phi \ln(8 \left| S \right|).$$

22.5 Rounding

To apply Cheeger's inequality, Theorem 21.1.3, we first change variables from $\boldsymbol{p}_t$ to $\boldsymbol{y} \overset{\text{def}}{=} \boldsymbol{D}^{-1}\boldsymbol{p}_t$. As $\boldsymbol{1}^T\boldsymbol{p}_t = 1$, the vector $\boldsymbol{y}$ satisfies $\boldsymbol{d}^T\boldsymbol{y} = 1$, and

$$\frac{\boldsymbol{p}_t^T \boldsymbol{D}^{-1}\boldsymbol{L}\boldsymbol{D}^{-1}\boldsymbol{p}_t}{\boldsymbol{p}_t^T \boldsymbol{D}^{-1}\boldsymbol{p}_t} = \frac{\boldsymbol{y}^T\boldsymbol{L}\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y}}.$$

So that we can be sure that the algorithm underlying Theorem 21.1.3 will find a set T that is not too big, we will round to zero all the small entries of $\boldsymbol{y}$ and call the result $\boldsymbol{x}$. While this is not necessary for the algorithm, it does facilitate analysis.

Define

$$\boldsymbol{x}(a) = \max(0, \boldsymbol{y}(a) - 1/16s). \tag{22.1}$$

If $s \leq \boldsymbol{d}(V)/32$, then $\boldsymbol{x}$ will be balanced with respect to $\boldsymbol{d}$. This is because at most half its entries (measured by degree) will be positive. Formally,

$$\sum_{a:\boldsymbol{y}(a)>1/16s} \boldsymbol{d}(a) = \sum_{a:\boldsymbol{p}_t(a)>\boldsymbol{d}(a)/16s} \boldsymbol{d}(a) < \sum_{a:\boldsymbol{p}_t(a)>\boldsymbol{d}(a)/16s} 16s\boldsymbol{p}_t(a) \leq \sum_a 16s\boldsymbol{p}_t(a) \leq 16s.$$

As Cheeger's inequality will produce a set of the form

$$T_\tau = \{a : \boldsymbol{y}(a) > \tau\},$$

this set will satisify $\boldsymbol{d}(T_\tau) \leq 16s$.

Lemma 22.5.1. *Let $\boldsymbol{y}$ be a vector such that $\boldsymbol{d}^T\boldsymbol{y} = 1$ and define the vector $\boldsymbol{x}$ by*

$$\boldsymbol{x}(a) = \max(0, \boldsymbol{y}(a) - \epsilon).$$

Then, $\boldsymbol{x}^T\boldsymbol{D}\boldsymbol{x} \geq \boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y} - 2\epsilon$.

Proof. We observe that for every number y and ϵ,

$$\max(0, y - \epsilon)^2 \geq y^2 - 2y\epsilon :$$

If $y \leq \epsilon$ then $y^2 - 2y\epsilon < 0$, and for $y > \epsilon$, $(y - \epsilon)^2 = y^2 - 2y + \epsilon^2$. Thus,

$$\begin{aligned}
\boldsymbol{x}^T\boldsymbol{D}\boldsymbol{x} &= \sum_a \boldsymbol{d}(a)\max(0, \boldsymbol{y}(a) - \epsilon)^2 \\
&\geq \sum_a \boldsymbol{d}(a)\boldsymbol{y}(a)^2 - 2\boldsymbol{y}(a)\epsilon \\
&= \boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y} - 2\epsilon\sum_a \boldsymbol{d}(a)\boldsymbol{y}(a) \\
&= \boldsymbol{y}^T\boldsymbol{D}\boldsymbol{y} - 2\epsilon.
\end{aligned}$$

□

If a is good for S and $\boldsymbol{p}_0 = \boldsymbol{\delta}_a$, then

$$\boldsymbol{x}^T \boldsymbol{D} \boldsymbol{x} \geq \boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y} - 1/8 \boldsymbol{d}(S) \geq \boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}/2.$$

Moreover, as shifting $\boldsymbol{y}$ and rounding entries to zero can not increase the length of any edge,

$$\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x} \leq \boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}.$$

Together these imply

$$\frac{\boldsymbol{x}^T \boldsymbol{L} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{D} \boldsymbol{x}} \leq 2 \frac{\boldsymbol{y}^T \boldsymbol{L} \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{D} \boldsymbol{y}} \leq 4 \ln(8 \left| S \right|) \phi.$$

As $\boldsymbol{y}$ is balanced with respect to $\boldsymbol{d}$, we may apply Cheeger's inequality to obtain a set T of conductance at most

$$\sqrt{8 \ln(8 \left| S \right|) \phi}.$$

22.6 Notes

Explain where these come from, and give some references to where they are used in practice.

Chapter 23

Spectral Partitioning in a Stochastic Block Model

In this chapter, show how eigenvectors can be used to partition graphs drawn from certain natural models. These are called *stochastic block models* or a *planted partition model*, depending on community and application.

The simplest model of this form is for the graph bisection problem. This is the problem of partitioning the vertices of a graph into two equal-sized sets while minimizing the number of edges bridging the sets. To create an instance of the planted bisection problem, we first choose a partition of the vertices into equal-sized sets X and Y. When then choose probabilities $p > q$, and place edges between vertices with the following probabilities:

$$\Pr\left[(u,v) \in E\right] = \begin{cases} p & \text{if } u \in X \text{ and } v \in X \\ p & \text{if } u \in Y \text{ and } v \in Y \\ q & \text{otherwise.} \end{cases}$$

The expected number of edges crossing between X and Y will be $q\left|X\right|\left|Y\right|$. If p is sufficiently larger than q, for example if $p = 1/2$ and $q = p - 24/\sqrt{n}$, we will show that the partition can be approximately recovered from the second eigenvector of the adjacency matrix of the graph. The result, of course, extends to over values of p and q. This will be a crude version of an analysis of McSherry [McS01].

If p is too close to q, then the partition given by X and Y will not be the smallest. For example, if $q = p - \epsilon/\sqrt{n}$ for small ϵ then one cannot hope to distinguish between X and Y.

McSherry analyzed more general models than this, including planted coloring problems, and sharp results have been obtained in a rich line of work. See, for example, [MNS14, DKMZ11, BLM15, Mas14, Vu14].

McSherry's analysis treats the adjacency matrix of the generated graph as a perturbation of one ideal probability matrix. In the probability matrix the second eigenvector provides a clean partition of the two blocks. McSherry shows that the difference between the generated matrix and the ideal one is small, and so the generated matrix can be viewed as a small perturbation of the idea one. He then uses matrix perturbation theory to show that the second eigenvector of the generated matrix will probably be close to the second eigenvector of the original, and so it reveals the partition. The idea of using perturbation theory to analyze random objects generated from nice models has been very powerful.

Warning: stochastic block models have been the focus of a lot of research lately, and there are now very good algorithms for solving problems on graphs generated from these models. But, these are just models and very little real data resembles that produced by these models. So, there is no reason to believe that algorithms that are optimized for these models will be useful in practice. Nevertheless, some of them are.

23.1 The Perturbation Approach

As long as we don't tell our algorithm, we can choose $X = \{1, \dots, n/2\}$ and $Y = \{n/2+1, \dots, n\}$. Let's do this for simplicity.

Define the matrix

$$\boldsymbol{A} = \begin{bmatrix} 0 & p & \cdots & p & p & q & q & \cdots & q & q \\ p & 0 & \cdots & p & p & q & q & \cdots & q & q \\ \vdots & & & & & & & & & \vdots \\ p & p & \cdots & 0 & p & q & q & \cdots & q & q \\ p & p & \cdots & p & 0 & q & q & \cdots & q & q \\ q & q & \cdots & q & q & 0 & p & \cdots & p & p \\ q & q & \cdots & q & q & p & 0 & \cdots & p & p \\ \vdots & & & & & & & & & \vdots \\ q & q & \cdots & q & q & p & p & \cdots & 0 & p \\ q & q & \cdots & q & q & p & p & \cdots & p & 0 \end{bmatrix} = \begin{bmatrix} p\boldsymbol{J}_{n/2} & q\boldsymbol{J}_{n/2} \\ q\boldsymbol{J}_{n/2} & p\boldsymbol{J}_{n/2} \end{bmatrix} - p\boldsymbol{I}_n,$$

where we write $\boldsymbol{J}_{n/2}$ for the square all-1s matrix of size $n/2$.

The adjacency matrix of the planted partition graph is obtained by setting $\boldsymbol{M}(a,b) = 1$ with probability $\boldsymbol{A}(a,b)$, subject to $\boldsymbol{M}(a,b) = \boldsymbol{M}(b,a)$ and $\boldsymbol{M}(a,a) = 0$. So, this is a random graph, but the probabilities of some edges are different from others.

We will study a very simple algorithm for finding an approximation of the planted bisection: compute $\boldsymbol{\psi}_2$, the eigenvector of the second-largest eigenvalue of $\boldsymbol{M}$. Then, set $S = \{a : \boldsymbol{\psi}_2(a) < 0\}$. We guess that S is one of the sets in the bisection. We will show that under reasonable conditions on p and q, S will be mostly right. For example, we might consider $p = 1/2$ and $q = 1/2 - 12/\sqrt{n}$. Intuitively, the reason this works is that $\boldsymbol{M}$ is a slight perturbation of $\boldsymbol{A}$, and so the eigenvectors of $\boldsymbol{M}$ should look like the eigenvectors of $\boldsymbol{A}$.

To simplify some formulas, we henceforth work with

$$\widehat{\boldsymbol{M}} \stackrel{\text{def}}{=} \boldsymbol{M} + p\boldsymbol{I} \quad \text{and} \quad \widehat{\boldsymbol{A}} \stackrel{\text{def}}{=} \boldsymbol{A} + p\boldsymbol{I}$$

Note that the eigenvectors of $\widehat{\boldsymbol{M}}$ and $\boldsymbol{M}$ are the same, and so are those of $\widehat{\boldsymbol{A}}$ and $\boldsymbol{A}$. So considering $\widehat{\boldsymbol{M}}$ and $\widehat{\boldsymbol{A}}$ won't change our analysis at all. The matrix $\widehat{\boldsymbol{A}}$ is convenient because it has rank 2. We now consider the difference between $\widehat{\boldsymbol{M}}$ and $\widehat{\boldsymbol{A}}$:

$$\boldsymbol{R} = \widehat{\boldsymbol{M}} - \widehat{\boldsymbol{A}} = \boldsymbol{M} - \boldsymbol{A}.$$

To see why this would be useful, let's look at the eigenvectors of $\widehat{\boldsymbol{A}}$. Of course, the constant vectors are eigenvectors of $\widehat{\boldsymbol{A}}$. We have

$$\widehat{\boldsymbol{A}}\mathbf{1} = \frac{n}{2}(p+q)\mathbf{1},$$

and so the corresponding eigenvalue is

$$\alpha_1 \overset{\text{def}}{=} \frac{n}{2}(p+q).$$

The second eigenvector of $\widehat{\boldsymbol{A}}$ has two values: one on X and one on Y. Let's be careful to make this a unit vector. We take

$$\boldsymbol{\phi}_2(a) = \begin{cases} \frac{1}{\sqrt{n}} & a \in X \\ -\frac{1}{\sqrt{n}} & a \in Y. \end{cases}$$

Then,

$$\widehat{\boldsymbol{A}}\boldsymbol{\phi}_2 = \frac{n}{2}(p-q)\boldsymbol{\phi}_2,$$

and the corresponding eigenvalue is

$$\alpha_2 \overset{\text{def}}{=} \frac{n}{2}(p-q).$$

As $\widehat{\boldsymbol{A}}$ has rank 2, all the other eigenvalues of $\widehat{\boldsymbol{A}}$ are zero.

For (a,b) in the same component,

$$\Pr\left[\boldsymbol{R}(a,b) = 1-p\right] = p \quad \text{and}$$
$$\Pr\left[\boldsymbol{R}(a,b) = -p\right] = 1-p,$$

and for (a,b) in different components,

$$\Pr\left[\boldsymbol{R}(a,b) = 1-q\right] = q \quad \text{and}$$
$$\Pr\left[\boldsymbol{R}(a,b) = -q\right] = 1-q.$$

We can use bounds similar to that proved in Chapter 8 to show that it is unlikely that $\boldsymbol{R}$ has large norm. The bounds that we proved on the norm of a matrix in which entries are chosen from $\{1-p, -p\}$ applies equally well if each entry (a,b) is chosen from $\{1-q_{a,b}, -q_{a,b}\}$ as long as $q_{a,b} < p$ and all have expectation 0, because (8.2) still applies. For a sharp result, we appeal to a theorem of Vu [Vu07, Theorem 1.4], which implies the following.

Theorem 23.1.1. *There exist constants c_1 and c_2 such that with probability approaching 1,*

$$\|\boldsymbol{R}\| \leq 2\sqrt{p(1-p)n} + c_1(p(1-p)n)^{1/4}\ln n,$$

provided that

$$p \geq c_2\frac{\ln^4 n}{n}.$$

We use a crude corollary of this result.

Corollary 23.1.2. *There exists a constant c_0 such that with probability approaching 1,*

$$\|\boldsymbol{R}\| \leq 3\sqrt{pn},$$

provided that

$$p \geq c_0 \frac{\ln^4 n}{n}.$$

In fact, Alon, Krivelevich and Vu [AKV02] prove that the probability that the norm of $\boldsymbol{R}$ exceeds this value by more than t is exponentially small in t. However, we will not need that fact for this lecture.

23.2 Perturbation Theory for Eigenvectors

Let $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_n$ be the eigenvalues of $\widehat{\boldsymbol{M}}$, and let $\alpha_1 > \alpha_2 > 0 = \alpha_3 = \cdots = \alpha_n$ be the eigenvalues of $\widehat{\boldsymbol{A}}$. Weyl's inequality, which one can prove using the Courant-Fischer theorem, says that

$$|\mu_i - \alpha_i| \leq \|R\|. \tag{23.1}$$

So, we can view μ_2 as a perturbation of α_2. We need a stronger fact, which is that we can view $\boldsymbol{\psi}_2$ as a perturbation of $\boldsymbol{\phi}_2$.

The Davis-Kahan theorem [DK70] says that $\boldsymbol{\psi}_2$ will be close to $\boldsymbol{\phi}_2$, in angle, if the norm of $\boldsymbol{R}$ is significantly less than the distance between α_2 and the other eigenvalues of $\widehat{\boldsymbol{A}}$. That is, the eigenvector does not move too much if its corresponding eigenvalue is isolated.

Theorem 23.2.1. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be symmetric matrices. Let $\boldsymbol{R} = \boldsymbol{A} - \boldsymbol{B}$. Let $\alpha_1 \geq \cdots \geq \alpha_n$ be the eigenvalues of $\boldsymbol{A}$ with corresponding eigenvectors $\boldsymbol{\phi}_1, \ldots, \boldsymbol{\phi}_n$ and let Let $\beta_1 \geq \cdots \geq \beta_n$ be the eigenvalues of $\boldsymbol{B}$ with corresponding eigenvectors $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n$. Let θ_i be the angle between $\pm\boldsymbol{\psi}_i$ and $\pm\boldsymbol{\phi}_i$. Then,*

$$\sin 2\theta_i \leq \frac{2\|R\|}{\min_{j \neq i} |\alpha_i - \alpha_j|}.$$

The angle is never more than $\pi/2$, because this theorem is bounding the angle between the eigenspaces rather than a particular choice of eigenvectors. We will prove and use a slightly weaker statement in which we replace 2θ with θ.

23.3 Partitioning

Consider

$$\boldsymbol{\delta} = \boldsymbol{\psi}_2 - \boldsymbol{\phi}_2,$$

and let θ be the angle between them. For every vertex a that is misclassified by $\boldsymbol{\psi}_2$, we have $|\boldsymbol{\delta}(a)| \geq \frac{1}{\sqrt{n}}$. So, if $\boldsymbol{\psi}_2$ misclassifies k vertices, then

$$\|\boldsymbol{\delta}\| \geq \sqrt{\frac{k}{n}}.$$

As $\boldsymbol{\phi}_2$ and $\boldsymbol{\psi}_2$ are unit vectors, we may apply the crude inequality

$$\|\boldsymbol{\delta}\| \leq \sqrt{2} \sin \theta$$

(the $\sqrt{2}$ disappears as θ gets small).

To combine this with the perturbation bound, we assume $q > p/3$, and find

$$\min_{j \neq 2} |\alpha_2 - \alpha_j| = \frac{n}{2}(p-q).$$

Assuming that $\|R\| \leq 3\sqrt{pn}$, we find

$$\sin \theta \leq \frac{2\|\boldsymbol{R}\|}{\frac{n}{2}(p-q)} \leq \frac{2 \cdot 3\sqrt{pn}}{\frac{n}{2}(p-q)} = \frac{12\sqrt{p}}{\sqrt{n}(p-q)}.$$

So, the number k of misclassified vertices satisfies

$$\sqrt{\frac{k}{n}} \leq \frac{\sqrt{2} \cdot 12\sqrt{p}}{\sqrt{n}(p-q)},$$

which implies

$$k \leq \frac{288p}{(p-q)^2}.$$

So, we expect to misclassify at most a constant number of vertices if p and q remain constant as n grows large. An interesting case to consider is $p = 1/2$ and $q = p - 24/\sqrt{n}$. This gives

$$\frac{288p}{(p-q)^2} = \frac{n}{4},$$

so we expect to misclassify at most a constant fraction of the vertices. Of course, once one gets most of the vertices correct is should be possible to use them to better classify the rest. Many of the advances in the study of algorithms for this problem involve better and more rigorous ways of doing this.

23.4 Proof of the Davis-Kahan Theorem

For simplicity, we will prove a statement that is weaker by a factor of 2.

Proof of a weak version of Theorem 23.2.1. By considering the matrices $\boldsymbol{A} - \alpha_i I$ and $\boldsymbol{B} - \alpha_i I$ instead of $\boldsymbol{A}$ and $\boldsymbol{B}$, we can assume that $\alpha_i = 0$. As the theorem is vacuous if α_i has multiplicity more than 1, we may also assume that α_i has multiplicity 1 as an eigenvalue, and that $\boldsymbol{\psi}_i$ is a unit vector in the nullspace of $\boldsymbol{B}$.

Our assumption that $\alpha_i = 0$ also leads to $|\beta_i| \leq \|R\|$ by Weyl's inequality (23.1).

Expand $\boldsymbol{\psi}_i$ in the eigenbasis of $\boldsymbol{A}$, as

$$\boldsymbol{\psi}_i = \sum_j c_j \boldsymbol{\phi}_j, \qquad \text{where } c_j = \boldsymbol{\phi}_j^T \boldsymbol{\psi}_i.$$

Setting

$$\delta = \min_{j \neq i} |\alpha_j| \,,$$

we compute

$$\begin{aligned}
\|\boldsymbol{A}\boldsymbol{\psi}_i\|^2 &= \sum_j c_j^2 \alpha_j^2 \\
&\geq \sum_{j \neq i} c_j^2 \delta^2 \\
&= \delta^2 \sum_{j \neq i} c_j^2 \\
&= \delta^2 (1 - c_i^2) \\
&= \delta^2 \sin^2 \theta_i .
\end{aligned}$$

On the other hand,

$$\|\boldsymbol{A}\boldsymbol{\psi}_i\| = \|(\boldsymbol{B} + \boldsymbol{R})\boldsymbol{\psi}_i\| \leq \|\boldsymbol{B}\boldsymbol{\psi}_i\| + \|\boldsymbol{R}\boldsymbol{\psi}_i\| = \beta_i + \|\boldsymbol{R}\boldsymbol{\psi}_i\| \leq 2 \|\boldsymbol{R}\| \,.$$

So,

$$\sin \theta_i \leq \frac{2 \|\boldsymbol{R}\|}{\delta}.$$

□

It may seem surprising that the amount by which eigenvectors move depends upon how close their respective eigenvalues are to the other eigenvalues. However, this dependence is necessary. To see why, first consider the matrices

$$\begin{bmatrix} 1+\epsilon & 0 \\ 0 & 1 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 1 & 0 \\ 0 & 1+\epsilon \end{bmatrix}.$$

While these two matrices are very close, their leading eigenvectors are $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$, which are 90 degrees from each other.

The heart of the problem is that there is no unique eigenvector of an eigenvalue that has multiplicity greater than 1.

23.5 Further Reading

If you would like to know more about bounding norms and eigenvalues of random matrices, I recommend [Ver10] and [Tro12].

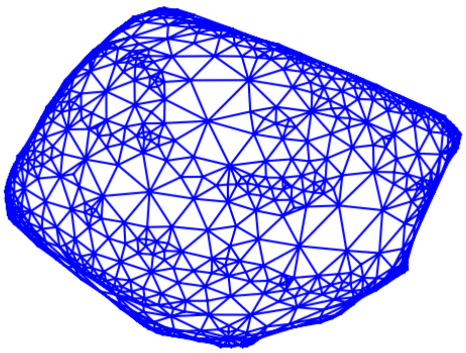
Chapter 24

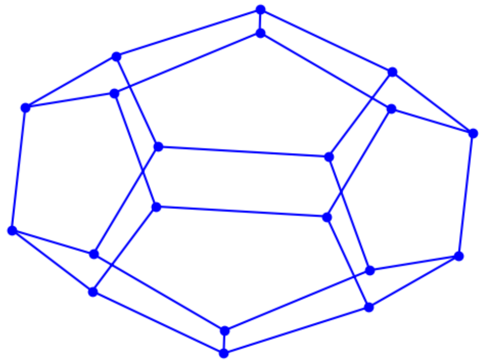
Nodal Domains

Lecture 7 from September 19, 2018

24.1 Overview

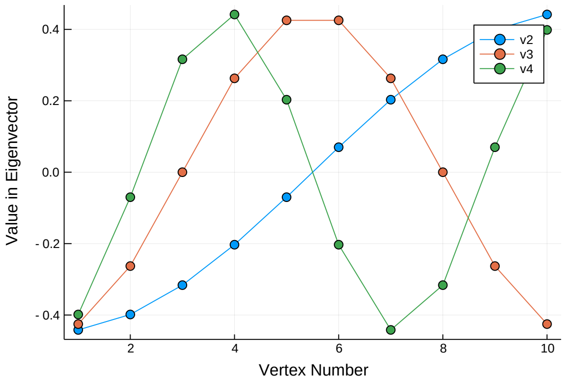
In today's lecture we will justify some of the behavior we observed when using eigenvectors to draw graphs in the first lecture. First, recall some of the drawings we made of graphs:

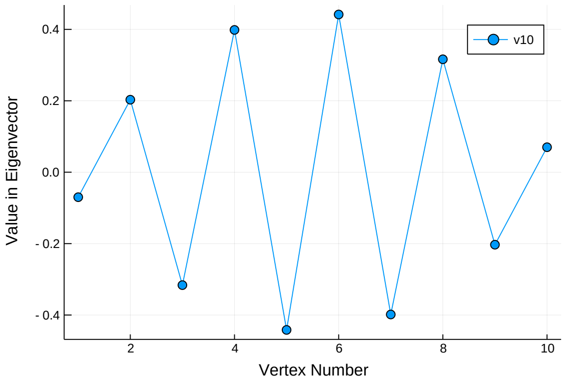




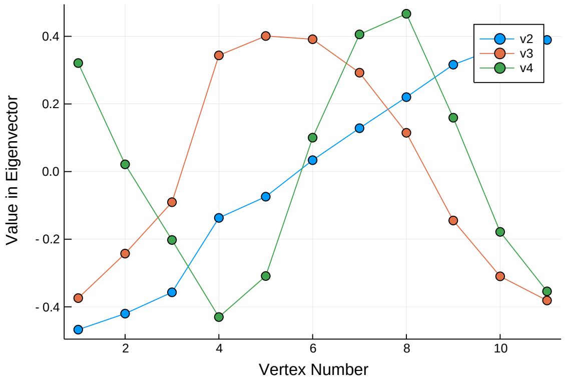
We will show that the subgraphs obtained in the right and left halfs of each image are connected.

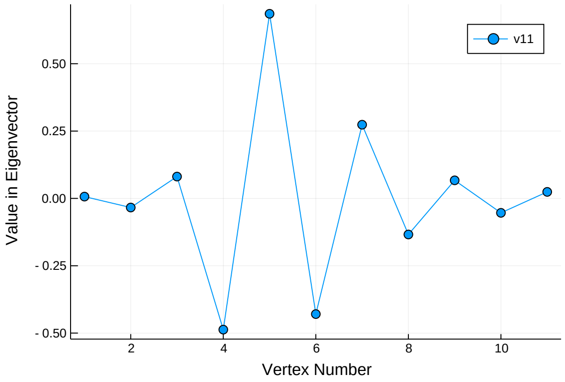
Path graphs exhibited more interesting behavior: their kth eigenvector changes sign k times:



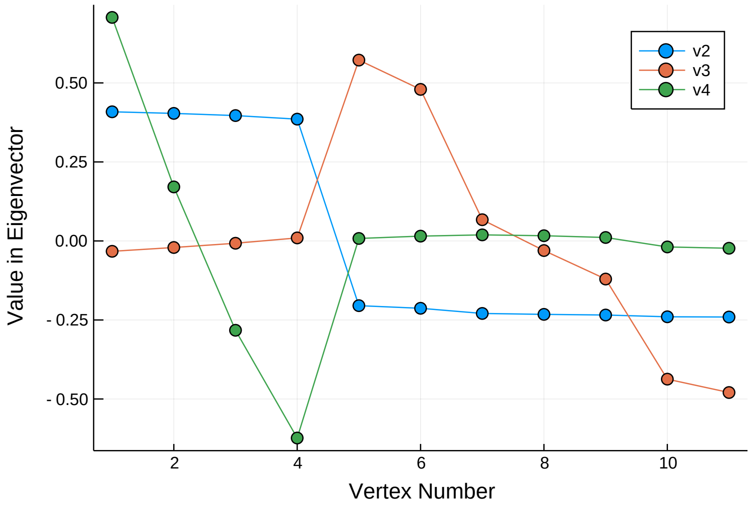


Here are the analogous plots for a path graph with edge weights randomly chosen in $[0, 1]$:





Here are the first few eigenvectors of another:



```
Random.seed!(1)
M = spdiagm(1=>rand(10))
M = M + M'
L = lap(M)
E = eigen(Matrix(L))
Plots.plot(E.vectors[:,2],label="v2",marker = 5)
Plots.plot!(E.vectors[:,3],label="v3",marker = 5)
Plots.plot!(E.vectors[:,4],label="v4",marker = 5)
xlabel!("Vertex Number")
ylabel!("Value in Eigenvector")
savefig("rpath2v24.pdf")
```

We see that the kth eigenvector still changes sign k times. We will see that this always happens. These are some of Fiedler's theorems about "nodal domains". Nodal domains are the connected parts of a graph on which an eigenvector is negative or positive.

24.2 Sylverter's Law of Interia

Let's begin with something obvious.

Claim 24.2.1. *If A is positive semidefinite, then so is $B^T AB$ for every matrix B.*

Proof. For any x,

$$x^T B^T ABx = (Bx)^T A(Bx) \geq 0,$$

since A is positive semidefinite. □

In this lecture, we will make use of Sylvester's law of intertia, which is a powerful generalization of this fact. I will state and prove it now.

Theorem 24.2.2 (Sylvester's Law of Intertia)**.** *Let A be any symmetric matrix and let B be any non-singular matrix. Then, the matrix BAB^T has the same number of positive, negative and zero eigenvalues as A.*

Note that if the matrix B were orthonormal, or if we used B^{-1} in place of B^T, then these matrices would have the same eigenvalues. What we are doing here is different, and corresponds to a change of variables.

Proof. It is clear that $\boldsymbol{A}$ and $\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$ have the same rank, and thus the same number of zero eigenvalues.

We will prove that $\boldsymbol{A}$ has at least as many positive eigenvalues as $\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$. One can similarly prove that that $\boldsymbol{A}$ has at least as many negative eigenvalues, which proves the theorem.

Let $\gamma_1, \dots, \gamma_k$ be the positive eigenvalues of $\boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T$ and let Y_k be the span of the corresponding eigenvectors. Now, let S_k be the span of the vectors $\boldsymbol{B}^T\boldsymbol{y}$, for $\boldsymbol{y} \in Y_k$. As $\boldsymbol{B}$ is non-singluar, S_k has dimension k. Let $\alpha_1 \geq \cdots \geq \alpha_n$ be the eigenvalues of $\boldsymbol{A}$. By the Courant-Fischer Theorem, we have

$$\alpha_k = \max_{\substack{S \subseteq \mathbb{R}^n \\ \dim(S)=k}} \min_{\boldsymbol{x} \in S} \frac{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} \geq \min_{\boldsymbol{x} \in S_k} \frac{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = \min_{\boldsymbol{y} \in Y_k} \frac{\boldsymbol{y}^T \boldsymbol{B}\boldsymbol{A}\boldsymbol{B}^T \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{B}\boldsymbol{B}^T \boldsymbol{y}} \geq \frac{\gamma_k \boldsymbol{y}^T \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{B}\boldsymbol{B}^T \boldsymbol{y}} > 0.$$

So, $\boldsymbol{A}$ has at least k positive eigenvalues (The point here is that the denominators are always positive, so we only need to think about the numerators.)

To finish, either apply the symmetric argument to the negative eigenvalues, or apply the same argument with B^{-1}. □

24.3 Weighted Trees

We will now examine a theorem of Fiedler [Fie75].

Theorem 24.3.1. *Let T be a weighted tree graph on n vertices, let $\boldsymbol{L}_T$ have eigenvalues $0 = \lambda_1 < \lambda_2 \cdots \leq \lambda_n$, and let $\boldsymbol{\psi}_k$ be an eigenvector of λ_k. If there is no vertex u for which $\boldsymbol{\psi}_k(u) = 0$, then there are exactly $k-1$ edges for which $\boldsymbol{\psi}_k(u)\boldsymbol{\psi}_k(v) < 0$.*

One can extend this theorem to accomodate zero entries and prove that the eigenvector changes $k-1$ times. We will just prove this theorem for weighted path graphs.

Our analysis will rest on an understanding of Laplacians of paths that are allowed to have negative edges weights.

Lemma 24.3.2. *Let $\boldsymbol{M}$ be the Laplacian matrix of a weighted path that can have negative edge weights:*

$$\boldsymbol{M} = \sum_{1 \leq a < n} w_{a,a+1} \boldsymbol{L}_{a,a+1},$$

where the weights $w_{a,a+1}$ are non-zero and we recall that $\boldsymbol{L}_{a,b}$ is the Laplacian of the edge (a,b). The number of negative eigenvalues of $\boldsymbol{M}$ equals the number of negative edge weights.

Proof. Note that

$$\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} = \sum_{(u,v)\in E} w_{u,v} (\boldsymbol{x}(u) - \boldsymbol{x}(v))^2.$$

We now perform a change of variables that will diagonalize the matrix $\boldsymbol{M}$. Let $\boldsymbol{\delta}(1) = \boldsymbol{x}(1)$, and for every $a > 1$ let $\boldsymbol{\delta}(a) = \boldsymbol{x}(a) - \boldsymbol{x}(a-1)$.

Every variable $\boldsymbol{x}(1), \dots, \boldsymbol{x}(n)$ can be expressed as a linear combination of the variables $\boldsymbol{\delta}(1), \dots, \boldsymbol{\delta}(n)$. In particular,

$$\boldsymbol{x}(a) = \boldsymbol{\delta}(1) + \boldsymbol{\delta}(2) + \cdots + \boldsymbol{\delta}(a).$$

So, there is a square matrix $\boldsymbol{L}$ of full rank such that

$$\boldsymbol{x} = \boldsymbol{L}\boldsymbol{\delta}.$$

By Sylvester's law of intertia, we know that

$$\boldsymbol{L}^T \boldsymbol{M} \boldsymbol{L}$$

has the same number of positive, negative, and zero eigenvalues as $\boldsymbol{M}$. On the other hand,

$$\boldsymbol{\delta}^T \boldsymbol{L}^T \boldsymbol{M} \boldsymbol{L} \boldsymbol{\delta} = \sum_{1 \leq a < n} w_{a,a+1} (\boldsymbol{\delta}(v))^2.$$

So, this matrix clearly has one zero eigenvalue, and as many negative eigenvalues as there are negative $w_{a,a+1}$. □

Proof of Theorem 24.3.1. **We assume that λ_k has multiplicity 1. One can prove it, but we will skip it.**

Let $\boldsymbol{\Psi}_k$ denote the diagonal matrix with $\boldsymbol{\psi}_k$ on the diagonal, and let λ_k be the corresponding eigenvalue. Consider the matrix

$$\boldsymbol{M} = \boldsymbol{\Psi}_k (\boldsymbol{L}_P - \lambda_k \boldsymbol{I}) \boldsymbol{\Psi}_k.$$

The matrix $\boldsymbol{L}_P - \lambda_k \boldsymbol{I}$ has one zero eigenvalue and $k-1$ negative eigenvalues. As we have assumed that $\boldsymbol{\psi}_k$ has no zero entries, $\boldsymbol{\Psi}_k$ is non-singular, and so we may apply Sylvester's Law of Intertia to show that the same is true of $\boldsymbol{M}$.

I claim that

$$\boldsymbol{M} = \sum_{(u,v)\in E} w_{u,v} \boldsymbol{\psi}_k(u) \boldsymbol{\psi}_k(v) \boldsymbol{L}_{u,v}.$$

To see this, first check that this agrees with the previous definition on the off-diagonal entries. To verify that these expression agree on the diagonal entries, we will show that the sum of the entries in each row of both expressions agree. As we know that all the off-diagonal entries agree, this implies that the diagonal entries agree. We compute

$$\boldsymbol{\Psi}_k(\boldsymbol{L}_P - \lambda_k \boldsymbol{I})\boldsymbol{\Psi}_k\mathbf{1} = \boldsymbol{\Psi}_k(\boldsymbol{L}_P - \lambda_k \boldsymbol{I})\boldsymbol{\psi}_k = \boldsymbol{\Psi}_k(\lambda_k\boldsymbol{\psi}_k - \lambda_k\boldsymbol{\psi}_k) = \mathbf{0}.$$

As $\boldsymbol{L}_{u,v}\mathbf{1} = \mathbf{0}$, the row sums agree. Lemma 24.3.2 now tells us that the matrix $\boldsymbol{M}$, and thus $\boldsymbol{L}_P - \lambda_k II$, has as many negative eigenvalues as there are edges (u, v) for which $\boldsymbol{\psi}_k(u)\boldsymbol{\psi}_k(v) < 0$. □

24.4 More linear algebra

There are a few more facts from linear algebra that we will need for the rest of this lecture. We stop to prove them now.

24.4.1 The Perron-Frobenius Theorem for Laplacians

In Lecture 3, we proved the Perron-Frobenius Theorem for non-negative matrices. I wish to quickly observe that this theory may also be applied to Laplacian matrices, to principal sub-matrices of Laplacian matrices, and to any matrix with non-positive off-diagonal entries. The difference is that it then involves the eigenvector of the smallest eigenvalue, rather than the largest eigenvalue.

Corollary 24.4.1. *Let M be a matrix with non-positive off-diagonal entries, such that the graph of the non-zero off-diagonally entries is connected. Let λ_1 be the smallest eigenvalue of M and let $\boldsymbol{v}_1$ be the corresponding eigenvector. Then $\boldsymbol{v}_1$ may be taken to be strictly positive, and λ_1 has multiplicity* 1.

Proof. Consider the matrix $A = \sigma I - M$, for some large σ. For σ sufficiently large, this matrix will be non-negative, and the graph of its non-zero entries is connected. So, we may apply the Perron-Frobenius theory to A to conclude that its largest eigenvalue α_1 has multiplicity 1, and the corresponding eigenvector $\boldsymbol{v}_1$ may be assumed to be strictly positive. We then have $\lambda_1 = \sigma - \alpha_1$, and $\boldsymbol{v}_1$ is an eigenvector of λ_1. □

24.4.2 Eigenvalue Interlacing

We will often use the following elementary consequence of the Courant-Fischer Theorem. I will assign it as homework.

Theorem 24.4.2 (Eigenvalue Interlacing). *Let $\boldsymbol{A}$ be an n-by-n symmetric matrix and let $\boldsymbol{B}$ be a principal submatrix of $\boldsymbol{A}$ of dimension $n-1$ (that is, $\boldsymbol{B}$ is obtained by deleting the same row and column from $\boldsymbol{A}$). Then,*

$$\alpha_1 \geq \beta_1 \geq \alpha_2 \geq \beta_2 \geq \cdots \geq \alpha_{n-1} \geq \beta_{n-1} \geq \alpha_n,$$

where $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_n$ and $\beta_1 \geq \beta_2 \geq \cdots \geq \beta_{n-1}$ are the eigenvalues of $\boldsymbol{A}$ and $\boldsymbol{B}$, respectively.

24.5 Fiedler's Nodal Domain Theorem

Given a graph $G = (V, E)$ and a subset of vertices, $W \subseteq V$, recall that the *graph induced by G on W* is the graph with vertex set W and edge set

$$\{(i,j) \in E, i \in W \text{ and } j \in W\}.$$

This graph is sometimes denoted $G(W)$.

Theorem 24.5.1 ([Fie75])**.** *Let $G = (V, E, w)$ be a weighted connected graph, and let L_G be its Laplacian matrix. Let $0 = \lambda_1 < \lambda_2 \leq \cdots \leq \lambda_n$ be the eigenvalues of L_G and let $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$ be the corresponding eigenvectors. For any $k \geq 2$, let*

$$W_k = \{i \in V : \boldsymbol{\psi}_k(i) \geq 0\}.$$

Then, the graph induced by G on W_k has at most $k-1$ connected components.

Proof. To see that W_k is non-empty, recall that $\boldsymbol{\psi}_1 = \mathbf{1}$ and that $\boldsymbol{\psi}_k$ is orthogonal $\boldsymbol{\psi}_1$. So, $\boldsymbol{\psi}_k$ must have both positive and negative entries.

Assume that $G(W_k)$ has t connected components. After re-ordering the vertices so that the vertices in one connected component of $G(W_k)$ appear first, and so on, we may assume that L_G and $\boldsymbol{\psi}_k$ have the forms

$$L_G = \begin{bmatrix} B_1 & \mathbf{0} & \mathbf{0} & \cdots & C_1 \\ \mathbf{0} & B_2 & \mathbf{0} & \cdots & C_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & B_t & C_t \\ C_1^T & C_2^T & \cdots & C_t^T & D \end{bmatrix} \qquad \boldsymbol{\psi}_k = \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_t \\ \boldsymbol{y} \end{pmatrix},$$

and

$$\begin{bmatrix} B_1 & \mathbf{0} & \mathbf{0} & \cdots & C_1 \\ \mathbf{0} & B_2 & \mathbf{0} & \cdots & C_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & B_t & C_t \\ C_1^T & C_2^T & \cdots & C_t^T & D \end{bmatrix} \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_t \\ \boldsymbol{y} \end{pmatrix} = \lambda_k \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_t \\ \boldsymbol{y} \end{pmatrix}.$$

The first t sets of rows and columns correspond to the t connected components. So, $\boldsymbol{x}_i \geq 0$ for $1 \leq i \leq t$ and $\boldsymbol{y} < 0$ (when I write this for a vector, I mean it holds for each entry). We also know that the graph of non-zero entries in each B_i is connected, and that each C_i is non-positive, and has at least one non-zero entry (otherwise the graph G would be disconnected).

We will now prove that the smallest eigenvalue of B_i is smaller than λ_k. We know that

$$B_i \boldsymbol{x}_i + C_i \boldsymbol{y} = \lambda_k \boldsymbol{x}_i.$$

As each entry in C_i is non-positive and $\boldsymbol{y}$ is strictly negative, each entry of $C_i \boldsymbol{y}$ is non-negative and some entry of $C_i \boldsymbol{y}$ is positive. Thus, $\boldsymbol{x}_i$ cannot be all zeros,

$$B_i \boldsymbol{x}_i = \lambda_k \boldsymbol{x}_i - C_i \boldsymbol{y} \leq \lambda_k \boldsymbol{x}_i$$

and

$$\boldsymbol{x}_i^T B_i \boldsymbol{x}_i \leq \lambda_k \boldsymbol{x}_i^T \boldsymbol{x}_i.$$

If $\boldsymbol{x}_i$ has *any* zero entries, then the Perron-Frobenius theorem tells us that $\boldsymbol{x}_i$ cannot be an eigenvector of smallest eigenvalue, and so the smallest eigenvalue of B_i is less than λ_k. On the other hand, if $\boldsymbol{x}_i$ is strictly positive, then $\boldsymbol{x}_i^T C_i \boldsymbol{y} > 0$, and

$$\boldsymbol{x}_i^T B_i \boldsymbol{x}_i = \lambda_k \boldsymbol{x}_i^T \boldsymbol{x}_i - \boldsymbol{x}_i^T C_i \boldsymbol{y} < \lambda_k \boldsymbol{x}_i^T \boldsymbol{x}_i.$$

Thus, the matrix

$$\begin{bmatrix} B_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & B_2 & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & B_t \end{bmatrix}$$

has at least t eigenvalues less than λ_k. By the eigenvalue interlacing theorem, this implies that L_G has at least t eigenvalues less than λ_k. We may conclude that t, the number of connected components of $G(W_k)$, is at most $k-1$. □

We remark that Fiedler actually proved a somewhat stronger theorem. He showed that the same holds for

$$W = \{i : \boldsymbol{\psi}_k(i) \geq t\},$$

for every $t \leq 0$.

This theorem breaks down if we instead consider the set

$$W = \{i : \boldsymbol{\psi}_k(i) > 0\}.$$

The star graphs provide counter-examples.

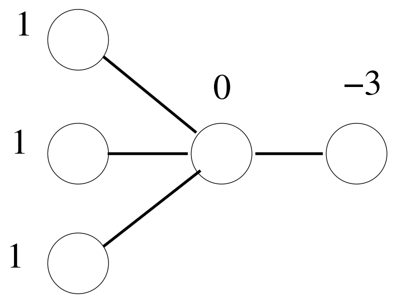


Figure 24.1: The star graph on 5 vertices, with an eigenvector of $\lambda_2 = 1$.

Chapter 25

The Second Eigenvalue of Planar Graphs

Lecture 16 from October 24, 2018

25.1 Overview

Spectral Graph theory first came to the attention of many because of the success of using the second Laplacian eigenvector to partition planar graphs and scientific meshes [DH72, DH73, Bar82, PSL90, Sim91].

In this lecture, we will attempt to explain this success by proving, at least for planar graphs, that the second smallest Laplacian eigenvalue is small. One can then use Cheeger's inequality to prove that the corresponding eigenvector provides a good cut.

This was already known for the model case of a 2-dimensional grid. If the grid is of size $\sqrt{n}$-by-$\sqrt{n}$, then it has $\lambda_2 \approx c/n$. Cheeger's inequality then tells us that it has a cut of conductance $c/\sqrt{n}$. And, this is in fact the cut that goes right accross the middle of one of the axes, which is the cut of minimum conductance.

Theorem 25.1.1 ([ST07])**.** *Let G be a planar graph with n vertices of maximum degree d, and let λ_2 be the second-smallest eigenvalue of its Laplacian. Then,*

$$\lambda_2 \leq \frac{8d}{n}.$$

The proof will involve almost no calculation, but will use some special properties of planar graphs. However, this proof has been generalized to many planar-like graphs, including the graphs of well-shaped 3d meshes.

25.2 Geometric Embeddings

We typically upper bound λ_2 by evidencing a test vector. Here, we will upper bound λ_2 by evidencing a test embedding. The bound we apply is:

Lemma 25.2.1. *For any $d \geq 1$,*

$$\lambda_2 = \min_{\boldsymbol{v}_1,\dots,\boldsymbol{v}_n \in \mathbb{R}^d : \sum \boldsymbol{v}_i = \boldsymbol{0}} \frac{\sum_{(i,j)\in E} \left\| \boldsymbol{v}_i - \boldsymbol{v}_j \right\|^2}{\sum_i \left\| \boldsymbol{v}_i \right\|^2 .} \tag{25.1}$$

Proof. Let $\boldsymbol{v}_i = (x_i, y_i, \dots, z_i)$. We note that

$$\sum_{(i,j)\in E} \left\| \boldsymbol{v}_i - \boldsymbol{v}_j \right\|^2 = \sum_{(i,j)\in E} (x_i - x_j)^2 + \sum_{(i,j)\in E} (y_i - y_j)^2 + \cdots + \sum_{(i,j)\in E} (z_i - z_j)^2.$$

Similarly,

$$\sum_i \left\| \boldsymbol{v}_i \right\|^2 = \sum_i x_i^2 + \sum_i y_i^2 + \cdots + \sum_i z_i^2.$$

It is now trivial to show that $\lambda_2 \geq RHS$: just let $x_i = y_i = \cdots = z_i$ be given by an eigenvector of λ_2. To show that $\lambda_2 \leq RHS$, we apply my favorite inequality: $\frac{A+B+\cdots+C}{A'+B'+\cdots+C'} \geq \min\left(\frac{A}{A'}, \frac{B}{B'}, \dots, \frac{C}{C'}\right)$, and then recall that $\sum x_i = 0$ implies

$$\frac{\sum_{(i,j)\in E} (x_i - x_j)^2}{\sum_i x_i^2} \geq \lambda_2.$$

□

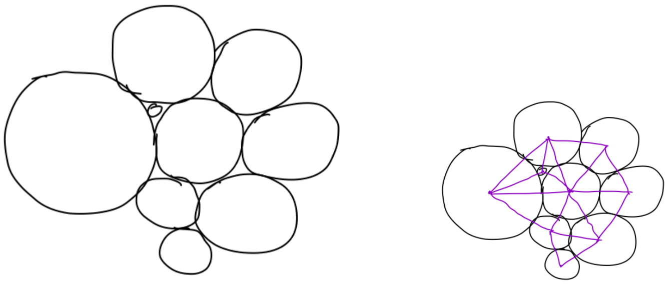
For an example, consider the natural embedding of the square with corners $(\pm 1, \pm 1)$.

The key to applying this embedding lemma is to obtain the right embedding of a planar graph. Usually, the right embedding of a planar graph is given by Koebe's embedding theorem, which I will now explain. I begin by considering one way of generating planar graphs. Consider a set of circles $\{C_1, \dots, C_n\}$ in the plane such that no pair of circles intersects in their interiors. Associate a vertex with each circle, and create an edge between each pair of circles that meet at a boundary. See Figure 25.2. The resulting graph is clearly planar. Koebe's embedding theorem says that *every planar graph results from such an embedding.*

Theorem 25.2.2 (Koebe). *Let $G = (V, E)$ be a planar graph. Then there exists a set of circles $\{C_1, \dots, C_n\}$ in $\mathbb{R}^2$ that are interior-disjoint such that circle C_i touches circle C_j if and only if $(i, j) \in E$.*

This is an amazing theorem, which I won't prove today. You can find a beautiful proof in the book "Combinatorial Geometry" by Agarwal and Pach.

Such an embedding is often called a *kissing disk* embedding of the graph. From a kissing disk embedding, we obtain a natural choice of $\boldsymbol{v}_i$: the center of disk C_i. Let r_i denote the radius of



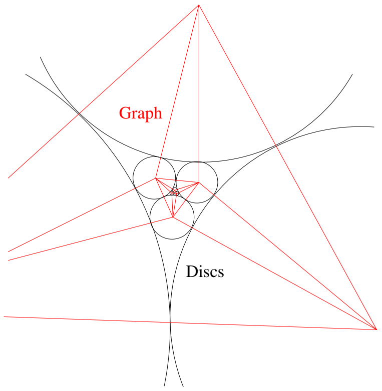
(a) Circles in the plane

(b) Circles with their intersection graph

this disk. We now have an easy upper bound on the numerator of (25.1): $\left\| \boldsymbol{v}_i - \boldsymbol{v}_j \right\|^2 = (r_i + r_j)^2 \leq 2r_i^2 + 2r_j^2$. On the other hand, it is trickier to obtain a lower bound on $\sum \left\| \boldsymbol{v}_i \right\|^2$. In fact, there are graphs whose kissing disk embeddings result in

$$(25.1) = \Theta(1).$$

These graphs come from triangles inside triangles inside triangles. . . Such a graph is depicted below:



We will fix this problem by lifting the planar embeddings to the sphere by stereographic projection. Given a plane, $\mathbb{R}^2$, and a sphere S tangent to the plane, we can define the stereographic projection map, Π, from the plane to the sphere as follows: let $\boldsymbol{s}$ denote the point where the sphere touches the plane, and let $\boldsymbol{n}$ denote the opposite point on the sphere. For any point $\boldsymbol{x}$ on the plane, consider the line from $\boldsymbol{x}$ to $\boldsymbol{n}$. It will intersect the sphere somewhere. We let this point of intersection be $\Pi(\boldsymbol{x})$.

The fundamental fact that we will exploit about stereographic projection is that *it maps circles to circles!* So, by applying stereographic projection to a kissing disk embedding of a graph in the plane, we obtain a kissing disk embedding of that graph on the sphere. Let $D_i = \Pi(C_i)$ denote the image of circle C_i on the sphere. We will now let $\boldsymbol{v}_i$ denote the center of D_i, on the sphere.

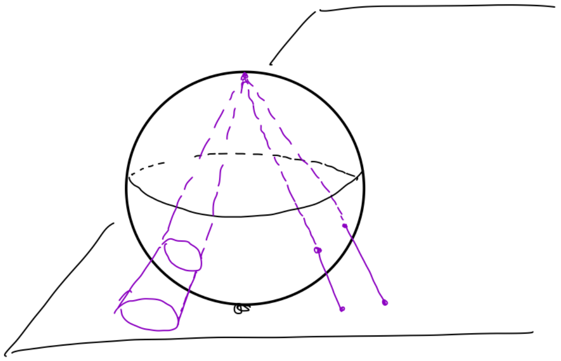


Figure 25.1: Stereographic Projection.

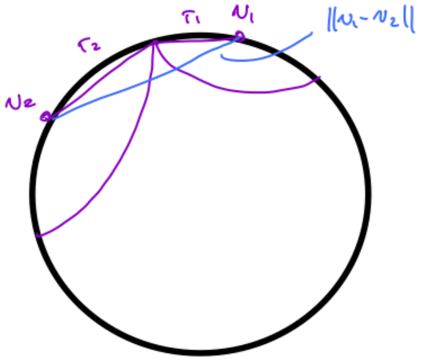


Figure 25.2: Stereographic Projection.

If we had $\sum_i \boldsymbol{v}_i = \boldsymbol{0}$, the rest of the computation would be easy. For each i, $\|\boldsymbol{v}_i\| = 1$, so the denominator of (25.1) is n. Let r_i denote the straight-line distance from $\boldsymbol{v}_i$ to the boundary of D_i. We then have (see Figure 25.2)

$$\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2 \leq (r_i + r_j)^2 \leq 2r_i^2 + 2r_j^2.$$

So, the denominator of (25.1) is at most $2d \sum_i r_i^2$. On the other hand, a theorem of Archimedes tells us that the area of the cap encircled by D_i is at exactly πr_i^2. Rather than proving it, I will convince you that it has to be true because it is true when r_i is small, it is true when the cap is a hemisphere and $r_i = \sqrt{2}$, and it is true when the cap is the whole sphere and $r_i = 2$.

As the caps are disjoint, we have

$$\sum_i \pi r_i^2 \leq 4\pi,$$

which implies that the denominator of (25.1) is at most

$$\sum_{(a,b) \in E} \|\boldsymbol{v}_a - \boldsymbol{v}_b\|^2 \leq 2r_a^2 + 2r_b^2 \leq 2d \sum_a r_a^2 \leq 8d.$$

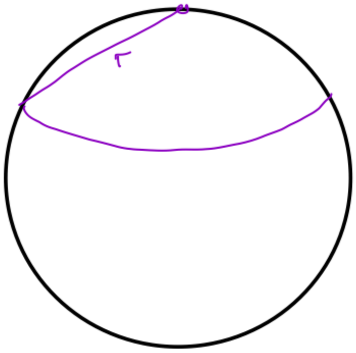


Figure 25.3: A Spherical Cap.

Putting these inequalities together, we see that

$$\min_{\boldsymbol{v}_1,\ldots,\boldsymbol{v}_n\in\mathbb{R}^d:\sum\boldsymbol{v}_i=\boldsymbol{0}} \frac{\sum_{(i,j)\in E}\left\|\boldsymbol{v}_i-\boldsymbol{v}_j\right\|^2}{\sum_i\left\|\boldsymbol{v}_i\right\|^2.} \leq \frac{8d}{n}.$$

Thus, we merely need to verify that we can ensure that

$$\sum_i \boldsymbol{v}_i = \boldsymbol{0}. \tag{25.2}$$

Note that there is enough freedom in our construction to believe that we could prove such a thing: we can put the sphere anywhere on the plane, and we could even scale the image in the plane before placing the sphere. By carefully combining these two operations, it is clear that we can place the center of gravity of the $\boldsymbol{v}_i$s close to any point on the boundary of the sphere. It turns out that this is sufficient to prove that we can place it at the origin.

25.3 The center of gravity

We need a nice family of maps that transform our kissing disk embedding on the sphere. It is particularly convenient to parameterize these by a point ω inside the sphere. For any point α on the surface of the unit sphere, I will let Π_α denote the stereographic projection from the plane tangent to the sphere at α.

I will also define Π_α^{-1}. To handle the point $-\alpha$, I let $\Pi_\alpha^{-1}(-\alpha)=\infty$, and $\Pi_\alpha(\infty)=-\alpha$. We also define the map that dilates the plane tangent to the sphere at α by a factor a: D_α^a. We then define the following map from the sphere to itself

$$f_\omega(\boldsymbol{x}) \stackrel{\text{def}}{=} \Pi_{\omega/\|\omega\|}\left(D_{\omega/\|\omega\|}^{1-\|\omega\|}\left(\Pi_{\omega/\|\omega\|}^{-1}(\boldsymbol{x})\right)\right).$$

For $\alpha\in S$ and $\omega=a\alpha$, this map pushes everything on the sphere to a point close to α. As a approaches 1, the mass gets pushed closer and closer to α.

Instead of proving that we can achieve (25.2), I will prove a slightly simpler theorem. The proof of the theorem we really want is similar, but about just a few minutes too long for class. We will prove

Theorem 25.3.1. *Let $\boldsymbol{v}_1, \dots, \boldsymbol{v}_n$ be points on the unit-sphere. Then, there exists an ω such that $\sum_i f_\omega(\boldsymbol{v}_i) = \mathbf{0}$.*

The reason that this theorem is different from the one that we want to prove is that if we apply a circle-preserving map from the sphere to itself, the center of the circle might not map to the center of the image circle.

To show that we can achieve $\sum_i \boldsymbol{v}_i = \mathbf{0}$, we will use the following topological lemma, which follows immediately from Brouwer's fixed point theorem. In the following, we let B denote the ball of points of norm less than 1, and S the sphere of points of norm 1.

Lemma 25.3.2. *If $\phi : B \to B$ be a continuous map that is the identity on S. Then, there exists an $\omega \in B$ such that*

$$\phi(\omega) = \mathbf{0}.$$

We will prove this lemma using Brouwer's fixed point theorem:

Theorem 25.3.3 (Brouwer)**.** *If $g : B \to B$ is continuous, then there exists an $\alpha \in B$ such that $g(\alpha) = \alpha$.*

Proof of Lemma 25.3.2. Let b be the map that sends $\boldsymbol{z} \in B$ to $\boldsymbol{z} / \|\boldsymbol{z}\|$. The map b is continuous at every point other than $\mathbf{0}$. Now, assume by way of contradiction that $\mathbf{0}$ is not in the image of ϕ, and let $g(\boldsymbol{z}) = -b(\phi(\boldsymbol{z}))$. By our assumption, g is continuous and maps B to B. However, it is clear that g has no fixed point, contradicting Brouwer's fixed point theorem. □

Lemma 25.3.2, was our motivation for defining the maps f_ω in terms of $\omega \in B$. Now consider setting

$$\phi(\omega) = \frac{1}{n} \sum_i f_\omega(\boldsymbol{v}_i).$$

The only thing that stops us from applying Lemma 25.3.2 at this point is that ϕ is not defined on S, because f_ω was not defined for $\omega \in S$. To fix this, we define for $\alpha \in S$

$$f_\alpha(\boldsymbol{z}) = \begin{cases} \alpha & \text{if } \boldsymbol{z} \neq -\alpha \\ -\alpha & \text{otherwise.} \end{cases}$$

We then encounter the problem that $f_\alpha(\boldsymbol{z})$ is not a continuous function of α because it is discontinuous at $\alpha = -\boldsymbol{v}_i$. But, this shouldn't be a problem because the point ω at which $\phi(\omega) = 0$ won't be on or near the boundary. The following argument makes this intuition formal.

We set

$$h_\omega(\boldsymbol{z}) = \begin{cases} 1 & \text{if } \operatorname{dist}(\omega, \boldsymbol{z}) < 2 - \epsilon, \text{ and} \\ (2 - \operatorname{dist}(\omega, \boldsymbol{z}))/\epsilon & \text{otherwise.} \end{cases}$$

Now, the function $f_\alpha(\boldsymbol{z})h_\alpha(\boldsymbol{z})$ is continuous on all of B. So, we may set

$$\phi(\omega) \overset{\text{def}}{=} \frac{\sum_i f_\omega(\boldsymbol{v}_i)h_\omega(\boldsymbol{v}_i)}{\sum_i h_\omega(\boldsymbol{v}_i),}$$

which is now continuous and is the identity map on S.

So, for any $\epsilon > 0$, we may now apply Lemma 25.3.2 to find an ω for which

$$\phi(\omega) = \mathbf{0}.$$

To finish the proof, we need to get rid of this ϵ. That is, we wish to show that ω is bounded away from S, say by μ, for all sufficiently small ϵ. If that is the case, then we will have $\text{dist}(\omega, \boldsymbol{v}_i) \geq \mu > 0$ for all sufficiently small ϵ. So, for $\epsilon < \mu$ and sufficiently small, $h_\omega(\boldsymbol{v}_i) = 1$ for all i, and we recover the $\epsilon = 0$ case.

One can verify that this holds provided that the points $\boldsymbol{v}_i$ are distinct and there are at least 3 of them.

Finally, recall that this is not exactly the theorem we wanted to prove: this theorem deals with $\boldsymbol{v}_i$, and not the centers of caps. The difficulty with centers of caps is that they move as the caps move. However, this can be overcome by observing that the centers remain inside the caps, and move continuously with ω. For a complete proof, see [ST07, Theorem 4.2]

25.4 Further progress

This result has been improved in many ways. Jonathan Kelner [Kel06] generalized this result to graphs of bounded genus. Kelner, Lee, Price and Teng [KLPT09] obtained analogous bounds for λ_k for $k \geq 2$. Biswal, Lee and Rao [BLR10] developed an entirely new set of techniques to prove these results. Their techniques improve these bounds, and extend them to graphs that do not have K_h minors for any constant h.

Chapter 26

Planar Graphs 2, the Colin de Verdière Number

Lecture 26 from December 4, 2009

26.1 Introduction

In this lecture, I will introduce the Colin de Verdière number of a graph, and sketch the proof that it is three for planar graphs. Along the way, I will recall two important facts about planar graphs:

1. Three-connected planar graphs are the skeletons of three-dimensional convex polytopes.
2. Planar graphs are the graphs that do not have K_5 or $K_{3,3}$ minors.

26.2 Colin de Verdière invariant

The Colin de Verdière graph parameter essentially measures the maximum multiplicity of the second eigenvalue of a generalized Laplacian matrix of the graph. It is less than or equal to three precisely for planar graphs.

We say that M is a *Generalized Laplacian Matrix* of a graph $G = (V, E)$ if M can be expressed as $M = L + D$ where L is a the Laplacian matrix of a weighted version of G and D is an arbitrary diagonal matrix. That is, we impose the restrictions:

$$\begin{aligned} &M(i,j) < 0 \quad \text{if } (i,j) \in E \\ &M(i,j) = 0 \quad \text{if } (i,j) \notin E \text{ and } i \neq j \\ &M(i,i) \text{ is arbitrary.} \end{aligned}$$

The Colin de Verdière graph parameter, which we denote $\mathsf{cdv}(G)$ is the maximum multiplicity of the second-smallest eigenvalue of a Generalized Laplacian Matrix M of G satisfying the following condition, known as the Strong Arnold Property.

> For every non-zero n-by-n matrix X such that $X(i,j) = 0$ for $i = j$ and $(i,j) \in E$, $MX \neq \mathbf{0}$.

That later restriction will be unnecessary for the results we will prove in this lecture.

Colin de Verdière [dV90] proved that $\mathsf{cdv}(G)$ is at most 2 if and only if the graph G is outerplanar. That is, it is a planar graph in which every vertex lies on one face. He also proved that it is at most 3 if and only if G is planar. Lovàsz and Schrijver [LS98] proved that it is at most 4 if and only if the graph is linkless embeddable.

In this lecture, I will sketch proofs from two parts of this work:

1. If G is a three-connected planar graph, then $\mathsf{cdv}(G) \geq 3$.
2. If G is a three-connected planar graph, then $\mathsf{cdv}(G) \leq 3$.

The first requires the construction of a matrix, which we do using the representation of the graph as a convex polytope. The second requires a proof that no Generalized Laplacian Matrix of the graph has a second eigenvalue of high multiplicity. We prove this by using graph minors.

26.3 Polytopes and Planar Graphs

Let me begin by giving two definitions of convex polytope: as the convex hull of a set of points and as the intersection of half-spaces.

Let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^d$ (think $d = 3$). Then, the *convex hull* of $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ is the set of points

$$\left\{ \sum_i a_i \boldsymbol{x}_i : \sum a_i = 1 \text{ and all } a_i \geq 0 \right\}.$$

Every convex polytope is the convex hull of its extreme vertices.

A convex polytope can also be defined by its faces. For example, given vectors $\boldsymbol{y}_1, \dots, \boldsymbol{y}_l$, the set of points

$$\left\{ \boldsymbol{x} : \boldsymbol{y}_i^T \boldsymbol{x} \leq 1, \text{ for all } i \right\}$$

is a convex polytope. Moreover, every convex polytope containing the origin in its interior can be described in this way. Each vector $\boldsymbol{y}_i$ defines a *face* of the polytope consisting of those points $\boldsymbol{x}$ in the polytope such that $\boldsymbol{y}_i^T \boldsymbol{x} = 1$.

The vertices of a convex polytope are those points $\boldsymbol{x}$ in the polytope that cannot be expressed non-trivially as a convex combination of any points other than themselves. The edges (or 1-faces) of a convex polytope are the line segments on the boundary of the polytope that go between two vertices of the polytope and such that every point on the edge cannot be expressed non-trivially as the convex hull of any vertices other than these two.

Theorem 26.3.1 (Steinitz's Theorem)**.** *For every three-connected planar graph $G = (V, E)$, there exists a set of vectors $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^3$ such that the line segment from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ is an edge of the convex hull of the vectors if and only if $(i, j) \in E$.*

That is, every planar graph may be represented by the edges of a three-dimensional convex polytope. We will use this representation to construct a Generalized Laplacian Matrix M whose second-smallest eigenvalue has multiplicity 3.

26.4 The Colin de Verdière Matrix

Let $G = (V, E)$ be a planar graph, and let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^3$ be the vectors given by Steinitz's Theorem. For $1 \leq i \leq 3$, let $\boldsymbol{v}_i \in \mathbb{R}^n$ be the vector given by

$$\boldsymbol{v}_i(j) = \boldsymbol{x}_j(i).$$

So, the vector $\boldsymbol{v}_i$ contains the ith coordinate of each vector $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$.

We will now see how to construct a generalized Laplacian matrix M having the vectors $\boldsymbol{v}_1, \boldsymbol{v}_2$ and $\boldsymbol{v}_3$ in its nullspace. One can also show that the matrix M has precisely one negative eigenvalue. But, we won't have time to do that in this lecture. You can find the details in [Lov01].

Our construction will exploit the *vector cross product.* Recall that for two vectors $\boldsymbol{x}$ and $\boldsymbol{y}$ in $\mathbb{R}^3$ that it is possible to define a vector $\boldsymbol{x} \times \boldsymbol{y}$ that is orthogonal to both $\boldsymbol{x}$ and $\boldsymbol{y}$, and whose length is the area of the parallelogram with sizes $\boldsymbol{x}$ and $\boldsymbol{y}$. This determines the cross product up to sign. You should recall that the sign is determined by an ordering of the basis of $\mathbb{R}^3$, or by the *right hand rule.* Also recall that

$$\begin{aligned} \boldsymbol{x} \times \boldsymbol{y} &= -\boldsymbol{y} \times \boldsymbol{x}, \\ (\boldsymbol{x}_1 + \boldsymbol{x}_2) \times \boldsymbol{y} &= \boldsymbol{x}_1 \times \boldsymbol{y} + \boldsymbol{x}_2 \times \boldsymbol{y}, \qquad \text{and} \\ \boldsymbol{x} \times \boldsymbol{y} &= 0 \text{ if and only if } \boldsymbol{x} \text{ and } \boldsymbol{y} \text{ are parallel.} \end{aligned}$$

We will now specify the entries $M(i, j)$ for $(i, j) \in E$. An edge (i, j) is on the boundary of two faces of the polytope. Let's say that the vectors defining these faces are $\boldsymbol{y}_a$ and $\boldsymbol{y}_b$. So,

$$\boldsymbol{y}_a^T \boldsymbol{x}_i = \boldsymbol{y}_a^T \boldsymbol{x}_j = \boldsymbol{y}_b^T \boldsymbol{x}_i = \boldsymbol{y}_b^T \boldsymbol{x}_j = 1.$$

So,

$$(\boldsymbol{y}_a - \boldsymbol{y}_b)^T \boldsymbol{x}_i = (\boldsymbol{y}_a - \boldsymbol{y}_b)^T \boldsymbol{x}_j = 0.$$

This implies that $\boldsymbol{y}_a - \boldsymbol{y}_b$ is parallel to $\boldsymbol{x}_i \times \boldsymbol{x}_j$.

Assume $\boldsymbol{y}_a$ comes before $\boldsymbol{y}_b$ in the clockwise order about vertex $\boldsymbol{x}_i$. So, $\boldsymbol{y}_b - \boldsymbol{y}_a$ points the same direction as $\boldsymbol{x}_i \times \boldsymbol{x}_j$. Set $M(i, j)$ so that

$$M(i, j) \boldsymbol{x}_i \times \boldsymbol{x}_j = \boldsymbol{y}_a - \boldsymbol{y}_b$$

and $M(i,j) < 0$.

I will now show that we can choose the diagonal entries $M(i,i)$ so that the coordinate vectors are in the nullspace of M. First, set

$$\hat{\boldsymbol{x}}_i = \sum_{j \sim i} M(i,j)\boldsymbol{x}_j.$$

We will show that $\hat{\boldsymbol{x}}_i$ is parallel to $\boldsymbol{x}_i$ by observing that $\hat{\boldsymbol{x}}_i \times \boldsymbol{x}_i = 0$. We compute

$$\boldsymbol{x}_i \times \hat{\boldsymbol{x}}_i = \boldsymbol{x}_i \times \sum_{j \sim i} M(i,j)\boldsymbol{x}_j = \sum_{j \sim i} M(i,j)\boldsymbol{x}_i \times \boldsymbol{x}_j.$$

This sum counts the difference $\boldsymbol{y}_b - \boldsymbol{y}_a$ between each adjacent pair of faces that touch $\boldsymbol{x}_i$. By going around $\boldsymbol{x}_i$ in counter-clockwise order, we see that each of these vectors occurs once positively and once negatively in the sum, so the sum is zero.

Thus, $\boldsymbol{x}_i$ and $\hat{\boldsymbol{x}}_i$ are parallel, and we may set $M(i,i)$ so that

$$M(i,i)\boldsymbol{x}_i + \hat{\boldsymbol{x}}_i = \boldsymbol{0}.$$

This implies that the coordinate vectors are in the nullspace of M, as

$$\left(M \begin{pmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_n \end{pmatrix} \right)_i = M(i,i)\boldsymbol{x}_i + \sum_{j \sim i} M(i,j)\boldsymbol{x}_j = M(i,i)\boldsymbol{x}_i + \hat{\boldsymbol{x}}_i.$$

One can also show that the matrix M has precisely one negative eigenvalue, so the multiplicity of its second-smallest eigenvalue is 3.

26.5 Minors of Planar Graphs

I will now show you that $\mathsf{cdv}(G) \leq 3$ for every 3-connected planar graph G. To begin, I mention one other characterization of planar graphs.

First, observe that if G is a planar graph, it remains planar when we remove an edge. Also observe that if (u,v) is an edge, then the graph obtained by contracting (u,v) to one vertex is also planar. Any graph H that can be obtained by removing and contracting edges from a graph G is called a *minor* of G. It is easy to show that every minor of a planar graph is also planar. Kuratowski's Theorem tells us that a graph is planar *if and only if* it does not have K_5 or $K_{3,3}$ (the complete bipartite graph between two sets of 3 vertices) as a minor. We will just use the fact that a planar graph does not have $K_{3,3}$ as a minor.

26.6 $\mathsf{cdv}(G) \leq 3$

We will now prove that if G is a 3-connected planar graph, then $\mathsf{cdv}(G) \leq 3$. Assume, by way of contradiction, that there is generalized Laplacian matrix M of G whose second eigenvalue λ_2 has multiplicity greater than or equal to 4. We will do this by showing that if G is three-connected and $\mathsf{cdv}(G) \geq 4$, then G contains a $K_{3,3}$ minor. Without loss of generality, we can assume $\lambda_2 = 0$ (by just adding a diagonal matrix).

Our proof will exploit a variant of Fiedler's Nodal Domain Theorem, which we proved back in the beginning of the semester. That theorem considered *any* eigenvector $\boldsymbol{v}$ of λ_2 (of a Laplacian), and proved that the set of vertices that are non-negative in $\boldsymbol{v}$ is connected. The variant we use is due to van der Holst [van95], which instead applies to eigenvectors $\boldsymbol{v}$ of λ_2 of *minimal support.* These are the eigenvectors of $\boldsymbol{v}$ of λ_2 for which there is no other eigenvector $\boldsymbol{w}$ of λ_2 such that the zeros of $\boldsymbol{v}$ are a subset of the zeros of $\boldsymbol{w}$. That is, $\boldsymbol{v}$ has as many zero entries as possible. One can then prove that the set of vertices that are positive in $\boldsymbol{v}$ is connected. And, one can of course do the same for the vertices that are negative.

Now, let F be any face of G, and let a, b and c be three vertices in F. As λ_2 has multiplicity at least 4, it has some eigenvector that is zero at each of a, b and c. Let $\boldsymbol{v}$ be an eigenvector of λ_2 with minimal support that is zero at each of a, b, and c. Let d be any vertex for which $\boldsymbol{v}(d) > 0$. As the graph is three-connected, it contains three vertex-disjoint paths from d to a, b, and c (this follows from Menger's Theorem, which I have not covered). As $\boldsymbol{v}(d) > 0$ and $\boldsymbol{v}(a) = 0$, there is some vertex a' on the path from d to a for which $\boldsymbol{v}(a') = 0$ but a' has a neighbor a^+ for which $\boldsymbol{v}(a^+) > 0$. As $\lambda_2 = 0$, a' must also have a neighbor a^- for which $\boldsymbol{v}(a^-) < 0$. Construct similar vertices for b and c.

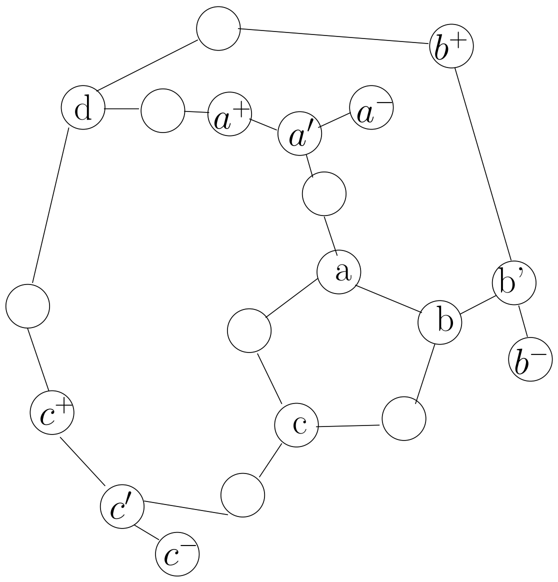


Figure 26.1: Vertices a, b, c, d, and the paths.

Now, contract every edge on the path from a to a', on the path from b to b' and on the path from c to c'. Also, contract all the vertices for which $\boldsymbol{v}$ is positive and contract all the vertices for which $\boldsymbol{v}$ is negative (which we can do because these sets are connected). Finally, contract every edge in the face F that does not involve one of a, b, or c. We obtain a graph with a triangular face abc such that each of a, b, and c have an edge to the positive supervertex and the negative supervertex. We would like to say that this graph cannot be planar.

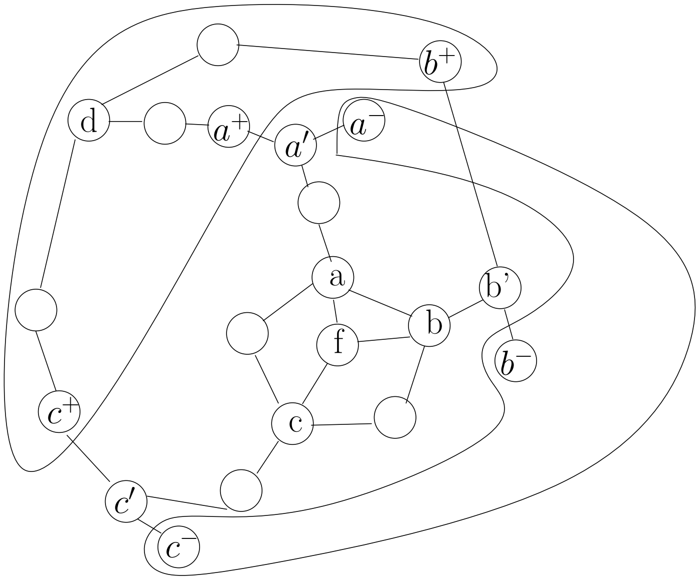


Figure 26.2: The set of positive and negative vertices that will be contracted. Vertex f has been inserted.

To do this, we add one additional vertex f inside the face and connected to each of a, b, and c. This does not violate planarity because a, b, and c were contained in a face. In fact, we can add f before we do the contractions. By throwing away all other edges, we have constructed a $K_{3,3}$ minor, so the graph cannot be planar.

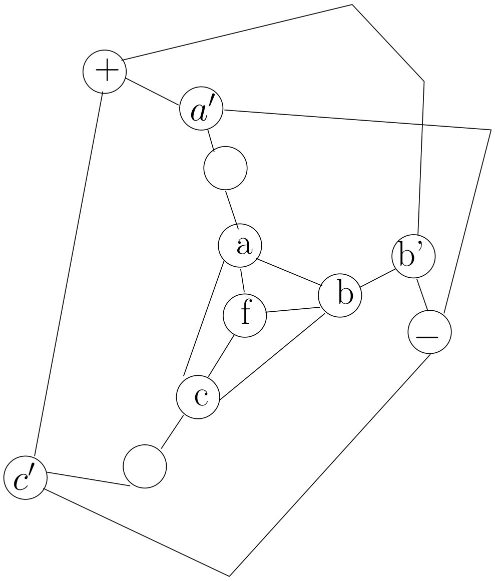


Figure 26.3: The edges in the cycle have been contracted, as have all the positive and negative vertices. After contracting the paths between a and a', between b and b' and between c and c', we obtain a $K_{3,3}$ minor.

Part V

Expander Graphs

Chapter 27

Properties of Expander Graphs

Lecture 17 from October 29, 2018

27.1 Overview

We say that a d-regular graph is a good expander if all of its adjacency matrix eigenvalues are small. To quantify this, we set a threshold $\epsilon > 0$, and require that each adjacency matrix eigenvalue, other than d, has absolute value at most ϵd. This is equivalent to requiring all non-zero eigenvalues of the Laplacian to be within ϵd of d.

In this lecture, we will:

1. Show that this condition is equivalent to approximating the complete graph.
2. Prove that this condition implies that the number of edges between sets of vertices in the graph is approximately the same as in a d-regular random graph.
3. Prove Tanner's Theorem: that small sets of vertices have many neighbors.
4. Derive the Alon-Boppana bound, which says that ϵ cannot be asymptotically smaller than $2\sqrt{d-1}/d$. This will tell us that the asymptotically best expanders are the Ramanujan graphs.

Random d-regular graphs are expander graphs. Explicitly constructed expander graphs have proved useful in a large number of algorithms and theorems. We will see some applications of them next week.

27.2 Expanders as Approximations of the Complete Graph

One way of measuring how well two matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ approximate each other is to measure the operator norm of their difference: $\boldsymbol{A} - \boldsymbol{B}$. Since I consider the operator norm by default, I will just refer to it as the norm. Recall that the norm of a matrix $\boldsymbol{M}$ is defined to be its largest singular value:

$$\|\boldsymbol{M}\| = \max_{\boldsymbol{x}} \frac{\|\boldsymbol{M}\boldsymbol{x}\|}{\|\boldsymbol{x}\|},$$

where the norms in the fraction are the standard Euclidean vector norms. The norm of a symmetric matrix is just the largest absolute value of one of its eigenvalues. It can be very different for a non symmetric matrix.

For this lecture, we define an ϵ-expander to be a d-regular graph whose adjacency matrix eigenvalues satisfy $|\mu_i| \leq \epsilon d$ for $\mu_i \geq 2$. As the Laplacian matrix eigenvalues are given by $\lambda_i = d - \mu_i$, this is equivalent to $|d - \lambda_i| \leq \epsilon d$ for $i \geq 2$. It is also equivalent to

$$\|\boldsymbol{L}_G - (d/n)\boldsymbol{L}_{K_n}\| \leq \epsilon d.$$

For this lecture, I define a graph G to be an ϵ-approximation of a graph H if

$$(1-\epsilon)H \preccurlyeq G \preccurlyeq (1+\epsilon)H,$$

where I recall that I say $H \preccurlyeq G$ if for all $\boldsymbol{x}$

$$\boldsymbol{x}^T \boldsymbol{L}_H \boldsymbol{x} \leq \boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x}.$$

I warn you that this definition is not symmetric. When I require a symmetric definition, I usually use the condition $(1+\epsilon)^{-1}H \preccurlyeq G$ instead of $(1-\epsilon)H \preccurlyeq G$.

If G is an ϵ-expander, then for all $\boldsymbol{x} \in \mathbb{R}^V$ that are orthogonal to the constant vectors,

$$(1-\epsilon)d\boldsymbol{x}^T\boldsymbol{x} \leq \boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x} \leq (1+\epsilon)d\boldsymbol{x}^T\boldsymbol{x}.$$

On the other hand, for the complete graph K_n, we know that all $\boldsymbol{x}$ orthogonal to the constant vectors satisfy

$$\boldsymbol{x}^T \boldsymbol{L}_{K_n} \boldsymbol{x} = n\boldsymbol{x}^T\boldsymbol{x}.$$

Let H be the graph

$$H = \frac{d}{n}K_n,$$

so

$$\boldsymbol{x}^T \boldsymbol{L}_H \boldsymbol{x} = d\boldsymbol{x}^T\boldsymbol{x}.$$

So, G is an ϵ-approximation of H.

This tells us that $\boldsymbol{L}_G - \boldsymbol{L}_H$ is a matrix of small norm. Observe that

$$(1-\epsilon)\boldsymbol{L}_H \preccurlyeq \boldsymbol{L}_G \preccurlyeq (1+\epsilon)\boldsymbol{L}_H \quad \text{implies} \quad -\epsilon\boldsymbol{L}_H \preccurlyeq \boldsymbol{L}_G - \boldsymbol{L}_H \preccurlyeq \epsilon\boldsymbol{L}_H.$$

As $\boldsymbol{L}_G$ and $\boldsymbol{L}_H$ are symmetric, and all eigenvalues of $\boldsymbol{L}_H$ are 0 or d, we may infer

$$\|\boldsymbol{L}_G - \boldsymbol{L}_H\| \leq \epsilon d. \tag{27.1}$$

27.3 Quasi-Random Properties of Expanders

There are many ways in which expander graphs act like random graphs. Conversely, one can prove that a random d-regular graph is an expander graph with reasonably high probability [Fri08].

We will see that all sets of vertices in an expander graph act like random sets of vertices. To make this precise, imagine creating a random set $S \subset V$ by including each vertex in S independently with probability α. How many edges do we expect to find between vertices in S? Well, for every edge (u, v), the probability that $u \in S$ is α and the probability that $v \in S$ is α, so the probability that both endpoints are in S is α^2. So, we expect an α^2 fraction of the edges to go between vertices in S. We will show that this is true *for all sufficiently large sets* S in an expander.

In fact, we will prove a stronger version of this statement for two sets S and T. Imagine including each vertex in S independently with probability α and each vertex in T with probability β. We allow vertices to belong to both S and T. For how many ordered pairs $(u, v) \in E$ do we expect to have $u \in S$ and $v \in T$? Obviously, it should hold for an $\alpha\beta$ fraction of the pairs.

For a graph $G = (V, E)$, define

$$\vec{E}(S,T) = \{(u,v) : u \in S, v \in T, (u,v) \in E\}.$$

We have put the arrow above the E in the definition, because we are considering ordered pairs of vertices. When S and T are disjoint

$$\left|\vec{E}(S,T)\right|$$

is precisely the number of edges between S and T, while

$$\left|\vec{E}(S,S)\right|$$

counts every edge inside S twice.

The following bound is a slight extension by Beigel, Margulis and Spielman [BMS93] of a bound originally proved by Alon and Chung [AC88].

Theorem 27.3.1. *Let $G = (V, E)$ be a d-regular graph that ϵ-approximates $\frac{d}{n}K_n$. Then, for every $S \subseteq V$ and $T \subseteq V$,*

$$\left|\left|\vec{E}(S,T)\right| - \alpha\beta dn\right| \leq \epsilon dn\sqrt{(\alpha-\alpha^2)(\beta-\beta^2)},$$

where $|S| = \alpha n$ and $|T| = \beta n$.

Observe that when α and β are greater than ϵ, the term on the right is less than $\alpha\beta dn$.

In class, we will just prove this in the case that S and T are disjoint.

Proof. The first step towards the proof is to observe

$$\boldsymbol{\chi}_S^T \boldsymbol{L}_G \boldsymbol{\chi}_T = d\,|S \cap T| - \left|\vec{E}(S,T)\right|.$$

Let $H = \frac{d}{n} K_n$. As G is a good approximation of H, let's compute

$$\boldsymbol{\chi}_S^T \boldsymbol{L}_H \boldsymbol{\chi}_T = \boldsymbol{\chi}_S^T \left(dI - \frac{d}{n} J \right) \boldsymbol{\chi}_T = d \left| S \cap T \right| - \frac{d}{n} \left| S \right| \left| T \right| = d \left| S \cap T \right| - \alpha \beta d n.$$

So,

$$\left| \left| \vec{E}(S,T) \right| - \alpha \beta d n \right| = \left| \boldsymbol{\chi}_S^T \boldsymbol{L}_G \boldsymbol{\chi}_T - \boldsymbol{\chi}_S^T \boldsymbol{L}_H \boldsymbol{\chi}_T \right|.$$

As

$$\left\| \boldsymbol{L}_G - \boldsymbol{L}_H \right\| \leq \epsilon d,$$

$$\begin{aligned}
\boldsymbol{\chi}_S^T \boldsymbol{L}_H \boldsymbol{\chi}_T - \boldsymbol{\chi}_S^T \boldsymbol{L}_G \boldsymbol{\chi}_T &= \boldsymbol{\chi}_S^T (\boldsymbol{L}_H - \boldsymbol{L}_G) \boldsymbol{\chi}_T \\
&\leq \left\| \boldsymbol{\chi}_S \right\| \left\| (\boldsymbol{L}_H - \boldsymbol{L}_G) \boldsymbol{\chi}_T \right\| \\
&\leq \left\| \boldsymbol{\chi}_S \right\| \left\| \boldsymbol{L}_H - \boldsymbol{L}_G \right\| \left\| \boldsymbol{\chi}_T \right\| \\
&\leq \epsilon d \left\| \boldsymbol{\chi}_S \right\| \left\| \boldsymbol{\chi}_T \right\| \\
&= \epsilon d n \sqrt{\alpha \beta}.
\end{aligned}$$

This is almost as good as the bound we are trying to prove. To prove the claimed bound, recall that $\boldsymbol{L}_H \boldsymbol{x} = \boldsymbol{L}_H (\boldsymbol{x} + c\boldsymbol{1})$ for all c. So, let $\boldsymbol{x}_S$ and $\boldsymbol{x}_T$ be the result of orthogonalizing $\boldsymbol{\chi}_S$ and $\boldsymbol{\chi}_T$ with respect to the constant vectors. By Claim 2.4.2 (from Lecture 2), $\left\| \boldsymbol{x}_S \right\| = n(\alpha - \alpha^2)$. So, we obtain the improved bound

$$\boldsymbol{x}_S^T (\boldsymbol{L}_H - \boldsymbol{L}_G) \boldsymbol{x}_T = \boldsymbol{\chi}_S^T (\boldsymbol{L}_H - \boldsymbol{L}_G) \boldsymbol{\chi}_T,$$

while

$$\left\| \boldsymbol{x}_S \right\| \left\| \boldsymbol{x}_T \right\| = n \sqrt{(\alpha - \alpha^2)(\beta - \beta^2)}.$$

So, we may conclude

$$\left| \left| \vec{E}(S,T) \right| - \alpha \beta d n \right| \leq \epsilon d n \sqrt{(\alpha - \alpha^2)(\beta - \beta^2)}.$$

□

We remark that when S and T are disjoint, the same proof goes through even if G is irregular and weighted if we replace $\vec{E}(S,T)$ with

$$w(S,T) = \sum_{(u,v) \in E, u \in S, v \in T} w(u,v).$$

We only need the fact that G ϵ-approximates $\frac{d}{n} K_n$. See [BSS12] for details.

27.4 Vertex Expansion

The reason for the name *expander graph* is that small sets of vertices in expander graphs have unusually large numbers of neighbors. For $S \subset V$, let $N(S)$ denote the set of vertices that are neighbors of vertices in S. The following theorem, called Tanner's Theorem, provides a lower bound on the size of $N(S)$.

Theorem 27.4.1 ([Tan84]). *Let $G = (V, E)$ be a d-regular graph on n vertices that ϵ-approximates $\frac{d}{n}K_n$. Then, for all $S \subseteq V$,*

$$|N(S)| \geq \frac{|S|}{\epsilon^2(1-\alpha)+\alpha},$$

where $|S| = \alpha n$.

Note that when α is much less than ϵ^2, the term on the right is approximately $|S|/\epsilon^2$, which can be much larger than $|S|$. We will derive Tanner's theorem from Theorem 27.3.1.

Proof. Let $R = N(S)$ and let $T = V - R$. Then, there are no edges between S and T. Let $|T| = \beta n$ and $|R| = \gamma n$, so $\gamma = 1 - \beta$. By Theorem 27.3.1, it must be the case that

$$\alpha\beta dn \leq \epsilon dn\sqrt{(\alpha-\alpha^2)(\beta-\beta^2)}.$$

The lower bound on γ now follows by re-arranging terms. Dividing through by dn and squaring both sides gives

$$\begin{aligned}
\alpha^2\beta^2 &\leq \epsilon^2(\alpha-\alpha^2)(\beta-\beta^2) && \iff \\
\alpha\beta &\leq \epsilon^2(1-\alpha)(1-\beta) && \iff \\
\frac{\beta}{1-\beta} &\leq \frac{\epsilon^2(1-\alpha)}{\alpha} && \iff \\
\frac{1-\gamma}{\gamma} &\leq \frac{\epsilon^2(1-\alpha)}{\alpha} && \iff \\
\frac{1}{\gamma} &\leq \frac{\epsilon^2(1-\alpha)+\alpha}{\alpha} && \iff \\
\gamma &\geq \frac{\alpha}{\epsilon^2(1-\alpha)+\alpha}.
\end{aligned}$$

□

If instead of $N(S)$ we consider $N(S) - S$, then T and S are disjoint, so the same proof goes through for weighted, irregular graphs that ϵ-approximate $\frac{d}{n}K_n$.

27.5 How well can a graph approximate the complete graph?

Consider applying Tanner's Theorem with $S = \{v\}$ for some vertex v. As v has exactly d neighbors, we find

$$\epsilon^2(1-1/n) + 1/n \geq 1/d,$$

from which we see that ϵ must be at least $1/\sqrt{d + d^2/n}$, which is essentially $1/\sqrt{d}$. But, how small can it be?

The Ramanujan graphs, constructed by Margulis [Mar88] and Lubotzky, Phillips and Sarnak [LPS88] achieve

$$\epsilon \leq \frac{2\sqrt{d-1}}{d}.$$

We will see that if we keep d fixed while we let n grow, ϵ cannot exceed this bound in the limit. We will prove an upper bound on ϵ by constructing a suitable test function.

As a first step, choose two vertices v and u in V whose neighborhoods to do not overlap. Consider the vector $\boldsymbol{x}$ defined by

$$\boldsymbol{x}(i) = \begin{cases} 1 & \text{if } i = u, \\ 1/\sqrt{d} & \text{if } i \in N(u), \\ -1 & \text{if } i = v, \\ -1/\sqrt{d} & \text{if } i \in N(v), \\ 0 & \text{otherwise.} \end{cases}$$

Now, compute the Rayleigh quotient with respect to $\boldsymbol{x}$. The numerator is the sum over all edges of the squares of differences across the edges. This gives $(1 - 1/\sqrt{d})^2$ for the edges attached to u and v, and $1/d$ for the edges attached to $N(u)$ and $N(v)$ but not to u or v, for a total of

$$2d(1 - 1/\sqrt{d})^2 + 2d(d-1)/d = 2\left(d - 2\sqrt{d} + 1 + (d-1)\right) = 2\left(2d - 2\sqrt{d}\right).$$

On the other hand, the denominator is 4, so we find

$$\frac{\boldsymbol{x}^T L \boldsymbol{x}}{\boldsymbol{x}^T \boldsymbol{x}} = d - \sqrt{d}.$$

If we use instead the vector

$$\boldsymbol{y}(i) = \begin{cases} 1 & \text{if } i = u, \\ -1/\sqrt{d} & \text{if } i \in N(u), \\ -1 & \text{if } i = v, \\ 1/\sqrt{d} & \text{if } i \in N(v), \\ 0 & \text{otherwise,} \end{cases}$$

we find

$$\frac{\boldsymbol{y}^T L \boldsymbol{y}}{\boldsymbol{y}^T \boldsymbol{y}} = d + \sqrt{d}.$$

This is not so impressive, as it merely tells us that $\epsilon \geq 1/\sqrt{d}$, which we already knew. But, we can improve this argument by pushing it further. We do this by modifying it in two ways. First, we extend $\boldsymbol{x}$ to neighborhoods of neighborhoods of u and v. Second, instead of basing the construction at vertices u and v, we base it at two edges. This way, each vertex has $d-1$ edges to those that are farther away from the centers of the construction.

The following theorem is attributed to A. Nilli [Nil91], but we suspect it was written by N. Alon.

Theorem 27.5.1. *Let G be a d-regular graph containing two edges (u_0, u_1) and (v_0, v_1) that are at distance at least $2k+2$. Then*

$$\lambda_2 \leq d - 2\sqrt{d-1} + \frac{2\sqrt{d-1} - 1}{k+1}.$$

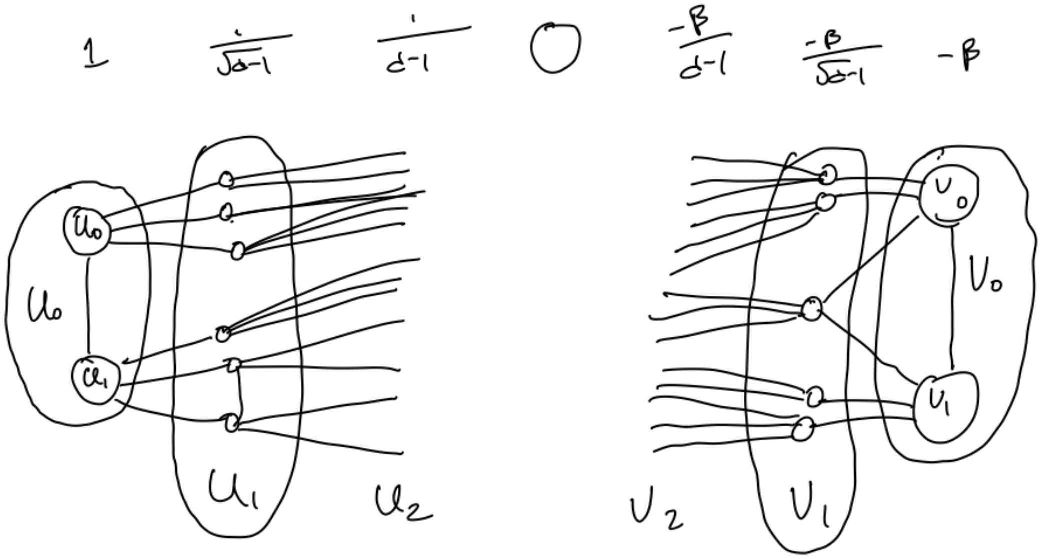


Figure 27.1: The construction of $\boldsymbol{x}$.

Proof. Define the following neighborhoods.

$$
\begin{aligned}
U_0 &= \{u_0, u_1\} \\
U_i &= N(U_{i-1}) - \cup_{j<i} U_j, \quad \text{for } 0 < i \leq k, \\
V_0 &= \{v_0, v_1\} \\
V_i &= N(V_{i-1}) - \cup_{j<i} V_j, \quad \text{for } 0 < i \leq k.
\end{aligned}
$$

That is, U_i consists of exactly those vertices at distance i from U_0. Note that there are no edges between any vertices in any U_i and any V_j.

Our test vector for λ_2 will be given by

$$
\boldsymbol{x}(a) = \begin{cases} \frac{1}{(d-1)^{-i/2}} & \text{for } a \in U_i \\ -\frac{\beta}{(d-1)^{-i/2}} & \text{for } a \in V_i \\ \\ 0 & \text{otherwise.} \end{cases}
$$

We choose β so that $\boldsymbol{x}$ is orthogonal to $\mathbf{1}$.

We now find that the Rayleigh quotient of $\boldsymbol{x}$ with respect to $\boldsymbol{L}$ is at most

$$
\frac{X_0 + \beta^2 Y_0}{X_1 + \beta^2 Y_1},
$$

where

$$
X_0 = \sum_{i=0}^{k-1} |U_i| \, (d-1) \left(\frac{1 - 1/\sqrt{d-1}}{(d-1)^{-i/2}} \right)^2 + |U_k| \, (d-1)^{-k+1}, \text{ and } X_1 = \sum_{i=0}^{k} |U_i| \, (d-1)^{-i}
$$

and

$$Y_0 = \sum_{i=0}^{k-1} |V_i| (d-1) \left(\frac{1 - 1/\sqrt{d-1}}{(d-1)^{-i/2}} \right)^2 + |V_k| (d-1)^{-k+1}, \text{ and } Y_1 = \sum_{i=0}^{k} |V_i| (d-1)^{-i}.$$

By my favorite inequality, it suffices to prove upper bounds on X_0/X_1 and Y_0/Y_1. So, consider

$$\frac{\sum_{i=0}^{k-1} |U_i| (d-1) \left(\frac{1-1/\sqrt{d-1}}{(d-1)^{-i/2}} \right)^2 + |U_k| (d-1)^{-k+1}}{\sum_{i=0}^{k} |U_i| (d-1)^{-i}}.$$

For now, let's focus on the numerator,

$$\begin{aligned}
&\sum_{i=0}^{k-1} |U_i| (d-1) \left(\frac{1 - 1/\sqrt{d-1}}{(d-1)^{-i/2}} \right)^2 + |U_k| (d-1)(d-1)^{-k} \\
&= \sum_{i=0}^{k-1} \frac{|U_i|}{(d-1)^i} (d - 2\sqrt{d-1}) + \frac{|U_k|}{(d-1)^k} (d-1) \\
&= \sum_{i=0}^{k-1} \frac{|U_i|}{(d-1)^i} (d - 2\sqrt{d-1}) + \frac{|U_k|}{(d-1)^k} (d - 2\sqrt{d-1}) + \frac{|U_k|}{(d-1)^k} (2\sqrt{d-1} - 1) \\
&= \sum_{i=0}^{k} \frac{|U_i|}{(d-1)^i} (d - 2\sqrt{d-1}) + \frac{|U_k|}{(d-1)^k} (2\sqrt{d-1} - 1).
\end{aligned}$$

To upper bound the Rayleigh quotient, we observe that the left-most of these terms contributes

$$\frac{\sum_{i=0}^{k} \frac{|U_i|}{(d-1)^i} (d - 2\sqrt{d-1})}{\sum_{i=0}^{k} |U_i| (d-1)^{-i}} = d - 2\sqrt{d-1}.$$

To bound the impact of the remaining term,

$$\frac{|U_k|}{(d-1)^k} (2\sqrt{d-1} - 1),$$

note that

$$|U_k| \leq (d-1)^{k-i} |U_i|.$$

So, we have

$$\frac{|U_k|}{(d-1)^k} \leq \frac{1}{k+1} \sum_{i=0}^{k} \frac{|U_i|}{(d-1)^i}.$$

Thus, the last term contributes at most

$$\frac{2\sqrt{d} - 1}{k+1}$$

to the Rayleigh quotient.

□

27.6 Open Problems

What can we say about λ_n? In a previous iteration of this course, I falsely asserted that the same proof tells us that

$$\lambda_n \geq d + 2\sqrt{d-1} - \frac{2\sqrt{d-1}-1}{k+1}.$$

But, the proof did not work.

Another question is how well a graph of average degree d can approximate the complete graph. That is, let G be a graph with $dn/2$ edges, but let G be irregular. While I doubt that irregularity helps one approximate the complete graph, I do not know how to prove it.

We can generalize this question further. Let $G = (V, E, w)$ be a weighted graph with $dn/2$ edges. Can we prove that G cannot approximate a complete graph any better than the Ramanujan graphs do? I conjecture that for every d and every $\beta > 0$ there is an n_0 so that for every graph of average degree d on $n \geq n_0$ vertices,

$$\frac{\lambda_2}{\lambda_n} \leq \frac{d - 2\sqrt{d-1}}{d + 2\sqrt{d-1}} + \beta.$$

Chapter 28

A brief introduction to Coding Theory

This chapter gives a short introduction to the combinatorial view of error-correcting codes. Our motivation is twofold: good error-correcting codes provide choices for the generators of generalized hypercubes that have high expansion, and in the next chapter we learn how to use expander graphs to construct good error-correcting codes.

We begin and end the chapter with a warning: the combinatorial, worst-case view of coding theory presented herein was very useful in the first few decades of the field. But, the problem of error-correction is at its heart probabilistic and great advances have been made by avoiding the worst-case formulation. For readers who would like to understand this perspective, we recommend "Modern Coding Theory" by Richardson and Urbanke. For those who wish to learn more about the worst-case approach, we recommend "The Theory of Error-Correcting Codes" by MacWilliams and Sloane.

28.1 Coding

Error-correcting codes are used to compensate for noise and interference in communication. They are used in practically all digital transmission and data storage schemes. We will only consider the problem of storing or transmitting bits[1], or maybe symbols from some small discrete alphabet.

The only type of interference we will consider is the flipping of bits. Thus, `0101` may become `1101`, but not `010`. More noise means more bits are flipped.

In our model problem, a transmitter wants to send m bits, which means that the transmitter's message is an element of $\mathbb{F}_2^m$. But, if the transmitter wants the receiver to correctly receive the message in the presence of noise, the transmitter should not send the plain message. Rather, the transmitter will send $n > m$ bits, encoded in such a way that the receiver can figure out what the message was even if there is a little bit of noise.

[1] Everything is bits. You think that's air you're breathing?

A naive way of doing this would be for the transmitter to send every bit 3 times. If only 1 bit were flipped during transmission, then the receiver would be able to figure out which one it was. But, this is a very inefficient coding scheme. Much better approaches exist.

28.2 Notation

When $\boldsymbol{x}$ is a vector, we let

$$|\boldsymbol{x}| \stackrel{\text{def}}{=} |\{a : \boldsymbol{x}(a) \neq 0\}|$$

denote the *hamming weight* of $\boldsymbol{x}$. This is often called the 0-norm, and written $\|\boldsymbol{x}\|_0$.

For a prime p, we denote the integers modulo p by $\mathbb{F}_p$. The reason is that the integers modulo p form the field with p elements: they may be summed and multiplied, have identities under addition and multiplication (0 and 1), the have inverses under addition $(-x)$, and all but zero have inverses under multiplication. We say *the* field because it is unique up to the names of the elements. In this chapter we mostly deal with the field of two elements $\mathbb{F}_2$, which we write $\mathbb{F}_2$.

28.3 Connection with Generalized Hypercubes

Recall that the Generalized Hypercubes we encountered in Section 7.4 have vertex set $\mathbb{F}_2^k$ and are defined by $d \geq k$ generators, $\boldsymbol{g}_1, \dots, \boldsymbol{g}_d \in \mathbb{F}_2^k$. For each $\boldsymbol{b} \in \mathbb{F}_2^k$, the graph defined by these generators has an adjacency matrix eigenvalue given by

$$\mu_{\boldsymbol{b}} = \sum_{i=1}^{d} (-1)^{\boldsymbol{g}_i^T \boldsymbol{b}}.$$

Let G be the d-by-k matrix whose ith row is $\boldsymbol{g}_i^T$. As $(-1)^x = 1 - 2x$, for $x \in \{0, 1\}$,

$$\mu_{\boldsymbol{b}} = \sum_{i=1}^{d} (-1)^{\boldsymbol{g}_i^T \boldsymbol{b}} = d - 2\,|G\boldsymbol{b}|\,.$$

The eigenvalue of d comes from $\boldsymbol{b} = \boldsymbol{0}$. If $G\boldsymbol{b}$ has small Hamming weight for every other vector $\boldsymbol{b}$, then all the other eigenvalues of the adjacency matrix will be small. We will see that this condition is satisfied when G is the generator matrix of a good code.

28.4 Hamming Codes

The first idea in coding theory was the parity bit. It allows one to detect one error. Let's say that the transmitter wants to send $b_1, \dots, b_m$. If the transmitter constructs

$$b_{m+1} = \sum_{i=1}^{m} b_i \mod 2, \tag{28.1}$$

and sends

$$b_1, \ldots, b_{m+1},$$

then the receiver will be able to detect one error, as it would cause (28.1) to be violated. But, the receiver won't know where the error is, and so won't be able to figure out the correct message unless it request a retransmit. And, of course, the receiver wouldn't be able to detect 2 errors.

Hamming codes combine parity bits in an interesting way to enable the receiver to correct one error. Let's consider the first interesting Hamming code, which transmits 4-bit messages by sending 7 bits in such a way that any one error can be corrected. Note that this is much better than repeating every bit 3 times, which would require 12 bits.

For reasons that will be clear soon, we will let b_3, b_5, b_6, and b_7 be the bits that the transmitter would like to send. The parity bits will be chosen by the rules

$$\begin{aligned} b_4 &= b_5 + b_6 + b_7 \\ b_2 &= b_3 + b_6 + b_7 \\ b_1 &= b_3 + b_5 + b_7. \end{aligned}$$

All additions, of course, are modulo 2. The transmitter will send the *codeword* $b_1, \ldots, b_7$.

If we write the bits as a vector, then we see that they satisfy the linear equations

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}.$$

For example, to transmit the message `1010`, we set

$$b_3 = 1, b_5 = 0, b_6 = 1, b_7 = 0,$$

and then compute

$$b_1 = 1, b_2 = 0, b_4 = 1.$$

Let's see what happens if some bit is flipped. Let the received transmission be $c_1, \ldots, c_7$, and assume that $c_i = b_i$ for all i except that $c_6 = 0$. This means that the parity check equations that involved the 6th bit will now fail to be satisfied, or

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix} \boldsymbol{c} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}.$$

Note that this is exactly the pattern of entries in the 6th column of the matrix. This will happen in general. If just one bit is flipped, and we multiply the received transmission by the matrix, the product will be the column of the matrix containing the flipped bit. As each column is different, we can tell which bit it was. To make this even easier, the columns have been arranged to be the binary representations of their index. For example, `110` is the binary representation of 6.

28.5 Terminology and Linear Codes

We will view an error-correcting code as a mapping

$$C : \mathbb{F}_2^m \to \mathbb{F}_2^n,$$

for n larger than m. Every string in the image of C is called a *codeword*. We will also abuse notation by identifying C with the set of codewords.

We define the *rate* of the code to be

$$r = \frac{m}{n}.$$

The rate of a code tells you how many bits of information you receive for each codeword bit. Of course, codes of higher rate are more efficient.

The *Hamming distance* between two words $\boldsymbol{c}^1$ and $\boldsymbol{c}^2$ is the number of bits in which they differ. It will be written

$$\mathrm{dist}(\boldsymbol{c}^1, \boldsymbol{c}^2) = \left| \boldsymbol{c}^1 - \boldsymbol{c}^2 \right|.$$

The *minimum distance* of a code is

$$d = \min_{\boldsymbol{c}^1 \neq \boldsymbol{c}^2 \in C} \mathrm{dist}(\boldsymbol{c}^1, \boldsymbol{c}^2)$$

(here we have used C to denote the set of codewords). It should be clear that if a code has large minimum distance then it is possible to correct many errors. In particular, it is possible to correct any number of errors less than $d/2$. To see why, let $\boldsymbol{c}$ be a codeword, and let $\boldsymbol{r}$ be the result of flipping $e < d/2$ bits of $\boldsymbol{c}$. As $\mathrm{dist}(\boldsymbol{c}, \boldsymbol{r}) < d/2$, $\boldsymbol{c}$ will be the closest codeword to $\boldsymbol{r}$. This is because for every $\boldsymbol{c}^1 \neq \boldsymbol{c}$,

$$d \leq \mathrm{dist}(\boldsymbol{c}^1, \boldsymbol{c}) \leq \mathrm{dist}(\boldsymbol{c}^1, \boldsymbol{r}) + \mathrm{dist}(\boldsymbol{r}, \boldsymbol{c}) < \mathrm{dist}(\boldsymbol{c}^1, \boldsymbol{r}) + d/2 \quad \text{implies} \quad d/2 < \mathrm{dist}(\boldsymbol{c}^1, \boldsymbol{r}).$$

So, large minimum distance is good.

The *minimum relative distance* of a code is

$$\delta = \frac{d}{n}.$$

It turns out that it is possible to keep both the rate and minimum relative distance of a code bounded below by constants, even as n grows. To formalize this notation, we will talk about a sequence of codes instead of a particular code. A sequence of codes $C_1, C_2, C_3, \dots$ is presumed to be a sequence of codes of increasing message lengths. Such a sequence is called *asymptotically good* if there are absolute constants r and δ such that for all i,

$$r(C_i) \geq r \qquad \text{and} \qquad \delta(C_i) \geq \delta.$$

One of the early goals of coding theory was to construct asymptotically good sequences of codes. Of course, one also needs to derive codes that have concise descriptions and that can be encoded and decoded efficiently.

A big step in this direction was the use of *linear codes* . In the same way that we defined Hamming codes, we may define a linear code as the set of vectors $\boldsymbol{c} \in \mathbb{F}_2^n$ such that $\boldsymbol{M}\boldsymbol{c} = \boldsymbol{0}$, for some $(n-m)$-by-n matrix $\boldsymbol{M}$. In this chapter, we will instead define a code by its generator matrix. Given an n-by-m matrix $\boldsymbol{G}$, we define the code $C_{\boldsymbol{G}}$ to be the of vectors of the form $\boldsymbol{G}\boldsymbol{b}$, where $\boldsymbol{b} \in \mathbb{F}_2^m$. One may view $\boldsymbol{b}$ as the message to be transmitted, and $\boldsymbol{G}\boldsymbol{b}$ as its encoding.

A linear code is called *linear* because the sum of two codewords in the code is always another codeword. In particular, $\boldsymbol{0}$ is always a codeword and the minimum distance of the code equals the minimum Hamming weight of a non-zero codeword, as

$$\mathrm{dist}(\boldsymbol{c}^1, \boldsymbol{c}^2) = \left|\boldsymbol{c}^1 - \boldsymbol{c}^2\right| = \left|\boldsymbol{c}^1 + \boldsymbol{c}^2\right|$$

over $\mathbb{F}_2$.

We now pause to make the connection back to generalized hypercubes: if $C_{\boldsymbol{G}}$ has minimum relative distance δ and maximum relative distance $1-\delta$, then the corresponding generalized hypercube is $1-2\delta$ expander.

28.6 Random Linear Codes

In the early years of coding theory, there were many papers published that contained special constructions of codes such as the Hamming code. But, as the number of bits to be transmitted became larger and larger, it became more and more difficult to find such exceptional codes. Thus, an asymptotic approach became reasonable. In his paper introducing coding theory, Shannon [Sha48] proved that random codes are asymptotically good. A few years later, Elias [Eli55] suggested using random linear codes.

We will now see that random linear codes are asymptotically good with high probability. We consider a code of the form $C_{\boldsymbol{G}}$, where $\boldsymbol{G}$ is an n-by-m matrix with independent uniformly chosen $\mathbb{F}_2$ entries. Clearly, the rate of the code will be m/n.

So, the minimum distance of $C_{\boldsymbol{G}}$ is

$$\min_{\boldsymbol{0} \neq \boldsymbol{b} \in \mathbb{F}_2^m} \mathrm{dist}(\boldsymbol{0}, \boldsymbol{G}\boldsymbol{b}) = \min_{\boldsymbol{0} \neq \boldsymbol{b} \in \mathbb{F}_2^m} \left|\boldsymbol{G}\boldsymbol{b}\right|,$$

where by $|\boldsymbol{c}|$ we mean the number of 1s in $\boldsymbol{c}$. This is sometimes called the *weight* of $\boldsymbol{c}$.

Here's what we can say about the minimum distance of a random linear code. The following argument is a refinement of the Chernoff based argument that appears in Section 7.5.

Lemma 28.6.1. *Let $\boldsymbol{G}$ be a random n-by-m matrix. For any d, the probability that $C_{\boldsymbol{G}}$ has minimum distance at least d is at least*

$$1 - \frac{2^m}{2^n} \sum_{i=0}^{d} \binom{n}{i}.$$

Proof. It suffices to upper bound the probability that there is some non-zero $\boldsymbol{b} \in \mathbb{F}_2^m$ for which

$$\left|\boldsymbol{G}\boldsymbol{b}\right| \leq d.$$

To this end, fix some non-zero vector $\boldsymbol{b}$ in $\mathbb{F}_2^m$. Each entry of $\boldsymbol{G}\boldsymbol{b}$ is the inner product of a column of $\boldsymbol{G}$ with $\boldsymbol{b}$. As each column of $\boldsymbol{G}$ consists of random $\mathbb{F}_2$ entries, each entry of $\boldsymbol{G}\boldsymbol{b}$ is chosen uniformly from $\mathbb{F}_2$. As the columns of $\boldsymbol{G}$ are chosen independently, we see that $\boldsymbol{G}\boldsymbol{b}$ is a uniform random vector in $\mathbb{F}_2^n$. Thus, the probability that $|\boldsymbol{G}\boldsymbol{b}|$ is at most d is precisely

$$\frac{1}{2^n}\sum_{i=0}^{d}\binom{n}{i}.$$

As the probability that one of a number of events holds is at most the sum of the probabilities that each holds (the "union bound"),

$$\begin{aligned}\Pr_{\boldsymbol{G}}\left[\exists \boldsymbol{b}\in\mathbb{F}_2^m, \boldsymbol{b}\neq\mathbf{0} : |\boldsymbol{G}\boldsymbol{b}|\leq d\right] &\leq \sum_{\mathbf{0}\neq\boldsymbol{b}\in\mathbb{F}_2^m}\Pr_{\boldsymbol{G}}\left[|\boldsymbol{G}\boldsymbol{b}|\leq d\right]\\ &\leq (2^m-1)\frac{1}{2^n}\sum_{i=0}^{d}\binom{n}{i}.\\ &\leq \frac{2^m}{2^n}\sum_{i=0}^{d}\binom{n}{i}.\end{aligned}$$

□

To see how this behaves asymptotically, recall that for a constant p,

$$\binom{n}{pn}\approx 2^{nH(p)},$$

where

$$H(p)\overset{\text{def}}{=}-p\log_2 p-(1-p)\log_2(1-p)$$

is the *binary entropy function.* If you are not familiar with this, you may derive it from Stirling's formula. For our purposes $2^{nH(p)}\approx\sum_{i=0}^{pn}\binom{n}{i}$. Actually, we will just use the fact that for $\beta > H(p)$,

$$\frac{\sum_{i=0}^{pn}\binom{n}{i}}{2^{n\beta}}\to 0$$

as n goes to infinity.

If we set $m = rn$ and $d = \delta n$, then Lemma 28.6.1 tells us that $C_{\boldsymbol{G}}$ probably has rate r and minimum relative distance δ if

$$\frac{2^{rn}}{2^n}2^{nH(\delta)}<1,$$

which happens when

$$H(\delta)<1-r.$$

For any constant $r<1$, we can find a δ for which $H(\delta)<1-r$, so there exist asymptotically good sequences of codes of every non-zero rate. This is called the Gilbert-Varshamov bound. It is still not known if binary codes exist whose relative minimum distance satisfies $H(\delta)>1-r$. This is a big open question in coding theory.

Of course, this does not tell us how to choose such a code in practice, how to efficiently check if a given code has large minimum distance, or how to efficiently decode such a code.

28.7 Reed-Solomon Codes

Reed-Solomon Codes are one of the workhorses of coding theory. The are simple to describe, and easy to encode and decode.

However, Reed-Solomon Codes are not binary codes. Rather, they are codes whose symbols are elements of a finite field. If you don't know what a finite field is, don't worry (yet). For now, we will just consider prime fields, F_p. These are the numbers modulo a prime p. Recall that such numbers may be added, multiplied, and *divided*.

A message in a Reed-Solomon code over a field F_p is identified with a polynomial of degree $m-1$. That is, the message $f_1, \ldots, f_m$ is viewed as providing the coefficients of the polynomial

$$Q(x) = \sum_{i=0}^{m-1} f_{i+1} x^i.$$

A Reed-Solomon code is encoded by evaluating it over every element of the field. That is, the codeword is

$$Q(0), Q(1), Q(2), \ldots, Q(p-1).$$

Sometimes, it is evaluated at a subset of the field elements.

We will now see that the minimum distance of such a Reed-Solomon code is $p-m$. We show this using the following standard fact from algebra.

Lemma 28.7.1. *Let Q be a polynomial of degree at most $m-1$ over a field F_p. If there exists distinct field elements $x_1, \ldots, x_m$ such that*

$$Q(x_i) = 0$$

then Q is identically zero.

Theorem 28.7.2. *The minimum distance of the Reed-Solomon code is at least $p-m$.*

Proof. Let Q^1 and Q^2 be two different polynomials of degree at most $m-1$. For a polynomial Q, let

$$E(Q) = (Q(0), Q(1), \ldots, Q(p))$$

be its encoding. If

$$\mathrm{dist}(E(Q^1), E(Q^2)) \leq p-k,$$

then there exists field elements $x_1, \ldots, x_k$ such that

$$Q^1(x_j) = Q^2(x_j).$$

Now, consider the polynomial

$$Q^1(x) - Q^2(x).$$

It also has degree at most $m-1$, and it is zero at k field elements. Lemma 28.7.1 tells us that if $k \geq m$, then $Q^1 - Q^2$ is exactly zero, which means that $Q^1 = Q^2$. Thus, for distinct Q^1 and Q^2, it must be the case that

$$\mathrm{dist}(E(Q^1), E(Q^2)) > p-m.$$

□

However, Reed-Solomon codes do not provide an asymptotically good family. If one represents each field element by $\log_2 p$ bits in the obvious way, then the code has length $p \log_2 p$, but can only correct at most p errors. That said, one can find an asymptotically good family by encoding each field element with its own small error-correcting code.

Next lecture, we will see how to make asymptotically good codes out of expander graphs. In the following lecture, we will use good error-correcting codes to construct graphs.

28.8 Caution

Explain defects of the worst-case view.

Chapter 29

Expander Codes

In this Chapter we will learn how to use expander graphs to construct and decode asymptotically good error correcting codes.

29.1 Bipartite Expander Graphs

Our construction of error-correcting codes will exploit bipartite expander graphs (as these give a much cleaner construction than the general case). Let's begin by examining what a bipartite expander graph should look like. It's vertex set will have two parts, U and V, each having n vertices. Every vertex will have degree d, and every edge will go from a vertex in U to a vertex in V.

In the same way that we view ordinary expanders as approximations of complete graphs, we will view bipartite expanders as approximations of complete bipartite graphs[1]. That is, if we let $K_{n,n}$ denote the complete bipartite graph, then we want a d-regular bipartite graph G such that

$$(1-\epsilon)\frac{d}{n}K_{n,n} \preccurlyeq G \preccurlyeq (1+\epsilon)\frac{d}{n}K_{n,n}.$$

As the eigenvalues of the Laplacian of $\frac{d}{n}K_{n,n}$ are 0 and $2d$ with multiplicity 1 each, and d otherwise, this means that we want a d-regular graph G whose Laplacian spectrum satisfies

$$\lambda_1 = 0, \quad \lambda_{2n} = 2d, \quad \text{and } |\lambda_i - d| \leq \epsilon d, \text{for all } 1 < i < 2n.$$

We can obtain such a graph by taking the *double-cover* of an ordinary expander graph.

Definition 29.1.1. *Let $G = (V, E)$ be a graph. The double-cover of G is the graph with vertex set $V \times \{0, 1\}$ and edges*

$$((a, 0), (b, 1)), \quad \textit{for } (a, b) \in E.$$

It is easy to determine the eigenvalues of the double-cover of a graph.

[1] The complete bipartite graph contains all edges between U and V

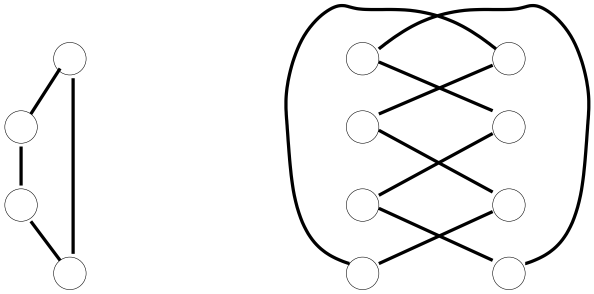


Figure 29.1: The cycle on 4 vertices, and its double-cover

Proposition 29.1.2. *Let H be the double-cover of G. Then, for every eigenvalue λ_i of the Laplacian of G, H has a pair of eigenvalues,*

$$\lambda_i \quad \textit{and} \quad 2d - \lambda_i.$$

The easiest way to prove this is to observe that if A is the adjacency matrix of G, then the adjacency matrix of H looks like

$$\begin{pmatrix} \mathbf{0} & A \\ A & \mathbf{0} \end{pmatrix}.$$

Our analysis of error-correcting codes will exploit the following theorem, which is analogous to Theorem 10.2.1.

Theorem 29.1.3. *Let $G = (U \cup V, E)$ be a d-regular bipartite graph that ϵ-approximates $\frac{d}{n}K_{n,n}$. Then, for all $S \subseteq U$ and $T \subseteq V$,*

$$\left| |E(S,T)| - \frac{d}{n}|S|\,|T| \right| \leq \epsilon d \sqrt{|S|\,|T|}.$$

Proof. Similar to the proof of Theorem 27.3.1. □

Let $G(S \cup T)$ denote the graph induced on vertex set $S \cup T$. We use the following simple corollary of Theorem 29.1.3.

Corollary 29.1.4. *For $S \subseteq U$ with $|S| = \sigma n$ and and $T \subseteq V$ with $|T| = \tau n$, the average degree of vertices in $G(S \cup T)$ is at most*

$$\frac{2d\sigma\tau}{\sigma + \tau} + \epsilon d.$$

Proof. The average degree of a graph is twice its number of edges, divided by the number of vertices. In our case, this is at most

$$\frac{2d}{n}\frac{|S|\,|T|}{|S| + |T|} + 2\epsilon d \frac{\sqrt{|S|\,|T|}}{|S| + |T|}.$$

The left-hand term is

$$\frac{2d\sigma\tau}{\sigma+\tau},$$

and the right-hand term is at most

$$\epsilon d.$$

□

29.2 Building Codes

Our construction of error-correcting codes will require two ingredients: a d-regular bipartite expander graph G on $2n$ vertices, and a linear error correcting code C_0 of length d. We will combine these to construct an error correcting code of length dn. We think of the code C_0 as being a small code that drives the construction. This is reasonable as we will keep d a small constant while n grows.

In our construction of the code, we associate one bit with each edge of the graph. As the graph has dn edges, this results in dn bits, which we label $y_1, \dots, y_{dn}$. We now describe the code by listing the linear constraints its codewords must satisfy. Each vertex requires that the bits on its attached edges resemble a codeword in the code C_0. That is, each vertex should list its attached edges in some order (which order doesn't matter, but it should be fixed). As a vertex has d attached edges, it is easy to require that the d bits on these edges are a codeword in the code C_0.

Let r_0 be the rate of code C_0. This means that the space of codewords has dimension $r_0 d$. But, since C_0 is a linear code, it means that its codewords are exactly the the vectors that satisfy some set of $d(1-r_0)$ linear equations. As there are $2n$ vertices in the graph, the constraints imposed by each vertex impose $2nd(1-r_0)$ linear constraints on the dn bits. Thus, the vector space of codewords that satisfy all of these constraints has dimension at least

$$dn - 2dn(1-r_0) = dn(2r_0 - 1),$$

and the code we have constructed has rate at least

$$r = 2r_0 - 1.$$

So, this rate will be a non-zero constant as long as $r_0 > 1/2$.

For the rest of the lecture, we will let C denote the resulting expander code.

29.3 Encoding

We have described the set of codewords, but have not said how one should encode. As the code is linear, it is relatively easy to find a way to encode it. In particular, one may turn the above description of the code into a matrix M with dn columns and $2dn(1-r_0)$ rows such that the codewords are precisely those $\boldsymbol{y}$ such that

$$M\boldsymbol{y} = \boldsymbol{0}.$$

So, the codewords form a vector space of dimension $dn(2r_0 - 1)$, and so there is a matrix G with $dn(2r_0 - 1)$ columns and dn rows for which the codewords are precisely the vectors $G\boldsymbol{x}$, for $\boldsymbol{x} \in \{0, 1\}^{dn(2r_0 - 1)}$. In fact, there are many such matrices G, and they are called *generator matrices* for the code. Such a matrix G may be computed from M by elementary linear algebra.

29.4 Minimum Distance

We will now see that if C_0 is a good code, then C has large minimum distance. Let $\delta_0 d$ be the minimum distance of the code C_0. You should think of δ_0 as being a constant.

Theorem 29.4.1. *If $\epsilon \leq \delta_0/2$, then the minimum relative distance δ of C satisfies*

$$\delta \geq \delta_0^2/2.$$

Proof. As C is a linear code, it suffices to prove that C has no nonzero codewords of small Hamming weight. To this end, we identify a codeword with the set of edges on which its bits are 1. Let F be such a set of edges, and let $|F| = \phi dn$. As the minimum distance of C_0 is $\delta_0 d$, every vertex v that is attached to an edge of F must be attached to at least $\delta_0 d$ edges of F. Let S be the subset of vertices of U adjacent to edges in F, and let T be the corresponding subset of V. We have just argued that every vertex in $G(S \cup T)$ must have degree at least $\delta_0 d$, and so in particular the average degree of $G(S \cup T)$ is at least $\delta_0 d$.

We may also use this fact to see that

$$|S|, |T| \leq \frac{|F|}{\delta_0 d}.$$

Setting $\sigma = |S|/n$ and $\tau = |T|/n$, the previous inequality becomes

$$\sigma, \tau \leq \frac{\phi}{\delta_0}.$$

Corollary 29.1.4 tells us that the average degree of $G(S \cup T)$ is at most

$$\frac{2d\sigma\tau}{\sigma + \tau} + \epsilon d.$$

As

$$2\sigma\tau \leq \sigma^2 + \tau^2 \leq \frac{\phi}{\delta_0}(\sigma + \tau),$$

the average degree of $G(S \cup T)$ is at most

$$d\frac{\phi}{\delta_0} + \epsilon d.$$

Combining the upper and lower bounds on the average degree of $G(S \cup T)$, we obtain

$$\delta_0 d \leq d\frac{\phi}{\delta_0} + \epsilon d,$$

which implies

$$\delta_0(\delta_0 - \epsilon) \leq \phi.$$

The assumption $\epsilon \leq \delta_0/2$ then yields

$$\phi \geq \delta_0^2/2.$$

As we assumed that F was the set of edges corresponding to a codeword and that $|F| = \phi dn$, we have shown that the minimum relative distance of C is at least $\delta_0^2/2$. □

29.5 Decoding

We will convert an algorithm that corrects errors in C_0 into an algorithm for correcting errors in C. The construction is fairly simple. We first apply the decoding algorithm at every vertex in U. We then do it at every vertex in V. We alternate in this fashion until we produce a codeword.

To make this more concrete, assume that we have an algorithm A that corrects up to $\delta_0 d/2$ errors in the code C_0. That is, on input any word $\boldsymbol{r} \in \{0,1\}^d$, A outputs another word in $\{0,1\}^d$ with the guarantee that if there is a $\boldsymbol{c} \in C_0$ such that $\operatorname{dist}(\boldsymbol{c}, \boldsymbol{r}) \leq \delta_0 d/2$, then A outputs $\boldsymbol{c}$. We apply the transformation A independently to the edges attached to each vertex of U. We then do the same for V, and then alternate sides for a logarithmic number of iterations. We refer to these alternating operations as $U-$ and V-decoding steps

We will prove that if $\epsilon \leq \delta_0/3$ then this algorithm will correct up to $\delta_0^2 dn/18$ errors in at most $\log_{4/3} n$ iterations. The idea is to keep track of which vertices are attached to edges that contain errors, rather than keeping track of the errors themselves. We will exploit the fact that any vertex that is attached to few edges in error will correct those errors. Let S be the set of vertices attached to edges in error after a U-decoding step. We will show that the set T of vertices attached to edges in error after the next V-decoding step will be much smaller.

Lemma 29.5.1. *Assume that $\epsilon \leq \delta_0/3$. Let $F \subset E$ be a set of edges, let S be the subset of vertices in U attached to edges in F and let T be the subset of vertices in V attached to at least $\delta_0 d/2$ edges in F. If*

$$|S| \leq \delta_0 n/9,$$

then

$$|T| \leq \frac{3}{4}|S|.$$

Proof. Let $|S| = \sigma n$ and $|T| = \tau n$. We have $|F| \geq (\delta_0 d/2)|T|$. As the average degree of $G(S \cup T)$ is twice the number of edges in the subgraph divided by the number of vertices, it is at least

$$\frac{\delta_0 d|T|}{|S| + |T|} = \frac{\delta_0 d\tau}{\sigma + \tau}.$$

Applying Corollary 29.1.4, we find

$$\frac{\delta_0 d\tau}{\sigma + \tau} \leq \frac{2d\sigma\tau}{\sigma + \tau} + \epsilon d.$$

This implies

$$\delta_0 \tau \leq 2\sigma\tau + \epsilon(\sigma + \tau),$$

which becomes

$$\tau \leq \frac{\epsilon\sigma}{\delta_0 - 2\sigma - \epsilon}.$$

Recalling that $\sigma \leq \delta_0/9$ and $\epsilon \leq \delta_0/3$, we obtain

$$\tau \leq \sigma \frac{\delta_0/3}{\delta_0(4/9)} \leq \frac{3}{4}\sigma.$$

□

Lemma 29.5.2. *Assume that $\epsilon \leq \delta_0/3$. Let F be the set of edges in error after a U-decoding step, and let S be the set of vertices in U attached to F. Now, perform a V-decoding step and let T be the set of vertices in V attached to edges in error afterwards. If*

$$|S| \leq \delta_0 n/9,$$

then

$$|T| \leq \frac{3}{4} |S| .$$

Proof. Every vertex in V that outputs an error after the V-decoding step must be attached to at least $\delta_0 d/2$ edges of F. Moreover, each of these edges is attached to a vertex of S. Thus, the lemma follows immediately from Lemma 29.5.1. □

Theorem 29.5.3. *If $\epsilon \leq \delta_0/3$, then the proposed decoding algorithm will correct every set of at most*

$$\frac{\delta_0^2}{18} dn$$

errors.

Proof. Let F denote the set of edges that are initially in error. Let S denote the set of vertices that output errors after the first U-decoding step. Every vertex in S must be adjacent to at least $\delta d/2$ edges in F, so

$$|F| \leq \frac{\delta_0^2}{18} dn \quad \Longrightarrow \quad |S| \leq \frac{|F|}{\delta_0 d/2} \leq \delta_0 n/9.$$

After this point, we may apply Lemma 29.5.2 to show that the decoding process converges in at most $\log_{4/3} n$ iterations. □

29.6 Historical Notes

Gallager [Gal63] first used graphs to construct error-correcting codes. His graphs were also bipartite, with one set of vertices representing bits and the other set of vertices representing constraints. Tanner [Tan81] was the first to put the vertices on the edges. The use of expansion in analyzing these codes we pioneered by Sipser and Spielman [SS96]. The construction we present here is due to Zemor [Zem01], although he presents a tighter analysis. Improved constructions and analyses may be found in [BZ02, BZ05, BZ06, AS06].

Surprisingly, encoding these codes is slower than decoding them. As the matrix G will be dense, leading to an encoding algorithm that takes time $\Theta((dn)^2)$. Of course, one would prefer to encode them in time $O(dn)$. Using Ramanujan expanders and the Fast Fourier Transform over the appropriate groups, Lafferty and Rockmore [LR97] reduced the time for encoding to $O(d^2 n^{4/3})$. Spielman [Spi96a] modifies the code construction to obtain codes with similar performance that may be encoded in linear time.

Related ideas have been used to design codes that approach channel capacity. See [**?**, **?**, **?**, **?**].

Chapter 30

A simple construction of expander graphs

Lecture 18 from October 31, 2018

30.1 Overview

Our goal is to prove that for every $\epsilon > 0$ there is a d for which we can efficiently construct an infinite family of d-regular ϵ-expanders. I recall that these are graphs whose adjacency matrix eigenvalues satisfy $|\mu_i| \leq \epsilon d$ and whose Laplacian matrix eigenvalues satisfy $|d - \lambda_i| \leq \epsilon d$. Viewed as a function of ϵ, the d that we obtain in this construction is rather large. But, it is a constant. The challenge here is to construct infinite families with fixed d and ϵ.

Before we begin, I remind you that in Lecture 5 we showed that random generalized hybercubes were ϵ expanders of degree $f(\epsilon) \log n$, for some function f. The reason they do not solve today's problem is that their degrees depend on the number of vertices. However, today's construction will require some small expander graph, and these graphs or graphs like them can serve in that role. So that we can obtain a construction for every number of vertices n, we will exploit random generalized ring graphs. Their analysis is similar to that of random generalized hypercubes.

Claim 30.1.1. *There exists a function $f(\epsilon)$ so that for every $\epsilon > 0$ and every sufficiently large n the Cayley graph with group $\mathbb{Z}/n$ and a random set of at least $f(\epsilon) \log n$ generators is an ϵ-expander with high probability.*

I am going to present the simplest construction of expanders that I have been able to find. By "simplest", I mean optimizing the tradeoff of simplicity of construction with simplicity of analysis. It is inspired by the Zig-Zag product and replacement product constructions presented by Reingold, Vadhan and Wigderson [RVW02].

For those who want the quick description, here it is. Begin with an expander. Take its line graph.

Observe that the line graph is a union of cliques. So, replace each clique by a small expander. We need to improve the expansion slightly, so square the graph. Square one more time. Repeat.

The analysis will be simple because all of the important parts are equalities, which I find easier to understand than inequalities.

While this construction requires the choice of two expanders of constant size, it is explicit in the sense that we can obtain a simple *implicit* representation of the graph: if the name of a vertex in the graph is written using b bits, then we can compute its neighbors in time polynomial in b.

30.2 Squaring Graphs

We will first show that we can obtain a family of ϵ expanders from a family of β-expanders for any $\beta < 1$. The reason is that squaring a graph makes it a better expander, although at the cost of increasing its degree.

Given a graph G, we define the graph G^2 to be the graph in which vertices u and v are connected if they are at distance 2 in G. Formally, G^2 should be a weighted graph in which the weight of an edge is the number of such paths. When first thinking about this, I suggest that you ignore the issue. When you want to think about it, I suggest treating such weighted edges as multiedges.

We may form the adjacency matrix of G^2 from the adjacency matrix of G. Let $\boldsymbol{M}$ be the adjacency matrix of G. Then $\boldsymbol{M}^2(u,v)$ is the number of paths of length 2 between u and v in G, and $\boldsymbol{M}^2(v,v)$ is always d. We will eliminate those self-loops. So,

$$\boldsymbol{M}_{G^2} = \boldsymbol{M}_G^2 - dI_n.$$

If G has no cycles of length up to 4, then all of the edges in its square will have weight 1. The following claim is immediate from this definition.

Claim 30.2.1. *The adjacency matrix eigenvalues of G^2 are precisely*

$$\mu_i^2 - d,$$

where $\mu_1, \ldots, \mu_n$ are the adjacency matrix eigenvalues of G.

Lemma 30.2.2. *If $\{G_i\}_i$ is an infinite family of d-regular β-expanders for $\beta \geq 1/\sqrt{d-1}$, then $\left\{G_i^2\right\}_i$ is an infinite family of $d(d-1)$-regular β^2 expanders.*

We remark that the case of $\beta > 1/\sqrt{d-1}$, or even larger, is the case of interest. We are not expecting to work with graphs that beat the Ramanujan bound, $2\sqrt{d-1}/d$.

Proof. For μ an adjacency matrix eigenvalue of G_i other than d, we have

$$\frac{\mu^2 - d}{d(d-1)} = \frac{\mu^2 - d}{d^2 - d} \leq \frac{\mu^2}{d^2} \leq \beta^2.$$

On the other hand, every adjacency eigenvalue of G_i^2 is at least $-d$, which is at least $-\beta^2 d(d-1)$. □

So, by squaring enough times, we can convert a family of β expanders for any $\beta < 1$ into a family of ϵ expanders.

30.3 The Relative Spectral Gap

To measure the qualities of the graphs that appear in our construction, we define a quantity that we will call the *relative spectral gap* of a d-regular graph:

$$r(G) \overset{\text{def}}{=} \min\left(\frac{\lambda_2(G)}{d}, \frac{2d - \lambda_n}{d}\right).$$

The graphs with larger relative spectral gaps are better expanders. An ϵ-expander has relative spectral gap at least $1 - \epsilon$, and vice versa. Because we can square graphs, we know that it suffices to find an infinite family of graphs with relative spectral gap strictly greater than 0.

We now state exactly how squaring impacts the relative spectral gap of a graph.

Corollary 30.3.1. *If G has relative spectral gap β, then G^2 has relative spectral gap at least*

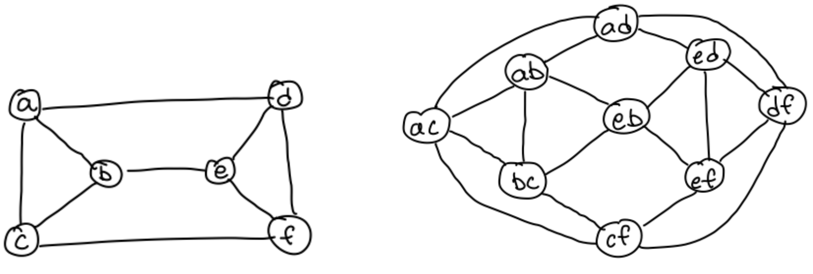
$$2\beta - \beta^2.$$

Note that when β is small, this gap is approximately 2β.

30.4 Line Graphs

Our construction will leverage small expanders to make bigger expanders. To begin, we need a way to make a graph bigger and still say something about its spectrum.

We use the *line graph* of a graph. Let $G = (V, E)$ be a graph. The line graph of G is the graph whose vertices are the edges of G in which two are connected if they share an endpoint in G. That is, $\big((u, v), (w, z)\big)$ is an edge of the line graph if one of $\{u, v\}$ is the same as one of $\{w, z\}$. The line graph is often written $L(G)$, but we won't do that in this class so that we can avoid confusion with the Laplacian.



(a) A graph

(b) Its line graph.

Let G be a d-regular graph with n vertices, and let H be its line graph[1].As G has $dn/2$ edges, H has $dn/2$ vertices. Each vertex of H, say (u, v), has degree $2(d-1)$: $d-1$ neighbors for the other edges attached to u and $d-1$ for v. In fact, if we just consider one vertex u in V, then all vertices in H of form (u, v) of G will be connected. That is, H contains a d-clique for every vertex in V. We see that each vertex of H is contained in exactly two of these cliques.

Here is the great fact about the spectrum of the line graph.

Lemma 30.4.1. *Let G be a d-regular graph with n vertices, and let H be its line graph. Then the spectrum of the Laplacian of H is the same as the spectrum of the Laplacian of G, except that it has $dn/2 - n$ extra eigenvalues of $2d$.*

Before we prove this lemma, we need to recall the factorization of a Laplacian as the product of the signed edge-vertex adjacency matrix times its transpose. We reserved the letter $\boldsymbol{U}$ for this matrix, and defined it by

$$\boldsymbol{U}((a,b),c) = \begin{cases} 1 & \text{if } a = c \\ -1 & \text{if } b = c \\ 0 & \text{otherwise.} \end{cases}$$

For an unweighted graph, we have

$$\boldsymbol{L}_G = \boldsymbol{U}^T \boldsymbol{U}.$$

Recall that each edge indexes one column, and that we made an arbitrary choice when we ordered the edge (a, b) rather than (b, a). But, this arbitrary choice factors out when we multiply by $\boldsymbol{U}^T$.

30.5 The Spectrum of the Line Graph

Define the matrix $|\boldsymbol{U}|$ to be the matrix obtained by replacing every entry of $\boldsymbol{U}$ by its absolute value. Now, consider $|\boldsymbol{U}|^T |\boldsymbol{U}|$. It looks just like the Laplacian, except that all of its off-diagonal entries are 1 instead of -1. So,

$$|\boldsymbol{U}|^T |\boldsymbol{U}| = \boldsymbol{D}_G + \boldsymbol{M}_G = d\boldsymbol{I} + \boldsymbol{M}_G,$$

as G is d-regular. We will also consider the matrix $|\boldsymbol{U}| |\boldsymbol{U}|^T$. This is a matrix with $nd/2$ rows and $nd/2$ columns, indexed by edges of G. The entry at the intersection of row (u, v) and column (w, z) is

$$(\boldsymbol{\delta}_u + \boldsymbol{\delta}_v)^T (\boldsymbol{\delta}_w + \boldsymbol{\delta}_z).$$

So, it is 2 if these are the same edge, 1 if they share a vertex, and 0 otherwise. That is

$$|\boldsymbol{U}| |\boldsymbol{U}|^T = 2I_{nd/2} + \boldsymbol{M}_H.$$

Moreover, $|\boldsymbol{U}| |\boldsymbol{U}|^T$ and $|\boldsymbol{U}|^T |\boldsymbol{U}|$ have the same eigenvalues, except that the later matrix has $nd/2 - n$ extra eigenvalues of 0.

[1] If G has multiedges, which is how we interpret integer weights, then we include a vertex in the line graph for each of those multiedges. These will be connected to each other by edges of weight two—one for each vertex that they share. All of the following statements then work out.

Proof of Lemma 30.4.1. First, let λ_i be an eigenvalue of $\boldsymbol{L}_G$. We see that

$$\begin{aligned}
&\lambda_i \text{ is an eigenvalue of } \boldsymbol{D}_G - \boldsymbol{M}_G \implies\\
&d - \lambda_i \text{ is an eigenvalue of } \boldsymbol{M}_G \implies\\
&2d - \lambda_i \text{ is an eigenvalue of } \boldsymbol{D}_G + \boldsymbol{M}_G \implies\\
&2d - \lambda_i \text{ is an eigenvalue of } 2I_{nd/2} + \boldsymbol{M}_H \implies\\
&2(d-1) - \lambda_i \text{ is an eigenvalue of } \boldsymbol{M}_H \implies\\
&\lambda_i \text{ is an eigenvalue of } \boldsymbol{D}_H - \boldsymbol{M}_H.
\end{aligned}$$

Of course, this last matrix is the Laplacian matrix of H. We can similarly show that the extra $dn/2 - n$ zero eigenvalues of $2I_{nd/2} + \boldsymbol{M}_H$ become $2d$ in $\boldsymbol{L}_H$. □

While the line graph operation preserves λ_2, it causes the degree of the graph to grow. So, we are going to need to do more than just take line graphs to construct expanders.

Proposition 30.5.1. *Let G be a d-regular graph with $d \geq 7$ and let H be its line graph. Then,*

$$r(H) = \frac{\lambda_2(G)}{2(d-1)} \geq r(G)/2.$$

Proof. For G a d-regular graph other than K_{d+1}, $\lambda_2(G) \leq d+1$. By the Perron-Frobenius theorem (Lemma 6.A.1) $\lambda_{max}(G) \leq 2d$ (with equality if and only G is bipartite). So, $\lambda_{max}(H) = 2d$ and $\lambda_2(H) = \lambda_2(G) \leq d$. So, the term in the definition of the relative spectral gap corresponding to the largest eigenvalue of H satisfies

$$\frac{2(2d-2) - \lambda_{max}(H)}{2d-2} = \frac{2(2d-2) - 2d}{2d-2} = 1 - \frac{2}{d} \geq 5/7,$$

as $d \geq 7$. On the other hand,

$$\frac{\lambda_2(H)}{2d-2} \leq \frac{d}{2d-2} \leq 2/3.$$

As $2/3 < 5/7$,

$$\min\left(\frac{\lambda_2(H)}{2d-2}, \frac{2(2d-2) - \lambda_{max}(H)}{2d-2}\right) = \frac{\lambda_2(H)}{2d-2} = \frac{\lambda_2(G)}{2d-2} \geq r(G/2).$$

□

While the line graph of G has more vertices, its degree is higher and its relative spectral gap is approximately half that of G. We can improve the relative spectral gap by squaring. In the next section, we show how to lower the degree.

30.6 Approximations of Line Graphs

Our next step will be to construct approximations of line graphs. We already know how to approximate complete graphs: we use expanders. As line graphs are sums of complete graphs, we will approximate them by sums of expanders. That is, we replace each clique in the line graph by an expander on d vertices. Since d will be a constant in our construction, we will be able to get these small expanders from known constructions, like the random generalized ring graphs.

Let G be a d-regular graph and let Z be a graph on d vertices of degree k (we will use a low-degree expander). We define the graph

$$G \textcircled{L} Z$$

to be the graph obtained by forming the edge graph of G, H, and then replacing every d-clique in H by a copy of Z. Actually, this does not uniquely define $G \textcircled{L} Z$, as there are many ways to replace a d-clique by a copy of Z. But, any choice will work. Note that every vertex of $G \textcircled{L} Z$ has degree $2k$.

Lemma 30.6.1. *Let G be a d-regular graph, let H be the line graph of G, and let Z be a k-regular α-expander. Then,*

$$(1-\alpha)\frac{k}{d}H \preccurlyeq G \textcircled{L} Z \preccurlyeq (1+\alpha)\frac{k}{d}H$$

Proof. As H is a sum of d-cliques, let $H_1, \dots, H_n$ be those d-cliques. So,

$$\boldsymbol{L}_H = \sum_{i=1}^{n} \boldsymbol{L}_{H_i}.$$

Let Z_i be the graph obtained by replacing H_i with a copy of Z, on the same set of vertices. To prove the lower bound, we compute

$$\boldsymbol{L}_{G \textcircled{L} Z} = \sum_{i=1}^{n} \boldsymbol{L}_{Z_i} \succcurlyeq (1-\alpha)\frac{k}{d}\sum_{i=1}^{n} \boldsymbol{L}_{H_i} = (1-\alpha)\frac{k}{d}\boldsymbol{L}_H.$$

The upper bound is proved similarly. □

Corollary 30.6.2. *Under the conditions of Lemma 30.6.1,*

$$r(G \textcircled{L} Z) \geq \frac{1-\alpha}{2} r(G).$$

Proof. The proof is similar to the proof of Proposition 30.5.1. We have

$$\lambda_2(G \textcircled{L} Z) \geq (1-\alpha)\frac{k\lambda_2(G)}{d},$$

and

$$\lambda_{max}(G \textcircled{L} Z) \leq (1+\alpha)2k.$$

So,

$$\min\left(\lambda_2(G \textcircled{L} Z), 2(2k) - \lambda_{max}(G \textcircled{L} Z)\right) \geq \min\left((1-\alpha)\frac{k\lambda_2(G)}{d}, (1-\alpha)2k\right) = (1-\alpha)\frac{k\lambda_2(G)}{d},$$

as $\lambda_2(G) \leq d$. So,

$$r(G \textcircled{L} Z) \geq \frac{1}{2k}(1-\alpha)kr(G) = \frac{1-\alpha}{2}r(G).$$

□

So, the relative spectral gap of $G \textcircled{L} Z$ is a little less than half that of G. But, the degree of $G \textcircled{L} Z$ is $2k$, which we will arrange to be much less than the degree of G, d.

We will choose k and d so that squaring this graph improves its relative spectral gap, but still leaves its degree less than d. If G has relative spectral gap β, then G^2 has relative spectral gap at least

$$2\beta - \beta^2.$$

It is easy to see that when β is small, this gap is approximately 2β. This is not quite enough to compensate for the loss of $(1-\epsilon)/2$ in the corollary above, so we will have to square the graph once more.

30.7 The whole construction

To begin, we need a "small" k-regular expander graph Z on

$$d \overset{\text{def}}{=} (2k(2k-1))^2 - 2k(2k-1)$$

vertices. It should be an ϵ-expander for some small ϵ. I believe that $\epsilon = 1/6$ would suffice. The other graph we will need to begin our construction will be a small d-regular expander graph G_0. We use Claim 30.1.1 to establish the existence of both of these. Let β be the relative spectral gap of G_0. We will assume that β is small, but greater than 0. I believe that $\beta = 1/5$ will work. Of course, it does not hurt to start with a graph of larger relative spectral gap.

We then construct $G_0 \textcircled{L} Z$. The degree of this graph is $2k$, and its relative spectral gap is a little less than $\beta/2$. So, we square the resulting graph, to obtain

$$(G_0 \textcircled{L} Z)^2.$$

It has degree approximately $4k^2$, and relative spectral gap slightly less than β. But, for induction, we need it to be more than β. So, we square one more time, to get a relative spectral gap a little less than 2β. We now set

$$G_1 = \left((G_0 \textcircled{L} Z)^2\right)^2.$$

The graph G_1 is at least as good an approximation of a complete graph as G_0, and it has degree approximately $16k^4$. In general, we set

$$G_{i+1} = \left((G_i \textcircled{L} Z)^2\right)^2.$$

To make the inductive construction work, we need for Z to be a graph of degree k whose number of vertices equals the degree of G. This is approximately $16k^4$, and is exactly

$$(2k(2k-1))^2 - 2k(2k-1).$$

I'll now carry out the computation of relative spectral gaps with more care. Let's assume that G_0 has a relative spectral gap of $\beta \geq 4/5$, and assume, by way of induction, that $\rho(G_i) \geq 4/5$. Also assume that Z is a 1/6-expander. We then find

$$r(G_i \textcircled{L} Z) \geq (1-\epsilon)(4/5)/2 = 1/3.$$

So, $G_i \textcircled{L} Z$ is a 2/3-expander. Our analysis of graph squares then tells us that G_{i+1} is a $(2/3)^4$-expander. So,

$$r(G_{i+1}) \geq 1 - (2/3)^4 = 65/81 > 4/5.$$

By induction, we conclude that every G_i has relative spectral gap at least 4/5.

To improve their relative spectral gaps of the graphs we produce, we can just square them a few times.

30.8 Better Constructions

There is a better construction technique, called the Zig-Zag product [RVW02]. The Zig-Zag construction is a little trickier to understand, but it achieves better expansion. I chose to present the line-graph based construction because its analysis is very closely related to an analysis of the Zig-Zag product.

Chapter 31

PSRGs via Random Walks on Graphs

31.1 Overview

There are three major approaches to designing pseudo-random generators (PSRGs). The most common is to use quick procedures that seem good enough. This is how the PSRGs that are standard in most languages arise. Cryptographers and Complexity Theorists try to design PSRGs that work for every polynomial-time algorithm. For example, one can construct PSRGs from cryptographic functions with the guarantee that if the output of a polynomial-time algorithm differs from random when using the PSRG, then one can use it to break the cryptographic function (see [HILL99, Gol07]). In this chapter we consider the construction of PSRGs that can be proved to work for specific algorithms or algorithms of specific forms. In particular, we will see w Impagliazzo and Zuckerman's [IZ89] approach of using of random walks on expanders to run the same algorithm many times. We are going to perform a very crude analysis that is easy to present. Rest assured that much tighter analyses are possible and much better PSRGs have been constructed since.

31.2 Why Study PSRGs?

Pseudo-random number generators take a seed which is presumably random (or which has a lot of randomness in it), and then generate a long string of random bits that are supposed to act random. We should first discuss why we would actually want such a thing. I can think of two reasons.

1. Random bits are scarce. This might be surprising. After all, if you look at the last few bits of the time that I last hit a key, it is pretty random. Similarly, the low-order bits of the temperature of the processor in my computer seem pretty random. While these bits are pretty random, there are not too many of them.

   Many randomized algorithms need *a lot* of random bits. Sources such as these just do not produce random bits with a frequency sufficient for many applications.

2. If you want to re-run an algorithm, say to de-bug it, it is very convenient to be able to use the same set of random bits by re-running the PSRG with the same seed. If you use truly random bits, you can't do this.

You may also wonder how good the standard pseudo-random number generators are. The first answer is that the default ones, such as `rand` in C, are usually terrible. There are many applications, such as those in my thesis, for which these generators produce behavior that is very different from what one would expect from truly random bits (yes, this is personal). On the other hand, one can use cryptographic functions to create bits that will act random for most purposes, unless one can break the underlying cryptography [HILL99]. But, the resulting generators are usually much slower than the fastest pseudo-random generators. Fundamentally, it comes down to a time-versus-quality tradeoff. The longer you are willing to wait, the better the pseudo-random bits you can get.

31.3 Expander Graphs

In today's lecture we will require an infinite family of d-regular $1/10$-expander graphs. We require that d be a constant, that the graphs have 2^r vertices for all sufficiently large r, and that we can construct the neighbors of a vertex in time polynomial in r. That is, we need the graphs to have a simple explicit description. One can construct expanders families of this form using the techniques from last lecture. For today's purposes, the best expanders are the Ramanujan graphs produced by Margulis [Mar88] and Lubotzky, Phillips and Sarnak [LPS88]. Ramanujan graphs of degree $d = 400$ are $1/10$-expanders. See also the work of Alon, Bruck, Naor, Naor and Roth [ABN+92] for even more explicit constructions.

While the explicit Ramanujan graphs only exist in certain sizes, none of which do have exactly 2^r vertices, some of them have just a little more that 2^r vertices. It is possible to trim these to make them work, say by ignoring all steps in which the vertex does not correspond to r bits.

31.4 Today's Application : repeating an experiment

Imagine you are given a black box that takes r bits as input and then outputs either 0 or 1. Moreover, let's assume that the black box is very consistent: we know that it returns the same answer at least 99% of the time. If it almost always returns 0, we will call it a 0-box and if it almost always returns 1, we will call it a 1-box. Our job is to determine whether a given box is a 0 or 1 box. We assume that r is big, so we don't have time to test the box on all 2^r settings of r bits. Instead, we could pick r bits at random, and check what the box returns. If it says"1", then it is probably a 1-box. But, what if we want more than 99% confidence? We could check the box on many choices of r random bits, and report the majority value returned by the box.[1]. But, this seems to require a new set of random bits for each run. In this lecture, we will prove that 9 new bits per run suffice. Note that the result would be interesting for any constant other than 9.

[1] Check for yourself that running it twice doesn't help

Since we will not make any assumptions about the black box, we will use truly random bits the first time we test it. But, we will show that we only need 9 new random bits for each successive test. In particular, we will show that if we use our PSRG to generate bits for $t+1$ test, then the probability that majority answer is wrong decreases exponentially in t.

You are probably wondering why we would want to do such a thing. The reason is to increase the accuracy of randomized algorithms. There are many randomized algorithms that provide weak guarantees, such as being correct 99% or 51% of the time. To obtain accurate answers from such algorithms, we run them many times with fresh random bits. You can view such an algorithm has having two inputs: the problem to be solved and its random bits. The black box is the behavior of the algorithm when the problem to be solved is fixed, so it is just working on the random bits.

31.5 The Random Walk Generator

Let r be the number of bits that our black box takes as input. So, the space of random bits is $\{0,1\}^r$. Let $X \subset \{0,1\}^r$ be the settings of the random bits on which the box gives the minority answer, and let Y be the settings on which it gives the majority answer.

Our pseudo-random generator will use a random walk on a 1/10-expander graph whose vertex set is $\{0,1\}^r$. Recall that we can use $d = 400$. For the first input we feed to the black box, we will require r truly random bits. We treat these bits as a vertex of our graph. For each successive test, we choose a random neighbor of the present vertex, and feed the corresponding bits to the box. That is, we choose a random i between 1 and 400, and move to the ith neighbor of the present vertex. Note that we only need $\log_2 400 \approx 9$ random bits to choose the next vertex. So, we will only need 9 new bits to generate each input we feed to the box after the first.

31.6 Formalizing the problem

Assume that we are going to test the box $t+1$ times. Our pseudo-random generator will begin at a truly random vertex v, and then take t random steps. Recall that we defined X to be the set of vertices on which the box outputs the minority answer, and we assume that $|X| \leq 2^r/100$. If we report the majority of the outcomes of the $t+1$ outputs of the box, we will return the correct answer as long as the random walk is inside X less than half the time. To analyze this, let v_0 be the initial random vertex, and let $v_1, \dots, v_t$ be the vertices produced by the t steps of the random walk. Let $T = \{0, \dots, t\}$ be the time steps, and let $S = \{i : v_i \in X\}$. We will prove

$$\Pr\left[|S| > t/2\right] \leq \left(\frac{2}{\sqrt{5}}\right)^{t+1}.$$

To begin our analysis, recall that the initial distribution of our random walk is $\boldsymbol{p}_0 = \boldsymbol{1}/n$. Let $\boldsymbol{\chi}_X$ and $\boldsymbol{\chi}_Y$ be the characteristic vectors of X and Y, respectively, and let $\boldsymbol{D}_X = \mathtt{diag}(\boldsymbol{\chi}_X)$ and $\boldsymbol{D}_Y = \mathtt{diag}(\boldsymbol{\chi}_Y)$. Let

$$\boldsymbol{W} = \frac{1}{d}\boldsymbol{M} \tag{31.1}$$

be the transition matrix of the ordinary random walk on G. We are not using the lazy random walk: it would be silly to use the lazy random walk for this problem, as there is no benefit to re-running the experiment with the same random bits as before. Let $\omega_1, \dots, \omega_n$ be the eigenvalues of $\boldsymbol{W}$. As the graph is a 1/10-expander, $|\omega_i| \leq 1/10$ for all $i \geq 2$.

Let's see how we can use these matrices to understand the probabilities under consideration. For a probability vector $\boldsymbol{p}$ on vertices, the probability that a vertex chosen according to $\boldsymbol{p}$ is in X may be expressed

$$\boldsymbol{\chi}_X^T \boldsymbol{p} = \mathbf{1}^T \boldsymbol{D}_X \boldsymbol{p}.$$

The second form will be more useful, as

$$\boldsymbol{D}_X \boldsymbol{p}$$

is the vector obtained by zeroing out the events in which the vertices are not in X. If we then want to take a step in the graph G, we multiply by $\boldsymbol{W}$. That is, the probability that the walk starts at vertex in X, and then goes to a vertex i is $\boldsymbol{q}(i)$ where

$$\boldsymbol{q} = \boldsymbol{W} \boldsymbol{D}_X \boldsymbol{p}_0.$$

Continuing this way, we see that the probability that the walk is in X at precisely the times $i \in R$ is

$$\mathbf{1}^T \boldsymbol{D}_{Z_t} \boldsymbol{W} \boldsymbol{D}_{Z_{t-1}} \boldsymbol{W} \cdots \boldsymbol{D}_{Z_1} \boldsymbol{W} \boldsymbol{D}_{Z_0} \boldsymbol{p}_0,$$

where

$$Z_i = \begin{cases} X & \text{if } i \in R \\ Y & \text{otherwise.} \end{cases}$$

We will prove that this probability is at most $(1/5)^{|R|}$. It will then follow that

$$\begin{aligned} \Pr\left[|S| > t/2\right] &\leq \sum_{|R| > t/2} \Pr\left[\text{the walk is in } X \text{ at precisely the times in } R\right] \\ &\leq \sum_{|R| > t/2} \left(\frac{1}{5}\right)^{|R|} \\ &\leq 2^{t+1} \left(\frac{1}{5}\right)^{(t+1)/2} \\ &= \left(\frac{2}{\sqrt{5}}\right)^{t+1}. \end{aligned}$$

31.7 Matrix Norms

Recall that the operator norm of a matrix M (also called the 2-norm) is defined by

$$\|\boldsymbol{M}\| = \max_{\boldsymbol{v}} \frac{\|\boldsymbol{M}\boldsymbol{v}\|}{\|\boldsymbol{v}\|}.$$

The matrix norm measures how much a vector can increase in size when it is multiplied by $\boldsymbol{M}$. When $\boldsymbol{M}$ is symmetric, the 2-norm is just the largest absolute value of an eigenvalue of $\boldsymbol{M}$ (prove this for yourself). It is also immediate that

$$\|\boldsymbol{M}_1\boldsymbol{M}_2\| \leq \|\boldsymbol{M}_1\| \, \|\boldsymbol{M}_2\| \, .$$

You should also verify this yourself. As $\boldsymbol{D}_X$, $\boldsymbol{D}_Y$ and $\boldsymbol{W}$ are symmetric, they each have norm 1.

Warning 31.7.1. *While the largest eigenvalue of a walk matrix is* 1*, the norm of an asymmetric walk matrix can be larger than* 1*. For instance, consider the walk matrix of the path on* 3 *vertices. Verify that it has norm* $\sqrt{2}$*.*

Our analysis rests upon the following bound on the norm of $\boldsymbol{D}_X W$.

Lemma 31.7.2.

$$\|\boldsymbol{D}_X \boldsymbol{W}\| \leq 1/5.$$

Let's see why this implies the theorem. For any set R, let Z_i be as defined above. As $\boldsymbol{p}_0 = \boldsymbol{W}\boldsymbol{p}_0$, we have

$$\boldsymbol{1}^T \boldsymbol{D}_{Z_t} \boldsymbol{W} \boldsymbol{D}_{Z_{t-1}} \boldsymbol{W} \cdots \boldsymbol{D}_{Z_1} \boldsymbol{W} \boldsymbol{D}_{Z_0} \boldsymbol{p}_0 = \boldsymbol{1}^T \left(\boldsymbol{D}_{Z_t} \boldsymbol{W}\right) \left(\boldsymbol{D}_{Z_{t-1}} \boldsymbol{W}\right) \cdots \left(\boldsymbol{D}_{Z_0} \boldsymbol{W}\right) \boldsymbol{p}_0.$$

Now,

$$\left\|\boldsymbol{D}_{Z_{t-1}} \boldsymbol{W}\right\| \leq \begin{cases} 1/5 & \text{for } i \in R, \text{ and} \\ 1 & \text{for } i \not\in R. \end{cases}$$

So,

$$\left\|\left(\boldsymbol{D}_{Z_t} \boldsymbol{W}\right) \left(\boldsymbol{D}_{Z_{t-1}} \boldsymbol{W}\right) \cdots \left(\boldsymbol{D}_{Z_0} \boldsymbol{W}\right)\right\| \leq (1/5)^{|R|}.$$

As $\|\boldsymbol{p}_0\| = 1/\sqrt{n}$ and $\|\boldsymbol{1}\| = \sqrt{n}$, we may conclude

$$\begin{aligned} \boldsymbol{1}^T \left(\boldsymbol{D}_{Z_t} \boldsymbol{W}\right) \left(\boldsymbol{D}_{Z_{t-1}} \boldsymbol{W}\right) \cdots \left(\boldsymbol{D}_{Z_0} \boldsymbol{W}\right) \boldsymbol{p}_0 &\leq \left\|\boldsymbol{1}^T\right\| \left\|\left(\boldsymbol{D}_{Z_t} \boldsymbol{W}\right) \left(\boldsymbol{D}_{Z_{t-1}} \boldsymbol{W}\right) \cdots \left(\boldsymbol{D}_{Z_0} \boldsymbol{W}\right) \boldsymbol{p}_0\right\| \\ &\leq \left\|\boldsymbol{1}^T\right\| (1/5)^{|R|} \left\|\boldsymbol{p}_0\right\| \\ &= (1/5)^{|R|}. \end{aligned}$$

31.8 The norm of $\boldsymbol{D}_X \boldsymbol{W}$

Proof of Lemma 31.7.2. Let $\boldsymbol{x}$ be any non-zero vector, and write

$$\boldsymbol{x} = c\boldsymbol{1} + \boldsymbol{y},$$

where $\boldsymbol{1}^T \boldsymbol{y} = 0$. We will show that $\|\boldsymbol{D}_X \boldsymbol{W} \boldsymbol{x}\| \leq \|\boldsymbol{x}\|/5$.

We know that the constant vectors are eigenvectors of $\boldsymbol{W}$. So, $\boldsymbol{W}\boldsymbol{1} = \boldsymbol{1}$ and

$$\boldsymbol{D}_X \boldsymbol{W} \boldsymbol{1} = \boldsymbol{\chi}_X.$$

This implies

$$\|\boldsymbol{D}_X \boldsymbol{W} c\boldsymbol{1}\| = c\,\|\boldsymbol{\chi}_X\| = c\sqrt{|X|} \leq c\sqrt{n}/10.$$

We will now show that $\|\boldsymbol{W}\boldsymbol{y}\| \leq \|\boldsymbol{y}\|/10$. The easiest way to see this is to consider the matrix

$$\boldsymbol{W} - \boldsymbol{J}/n,$$

where we recall that $\boldsymbol{J}$ is the all-1 matrix. This matrix is symmetric and all of its eigenvalues have absolute value at most 1/10. So, it has norm at most 1/10. Moreover, $(\boldsymbol{W} - \boldsymbol{J}/n)\boldsymbol{y} = \boldsymbol{W}\boldsymbol{y}$, which implies $\|\boldsymbol{W}\boldsymbol{y}\| \leq \|\boldsymbol{y}\|/10$. Another way to prove this is to expand $\boldsymbol{y}$ in the eigenbasis of $\boldsymbol{W}$, as in the proof of Lemma 2.1.3.

Finally, as $\boldsymbol{1}$ is orthogonal to $\boldsymbol{y}$,

$$\|\boldsymbol{x}\| = \sqrt{c^2 n + \|\boldsymbol{y}\|^2}.$$

So,

$$\|\boldsymbol{D}_X \boldsymbol{W}\boldsymbol{x}\| \leq \|\boldsymbol{D}_X \boldsymbol{W} c\boldsymbol{1}\| + \|\boldsymbol{D}_X \boldsymbol{W}\boldsymbol{y}\| \leq c\sqrt{n}/10 + \|\boldsymbol{y}\|/10 \leq \|\boldsymbol{x}\|/10 + \|\boldsymbol{x}\|/10 \leq \|\boldsymbol{x}\|/5.$$

□

31.9 Conclusion

Observe that this is a very strange proof. When considering probabilities, it seems that it would be much more natural to sum them. But, here we consider 2-norms of probability vectors.

31.10 Notes

For the best results on the number of bits one needs for each run of an algorithm, see [?].

For tighter results on the concentration on variables drawn from random walks on expanders, see Gillman [Gil98]. For matrices, see [GLSS18].

Part VI

Algorithms

Chapter 32

Sparsification by Random Sampling

32.1 Overview

Two weeks ago, we learned that expander graphs are sparse approximations of the complete graph. This week we will learn that every graph can be approximated by a sparse graph. Today, we will see how a sparse approximation can be obtained by careful random sampling: every graph on n vertices has an ϵ-approximation with only $O(\epsilon^{-2} n \log n)$ edges (a result of myself and Srivastava [SS11]). We will prove this using a matrix Chernoff bound due to Tropp [Tro12].

We originally proved this theorem using a concentration bound of Rudelson [Rud99]. This required an argument that used sampling with replacement. When I taught this result in 2012, I asked if one could avoid sampling with replacement. Nick Harvey pointed out to me the argument that avoids replacement that I am presenting today.

In the next lecture, we will see that the $\log n$ term is unnecessary. In fact, almost every graph can be approximated by a sparse graph almost as well as the Ramanujan graphs approximate complete graphs.

32.2 Sparsification

For this lecture, I define a graph H to be an ϵ-approximation of a graph G if

$$(1-\epsilon)\boldsymbol{L}_G \preccurlyeq \boldsymbol{L}_H \preccurlyeq (1+\epsilon)\boldsymbol{L}_G.$$

We will show that every graph G has a good approximation by a sparse graph. This is a very strong statement, as graphs that approximate each other have a lot in common. For example,

1. the effective resistance between all pairs of vertices are similar in the two graphs,
2. the eigenvalues of the graphs are similar,
3. the boundaries of all sets are similar, as these are given by $\boldsymbol{\chi}_S^T \boldsymbol{L}_G \boldsymbol{\chi}_S$, and

4. the solutions of linear equations in the two matrices are similar.

We will prove this by using a very simple random construction. We first carefully[1] choose a probability $p_{a,b}$ for each edge (a, b). We then include each edge (a, b) with probabilty $p_{a,b}$, independently. If we do include edge (a, b), we give it weight $w_{a,b}/p_{a,b}$. We will show that our choice of probabilities ensures that the resulting graph H has at most $4n \ln n/\epsilon^2$ edges and is an ϵ approximation of G with high probability.

The reason we employ this sort of sampling–blowing up the weight of an edge by dividing by the probability that we choose it—is that it preserves the matrix in expectation. Let $\boldsymbol{L}_{a,b}$ denote the elementary Laplacian on edge (a, b) with weight 1, so that

$$\boldsymbol{L}_G = \sum_{(a,b)\in E} w_{a,b} \boldsymbol{L}_{a,b}.$$

We then have that

$$\mathbb{E}\boldsymbol{L}_H = \sum_{(a,b)\in E} p_{a,b}(w_{a,b}/p_{a,b})\boldsymbol{L}_{a,b} = \boldsymbol{L}_G.$$

32.3 Matrix Chernoff Bounds

The main tool that we will use in our analysis is a theorem about the concentration of random matrices. These may be viewed as matrix analogs of the Chernoff bound that we saw in Lecture 5. These are a surprisingly recent development, with the first ones appearing in the work of Rudelson and Vershynin [Rud99, RV07] and Ahlswede and Winter [AW02]. The best present source for these bounds is Tropp [Tro12], in which the following result appears as Corollary 5.2.

Theorem 32.3.1. *Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m$ be independent random n-dimensional symmetric positive semidefinite matrices so that $\|\boldsymbol{X}_i\| \leq R$ almost surely. Let $\boldsymbol{X} = \sum_i \boldsymbol{X}_i$ and let μ_{min} and μ_{max} be the minimum and maximum eigenvalues of*

$$\mathbb{E}[\boldsymbol{X}] = \sum_i \mathbb{E}[\boldsymbol{X}_i].$$

Then,

$$Pr\left[\lambda_{min}(\sum_i \boldsymbol{X}_i) \leq (1-\epsilon)\mu_{min}\right] \leq n\left(\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right)^{\mu_{min}/R}, \qquad \text{for } 0 < \epsilon < 1, \text{ and}$$

$$Pr\left[\lambda_{max}(\sum_i \boldsymbol{X}_i) \geq (1+\epsilon)\mu_{max}\right] \leq n\left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right)^{\mu_{max}/R}, \qquad \text{for } 0 < \epsilon.$$

It is important to note that the matrices $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_m$ can have different distributions. Also note that as the norms of these matrices get bigger, the bounds above become weaker. As the

[1] For those who can't stand the suspense, we reveal that we will choose the probabilities to be proportional to leverage scores of the edges.

expressions above are not particularly easy to work with, we often use the following approximations.

$$\begin{aligned} \left(\frac{e^{-\epsilon}}{(1-\epsilon)^{1-\epsilon}}\right) &\leq e^{-\epsilon^2/2}, && \text{for } 0 < \epsilon < 1, \text{ and} \\ \left(\frac{e^{\epsilon}}{(1+\epsilon)^{1+\epsilon}}\right) &\leq e^{-\epsilon^2/3}, && \text{for } 0 < \epsilon < 1. \end{aligned}$$

Chernoff (and Hoeffding and Bernstein) bounds rarely come in exactly the form you want. Sometimes you can massage them into the needed form. Sometimes you need to prove your own. For this reason, you may some day want to spend a lot of time reading how these are proved.

32.4 The key transformation

Before applying the matrix Chernoff bound, we make a transformation that will cause $\mu_{min} = \mu_{\max} = 1$.

For positive definite matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we have

$$\boldsymbol{A} \preccurlyeq (1+\epsilon)\boldsymbol{B} \quad \iff \quad \boldsymbol{B}^{-1/2}\boldsymbol{A}\boldsymbol{B}^{-1/2} \preccurlyeq (1+\epsilon)\boldsymbol{I}.$$

The same things holds for singular semidefinte matrices that have the same nullspace:

$$\boldsymbol{L}_H \preccurlyeq (1+\epsilon)\boldsymbol{L}_G \quad \iff \quad \boldsymbol{L}_G^{+/2}\boldsymbol{L}_H\boldsymbol{L}_G^{+/2} \preccurlyeq (1+\epsilon)\boldsymbol{L}_G^{+/2}\boldsymbol{L}_G\boldsymbol{L}_G^{+/2},$$

where $\boldsymbol{L}_G^{+/2}$ is the square root of the pseudo-inverse of $\boldsymbol{L}_G$. Let

$$\boldsymbol{\Pi} = \boldsymbol{L}_G^{+/2}\boldsymbol{L}_G\boldsymbol{L}_G^{+/2},$$

which is the projection onto the range of $\boldsymbol{L}_G$. We now know that $\boldsymbol{L}_G$ is an ϵ-approximation of $\boldsymbol{L}_H$ if and only if $\boldsymbol{L}_G^{+/2}\boldsymbol{L}_H\boldsymbol{L}_G^{+/2}$ is an ϵ-approximation of $\boldsymbol{\Pi}$.

As multiplication by a fixed matrix is a linear operation and expectation commutes with linear operations,

$$\mathbb{E}\boldsymbol{L}_G^{+/2}\boldsymbol{L}_H\boldsymbol{L}_G^{+/2} = \boldsymbol{L}_G^{+/2}\left(\mathbb{E}\boldsymbol{L}_H\right)\boldsymbol{L}_G^{+/2} = \mathbb{E}\boldsymbol{L}_G^{+/2}\boldsymbol{L}_G\boldsymbol{L}_G^{+/2} = \boldsymbol{\Pi}.$$

So, we really just need to show that this random matrix is probably close to its expectation, $\boldsymbol{\Pi}$. It would probably help to pretend that $\boldsymbol{\Pi}$ is in fact the identity, as it will make it easier to understand the analysis. In fact, you don't have to pretend: you could project all the vectors and matrices onto the span of $\boldsymbol{\Pi}$ and carry out the analysis there.

32.5 The probabilities

Let

$$\boldsymbol{X}_{a,b} = \begin{cases} (w_{a,b}/p_{a,b})\boldsymbol{L}_G^{+/2}\boldsymbol{L}_{(a,b)}\boldsymbol{L}_G^{+/2} & \text{with probability } p_{a,b} \\ 0 & \text{otherwise,} \end{cases}$$

so that

$$\boldsymbol{L}_G^{+/2} \boldsymbol{L}_H \boldsymbol{L}_G^{+/2} = \sum_{(a,b) \in E} \boldsymbol{X}_{a,b}.$$

We will choose the probabilities to be

$$p_{a,b} \overset{\text{def}}{=} \frac{1}{R} w_{a,b} \left\| \boldsymbol{L}_G^{+/2} \boldsymbol{L}_{(a,b)} \boldsymbol{L}_G^{+/2} \right\|,$$

for an R to be chosen later. Thus, when edge (a, b) is chosen, $\|X_{a,b}\| = R$. Making this value uniform for every edge optimizes one part of Theorem 32.3.1.

You may wonder what we should do if one of these probabilities $p_{a,b}$ exceeds one. There are many ways of addressing this issue. For now, pretend that it does not happen. We will then explain how to deal with this at the end of lecture.

Recall that the leverage score of edge (a, b) written $\ell_{a,b}$ was defined in Lecture 14 to be the weight of an edge times the effective resistance between its endpoints:

$$\ell_{a,b} = w_{a,b} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}_G^{+} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b).$$

To see the relation between the leverage score and $p_{a,b}$, compute

$$\begin{aligned} \left\| \boldsymbol{L}_G^{+/2} \boldsymbol{L}_{(a,b)} \boldsymbol{L}_G^{+/2} \right\| &= \left\| \boldsymbol{L}_G^{+/2} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}_G^{+/2} \right\| \\ &= \left\| (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}_G^{+/2} \boldsymbol{L}_G^{+/2} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b) \right\| \\ &= (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}_G^{+} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b) \\ &= \mathrm{R}_{\mathrm{eff}}(a, b). \end{aligned}$$

As we can quickly approximate the effective resistance of every edge, we can quickly compute sufficient probabilities.

Recall that the leverage score of an edge equals the probability that the edge appears in a random spanning tree. As every spanning tree has $n - 1$ edges, this means that the sum of the leverage scores is $n - 1$, and thus

$$\sum_{(a,b) \in E} p_{a,b} = \frac{n-1}{R} \leq \frac{n}{R}.$$

This is a very clean bound on the expected number of edges in H. One can use a Chernoff bound (on real variables rather than matrices) to prove that it is exponentially unlikely that the number of edges in H is more than any small multiple of this.

For your convenience, I recall another proof that the sum of the leverage scores is $n-1$:

$$\begin{aligned}
\sum_{(a,b)\in E} \ell_{a,b} &= \sum_{(a,b)\in E} w_{a,b}\mathrm{R_{eff}}(a,b) \\
&= \sum_{(a,b)\in E} w_{a,b}(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T \boldsymbol{L}_G^+ (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b) \\
&= \sum_{(a,b)\in E} w_{a,b}\mathrm{Tr}\left(\boldsymbol{L}_G^+ (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T\right) \\
&= \mathrm{Tr}\left(\sum_{(a,b)\in E} \boldsymbol{L}_G^+ w_{a,b}(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)(\boldsymbol{\delta}_a - \boldsymbol{\delta}_b)^T\right) \\
&= \mathrm{Tr}\left(\boldsymbol{L}_G^+ \sum_{(a,b)\in E} w_{a,b}\boldsymbol{L}_{a,b}\right) \\
&= \mathrm{Tr}\left(\boldsymbol{L}_G^+ \boldsymbol{L}_G\right) \\
&= \mathrm{Tr}\left(\boldsymbol{\Pi}\right) \\
&= n-1.
\end{aligned}$$

32.6 The analysis

We will choose

$$R = \frac{\epsilon^2}{3.5\ln n}.$$

Thus, the number of edges in H will be at most $4(\ln n)\epsilon^{-2}$ with high probability.

We have

$$\sum_{(a,b)\in E} \mathbb{E}\boldsymbol{X}_{a,b} = \boldsymbol{\Pi}.$$

It remains to show that it is unlikely to deviate from this by too much.

We first consider the case in which $p_{(a,b)} \leq 1$ for all edges (a,b). If this is the case, then Theorem 32.3.1 tells us that

$$\Pr\left[\sum_{a,b} \boldsymbol{X}_{a,b} \geq (1+\epsilon)\boldsymbol{\Pi}\right] \leq n\exp\left(-\epsilon^2/3R\right) = n\exp\left(-(3.5/3)\ln n\right) = n^{-1/6}.$$

For the lower bound, we need to remember that we can just work orthogonal to the all-1s vector, and so treat the smallest eigenvalue of $\boldsymbol{\Pi}$ as 1. We then find that

$$\Pr\left[\sum_{a,b} \boldsymbol{X}_{a,b} \leq (1-\epsilon)\boldsymbol{\Pi}\right] \leq n\exp\left(-\epsilon^2/2R\right) = n\exp\left(-(3.5/2)\ln n\right) = n^{-3/2},$$

We finally return to deal with the fact that there might be some edges for which $p_{a,b} \geq 1$ and so definitely appear in H. There are two natural ways to deal with these—one that is easiest algorithmically and one that simplifies the proof. The algorithmically natural way to handle these is to simply include these edges in H, and remove them from the analysis above. This requires a small adjustment to the application of the Matrix Chernoff bound, but it does go through.

From the perspective of the proof, the simplest way to deal with these is to split each such $\boldsymbol{X}_{a,b}$ into many independent random edges: $k = \lfloor \ell_{a,b}/R \rfloor$ that appear with probability exactly 1, and one more that appears with probability $\ell_{a,b}/R - k$. This does not change the expectation of their sum, or the expected number of edges once we remember to add together the weights of edges that appear multiple times. The rest of the proof remains unchanged.

32.7 Open Problem

If I have time in class, I will sketch a way to quickly approximate the effective resistances of every edge in the graph. The basic idea, which can be found in [SS11] and which is carried out better in [KLP12], is that we can compute the effective resistance of an edge (a, b) from the solution to a logarithmic number of systems of random linear equations in $\boldsymbol{L}_G$. That is, after solving a logarithmic number of systems of linear equations in $\boldsymbol{L}_G$, we have information from which we can estimates all of the effective resistances.

In order to sparsify graphs, we do not actually need estimates of effective resistances that are always accurate. We just need a way to identify many edges of low effective resistance, without listing any that have high effective resistance. I believe that better algorithms for doing this remain to be found. Current fast algorithms that make progress in this direction and that exploit such estimates may be found in [KLP12, Kou14, CLM$^+$15, LPS15]. These, however, rely on fast Laplacian equation solvers. It would be nice to be able to estimate effective resistances without these. A step in this direction was recently taken in the works [CGP$^+$18, LSY18], which quickly decompose graphs into the union of short cycles plus a few edges.

Chapter 33

Linear Sized Sparsifiers

33.1 Overview

In this lecture, we will prove a slight simplification of the main result of [BSS12, BSS14]. This will tell us that every graph with n vertices has an ϵ-approximation with approximately $4\epsilon^{-2}n$ edges. To translate this into a relation between approximation quality and average degree, note that such a graph has average degree $d_{ave} = 8\epsilon^{-2}$. So,

$$\epsilon \approx \frac{2\sqrt{2}}{\sqrt{d}},$$

which is about twice what you would get from a Ramanujan graph. Interestingly, this result even works for average degree just a little bit more than 1.

33.2 Turning edges into vectors

In the last lecture, we considered the Laplacian matrix of a graph G times the square root of the pseudoinverse on either side. That is,

$$\boldsymbol{L}_G^{+/2} \left(\sum_{(a,b) \in E} w_{a,b} \boldsymbol{L}_{(a,b)} \right) \boldsymbol{L}_G^{+/2}.$$

Today, it will be convenient to view this as a sum of outer products of vectors. Set

$$\boldsymbol{v}_{(a,b)} = \sqrt{w_{a,b}} \boldsymbol{L}_G^{+/2} (\boldsymbol{\delta}_a - \boldsymbol{\delta}_b).$$

Then,

$$\boldsymbol{L}_G^{+/2} \left(\sum_{(a,b) \in E} w_{a,b} \boldsymbol{L}_{(a,b)} \right) \boldsymbol{L}_G^{+/2} = \sum_{(a,b) \in E} \boldsymbol{v}_{(a,b)} \boldsymbol{v}_{(a,b)}^T = \boldsymbol{\Pi},$$

where we recall that $\boldsymbol{\Pi} = \frac{1}{n}\boldsymbol{L}_{K_n}$ is the projection orthogonal to the constant vectors.

The problem of sparsification is then the problem of finding a small subset of these vectors, $S \subseteq E$, along with scaling factors, $c : S \rightarrow \mathbb{R}$, so that

$$(1-\epsilon)\boldsymbol{\Pi} \preccurlyeq \sum_{(a,b)\in S} c_{a,b}\boldsymbol{v}_{(a,b)}\boldsymbol{v}_{(a,b)}^T \preccurlyeq (1+\epsilon)\boldsymbol{\Pi}$$

If we project onto the span of the Laplacian, then the sum of the outer products of vectors $\boldsymbol{v}_{(a,b)}$ becomes the identity, and our goal is to find a set S and scaling factors $c_{a,b}$ so that

$$(1-\epsilon)\boldsymbol{I}_{n-1} \preccurlyeq \sum_{(a,b)\in S} c_{a,b}\boldsymbol{v}_{(a,b)}\boldsymbol{v}_{(a,b)}^T \preccurlyeq (1+\epsilon)\boldsymbol{I}_{n-1}.$$

That is, so that all the eigenvalues of the matrix in the middle lie between $(1-\epsilon)$ and $(1+\epsilon)$.

33.3 The main theorem

Theorem 33.3.1. *Let $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ be vectors in $\mathbb{R}^n$ so that*

$$\sum_i \boldsymbol{v}_i\boldsymbol{v}_i^T = \boldsymbol{I}.$$

Then, for every $\epsilon > 0$ there exists a set S along with scaling factors c_i so that

$$(1-\epsilon)^2\boldsymbol{I} \preccurlyeq \sum_{i\in S} c_i\boldsymbol{v}_i\boldsymbol{v}_i^T \preccurlyeq (1+\epsilon)^2\boldsymbol{I},$$

and

$$|S| \leq \left\lceil n/\epsilon^2 \right\rceil.$$

The condition that the sum of the outer products of the vectors sums to the identity has a name, *isotropic position.* I now mention one important property of vectors in isotropic position

Lemma 33.3.2. *Let $\boldsymbol{v}_1, \dots, \boldsymbol{v}_m$ be vectors in isotropic position. Then, for every matrix $\boldsymbol{M}$,*

$$\sum_i \boldsymbol{v}_i^T\boldsymbol{M}\boldsymbol{v}_i = \operatorname{Tr}(\boldsymbol{M}).$$

Proof. We have

$$\boldsymbol{v}^T\boldsymbol{M}\boldsymbol{v} = \operatorname{Tr}\left(\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{M}\right),$$

so

$$\sum_i \boldsymbol{v}_i^T\boldsymbol{M}\boldsymbol{v}_i = \sum_i \operatorname{Tr}\left(\boldsymbol{v}_i\boldsymbol{v}_i^T\boldsymbol{M}\right) = \operatorname{Tr}\left(\left(\sum_i \boldsymbol{v}_i\boldsymbol{v}_i^T\right)\boldsymbol{M}\right) = \operatorname{Tr}(\boldsymbol{I}\boldsymbol{M}) = \operatorname{Tr}(\boldsymbol{M}).$$

□

Today, we will prove that we can find a set of $6n$ vectors for which all eigenvalues lie between $1n$ and $13n$. If you divide all scaling factors by $\sqrt{13}n$, this puts the eigenvalues between $1/\sqrt{13}$ and $\sqrt{13}$. You can tighten the argument to prove Theorem 33.3.1.

We will prove this theorem by an iterative argument in which we choose one vector at a time to add to the set S. We will set the scaling factor of a vector when we add it to S. It is possible that we will add a vector to S more than once, in which case we will increase its scaling factor each time. Throughout the argument we will maintain the invariant that the eigenvalues of the scaled sum of outer produces is in the interval $[l, u]$, where l and u are quantities that will change with each addition to S. At the start of the algorithm, when S is empty, we will have

$$l_0 = -n \quad \text{and} \quad u_0 = n.$$

Every time we add a vector to S, we increase l by δ_L and u by δ_U, where

$$\delta_L = 1/3 \quad \text{and} \quad \delta_U = 2.$$

After we have done this $6n$ times, we will have $l = n$ and $u = 13n$.

33.4 Rank-1 updates

We will need to understand what happens to a matrix when we add the outer product of a vector.

Theorem 33.4.1 (Sherman-Morrison)**.** *Let $\boldsymbol{A}$ be a nonsingular symmetric matrix and let $\boldsymbol{v}$ be a vector and let c be a real number. Then,*

$$(\boldsymbol{A} - c\boldsymbol{v}\boldsymbol{v}^T)^{-1} = \boldsymbol{A}^{-1} + c\frac{\boldsymbol{A}^{-1}\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 - c\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}}.$$

Proof. The easiest way to prove this is to multiply it out, gathering $\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}$ terms into scalars:

$$\begin{aligned}(\boldsymbol{A} - c\boldsymbol{v}\boldsymbol{v}^T)\left(\boldsymbol{A}^{-1} + c\frac{\boldsymbol{A}^{-1}\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 - c\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}}\right) &= \boldsymbol{I} - c\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{A}^{-1} + c\frac{\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 - c\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}} - c^2\frac{\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{A}^{-1}}{1 - c\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}} \\ &= \boldsymbol{I} - c\boldsymbol{v}\boldsymbol{v}^T\boldsymbol{A}^{-1}\left(1 - \frac{1}{1 - c\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}} + \frac{c\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v}}{1 - c\boldsymbol{v}^T\boldsymbol{A}^{-1}\boldsymbol{v}}\right) \\ &= \boldsymbol{I}.\end{aligned}$$

□

33.5 Barrier Function Arguments

To prove the main theorem we need a good way to measure progress. We would like to keep all the eigenvalues of the matrix we have constructed at any point to lie in a nice range. But, more than that, we need them to be nicely distributed within this range. To enforce this, we need to measure how close the eigenvalues are to the limits.

Let $\boldsymbol{A}$ be a symmetric matrix with eigenvalues $\lambda_1 \leq \ldots \leq \lambda_n$. If u is larger than all of the eigenvalues of $\boldsymbol{A}$, then we call u an upper bound on $\boldsymbol{A}$. To make this notion quantitive, we define the *upper barrier function*

$$\Phi^u(\boldsymbol{A}) = \sum_i \frac{1}{u - \lambda_i}.$$

This is positive for all upper bounds u, goes to infinity as u approaches the largest eigenvalue, decreases as u grows, and is convex for $u > \lambda_n$. In particular, we will use

$$\Phi^{u+\delta}(\boldsymbol{A}) < \Phi^u(\boldsymbol{A}), \qquad \text{for } \delta > 0. \tag{33.1}$$

Also, observe that

$$\lambda_n \leq u - 1/\Phi^u(\boldsymbol{A}). \tag{33.2}$$

We will exploit the following formula for the upper barrier function:

$$\Phi^u(\boldsymbol{A}) = \operatorname{Tr}\left((u\boldsymbol{I} - \boldsymbol{A})^{-1}\right).$$

For a lower bound on the eigenvalues l, we will define an analogous lower barrier function

$$\Phi_l(\boldsymbol{A}) = \sum_i \frac{1}{\lambda_i - l} = \operatorname{Tr}\left((\boldsymbol{A} - l\boldsymbol{I})^{-1}\right).$$

This is positive whenever l is smaller than all the eigenvalues, goes to infinity as l approaches the smallest eigenvalue, and decreases as l becomes smaller. In particular,

$$l + 1/\Phi_l(\boldsymbol{A}) \leq \lambda_1. \tag{33.3}$$

The analog of (33.1) is the following.

Claim 33.5.1. *Let l be a lower bound on $\boldsymbol{A}$ and let $\delta < 1/\Phi_l(\boldsymbol{A})$. Then,*

$$\Phi_{l+\delta}(\boldsymbol{A}) \leq \frac{1}{1/\Phi_l(\boldsymbol{A}) - \delta}.$$

Note that this inequality is an equality when A is one-dimensional. In that case,

$$\frac{1}{\lambda_1 - l - \delta} = \frac{1}{1/(1/\lambda_1 - l) - \delta}.$$

Proof. After rearranging terms, we see that the inequality is equivalent to

$$\Phi_{l+\delta}(\boldsymbol{A}) - \Phi_l(\boldsymbol{A}) \leq \delta \Phi_{l+\delta}(\boldsymbol{A})\Phi_l(\boldsymbol{A}).$$

We then prove this by expanding in the eigenvalues, keeping in mind that all the terms $\lambda_i - l - \delta$ are positive:

$$\begin{aligned}
\Phi_{l+\delta}(\boldsymbol{A}) - \Phi_l(\boldsymbol{A}) &= \sum_i \frac{1}{\lambda_i - l - \delta} - \sum_i \frac{1}{\lambda_i - l} \\
&= \sum_i \frac{\delta}{(\lambda_i - l - \delta)(\lambda_i - l)} \\
&\leq \delta \left(\sum_i \frac{1}{(\lambda_i - l - \delta)}\right)\left(\sum_i \frac{1}{(\lambda_i - l)}\right).
\end{aligned}$$

□

Initially, we will have

$$\Phi_{l_0}(0) = \Phi_{-n}(0) = 1 \quad \text{and} \quad \Phi^{u_0}(0) = \Phi^{n}(0) = 1.$$

33.6 Barrier Function Updates

The most important thing to understand about the barrier functions is how they change when we add a vector to S. The Sherman-Morrison theorem tells us that happens when we change $\boldsymbol{A}$ to $\boldsymbol{A} + c\boldsymbol{v}\boldsymbol{v}^T$:

$$\begin{aligned}\Phi^u(\boldsymbol{A} + c\boldsymbol{v}\boldsymbol{v}^T) &= \operatorname{Tr}\left((u\boldsymbol{I} - \boldsymbol{A} - c\boldsymbol{v}\boldsymbol{v}^T)^{-1}\right) \\ &= \operatorname{Tr}\left((u\boldsymbol{I} - \boldsymbol{A})^{-1}\right) + c\frac{\operatorname{Tr}\left((u\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{v}\boldsymbol{v}^T(u\boldsymbol{I} - \boldsymbol{A})^{-1}\right)}{1 - c\boldsymbol{v}^T(u\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{v}} \\ &= \Phi^u(\boldsymbol{A}) + c\frac{\operatorname{Tr}\left(\boldsymbol{v}^T(u\boldsymbol{I} - \boldsymbol{A})^{-1}(u\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{v}\right)}{1 - c\boldsymbol{v}^T(u\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{v}} \\ &= \Phi^u(\boldsymbol{A}) + c\frac{\boldsymbol{v}^T(u\boldsymbol{I} - \boldsymbol{A})^{-2}\boldsymbol{v}}{1 - c\boldsymbol{v}^T(u\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{v}}.\end{aligned}$$

This increases the upper barrier function, and we would like to counteract this increase by increasing u at the same time. If we advance u to $\widehat{u} = u + \delta_U$, then we find

$$\begin{aligned}\Phi^{u+\delta_U}(\boldsymbol{A} + c\boldsymbol{v}\boldsymbol{v}^T) &= \Phi^{u+\delta_U}(\boldsymbol{A}) + c\frac{\boldsymbol{v}^T(\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-2}\boldsymbol{v}}{1 - c\boldsymbol{v}^T(\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{v}} \\ &= \Phi^u(\boldsymbol{A}) - \left(\Phi^u(\boldsymbol{A}) - \Phi^{u+\delta_U}(\boldsymbol{A})\right) + \frac{\boldsymbol{v}^T(\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-2}\boldsymbol{v}}{1/c - \boldsymbol{v}^T(\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{v}}.\end{aligned}$$

We would like for this to be less than $\Phi^u(\boldsymbol{A})$. If we commit to how much we are going to increase u, then this gives an upper bound on how large c can be. We want

$$\left(\Phi^u(\boldsymbol{A}) - \Phi^{u+\delta_U}(\boldsymbol{A})\right) \geq \frac{\boldsymbol{v}^T(\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-2}\boldsymbol{v}}{1/c - \boldsymbol{v}^T(\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{v}},$$

which is equivalent to

$$\frac{1}{c} \geq \frac{\boldsymbol{v}^T(\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-2}\boldsymbol{v}}{(\Phi^u(\boldsymbol{A}) - \Phi^{u+\delta_U}(\boldsymbol{A}))} + \boldsymbol{v}^T(\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-1}\boldsymbol{v}.$$

Define

$$\boldsymbol{U}_{\boldsymbol{A}} = \frac{((u + \delta_u)\boldsymbol{I} - \boldsymbol{A})^{-2}}{(\Phi^u(\boldsymbol{A}) - \Phi^{u+\delta_U}(\boldsymbol{A}))} + ((u + \delta_u)\boldsymbol{I} - \boldsymbol{A})^{-1}.$$

We have established a clean condition for when we can add $c\boldsymbol{v}\boldsymbol{v}^T$ to S and increase u by δ_U without increasing the upper barrier function.

Lemma 33.6.1. *If*

$$\frac{1}{c} \geq \boldsymbol{v}^T \boldsymbol{U}_{\boldsymbol{A}} \boldsymbol{v},$$

then

$$\Phi^{u+\delta_U}(\boldsymbol{A} + c\boldsymbol{v}\boldsymbol{v}^T) \leq \Phi^u(\boldsymbol{A}).$$

The miracle in the above formula is that the condition in the lemma just involves the vector $\boldsymbol{v}$ as the argument of a quadratic form.

We also require the following analog for the lower barrier function. The difference is that increasing l by setting $\hat{l} = l + \delta_L$ increases the barrier function, and adding a vector decreases it.

Lemma 33.6.2. *Define*

$$\boldsymbol{L}_{\boldsymbol{A}} = \frac{(\boldsymbol{A} - \hat{l}\boldsymbol{I})^{-2}}{(\Phi_{l+\delta_L}(\boldsymbol{A}) - \Phi_l(\boldsymbol{A}))} - (\boldsymbol{A} - \hat{l}\boldsymbol{I})^{-1}.$$

If

$$\frac{1}{c} \leq \boldsymbol{v}^T \boldsymbol{L}_{\boldsymbol{A}} \boldsymbol{v},$$

then

$$\Phi_{l+\delta_L}(\boldsymbol{A} + c\boldsymbol{v}\boldsymbol{v}^T) \leq \Phi_l(\boldsymbol{A}).$$

If we fix the vector $\boldsymbol{v}$ and an increment δ_L, then this gives a lower bound on the scaling factor by which we need to multiply it for the lower barrier function not to increase.

33.7 The inductive argument

It remains to show that there exits a vector $\boldsymbol{v}$ and a scaling factor c so that

$$\Phi^{u+\delta_U}(\boldsymbol{A} + c\boldsymbol{v}\boldsymbol{v}^T) \leq \Phi^u(\boldsymbol{A}) \quad \text{and} \quad \Phi_{l+\delta_L}(\boldsymbol{A} + c\boldsymbol{v}\boldsymbol{v}^T) \leq \Phi_l(\boldsymbol{A}).$$

That is, we need to show that there is a vector $\boldsymbol{v}_i$ so that

$$\boldsymbol{v}_i^T \boldsymbol{U}_{\boldsymbol{A}} \boldsymbol{v}_i \leq \boldsymbol{v}_i^T \boldsymbol{L}_{\boldsymbol{A}} \boldsymbol{v}_i.$$

Once we know this, we can set c so that

$$\boldsymbol{v}_i^T \boldsymbol{U}_{\boldsymbol{A}} \boldsymbol{v}_i \leq \frac{1}{c} \leq \boldsymbol{v}_i^T \boldsymbol{L}_{\boldsymbol{A}} \boldsymbol{v}_i.$$

Lemma 33.7.1.

$$\sum_i \boldsymbol{v}_i^T \boldsymbol{U}_{\boldsymbol{A}} \boldsymbol{v}_i \leq \frac{1}{\delta_U} + \Phi_u(\boldsymbol{A}).$$

Proof. By Lemma 33.3.2, we know

$$\sum_i \boldsymbol{v}_i^T \boldsymbol{U}_{\boldsymbol{A}} \boldsymbol{v}_i = \mathrm{Tr}\left(\boldsymbol{U}_{\boldsymbol{A}}\right).$$

To bound this, we break it into two parts

$$\frac{\mathrm{Tr}\left((\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-2}\right)}{(\Phi^u(\boldsymbol{A}) - \Phi^{u+\delta_U}(\boldsymbol{A}))}$$

and

$$\mathrm{Tr}\left((\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-1}\right).$$

The second term is easiest

$$\mathrm{Tr}\left((\widehat{u}\boldsymbol{I} - \boldsymbol{A})^{-1}\right) = \Phi^{u+\delta}(\boldsymbol{A}) \leq \Phi^u(\boldsymbol{A}).$$

To bound the first term, consider the derivative of the barrier function with respect to u:

$$\frac{\partial}{\partial u}\Phi^u(\boldsymbol{A}) = \frac{\partial}{\partial u}\sum_i \frac{1}{u - \lambda_i} = -\sum_i \left(\frac{1}{u - \lambda_i}\right)^2 = -\mathrm{Tr}\left(u\boldsymbol{I} - \boldsymbol{A}\right)^{-2}.$$

As $\Phi^u(\boldsymbol{A})$ is convex in u, we may conclude that

$$\Phi^u(\boldsymbol{A}) - \Phi^{u+\delta_U}(\boldsymbol{A}) \geq -\delta_U \frac{\partial}{\partial u}\Phi^{u+\delta_u}(\boldsymbol{A}) = \delta_U \mathrm{Tr}\left(\widehat{u}\boldsymbol{I} - \boldsymbol{A}\right)^{-2}.$$

□

The analysis for the lower barrier is similar, but the second term is slightly more complicated.

Lemma 33.7.2.

$$\sum_i \boldsymbol{v}_i^T \boldsymbol{L}_{\boldsymbol{A}} \boldsymbol{v}_i \geq \frac{1}{\delta_L} - \frac{1}{1/\Phi_l(\boldsymbol{A}) - \delta_L}.$$

Proof. As before, we bound

$$\frac{\mathrm{Tr}\left((\boldsymbol{A} - (l + \delta_L\boldsymbol{I}))^{-2}\right)}{\Phi_{l+\delta_L}(\boldsymbol{A}) - \Phi_l(\boldsymbol{A})}$$

by recalling that

$$\frac{\partial}{\partial l}\Phi_l(\boldsymbol{A}) = \mathrm{Tr}\left(\boldsymbol{A} - l\boldsymbol{I}\right)^{-2}.$$

As $\Phi_l(\boldsymbol{A})$ is convex in l, we have

$$\Phi_{l+\delta_L}(\boldsymbol{A}) - \Phi_l(\boldsymbol{A}) \leq \delta_L \frac{\partial}{\partial l}\Phi_{l+\delta_L}(\boldsymbol{A}) = \delta_L \mathrm{Tr}\left(\boldsymbol{A} - (l + \delta_L)\boldsymbol{I}\right)^{-2}.$$

To bound the other term, we use Claim 33.5.1 to prove

$$\mathrm{Tr}\left((\boldsymbol{A} - (l + \delta_L\boldsymbol{I})^{-1}\right) \leq \frac{1}{1/\Phi_l(\boldsymbol{A}) - \delta_L}.$$

□

So, for there to exist a $\boldsymbol{v}_i$ that we can add to S with scale factor c so that neither barrier function increases, we just need that

$$\frac{1}{\delta_U} + \Phi^u(\boldsymbol{A}) \leq \frac{1}{\delta_L} - \frac{1}{1/\Phi_l(\boldsymbol{A}) - \delta}.$$

If this holds, then there is a $\boldsymbol{v}_i$ so that

$$\boldsymbol{v}_i \boldsymbol{U}_{\boldsymbol{A}} \boldsymbol{v}_i \leq \boldsymbol{v}_i \boldsymbol{L}_{\boldsymbol{A}} \boldsymbol{v}_i.$$

We then set c so that

$$\boldsymbol{v}_i \boldsymbol{U}_{\boldsymbol{A}} \boldsymbol{v}_i \leq \frac{1}{c} \leq \boldsymbol{v}_i \boldsymbol{L}_{\boldsymbol{A}} \boldsymbol{v}_i.$$

We now finish the proof by checking that the numbers I gave earlier satisfy the necessary conditions. At the start both barrier functions are less than 1, and we need to show that this holds throughout the algorithm. At every step, we will have by induction

$$\frac{1}{\delta_U} + \Phi_u(\boldsymbol{A}) \leq \frac{1}{2} + 1 = \frac{3}{2},$$

and

$$\frac{1}{\delta_L} - \frac{1}{1/\Phi_l(\boldsymbol{A}) - \delta_L} \geq 3 - \frac{1}{1 - 1/3} = \frac{3}{2}.$$

So, there is always a $\boldsymbol{v}_i$ that we can add to S and a scaling factor c so that both barrier function remain upper bounded by 1.

If we now do this for $6n$ steps, we will have

$$l = -n + 6n/3 = n \qquad \text{and} \qquad u = n + 2 \cdot 6n = 13n.$$

The bound stated at the beginning of the lecture comes from tightening the analysis. In particular, it is possible to improve Lemma 33.7.2 so that it says

$$\sum_i \boldsymbol{v}_i^T \boldsymbol{L}_{\boldsymbol{A}} \boldsymbol{v}_i \geq \frac{1}{\delta_L} - \frac{1}{1/\Phi_l(\boldsymbol{A})}.$$

I recommend the paper for details.

33.8 Progress and Open Problems

- It is possible to generalize this result to sums of positive semidefinite matrices, instead of outer products of vectors [dCSHS11].
- It is now possible to compute sparsifiers that are almost this good in something close to linear time. [AZLO15, LS15].
- Given last lecture, it seems natural to conjecture that the scaling factors of edges should be proportional to their weights times effective resistances. Similarly, one might conjecture that if all vectors $\boldsymbol{v}_i$ have the same norm, then the scaling factors are unnecessary. This is true, but not obvious. In fact, it is essentially equivalent to the Kadison-Singer problem [MSS14, MSS15c].

Chapter 34

Iterative solvers for linear equations

We introduce basic iterative solvers for systems of linear equations: Richardson iteration and Chebyshev's method. We discuss Conjugate Gradient in the next Chapter, and iterative refinement and preconditioning in Chapter 36.

34.1 Why iterative methods?

One is first taught to solve linear systems like

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$$

by *direct methods* such as Gaussian elimination, computing the inverse of $\boldsymbol{A}$, or the LU factorization. However, elimination algorithms can be very slow. This is especially true when $\boldsymbol{A}$ is sparse. Just writing down the inverse takes $O(n^2)$ space, and computing the inverse takes $O(n^3)$ time if we do it naively. This might be OK if $\boldsymbol{A}$ is dense. But, it is very wasteful if $\boldsymbol{A}$ only has $O(n)$ non-zero entries.

In general, we prefer algorithms whose running time is proportional to the number of non-zero entries in the matrix $\boldsymbol{A}$, and which do not require much more space than that used to store $\boldsymbol{A}$.

Iterative algorithms solve linear equations *while only performing multiplications* by $\boldsymbol{A}$, and performing a few vector operations. Unlike the *direct methods* which are based on elimination, the iterative algorithms do not find exact solutions. Rather, they get closer and closer to the solution the longer they work. The advantage of these methods is that they need to store very little, and are often much faster than the direct methods. When $\boldsymbol{A}$ is symmetric, the running times of these methods are determined by the eigenvalues of $\boldsymbol{A}$.

Throughout this lecture we will assume that $\boldsymbol{A}$ is positive definite or positive semidefinite.

34.2 First-Order Richardson Iteration

To get started, we will examine a simple, but sub-optimal, iterative method, Richardson's iteration. The idea of the method is to find an iterative process that has the solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ as a fixed point, and which converges. We observe that if $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$, then for any α,

$$\begin{aligned} \alpha \boldsymbol{A}\boldsymbol{x} &= \alpha \boldsymbol{b}, \quad \Longrightarrow \\ \boldsymbol{x} + (\alpha \boldsymbol{A} - I)\boldsymbol{x} &= \alpha \boldsymbol{b}, \quad \Longrightarrow \\ \boldsymbol{x} &= (I - \alpha \boldsymbol{A})\boldsymbol{x} + \alpha \boldsymbol{b}. \end{aligned}$$

This leads us to the following iterative process:

$$\boldsymbol{x}^t = (I - \alpha \boldsymbol{A})\boldsymbol{x}^{t-1} + \alpha \boldsymbol{b}, \tag{34.1}$$

where we will take $\boldsymbol{x}^0 = \boldsymbol{0}$. We will show that this converges if

$$I - \alpha \boldsymbol{A}$$

has norm less than 1, and that the convergence rate depends on how much the norm is less than 1. This is analogous to our analysis of random walks on graphs from Chapter 10.

As we are assuming $\boldsymbol{A}$ is symmetric, $I - \alpha \boldsymbol{A}$ is symmetric as well, and so its norm is the maximum absolute value of its eigenvalues. Let $0 < \lambda_1 \leq \lambda_2 \ldots \leq \lambda_n$ be the eigenvalues of $\boldsymbol{A}$. Then, the eigenvalues of $I - \alpha \boldsymbol{A}$ are

$$1 - \alpha \lambda_i,$$

and the norm of $I - \alpha \boldsymbol{A}$ is

$$\max_i |1 - \alpha \lambda_i| = |\max (1 - \alpha \lambda_1, 1 - \alpha \lambda_n)| .$$

This is minimized by taking

$$\alpha = \frac{2}{\lambda_n + \lambda_1},$$

in which case the smallest and largest eigenvalues of $I - \alpha \boldsymbol{A}$ become

$$\pm \frac{\lambda_n - \lambda_1}{\lambda_n + \lambda_1},$$

and the norm of $I - \alpha \boldsymbol{A}$ becomes

$$1 - \frac{2\lambda_1}{\lambda_n + \lambda_1}.$$

While we might not know $\lambda_n + \lambda_1$, a good guess is often sufficient. If we choose an $\alpha < 2/(\lambda_n + \lambda_1)$, then the norm of $I - \alpha \boldsymbol{A}$ is at most

$$1 - \alpha \lambda_1.$$

To show that $\boldsymbol{x}^t$ converges to the solution, $\boldsymbol{x}$, consider the difference $\boldsymbol{x} - \boldsymbol{x}^t$. We have

$$\begin{aligned} \boldsymbol{x} - \boldsymbol{x}^t &= ((I - \alpha \boldsymbol{A})\boldsymbol{x} + \alpha \boldsymbol{b}) - \big((I - \alpha \boldsymbol{A})\boldsymbol{x}^{t-1} + \alpha \boldsymbol{b}\big) \\ &= (I - \alpha \boldsymbol{A})(\boldsymbol{x} - \boldsymbol{x}^{t-1}). \end{aligned}$$

So,

$$\boldsymbol{x} - \boldsymbol{x}^t = (I - \alpha \boldsymbol{A})^t (\boldsymbol{x} - \boldsymbol{x}^0) = (I - \alpha \boldsymbol{A})^t \boldsymbol{x}.$$

and

$$\begin{aligned}\left\| \boldsymbol{x} - \boldsymbol{x}^t \right\| = \left\| (I - \alpha \boldsymbol{A})^t \boldsymbol{x} \right\| &\leq \left\| (I - \alpha \boldsymbol{A})^t \right\| \left\| \boldsymbol{x} \right\| \\ &= \left\| (I - \alpha \boldsymbol{A}) \right\|^t \left\| \boldsymbol{x} \right\| \\ &\leq \left(1 - \frac{2\lambda_1}{\lambda_n + \lambda_1} \right)^t \left\| \boldsymbol{x} \right\| . \\ &\leq e^{-2\lambda_1 t/(\lambda_n + \lambda_1)} \left\| \boldsymbol{x} \right\| .\end{aligned}$$

So, if we want to get a solution $\boldsymbol{x}^t$ with

$$\frac{\left\| \boldsymbol{x} - \boldsymbol{x}^t \right\|}{\left\| \boldsymbol{x} \right\|} \leq \epsilon,$$

it suffices to run for

$$\frac{\lambda_n + \lambda_1}{2\lambda_1} \ln(1/\epsilon) = \left(\frac{\lambda_n}{2\lambda_1} + \frac{1}{2} \right) \ln(1/\epsilon).$$

iterations. The term

$$\frac{\lambda_n}{\lambda_1}$$

is called the *condition number*[1] of the matrix $\boldsymbol{A}$, when $\boldsymbol{A}$ is symmetric. It is often written $\kappa(\boldsymbol{A})$, and the running time of iterative algorithms is often stated in terms of this quantity. We see that if the condition number is small, then this algorithm quickly provides an approximate solution.

34.3 Expanders

Let's pause a moment to consider the problem of solving systems in the Laplacians of expander graphs. These are singular, but we know that their nullspace is spanned by the constant vectors. So, if we work orthogonal to the constant vectors their effective smallest eigenvalue is λ_2. If the graph is an ϵ-expander, then its condition number, λ_n/λ_2, will be approximately $1 + 2\epsilon$. Thus, we can solve systems of linear equations in this Laplacian very quickly.

This should make intuitive sense: the Laplacian of an expander is an approximation of the Laplacian of a complete graph. And, the Laplacians of complete graphs act as multiples of the identity on the space orthogonal to constant vectors.

In contrast, Gaussian elimination on expanders is slow: it takes time $\Omega(n^3)$ and requires space $\Omega(n^2)$ [LRT79].

[1] For general matrices, the condition number is defined to be the ratio of the largest to smallest singular value.

34.4 The norm of the residual

Thinking about $\|\boldsymbol{x} - \boldsymbol{x}_t\|$ is a little awkward because we do not know $\boldsymbol{x}$. For this reason, people often measure the quality of approximation of a solution to a system of linear equations by $\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_t\|$. For this quantity, the same sort of convergence results hold. First observe that

$$\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_t = \boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}\boldsymbol{x}_t = \boldsymbol{A}(\boldsymbol{I} - \alpha\boldsymbol{A})^t\boldsymbol{x} = (\boldsymbol{I} - \alpha\boldsymbol{A})^t\boldsymbol{A}\boldsymbol{x} = (\boldsymbol{I} - \alpha\boldsymbol{A})^t\boldsymbol{b}.$$

So, the right choice of α guarantees that

$$\frac{\|\boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_t\|}{\|\boldsymbol{b}\|} \leq e^{-2\lambda_1 t/(\lambda_n + \lambda_1)}.$$

In Chapter 35 we will encounter a more useful measure of convergence—convergence in the $\boldsymbol{A}$-norm.

34.5 A polynomial approximation of the inverse

I am now going to give another interpretation of Richardson's iteration. It provides us with a polynomial in $\boldsymbol{A}$ that approximates $\boldsymbol{A}^{-1}$. In particular, the tth iterate, $\boldsymbol{x}^t$ can be expressed in the form

$$p_t(\boldsymbol{A})\boldsymbol{b},$$

where p_t is a polynomial of degree t.

We will view $p_t(\boldsymbol{A})$ as a good approximation of $\boldsymbol{A}^{-1}$ if

$$\|\boldsymbol{A}p_t(\boldsymbol{A}) - I\|$$

is small. From the formula defining Richardson's iteration (34.1), we find

$$\begin{aligned}
\boldsymbol{x}^0 &= \boldsymbol{0}, \\
\boldsymbol{x}^1 &= \alpha\boldsymbol{b}, \\
\boldsymbol{x}^2 &= (\boldsymbol{I} - \alpha\boldsymbol{A})\alpha\boldsymbol{b} + \alpha\boldsymbol{b}, \\
\boldsymbol{x}^3 &= (\boldsymbol{I} - \alpha\boldsymbol{A})^2\alpha\boldsymbol{b} + (\boldsymbol{I} - \alpha\boldsymbol{A})\alpha\boldsymbol{b} + \alpha\boldsymbol{b}, \text{and} \\
\boldsymbol{x}^t &= \sum_{i=0}^{t}(\boldsymbol{I} - \alpha\boldsymbol{A})^i\alpha\boldsymbol{b}.
\end{aligned}$$

To get some idea of why this should be an approximation of $\boldsymbol{A}^{-1}$, consider the limit as t goes to infinity. Assuming that the infinite sum converges, we obtain

$$\alpha\sum_{i=0}^{\infty}(\boldsymbol{I} - \alpha\boldsymbol{A})^i = \alpha\left(\boldsymbol{I} - (\boldsymbol{I} - \alpha\boldsymbol{A})\right)^{-1} = \alpha(\alpha\boldsymbol{A})^{-1} = \boldsymbol{A}^{-1}.$$

So, the Richardson iteration can be viewed as a truncation of this infinite summation.

In general, a polynomial p_t will enable us to compute a solution to precision ϵ if

$$\|p_t(\boldsymbol{A})\boldsymbol{b} - \boldsymbol{x}\| \leq \epsilon \|\boldsymbol{x}\|.$$

As $\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}$, this is equivalent to

$$\|p_t(\boldsymbol{A})\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x}\| \leq \epsilon \|\boldsymbol{x}\|,$$

which is equivalent to

$$\|\boldsymbol{A}p_t(\boldsymbol{A}) - \boldsymbol{I}\| \leq \epsilon$$

34.6 Better Polynomials

This leads us to the question of whether we can find better polynomial approximations to $\boldsymbol{A}^{-1}$. The reason I ask is that the answer is yes! As $\boldsymbol{A}$, $p_t(\boldsymbol{A})$ and $\boldsymbol{I}$ all commute, the matrix

$$\boldsymbol{A}p_t(\boldsymbol{A}) - \boldsymbol{I}$$

is symmetric and its norm is the maximum absolute value of its eigenvalues. So, it suffices to find a polynomial p_t such that

$$|\lambda_i p_t(\lambda_i) - 1| \leq \epsilon,$$

for all eigenvalues λ_i of $\boldsymbol{A}$.

To reformulate this, define

$$q_t(x) = 1 - xp(x).$$

Then, it suffices to find a polynomial q_t of degree $t+1$ for which

$$\begin{aligned} q_t(0) &= 1, \text{ and} \\ |q_t(x)| &\leq \epsilon, \text{ for } \lambda_1 \leq x \leq \lambda_n. \end{aligned}$$

We will see that there are polynomials of degree

$$\ln(2/\epsilon)\left(\sqrt{\lambda_n/\lambda_1} + 1\right)/2$$

that satisfy these conditions and thus allow us to compute solutions of accuracy ϵ. In terms of the condition number of $\boldsymbol{A}$, this is a quadratic improvement over Richardson's first-order method.

Theorem 34.6.1. *For every $t \geq 1$, and $0 < \lambda_{min} \leq \lambda_{max}$, there exists a polynomial $q_t(x)$ such that*

1. *$|q_t(x)| \leq \epsilon$, for $\lambda_{min} \leq x \leq \lambda_{min}$, and*
2. *$q_t(0) = 1$,*

for

$$\epsilon \leq 2(1 + 2/\sqrt{\kappa})^{-t} \leq 2e^{-2t/\sqrt{\kappa}},$$

where

$$\kappa = \frac{\lambda_{max}}{\lambda_{min}}.$$

34.7 Chebyshev Polynomials

I'd now like to explain how we find these better polynomials. The key is to transform one of the most fundamental polynomials: the Chebyshev polynomials. These polynomials are as small as possible on $[-1,1]$, and grow quickly outside this interval. We will translate the interval $[-1,1]$ to obtain the polynomials we need.

The tth Chebyshev polynomial, $T_t(x)$ has degree t, and may be defined by setting

$$T_0(x) = 1, \quad T_1(x) = x,$$

and for $t \geq 2$

$$T_t(x) = 2xT_{t-1}(x) - T_{t-2}(x).$$

These polynomials are best understood by realizing that they are the polynomials for which

$$\cos(t\theta) = T_t(\cos(\theta)) \quad \text{and} \quad \cosh(t\theta) = T_t(\cosh(\theta)).$$

It might not be obvious that one can express $\cos(t\theta)$ as a polynomial in $\cos(\theta)$. To see this, and the correctness of the above formulas, recall that

$$\cos(\theta) = \frac{1}{2}\left(e^{i\theta} + e^{-i\theta}\right), \quad \text{and} \quad \cosh(\theta) = \frac{1}{2}\left(e^{\theta} + e^{-\theta}\right).$$

To verify that these satisfy the stated recurrences with $x = \cos(\theta)$, compute

$$\begin{aligned}
2T_{t-1}(x) - T_{t-2}(x) &= \frac{1}{2}\left(e^{\theta} + e^{-\theta}\right)\left(e^{(t-1)\theta} + e^{-(t-1)\theta}\right) - \frac{1}{2}\left(e^{(t-2)\theta} + e^{-(t-2)\theta}\right) \\
&= \frac{1}{2}\left(e^{(t\theta} + e^{-t\theta}\right) + \frac{1}{2}\left(e^{(t-2)\theta} + e^{-(t-2)\theta}\right) - \frac{1}{2}\left(e^{(t-2)\theta} + e^{-(t-2)\theta}\right) \\
&= \frac{1}{2}\left(e^{(t\theta} + e^{-t\theta}\right).
\end{aligned}$$

Thus,

$$T_t(x) = \begin{cases} \cos(t\,\mathrm{acos}(x)) & \text{for } |x| \leq 1, \text{ and} \\ \cosh(t\,\mathrm{acosh}(x)) & \text{for } x \geq 1. \end{cases}$$

Claim 34.7.1. *For $x \in [-1,1]$, $|T_t(x)| \leq 1$.*

Proof. For $x \in [-1,1]$, there is a θ so that $\cos(\theta) = x$. We then have $T_t(x) = \cos(t\theta)$, which must also be between -1 and 1. □

To compute the values of the Chebyshev polynomials outside $[-1,1]$, we use the hyperbolic cosine function. Hyperbolic cosine maps the real line to $[1,\infty]$ and is symmetric about the origin. So, the inverse of hyperbolic cosine may be viewed as a map from $[1,\infty]$ to $[0,\infty]$, and satisfies

$$\mathrm{acosh}(x) = \ln\left(x + \sqrt{x^2 - 1}\right), \text{ for } x \geq 1.$$

Claim 34.7.2. *For $\gamma > 0$,*

$$T_t(1+\gamma) \geq (1+\sqrt{2\gamma})^t/2.$$

Proof. Setting $x = 1+\gamma$, we compute

$$\begin{aligned}
T_t(x) &= \frac{1}{2}\left(e^{t\,\mathrm{acosh}(x)} + e^{-t\,\mathrm{acosh}(x)}\right)\\
&\geq \frac{1}{2}\left(e^{t\,\mathrm{acosh}(x)}\right)\\
&= \frac{1}{2}(x+\sqrt{x^2-1})^t\\
&= \frac{1}{2}(1+\gamma+\sqrt{(1+\gamma)^2-1})^t\\
&= \frac{1}{2}(1+\gamma+\sqrt{2\gamma+\gamma^2})^t\\
&\geq \frac{1}{2}(1+\sqrt{2\gamma})^t.
\end{aligned}$$

□

34.8 Proof of Theorem 34.6.1

We will exploit the following properties of the Chebyshev polynomials:

1. T_t has degree t.
2. $T_t(x) \in [-1,1]$, for $x \in [-1,1]$.
3. $T_t(x)$ is monotonically increasing for $x \geq 1$.
4. $T_t(1+\gamma) \geq (1+\sqrt{2\gamma})^t/2$, for $\gamma > 0$.

To express $q_t(x)$ in terms of a Chebyshev polynomial, we should map the range on which we want q_t to be small, $[\lambda_{min}, \lambda_{max}]$ to $[-1,1]$. We will accomplish this with the linear map:

$$l(x) \stackrel{\text{def}}{=} \frac{\lambda_{max}+\lambda_{min}-2x}{\lambda_{max}-\lambda_{min}}.$$

Note that

$$l(x) = \begin{cases} -1 & \text{if } x = \lambda_{max}\\ 1 & \text{if } x = \lambda_{min}\\ \frac{\lambda_{max}+\lambda_{min}}{\lambda_{max}-\lambda_{min}} & \text{if } x = 0.\end{cases}$$

To guarantee that the constant coefficient in $q_t(x)$ is one ($q_t(0) = 1$), we should set

$$q_t(x) \stackrel{\text{def}}{=} \frac{T_t(l(x))}{T_t(l(0))}.$$

We know that $|T_t(l(x))| \leq 1$ for $x \in [\lambda_{min}, \lambda_{max}]$. To find $q(x)$ for x in this range, we must compute $T_t(l(0))$. We have

$$l(0) \geq 1 + 2/\kappa(\boldsymbol{A}),$$

and so by properties 3 and 4 of Chebyshev polynomials,

$$T_t(l(0)) \geq (1 + 2/\sqrt{\kappa})^t/2.$$

Thus,

$$q(x) \leq 2(1 + 2/\sqrt{\kappa})^{-t},$$

for $x \in [\lambda_{min}, \lambda_{max}]$, and so all eigenvalues of $q(\boldsymbol{A})$ will have absolute value at most $2(1+2/\sqrt{\kappa})^{-t}$.

34.9 Laplacian Systems

One might at first think that these techniques do not apply to Laplacian systems, as these are always singular. However, we can apply these techniques without change if $\boldsymbol{b}$ is in the span of $\boldsymbol{L}$. That is, if $\boldsymbol{b}$ is orthogonal to the all-1s vector and the graph is connected. In this case the eigenvalue $\lambda_1 = 0$ has no role in the analysis, and it is replaced by λ_2. One way of understanding this is to just view $\boldsymbol{L}$ as an operator acting on the space orthogonal to the all-1s vector.

By considering the example of the Laplacian of the path graph, one can show that it is impossible to do much better than the $\sqrt{\kappa}$ iteration bound that I claimed at the end of the last section. To see this, first observe that when one multiplies a vector $\boldsymbol{x}$ by $\boldsymbol{L}$, the entry $(\boldsymbol{L}\boldsymbol{x})(i)$ just depends on $\boldsymbol{x}(i-1), \boldsymbol{x}(i)$, and $\boldsymbol{x}(i+1)$. So, if we apply a polynomial of degree at most t, $\boldsymbol{x}^t(i)$ will only depend on $\boldsymbol{b}(j)$ with $i - t \leq j \leq i + t$. This tells us that we will need a polynomial of degree on the order of n to solve such a system.

On the other hand, $\sqrt{\lambda_n/\lambda_2}$ is on the order of n as well. So, we should not be able to solve the system with a polynomial whose degree is significantly less than $\sqrt{\lambda_n/\lambda_2}$.

34.10 Warning

The polynomial-based approach that I have described here only works in infinite precision arithmetic. In finite precision arithmetic one has to be more careful about how one implements these algorithms. This is why the descriptions of methods such as the Chebyshev method found in Numerical Linear Algebra textbooks are more complicated than that presented here. The algorithms that are actually used are mathematically identical in infinite precision, but they actually work. The problem with the naive implementations are the typical experience: in double-precision arithmetic the polynomial approach to Chebyshev will fail to solve linear systems in random positive definite matrices in 60 dimensions!

Chapter 35

The Conjugate Gradient and Diameter

We introduce the matrix norm as the measure of convergence of iterative methods, and show how the Conjugate Gradient method efficiently minimizes it. We finish by relating the rate of convergence of any iterative method on a Laplacian matrix to the diameter of the underlying graph.

My description of the Conjugate Gradient method is inspired by Vishnoi's [Vis12]. It is the simplest explanation of the Conjugate Gradient that I have seen.

35.1 The Matrix Norm

Recall from Chapter 14 that for a positive semidefinite matrix $\boldsymbol{A}$, the matrix norm in $\boldsymbol{A}$ is defined by

$$\|\boldsymbol{x}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}} = \left\| \boldsymbol{A}^{1/2} \boldsymbol{x} \right\|.$$

For many applications, the right way to measure the quality of approximation of a system of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ is by $\|\boldsymbol{x} - \boldsymbol{x}_t\|_{\boldsymbol{A}}$. Many algorithms naturally produce bounds on the error in the matrix norm. And, for many applications that use linear equation solvers as subroutines, this is the measure of accuracy in the subroutine that most naturally translates to accuracy of the outside algorithm.

We should observe that both the Richardon and Chebyshev methods achieve ϵ error in the $\boldsymbol{A}$-norm. Let p be a polynomial such that

$$\|p(\boldsymbol{A})\boldsymbol{A} - \boldsymbol{I}\| \leq \epsilon.$$

Then,

$$\|p(\boldsymbol{A})\boldsymbol{b} - \boldsymbol{x}\|_{\boldsymbol{A}} = \left\| \boldsymbol{A}^{1/2} p(\boldsymbol{A}) \boldsymbol{A} \boldsymbol{x} - \boldsymbol{A}^{1/2} \boldsymbol{x} \right\| = \left\| (p(\boldsymbol{A})\boldsymbol{A} - \boldsymbol{I}) \boldsymbol{A}^{1/2} \boldsymbol{x} \right\| \leq \epsilon \left\| \boldsymbol{A}^{1/2} \boldsymbol{x} \right\| = \epsilon \|\boldsymbol{x}\|_{\boldsymbol{A}}.$$

The analysis above works because these methods produce $\boldsymbol{x}_t$ by applying a linear operator, $\boldsymbol{p}(\boldsymbol{A})$, to $\boldsymbol{b}$ that commutes with $\boldsymbol{A}$. While most of the algorithms we use to solve systems of equations in $\boldsymbol{A}$ will be linear operators, they will typically not commute with $\boldsymbol{A}$. But, they will produce small error in the $\boldsymbol{A}$-norm.

The following theorem shows that a linear operator $\boldsymbol{Z}$ is an ϵ approximation of $\boldsymbol{A}^{-1}$ if and only if it produces at most ϵ error in the $\boldsymbol{A}$-norm when used to solve systems of linear equations in $\boldsymbol{A}$.

Theorem 35.1.1. *Let $\boldsymbol{A}$ and $\boldsymbol{Z}$ be positive definite matrices. Then*

$$\left\| \boldsymbol{Z}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x} \right\|_{\boldsymbol{A}} \leq \epsilon \left\| \boldsymbol{x} \right\|_{\boldsymbol{A}} \tag{35.1}$$

for all $\boldsymbol{x}$ if and only if

$$(1-\epsilon)\boldsymbol{A}^{-1} \preccurlyeq \boldsymbol{Z} \preccurlyeq (1+\epsilon)\boldsymbol{A}^{-1}.$$

Proof. The assertion that (35.1) holds for all $\boldsymbol{x}$ is equivalent to the assertion that for all $\boldsymbol{x}$,

$$\left\| \boldsymbol{A}^{1/2}(\boldsymbol{Z}\boldsymbol{A} - \boldsymbol{I})\boldsymbol{x} \right\| \leq \epsilon \left\| \boldsymbol{A}^{1/2}\boldsymbol{x} \right\|.$$

Setting $\boldsymbol{y} = \boldsymbol{A}^{1/2}\boldsymbol{x}$, this becomes equivalent to saying that for all $\boldsymbol{y}$,

$$\left\| (\boldsymbol{A}^{1/2}\boldsymbol{Z}\boldsymbol{A}^{1/2} - \boldsymbol{I})\boldsymbol{y} \right\| \leq \epsilon \left\| \boldsymbol{y} \right\|,$$

which we usually write

$$\left\| \boldsymbol{A}^{1/2}\boldsymbol{Z}\boldsymbol{A}^{1/2} - \boldsymbol{I} \right\| \leq \epsilon.$$

This is in turn equivalent to

$$\begin{aligned}
-\epsilon\boldsymbol{I} \preccurlyeq \boldsymbol{A}^{1/2}\boldsymbol{Z}\boldsymbol{A}^{1/2} - \boldsymbol{I} &\preccurlyeq \epsilon\boldsymbol{I} && \Longleftrightarrow \\
(1-\epsilon)\boldsymbol{I} \preccurlyeq \boldsymbol{A}^{1/2}\boldsymbol{Z}\boldsymbol{A}^{1/2} &\preccurlyeq (1+\epsilon)\boldsymbol{I} && \Longleftrightarrow \\
(1-\epsilon)\boldsymbol{A}^{-1} \preccurlyeq \boldsymbol{Z} &\preccurlyeq (1+\epsilon)\boldsymbol{A}^{-1},
\end{aligned}$$

where the last statement follows from multiplying on the left and right by $\boldsymbol{A}^{-1/2}$. □

35.2 Application: Approximating Fiedler Vectors

Approximately computing eigenvectors of the smallest eigenvalues of matrices, such as Fiedler vectors, is one application in which approximation in the $\boldsymbol{A}$-norm is the right thing to do. In problem [?], we saw that the largest eigenvalue of a matrix can be approximated using the power method. If we want the smallest eigenvalue, it is natural to use the power method on the inverse of the matrix.

As we are only going to compute an approximation of the eigenvalue and its corresponding eigenvector, we might as well use an approximation of the matrix inverse. If $\boldsymbol{Z}$ is an operator that ϵ-approximates $\boldsymbol{A}^{-1}$, then the largest eigenvalue of $\boldsymbol{Z}$ is within $1 \pm \epsilon$ of the largest eigenvalue of $\boldsymbol{A}^{-1}$, and the corresponding eigenvector has large Rayleigh quotient with respect to $\boldsymbol{A}^{-1}$. As we learned in problem [?], if there is a gap between this and the next eigenvalue, then this vector makes a small angle with the eigenvector. See [ST14, Section 7] for a more detailed discussion.

35.3 Optimality in the $\boldsymbol{A}$-norm

The iterative methods that we consider begin with the vector $\boldsymbol{b}$, and then perform multiplications by $\boldsymbol{A}$ and take linear combinations with vectors that have already been produced. So, after t iterations they produce a vector that is in the span of

$$\left\{\boldsymbol{b}, \boldsymbol{A}\boldsymbol{b}, \boldsymbol{A}^2\boldsymbol{b}, \ldots, \boldsymbol{A}^t\boldsymbol{b}\right\}.$$

This subspace is called the $t+1$st Krylov subspace generated by $\boldsymbol{A}$ and $\boldsymbol{b}$.

The Conjugate Gradient will find the vector $\boldsymbol{x}_t$ in this subspace that minimizes the error in the $\boldsymbol{A}$-norm. It will do so by computing a very useful basis of this subspace. But, before we describe this basis, let's examine the error in the $\boldsymbol{A}$ norm.

We have

$$\|\boldsymbol{x}_t - \boldsymbol{x}\|_{\boldsymbol{A}}^2 = \boldsymbol{x}_t^T\boldsymbol{A}\boldsymbol{x}_t - 2\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}_t^T\boldsymbol{A}\boldsymbol{x}_t - 2\boldsymbol{b}^T\boldsymbol{x}_t + \boldsymbol{b}^T\boldsymbol{x}.$$

While we do not know $\boldsymbol{b}^T\boldsymbol{x}$, we do know that $\|\boldsymbol{x}_t - \boldsymbol{x}\|_{\boldsymbol{A}}^2$ is minimized when we minimize

$$\frac{1}{2}\boldsymbol{x}_t^T\boldsymbol{A}\boldsymbol{x}_t - \boldsymbol{b}^T\boldsymbol{x}_t. \tag{35.2}$$

So, we will work to minimize (35.2).

Let $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_t$ be a basis of the $t+1$st Krylov subspace, and let

$$\boldsymbol{x}_t = \sum_{i=0}^{t} c_i\boldsymbol{p}_i.$$

We would like to find the coefficients c_i that minimize (35.2). Expanding $\boldsymbol{x}$ gives

$$\begin{aligned}\frac{1}{2}\boldsymbol{x}_t^T\boldsymbol{A}\boldsymbol{x}_t - \boldsymbol{b}^T\boldsymbol{x}_t &= \frac{1}{2}\left(\sum_{i=0}^{t} c_i\boldsymbol{p}_i\right)^T\boldsymbol{A}\left(\sum_{i=0}^{t} c_i\boldsymbol{p}_i\right) - \boldsymbol{b}^T\left(\sum_{i=0}^{t} c_i\boldsymbol{p}_i\right)\\ &= \frac{1}{2}\sum_{i=0}^{t} c_i^2\boldsymbol{p}_i^T\boldsymbol{A}\boldsymbol{p}_i - \sum_{i=0}^{t} c_i\boldsymbol{b}^T\boldsymbol{p}_i + \frac{1}{2}\sum_{i\neq j} c_ic_j\boldsymbol{p}_i^T\boldsymbol{A}\boldsymbol{p}_j.\end{aligned}$$

To simplify the selection of the optimal constants c_i, the Conjugate Gradient will compute a basis $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_t$ that makes the rightmost term 0. That is, it will compute a basis such that $\boldsymbol{p}_i^T\boldsymbol{A}\boldsymbol{p}_j = 0$ for all $i \neq j$. Such a basis is called an $\boldsymbol{A}$-orthogonal basis.

When the last term is zero, the objective function becomes

$$\sum_{i=0}^{t}\left(\frac{1}{2}c_i^2\boldsymbol{p}_i^T\boldsymbol{A}\boldsymbol{p}_i - c_i\boldsymbol{b}^T\boldsymbol{p}_i\right).$$

So, the terms corresponding to different is do not interact, and we can minimize the sum by minimizing each term individually. The term

$$\frac{1}{2}c_i^2\boldsymbol{p}_i^T\boldsymbol{A}\boldsymbol{p}_i - c_i\boldsymbol{b}^T\boldsymbol{p}_i$$

is minimized by setting its derivative in c_i equal to zero, which gives

$$c_i = \frac{\boldsymbol{b}^T \boldsymbol{p}_i}{\boldsymbol{p}_i^T \boldsymbol{A} \boldsymbol{p}_i}.$$

It remains to describe how we compute this $\boldsymbol{A}$-orthogonal basis. The algorithm begins by setting

$$\boldsymbol{p}_0 = \boldsymbol{b}.$$

The next vector should be $\boldsymbol{A}\boldsymbol{p}_0$, but $\boldsymbol{A}$-orthogonalized with respect to $\boldsymbol{p}_0$. That is,

$$\boldsymbol{p}_1 = \boldsymbol{A}\boldsymbol{p}_0 - \boldsymbol{p}_0 \frac{(\boldsymbol{A}\boldsymbol{p}_0)^T \boldsymbol{A}\boldsymbol{p}_0}{\boldsymbol{p}_0^T \boldsymbol{A}\boldsymbol{p}_0}.$$

It is immediate that

$$\boldsymbol{p}_0^T \boldsymbol{A}\boldsymbol{p}_1 = 0.$$

In general, we set

$$\boldsymbol{p}_{t+1} = \boldsymbol{A}\boldsymbol{p}_t - \sum_{i=0}^{t} \boldsymbol{p}_i \frac{(\boldsymbol{A}\boldsymbol{p}_t)^T \boldsymbol{A}\boldsymbol{p}_i}{\boldsymbol{p}_i^T \boldsymbol{A}\boldsymbol{p}_i}. \tag{35.3}$$

Let's verify that $\boldsymbol{p}_{t+1}$ is $\boldsymbol{A}$-orthogonal to $\boldsymbol{p}_i$ for $i \leq t$, assuming that $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_t$ are $\boldsymbol{A}$-orthogonal. We have

$$\begin{aligned}
\boldsymbol{p}_j^T \boldsymbol{A}\boldsymbol{p}_{t+1} &= \boldsymbol{p}_j^T \boldsymbol{A}\boldsymbol{A}\boldsymbol{p}_t - \sum_{i=0}^{t} \boldsymbol{p}_j^T \boldsymbol{A}\boldsymbol{p}_i \frac{(\boldsymbol{A}\boldsymbol{p}_t)^T \boldsymbol{A}\boldsymbol{p}_i}{\boldsymbol{p}_i^T \boldsymbol{A}\boldsymbol{p}_i} \\
&= \boldsymbol{p}_j^T \boldsymbol{A}^2 \boldsymbol{p}_t - \boldsymbol{p}_j^T \boldsymbol{A}\boldsymbol{p}_j \frac{(\boldsymbol{A}\boldsymbol{p}_t)^T \boldsymbol{A}\boldsymbol{p}_i}{\boldsymbol{p}_j^T \boldsymbol{A}\boldsymbol{p}_j} \\
&= 0.
\end{aligned}$$

The computation of $\boldsymbol{p}_{t+1}$ is greatly simplified by the observation that all but two of the terms in the sum (35.3) are zero: for $i < t-1$,

$$(\boldsymbol{A}\boldsymbol{p}_t)^T \boldsymbol{A}\boldsymbol{p}_i = 0.$$

To see this, note that

$$(\boldsymbol{A}\boldsymbol{p}_t)^T \boldsymbol{A}\boldsymbol{p}_i = \boldsymbol{p}_t^T \boldsymbol{A}(\boldsymbol{A}\boldsymbol{p}_i),$$

and that $\boldsymbol{A}\boldsymbol{p}_i$ is in the span of $\{\boldsymbol{p}_0, \ldots, \boldsymbol{p}_{i+1}\}$. So, this term will be zero if $i+1 < t$.

That means that

$$\boldsymbol{p}_{t+1} = \boldsymbol{A}\boldsymbol{p}_t - \boldsymbol{p}_t \frac{(\boldsymbol{A}\boldsymbol{p}_t)^T \boldsymbol{A}\boldsymbol{p}_t}{\boldsymbol{p}_t^T \boldsymbol{A}\boldsymbol{p}_t} - \boldsymbol{p}_{t-1} \frac{(\boldsymbol{A}\boldsymbol{p}_t)^T \boldsymbol{A}\boldsymbol{p}_{t-1}}{\boldsymbol{p}_{t-1}^T \boldsymbol{A}\boldsymbol{p}_{t-1}}.$$

So, one can compute $\boldsymbol{p}_{t+1}$ from $\boldsymbol{p}_t$ and $\boldsymbol{p}_{t-1}$ while using only a constant number of multiplications by $\boldsymbol{A}$ and a constant number of vector operations. This means that one can compute the entire basis $\boldsymbol{p}_0, \ldots, \boldsymbol{p}_t$ while performing only $O(t)$ multiplications of vectors by $\boldsymbol{A}$ and $O(t)$ vector operations.

The computation of $\boldsymbol{x}_t$ by

$$\boldsymbol{x}_t = \sum_{i=0}^{t} \boldsymbol{p}_i \frac{\boldsymbol{b}^T \boldsymbol{p}_i}{\boldsymbol{p}_i^T \boldsymbol{A} \boldsymbol{p}_i}.$$

Only requires an additional $O(t)$ more such operations.

In fact, only t multiplications by $\boldsymbol{A}$ are required to compute $\boldsymbol{p}_0, \dots, \boldsymbol{p}_t$ and $\boldsymbol{x}_1, \dots, \boldsymbol{x}_t$: every term in the expressions for these vectors can be derived from the products $\boldsymbol{A}\boldsymbol{p}_i$. Thus, the Conjugate Gradient algorithm can find the $\boldsymbol{x}_t$ in the $t+1$st Krylov subspace that minimizes the error in the $\boldsymbol{A}$-norm in time $O(tn)$ plus the time required to perform t multiplications by $\boldsymbol{A}$.

Caution: the algorithm that I have presented here differs from the implemented Conjugate gradient in that the implemented Conjugate Gradient re-arranges this computation to keep the norms of the vectors involved reasonably small. Without this adjustment, the algorithm that I've described will fail in practice as the vectors $\boldsymbol{p}_i$ will become too large.

35.4 How Good is CG?

The Conjugate Gradient is at least as good as the Chebyshev iteration, in that it finds a vector of smaller error in the $\boldsymbol{A}$-norm in any given number of iterations. The optimality property of the Conjugate Gradient causes it to perform remarkably well.

For example, one can see that it should never require more than n iterations. The vector $\boldsymbol{x}$ is always in nth Krylov subspace. Here's an easy way to see this. Let the *distinct* eigenvalues of $\boldsymbol{A}$ be $\lambda_1, \dots, \lambda_k$. Now, consider the polynomial

$$q(x) \stackrel{\text{def}}{=} \frac{\prod_{i=1}^{k} (\lambda_i - x)}{\prod_{i=1}^{k} \lambda_i}.$$

You can verify that q is a degree k polynomial such that

$$\begin{aligned} q(0) &= 1, \text{ and} \\ q(\lambda_i) &= 0, \text{ for all } i. \end{aligned}$$

So, CG should be able to find the exact answer to a system in $\boldsymbol{A}$ in $k-1$ iterations. I say "should" because, while this statement is true with infinite precision arithmetic, it doesn't work out quite this well in practice.

Ignoring for now issues of finite arithmetic, let's consider the importance of this for sparse matrices $\boldsymbol{A}$. By a sparse matrix, I mean one with at most cn non-zero entries, for some constant c. That's not a rigorous definition, but it will help guide our discussion. Multiplication by a sparse matrix can be done in time $O(n)$. So, CG can solve a system of equations in a sparse matrix in time $O(n^2)$. Note that this is proportional to how long it would take to just write the inverse of $\boldsymbol{A}$, and will probably be faster than any algorithm for computing the inverse. On the other hand, it only provides the solution to one system in $\boldsymbol{A}$.

For another interesting example, consider the hypercube graph on n vertices. It only has $\log_2 n$ distinct eigenvalues. So, CG will only need $\log_2 n$ iterations to solve linear systems in the

Laplacian of the hypercube. While there are other fast algorithms the exploit the special structure of the hypercube, CG works well when one has a graph that is merely very close to the hypercube.

In general, CG works especially quickly on matrices in which the eigenvalues appear in just a few clusters, and on matrices in which there are just a few extreme eigenvalues. We will learn more about this in the next lecture.

35.5 Laplacian Systems, again

This would be a good time to re-examine what we want when our matrix is a Laplacian. The Laplacian does not have an inverse. Rather, we want a polynomial in the Laplacian that approximates its pseudo-inverse (which we defined back in Lecture 8). If we were exactly solving the system of linear equations, we would have found a polynomial p such that

$$p(\boldsymbol{L})\boldsymbol{b} = \boldsymbol{x},$$

where $\boldsymbol{b} = \boldsymbol{L}\boldsymbol{x}$, so this gives

$$p(\boldsymbol{L})\boldsymbol{L}\boldsymbol{x} = \boldsymbol{x}.$$

Of course, this is only reasonable if $\boldsymbol{x}$ is in the span of $\boldsymbol{L}$. If the underlying graph is connected, this only happens if $\boldsymbol{x}$ is orthogonal to the all-1s vector. Of course, $\boldsymbol{L}$ sends constant vectors to zero. So, we want

$$p(\boldsymbol{L})\boldsymbol{L} = \boldsymbol{\Pi},$$

where $\boldsymbol{\Pi}$ is the projection matrix that sends the constant vectors to zero, and acts as an identity on the vectors that are orthogonal to the constant vectors. Recall that $\boldsymbol{\Pi} = \frac{1}{n}\boldsymbol{L}_{K_n}$.

Similarly, p gives an ϵ-approximation of the pseudo-inverse if

$$\left\| p(\boldsymbol{L})\boldsymbol{L} - \boldsymbol{\Pi} \right\| \leq \epsilon.$$

35.6 Bounds on the Diameter

Our intuition tells us that if we can quickly solve linear equations in the Laplacian matrix of a graph by an iterative method, then the graph should have small diameter. We now make that intuition precise.

If s and t are vertices that are at distance greater than d from each other, then

$$\boldsymbol{\chi}_s^T \boldsymbol{L}^d \boldsymbol{\chi}_t = 0.$$

On the other hand, if $\boldsymbol{L}$ only has k distinct eigenvalues other than 0, then we can form a polynomial p of degree $k-1$ such that

$$\boldsymbol{L}p(\boldsymbol{L}) = \boldsymbol{\Pi}.$$

This allows us to prove the following theorem.

Theorem 35.6.1. *Let G be a connected graph whose Laplacian has at most k distinct eigenvalues other than 0. Then, the diameter of G is at most k.*

Proof. Let d be the diameter of the graph and let s and t be two vertices at distance d from each other. We have

$$\boldsymbol{e}_s^T \boldsymbol{\Pi} \boldsymbol{e}_t = -1/n.$$

On the other hand, we have just described a polynomial in $\boldsymbol{L}$ with zero constant term, given by $\boldsymbol{L}p(\boldsymbol{L})$, that has degree k and such that

$$\boldsymbol{L}p(\boldsymbol{L}) = \boldsymbol{\Pi}.$$

If the degree of this polynomial were less than d, we would have

$$\boldsymbol{e}_s^T \boldsymbol{L}p(\boldsymbol{L}) \boldsymbol{e}_t = 0.$$

As this is not the case, we have $d \leq k$. □

We can similarly obtain bounds on the diameter from approximate pseudo-inverses. If p is a polynomial such that

$$\left\| p(\boldsymbol{L})\boldsymbol{L} - \boldsymbol{\Pi} \right\| \leq \epsilon,$$

then

$$\boldsymbol{e}_s^T (p(\boldsymbol{L})\boldsymbol{L} - \boldsymbol{\Pi}) \boldsymbol{e}_t \leq \left\| \boldsymbol{e}_s \right\| \left\| p(\boldsymbol{L})\boldsymbol{L} - \boldsymbol{\Pi} \right\| \left\| \boldsymbol{e}_t \right\| \leq \epsilon.$$

If s and t are at distance d from each other in the graph, and if the degree of $p(\boldsymbol{L})\boldsymbol{L}$ has degree less than d, then

$$\boldsymbol{e}_s^T (p(\boldsymbol{L})\boldsymbol{L} - \boldsymbol{\Pi}) \boldsymbol{e}_t = \boldsymbol{e}_s^T (-\boldsymbol{\Pi}) \boldsymbol{e}_t = 1/n.$$

This is a contradiction if $\epsilon < 1/n$. So, the polynomials we constructed from Chebyshev polynomials imply the following theorem of Chung, Faber and Manteuffel [CFM94]

Theorem 35.6.2. *Let $G = (V, E)$ be a connected graph, and let $\lambda_2 \leq \cdots \leq \lambda_n$ be its Laplacian eigenvalues. Then, the diameter of G is at most*

$$\left(\frac{1}{2}\sqrt{\frac{\lambda_n}{\lambda_2}} + 1 \right) \ln 2n.$$

Chapter 36

Preconditioning Laplacians

A *preconditioner* for a positive semidefinite matrix $\boldsymbol{A}$ is a positive semidefinite matrix $\boldsymbol{B}$ such that it is easy to solve systems of linear equations in $\boldsymbol{B}$ and the condition number of $\boldsymbol{B}^{-1}\boldsymbol{A}$ is small. A good preconditioner allows one to quickly solve systems of equations in $\boldsymbol{A}$.

In this lecture, we will measure the quality of preconditioners in terms of the ratio

$$\kappa(\boldsymbol{A}, \boldsymbol{B}) \stackrel{\text{def}}{=} \beta/\alpha,$$

where α is the largest number and β is the smallest such that

$$\alpha \boldsymbol{B} \preccurlyeq \boldsymbol{A} \preccurlyeq \beta \boldsymbol{B}.$$

Lemma 36.0.1. *Let α and β be as defined above. Then, α and β are the smallest and largest eigenvalues of $\boldsymbol{B}^{-1}\boldsymbol{A}$, excluding possible zero eigenvalues corresponding to a common nullspace of $\boldsymbol{A}$ and $\boldsymbol{B}$.*

We need to exclude the common nullspace when $\boldsymbol{A}$ and $\boldsymbol{B}$ are the Laplacian matrices of connected graphs. If these matrices have different nullspaces $\alpha = 0$ or $\beta = \infty$ and the condition number β/α is infinite.

Proof of Lemma 36.0.1. We just prove the statement for β, in the case where neither matrix is singular. We have

$$\begin{aligned} \lambda_{max}(\boldsymbol{B}^{-1}\boldsymbol{A}) &= \lambda_{max}(\boldsymbol{B}^{-1/2}\boldsymbol{A}\boldsymbol{B}^{-1/2}) \\ &= \max_{\boldsymbol{x}} \frac{\boldsymbol{x}^T\boldsymbol{B}^{-1/2}\boldsymbol{A}\boldsymbol{B}^{-1/2}\boldsymbol{x}}{\boldsymbol{x}^T\boldsymbol{x}} \\ &= \max_{\boldsymbol{y}} \frac{\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{y}}{\boldsymbol{y}^T\boldsymbol{B}\boldsymbol{y}}, \qquad \text{settting } \boldsymbol{y} = B^{-1/2}\boldsymbol{x}, \end{aligned}$$

which equals β. □

Recall that the eigenvalues of $\boldsymbol{B}^{-1}\boldsymbol{A}$ are the same as those of $\boldsymbol{B}^{-1/2}\boldsymbol{A}\boldsymbol{B}^{-1/2}$ and $\boldsymbol{A}^{1/2}\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2}$.

36.1 Approximate Solutions

Recall the $\boldsymbol{A}$-norm:

$$\|\boldsymbol{x}\|_{\boldsymbol{A}} = \sqrt{\boldsymbol{x}^T \boldsymbol{A} \boldsymbol{x}} = \left\|\boldsymbol{A}^{1/2}\boldsymbol{x}\right\|.$$

We say that $\widetilde{\boldsymbol{x}}$ is an ϵ-approximate solution to the problem $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ if

$$\|\widetilde{\boldsymbol{x}} - \boldsymbol{x}\|_{\boldsymbol{A}} \leq \epsilon \|\boldsymbol{x}\|_{\boldsymbol{A}}.$$

36.2 Iterative Refinement

We will now see how to use a *very* good preconditioner to solve a system of equations. Let's consider a preconditioner $\boldsymbol{B}$ that satisfies

$$(1-\epsilon)\boldsymbol{B} \preccurlyeq \boldsymbol{A} \preccurlyeq (1+\epsilon)\boldsymbol{B}.$$

So, all of the eigenvalues of

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

have absolute value at most ϵ.

The vector $\boldsymbol{B}^{-1}\boldsymbol{b}$ is a good approximation of $\boldsymbol{x}$ in the $\boldsymbol{A}$-norm. We have

$$\begin{aligned}
\left\|\boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{x}\right\|_{\boldsymbol{A}} &= \left\|\boldsymbol{A}^{1/2}\boldsymbol{B}^{-1}\boldsymbol{b} - \boldsymbol{A}^{1/2}\boldsymbol{x}\right\| \\
&= \left\|\boldsymbol{A}^{1/2}\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^{1/2}\boldsymbol{x}\right\| \\
&= \left\|\boldsymbol{A}^{1/2}\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2}(\boldsymbol{A}^{1/2}\boldsymbol{x}) - \boldsymbol{A}^{1/2}\boldsymbol{x}\right\| \\
&\leq \left\|\boldsymbol{A}^{1/2}\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2} - \boldsymbol{I}\right\| \left\|\boldsymbol{A}^{1/2}\boldsymbol{x}\right\| \\
&\leq \epsilon \left\|\boldsymbol{A}^{1/2}\boldsymbol{x}\right\| \\
&= \epsilon \|\boldsymbol{x}\|_{\boldsymbol{A}}.
\end{aligned}$$

Remark: This result crucially depends upon the use of the $\boldsymbol{A}$-norm. It fails under the Euclidean norm.

If we want a better solution, we can just compute the residual and solve the problem in the residual. That is, we set

$$\boldsymbol{x}_1 = \boldsymbol{B}^{-1}\boldsymbol{b},$$

and compute

$$\boldsymbol{r}_1 = \boldsymbol{b} - \boldsymbol{A}\boldsymbol{x}_1 = \boldsymbol{A}(\boldsymbol{x} - \boldsymbol{x}_1).$$

We then use one solve in $\boldsymbol{B}$ to compute a vector $\boldsymbol{x}_2$ such that

$$\|(\boldsymbol{x} - \boldsymbol{x}_1) - \boldsymbol{x}_2\|_{\boldsymbol{A}} \leq \epsilon \|\boldsymbol{x} - \boldsymbol{x}_1\|_{\boldsymbol{A}} \leq \epsilon^2 \|\boldsymbol{x}\|_{\boldsymbol{A}}.$$

So, $\boldsymbol{x}_1 + \boldsymbol{x}_2$, our new estimate of $\boldsymbol{x}$, differs from $\boldsymbol{x}$ by at most an ϵ^2 factor. Continuing in this way, we can find an ϵ^k approximation of $\boldsymbol{x}$ after solving k linear systems in $\boldsymbol{B}$. This procedure is called *iterative refinement*.

36.3 Iterative Methods in the Matrix Norm

The iterative methods we studied last class can also be shown to produce good approximate solutions in the matrix norm. Given a matrix $\boldsymbol{A}$, these produce ϵ-approximation solutions after t iterations if there is a polynomial q of degree t for which $q(0) = 1$ and $|q(\lambda_i)| \leq \epsilon$ for all eigenvalues of $\boldsymbol{A}$. To see this, recall that we can define $p(x)$ so that $q(x) = 1 - xp(x)$, and set

$$\widetilde{\boldsymbol{x}} = p(\boldsymbol{A})\boldsymbol{b},$$

to get

$$\left\| \widetilde{\boldsymbol{x}} - \boldsymbol{x} \right\|_{\boldsymbol{A}} = \left\| p(\boldsymbol{A})\boldsymbol{b} - \boldsymbol{x} \right\|_{\boldsymbol{A}} = \left\| p(\boldsymbol{A})\boldsymbol{A}\boldsymbol{x} - \boldsymbol{x} \right\|_{\boldsymbol{A}}.$$

As $\boldsymbol{I}$, $\boldsymbol{A}$, $p(\boldsymbol{A})$ and $\boldsymbol{A}^{1/2}$ all commute, this equals

$$\begin{aligned} \left\| \boldsymbol{A}^{1/2} p(\boldsymbol{A})\boldsymbol{A}\boldsymbol{x} - \boldsymbol{A}^{1/2}\boldsymbol{x} \right\| &= \left\| p(\boldsymbol{A})\boldsymbol{A}\boldsymbol{A}^{1/2}\boldsymbol{x} - \boldsymbol{A}^{1/2}\boldsymbol{x} \right\| \\ &\leq \left\| p(\boldsymbol{A})\boldsymbol{A} - \boldsymbol{I} \right\| \left\| \boldsymbol{A}^{1/2}\boldsymbol{x} \right\| \\ &\leq \epsilon \left\| \boldsymbol{x} \right\|_{\boldsymbol{A}}. \end{aligned}$$

36.4 Preconditioned Iterative Methods

Preconditioned iterative methods can be viewed as the extension of Iterative Refinement by algorithms like Chebyshev iteration and the Preconditioned Conjugate Gradient. These usually work with condition numbers much larger than 2.

In each iteration of a preconditioned method we will solve a system of equations in $\boldsymbol{B}$, multiply a vector by $\boldsymbol{A}$, and perform a constant number of other vector operations. For this to be worthwhile, the cost of solving equations in $\boldsymbol{B}$ has to be low.

We begin by seeing how the analysis with polynomials translates. Let λ_i be the ith eigenvalue of $\boldsymbol{B}^{-1}\boldsymbol{A}$. If $q_t(x) = 1 - xp_t(x)$ is a polynomial such that $|q_t(\lambda_i)| \leq \epsilon$ for all i, then

$$\boldsymbol{x}_t \stackrel{\text{def}}{=} p_t(\boldsymbol{B}^{-1}\boldsymbol{A})\boldsymbol{B}^{-1}\boldsymbol{b}$$

will be an ϵ-approximate solution to $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$:

$$\begin{aligned}
\|\boldsymbol{x} - \boldsymbol{x}_t\|_{\boldsymbol{A}} &= \left\|\boldsymbol{A}^{1/2}\boldsymbol{x} - \boldsymbol{A}^{1/2}\boldsymbol{x}_t\right\| \\
&= \left\|\boldsymbol{A}^{1/2}\boldsymbol{x} - \boldsymbol{A}^{1/2}p_t(\boldsymbol{B}^{-1}\boldsymbol{A})\boldsymbol{B}^{-1}\boldsymbol{b}\right\| \\
&= \left\|\boldsymbol{A}^{1/2}\boldsymbol{x} - \boldsymbol{A}^{1/2}p_t(\boldsymbol{B}^{-1}\boldsymbol{A})\boldsymbol{B}^{-1}\boldsymbol{A}\boldsymbol{x}\right\| \\
&= \left\|\boldsymbol{A}^{1/2}\boldsymbol{x} - \boldsymbol{A}^{1/2}p_t(\boldsymbol{B}^{-1}\boldsymbol{A})\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2}(\boldsymbol{A}^{1/2}\boldsymbol{x})\right\| \\
&\leq \left\|\boldsymbol{I} - \boldsymbol{A}^{1/2}p_t(\boldsymbol{B}^{-1}\boldsymbol{A})\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2}\right\| \left\|(\boldsymbol{A}^{1/2}\boldsymbol{x})\right\|.
\end{aligned}$$

We now prod this matrix into a more useful form:

$$\boldsymbol{I} - \boldsymbol{A}^{1/2}p_t(\boldsymbol{B}^{-1}\boldsymbol{A})\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2} = \boldsymbol{I} - p_t(\boldsymbol{A}^{1/2}\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2})\boldsymbol{A}^{1/2}\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2} = q_t(\boldsymbol{A}^{1/2}\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2}).$$

So, we find

$$\|\boldsymbol{x} - \boldsymbol{x}_t\|_{\boldsymbol{A}} \leq \left\|q_t(\boldsymbol{A}^{1/2}\boldsymbol{B}^{-1}\boldsymbol{A}^{1/2})\right\| \left\|(\boldsymbol{A}^{1/2}\boldsymbol{x})\right\| \leq \epsilon \|\boldsymbol{x}\|_{\boldsymbol{A}}.$$

The Preconditioned Conjugate Gradient (PCG) is a magical algorithm that after t steps (each of which involves solving a system in $\boldsymbol{B}$, multiplying a vector by $\boldsymbol{A}$, and performing a constant number of vector operations) produces the vector $\boldsymbol{x}_t$ that *minimizes*

$$\|\boldsymbol{x}_t - \boldsymbol{x}\|_{\boldsymbol{A}}$$

over all vectors $\boldsymbol{x}_t$ that can be written in the form $p_t(\boldsymbol{b})$ for a polynomial of degree at most t. That is, the algorithm finds the best possible solution among all iterative methods of the form we have described. We first bound the quality of PCG by saying that it is at least as good as Preconditioned Chebyshev, but it has the advantage of not needing to know α and β. We will then find an improved analysis.

36.5 Preconditioning by Trees

Vaidya [Vai90] had the remarkable idea of preconditioning the Laplacian matrix of a graph by the Laplacian matrix of a subgraph. If H is a subgraph of G, then

$$\boldsymbol{L}_H \preccurlyeq \boldsymbol{L}_G,$$

so all eigenvalues of $\boldsymbol{L}_H^{-1}\boldsymbol{L}_G$ are at least 1. Thus, we only need to find a subgraph H such that $\boldsymbol{L}_H$ is easy to invert and such that the largest eigenvalue of $\boldsymbol{L}_H^{-1}\boldsymbol{L}_G$ is not too big.

It is relatively easy to show that linear equations in the Laplacian matrices of trees can be solved exactly in linear time. One can either do this by finding an LU-factorization with a linear number of non-zeros, or by viewing the process of solving the linear equation as a dynamic program that passes up once from the leaves of the tree to a root, and then back down.

We will now show that a special type of tree, called a *low-stretch spanning tree* provides a very good preconditioner. To begin, let T be a spanning tree of G. Write

$$\boldsymbol{L}_G = \sum_{(u,v)\in E} w_{u,v} \boldsymbol{L}_{u,v} = \sum_{(u,v)\in E} w_{u,v} (\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)^T.$$

We will actually consider the trace of $\boldsymbol{L}_T^{-1}\boldsymbol{L}_G$. As the trace is linear, we have

$$\begin{aligned}
\operatorname{Tr}\left(\boldsymbol{L}_T^{-1}\boldsymbol{L}_G\right) &= \sum_{(u,v)\in E} w_{u,v} \operatorname{Tr}\left(\boldsymbol{L}_T^{-1}\boldsymbol{L}_{u,v}\right) \\
&= \sum_{(u,v)\in E} w_{u,v} \operatorname{Tr}\left(\boldsymbol{L}_T^{-1}(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)^T\right) \\
&= \sum_{(u,v)\in E} w_{u,v} \operatorname{Tr}\left((\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)^T \boldsymbol{L}_T^{-1}(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)\right) \\
&= \sum_{(u,v)\in E} w_{u,v} (\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)^T \boldsymbol{L}_T^{-1}(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v).
\end{aligned}$$

To evaluate this last term, we need to know the value of $(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)^T \boldsymbol{L}_T^{-1}(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)$. You already know something about it: it is the effective resistance in T between u and v. In a tree, this equals the distance in T between u and v, when we view the length of an edge as the reciprocal of its weight. This is because it is the resistance of a path of resistors in series. Let $T(u,v)$ denote the path in T from u to v, and let $w_1, \dots, w_k$ denote the weights of the edges on this path. As we view the weight of an edge as the reciprocal of its length,

$$(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)^T \boldsymbol{L}_T^{-1}(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v) = \sum_{i=1}^{k} \frac{1}{w_i}. \tag{36.1}$$

Even better, the term (36.1) is something that has been well-studied. It was defined by Alon, Karp, Peleg and West [AKPW95] to be the *stretch* of the unweighted edge (u,v) with respect to the tree T. Moreover, the *stretch* of the edge (u,v) with weight $w_{u,v}$ with respect to the tree T is defined to be exactly

$$w_{u,v} \sum_{i=1}^{k} \frac{1}{w_i},$$

where again $w_1, \dots, w_k$ are the weights on the edges of the unique path in T from u to v. A sequence of works, begining with [AKPW95], has shown that every graph G has a spanning tree in which the sum of the stretches of the edges is low. The best result so far is due to [AN12], who prove the following theorem.

Theorem 36.5.1. *Every weighted graph G has a spanning tree subgraph T such that the sum of the stretches of all edges of G with respect to T is at most*

$$O(m \log n \log\log n),$$

where m is the number of edges G. Moreover, one can compute this tree in time $O(m \log n \log\log n)$.

Thus, if we choose a low-stretch spanning tree T, we will ensure that

$$\operatorname{Tr}\left(\boldsymbol{L}_T^{-1}\boldsymbol{L}_G\right) = \sum_{(u,v)\in E} w_{u,v}(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v)^T \boldsymbol{L}_T^{-1}(\boldsymbol{\chi}_u - \boldsymbol{\chi}_v) \leq O(m \log n \log\log n).$$

In particular, this tells us that $\lambda_{max}(L_T^{-1}\boldsymbol{L}_G)$ is at most $O(m \log n \log\log n)$, and so the Preconditioned Conjugate Gradient will require at most $O(m^{1/2}\log n)$ iterations, each of which requires one multiplication by $\boldsymbol{L}_G$ and one linear solve in $\boldsymbol{L}_T$. This gives an algorithm that runs in time $O(m^{3/2}\log n \log 1/\epsilon)$, which is much lower than the $O(n^3)$ of Gaussian elimination when m, the number of edges in G, is small.

This result is due to Boman and Hendrickson [BH01].

36.6 Improving the Bound on the Running Time

We can show that the Preconditioned Conjugate Gradient will actually run in closer to $O(m^{1/3})$ iterations. Since the trace is the sum of the eigenvalues, we know that for every $\beta > 0$, $\boldsymbol{L}_T^{-1}\boldsymbol{L}_G$ has at most

$$\operatorname{Tr}\left(\boldsymbol{L}_T^{-1}\boldsymbol{L}_G\right)/\beta$$

eigenvalues that are larger than β.

To exploit this fact, we use the following lemma. It basically says that we can ignore the largest eigenvalues of $\boldsymbol{B}^{-1}\boldsymbol{A}$ if we are willing to spend one iteration for each.

Lemma 36.6.1. *Let $\lambda_1, \dots, \lambda_n$ be positive numbers such that all of them are at least α and at most k of them are more than β. Then, for every $t \geq k$, there exists a polynomial $p(X)$ of degree t such that $p(0) = 1$ and*

$$|p(\lambda_i)| \leq 2\left(1 + \frac{2}{\sqrt{\beta/\alpha}}\right)^{-(t-k)},$$

for all λ_i.

Proof. Let $r(X)$ be the polynomial we constructed using Chebyshev polynomials of degree $t-k$ for which

$$|r(X)| \leq 2\left(1 + \frac{2}{\sqrt{\beta/\alpha}}\right)^{-(t-k)},$$

for all X between α and β. Now, set

$$p(X) = r(X)\prod_{i:\lambda_i>\beta}(1 - X/\lambda_i).$$

This new polynomial is zero at every λ_i greater than β, and for X between α and β

$$|p(X)| = |r(X)|\prod_{i:\lambda_i>\beta}|(1 - X/\lambda_i)| \leq |r(X)|,$$

as we always have $X < \lambda_i$ in the product. □

Applying this lemma to the analysis of the Preconditioned Conjugate Gradient, with $\beta = \mathrm{Tr}\left(\boldsymbol{L}_T^{-1}\boldsymbol{L}_G\right)^{2/3}$ and $k = \mathrm{Tr}\left(\boldsymbol{L}_T^{-1}\boldsymbol{L}_G\right)^{1/3}$, we find that the algorithm produces ϵ-approximate solutions within

$$O(\mathrm{Tr}\left(\boldsymbol{L}_T^{-1}\boldsymbol{L}_G\right)^{1/3}\ln(1/\epsilon)) = O(m^{1/3}\log n \ln 1/\epsilon)$$

iterations.

This result is due to Spielman and Woo [SW09].

36.7 Further Improvements

We now have three families of algorithms for solving systems of equations in Laplaican matrices in nearly-linear time.

- By subgraph preconditioners. These basically work by adding back edges to the low-stretch trees. The resulting systems can no longer be solved directly in linear time. Instead, we use Gaussian elimination to eliminate the degree 1 and 2 vertices to reduce to a smaller system, and then solve that system recursively. The first nearly linear time algorithm of this form ran in time $O(m \log^c n \log 1/\epsilon)$, for some constant c [ST14]. An approach of this form was first made practical (and much simpler) by Koutis, Miller, and Peng [KMP11]. The asymptotically fastest method also works this way. It runs in time $O(m \log^{1/2} m \log^c \log n \log 1/\epsilon)$, [CKM+14] (Cohen, Kyng, Miller, Pachocki, Peng, Rao, Xu).
- By sparsification (see my notes from Lecture 19 from 2015). These algorithms work rather differently, and do not exploit low-stretch spanning trees. They appear in the papers [PS14, KLP+16].
- Accelerating Gaussian elimination by random sampling, by Kyng and Sachdeva [KS16]. This is the most elegant of the algorithms. While the running time of the algorithms, $O(m \log^2 n \log 1/\epsilon)$ is not the asymptotically best, the algorithm is so simple that it is the best in practice. An optimized implementation appears in the package `Laplacian.jl`.

There are other algorithms that are often fast in practice, but for which we have no theoretical analysis. I suggest the Algebraic Multigrid of Livne and Brandt, and the Combinatorial Multigrid of Yiannis Koutis.

36.8 Questions

I conjecture that it is possible to construct spanning trees of even lower stretch. Does every graph have a spanning tree of average stretch $2 \log_2 n$? I do not see any reason this should not be true. I also believe that this should be achievable by a practical algorithm. The best code that I know for computing low-stretch spanning trees, and which I implemented in `Laplacians.jl`, is a heuristic based on the algorithm of Alon, Karp, Peleg and West. However, I do not know an analysis of their algorithm that gives stretch better than $O(m 2^{\sqrt{\log n}})$. The theoretically better low-stretch trees of Abraham and Neiman are obtained by improving constructions of [EEST08, ABN08]. However, they seem too complicated to be practical.

The eigenvalues of $\boldsymbol{L}_H^{-1}\boldsymbol{L}_G$ are called generalized eigenvalues. The relation between generalized eigenvalues and stretch is the first result of which I am aware that establishes a combinatorial interpretation of generalized eigenvalues. Can you find any others?

Chapter 37

Augmented Spanning Tree Preconditioners

under construction

The first algorithms that solved Laplacian systems in nearly linear time used augmented spanning tree preconditioners. These are formed by adding edges of G back to a spanning tree of G. Vaidya [Vai90] first suggested doing this with maximum spanning trees. The first nearly linear time solvers were developed by Spielman and Teng [ST14] by augmenting low stretch spanning trees. The elegant algorithm described in this chapter is from two papers by Koutis, Miller, and Peng [KMP10, KMP11]. It solves systems to ϵ accuracy in time $\widetilde{O}(m \log n \log \epsilon^{-1})$.

Using the Iterative Refinement algorithm from the previous chapter, we know that it suffices to show this with any constant $\epsilon < 1$. You should assume throughout this chapter that ϵ is some absolute constant like 1/20.

I recall that $\widetilde{O}$ is like O-notation, but it hides low order logarithmic terms. That is, when we write $f(n) \leq \widetilde{O}(g(n))$, we mean that there is a constant c such that $f(n) \leq O(g(n) \log^c g(n))$. For example, in this notation we can say that every graph G has a spanning tree T of average stretch $\widetilde{O}(\log n)$. In this Chapter we will want to specify that many statements are true given some choice of constants c. For this purpose, we will often let c be a constant, but not the same constant, where it appears throughout the chapter. We do this instead of using O-notation, as it simplifies making the constants explicit later.

37.1 Recursion

Let H be obtained by adding a few edges back to a spanning tree T of G. As a large fraction of the vertices of T will have degree 1 or 2, the same is true of H. We can eliminate these degree 1 and 2 vertices to obtain a Schur complement $\widetilde{H}$ and an upper triangular matrix[1] $\boldsymbol{U}$ such that

$$\boldsymbol{U}^T \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}_{\widetilde{H}} \end{pmatrix} \boldsymbol{U} = \boldsymbol{L}_H.$$

This means that we can solve a system of equations in $\boldsymbol{L}_H$ by solving systems in $\boldsymbol{U}^T$, $\boldsymbol{L}_{\widetilde{H}}$, and $\boldsymbol{U}$. As elimination of a degree 1 vertex only decreases the degree of its neighbor and the elimination of a degree 2 vertex does not change the degrees of its neighbors, the matrix $\boldsymbol{U}$ has at most $2n$ nonzero entries. As $\boldsymbol{U}$ is upper triangular, systems in $\boldsymbol{U}$ and $\boldsymbol{U}^T$ can be solved in time proportional to their number of nonzero entries, $O(n)$. This inspires a recursive algorithm for solving equations in $\boldsymbol{L}_G$: we construct a good preconditioner H with many degree 1 and 2 vertices, and then solve systems in $\boldsymbol{L}_H$ by approximately solving in $\boldsymbol{L}_{\widetilde{H}}$.

We now explore this idea in a little more detail. First observe that because we are applying a recursive algorithm, we will not solve systems in $\boldsymbol{L}_{\widetilde{H}}$ exactly. Rather, we will be applying an algorithm to approximately solve these systems. The one guarantee we make about this algorithm is that it acts as a linear operator. That is, the action of this algorithm corresponds to multiplication by some matrix $\boldsymbol{Z}$ that we never construct. But, we know that for some ϵ

$$(1-\epsilon)\boldsymbol{Z}^+ \preccurlyeq \boldsymbol{L}_{\widetilde{H}} \preccurlyeq (1+\epsilon)\boldsymbol{Z}^+.$$

This immediately implies that

$$(1-\epsilon)\boldsymbol{U}^T \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}^+ \end{pmatrix} \boldsymbol{U} \preccurlyeq \boldsymbol{U}^T \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L}_{\widetilde{H}} \end{pmatrix} \boldsymbol{U} \preccurlyeq (1+\epsilon)\boldsymbol{U}^T \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}^+ \end{pmatrix} \boldsymbol{U}.$$

Thus, we can obtain ϵ-approximate solutions to systems in $\boldsymbol{L}_H$ by solving a system in $\boldsymbol{U}$, applying $\boldsymbol{Z}$, and solving a system in $\boldsymbol{U}^T$.

Define

$$\boldsymbol{M} = \boldsymbol{U}^T \begin{pmatrix} \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{Z}^+ \end{pmatrix} \boldsymbol{U}.$$

This will imply that $\kappa(\boldsymbol{L}_G, \boldsymbol{M})$ is at most $((1+\epsilon)/(1-\epsilon))\kappa(\boldsymbol{L}_G, \boldsymbol{L}_H)$, which will be just a little more than $\kappa(\boldsymbol{L}_G, \boldsymbol{L}_H)$ and thus fine for our purposes.

Lemma 37.1.1. *Let T be a tree on n vertices. Then, more than half the vertices of T have degree 1 or 2.*

Proof. The number of edges in T is $n-1$, so the average degree of vertices in T is less than 2. Thus T must contain at least one degree 1 vertex for every vertex of degree at least 3. The other vertices have degree 2. □

We learned last lecture that if we keep eliminating degree 1 vertices from trees, then we will eventually eliminate all the vertices. An analogous fast is true for a graph that equals a tree plus k edges.

[1] Whether this matrix is actually upper triangular depends on the ordering of the vertices. We assume, without loss of generality, that the vertices are ordered so that the matrix is upper triangular.

Lemma 37.1.2. *Let H be a tree on n vertices plus k edges. If we eliminate degree 1 and 2 vertices of the tree that do not touch the extra k edges until none remain, we will be left with at most $4k$ vertices and $5k$ edges.*

Proof. If we eliminate a degree 1 or 2 vertex of the tree that does not touch one of the extra k edges, we will obtain a graph that looks like a tree on one fewer vertex, plus k edges. As a tree on $4k$ vertices must have at least $2k+1$ vertices of degree 1 or 2, at least one of these does not touch one of the extra k edges, and so can be eliminated. □

37.2 Heavy Trees

Koutis, Miller, and Peng observe that we do not necessarily have to produce a subgraph H of G that looks like a tree plus a few edges. All we really need is for H to have many fewer edges than G. This still leaves the question of how we will find such an H that is a good approximation of G. The trick is to use a variant of the random-sampling based approach of Chapter 32. But, we avoid the cost of computing effective resistances of edges by estimating them by their stretches, at the cost of a worse approximation.

We begin by formally stating the result of that chapter for graphs.

Theorem 37.2.1. *Let $G = (V, E, w)$ be a graph, let $\epsilon > 0$, and for every edge (a, b) and let $p_{a,b} \in (0, 1]$ satisfy*

$$p_{a,b} \geq \min\left(1, \frac{4\ln n}{\epsilon^2} w_{a,b} \mathrm{Reff}_G(a, b)\right).$$

Form the random graph $H = (V, F, u)$ by setting for every edge independently

$$u_{a,b} = \begin{cases} w_{a,b}/p_{a,b} & \text{with probability } p_{a,b} \\ 0 & \text{with probability } 1 - p_{a,b}. \end{cases}$$

Then there exists a constant c so that with probability at least $1 - n^{-c}$, H is an ϵ approximation of G and the number of edges in H is at most $2\sum_{a,b} p_{a,b}$.

Lemma 37.2.2. *For every weighted graph $G = (V, E, w)$, spanning tree $T = (V, F, w)$ of G, and $a, b \in V$, $\mathrm{Reff}_G(a, b) \leq \mathrm{Stretch}_T(a, b)$.*

Proof. Rayleigh's Monotonicity Theorem tells us that $\mathrm{Reff}_G(a, b) \leq \mathrm{Reff}_T(a, b)$, and this latter term equals $\mathrm{Stretch}_T(a, b)$. □

The problem with sampling edges with probability proportional to their effective resistance, or stretches, is that this will produce too many edges. Koutis, Miller, and Peng solve this problem by multiplicatively increasing the weights of the edges in a low-stretch spanning tree of G. Define

$$\widetilde{G} = G + (s-1)T.$$

That is, $\widetilde{G}$ is the same as G, but every edge in the tree T has its weight multiplied by s.

Thus, for (a, b) not in the tree,

$$\mathrm{Reff}_{\widetilde{G}}(a, b) \leq \mathrm{Reff}_{sT}(a, b) \leq (1/s)\mathrm{Stretch}_T(a, b).$$

For every edge $(a, b) in T$ we set $p_{a,b} = 1$ and for every edge $(a, b) \notin T$, we set

$$p_{a,b} \geq \min\left(1, \frac{4 \ln n}{s\epsilon^2} w_{a,b} \mathrm{Stretch}_T(a, b)\right).$$

Define σ to be the average stretch of edges of G with respect to T:

$$\sigma = (1/m)\mathrm{Stretch}_T(G) = (1/m) \sum_{(a,b)\in E} w_{a,b}\mathrm{Stretch}_T(a, b),$$

and recall that $\sigma \leq \widetilde{O}(\log n)$. If we now form H by including edge (a, b) with probability $p_{a,b}$, then Theorem 37.2.1 tells us that with high probability H is an ϵ approximation of $\widetilde{G}$ and that the number of edges of H that are not in T is at most

$$\sum_{(a,b)\notin T} p_{a,b} \leq \frac{4m\sigma \ln n}{s\epsilon^2}.$$

So, by making s a little more than some constant times $\sigma \ln n$, we can make sure that the number of edges of H not in T is less than the number of edges of G not in T.

But, we need to solve systems in G, not $\widetilde{G}$. To this end, we use the following multiplicative property of condition numbers.

Claim 37.2.3.

$$\kappa(\boldsymbol{L}_G, \boldsymbol{L}_H) \leq \kappa(\boldsymbol{L}_G, \boldsymbol{L}_{\widetilde{G}})\kappa(\boldsymbol{L}_{\widetilde{G}}, \boldsymbol{L}_H).$$

As $\widetilde{G}$ differs from G by having the weights of some edges multiplied by s, $\kappa(\boldsymbol{L}_G, \boldsymbol{L}_{\widetilde{G}}) \leq s$. Thus, we will have $\kappa(\boldsymbol{L}_G, \boldsymbol{L}_H) \leq s(1 + \epsilon)/(1 - \epsilon)$, and to get ϵ accurate solutions to systems in $\boldsymbol{L}_G$ we will need to solve some constant times $\kappa(\boldsymbol{L}_G, \boldsymbol{L}_H)^{1/2}$ systems in $\boldsymbol{L}_H$. As we are going to keep ϵ constant, this will be around $s^{1/2}$.

To make an efficient algorithm for solving systems in G out of an algorithm for solving systems in H, it would be easiest if the cost of the solves in H is less than the cost of a multiply by G. As we will solve the system in H around $s^{1/2}$ times, it seems natural to ensure that the number of edges of H that are not in T is at most the number of edges in G divide by $s^{1/2}$. That is, we want

$$s^{1/2}\frac{4m\sigma \ln n}{s\epsilon^2} \leq m,$$

which requires

$$s \geq c(\sigma \ln n)^2,$$

for some constant c. We will now show that such a choice of c yields an algorithm for solving linear equations in $\boldsymbol{L}_G$ to constant accuracy in time $\widetilde{O}(m \log^2 n)$.

We now describe the recursion. Let $G_0 = G$, the input graph. We will eventually solve systems in G_i by recursively solving systems in G_{i+1}. Each system G_{i+1} will have fewer edges than G_i, and thus we can use a brute force solve when the system becomes small enough. We will bound the running time of solvers for systems in G_i in terms of the number of edges that are *not* in their spanning trees. We denote this by $o_i = m_i - (n_i - 1)$. There is some issue with o_0, so let's assume without much loss of generality that G_0 does not have any degree 1 or 2 vertices, and thus the $o_0 \geq n_0$.

Form $\widetilde{G}_i$ by multiplying a low-stretch spanning tree of G by s, and use random sampling to produce H_i. We know that the number of off-tree edges in H_i is at most a $1/(c\sigma \ln n)$ fraction of the number of off-tree edges in G_i. If the number of off-tree edges in H_i is less than $n_i/4$, then we know that after eliminating degree 1 and 2 vertices we will be left with a graph having at most $4n_i$ vertices and $5n_i$ edges. We let G_{i+1} be this graph. If this number off of-tree edges in H_i is more than $n_i/4$, then we just set $G_{i+1} = H_i$.

In this way, we ensure that $o_{i+1} \leq o_i/(c\sigma \ln n)$. We can now prove by backwards induction on i that the time required to solve systems of equations in $\boldsymbol{L}_{G_i}$ is at most $O(o_i \sigma \ln n)$. A solve in G_i to constant accuracy requires performing $O(s^{1/2})$ solves in G_{i+1} and as many multiplies by $\boldsymbol{L}_{G_i}$. By induction we know that this takes time at most

$$O(s^{1/2}\,(o_i + o_{i-1}\sigma \ln n)) \leq O(s^{1/2}\,(2o_i)) \leq O(o_i \sigma \ln n).$$

37.3 Saving a log

Chapter 38

Fast Laplacian Solvers by Sparsification

Lecture 19 from November 9, 2015

38.1 Overview

We will see how sparsification allows us to solve systems of linear equations in Laplacian matrices and their sub-matrices in nearly linear time. By "nearly-linear", I mean time $O(m \log^c(n\kappa^{-1}) \log \epsilon^{-1})$ for systems with m nonzero entries, n dimensions, condition number κ. and accuracy ϵ.

This algorithm comes from [PS14].

38.2 Today's notion of approximation

In today's lecture, I will find it convenient to define matrix approximations slightly differently from previous lectures. Today, I define $\boldsymbol{A} \approx_\epsilon \boldsymbol{B}$ to mean

$$e^{-\epsilon}\boldsymbol{A} \preccurlyeq \boldsymbol{B} \preccurlyeq e^{\epsilon}\boldsymbol{A}.$$

Note that this relation is symmetric in $\boldsymbol{A}$ and $\boldsymbol{B}$, and that for ϵ small $e^\epsilon \approx 1 + \epsilon$.

The advantage of this definition is that

$$\boldsymbol{A} \approx_\alpha \boldsymbol{B} \quad \text{and} \quad \boldsymbol{B} \approx_\beta \boldsymbol{C} \quad \text{implies} \quad \boldsymbol{A} \approx_{\alpha+\beta} \boldsymbol{C}.$$

38.3 The Idea

I begin by describing the idea behind the algorithm. This idea won't quite work. But, we will see how to turn it into one that does.

We will work with matrices that look like $\boldsymbol{M} = \boldsymbol{L} + \boldsymbol{X}$ where $\boldsymbol{L}$ is a Laplacian and $\boldsymbol{X}$ is a non-zero, non-negative diagonal matrix. Such matrices are called M-matrices. A *symmetric M-matrix* is a matrix $\boldsymbol{M}$ with nonpositive off-diagonal entries such that $\boldsymbol{M}\mathbf{1}$ is nonnegative and nonzero. We have encountered M-matrices before without naming them. If $G = (V, E)$ is a graph, $S \subset V$, and $G(S)$ is connected, then the submatrix of $\boldsymbol{L}_G$ indexed by rows and columns in S is an M-matrix. Algorithmically, the problems of solving systems of equations in Laplacians and symmetric M-matrices are equivalent.

The sparsification results that we learned for Laplacians translate over to M-matrices. Every M-matrix $\boldsymbol{M}$ can be written in the form $\boldsymbol{X} + \boldsymbol{L}$ where $\boldsymbol{L}$ is a Laplacian and $\boldsymbol{X}$ is a nonnegative diagonal matrix. If $\widehat{\boldsymbol{L}} \approx_\epsilon \boldsymbol{L}$, then it is easy to show (too easy for homework) that

$$\boldsymbol{X} + \widehat{\boldsymbol{L}} \approx_\epsilon \boldsymbol{X} + \boldsymbol{L}.$$

In Lecture 7, Lemma 7.3.1, we proved that if $\boldsymbol{X}$ has at least one nonzero entry and if $\boldsymbol{L}$ is connected, then $\boldsymbol{X} + \boldsymbol{L}$ is nonsingular. We write such a matrix in the form $\boldsymbol{M} = \boldsymbol{D} - \boldsymbol{A}$ where $\boldsymbol{D}$ is positive diagonal and $\boldsymbol{A}$ is nonnegative, and note that its being nonsingular and positive semidefinite implies

$$\boldsymbol{D} - \boldsymbol{A} \succ 0 \quad \iff \quad \boldsymbol{D} \succ \boldsymbol{A}. \tag{38.1}$$

Using the Perron-Frobenius theorem, one can also show that

$$\boldsymbol{D} \succ -\boldsymbol{A}. \tag{38.2}$$

Multiplying $\boldsymbol{M}$ by $\boldsymbol{D}^{-1/2}$ on either side, we obtain

$$\boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2}.$$

Define

$$\boldsymbol{B} = \boldsymbol{D}^{-1/2}\boldsymbol{A}\boldsymbol{D}^{-1/2},$$

and note that inequalities (38.1) and (38.2) imply that all eigenvalues of $\boldsymbol{B}$ have absolute value strictly less than 1.

It suffices to figure out how to solve systems of equations in $\boldsymbol{I} - \boldsymbol{B}$. One way to do this is to exploit the power series expansion:

$$(\boldsymbol{I} - \boldsymbol{B})^{-1} = \boldsymbol{I} + \boldsymbol{B} + \boldsymbol{B}^2 + \boldsymbol{B}^3 + \cdots$$

However, this series might need many terms to converge. We can figure out how many. If the largest eigenvalue of $\boldsymbol{B}$ is $(1 - \kappa) < 1$, then we need at least $1/\kappa$ terms.

We can write a series with fewer terms if we express it as a product instead of as a sum:

$$\sum_{i \geq 0} \boldsymbol{B}^i = \prod_{j \geq 1} (\boldsymbol{I} + \boldsymbol{B}^{2^j}).$$

To see why this works, look at the first few terms

$$(\boldsymbol{I}+\boldsymbol{B})(\boldsymbol{I}+\boldsymbol{B}^2)(\boldsymbol{I}+\boldsymbol{B}^4) = (\boldsymbol{I}+\boldsymbol{B}+\boldsymbol{B}^2+\boldsymbol{B}^3)(\boldsymbol{I}+\boldsymbol{B}^4) = (\boldsymbol{I}+\boldsymbol{B}+\boldsymbol{B}^2+\boldsymbol{B}^3)+\boldsymbol{B}^4(\boldsymbol{I}+\boldsymbol{B}+\boldsymbol{B}^2+\boldsymbol{B}^3).$$

We only need $O(\log \kappa^{-1})$ terms of this product to obtain a good approximation of $(\boldsymbol{I}-\boldsymbol{B})^{-1}$. The obstacle to quickly applying a series like this is that the matrices $\boldsymbol{I}+\boldsymbol{B}^{2^j}$ are probably dense. We know how to solve this problem: we can sparsify them! I'm not saying that flippantly. We actually do know how to sparsify matrices of this form.

But, simply sparsifying the matrices $\boldsymbol{I}+\boldsymbol{B}^{2^j}$ does not solve our problem because approximation is not preserved by products. That is, even if $\boldsymbol{A} \approx_\epsilon \widehat{\boldsymbol{A}}$ and $\boldsymbol{B} \approx_\epsilon \widehat{\boldsymbol{B}}$, $\widehat{\boldsymbol{A}}\widehat{\boldsymbol{B}}$ could be a very poor approximation of $\boldsymbol{A}\boldsymbol{B}$. In fact, since the product $\widehat{\boldsymbol{A}}\widehat{\boldsymbol{B}}$ is not necessarily symmetric, we haven't even defined what it would mean for it to approximate $\boldsymbol{A}\boldsymbol{B}$.

38.4 A symmetric expansion

We will now derive a way of expanding $(\boldsymbol{I}-\boldsymbol{B})^{-1}$ that is amenable to approximation. We begin with an alternate derivation of the series we saw before. Note that

$$(\boldsymbol{I}-\boldsymbol{B})(\boldsymbol{I}+\boldsymbol{B}) = (\boldsymbol{I}-\boldsymbol{B}^2),$$

and so

$$(\boldsymbol{I}-\boldsymbol{B}) = (\boldsymbol{I}-\boldsymbol{B}^2)(\boldsymbol{I}+\boldsymbol{B})^{-1}.$$

Taking the inverse of both sides gives

$$(\boldsymbol{I}-\boldsymbol{B})^{-1} = (\boldsymbol{I}+\boldsymbol{B})(\boldsymbol{I}-\boldsymbol{B}^2)^{-1}.$$

We can then apply the same expansion to $(\boldsymbol{I}-\boldsymbol{B}^2)^{-1}$ to obtain

$$(\boldsymbol{I}-\boldsymbol{B})^{-1} = (\boldsymbol{I}+\boldsymbol{B})(\boldsymbol{I}+\boldsymbol{B}^2)(\boldsymbol{I}-\boldsymbol{B}^4)^{-1}.$$

What we need is a symmetric expansion. We use

$$(\boldsymbol{I}-\boldsymbol{B})^{-1} = \frac{1}{2}\boldsymbol{I} + \frac{1}{2}(\boldsymbol{I}+\boldsymbol{B})(\boldsymbol{I}-\boldsymbol{B}^2)^{-1}(\boldsymbol{I}+\boldsymbol{B}). \tag{38.3}$$

We will verify this by multiplying the right hand side by $(\boldsymbol{I}-\boldsymbol{B})$:

$$(\boldsymbol{I}+\boldsymbol{B})(\boldsymbol{I}-\boldsymbol{B}^2)^{-1}(\boldsymbol{I}+\boldsymbol{B})(\boldsymbol{I}-\boldsymbol{B}) = (\boldsymbol{I}+\boldsymbol{B})(\boldsymbol{I}-\boldsymbol{B}^2)^{-1}(\boldsymbol{I}-\boldsymbol{B}^2) = \boldsymbol{I}+\boldsymbol{B};$$

so

$$\frac{1}{2}\left[\boldsymbol{I}+(\boldsymbol{I}+\boldsymbol{B})(\boldsymbol{I}-\boldsymbol{B}^2)^{-1}(\boldsymbol{I}+\boldsymbol{B})\right](\boldsymbol{I}-\boldsymbol{B}) = \frac{1}{2}\left[(\boldsymbol{I}-\boldsymbol{B})+(\boldsymbol{I}+\boldsymbol{B})\right] = \boldsymbol{I}.$$

This expression for $(\boldsymbol{I}-\boldsymbol{B})^{-1}$ plays nicely with matrix approximations. If

$$\boldsymbol{M}_1 \approx_\epsilon (\boldsymbol{I}-\boldsymbol{B}^2),$$

then you can show

$$(\boldsymbol{I}-\boldsymbol{B})^{-1} \approx_\epsilon \frac{1}{2}\left[\boldsymbol{I}+(\boldsymbol{I}+\boldsymbol{B})\boldsymbol{M}_1^{-1}(\boldsymbol{I}+\boldsymbol{B})\right].$$

If we can apply $\boldsymbol{M}_1^{-1}$ quickly and if $\boldsymbol{B}$ is sparse, then we can quickly approximate $(\boldsymbol{I}-\boldsymbol{B})^{-1}$. You may now be wondering how we will construct such an $\boldsymbol{M}_1$. The answer, in short, is "recursively".

38.5 $\boldsymbol{D}$ and $\boldsymbol{A}$

Unfortunately, we are going to need to stop writing matrices in terms of $\boldsymbol{I}$ and $\boldsymbol{B}$, and return to writing them in terms of $\boldsymbol{D}$ and $\boldsymbol{A}$. The reason this is unfortunate is that it makes for longer expressions.

The analog of (38.3) is

$$(\boldsymbol{D}-\boldsymbol{A})^{-1} = \frac{1}{2}\left[\boldsymbol{D}^{-1} + (\boldsymbol{I}+\boldsymbol{D}^{-1}\boldsymbol{A})(\boldsymbol{D}-\boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{A})^{-1}(\boldsymbol{I}+\boldsymbol{A}\boldsymbol{D}^{-1})\right]. \tag{38.4}$$

In order to be able to work with this expression inductively, we need to check that the middle matrix is an M-matrix.

Lemma 38.5.1. *If $\boldsymbol{D}$ is a diagonal matrix and $\boldsymbol{A}$ is a nonnegative matrix so that $\boldsymbol{M} = \boldsymbol{D}-\boldsymbol{A}$ is an M-matrix, then*

$$\boldsymbol{M}_1 = \boldsymbol{D} - \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{A}$$

is also an M-matrix.

Proof. As the off-diagonal entries of this matrix are symmetric and nonpositive, it suffices to prove that $\boldsymbol{M}\boldsymbol{1} \geq 0$ and $\boldsymbol{M}\boldsymbol{1} \neq \boldsymbol{0}$. To compute the row sums set

$$\boldsymbol{d} = \boldsymbol{D}\boldsymbol{1} \quad \text{and} \quad \boldsymbol{a} = \boldsymbol{A}\boldsymbol{1},$$

and note that $\boldsymbol{d}-\boldsymbol{a} \geq 0$ and $\boldsymbol{d}-\boldsymbol{a} \neq \boldsymbol{0}$. For $\boldsymbol{M}_1$, we have

$$(\boldsymbol{D} - \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{A})\boldsymbol{1} = \boldsymbol{d} - \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{a} \geq \boldsymbol{d} - \boldsymbol{A}\boldsymbol{1} = \boldsymbol{d}-\boldsymbol{a},$$

which is nonnegative and not exactly zero. □

We will apply transformation like this many times during our algorithm. To keep track of progress, I say that $(\boldsymbol{D}, \boldsymbol{A})$ is an (α, β)-pair if

a. $\boldsymbol{D}$ is positive diagonal,

b. $\boldsymbol{A}$ is nonnegative (and can have diagonal entries), and

c. $\alpha\boldsymbol{D} \succcurlyeq \boldsymbol{A}$ and $\beta\boldsymbol{D} \succcurlyeq -\boldsymbol{A}$.

For our initial matrix $\boldsymbol{M} = \boldsymbol{D}-\boldsymbol{A}$, we know that there is some number $\kappa > 0$ for which $(\boldsymbol{D}, \boldsymbol{A})$ is a $(1-\kappa, 1-\kappa)$-pair.

At the end of our recursion we will seek a $(1/4, 1/4)$-pair. When we have such a pair, we can just approximate $\boldsymbol{D}-\boldsymbol{A}$ by $\boldsymbol{D}$.

Lemma 38.5.2. *If $\boldsymbol{M} = \boldsymbol{D}-\boldsymbol{A}$ and $(\boldsymbol{D}, \boldsymbol{A})$ is a $(1/4, 1/4)$-pair, then*

$$\boldsymbol{M} \approx_{1/3} \boldsymbol{D}.$$

Proof. We have

$$\boldsymbol{M} = \boldsymbol{D} - \boldsymbol{A} \preccurlyeq (1 + 1/4)\boldsymbol{D} \leq e^{1/4}\boldsymbol{D},$$

and

$$\boldsymbol{M} = \boldsymbol{D} - \boldsymbol{A} \succcurlyeq \boldsymbol{D} - (1/4)\boldsymbol{D} = (3/4)\boldsymbol{D} \succcurlyeq e^{-1/3}\boldsymbol{D}.$$

□

Lemma 38.5.3. *If* $(\boldsymbol{D}, \boldsymbol{A})$ *is an* (α, α)*-pair, then* $(\boldsymbol{D}, \boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{A})$ *is an* $(\alpha^2, 0)$*-pair.*

Proof. From Lecture 14, Lemma 3.1, we know that the condition of the lemma is equivalent to the assertion that all eigenvalues of $\boldsymbol{D}^{-1}\boldsymbol{A}$ have absolute value at most α, and that the conclusion is equivalent to the assertion that all eigenvalues of $\boldsymbol{D}^{-1}\boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{A}$ lie between 0 and α^2, which is immediate as they are the squares of the eigenvalues of $\boldsymbol{D}^{-1}\boldsymbol{A}$. □

So, if we start with matrices $\boldsymbol{D}$ and $\boldsymbol{A}$ that are a $(1 - \kappa, 1 - \kappa)$-pair, then after applying this transformation approximately $\log \kappa^{-1} + 2$ times we obtain a $(1/4, 0)$-pair. But, the matrices in this pair could be dense. To keep them sparse, we need to figure out how approximating $\boldsymbol{D} - \boldsymbol{A}$ degrades its quality.

Lemma 38.5.4. *If* $\epsilon \leq 1/3$*,*

a. $(\boldsymbol{D}, \boldsymbol{A})$ *is a* $(1 - \kappa, 0)$ *pair,*

b. $\boldsymbol{D} - \boldsymbol{A} \approx_\epsilon \widehat{\boldsymbol{D}} - \widehat{\boldsymbol{A}}$*, and*

c. $\boldsymbol{D} \approx_\epsilon \widehat{\boldsymbol{D}}$*,*

then $\widehat{\boldsymbol{D}} - \widehat{\boldsymbol{A}}$ *is an* $(1 - \kappa e^{-2\epsilon}, 3\epsilon)$*-pair.*

Proof. First observe that

$$(1 - \kappa)\boldsymbol{D} \succcurlyeq \boldsymbol{A} \iff \boldsymbol{D} - \boldsymbol{A} \succcurlyeq \kappa \boldsymbol{D}.$$

Then, compute

$$\widehat{\boldsymbol{D}} - \widehat{\boldsymbol{A}} \succcurlyeq e^{-\epsilon}(\boldsymbol{D} - \boldsymbol{A}) \succcurlyeq e^{-\epsilon}\kappa\boldsymbol{D} \succcurlyeq e^{-2\epsilon}\kappa\widehat{\boldsymbol{D}}.$$

For the other side, compute

$$e^{2\epsilon}\widehat{\boldsymbol{D}} \succcurlyeq e^{\epsilon}\boldsymbol{D} \succcurlyeq e^{\epsilon}(\boldsymbol{D} - \boldsymbol{A}) \succcurlyeq (\widehat{\boldsymbol{D}} - \widehat{\boldsymbol{A}}).$$

For $\epsilon \leq 1/3$, $3\epsilon \geq e^{2\epsilon} - 1$, so

$$3\epsilon\widehat{\boldsymbol{D}} \succcurlyeq (e^{2\epsilon} - 1)\widehat{\boldsymbol{D}} \succcurlyeq -\widehat{\boldsymbol{A}}.$$

□

It remains to confirm that sparsification satisfies the requirements of this lemma. The reason this might not be obvious is that we allow $\boldsymbol{A}$ to have nonnegative diagonal elements. While this does not interfere with condition b, you might be concerned that it would interfere with condition c. It need not.

Let $\boldsymbol{C}$ be the diagonal of $\boldsymbol{A}$, and let $\boldsymbol{L}$ be the Laplacian of the graph with adjacency matrix $\boldsymbol{A} - \boldsymbol{C}$, and set $\boldsymbol{X}$ so that $\boldsymbol{X} + \boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$. Let $\widetilde{\boldsymbol{L}}$ be a sparse ϵ-approximation of $\boldsymbol{L}$. By computing the quadratic form in elementary unit vectors, you can check that the diagonals of $\boldsymbol{L}$ and $\widetilde{\boldsymbol{L}}$ approximate each other. If we now write $\widetilde{\boldsymbol{L}} = \widetilde{\boldsymbol{D}} - \widetilde{\boldsymbol{A}}$, where $\widetilde{\boldsymbol{A}}$ has zero diagonal, and set

$$\widehat{\boldsymbol{D}} = \widetilde{\boldsymbol{D}} + \boldsymbol{C} \quad \text{and} \quad \widehat{\boldsymbol{A}} = \widetilde{\boldsymbol{A}} + \boldsymbol{C}$$

You can now check that $\widehat{\boldsymbol{D}}$ and $\widehat{\boldsymbol{A}}$ satisfy the requirements of Lemma 38.5.4.

You might wonder why we bother to keep diagonal elements in a matrix like $\boldsymbol{A}$. It seems simpler to get rid of them. However, we want $(\boldsymbol{D}, \boldsymbol{A})$ to be an (α, β) pair, and removing subtracting $\boldsymbol{C}$ from both of them would make β worse. This might not matter too much as we have good control over β. But, I don't yet see a nice way to carry out a proof that exploits this.

38.6 Sketch of the construction

We begin with an M-matrix $\boldsymbol{M}_0 = \boldsymbol{D}_0 - \boldsymbol{A}_0$. Since this matrix is nonsingular, there is a $\kappa_0 > 0$ so that $(\boldsymbol{D}_0, \boldsymbol{A}_0)$ is a $(1 - \kappa_0, 1 - \kappa_0)$ pair.

We now know that the matrix

$$\boldsymbol{D}_0 - \boldsymbol{A}_0 \boldsymbol{D}_0^{-1} \boldsymbol{A}_0$$

is an M-matrix and that $(\boldsymbol{D}_0, \boldsymbol{A}_0 \boldsymbol{D}_0^{-1} \boldsymbol{A}_0)$ is a $((1-\kappa_0)^2, 0)$-pair. Define κ_1 so that $1 - \kappa_1 = (1-\kappa)_0^2$, and note that κ_1 is approximately $2\kappa_0$. Lemma 38.5.4 and the discussion following it tells us that there is a $(1 - \kappa_1 e^{-2\epsilon}, 3\epsilon)$-pair $(\boldsymbol{D}_1, \boldsymbol{A}_1)$ so that

$$\boldsymbol{D}_1 - \boldsymbol{A}_1 \approx_\epsilon \boldsymbol{D}_0 - \boldsymbol{A}_0 \boldsymbol{D}_0^{-1} \boldsymbol{A}_0$$

and so that $\boldsymbol{A}_1$ has $O(n/\epsilon^2)$ nonzero entries.

Continuing inductively for some number k steps, we find $(1 - \kappa_i, 3\epsilon)$ pairs $(\boldsymbol{D}_i, \boldsymbol{A}_i)$ so that

$$\boldsymbol{M}_i = \boldsymbol{D}_i - \boldsymbol{A}_i$$

has $O(n/\epsilon^2)$ nonzero entries, and

$$\boldsymbol{M}_i \approx_\epsilon \boldsymbol{D}_i - \boldsymbol{A}_{i-1} \boldsymbol{D}_{i-1}^{-1} \boldsymbol{A}_{i-1}.$$

For the i such that κ_i is small, κ_{i+1} is approximately twice κ_i. So, for $k = 2 + \log_2 1/\kappa$ and ϵ close to zero, we can guarantee that $(\boldsymbol{D}_k, \boldsymbol{A}_k)$ is a $(1/4, 1/4)$ pair.

We now see how this construction allows us to approximately solve systems of equations in $\boldsymbol{D}_0 - \boldsymbol{A}_0$, and how we must set ϵ for it to work. For every $0 \leq i < k$, we have

$$(\boldsymbol{D}_i - \boldsymbol{A}_i)^{-1} \frac{1}{2} \boldsymbol{D}_i^{-1} + \frac{1}{2} (\boldsymbol{I} + \boldsymbol{D}_i^{-1} \boldsymbol{A}_i)(\boldsymbol{D}_i - \boldsymbol{A}_i \boldsymbol{D}_i^{-1} \boldsymbol{A}_i)^{-1} (\boldsymbol{I} + \boldsymbol{A}_i \boldsymbol{D}_i^{-1}) \approx_\epsilon \frac{1}{2} \boldsymbol{D}_i^{-1} + \frac{1}{2} (\boldsymbol{I} + \boldsymbol{D}_i^{-1} \boldsymbol{A}_i)(\boldsymbol{D}_{i+1} - \boldsymbol{A}_{i+1})^{-1} (\boldsymbol{I} + \boldsymbol{A}_i$$

and

$$(\boldsymbol{D}_k - \boldsymbol{A}_k)^{-1} \approx_{1/3} \boldsymbol{D}_k^{-1}.$$

By substituting through each of these approximations, we obtain solutions to systems of equations in $\boldsymbol{D}_0 - \boldsymbol{A}_0$ with accuracy $1/3 + k\epsilon$. So, we should set $k\epsilon = 1/3$, and thus

$$\epsilon = 1/(2 + \log_2 \kappa^{-1}).$$

The dominant cost of the resulting algorithm will be the multiplication of vectors by $2k$ matrices of $O(n/\epsilon^2)$ entries, with a total cost of

$$O(n(\log_2(1/\kappa))^3).$$

38.7 Making the construction efficient

In the above construction, I just assumed that appropriate sparsifiers exist, rather than constructing them efficiently. To construct them efficiently, we need two ideas. The first is that we need to be able to quickly approximate effective resistances so that we can use the sampling algorithm from Lecture 17.

The second is to observe that we do not actually want to form the matrix $\boldsymbol{A}\boldsymbol{D}^{-1}\boldsymbol{A}$ before sparsifying it, as that could take too long. Instead, we express it as a product of cliques that have succinct descriptions, and we form the sum of approximations of each of those.

38.8 Improvements

The fastest known algorithms for solving systems of equations run in time $O(m\sqrt{\log n} \log \epsilon^{-1})$ [CKM+14]. The algorithm I have presented here can be substantially improved by combining it with Cholesky factorization. This both gives an efficient parallel algorithm, and proves the existence of an approximate inverse for every M-matrix that has a linear number of nonzeros [LPS15].

Chapter 39

Testing Isomorphism of Graphs with Distinct Eigenvalues

Lecture 8 from September 24, 2018

39.1 Introduction

I will present an algorithm of Leighton and Miller [LM82] for testing isomorphism of graphs in which all eigenvalues have multiplicity 1. This algorithm was never published, as the results were technically subsumed by those in a paper of Babai, Grigoriev and Mount [BGM82], which gave a polynomial time algorithm for testing isomorphism of graphs in which all eigenvalues have multiplicity bounded by a constant.

I present the weaker result in the interest of simplicity.

Testing isomorphism of graphs is a notorious problem. Until very recently, the fastest-known algorithm for it took time time $2^{\sqrt{O(n \log n)}}$ (See [Bab81, BL83, ZKT85]). Babai [Bab16] recently announced a breakthrough that reduces the complexity to $2^{(\log n)^{O(1)}}$.

However, testing graph isomorphism seems easy in almost all practical instances. Today's lecture and one next week will give you some idea as to why.

39.2 Graph Isomorphism

Recall that two graphs $G = (V, E)$ and $H = (V, F)$ are isomorphic if there exists a permutation π of V such that

$$(a, b) \in E \iff (\pi(a), \pi(b)) \in F.$$

Of course, we can express this relation in terms of matrices associated with the graphs. It doesn't matter much which matrices we use. So for this lecture we will use the adjacency matrices.

Every permutation may be realized by a *permutation matrix*. For the permutation π, this is the matrix $\mathbf{\Pi}$ with entries given by

$$\mathbf{\Pi}(a,b) = \begin{cases} 1 & \text{if } \pi(a) = b \\ 0 & \text{otherwise.} \end{cases}$$

For a vector $\boldsymbol{\psi}$, we see[1] that

$$\left(\mathbf{\Pi}\boldsymbol{\psi}\right)(a) = \boldsymbol{\psi}(\pi(a)).$$

Let A be the adjacency matrix of G and let B be the adjacency matrix of H. We see that G and H are isomorphic if and only if there exists a permutation matrix $\mathbf{\Pi}$ such that

$$\mathbf{\Pi} A \mathbf{\Pi}^T = B.$$

39.3 Using Eigenvalues and Eigenvectors

If G and H are isomorphic, then A and B must have the same eigenvalues. However, there are many pairs of graphs that are non-isomorphic but which have the same eigenvalues. We will see some tricky ones next lecture. But, for now, we note that if A and B have different eigenvalues, then we know that the corresponding graphs are non-isomorphic, and we don't have to worry about them.

For the rest of this lecture, we will assume that A and B have the same eigenvalues, and that each of these eigenvalues has multiplicity 1. We will begin our study of this situation by considering some cases in which testing isomorphism is easy.

Recall that we can write

$$A = \boldsymbol{\Psi} \Lambda \boldsymbol{\Psi}^T,$$

where Λ is the diagonal matrix of eigenvalues of A and $\boldsymbol{\Psi}$ is an orthonormal matrix holding its eigenvectors. If B has the same eigenvalues, we can write

$$B = \boldsymbol{\Phi} \Lambda \boldsymbol{\Phi}^T.$$

If $\mathbf{\Pi}$ is the matrix of an isomorphism from G to H, then

$$\mathbf{\Pi} \boldsymbol{\Psi} \Lambda \boldsymbol{\Psi}^T \mathbf{\Pi}^T = \boldsymbol{\Phi} \Lambda \boldsymbol{\Phi}^T.$$

As each entry of Λ is distinct, this looks like it would imply $\mathbf{\Pi} \boldsymbol{\Psi} = \boldsymbol{\Phi}$. But, the eigenvectors (columns of $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$) are only determined up to sign. So, it just implies

$$\mathbf{\Pi} \boldsymbol{\Psi} = \boldsymbol{\Phi} S,$$

where S is a diagonal matrix with ±1 entries on its diagonal.

[1] I hope I got that right. It's very easy to confuse the permutation and its inverse.

Lemma 39.3.1. *Let $\boldsymbol{A} = \boldsymbol{\Psi} \Lambda \boldsymbol{\Psi}^T$ and $\boldsymbol{B} = \boldsymbol{\Phi} \Lambda \boldsymbol{\Phi}^T$ where Λ is a diagonal matrix with distinct entries and $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$ are orthogonal matrices. A permutation matrix $\mathbf{\Pi}$ satisfies $\mathbf{\Pi} \boldsymbol{A} \mathbf{\Pi}^T = \boldsymbol{B}$ if and only if there exists a diagonal ± 1 matrix $\boldsymbol{S}$ for which*

$$\mathbf{\Pi} \boldsymbol{\Psi} = \boldsymbol{\Phi} \boldsymbol{S}.$$

Proof. Let $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$ be the columns of $\boldsymbol{\Psi}$ and let $\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n$ be the columns of $\boldsymbol{\Phi}$. Assuming there is a $\mathbf{\Pi}$ for which $\mathbf{\Pi} \boldsymbol{A} \mathbf{\Pi}^T = \boldsymbol{B}$,

$$\boldsymbol{\Phi} \Lambda \boldsymbol{\Phi}^T = \sum_{i=1}^{n} \boldsymbol{\phi}_i \lambda_i \boldsymbol{\phi}_i^T = \sum_{i=1}^{n} (\mathbf{\Pi} \boldsymbol{\psi}_i) \lambda_i (\boldsymbol{\psi}_i^T \mathbf{\Pi}^T),$$

which implies that for all i

$$\boldsymbol{\phi}_i \boldsymbol{\phi}_i^T = (\mathbf{\Pi} \boldsymbol{\psi}_i)(\mathbf{\Pi} \boldsymbol{\psi}_i)^T.$$

This in turn implies that

$$\boldsymbol{\phi}_i = \pm \mathbf{\Pi} \boldsymbol{\psi}_i.$$

To go the other direction, assume $\mathbf{\Pi} \boldsymbol{\Psi} = \boldsymbol{\Phi} \boldsymbol{S}$. Then,

$$\mathbf{\Pi} \boldsymbol{A} \mathbf{\Pi}^T = \mathbf{\Pi} \boldsymbol{\Psi} \boldsymbol{\Lambda} \boldsymbol{\Psi}^T \mathbf{\Pi}^T = \boldsymbol{\Phi} \boldsymbol{S} \boldsymbol{\Lambda} \boldsymbol{S} \boldsymbol{\Phi}^T = \boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{S} \boldsymbol{S} \boldsymbol{\Phi}^T = \boldsymbol{\Phi} \boldsymbol{\Lambda} \boldsymbol{\Phi}^T = \boldsymbol{B},$$

as $\boldsymbol{S}$ and $\boldsymbol{\Lambda}$ are diagonal and thus commute, and $\boldsymbol{S}^2 = \boldsymbol{I}$. □

Our algorithm for testing isomorphism will determine all such matrices $\boldsymbol{S}$. Let $\mathcal{S}$ be the set of all diagonal ± 1 matrices. We will find diagonal matrices $\boldsymbol{S} \in \mathcal{S}$ such that the set of rows of $\boldsymbol{\Phi} \boldsymbol{S}$ is the same as the set of rows of $\boldsymbol{\Psi}$. As the rows of $\boldsymbol{\Psi}$ are indexed by vertices $a \in V$, we will write the row indexed by a as the row-vector

$$\boldsymbol{v}_a \stackrel{\text{def}}{=} (\boldsymbol{\psi}_1(a), \dots, \boldsymbol{\psi}_n(a)).$$

Similarly denote the rows of $\boldsymbol{\Phi}$ by vectors $\boldsymbol{u}_a$. In this notation, we are searching for matrices $\boldsymbol{S} \in \mathcal{S}$ for which the set of vectors $\{\boldsymbol{v}_a\}_{a \in V}$ is identical to the set of vectors $\{\boldsymbol{u}_a \boldsymbol{S}\}_{a \in V}$ We have thus transformed the graph isomorphism problem into a problem about vectors:

39.4 An easy case

I will say that an eigenvector $\boldsymbol{\psi}_i$ is *helpful* if for all $a \neq b \in V$, $|\boldsymbol{\psi}_i(a)| \neq |\boldsymbol{\psi}_i(b)|$. In this case, it is very easy to test if G and H are isomorphic, because this helpful vector gives us a canonical name for every vertex. If $\mathbf{\Pi}$ is an isomorphism from G to H, then $\mathbf{\Pi} \boldsymbol{\psi}_i$ must be an eigenvector of B. In fact, is must be $\pm \boldsymbol{\phi}_i$. If the sets of absolute values of entries of $\boldsymbol{\psi}_i$ and $\boldsymbol{\phi}_i$ are the same, then we may find the permutation that maps $\boldsymbol{A}$ to $\boldsymbol{B}$ by mapping every vertex a to the vertex b for which $|\boldsymbol{\psi}_i(a)| = |\boldsymbol{\phi}_i(b)|$.

The reason that I put absolute values in the definition of helpful, rather than just taking values, is that eigenvectors are only determined up to sign. On the other hand, a single eigenvector determines the isomorphism if $\boldsymbol{\psi}_i(a) \neq \boldsymbol{\psi}_i(b)$ for all $a \neq b$ and there is a canonical way to choose a sign for the vector $\boldsymbol{\psi}_i$. For example, if the sum of the entries in $\boldsymbol{\psi}_i$ is not zero, we can choose its sign to make the sum positive. In fact, unless $\boldsymbol{\psi}_i$ and $-\boldsymbol{\psi}_i$ have exactly the same set of values, there is a canonical choice of the sign for this vector.

Even if there is no canonical choice of sign for this vector, it leaves at most two choices for the isomorphism.

39.5 All the Automorphisms

The graph isomorphism problem is complicated by the fact that there can be many isomorphisms from one graph to another. So, any algorithm for finding isomorphisms must be able to find many of them.

Recall that an *automorphism* of a graph is an isomorphism from the graph to itself. These form a group which we denote $\mathsf{aut}(G)$: if $\boldsymbol{\Pi}$ and $\boldsymbol{\Gamma}$ are automorphisms of $\boldsymbol{A}$ then so is $\boldsymbol{\Pi}\boldsymbol{\Gamma}$. Let $\mathcal{A} \subseteq \mathcal{S}$ denote the corresponding set of diagonal ±1 matrices. The set $\mathcal{A}$ is in fact a group and is isomorphic to $\mathsf{aut}(G)$.

Here is a way to make this isomorphism very concrete: Lemma 39.3.1 implies that the $\boldsymbol{\Pi} \in \mathsf{aut}(G)$ and the $\boldsymbol{S} \in \mathcal{A}$ are related by

$$\boldsymbol{\Pi} = \boldsymbol{\Psi} \boldsymbol{S} \boldsymbol{\Psi}^T \quad \text{and} \quad \boldsymbol{S} = \boldsymbol{\Psi}^T \boldsymbol{\Pi} \boldsymbol{\Psi}.$$

As diagonal matrices commute, we have that for every $\boldsymbol{\Pi}_1$ and $\boldsymbol{\Pi}_2$ in $\mathsf{aut}(G)$ and for $\boldsymbol{S}_1 = \boldsymbol{\Psi}^T \boldsymbol{\Pi}_1 \boldsymbol{\Psi}$ and $\boldsymbol{S}_2 = \boldsymbol{\Psi}^T \boldsymbol{\Pi}_2 \boldsymbol{\Psi}$,

$$\boldsymbol{\Pi}_1 \boldsymbol{\Pi}_2 = \boldsymbol{\Psi} \boldsymbol{S}_1 \boldsymbol{\Psi}^T \boldsymbol{\Psi} \boldsymbol{S}_2 \boldsymbol{\Psi}^T = \boldsymbol{\Psi} \boldsymbol{S}_1 \boldsymbol{S}_2 \boldsymbol{\Psi}^T = \boldsymbol{\Psi} \boldsymbol{S}_2 \boldsymbol{S}_1 \boldsymbol{\Psi}^T = \boldsymbol{\Psi} \boldsymbol{S}_2 \boldsymbol{\Psi}^T \boldsymbol{\Psi} \boldsymbol{S}_1 \boldsymbol{\Psi}^T = \boldsymbol{\Pi}_2 \boldsymbol{\Pi}_1.$$

Thus, the automorphism group of a graph with distinct eigenvalues is commutative, and it is isomorphic to a subgroup of $\mathcal{S}$.

It might be easier to think about these subgroups by realizing that they are isomorphic to subspaces of $(\mathbf{Z}/2\mathbf{Z})^n$. Let $f : \mathcal{S} \to (\mathbf{Z}/2\mathbf{Z})^n$ be the function that maps the group of diagonal matrices with ±1 entries to vectors $\boldsymbol{t}$ modulo 2 by setting $\boldsymbol{t}(i)$ so that $\boldsymbol{S}(i,i) = (-1)^{\boldsymbol{t}(i)}$. You should check that this is a group homomorphism: $f(\boldsymbol{S}_1 \boldsymbol{S}_2) = f(\boldsymbol{S}_1) + f(\boldsymbol{S}_2)$. You should also confirm that f is invertible.

For today's lecture, we will focus on the problem of finding the group of automorphisms of a graph with distinct eigenvalues. We will probably save the slight extension to finding isomorphisms for homework. Note that we will not try to list all the isomorphisms, as there could be many. Rather, we will give a basis of the corresponding subspace of $(\mathbf{Z}/2\mathbf{Z})^n$.

39.6 Equivalence Classes of Vertices

Recall that the *orbit* of an element under the action of a group is the set of elements to which it is mapped by the elements of the group. Concretely, the orbit of a vertex a in the graph is the set of vertices to which it can be mapped by automorphisms. We will discover the orbits by realizing that the orbit of a vertex a is the set of b for which $\boldsymbol{v}_a \boldsymbol{S} = \boldsymbol{v}_b$ for some $\boldsymbol{S} \in \mathcal{A}$.

The set of orbits of vertices forms a partition of the vertices. We say that a partition of the vertices is *valid* if every orbit is contained entirely within one set in the partition. That is, each class of the partition is a union of orbits. Our algorithm will proceed by constructing a valid partition of the vertices and then splitting classes in the partition until each is exactly an orbit.

Recall that a set is *stabilized* by a group if the set is unchanged when the group acts on all of its members. We will say that a group $\mathcal{G} \subseteq \mathcal{S}$ stabilizes a set of vertices C if it stabilizes the set of vectors $\{\boldsymbol{v}_a\}_{a \in C}$. Thus, $\mathcal{A}$ is the group that stabilizes V.

An orbit is stabilized by $\mathcal{A}$, and so are unions of orbits and thus classes of valid partitions. We would like to construct the subgroup of $\mathcal{S}$ that stabilizes each orbit C_j. *However, I do not yet see how to do that directly.* Instead, we will construct a particular valid partition of the vertices, and find for each class in the partition C_j the subgroup of $\mathcal{A}_j \subseteq \mathcal{S}$ that stabilizes C_j, where here we are considering the actions of matrices $\boldsymbol{S} \in \mathcal{S}$ on vectors $\boldsymbol{v}_a$. In fact, $\mathcal{A}_j$ will act transitively[2] on the class C_j. As $\mathcal{A}$ stabilizes every orbit, and thus every union of orbits, it is a subgroup of $\mathcal{A}_j$. In fact, $\mathcal{A}$ is exactly the intersection of all the groups $\mathcal{A}_j$.

We now observe that we can use linear algebra to efficiently construct $\mathcal{A}$ from the groups $\mathcal{A}_j$ by exploiting the isomorphism between $\boldsymbol{S}$ and $(\mathbf{Z}/2)^n$. Each subgroup $\mathcal{A}_j$ is isomorphic to a subgroup of $(\mathbf{Z}/2)^n$. Each subgroup of $(\mathbf{Z}/2)^n$ is precisely a vector space modulo 2, and thus may be described by a basis. It will eventually become clear that by "compute $\mathcal{A}_j$" we mean to compute such a basis. From the basis, we may compute a basis of the nullspace. The subgroup of $(\mathbf{Z}/2)^n$ corresponding to $\mathcal{A}$ is then the nullspace of the span of the nullspaces of the subspaces corresponding to the $\mathcal{A}_j$. We can compute all these using Gaussian elimination.

39.7 The first partition

We may begin by dividing vertices according to the absolute values of their entries in eigenvectors. That is, if $|\boldsymbol{\psi}_i(a)| \neq |\boldsymbol{\psi}_i(b)|$ for some i, then we may place vertices a and b in different classes, as there can be no $\boldsymbol{S} \in \mathcal{S}$ for which $\boldsymbol{v}_a \boldsymbol{S} = \boldsymbol{v}_b$. The partition that we obtain this way is thus valid, and is the starting point of our algorithm.

39.8 Unbalanced vectors

We say that an eigenvector $\boldsymbol{\psi}_i$ is *unbalanced* if there is some value x for which

$$|\{a : \boldsymbol{\psi}_i(a) = x\}| \neq |\{a : \boldsymbol{\psi}_i(a) = -x\}| \,.$$

Such vectors cannot change sign in an automorphism. That is, $\boldsymbol{S}(i,i)$ must equal 1. The reason is that an automorphism with $\boldsymbol{S}(i,i) = -1$ must induce a bijection between the two sets above, but this is impossible if their sizes are different.

[2] That is, for every a and b in C_j, there is an $\boldsymbol{S} \in \mathcal{A}_j$ for which $\boldsymbol{v}_a \boldsymbol{S} = \boldsymbol{b}_b$.

Thus, an unbalanced vector tells us that all vertices for which $\boldsymbol{\psi}_i(a) = x$ are in different orbits from those for which $\boldsymbol{\psi}_i(a) = -x$. This lets us refine classes.

We now extend this idea in two ways. First, we say that $\boldsymbol{\psi}_i$ is *unbalanced* on a class C if there is some value x for which

$$|\{a \in C : \boldsymbol{\psi}_i(a) = x\}| \neq |\{a \in C : \boldsymbol{\psi}_i(a) = -x\}| .$$

By the same reasoning, we can infer that the sign of $\boldsymbol{S}(i,i)$ must be fixed to 1. Assuming, as will be the case, that C is a class in a valid partition and thus a union of orbits, we are now able to split C into two smaller classes

$$C_0 = \{a \in C : \boldsymbol{\psi}_i(a) = x\} \quad \text{and} \quad C_1 = \{a \in C : \boldsymbol{\psi}_i(a) = -x\} .$$

The partition we obtain by splitting C into C_1 and C_2 is thus also valid. Of course, it is only useful if both sets are non-empty.

Finally, we consider vectors formed from products of eigenvectors. For $R \subseteq \{1, \dots, n\}$, define $\boldsymbol{\psi}_R$ to be the component-wise product of the $\boldsymbol{\psi}_i$ for $i \in R$:

$$\boldsymbol{\psi}_R(a) = \prod_{i \in R} \boldsymbol{\psi}_i(a).$$

We say that the vector $\boldsymbol{\psi}_R$ is unbalanced on class C if there is some value x for which

$$|\{a \in C : \boldsymbol{\psi}_R(a) = x\}| \neq |\{a \in C : \boldsymbol{\psi}_R(a) = -x\}| .$$

An unbalanced vector of this form again tells us that the vertices in the two sets belong to different orbits. So, if both sets are nonempty we can use such a vector to split the class C in two to obtain a more refined valid partition. It also provides some relations between the entries of $\boldsymbol{S}$, but we will not exploit those.

We say that a vector is *balanced* if it is not unbalanced.

We say that a subset of the vertices $C \subseteq V$ is *balanced* if *every* non-constant product of eigenvectors is balanced on C. Thus, orbits are balanced. Our algorithm will partition the vertices into balanced classes.

My confusion over this lecture stemmed from thinking that all balanced classes must be orbits. But, I don't know if this is true.

Question: Is every balanced class an orbit of $\mathcal{A}$?

39.9 The structure of the balanced classes

Let C_j be a balanced class. By definition, the product of every subset of eigenvectors is either constant or balanced on C_j. We say that a subset of eigenvectors Q is *independent* on C_j if all products of subsets of eigenvectors in Q are balanced on C_j (except for the empty product). In particular, none of these eigenvectors is zero or constant on C_j. Construct a matrix $M_{C_j,Q}$ whose rows are indexed by vertices in $a \in C_j$, whose columns are indexed by subsets $R \subseteq Q$, and whose entries are given by

$$M_{C_j,Q}(a,R) = \mathsf{sgn}(\boldsymbol{\psi}_R(a)), \text{where I recall } \mathsf{sgn}(x) = \begin{cases} 1 & \text{if } x > 0 \\ -1 & \text{if } x < 0, \text{and} \\ 0 & \text{if } x = 0. \end{cases}$$

Lemma 39.9.1. *If Q is independent on C then the columns of $M_{C,Q}$ are orthogonal.*

Proof. Let R_1 and R_2 index two columns of $M_{C,Q}$. That is, R_1 and R_2 are two different subsets of Q. Let R_0 be their symmetric difference. We have

$$M_{C,Q}(a,R_1)M_{C,Q}(a,R_2) = \mathsf{sgn}(\boldsymbol{\psi}_{R_1}(a))\mathsf{sgn}(\boldsymbol{\psi}_{R_2}(a)) = \\ \prod_{i \in R_1} \mathsf{sgn}(\boldsymbol{\psi}_i(a)) \prod_{i \in R_2} \mathsf{sgn}(\boldsymbol{\psi}_i(a)) = \prod_{i \in R_0} \mathsf{sgn}(\boldsymbol{\psi}_i(a)) = \mathsf{sgn}(\boldsymbol{\psi}_{R_0}(a)) = M_{C,Q}(a,R_0).$$

As all the nonempty products of subsets of eigenvectors in Q are balanced on C, $M_{C,Q}(a,R_0)$ is positive for half the $a \in C$ and negative for the other half. So,

$$M_{C,Q}(:,R_1)^T M_{C,Q}(:,R_2) = \sum_{a \in C} M_{C,Q}(a,R_1)M_{C,Q}(a,R_2) = \sum_{a \in C} M_{C,Q}(a,R_0) = 0.$$

□

Lemma 39.9.2. *If C is a balanced class of vertices and Q is a maximal set of eigenvectors that are independent on C, then for every a and b in C there is an $i \in Q$ for which $\boldsymbol{\psi}_i(a) \neq \boldsymbol{\psi}_i(b)$.*

Proof. Assume by way of contradiction that this does not hold. There must be some eigenvector i for which $\boldsymbol{\psi}_i(a) \neq \boldsymbol{\psi}_i(b)$. We will show that if we added i to Q, the product of every subset would still be balanced. As we already know this for subsets of Q, we just have to prove it for subsets of the form $R \cup \{i\}$, where $R \subseteq Q$. As $\boldsymbol{\psi}_h(a) = \boldsymbol{\psi}_h(b)$ for every $h \in Q$, $\boldsymbol{\psi}_R(a) = \boldsymbol{\psi}_R(b)$. This implies $\boldsymbol{\psi}_{R\cup\{i\}}(a) \neq \boldsymbol{\psi}_{R\cup\{i\}}(b)$. Thus, $\boldsymbol{\psi}_{R\cup\{i\}}$ is not uniform on C, and so it must be balanced on C. □

Lemma 39.9.3. *If C is a balanced class of vertices and Q is a maximal set of eigenvectors that are independent on C, then the rows of $M_{C,Q}$ are orthogonal.*

Proof. Let a and b be in C. From Lemma 39.9.2 we know that there is an $i \in Q$ for which $\boldsymbol{\psi}_i(a) = -\boldsymbol{\psi}_i(b)$. To prove that the rows $M_{C,Q}(a,:)$ and $M_{C,Q}(b,:)$ are orthogonal, we compute their inner product:

$$\begin{aligned}\sum_{R\subseteq Q} \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) &= \sum_{R\subseteq Q-\{i\}} \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) + \mathsf{sgn}(\boldsymbol{\psi}_{R\cup\{i\}}(a)\boldsymbol{\psi}_{R\cup\{i\}}(b))\\ &= \sum_{R\subseteq Q-\{i\}} \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) + \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_i(a)\boldsymbol{\psi}_R(b)\boldsymbol{\psi}_i(b))\\ &= \sum_{R\subseteq Q-\{i\}} \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) + \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b))\mathsf{sgn}(\boldsymbol{\psi}_i(a)\boldsymbol{\psi}_i(b))\\ &= \sum_{R\subseteq Q-\{i\}} \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b)) - \mathsf{sgn}(\boldsymbol{\psi}_R(a)\boldsymbol{\psi}_R(b))\\ &= 0.\end{aligned}$$

□

Corollary 39.9.4. *Let C be a balanced subset of vertices. Then the size of C is a power of 2. If Q is an independent set of eigenvectors on C, then $|Q| \leq \log_2 |C|$.*

Proof. Let C be an orbit and let Q be a maximal set of eigenvectors that are independent on C. As the rows and columns of $M_{C,Q}$ are both orthogonal, $M_{C,Q}$ must be square. This implies that $|C| = 2^{|Q|}$. If we drop the assumption that Q is maximal, we still know that all the columns of $M_{C,Q}$ are orthogonal. This matrix has $2^{|Q|}$ columns. As they are vectors in $|C|$ dimensions, there can be at most $|C|$ of them. □

We can now describe the structure of a balanced subset of vertices C. We call a maximal set of eigenvectors that are independent on C a *base* for C. Every other eigenvector j is either constant on C or becomes constant when multiplied by the product of some subset R of eigenvectors in Q. In either case, we can write

$$\boldsymbol{\psi}_j(a) = \gamma \prod_{i\in R} \boldsymbol{\psi}_i(a) \quad \text{for all } a \in C, \tag{39.1}$$

for some constant γ.

Let $\boldsymbol{v}_a(Q)$ denote the vector $(\boldsymbol{v}_a(i))_{i\in Q}$—the restriction of the vector $\boldsymbol{v}_a$ to the coordinates in Q. I claim that every one of the $2^{|Q|}$ ± sign patterns of length $|Q|$ must appear in exactly one of the vectors $\boldsymbol{v}_q(Q)$. The reason is that there are $|C| = 2^{|Q|}$ of these vectors, and we established in Lemma 39.9.2 that $\boldsymbol{v}_a(Q) \neq \boldsymbol{v}_b(Q)$ for all $a \neq b$ in Q. Thus, for every diagonal ± matrix $\boldsymbol{S}_Q$ of dimension $|Q|$, we have

$$\{\boldsymbol{v}_a(Q)\boldsymbol{S}_Q : a \in C\} = \{\boldsymbol{v}_a(Q) : a \in C\}.$$

That is, this set of vectors is stabilized by ±1 diagonal matrices.

As equation (39.1) gives a formula for the value taken on C by every eigenvector not in Q in terms of the eigenvectors in Q, we have described the structure of the subgroup of $\boldsymbol{S}$ that stabilizes C: the diagonals corresponding to Q are unconstrained, and every other diagonal is some product of these. This structure is something that you are used to seeing in subspaces.

Apply f to map this subgroup of $\boldsymbol{S}$ to $(\mathbf{Z}/2)^n$, and let $\boldsymbol{B}$ be a n-by-$\log_2(|C|)$ matrix containing a basis of the subspace in its columns. Any independent subset of $\log_2(|C|)$ rows of $\boldsymbol{B}$ will form a basis of the row-space, and is isomorphic to a base for C of the eigenvectors.

39.10 Algorithms

Let C_j be a balanced class. We just saw how to compute $\mathcal{A}_j$, assuming that we know C_j and a base Q for it. Of course, by "compute" we mean computing a basis of $f(\mathcal{A}_j)$. We now show how to find a base for a balanced class C_j. We do this by building up a set Q of eigenvectors that are independent on C_j. To do this, we go through the eigenvectors in order. For each eigenvector $\boldsymbol{\psi}_i$, we must determine whether or not its values on C_j can be expressed as a product of eigenvectors already present in Q. If it can be, then we record this product as part of the structure of $\mathcal{A}_j$. If not, we add i to Q.

The eigenvector $\boldsymbol{\psi}_i$ is a product of eigenvectors in Q on C_j if and only if there is a constant γ and $y_h \in \{0, 1\}$ for $h \in Q$ such that

$$\boldsymbol{\psi}_i(a) = \gamma \prod_{h \in Q} (\boldsymbol{\psi}_h(a))^{y_h},$$

for all vertices $a \in C_j$. This happens if and only if

$$\mathsf{sgn}(\boldsymbol{\psi}_i(a)) = \prod_{h \in Q} \mathsf{sgn}(\boldsymbol{\psi}_h(a))^{y_h}.$$

We can tell whether or not these equations have a solution using linear algebra modulo 2. Let $\boldsymbol{B}$ be the matrix over $\mathbf{Z}/2$ such that

$$\boldsymbol{\psi}_i(a) = (-1)^{\boldsymbol{B}(i,a)}.$$

Then, the above equations become

$$\boldsymbol{B}(i,a) = \sum_{h \in Q} y_h \boldsymbol{B}(h,a) \quad \text{for all } a \in C_j.$$

Thus, we can solve for the coefficients y_h in polynomial time, if they exist. If they do not, we add i to Q.

Once we have determined a base Q and how to express on C_j the values of every other eigenvector as a product of eigenvectors in Q, we have determine A_j.

It remains to explain how we partition the vertices into balanced classes. Consider applying the above procedure to a class C_j that is not balanced. We will discover that C_j is not balanced by finding a product of eigenvectors that is neither constant nor balanced on C_j. Every time we add an eigenvector $\boldsymbol{\psi}_i$ to Q, we will examine *every* product of vectors in Q to check if any are unbalanced on C_j. We can do this efficiently, because there are at most $2^{|Q|} \leq |C_j|$ such products to consider. As we have added $\boldsymbol{\psi}_i$ to Q, none of the products of vectors in Q can be constant on C_j. If we find a product that it not balanced on C_j, then it must also be non-constant, and thus provide a way of splitting class C_j into two.

We can now summarize the entire algorithm. We first compute the partition by absolute values of entries described in section 39.7. We then go through the classes of the partition one-by-one. For each, we use the above procedure until we have either split it in two or we have determined that it is balanced and we have computed its automorphism group. If we do split the class in two, we refine the partition and start over. As the total number of times we split classes is at most n, this algorithm runs in polynomial time.

After we have computed a partition into balanced classes and have computed their automorphisms groups, we combine them to find the automorphisms group of the entire graph as described at the end of section 39.6.

Chapter 40

Testing Isomorphism of Strongly Regular Graphs

Lecture 9 from September 26, 2018

40.1 Introduction

In the last lecture we saw how to test isomorphism of graphs in which every eigenvalue is distinct. So, in this lecture we will consider the opposite case: graphs that only have 3 distinct eigenvalues. These are the strongly regular graphs.

Our algorithm for testing isomorphism of these will not run in polynomial time. Rather, it takes time $n^{O(n^{1/2} \log n)}$. This is at least much faster than the naive algorithm of checking all $n!$ possible permutations. In fact, this was the best known running time for general algorithms for graph isomorphism until three years ago.

40.2 Definitions

A graph G is strongly regular if

1. it is d-regular, for some integer d;
2. there exists an integer α such that for every pair of vertices x and y that are neighbors in G, there are exactly α vertices z that are neighbors of *both* x and y;
3. there exists an integer β such that for every pair of vertices x and y that are *not* neighbors in G, there are exactly β vertices z that are neighbors of *both* x and y.

These conditions are very strong, and it might not be obvious that there are any non-trivial graphs that satisfy these conditions. Of course, the complete graph and disjoint unions of complete graphs satisfy these conditions. Before proceeding, I warn you that there is a standard notation in the literature about strongly regular graphs, and I am trying not to use it. In this literature, d becomes k, α becomes λ and β becomes μ. Many other letters are bound as well.

For the rest of this lecture, we will only consider strongly regular graphs that are connected and that are not the complete graph. I will now give you some examples.

40.3 Paley Graphs and The Pentagon

The Paley graphs we encountered are strongly regular. The simplest of these is the pentagon. It has parameters

$$n = 5, \quad d = 2, \quad \alpha = 0, \quad \beta = 1.$$

40.4 Lattice Graphs

For a positive integer n, the *lattice graph* L_n is the graph with vertex set $\{1, \dots n\}^2$ in which vertex (a, b) is connected to vertex (c, d) if $a = c$ or $b = d$. Thus, the vertices may be arranged at the points in an n-by-n grid, with vertices being connected if they lie in the same row or column. Alternatively, you can understand this graph as the product of two complete graphs on n vertices.

The parameters of this graph are:

$$d = 2(n - 1), \quad \alpha = n - 2, \quad \beta = 2.$$

40.5 Latin Square Graphs

A Latin square is an n-by-n grid, each entry of which is a number between 1 and n, such that no number appears twice in any row or column. For example,

$$\begin{bmatrix} 1 & 2 & 3 & 4 \\ 4 & 1 & 2 & 3 \\ 3 & 4 & 1 & 2 \\ 2 & 3 & 4 & 1 \end{bmatrix}, \quad \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 1 & 4 & 3 \\ 3 & 4 & 1 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 1 & 2 & 3 & 4 \\ 2 & 4 & 1 & 3 \\ 3 & 1 & 4 & 2 \\ 4 & 3 & 2 & 1 \end{bmatrix}$$

are Latin squares. Let me remark that the number of different Latin squares of size n grows very quickly—at least as fast as $n!(n - 1)!(n - 2)! \dots 2!$. Two Latin squares are said to be *isomorphic* if there is a renumbering of their rows, columns, and entries, or a permutation of these, that makes them the same. As this provides $6(n!)^3$ isomorphisms, and this is much less than the number of Latin squares, there must be many non-isomorphic Latin squares of the same size. The two of the Latin squares above are isomorphic, but one is not.

From such a Latin square, we construct a Latin square graph. It will have n^2 nodes, one for each cell in the square. Two nodes are joined by an edge if

1. they are in the same row,
2. they are in the same column, or
3. they hold the same number.

So, such a graph has degree $d = 3(n-1)$. Any two nodes in the same row will both be neighbors with every other pair of nodes in their row. They will have two more common neighors: the nodes in their columns holding the other's number. So, they have n common neighbors. The same obviously holds for columns, and is easy to see for nodes that have the same number. So, every pair of nodes that are neighbors have exactly $\alpha = n$ common neighbors.

On the other hand, consider two vertices that are not neighbors, say $(1,1)$ and $(2,2)$. They lie in different rows, lie in different columns, and we are assuming that they hold different numbers. The vertex $(1,1)$ has two common neighbors of $(2,2)$ in its row: the vertex $(1,2)$ and the vertex holding the same number as $(2,2)$. Similarly, it has two common neighbors of $(2,2)$ in its column. Finally, we can find two more common neighbors of $(2,2)$ that are in different rows and columns by looking at the nodes that hold the same number as $(1,1)$, but which are in the same row or column as $(2,2)$. So, $\beta = 6$.

40.6 The Eigenvalues of Strongly Regular Graphs

We will consider the adjacency matrices of strongly regular graphs. Let A be the adjacency matrix of a strongly regular graph with parameters (d, α, β). We already know that A has an eigenvalue of d with multiplicity 1. We will now show that A has just two other eigenvalues.

To prove this, first observe that the (a,b) entry of A^2 is the number of common neighbors of vertices a and b. For $a = b$, this is just the degree of vertex a. We will use this fact to write A^2 as a linear combination of A, I and J, the all 1s matrix. To this end, observe that the adjacency matrix of the complement of A (the graph with non-edges where A has edges) is $J - I - A$. So,

$$A^2 = \alpha A + \beta(J - I - A) + dI = (\alpha - \beta)A + \beta J + (d - \beta)I.$$

For every vector $\boldsymbol{v}$ orthogonal to $\mathbf{1}$,

$$A^2\boldsymbol{v} = (\alpha - \beta)A\boldsymbol{v} + (d - \beta)\boldsymbol{v}.$$

So, every eigenvalue λ of A other than d satisfies

$$\lambda^2 = (\alpha - \beta)\lambda + d - \beta.$$

Thus, these are given by

$$\lambda = \frac{\alpha - \beta \pm \sqrt{(\alpha - \beta)^2 + 4(d - \beta)}}{2}.$$

These eigenvalues are traditionally denoted r and s, with $r > s$. By convention, the multiplicty of the eigenvalue r is always denoted f, and the multiplicty of s is always denoted g.

For example, for the pentagon we have

$$r = \frac{\sqrt{5}-1}{2}, \quad s = -\frac{\sqrt{5}+1}{2}.$$

For the lattice graph L_n, we have

$$r = n-2, \quad s = -2.$$

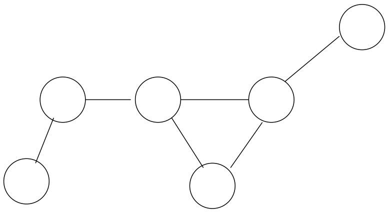
For the Latin square graphs of order n, we have

$$r = n-3, \quad s = -3.$$

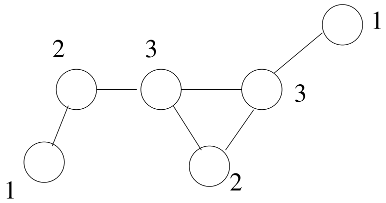
One can prove that every connected regular graph whose adjacency (or Laplacian) matrix has just three distinct eigenvalues is a strongly regular graph.

40.7 Testing Isomorphism by Individualization and Refinement

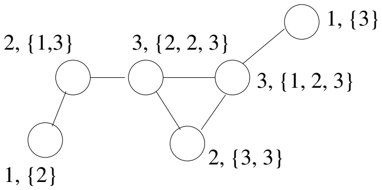
The problem of testing isomorphism of graphs is often reduced to the problem of giving each vertex in a graph a unique name. If we have a way of doing this that does not depend upon the initial ordering of the vertices, then we can use it to test graph isomorphism: find the unique names of vertices in both graphs, and then see if it provides an isomorphism. For example, consider the graph below.



We could begin by labeling every vertex by its degree.



The degrees distinguish between many nodes, but not all of them. We may refine this labeling by appending the labels of every neighbor of a node.



Now, every vertex has its own unique label. If we were given another copy of this graph, we could use these labels to determine the isomorphism between them. This procedure is called *refinement*, and it can be carried out until it stops producing new labels. However, it is clear that this procedure will fail to produce unique labels if the graph has automorphisms, or if it is a regular graph. In these cases, we need a way to break symmetry.

The procedure called *individualization* breaks symmetry arbitrarily. It chooses some nodes in the graph, arbitrarily, to give their own unique names. Ideally, we pick one vertex to give a unique name, and then refine the resulting labeling. We could then pick another troubling vertex, and continue. We call a set of vertices $S \subset V$ a *distinguishing set* if individualizing this set of nodes results in a unique name for every vertex, after refinement. How would we use a distinguishing set to test isomorphism? Assume that S is a distinguishing set for $G = (V, E)$. To test if $H = (W, F)$ is isomorphic to G, we could enumerate over *every* possible set of $|S|$ vertices of W, and check if they are a distinguishing set for H. If G and H are isomorphic, then H will also have an isomorphic distinguishing set that we can use to find an isomorphism between G and H. We would have to check $\binom{n}{|S|}$ sets, and try $|S|!$ labelings for each, so we had better hope that S is small.

40.8 Distinguishing Sets for Strongly Regular Graphs

We will now prove a result of Babai [Bab80] which says that every strongly regular graph has a distinguishing set of size $O(\sqrt{n} \log n)$. Babai's result won't require any refinement beyond naming every vertex by the set of individualized nodes that are its neighbors. So, we will prove that a set of nodes S is a distinguishing set by proving that for every pair of distinct vertices a and b, either there is an $s \in S$ that is a neighbor of a but not of b, or the other way around. This will suffice to distinguish a and b. As our algorithm will work in a brute-force fashion, enumerating over all sets of a given size, we merely need to show that such a set S exists. We will do so by proving that a random set of vertices probably works.

I first observe that it suffices to consider strongly-regular graphs with $d < n/2$, as the complement of a strongly regular graph is also a strongly regular graph (that would have been too easy to assign as a homework problem). We should also observe that every strongly-regular graph has diameter 2, and so $d \geq \sqrt{n-1}$.

Lemma 40.8.1. *Let $G = (V, E)$ be a connected strongly regular graph with n vertices and degree $d < n/2$. Then for every pair of vertices a and b, there are at least $d/3$ vertices that are neighbors of a but not b.*

Before I prove this, let me show how we may use it to prove the theorem. This lemma tells us that there are at least $\sqrt{n-1}/3$ nodes that are neighbors of a but not of b. Let T be the set of nodes that are neighbors of a but not neighbors of b. So, if we choose a vertex at random, the probability that it is in T is at least

$$\frac{|T|}{n} \geq \frac{\sqrt{n-1}}{3n} \geq \frac{1}{3\sqrt{n+2}}.$$

If we choose a set S of $3\sqrt{n+2}\ln n^2$ vertices at random, the probability that none of them is in T is

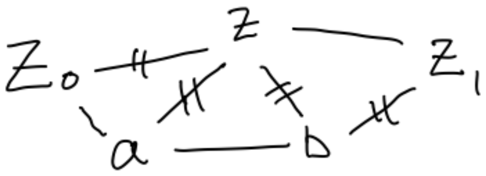
$$\left(1 - \frac{1}{3\sqrt{n+2}}\right)^{3\sqrt{n+2}\ln n^2} \leq \frac{1}{n^2}.$$

So, the probability that a random set of this many nodes fails to distinguish all $\binom{n}{2}$ pairs is at most $1/2$.

Proof of Lemma 40.8.1. Write $a \sim b$ if a is a neighbor of b, and $a \not\sim b$ otherwise. If $a \sim b$, then the number of nodes that are neighbors of a but not of b is $d - 1 - \alpha$, and if $a \not\sim b$ the number if $d - \beta$. So, we need to prove that neither α nor β is too close to d.

We will do this by establishing some elementary relations between these parameters. First, consider the case in which $a \sim b$. Let z be any vertex such that $a \not\sim z$ and $b \not\sim z$. We will use z to prove an upper bound on the number of vertices w that are neighbors of a but not of b (I know this looks like the wrong direction, but be patient). Let

$$Z_0 = \{w : w \sim a, w \not\sim z\}, \quad \text{and} \quad Z_1 = \{w : w \not\sim b, w \sim z\}.$$



Clearly, every w that is a neighbor of a but not of b lies in either Z_0 or Z_1. As z is neither a neighbor of a nor of b,

$$|Z_0| = |Z_1| = d - \beta.$$

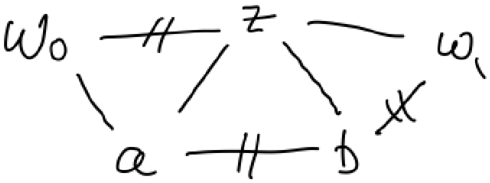
So,

$$d - \alpha - 1 \leq 2(d - \beta) \quad \Longrightarrow \quad 2\beta \leq d + \alpha + 1. \tag{40.1}$$

So, if β is close to d, α must also be close to d.

We will obtain an inequality in the other direction when $a \not\sim b$ by exploiting a z such that $z \sim a$ and $z \sim b$. Now, for any $w \sim a$ but $w \not\sim b$, we have either

$$(w \sim a \text{ and } w \not\sim z) \text{ or } (w \sim z \text{ and } w \not\sim b).$$

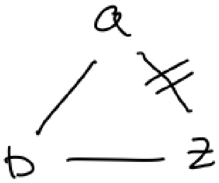


So,

$$d - \beta \leq 2(d - \alpha - 1) \quad \Longrightarrow \quad 2(\alpha + 1) \leq d + \beta. \tag{40.2}$$

This tells us that if α is close to d, then β is also.

We require one more relation between α and β. We obtain this relation by picking any vertex a, and counting the pairs b, z such that $b \sim z$, $a \sim b$ and $a \not\sim z$.



Every node b that is a neighbor of a has α neighbors in common with a, and so has $d - \alpha - 1$ neighbors that are not neighbors of a. This gives

$$|\{(b, z) : b \sim z, a \sim b, a \not\sim z\}| = d(d - \alpha - 1).$$

On the other hand, there are $n - d - 1$ nodes z that are not neighbors of a, and each of them has β neighbors in common with a, giving

$$|\{(b, z) : b \sim z, a \sim b, a \not\sim z\}| = (n - d - 1)\beta.$$

Combining, we find

$$(n - d - 1)\beta = d(d - \alpha - 1). \tag{40.3}$$

As $d < n/2$, this equation tells us

$$d(d - \alpha - 1) \geq d\beta \quad \Longrightarrow \quad d - \alpha - 1 \geq \beta. \tag{40.4}$$

Adding inequality 40.1 to (40.4) gives

$$2d \geq 3\beta \quad \Longrightarrow \beta \leq \frac{2}{3}d,$$

while adding inequality 40.2 to (40.4) gives

$$\alpha + 1 \leq \frac{2}{3}d.$$

Thus, for every $a \neq b$ the number of vertices that are neighbors of a but not of b is at least $\min(d - \alpha - 1, d - \beta) \geq d/3$.

□

40.9 Notes

You should wonder if we can make this faster by analyzing refinement steps. In, [Spi96b], I improved the running time bound to $2^{O(n^{1/3} \log n)}$ by analyzing two refinement phases. The algorithm required us to handle certain special families of strongly regular graphs separately: Latin square graphs and Steiner graphs. Algorithms for testing isomorphism of strongly regular graphs were recently improved by Babai, Chen, Sun, Teng, and Wilmes [BCS+13, BW13, SW15]. The running times of all these algorithms are subsumed by that in Babai's breakthrough algorithm for testing graph isomorphism [Bab16].

Part VII

Interlacing Families

Chapter 41

Bipartite Ramanujan Graphs

Lecture 25 from December 3, 2018

41.1 Overview

Margulis [Mar88] and Lubotzky, Phillips and Sarnak [LPS88] presented the first explicit constructions of infinite families of Ramanujan graphs. These had degrees $p+1$, for primes p. There have been a few other explicit constructions, [Piz90, Chi92, JL97, Mor94], all of which produce graphs of degree $q+1$ for some prime power q. Over this lecture and the next we will prove the existence of infinite families of bipartite Ramanujan of every degree. While today's proof of existence does not lend itself to an explicit construction, it is easier to understand than the presently known explicit constructions.

We think that much stronger results should be true. There is good reason to think that random d-regular graphs should be Ramanujan [MNS08]. And, Friedman [Fri08] showed that a random d-regular graph is almost Ramanujan: for sufficiently large n such a graph is a $2\sqrt{d-1}+\epsilon$ approximation of the complete graph with high probability, for every $\epsilon > 0$.

In today's lecture, we will use the method of interlacing families of polynomials to prove (half) a conjecture of Bilu and Linial [BL06] that every bipartite Ramanujan graph has a 2-lift that is also Ramanujan. This theorem comes from [MSS15b], but today's proof is informed by the techniques of [HPS15]. We will use theorems about the matching polynomials of graphs that we will prove next lecture.

In the same way that a Ramanujan graph approximates the complete graph, a bipartite Ramanujan graph approximates a complete bipartite graph. We say that a d-regular graph is a bipartite Ramanujan graph if all of its adjacency matrix eigenvalues, other than d and $-d$, have absolute value at most $2\sqrt{d-1}$. The eigenvalue of d is a consequence of being d-regular and the eigenvalue of $-d$ is a consequence of being bipartite. In particular, recall that the adjacency matrix eigenvalues of a bipartite graph are symmetric about the origin. This is a special case of the following claim, which you can prove when you have a sparse moment.

Claim 41.1.1. *The eigenvalues of a symmetric matrix of the form*

$$\begin{pmatrix} \mathbf{0} & \boldsymbol{A} \\ \boldsymbol{A}^T & \mathbf{0} \end{pmatrix}$$

are symmetric about the origin.

We remark that one can derive bipartite Ramanujan graphs from ordinary Ramanujan graphs—just take the double cover. However, we do not know any way to derive ordinary Ramanujan graphs from the bipartite ones.

As opposed to reasoning directly about eigenvalues, we will work with characteristic polynomials. For a matrix $\boldsymbol{M}$, we write its characteristic polynomial in the variable x as

$$\chi_x(\boldsymbol{M}) \overset{\text{def}}{=} \det(x\boldsymbol{I} - \boldsymbol{M}).$$

41.2 2-Lifts

We saw 2-lifts of graphs in Problem 3 from Problem Set 2:

> We define a *signed adjacency matrix* of G to be a symmetric matrix $\boldsymbol{S}$ with the same nonzero pattern as the adjacency matrix $\boldsymbol{A}$, but such that each nonzero entry is either 1 or -1.
>
> We will use it to define a graph $G^{\boldsymbol{S}}$. Like the double-cover, the graph $G^{\boldsymbol{S}}$ will have two vertices for every vertex of G and two edges for every edge of G. For each edge $(u,v) \in E$, if $\boldsymbol{S}(u,v) = -1$ then $G^{\boldsymbol{S}}$ has the two edges
>
> $$(u_1, v_2) \quad \text{and} \quad (v_1, u_2),$$
>
> just like the double-cover. If $\boldsymbol{S}(u,v) = 1$, then $G^{\boldsymbol{S}}$ has the two edges
>
> $$(u_1, v_1) \quad \text{and} \quad (v_2, u_2).$$
>
> You should check that $G^{-\boldsymbol{A}}$ is the double-cover of G and that $G^{\boldsymbol{A}}$ consists of two disjoint copies of G.
>
> Prove that the eigenvalues of the adjacency matrix of $G^{\boldsymbol{S}}$ are the union of the eigenvalues of $\boldsymbol{A}$ and the eigenvalues of $\boldsymbol{S}$.

The graphs $G^{\boldsymbol{S}}$ that we form this way are called 2-lifts of G.

Bilu and Linial [BL06] conjectured that every d-regular graph G has a signed adjacency matrix $\boldsymbol{S}$ so that $\|\boldsymbol{S}\| \leq 2\sqrt{d-1}$. This would give a simple procedure for constructing infinite families of Ramanujan graphs. We would begin with any small d-regular Ramanujan graph, such as the complete graph on $d+1$ vertices. Then, given any d-regular Ramanujan graph we could construct a new Ramanujan graph on twice as many vertices by using $G^{\boldsymbol{S}}$ where $\|\boldsymbol{S}\| \leq 2\sqrt{d-1}$.

We will prove something close to their conjecture.

Theorem 41.2.1. *Every d-regular graph G has a signed adjacency matrix $\boldsymbol{S}$ for which the minimum eigenvalue of $\boldsymbol{S}$ is at least $-2\sqrt{d-1}$.*

We can use this theorem to build infinite families of bipartite Ramanujan graphs, because their eigenvalues are symmetric about the origin. Thus, if $\mu_n \geq -2\sqrt{d-1}$, then we know that $|\mu_i| \leq 2\sqrt{d-1}$ for all $1 < i < n$. Note that every 2-lift of a bipartite graph is also a bipartite graph.

41.3 Random 2-Lifts

We will prove Theorem 46.2.1 by considering a random 2-lift. In particular, we consider the expected characteristic polynomial of a random signed adjacency matrix $\boldsymbol{S}$:

$$\mathbb{E}_{\boldsymbol{S}}\left[\chi_x(\boldsymbol{S})\right]. \tag{41.1}$$

Godsil and Gutman [GG81] proved that this is equal to the matching polynomial of G! We will learn more about the matching polynomial next lecture.

For now, we just need the following bound on its zeros which was proved by Heilmann and Lieb [HL72].

Theorem 41.3.1. *The eigenvalues of the matching polynomial of a graph of maximum degree at most d are real and have absolute value at most $2\sqrt{d-1}$.*

Now that we know that the smallest zero of (46.1) is at least $-2\sqrt{d-1}$, all we need to do is to show that there is some signed adjacency matrix whose smallest eigenvalue is at least this bound. This is not necessarily as easy as it sounds, because the smallest zero of the average of two polynomials is not necessarily related to the smallest zeros of those polynomials. We will show that, in this case, it is.

41.4 Laplacianized Polynomials

Instead of directly reasoning about the characteristic polynomials of signed adjacency matrices $\boldsymbol{S}$, we will work with characteristic polynomials of $d\boldsymbol{I} - \boldsymbol{S}$. It suffices for us to prove that there exists an $\boldsymbol{S}$ for which the largest eigenvalue of $d\boldsymbol{I} - \boldsymbol{S}$ is at most $d + 2\sqrt{d-1}$.

Fix an ordering on the m edges of the graph, associate each $\boldsymbol{S}$ with a vector $\sigma \in \{\pm 1\}^m$, and define

$$p_\sigma(x) = \chi_x(d\boldsymbol{I} - \boldsymbol{S}).$$

The expected polynomial is the average of all these polynomials.

We define two vectors for each edge in the graph. If the ith edge is (a, b), then we define

$$\boldsymbol{v}_{i,\sigma_i} = \boldsymbol{\delta}_a - \sigma_i \boldsymbol{\delta}_b.$$

For every $\sigma \in \{\pm 1\}^m$, we have

$$\sum_{i=1}^{m} \boldsymbol{v}_{i,\sigma_i} \boldsymbol{v}_{i,\sigma_i}^T = d\boldsymbol{I} - \boldsymbol{S},$$

where $\boldsymbol{S}$ is the signed adjacency matrix corresponding to σ. So, for every $\sigma \in \{\pm 1\}^m$,

$$p_\sigma(x) = \chi_x \left(\sum_{i=1}^{m} \boldsymbol{v}_{i,\sigma_i} \boldsymbol{v}_{i,\sigma_i}^T \right).$$

41.5 Interlacing Families of Polynomials

Here is the problem we face. We have a large family of polynomials, say $p_1(x), \dots, p_m(x)$, for which we know each p_i is real-rooted and that their sum is real rooted. We would like to show that there is some polynomial p_i whose largest zero is at most the largest zero of the sum. This is not true in general. But, it is true in our case because the polynomials form an *interlacing family.*

For a polynomial $p(x) = \prod_{i=1}^n (x - \lambda_i)$ of degree n and a polynomial $q(x) = \prod_{i=1}^{n-1} (x - \mu_i)$ of degree $n-1$, we say that $q(x)$ *interlaces* $p(x)$ if

$$\lambda_n \leq \mu_{n-1} \leq \lambda_{n-1} \leq \cdots \leq \lambda_2 \leq \mu_1 \leq \lambda_1.$$

If $r(x) = \prod_{i=1}^n (x - \mu_i)$ has degree n, we write $r(x) \to p(x)$ if

$$\mu_n \leq \lambda_n \leq \mu_{n-1} \leq \cdots \leq \lambda_2 \leq \mu_1 \leq \lambda_1.$$

That is, if the zeros of p and r interlace, with the zeros of p being larger. We also make these statements if they hold of positive multiples of p, r and q.

The following lemma gives the examples of interlacing polynomials that motivate us.

Lemma 41.5.1. *Let $\boldsymbol{A}$ be a symmetric matrix and let $\boldsymbol{v}$ be a vector. For a real number t let*

$$p_t(x) = \chi_x(\boldsymbol{A} + t\boldsymbol{v}\boldsymbol{v}^T).$$

Then, for $t > 0$, $p_0(x) \to p_t(x)$ and there is a monic[1] *degree $n-1$ polynomial $q(x)$ so that for all t*

$$p_t(x) = \chi_x(\boldsymbol{A}) - tq(x).$$

Proof. The fact that $p_0(x) \to p_t(x)$ for $t > 0$ follows from the Courant-Fischer Theorem.

We first establish the existence of $q(x)$ in the case that $\boldsymbol{v} = \boldsymbol{\delta}_1$. As the matrix $t\boldsymbol{\delta}_1\boldsymbol{\delta}_1^T$ is zeros everywhere except for the element t in the upper left entry and the determinant is linear in each entry of the matrix,

$$\chi_x(\boldsymbol{A} + t\boldsymbol{\delta}_1\boldsymbol{\delta}_1^T) = \det(x\boldsymbol{I} - \boldsymbol{A} - t\boldsymbol{\delta}_1\boldsymbol{\delta}_1^T) = \det(x\boldsymbol{I} - \boldsymbol{A}) - t\det(x\boldsymbol{I}^{(1)} - \boldsymbol{A}^{(1)}) = \chi_x(\boldsymbol{A}) - t\chi_x(\boldsymbol{A}^{(1)}),$$

[1] A *monic* polynomial is one whose leading coefficient is 1.

where $\boldsymbol{A}^{(1)}$ is the submatrix of $\boldsymbol{A}$ obtained by removing its first row and column. The polynomial $q(x) = \chi_x(\boldsymbol{A}^{(1)})$ has degree $n-1$.

For arbitrary, $\boldsymbol{v}$, let $\boldsymbol{Q}$ be a rotation matrix for which $\boldsymbol{Q}\boldsymbol{v} = \boldsymbol{\delta}_1$. As determinants, and thus characteristic polynomials, are unchanged by multiplication by rotation matrices,

$$\chi_x(\boldsymbol{A} + t\boldsymbol{v}\boldsymbol{v}^T) = \chi_x(\boldsymbol{Q}(\boldsymbol{A} + t\boldsymbol{v}\boldsymbol{v}^T)\boldsymbol{Q}^T)$$
$$= \chi_x(\boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}^T + t\boldsymbol{\delta}_1\boldsymbol{\delta}_1^T)) = \chi_x(\boldsymbol{Q}\boldsymbol{A}\boldsymbol{Q}^T) - tq(x) = \chi_x(\boldsymbol{A}) - tq(x),$$

for some $q(x)$ of degree $n-1$. □

For a polynomial p, let $\lambda_{max}(p)$ denote its largest zero. When polynomials interlace, we can relate the largest zero of their sum to the largest zero of at least one of them.

Lemma 41.5.2. *Let $p_1(x)$, $p_2(x)$ and $r(x)$ be polynomials so that $r(x) \to p_i(x)$. Then, $r(x) \to p_1(x) + p_2(x)$ and there is an $i \in \{1, 2\}$ for which*

$$\lambda_{max}(p_i) \leq \lambda_{max}(p_1 + p_2).$$

Proof. Let μ_1 be the largest zero of $r(x)$. As each polynomial $p_i(x)$ has a positive leading coefficient, each is eventually positive and so is their sum. As each has exactly one zero that is at least μ_1 each is nonpositive at μ_1, and the same is also true of their sum. Let λ be the largest zero of $p_1 + p_2$. We have established that $\lambda \geq \mu_1$.

If $p_i(\lambda) = 0$ for some i, then we are done. If not, there is an i for which $p_i(\lambda) > 0$. As p_i only has one zero larger than μ_1, and it is eventually positive, the largest zero of p_i must be less than λ. □

If $p_1, \dots, p_m$ are polynomials such that there exists an $r(x)$ for which $r(x) \to p_i(x)$ for all i, then these polynomials are said to have a *common interlacing*. Such polynomials satisfy the natural generalization of Lemma 44.3.1.

The polynomials $p_\sigma(x)$ do not all have a common interlacing. However, they satisfy a property that is just as useful: they form an *interlacing family*. Rather than defining these in general, we will just explain the special case we need for today's theorem.

We define polynomials that correspond to fixing the signs of the first k edges and then choosing the rest at random. We indicate these by shorter sequences $\sigma \in \{\pm 1\}^k$. For $k < m$ and $\sigma \in \{\pm 1\}^k$ we define

$$p_\sigma(x) \stackrel{\text{def}}{=} \mathbb{E}_{\rho \in \{\pm 1\}^{n-k}} \left[p_{\sigma,\rho}(x) \right].$$

So,

$$p_\emptyset(x) = \mathbb{E}_{\sigma \in \{\pm 1\}^m} \left[p_\sigma(x) \right].$$

We view the strings σ, and thus the polynomials p_σ, as vertices in a complete binary tree. The nodes with σ of length m are the leaves, and $\emptyset$ corresponds to the root. For σ of length less than n, the children of σ are $(\sigma, 1)$ and $(\sigma, -1)$. We call such a pair of nodes *siblings*. We will eventually prove in Lemma 41.6.1 that all the polynomials $p_\sigma(x)$ are real rooted and in Corollary 41.6.2 that every pair of siblings has a common interlacing.

But first, we show that this implies that there is a leaf indexed by $\sigma \in \{\pm 1\}^m$ for which

$$\lambda_{max}(p_\sigma) \leq \lambda_{max}(p_\emptyset).$$

This implies Theorem 46.2.1, as we know from Theorem 41.3.1 that $\lambda_{max}(p_\emptyset) \leq d + 2\sqrt{d-1}$.

Lemma 41.5.3. *There is a $\sigma \in \{\pm 1\}^m$ for which*

$$\lambda_{max}(p_\sigma) \leq \lambda_{max}(p_\emptyset).$$

Proof. Corollary 41.6.2 and Lemma 44.3.1 imply that every non-leaf node in the tree has a child whose largest zero is at most the largest zero of that node. Starting at the root of the tree, we find a node whose largest zero is at most the largest zero of $p_\emptyset$. We then proceed down the tree until we reach a leaf, at each step finding a node labeled by a polynomial whose largest zero is at most the largest zero of the previous polynomial. The leaf we reach, σ, satisfies the desired inequality. □

41.6 Common Interlacings

We can now use Lemmas 46.3.2 and 44.3.1 to show that every $\sigma \in \{\pm 1\}^{m-1}$ has a child (σ, s) for which $\lambda_{max}(p_{\sigma,s}) \leq \lambda_{max}(p_\sigma)$. Let

$$\boldsymbol{A} = \sum_{i=1}^{m-1} \boldsymbol{v}_{i,\sigma_i} \boldsymbol{v}_{i,\sigma_i}^T.$$

The children of σ, $(\sigma, 1)$ and $(\sigma, -1)$ have polynomials $p_{(\sigma,1)}$ and $p_{(\sigma,-1)}$ that equal

$$\chi_x(\boldsymbol{A} + \boldsymbol{v}_{m,1}\boldsymbol{v}_{m,1}^T) \quad \text{and} \quad \chi_x(\boldsymbol{A} + \boldsymbol{v}_{m,-1}\boldsymbol{v}_{m,-1}^T).$$

By Lemma 46.3.2, $\chi_x(\boldsymbol{A}) \to \chi_x(\boldsymbol{A} + \boldsymbol{v}_{m,s}\boldsymbol{v}_{m,s}^T)$ for $s \in \{\pm 1\}$, and Lemma 44.3.1 implies that there is an s for which the largest zero of $p_{(\sigma,s)}$ is at most the largest zero of their average, which is p_σ.

To extend this argument to nodes higher up in the tree, we will prove the following statement.

Lemma 41.6.1. *Let $\boldsymbol{A}$ be a symmetric matrix and let $\boldsymbol{w}_{i,s}$ be vectors for $1 \leq i \leq k$ and $s \in \{0,1\}$. Then the polynomial*

$$\sum_{\rho \in \{0,1\}^k} \chi_x\left(\boldsymbol{A} + \sum_{i=1}^{k} \boldsymbol{w}_{i,\rho_i}\boldsymbol{w}_{i,\rho_i}^T\right)$$

is real rooted, and for each $s \in \{0,1\}$,

$$\sum_{\rho \in \{0,1\}^k} \chi_x\left(\boldsymbol{A} + \sum_{i=1}^{k-1} \boldsymbol{w}_{i,\rho_i}\boldsymbol{w}_{i,\rho_i}^T\right) \to \sum_{\rho \in \{0,1\}^k} \chi_x\left(\boldsymbol{A} + \sum_{i=1}^{k-1} \boldsymbol{w}_{i,\rho_i}\boldsymbol{w}_{i,\rho_i}^T + \boldsymbol{w}_{k,s}\boldsymbol{w}_{k,s}^T\right).$$

Corollary 41.6.2. *For every $k < n$ and $\sigma \in \{\pm 1\}^k$, the polynomials $p_{\sigma,s}(x)$ for $s \in \{\pm 1\}$ are real rooted and have a common interlacing.*

41.7 Real Rootedness

To prove Lemma 41.6.1, we use the following two lemmas which are known collectively as Obreschkoff's Theorem [Obr63].

Lemma 41.7.1. *Let p and q be polynomials of degree n and $n-1$, and let $p_t(x) = p(x) - tq(x)$. If p_t is real rooted for all $t \in \mathbb{R}$, then q interlaces p.*

Proof Sketch. Recall that the roots of a polynomial are continuous functions of its coefficients, and thus the roots of p_t are continuous functions of t. We will use this fact to obtain a contradiction.

For simplicity,[2] I just consider the case in which all of the roots of p and q are distinct. If they are not, one can prove this by dividing out their common divisors.

If p and q do not interlace, then p must have two roots that do not have a root of q between them. Let these roots of p be λ_{i+1} and λ_i. Assume, without loss of generality, that both p and q are positive between these roots. We now consider the behavior of p_t for positive t.

As we have assumed that the roots of p and q are distinct, q is positive at these roots, and so p_t is negative at λ_{i+1} and λ_i. If t is very small, then p_t will be close to p in value, and so there must be some small t_0 for which $p_{t_0}(x) > 0$ for some $\lambda_{i+1} < x < \lambda_i$. This means that p_{t_0} must have two roots between λ_{i+1} and λ_i.

As q is positive on the entire closed interval $[\lambda_{i+1}, \lambda_i]$, when t is large p_t will be negative on this entire interval, and thus have no roots inside. As we vary t between t_0 and infinity, the two roots at t_0 must vary continuously and cannot cross λ_{i+1} or λ_i. This means that they must become complex, contradicting our assumption that p_t is always real rooted. □

Lemma 41.7.2. *Let p and q be polynomials of degree n and $n-1$ that interlace and have positive leading coefficients. For every $t > 0$, define $p_t(x) = p(x) - tq(x)$. Then, $p_t(x)$ is real rooted and*

$$p(x) \rightarrow p_t(x).$$

Proof Sketch. For simplicity, I consider the case in which all of the roots of p and q are distinct. One can prove the general case by dividing out the common repeated roots.

To see that the largest root of p_t is larger than λ_1, note that $q(x)$ is positive for all $x > \mu_1$, and $\lambda_1 > \mu_1$. So, $p_t(\lambda_1) = p(\lambda_1) - tq(\lambda_1) < 0$. As p_t is monic, it is eventually positive and it must have a root larger than λ_1.

We will now show that for every $i \geq 1$, p_t has a root between λ_{i+1} and λ_i. As this gives us $d-1$ more roots, it accounts for all d roots of p_t. For i odd, we know that $q(\lambda_i) > 0$ and $q(\lambda_{i+1}) < 0$. As p is zero at both of these points, $p_t(\lambda_i) > 0$ and $p_t(\lambda_{i+1}) < 0$, which means that p_t has a root between λ_i and λ_{i+1}. The case of even i is similar. □

Lemma 41.7.3. *Let $p_0(x)$ and $p_1(x)$ be degree n monic polynomials for which there is a third polynomial $r(x)$ Such that*

$$r(x) \rightarrow p_0(x) \quad \text{and} \quad r(x) \rightarrow p_1(x).$$

[2] I thank Sushant Sachdeva for helping me work out this particularly simple proof.

Then

$$r(x) \to (1/2)p_0(x) + (1/2)p_1(x),$$

and the latter is a real rooted polynomial.

Sketch. Assume for simplicity that all the roots of r are distinct and different from the roots of p_0 and p_1. Let $\mu_n < \mu_{n-1} < \cdots < \mu_1$ be the roots of r. Our assumptions imply that both p_0 and p_1 are negative at μ_i for odd i and positive for even i. So, the same is true of their average. This tells us that their average must have at least $n-1$ real roots between μ_n and μ_1. As their average is monic, it must be eventually positive and so must have a root larger than μ_1. That accounts for all n of its roots. □

Proof of Lemma 41.6.1. We prove this by induction on k. Assuming that we have proved it for $k-1$, we now prove it for k. Let $\boldsymbol{u}$ be any vector and let $t \in \mathbb{R}$. Define

$$p_t(x) = \sum_{\rho \in \{0,1\}^k} \chi_x \left(\boldsymbol{A} + \sum_{i=1}^{k-1} \boldsymbol{w}_{i,\rho_i} \boldsymbol{w}_{i,\rho_i}^T + t\boldsymbol{u}\boldsymbol{u}^T \right).$$

By Lemma 46.3.2, we can express this polynomial in the form

$$p_t(x) = p_0(x) - tq(x),$$

where q has positive leading coefficient and degree $n-1$. By absorbing $t\boldsymbol{u}\boldsymbol{u}^T$ into $\boldsymbol{A}$ we may use induction on k to show that $p_t(x)$ is real rooted for all t. Thus, Lemma 46.3.3 implies that $q(x)$ interlaces $p_0(x)$, and Lemma 46.3.4 tells us that for $t > 0$

$$p_0(x) \to p_t(x).$$

So, we may conclude that for every $s \in \{\pm 1\}$,

$$\sum_{\rho \in \{0,1\}^{k-1}} \chi_x \left(\boldsymbol{A} + \sum_{i=1}^{k-1} \boldsymbol{w}_{i,\rho_i} \boldsymbol{w}_{i,\rho_i}^T \right) \to \sum_{\rho \in \{0,1\}^k} \chi_x \left(\boldsymbol{A} + \sum_{i=1}^{k-1} \boldsymbol{w}_{i,\rho_i} \boldsymbol{w}_{i,\rho_i}^T + \boldsymbol{w}_{k,s} \boldsymbol{w}_{k,s}^T \right).$$

So, Lemma 46.3.5 implies that

$$\sum_{\rho \in \{0,1\}^{k-1}} \chi_x \left(\boldsymbol{A} + \sum_{i=1}^{k-1} \boldsymbol{w}_{i,\rho_i} \boldsymbol{w}_{i,\rho_i}^T \right) \to \sum_{\rho \in \{0,1\}^k} \chi_x \left(\boldsymbol{A} + \sum_{i=1}^{k} \boldsymbol{w}_{i,\rho_i} \boldsymbol{w}_{i,\rho_i}^T \right)$$

and that the latter polynomial is real rooted. □

41.8 Conclusion

The major open problem left by this work is establishing the existence of regular (non-bipartite) Ramanujan graphs. The reason we can not prove this using the techniques in this lecture is that the interlacing techniques only allow us to reason about the largest or smallest eigenvalue of a matrix, but not both.

To see related papers establishing the existence of Ramanujan graphs, see [MSS15d, HPS15]. For a survey on this and related material, see [MSS14].

41.9 Overview

The coefficients of the matching polynomial of a graph count the numbers of matchings of various sizes in that graph. It was first defined by Heilmann and Lieb [HL72], who proved that it has some amazing properties, including that it is real rooted. They also proved that all roots of the matching polynomial of a graph of maximum degree d are at most $2\sqrt{d-1}$. Our proofs today come from a different approach to the matching polynomial that appears in the work of Godsil [God93, God81]. A theorem of Godsil and Gutman [GG81] implies that the expected characteristic polynomial of a randomly signed adjacency matrix is the matching polynomial of a graph. Last lecture we used these results to establish the existence of infinite families of bipartite Ramanujan graphs.

41.10 $2\sqrt{d-1}$

We begin by explaining where the number $2\sqrt{d-1}$ comes from: it is an upper bound on the eigenvalues of a tree of maximum degree at most d. One can also show that the largest eigenvalue of an d-ary tree approaches $2\sqrt{d-1}$ as the depth of the tree (and number of vertices) increases.

We prove this statement in two steps. The first is similar to proofs we saw at the beginning of the semester.

Lemma 41.10.1. *Let $\boldsymbol{M}$ be a (not necessarily symmetric) nonnegative matrix. Let $s = \|M\mathbf{1}\|_\infty$ be the maximum row sum of $\boldsymbol{M}$. Then, $|\lambda| \leq s$ for every eigenvalue of $\boldsymbol{M}$.*

Proof. Let $\boldsymbol{M}\boldsymbol{\psi} = \lambda\boldsymbol{\psi}$, and let a be an entry of $\boldsymbol{\psi}$ of largest absolute value. Then,

$$\begin{aligned}
|\lambda|\,|\boldsymbol{\psi}(a)| &= |\lambda\boldsymbol{\psi}(a)| \\
&= |(\boldsymbol{M}\boldsymbol{\psi})(a)| \\
&= \left|\sum_b \boldsymbol{M}(b,a)\boldsymbol{\psi}(a)\right| \\
&\leq \left|\sum_b \boldsymbol{M}(b,a)\right| |\boldsymbol{\psi}(a)| \\
&\leq s\,|\boldsymbol{\psi}(a)|\,.
\end{aligned}$$

This implies $|\lambda| \leq s$. □

Theorem 41.10.2. *Let T be a tree in which every vertex has degree at most d. Then, all eigenvalues of $\chi_x(\boldsymbol{M}_T)$ have absolute value at most $2\sqrt{d-1}$.*

Proof. Let $\boldsymbol{M}$ be the adjacency matrix of T. Choose some vertex to be the root of the tree, and define its height to be 0. For every other vertex a, define its height, $h(a)$, to be its distance to the root. Define $\boldsymbol{D}$ to be the diagonal matrix with

$$\boldsymbol{D}(a,a) = \left(\sqrt{d-1}\right)^{h(a)}.$$

Recall that the eigenvalues of $\boldsymbol{M}$ are the same as the eigenvalues of $\boldsymbol{D}\boldsymbol{M}\boldsymbol{D}^{-1}$. We will use the fact that all eigenvalues of a nonnegative matrix are upper bounded in absolute value by its maximum row sum.

So, we need to prove that all row sums of $\boldsymbol{D}\boldsymbol{M}\boldsymbol{D}^{-1}$ are at most $2\sqrt{d-1}$. There are three types of vertices to consider. First, the row of the root has up to d entries that are all $1/\sqrt{d-1}$. For $d \geq 2$, $d/\sqrt{d-1} \leq 2\sqrt{d-1}$. The intermediate vertices have one entry in their row that equals $\sqrt{d-1}$, and up to $d-1$ entries that are equal to $1/\sqrt{d-1}$, for a total of $2\sqrt{d-1}$. Finally, every leaf only has one nonzero entry in its row, and that entry equals $\sqrt{d-1}$. □

41.11 The Matching Polynomial

A matching in a graph $G = (V, E)$ is a subgraph of G in which every vertex has degree 1. We say that a matching has size k if it has k edges. We let

$$m_k(G)$$

denote the number of matchings in G of size k. Throughout this lecture, we let $|V| = n$. Observe that $m_1(G)$ is the number of edges in G, and that $m_{n/2}(G)$ is the number of perfect matchings in G. By convention we set $m_0(G) = 1$, as the empty set is matching with no edges. Computing the number of perfect matchings is a $\#P$-hard problem [Val79]. This means that it is much harder than solving NP-hard problems, so you shouldn't expect to do it quickly on large graphs.

The *matching polynomial of* G, written $\mu_x[G]$, is

$$\mu_x[G] \overset{\text{def}}{=} \sum_{k=0}^{n/2} x^{n-2k}(-1)^k m_k(G).$$

Our convention that $m_0(G) = 1$ ensures that this is a polynomial of degree n.

This is a fundamental example of a polynomial that is defined so that its coefficients count something. When the "something" is interesting, the polynomial usually is as well.

Godsil and Gutman [GG81] proved that this is equal to the matching polynomial of G!

Lemma 41.11.1. *Let G be a graph and let $\boldsymbol{S}$ be a uniform random signed adjacency matrix of G. Then,*

$$\mathbb{E}\left[\chi_x(\boldsymbol{S})\right] = \mu_x[G].$$

Proof. Expand the expected characterstic polynomial as

$$\begin{aligned}\mathbb{E}\left[\chi_x(\boldsymbol{S})\right] &= \mathbb{E}\left[\det(x\boldsymbol{I} - \boldsymbol{S})\right] \\ &= \mathbb{E}\left[\det(x\boldsymbol{I} + \boldsymbol{S})\right] \\ &= \mathbb{E}\left[\sum_{\pi \in S_n} \mathsf{sgn}(\pi) x^{|\{a : \pi(a) = a\}|} \prod_{a : \pi(a) \neq a} (\boldsymbol{S}(a, \pi(a))).\right] \\ &= \sum_{\pi \in S_n} \mathsf{sgn}(\pi) x^{|\{a : \pi(a) = a\}|} \mathbb{E}\left[\prod_{a : \pi(a) \neq a} (\boldsymbol{S}(a, \pi(a)))\right].\end{aligned}$$

As $\mathbb{E}\left[\boldsymbol{S}(a, \pi(a))\right] = 0$ for every a so that $\pi(a) \neq a$, the only way we can get a nonzero contribution from a permutation π is if for all a so that $\pi(a) \neq a$,

a. $(a, \pi(a)) \in E$, and

b. $\pi(\pi(a)) = a$.

The latter condition guarantees that whenever $\boldsymbol{S}(a, \pi(a))$ appears in the product, $\boldsymbol{S}(\pi(a), a)$ does as well. As these entries are constrained to be the same, their product is 1.

Thus, the only permtuations that count are the involuations (the permutations in which all cycles have length 1 or 2). These correspond exactly to the matchings in the graph. Finally, the sign of an involution is exactly its number of two-cycles, which is exactly its number of edges. □

We will prove that the matching polynomial of every d-regular graph divides the matching polynomial of a larger tree of maximum degree d.

The matching polynomials of trees are very special—they are exactly the same as the characteristic polynomial of the adjacency matrix.

Theorem 41.11.2. *Let G be a tree and let $\boldsymbol{M}$ be its adjacency matrix. Then*

$$\mu_x\left[G\right] = \chi_x(\boldsymbol{M}).$$

Proof. Expand

$$\chi_x(\boldsymbol{M}) = \det(x\boldsymbol{I} - \boldsymbol{M})$$

by summing over permutations. We obtain

$$\sum_{\pi \in S_n} \mathsf{sgn}(\pi) x^{|\{a : \pi(a) = a\}|} \prod_{a : \pi(a) \neq a} (-\boldsymbol{M}(a, \pi(a))).$$

We will prove that the only permutations that contribute to this sum are those for which $\pi(\pi(a)) = a$ for every a. And, these correspond to matchings.

If π is a permutation for which there is an a so that $\pi(\pi(a)) \neq a$, then there are $a = a_1, \dots, a_k$ with $k > 2$ so that $\pi(a_i) = a_{i+1}$ for $1 \leq i < k$, and $\pi(a_k) = a_1$. For this term to contribute, it must be the case that $\boldsymbol{M}(a_i, a_{i+1}) = 1$ for all i, and that $\boldsymbol{M}(a_k, a_1) = 1$. For $k > 2$, this would be a cycle of length k in G. However, G is a tree and so cannot have a cycle.

So, the only permutations that contribute are the *involutions*: the permutations π that are their own inverse. An involution has only fixed points and cycles of length 2. Each cycle of length 2 that contributes a nonzero term corresponds to an edge in the graph. Thus, the number of permutations with k cycles of length 2 is equal to the number of matchings with k edges. As the sign of an involution with k cycles of length 2 is $(-1)^k$, the coefficient of x^{n-2k} is $(-1)^k m_k(G)$. □

41.12 Properties of the Matching Polynomial

We begin by establishing some fundamental properties of the matching polynomial. For graphs G and H on different vertex sets, we write $G \cup H$ for their disjoint union.

Lemma 41.12.1. *Let G and H be graphs on different vertex sets. Then,*

$$\mu_x [G \cup H] = \mu_x [G] \, \mu_x [H] \, .$$

Proof. Every matching in $G \cup H$ is the union of a matching in G and a matching in H. Thus,

$$m_k(G \cup H) = \sum_{j=0}^{k} m_j(G) m_{k-j}(H).$$

The lemma follows. □

For a a vertex of $G = (V, E)$, we write $G - a$ for the graph $G(V - \{a\})$. This notation will prove very useful when reasoning about matching polynmomials. Fix a vertex a of G, and divide the matchings in G into two classes: those that involve vertex a and those that do not. The number of matchings of size k that do not involve a is $m_k(G - a)$. On the other hand, those that do involve a connect a to one of its neighbors. To count these, we enumerate the neighbors b of a. A matching of size k that includes edge (a, b) can be written as the union of (a, b) and a matching of size $k - 1$ in $G - a - b$. So, the number of matchings that involve a is

$$\sum_{b \sim a} m_{k-1}(G - a - b).$$

This gives a recurrence for the number of matchings of size k in G:

$$m_k(G) = m_k(G - a) + \sum_{b \sim a} m_{k-1}(G - a - b).$$

To turn this into a recurrence for $\mu_x [G]$, write

$$x^{n-2k}(-1)^k m_k(G) = x \cdot x^{(n-1)-2k}(-1)^k m_k(G - a) - x^{(n-2)-2(k-1)}(-1)^{k-1} m_{k-1}(G - a - b).$$

This establishes the following formula.

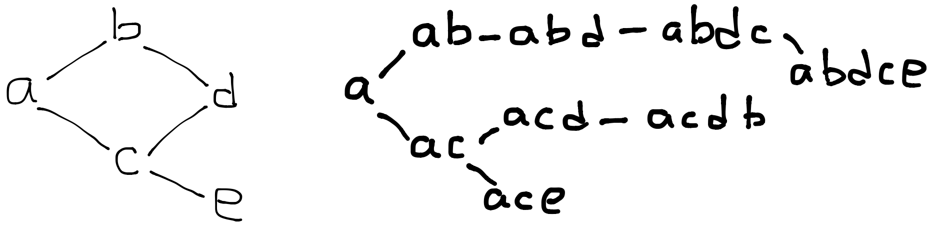
Lemma 41.12.2.

$$\mu_x[G] = x\mu_x[G-a] - \sum_{b\sim a} \mu_x[G-a-b].$$

41.13 The Path Tree

Godsil proves that the matching polynomial of a graph is real rooted by proving that it divides the matching polynomial of a tree. Moreover, the maximum degree of vertices in the tree is at most the maximum degree of vertices in the graph. As the matching polynomial of a tree is the same as its characteristic polynomial, and all zeros of the characteristic polynomial of a tree of maximum degree at most d have absolute value at most $2\sqrt{d-1}$, all the zeros of the matching polynomial of a d-regular graph have absolute value at most $2\sqrt{d-1}$.

The tree that Godsil uses is the *path tree of* G starting at a vertex of G. For a a vertex of G, the path tree of G starting at a, written $T_a(G)$ is a tree whose vertices correspond to paths in G that start at a and do not contain any vertex twice. One path is connected to another if one extends the other by one vertex. For example, here is a graph and its path tree starting at a.



When G is a tree, $T_a(G)$ is isomorphic to G.

Godsil's proof begins by deriving a somewhat strange equality. Since I haven't yet found a better proof, I'll take this route too.

Theorem 41.13.1. *For every graph G and vertex a of G,*

$$\frac{\mu_x[G]}{\mu_x[G-a]} = \frac{\mu_x[T_a(G)]}{\mu_x[T_a(G)-a]}.$$

The term on the upper-right hand side is a little odd. It is a forrest obtained by removing the root of the tree $T_a(G)$. We may write it as a disjoint union of trees as

$$T_a(G) - a = \bigcup_{b\sim a} T_b(G-a).$$

Before proving this, we use it to prove our main theorem.

Theorem 41.13.2. *For every vertex a of G, the polynomial $\mu_x[G]$ divides the polynomial $\mu_x[T_a(G)]$.*

Proof. We prove this by induction on the number of vertices in G, using as our base case graphs with at most 2 vertices. We then know, by induction, that for $b \sim a$,

$$\mu_x\left[G-a\right] \quad \text{divides} \quad \mu_x\left[T_b(G-a)\right].$$

As

$$T_a(G) - a = \cup_{b\sim a} T_b(G-a),$$

$$\mu_x\left[T_b(G-a)\right] \quad \text{divides} \quad \mu_x\left[T_a(G)-a\right].$$

Thus,

$$\mu_x\left[G-a\right] \quad \text{divides} \quad \mu_x\left[T_a(G)-a\right],$$

and so

$$\frac{\mu_x\left[T_a(G)-a\right]}{\mu_x\left[G-a\right]}$$

is a polynomial in x. To finish the proof, we apply Theorem 45.4.1, which implies

$$\mu_x\left[T_a(G)\right] = \mu_x\left[T_a(G)-a\right]\frac{\mu_x\left[G\right]}{\mu_x\left[G-a\right]} = \mu_x\left[G\right]\frac{\mu_x\left[T_a(G)-a\right]}{\mu_x\left[G-a\right]}.$$

□

Proof of Theorem 45.4.1. If G is a tree, then the left and right sides are identical, and so the equality holds. As the only graphs on less than 3 vertices are trees, the theorem holds for all graphs on at most 2 vertices. We will now prove it by induction on the number of vertices.

We may use Lemma 45.3.2 to expand the the left-hand side:

$$\frac{\mu_x\left[G\right]}{\mu_x\left[G-a\right]} = \frac{x\mu_x\left[G-a\right] - \sum_{b\sim a}\mu_x\left[G-a-b\right]}{\mu_x\left[G-a\right]} = x - \sum_{b\sim a}\frac{\mu_x\left[G-a-b\right]}{\mu_x\left[G-a\right]}.$$

By applying the inductive hypothesis to $G-a$, we see that this equals

$$x - \sum_{b\sim a}\frac{\mu_x\left[T_b(G-a)-b\right]}{\mu_x\left[T_b(G-a)\right]}. \tag{41.2}$$

To simplify this expression, we examine these graphs carefully. By the observtion we made before the proof,

$$T_b(G-a) - b = \bigcup_{c\sim b, c\neq a} T_c(G-a-b).$$

Similarly,

$$T_a(G) - a = \bigcup_{c\sim a} T_c(G-a),$$

which implies

$$\mu_x\left[T_a(G)-a\right] = \prod_{c\sim a}\mu_x\left[T_c(G-a)\right].$$

Let ab be the vertex in $T_a(G)$ corresponding to the path from a to b. We also have

$$\begin{aligned} T_a(G) - a - ab &= \left(\bigcup_{c \sim a, c \neq b} T_c(G-a) \right) \cup \left(\bigcup_{c \sim b, c \neq a} T_c(G-a-b) \right) \\ &= \left(\bigcup_{c \sim a, c \neq b} T_c(G-a) \right) \cup \left(T_b(G-a) - b \right). \end{aligned}$$

which implies

$$\mu_x \left[T_a(G) - a - ab \right] = \left(\prod_{c \sim a, c \neq b} \mu_x \left[T_c(G-a) \right] \right) \mu_x \left[T_b(G-a) - b \right].$$

Thus,

$$\begin{aligned} \frac{\mu_x \left[T_a(G) - a - ab \right]}{\mu_x \left[T_a(G) - a \right]} &= \frac{\left(\prod_{c \sim a, c \neq b} \mu_x \left[T_c(G-a) \right] \right) \mu_x \left[T_b(G-a) - b \right]}{\prod_{c \sim a} \mu_x \left[T_c(G-a) \right]} \\ &= \frac{\mu_x \left[T_b(G-a) - b \right]}{\mu_x \left[T_b(G-a) \right]}. \end{aligned}$$

Plugging this in to (45.1), we obtain

$$\begin{aligned} \frac{\mu_x [G]}{\mu_x [G-a]} &= x - \sum_{b \sim a} \frac{\mu_x \left[T_a(G) - a - ab \right]}{\mu_x \left[T_a(G) - a \right]} \\ &= \frac{x \mu_x \left[T_a(G) - a \right] - \sum_{b \sim a} \mu_x \left[T_a(G) - a - ab \right]}{\mu_x \left[T_a(G) - a \right]} \\ &= \frac{\mu_x \left[T_a(G) \right]}{\mu_x \left[T_a(G) - a \right]}. \end{aligned}$$

□

Chapter 42

Expected Characteristic Polynomials

Lecture 22 from November 18, 2015

42.1 Overview

Over the next few lectures, we will see two different proofs that infinite families of bipartite Ramanujan graphs exist. Both proofs will use the theory of interlacing polynomials, and will consider the expected characteristic polynomials of random matrices. In today's lecture, we will see a proof that some of these polynomials are real rooted.

At present, we do not know how to use these techniques to prove the existence of infinite families of non-bipartite Ramanujan graphs.

The material in today's lecture comes from [MSS15d], but the proof is inspired by the treatment of that work in [HPS15].

42.2 Random sums of graphs

We will build Ramanujan graphs on n vertices of degree d, for every d and even n. We begin by considering a random graph on n vertices of degree d. When n is even, the most natural way to generate such a graph is to choose d perfect matchings uniformly at random, and to then take their sum. I should mention one caveat: some edge could appear in many of the matchings. In this case, we add the weights of the corresponding edges together. So, the weight of an edge is the number of matchings in which it appears.

Let $\boldsymbol{M}$ be the adjacency matrix of some perfect matching on n vertices. We can generate the adjacency matrix of a random perfect matching by choosing a permutation matrix Π uniformly at random, and then forming $\Pi \boldsymbol{M} \Pi^T$. The sum of d independent uniform random perfect machings is then

$$\sum_{i=1}^{d} \Pi_i \boldsymbol{M} \Pi_i^T .$$

In today's lecture, we will consider the *expected characteristic polynomial* of such a graph. For a matrix $\boldsymbol{M}$, we let

$$\chi_x(\boldsymbol{M}) \stackrel{\text{def}}{=} \det(x\boldsymbol{I} - \boldsymbol{M})$$

denote the characteristic polynomial of $\boldsymbol{M}$ in the variable x.

For simplicity, we will consider the expected polynomial of the sum of just two graphs. For generality, we will let them be any graphs, or any symmetric matrices.

Our goal for today is to prove that these expected polynomials are real rooted.

Theorem 42.2.1. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be symmetric n-by-n matrices and let Π be a uniform random permutation. Then,*

$$\mathbb{E}_{\Pi}\left[\chi_x(\boldsymbol{A} + \Pi \boldsymbol{B} \Pi^T)\right]$$

has only real roots.

So that you will be surprised by this, I remind you that the sum of real rooted polynomials might have no real roots. For example, both $(x-2)^2$ and $(x+2)^2$ have only real roots, but their sum, $2x^2 + 8$, has no real roots.

Theorem 42.2.1 also holds for sums of many matrices. But, for simplicity, we restrict ourselves to considering the sum of two.

42.3 Interlacing

Our first tool for establishing real rootedness of polynomials is interlacing.

If $p(x)$ is a real rooted polynomial of degree n and $q(x)$ is a real rooted polynomial of degree $n-1$, then we say that p and q *interlace* if p has roots $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ and q has roots $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_{n-1}$ that satisfy

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \cdots \geq \lambda_{n-1} \geq \mu_{n-1} \geq \lambda_n .$$

We have seen two important examples of interlacing in this class so far. A real rooted polynomial and its derivative interlace. Similarly, the characteristic polynomial of a symmetric matrix and the characteristic polynomial of a principal submatrix interlace.

When p and q have the same degree, we also say that they interlace if their roots alternate. But, now there are two ways in which their roots can do so, depending on which polynomial has the largest root. If

$$p(x) = \prod_{i=1}^{n} (x - \lambda_i) \quad \text{and} \quad q(x) = \prod_{i=1}^{n} (x - \mu_i),$$

we write $q \rightarrow p$ if p and q interlace and for every i the ith root of p is at least as large as the ith root of q. That is, if

$$\lambda_1 \geq \mu_1 \geq \lambda_2 \geq \mu_2 \geq \cdots \geq \lambda_n \geq \mu_n.$$

Lemma 42.3.1. *Let p and q be polynomials of degree n and $n-1$ that interlace and have positive leading coefficients. For every $t > 0$, define $p_t(x) = p(x) - tq(x)$. Then, $p_t(x)$ is real rooted and*

$$p(x) \rightarrow p_t(x).$$

Proof Sketch. For simplicity, I consider the case in which all of the roots of p and q are distinct. One can prove the general case by dividing out the common repeated roots.

To see that the largest root of p_t is larger than λ_1, note that $q(x)$ is positive for all $x > \mu_1$, and $\lambda_1 > \mu_1$. So, $p_t(\lambda_1) = p(\lambda_1) - tq(\lambda_1) < 0$. As p_t is monic, it is eventually positive and it must have a root larger than λ_1.

We will now show that for every $i \geq 1$, p_t has a root between λ_{i+1} and λ_i. As this gives us $d-1$ more roots, it accounts for all d roots of p_t. For i odd, we know that $q(\lambda_i) > 0$ and $q(\lambda_{i+1}) < 0$. As p is zero at both of these points, $p_t(\lambda_i) > 0$ and $p_t(\lambda_{i+1}) < 0$, which means that p_t has a root between λ_i and λ_{i+1}. The case of even i is similar. □

The converse of this theorem is also true.

Lemma 42.3.2. *Let p and q be polynomials of degree n and $n-1$, and let $p_t(x) = p(x) - tq(x)$. If p_t is real rooted for all $t \in \mathbb{R}$, then p and q interlace.*

Proof Sketch. Recall that the roots of a polynomial are continuous functions of its coefficients, and thus the roots of p_t are continuous functions of t. We will use this fact to obtain a contradiction.

For simplicity,[1] I again just consider the case in which all of the roots of p and q are distinct.

If p and q do not interlace, then p must have two roots that do not have a root of q between them. Let these roots of p be λ_{i+1} and λ_i. Assume, without loss of generality, that both p and q are positive between these roots. We now consider the behavior of p_t for positive t.

As we have assumed that the roots of p and q are distinct, q is positive at these roots, and so p_t is negative at λ_{i+1} and λ_i. If t is very small, then p_t will be close to p in value, and so there must be some small t_0 for which $p_{t_0}(x) > 0$ for some $\lambda_{i+1} < x < \lambda_i$. This means that p_{t_0} must have two roots between λ_{i+1} and λ_i.

As q is positive on the entire closed interval $[\lambda_{i+1}, \lambda_i]$, when t is large p_t will be negative on this entire interval, and thus have no roots inside. As we vary t between t_0 and infinity, the two roots at t_0 must vary continuously and cannot cross λ_{i+1} or λ_i. This means that they must become complex, contradicting our assumption that p_t is always real rooted. □

Together, Lemmas 46.3.4 and 46.3.3 are known as Obreschkoff's Theorem [Obr63].

The following example will be critical.

[1] I thank Sushant Sachdeva for helping me work out this particularly simple proof.

Lemma 42.3.3. *Let $\boldsymbol{A}$ be an n-dimensional symmetric matrix and let $\boldsymbol{v}$ be a vector. Let*

$$p_t(x) = \chi_x(\boldsymbol{A} + t\boldsymbol{v}\boldsymbol{v}^T).$$

Then there is a degree $n-1$ polynomial $q(x)$ so that

$$p_t(x) = \chi_x(\boldsymbol{A}) - tq(x).$$

Proof. Consider the case in which $\boldsymbol{v} = \boldsymbol{\delta}_1$. It suffices to consider this case as determinants, and thus characteristic polynomials, are unchanged by multiplication by rotation matrices.

Then, we know that

$$\chi_x(\boldsymbol{A} + t\boldsymbol{\delta}_1\boldsymbol{\delta}_1^T) = \det(x\boldsymbol{I} - \boldsymbol{A} - t\boldsymbol{\delta}_1\boldsymbol{\delta}_1^T).$$

Now, the matrix $t\boldsymbol{\delta}_1\boldsymbol{\delta}_1^T$ is zeros everywhere except for the element t in the upper left entry. So,

$$\det(x\boldsymbol{I} - \boldsymbol{A} - t\boldsymbol{\delta}_1\boldsymbol{\delta}_1^T) = \det(x\boldsymbol{I} - \boldsymbol{A}) - t\det(x\boldsymbol{I}^{(1)} - \boldsymbol{A}^{(1)}) = \chi_x(\boldsymbol{A}) - t\chi_x(\boldsymbol{A}^{(1)}),$$

where $\boldsymbol{A}^{(1)}$ is the submatrix of $\boldsymbol{A}$ obtained by removing its first row and column. □

We know that $\chi_x(\boldsymbol{A} + t\boldsymbol{v}\boldsymbol{v}^T)$ is real rooted for all t, and we can easily show using the Courant Fischer Theorem that for $t > 0$ it interlaces $\chi_x(\boldsymbol{A})$ from above. Lemmas 46.3.4 and 46.3.3 tell us that these facts imply each other.

We need one other fact about interlacing polynomials.

Lemma 42.3.4. *Let $p_0(x)$ and $p_1(x)$ be two degree n monic polynomials for which there is a third polynomial $r(x)$ that has the same degree as p_0 and p_1 and so that*

$$p_0(x) \to r(x) \quad \text{and} \quad p_1(x) \to r(x).$$

Then for all $0 \leq s \leq 1$,

$$p_s(x) \overset{\text{def}}{=} sp_1(x) + (1-s)p_0(x)$$

is a real rooted polynomial.

Sketch. Assume for simplicity that all the roots of r are distinct. Let $\mu_1 > \mu_2 > \cdots > \mu_n$ be the roots of r. Our assumptions imply that both p_0 and p_1 are positive at μ_i for odd i and negative for even i. So, the same is true of their sum p_s. This tells us that p_s must have at least $n-1$ real roots.

We can also show that p_s has a root that is less than μ_n. One way to do it is to recall that the complex roots of a polynomial with real coefficients come in conjugate pairs. So, p_s can not have only one complex root. □

42.4 Sums of polynomials

Our goal is to show that

$$\sum_{\Pi \in S_n} \chi_x(\boldsymbol{A} + \Pi \boldsymbol{B} \Pi^T)$$

is a real rooted polynomial for all symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, where S_n is the set of n-by-n permuation matrices. We will do this by proving it for smaller sets of permutation matrices. To begin, we know it for $S = \{\boldsymbol{I}\}$. We will build up larger sets by swapping coordinates.

This will actually result in a distribution on permuations, so we consider $\sigma : S_n \rightarrow \mathbb{R}_{\geq 0}$ and consider sums of the form

$$\sum_{\Pi} \sigma(\Pi) \chi_x(\boldsymbol{A} + \Pi \boldsymbol{B} \Pi^T).$$

For coordinates i and j, let $\Gamma_{i,j}$ be the permutation matrix that just swaps i and j. We call such a permutation a swap. We need the following important fact about the action of swaps on matrices.

Lemma 42.4.1. *Let $\boldsymbol{A}$ be a symmetric matrix. Then, for all i and j, there are vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ so that*

$$\Gamma_{i,j} \boldsymbol{A} \Gamma_{i,j} = \boldsymbol{A} - \boldsymbol{u}\boldsymbol{u}^T + \boldsymbol{v}\boldsymbol{v}^T.$$

Proof. Without loss of generality, let $i = 1$ and $j = 2$. We prove that $\boldsymbol{A} - \Gamma_{i,j} \boldsymbol{A} \Gamma_{i,j}$ has rank 2 and trace 0.

We can write this difference in the form

$$\begin{bmatrix} a_{11} - a_{22} & a_{12} - a_{21} & a_{13} - a_{23} & a_{14} - a_{24} & \dots \\ a_{21} - a_{12} & a_{22} - a_{11} & a_{23} - a_{13} & a_{24} - a_{14} & \dots \\ a_{31} - a_{32} & a_{32} - a_{31} & 0 & \dots & \\ a_{41} - a_{42} & a_{42} - a_{41} & 0 & \dots & \\ \dots & & & & \end{bmatrix} = \begin{bmatrix} \alpha & \beta & \boldsymbol{y}^T \\ -\beta & -\alpha & -\boldsymbol{y}^T \\ \boldsymbol{y} & -\boldsymbol{y} & 0_{n-2} \end{bmatrix}$$

for some numbers α, β and some column vector $\boldsymbol{y}$ of length $n-2$. If $\alpha \neq \beta$ then the sum of the first two rows is equal to $(c, -c, 0, \dots, 0)$ for some $c \neq 0$, and every other row is a scalar multiple of this. On the other hand, if $\alpha = \beta$ then the first two rows are linearly dependent, and all of the other rows are multiples of $(1, -1, 0, \dots, 0)$. □

Lemma 42.4.2. *Let σ be such that for all symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$,*

$$p_x(\boldsymbol{A}, \boldsymbol{B}) \stackrel{\text{def}}{=} \sum_{\Pi \in S} \sigma(\Pi) \chi_x(\boldsymbol{A} + \Pi \boldsymbol{B} \Pi^T)$$

is real rooted. Then, for every $0 < s < 1$ and pair of vectors $\boldsymbol{u}$ and $\boldsymbol{v}$, for every symmetric $\boldsymbol{A}$ and $\boldsymbol{B}$ the polynomial

$$(1-s) p_x(\boldsymbol{A}, \boldsymbol{B}) + s p_x(\boldsymbol{A} - \boldsymbol{u}\boldsymbol{u}^T + \boldsymbol{v}\boldsymbol{v}^T, \boldsymbol{B})$$

is real rooted.

Proof. Define

$$r_t(x) = p_x(\boldsymbol{A} + t\boldsymbol{v}\boldsymbol{v}^T, \boldsymbol{B}).$$

By assumption, $r_t(x)$ is real rooted for every $t \in \mathbb{R}$. By Lemma 46.3.2, we can write

$$r_t(x) = r_0(x) - tq(x),$$

where $q(x)$ has degree $n-1$ and both r_0 and q have positive leading coefficients. So, by Lemma 46.3.3 $q(x)$ interlaces $r_0(x) = p_x(\boldsymbol{A}, \boldsymbol{B})$. Lemma 46.3.4 then tells us that

$$p_x(\boldsymbol{A}, \boldsymbol{B}) \to p_x(\boldsymbol{A} + \boldsymbol{v}\boldsymbol{v}^T, \boldsymbol{B}).$$

The same argument tells us that

$$p_x(\boldsymbol{A} - \boldsymbol{u}\boldsymbol{u}^T + \boldsymbol{v}\boldsymbol{v}^T, \boldsymbol{B}) \to p_x(\boldsymbol{A} + \boldsymbol{v}\boldsymbol{v}^T, \boldsymbol{B}).$$

This tells us that $p_x(\boldsymbol{A}, \boldsymbol{B})$ and $p_x(\boldsymbol{A} - \boldsymbol{u}\boldsymbol{u}^T + \boldsymbol{v}\boldsymbol{v}^T, \boldsymbol{B})$ both interlace $r_1(x)$ from below. We finish by applying Lemma 46.3.5 to conclude that every convex combination of these polynomials is real rooted. □

Corollary 42.4.3. *Let σ be such that for all symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$,*

$$p_x(\boldsymbol{A}, \boldsymbol{B}) \stackrel{\text{def}}{=} \sum_{\Pi \in S} \sigma(\Pi)\chi_x(\boldsymbol{A} + \Pi\boldsymbol{B}\Pi^T)$$

is real rooted. Then, for every $0 < s < 1$ and for every symmetric $\boldsymbol{A}$ and $\boldsymbol{B}$ the polynomial

$$\sum_{\Pi \in S} s\sigma(\Pi)\chi_x(\boldsymbol{A} + \Pi\boldsymbol{B}\Pi^T) + (1-s)\sigma(\Pi)\chi_x(\boldsymbol{A} + \Gamma_{i,j}\Pi\boldsymbol{B}\Pi^T\Gamma_{i,j}^T)$$

is real rooted.

Proof. Recall that

$$\chi_x(\boldsymbol{A} + \Gamma_{i,j}\Pi\boldsymbol{B}\Pi^T\Gamma_{i,j}^T) = \chi_x(\Gamma_{i,j}^T\boldsymbol{A}\Gamma_{i,j} + \Pi\boldsymbol{B}\Pi^T) = \chi_x(\Gamma_{i,j}\boldsymbol{A}\Gamma_{i,j}^T + \Pi\boldsymbol{B}\Pi^T).$$

The corollary now follows from the previous lemma. □

42.5 Random Swaps

We will build a random permutation out of random swaps. A random swap is specified by coordinates i and j and a swap probability s. It is a random matrix is that is equal to the identity with probability $1-s$ and $\Gamma_{i,j}$ with probability s. Let $\boldsymbol{S}$ be a random swap.

In the language of random swaps, we can express Corollary 42.5.1 as follows.

Corollary 42.5.1. *Let Π be a random permutation matrix drawn from a distribution so that for all symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$,*

$$\mathbb{E}\left[\chi_x(\boldsymbol{A} + \Pi \boldsymbol{B} \Pi^T)\right]$$

is real rooted. Let $\boldsymbol{S}$ be a random swap. Then,

$$\mathbb{E}\left[\chi_x(\boldsymbol{A} + \boldsymbol{S} \Pi \boldsymbol{B} \Pi^T \boldsymbol{S}^T)\right]$$

is real rooted for every symmetric $\boldsymbol{A}$ and $\boldsymbol{B}$.

All that remains is to show that a uniform random permutation can be assembled out of random swaps. The trick to doing this is to choose the random swaps with swap probabilities other than 1/2. If you didn't do this, it would be impossible as there are $n!$ permutations, which is not a power of 2.

Lemma 42.5.2. *For every n, there exists a finite sequence of random swaps $\boldsymbol{S}_1, \dots, \boldsymbol{S}_k$ so that*

$$\boldsymbol{S}_1 \boldsymbol{S}_2 \dots \boldsymbol{S}_k$$

is a uniform random permutation.

Proof. We prove this by induction. We can generate a random permutation on $1, \dots, n$ by first choosing which item maps to n, and then generating a random permutation on those that remain. To this end, we first form a sequence that gives a random permtuation on the first $n-1$ elements. We then compose this with a random swap that exchanges elements 1 and n with probability $1 - 1/n$. At this point, the element that maps to n will be uniformly random. We then compose with yet another sequence that gives a random permutation on the first $n-1$ elements. □

Chapter 43

Quadrature for the Finite Free Convolution

Lecture 23 from November 30, 2015

43.1 Overview

The material in today's lecture comes from [MSS15d] and [MSS15a]. My goal today is to prove simple analogs of the main quadrature results, and then give some indication of how the other quadrature statements are proved. I will also try to explain what led us to believe that these results should be true.

Recall that last lecture we considered the expected characteristic polynomial of a random matrix of the form $\boldsymbol{A} + \Pi \boldsymbol{B} \Pi^T$, where $\boldsymbol{A}$ and $\boldsymbol{B}$ are symmetric. We do not know a nice expression for this expected polynomial for general $\boldsymbol{A}$ and $\boldsymbol{B}$. However, we will see that there is a very nice expression when $\boldsymbol{A}$ and $\boldsymbol{B}$ are Laplacian matrices or the adjacency matrices of regular graphs.

43.2 The Finite Free Convolution

In Free Probability [Voi97], one studies operations on matrices in a large dimensional limit. These matrices are determined by the moments of their spectrum, and thus the operations are independent of the eigenvectors of the matrices. We consider a finite dimensional analog.

For n-dimensional symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, we consider the expected characteristic polynomial

$$\mathbb{E}_{\boldsymbol{Q} \in \mathcal{O}(n)} \left[\chi_x(\boldsymbol{A} + \boldsymbol{Q} \boldsymbol{B} \boldsymbol{Q}^T) \right],$$

where $\mathcal{O}(n)$ is the group of n-by-n orthonormal matrices, and $\boldsymbol{Q}$ is a random orthonormal matrix chosen according to the Haar measure. In case you are not familiar with "Haar measure", I'll quickly explain the idea. It captures our most natural idea of a random orthnormal matrix. For example, if $\boldsymbol{A}$ is a Gaussian random symmetric matrix, and $\boldsymbol{V}$ is its matrix of eigenvectors, then $\boldsymbol{V}$ is a random orthonormal matrix chosen according to Haar measure. Formally, it is the measure that is invariant under group operations, which in this case are multiplication by orthnormal matrices. That is, the Haar measure is the measure under which for every $S \subseteq \mathcal{O}(n)$ and $\boldsymbol{P} \in \mathcal{O}(n)$, S has the same measure as $\{\boldsymbol{Q}\boldsymbol{P} : \boldsymbol{Q} \in S\}$.

This expected characteristic polynomial does not depend on the eigenvectors of $\boldsymbol{A}$ and $\boldsymbol{B}$, and thus can be written as a function of the characteristic polynomials of these matrices. To see this, write $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T$ and $\boldsymbol{B} = \boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T$ where $\boldsymbol{U}$ and $\boldsymbol{V}$ are the orthnormal eigenvectors matrices and $\boldsymbol{C}$ and $\boldsymbol{D}$ are the diagonal matrices of eigenvalues. We have

$$\chi_x(\boldsymbol{V}\boldsymbol{D}\boldsymbol{V}^T + \boldsymbol{Q}\boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T\boldsymbol{Q}^T) = \chi_x(\boldsymbol{D} + \boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{U}\boldsymbol{C}\boldsymbol{U}^T\boldsymbol{Q}^T\boldsymbol{V}) = \chi_x(\boldsymbol{D} + (\boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{U})\boldsymbol{C}(\boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{U})^T).$$

If $\boldsymbol{Q}$ is distributed according to the Haar measure on $\mathcal{O}(n)$, then so is $\boldsymbol{V}^T\boldsymbol{Q}\boldsymbol{U}$.

If $p(x)$ and $q(x)$ are the characteristic polynomials of $\boldsymbol{A}$ and $\boldsymbol{B}$, then we define their *finite free convolution* to be the polynomial

$$p(x) \boxplus_n q(x) \overset{\text{def}}{=} \mathbb{E}_{\boldsymbol{Q}\in\mathcal{O}(n)}\left[\chi_x(\boldsymbol{A} + \boldsymbol{Q}\boldsymbol{B}\boldsymbol{Q}^T)\right].$$

In today's lecture, we will establish the following formula for the finite free convolution.

Theorem 43.2.1. *Let*

$$p(x) = \sum_{i=0}^{n} x^{n-i}(-1)^i a_i \quad \text{and} \quad q(x) = \sum_{i=0}^{n} x^{n-i}(-1)^i b_i.$$

Then,

$$p(x) \boxplus_n q(x) = \sum_{k=0}^{n} x^{n-k}(-1)^k \sum_{i+j=k} \frac{(n-i)!(n-j)!}{n!(n-i-j)!} a_i b_j. \tag{43.1}$$

This convolution was studied by Walsh [Wal22], who proved that when p and q are real rooted, so is $p \boxplus_n q$.

Our interest in the finite free convolution comes from the following theorem, whose proof we will also sketch today.

Theorem 43.2.2. *Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be symmetric matrices with constant row sums. If $\boldsymbol{A}\boldsymbol{1} = a\boldsymbol{1}$ and $\boldsymbol{B}\boldsymbol{1} = b\boldsymbol{1}$, we may write their characteristic polynomials as*

$$\chi_x(\boldsymbol{A}) = (x-a)p(x) \quad \text{and} \quad \chi_x(\boldsymbol{B}) = (x-b)q(x).$$

We then have

$$\mathbb{E}_{\Pi\in S_n}\left[\chi_x(\boldsymbol{A} + \Pi\boldsymbol{B}\Pi^T)\right] = (x-(a+b))(p(x) \boxplus_{n-1} q(x)).$$

We know that $\boldsymbol{1}$ is an eigenvector of eigenvalue $a+b$ of $\boldsymbol{A} + \Pi\boldsymbol{B}\Pi^T$ for every permutation matrix Π. Once we work orthogonal to this vector, we discover the finite free convolution.

We describe this theorem as a *quadrature* result, because it obtains an integral over a continuous space as a sum over a finite number of points.

Before going in to the proof of the theorem, I would like to explain why one might think something like this could be true. The first answer is that it was a lucky guess. We hoped that this expectation would have a nice formula. The nicest possible formula would be a bi-linear map: a function that is linear in p when q is held fixed, and vice versa. So, we computed some examples by holding $\boldsymbol{B}$ and q fixed and varying $\boldsymbol{A}$. We then observed that the coefficients of the resulting expected polynomial are in fact a linear functions of the coefficients of p. Once we knew this, it didn't take too much work to guess the formula.

I now describe the main quadrature result we will prove today. Let $\mathcal{B}(n)$ be the nth hyperoctahedral group. This is the group of symmetries of the generalized octahedron in n dimensions. It may be described as the set of matrices that can be written in the form $\boldsymbol{D}\Pi$, where $\boldsymbol{D}$ is a diagonal matrix of ± 1 entries and Π is a permutation. It looks like the family of permutation matrices, except that both 1 and -1 are allowed as nonzero entries. $\mathcal{B}(n)$ is a subgroup of $\mathcal{O}(n)$.

Theorem 43.2.3. *For all symmetric matrices* $\boldsymbol{A}$ *and* $\boldsymbol{B}$*,*

$$\mathbb{E}_{\boldsymbol{Q}\in\mathcal{O}(n)}\left[\chi_x(\boldsymbol{A}+\boldsymbol{Q}\boldsymbol{B}\boldsymbol{Q}^T)\right]=\mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[\chi_x(\boldsymbol{A}+\boldsymbol{P}\boldsymbol{B}\boldsymbol{P}^T)\right].$$

We will use this result to prove Theorem 43.2.1. The proof of Theorem 43.2.2 is similar to the proof of Theorem 43.2.3. So, we will prove Theorem 43.2.3 and then explain the major differences.

43.3 Quadrature

In general, quadrature formulas allow one to evaluate integrals of a family of functions over a fixed continuous domain by summing the values of those functions at a fixed number of points. There is an intimate connection between families of orthogonal polynomials and quadrature formulae that we unfortunately do not have time to discuss.

The best known quadrature formula allows us to evalue the integral of a polynomial around the unit circle in the complex plane. For a polynomial $p(x)$ of degree less than n,

$$\int_{\theta=0}^{2\pi} p(e^{i\theta})d\theta=\frac{1}{n}\sum_{k=0}^{n-1}p(\omega^k),$$

where $\omega=e^{2\pi i/n}$ is a primitive nth root of unity.

We may prove this result by establishing it separately for each monomial. For $p(x)=x^k$ with $k\neq 0$,

$$\int_{\theta=0}^{2\pi} p(e^{i\theta})d\theta=\int_{\theta=0}^{2\pi} e^{i\theta k}d\theta=0.$$

And, for $|k| < n$, the corresponding sum is the sum of nth roots of unity distributed symmetrically about the unit circle. So,

$$\sum_{j=0}^{n-1} \omega^{jk} = 0.$$

We used this fact in the start of the semester when we computed the eigenvectors of the ring graph and observed that all but the dominant are orthogonal to the all-1s vector.

On the other hand, for $p(x) = 1$ both the integral and sum are 1.

We will use an alternative approach to quadrature on groups, encapsulted by the following lemma.

Lemma 43.3.1. *For every n and function $p(x) = \sum_{|k|<n} c_k x^k$, and every $\theta \in [0, 2\pi]$,*

$$\sum_{j=0}^{n} p(e^{i(2\pi j/n+\theta)}) = \sum_{j=0}^{n} p(e^{i(2\pi j/n)}).$$

This identity implies the quadrature formula above, and has the advantage that it can be experimentally confirmed by evaluating both sums for a random θ.

Proof. We again evaluate the sums monomial-by-monomial. For $p(x) = x^k$, with $|k| < n$, we have

$$\sum_{j=0}^{n} (e^{i(2\pi j/n+\theta)})^k = e^{i\theta k} \sum_{j=0}^{n} (e^{i(2\pi j/n)})^k.$$

For $k \neq 0$, the latter sum is zero. For $k = 0$, $e^{i\theta k} = 1$. □

43.4 Quadrature by Invariance

For symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$, define the function

$$f_{\boldsymbol{A},\boldsymbol{B}}(\boldsymbol{Q}) = \det(\boldsymbol{A} + \boldsymbol{Q}\boldsymbol{B}\boldsymbol{Q}^T).$$

We will derive Theorem 43.2.3 from the following theorem.

Theorem 43.4.1. *For all $\boldsymbol{Q} \in \mathcal{O}(n)$,*

$$\mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)} \left[f(\boldsymbol{P}) \right] = \mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)} \left[f(\boldsymbol{Q}\boldsymbol{P}) \right].$$

Proof of Theorem 43.2.3. First, observe that it suffices to consider determinants. For every $\boldsymbol{P} \in \mathcal{B}(n)$, we have

$$\int_{\boldsymbol{Q}\in\mathcal{O}(n)} \det(\boldsymbol{A} + \boldsymbol{Q}\boldsymbol{B}\boldsymbol{Q}^T) = \int_{\boldsymbol{Q}\in\mathcal{O}(n)} f(\boldsymbol{Q}) = \int_{\boldsymbol{Q}\in\mathcal{O}(n)} f(\boldsymbol{Q}\boldsymbol{P}).$$

So,

$$\mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[\int_{\boldsymbol{Q}\in\mathcal{O}(n)} f(\boldsymbol{Q}\boldsymbol{P})\right] = \int_{\boldsymbol{Q}\in\mathcal{O}(n)} f(\boldsymbol{Q}).$$

On the other hand, as $\mathcal{B}(n)$ is discrete we can reverse the order of integration to obtain

$$\int_{\boldsymbol{Q}\in\mathcal{O}(n)} f(\boldsymbol{Q}) = \int_{\boldsymbol{Q}\in\mathcal{O}(n)} \mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[f(\boldsymbol{Q}\boldsymbol{P})\right] = \int_{\boldsymbol{Q}\in\mathcal{O}(n)} \mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[f(\boldsymbol{P})\right] = \mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[f(\boldsymbol{P})\right],$$

where the second equality follows from Theorem 43.4.1. □

43.5 Structure of the Orthogonal Group

To prove Theorem 43.4.1, we need to know a little more about the orthogonal group. We divide the orthonormal matrices into two types, those of determinant 1 and those of determinant -1. The orthonormal matrices of determinant 1 form the *special orthogonal group*, $\mathcal{SO}(n)$, and every matrix in $\mathcal{O}(n)$ may be written in the form $\boldsymbol{D}\boldsymbol{Q}$ where $\boldsymbol{Q}\in\mathcal{SO}(n)$ and $\boldsymbol{D}$ is a diagonal matrix in which the first entry is ± 1 and all others are 1. Every matrix in $\mathcal{SO}(n)$ may be expressed as a product of 2-by-2 rotation matrices. That is, for every $\boldsymbol{Q}\in\mathcal{SO}(n)$ there are matrices $\boldsymbol{Q}_{i,j}$ for $1\leq i<j\leq n$ so that $\boldsymbol{Q}_{i,j}$ is a rotation in the span of $\boldsymbol{\delta}_i$ and $\boldsymbol{\delta}_j$ and so that

$$\boldsymbol{Q} = \boldsymbol{Q}_{1,2}\boldsymbol{Q}_{1,3}\cdots\boldsymbol{Q}_{1,n}\boldsymbol{Q}_{2,3}\cdots\boldsymbol{Q}_{2,n}\cdots\boldsymbol{Q}_{n-1,n}.$$

If you learned the QR-factorization of a matrix, then you learned an algorithm for computing this decomposition.

These facts about the structure of $\mathcal{O}(n)$ tell us that it suffices to prove Theorem 43.4.1 for the special cases in which $\boldsymbol{Q} = \mathsf{diag}(-1,1,1,\dots,1)$ and when $\boldsymbol{Q}$ is rotation of the plane spanned by $\boldsymbol{\delta}_i$ and $\boldsymbol{\delta}_j$. As the diagonal matrix is contained in $\mathcal{B}(n)$, the result is immediate in that case.

For simplicity, consider the case $i=1$ and $j=2$, and let $\boldsymbol{R}_\theta$ denote the rotation by angle θ in the first two coordinates:

$$\boldsymbol{R}_\theta \stackrel{\text{def}}{=} \begin{bmatrix} \cos\theta & \sin\theta & \boldsymbol{0} \\ -\sin\theta & \cos\theta & \boldsymbol{0} \\ 0 & 0 & \boldsymbol{I}_{n-2} \end{bmatrix}.$$

The hyperoctahedral group $\mathcal{B}(n)$ contains the matrices $\boldsymbol{R}_\theta$ for $\theta\in\{0,\pi/2,\pi,3\pi/2\}$. As $\mathcal{B}(n)$ is a group, for these θ we know

$$\mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[f(\boldsymbol{P})\right] = \mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[f(\boldsymbol{R}_\theta\boldsymbol{P})\right],$$

as the set of matrices in the expectations are identical. This identity implies

$$\frac{1}{4}\sum_{j=0}^{3}\mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[f_{\boldsymbol{A},\boldsymbol{B}}(\boldsymbol{R}_{2\pi j/4}\boldsymbol{P})\right] = \mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[f(\boldsymbol{P})\right].$$

We will prove the following lemma, and then show it implies Theorem 43.4.1.

Lemma 43.5.1. *For every symmetric $\boldsymbol{A}$ and $\boldsymbol{B}$, and every θ*

$$\frac{1}{4}\sum_{j=0}^{3} f_{\boldsymbol{A},\boldsymbol{B}}(\boldsymbol{R}_{\theta+2\pi j/4}) = \frac{1}{4}\sum_{j=0}^{3} f_{\boldsymbol{A},\boldsymbol{B}}(\boldsymbol{R}_{2\pi j/4}).$$

This lemma implies that for every $\boldsymbol{Q}_{1,2}$,

$$\mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[f(\boldsymbol{P})\right] = \mathbb{E}_{\boldsymbol{P}\in\mathcal{B}(n)}\left[f(\boldsymbol{Q}_{1,2}\boldsymbol{P})\right].$$

This, in turn, implies Theorem 43.4.1 and thus Theorem 43.2.3.

We can use Lemma 43.3.1 to derive Lemma 43.5.1 follows from the following.

Lemma 43.5.2. *For every symmetric $\boldsymbol{A}$ and $\boldsymbol{B}$, there exist $c_{-2}, c_{-1}, c_0, c_1, c_2$ so that*

$$f_{\boldsymbol{A},\boldsymbol{B}}(\boldsymbol{R}_\theta) = \sum_{k=-2}^{2} c_k (e^{i\theta})^k.$$

Proof. We need to express $f(\boldsymbol{R}_\theta)$ as a function of $e^{i\theta}$. To this end, recall that

$$\cos\theta = \frac{1}{2}(e^{i\theta}+e^{-i\theta}) \quad \text{and} \quad \sin\theta = \frac{-i}{2}(e^{i\theta}-e^{-i\theta}).$$

From these identities, we see that all two-by-two rotation matrices can be simultaneously diagonalized by writing

$$\begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix} = \boldsymbol{U}\begin{pmatrix} e^{i\theta} & 0 \\ 0 & e^{-i\theta} \end{pmatrix}\boldsymbol{U}^*,$$

where

$$\boldsymbol{U} = \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix},$$

and we recall that $\boldsymbol{U}^*$ is the conjugate transpose:

$$\boldsymbol{U}^* = \begin{pmatrix} 1 & -i \\ 1 & i \end{pmatrix}.$$

Let $\boldsymbol{D}_\theta$ be the digaonal matrix having its first two entries $e^{i\theta}$ and $e^{-i\theta}$, and the rest 1, and let $\boldsymbol{U}_n$ be the matrix with $\boldsymbol{U}$ in its upper 2-by-2 block and 1s on the diagonal beneath. So,

$$\boldsymbol{R}_\theta = \boldsymbol{U}_n\boldsymbol{D}_\theta\boldsymbol{U}_n^*.$$

Now, examine

$$\begin{aligned} f_{\boldsymbol{A},\boldsymbol{B}}(\boldsymbol{R}_\theta) &= \det(\boldsymbol{A}+\boldsymbol{R}_\theta\boldsymbol{B}\boldsymbol{R}_\theta^*) \\ &= \det(\boldsymbol{A}+\boldsymbol{U}_n\boldsymbol{D}_\theta\boldsymbol{U}_n^*\boldsymbol{B}\boldsymbol{U}_n\boldsymbol{D}_\theta^*\boldsymbol{U}_n^*) \\ &= \det(\boldsymbol{U}_n^*\boldsymbol{A}\boldsymbol{U}_n+\boldsymbol{D}_\theta\boldsymbol{U}_n^*\boldsymbol{B}\boldsymbol{U}_n\boldsymbol{D}_\theta^*) \\ &= \det(\boldsymbol{U}_n^*\boldsymbol{A}\boldsymbol{U}_n\boldsymbol{D}_\theta+\boldsymbol{D}_\theta\boldsymbol{U}_n^*\boldsymbol{B}\boldsymbol{U}_n). \end{aligned}$$

The term $e^{i\theta}$ only appears in the first row and column of this matrix, and the term $e^{-i\theta}$ only appears in the second row and column. As a determinant can be expressed as a sum of products of matrix entries with one in each row and column, it is immediate that this determinant can be expressed in terms of $e^{ki\theta}$ for $|k| \leq 4$. As each such product can have at most 2 terms of the form $e^{i\theta}$ and at most two of the form $e^{-i\theta}$, we have $|k| \leq 2$. □

The difference between Theorem 43.2.3 and Theorem 43.2.2 is that the first involves a sum over the isometries of hyperoctahedron, while the second involves a sum over the symmetries of the regular n-simplex in $n-1$ dimensions. The proof of the appropriate quadrature theorem for the symmetries of the regular simplex is very similar to the proof we just saw, except that rotations of the plane through $\boldsymbol{\delta}_i$ and $\boldsymbol{\delta}_j$ are replaced by rotations of the plane parallel to the affine subspace spanned by triples of vertices of the simplex.

43.6 The Formula

To establish the formula in Theorem 43.2.1, we observe that it suffices to compute the formula for diagonal matrices, and that Theorem 43.2.3 makes this simple. Every matrix in $\mathcal{B}(n)$ can be written as a product $\Pi \boldsymbol{D}$ where $\boldsymbol{D}$ is a ± 1 diagonal matrix. If $\boldsymbol{B}$ is the diagonal matrix with entries $\mu_1, \ldots, \mu_n$, then $\Pi \boldsymbol{D}\boldsymbol{B}\boldsymbol{D}\Pi^T = \Pi \boldsymbol{B} \Pi^T$, which is the diagonal matrix with entries $\mu_{\pi(1)}, \ldots, \mu_{\pi(n)}$, where π is the permutation corresponding to Π.

Let $\boldsymbol{A}$ be diagonal with entries $\lambda_1, \ldots, \lambda_n$. For a subset S of $\{1, \ldots, n\}$, define

$$\lambda^S = \prod_{i \in S} \lambda_i.$$

We then have

$$a_i = \sum_{|S|=i} \lambda^S.$$

Let

$$p \boxplus_n q = \sum_{k=0}^{n} x^{n-k} (-1)^k c_k.$$

We first compute the expected determinant, c_n.

$$c_n = \frac{1}{n!} \sum_{\pi} \prod_{h} (\lambda_h + \mu_{\pi(h)}) = \frac{1}{n!} \sum_{\pi} \sum_{|S|=i} \lambda^S \prod_{h : \pi(h) \not\in S} \mu_h.$$

As opposed to expanding this out, let's just figure out how often the product $\lambda^S \mu^T$ appears. We must have $|T| = n - |S|$, and then this term appears for each permutation such that $\pi(T) \cap S = \emptyset$. This happens $1/\binom{n}{i}$ fraction of the time, giving the formula

$$c_n = \sum_{i=0}^{n} \frac{1}{\binom{n}{i}} \sum_{|S|=i} \lambda^S \sum_{|T|=n-i} \mu^T = \sum_{i=0}^{n} \frac{1}{\binom{n}{i}} a_i b_{n-i} = \sum_{i=0}^{n} \frac{i!(n-i)!}{n!} a_i b_{n-i}.$$

For general c_k and $i + j = k$, we see that λ^S and μ^T appear whenever $\mu(T)$ is disjoint from S. The probability of this happening is

$$\frac{\binom{n-i}{j}}{\binom{n}{j}} = \frac{(n-i)!(n-j)!j!}{n!(n-i-j)!j!} = \frac{(n-i)!(n-j)!}{n!(n-i-j)!},$$

and so

$$c_k = \sum_{i+j=k} a_i b_j \frac{(n-i)!(n-j)!}{n!(n-i-j)!}.$$

43.7 Question

For which discrete subgroups of $\mathcal{O}(n)$ does a result like Theorem 43.2.3 hold? Can it hold for a substantially smaller subgroup than the symmetries of the simplex (which has size $(n+1)!$ in n dimensions).

Chapter 44

Ramanujan Graphs of Every Size

Lecture 24 from December 2, 2015

44.1 Overview

We will *mostly* prove that there are Ramanujan graphs of every number of vertices and degree. The material in today's lecture comes from [MSS15d] and [MSS15a]. In those papers, we prove that for every even n and degree $d < n$ there is a **bipartite** Ramanujan graph of degree d on n vertices. A bipartite Ramanujan graph of degree d is an approximation of a complete bipartite graph. It's adjacency matrix thus has eigenvalues d and $-d$, and all other eigenvalues bounded in absolute value by $2\sqrt{d-1}$.

The difference between this result and that which we prove today is that we will show that for every $d < n$ there is a d-regular (multi) graph in whose second adjacency matrix eigenvalue is at most $2\sqrt{d-1}$. This bound is sufficient for many applications of expanders, but not all. We will not control the magnitude of the negative eigenvalues. The reason will simply be for simplicity: the proofs to bound the negative eigenvalues would take more lectures.

Next week we will see a different technique that won't produce a multigraph and that will produce a bipartite Ramanujan graph.

44.2 The Approach

We will consider the sum of d random perfect matchings on n vertices. This produces a d-regular graph that might be a multigraph. Friedman [Fri08] proves that such a graph is probably very close to being Ramanujan if n is big enough relative to d. In particular, he proves that for all d and $\epsilon > 0$ there is an n_0 so that for all $n > n_0$, such a graph will probably have all eigenvalues other than μ_1 bounded in absolute value by $2\sqrt{d-1} + \epsilon$. We remove the asymptotics and the ϵ, but merely prove the existence of one such graph. We do not estimate the probability with which such a graph is Ramanujan. But, it is predicted to be a constant [**?**].

The fundamental difference between our technique and that of Friedman is that Friedman bounds the moments of the distribution of the eigenvalues of such a random graph. I suspect that there is no true bound on these moments that would allow one to conclude that a random graph is probably Ramanujan. We consider the expected characteristic polynomial.

Let $\boldsymbol{M}$ be the adjacency matrix of a perfect matching, and let $\Pi_1, \dots, \Pi_d$ be independent uniform random permutation matrices. We will consider the expected characteristic polynomial

$$\mathbb{E}_{\Pi_1,\dots,\Pi_d}\left[\chi_x(\Pi_1 \boldsymbol{M} \Pi_1^T + \cdots + \Pi_d \boldsymbol{M} \Pi_d^T)\right].$$

In Lecture 22, we learned that this polynomial is real rooted. In Lecture 23, we learned a technique that allows us to compute this polynomial. Today we will prove that the second largest root of this polynomial is at most $2\sqrt{d-1}$. First, we show why this matters: it implies that there is some choice of the matrices $\Pi_1, \dots, \Pi_d$ so that resulting polynomial has second largest root at most $2\sqrt{d-1}$. These matrices provide the desired graph.

44.3 Interlacing Families of Polynomials

The general problem we face is the following. We have a large family of polynomials, say $p_1(x), \dots, p_m(x)$, for which we know each p_i is real-rooted and such that their sum is real rooted. We would like to show that there is some polynomial p_i whose largest root is at most the largest root of the sum, or rather we want to do this for the second-largest root. This is not true in general. But, it is true in our case. We will show that it is true whenever the polynomials form what we call an *interlacing family.*

Recall from Lecture 22 that we say that for monic degree n polynomials $p(x)$ and $r(x)$, $p(x) \rightarrow r(x)$ if the roots of p and r interlace, with the roots of r being larger. We proved that if $p_1(x) \rightarrow r(x)$ and $p_2(x) \rightarrow r(x)$, then every convex combination of p_1 and p_2 is real rooted. If we go through the proof, we will also see that for all $0 \leq s \leq 1$,

$$sp_2(x) + (1-s)p_1(x) \rightarrow r(x).$$

Proceeding by induction, we can show that if $p_i(x) \rightarrow r(x)$ for each i, then every convex combination of these polynomials interlaces $r(x)$, and is thus real rooted. That is, for every $s_1, \dots, s_m$ so that $s_i \geq 0$ (but not all are zero),

$$\sum_i s_i p_i(x) \rightarrow r(x).$$

Polynomials that satisfy this condition are said to have a *common interlacing.* By a technique analogous to the one we used to prove Lemma 22.3.2, one can prove that the polynomials $p_1, \dots, p_m$ have a common interlacing if and only if every convex combination of these polynomials is real rooted.

Lemma 44.3.1. *Let $p_1, \ldots, p_m$ be polynomials so that $p_i(x) \to r(x)$, and let $s_1, \ldots, s_m \geq 0$ be not identically zero. Define*

$$p_{\emptyset}(x) = \sum_{i=1}^{m} s_i p_i(x).$$

Then, there is an i so that the largest root of $p_i(x)$ is at most the largest root of $p_{\emptyset}(x)$. In general, for every j there is an i so that the jth largest root of $p_i(x)$ is at most the jth largest root of $p_{\emptyset}(x)$.

Proof. We prove this for the largest root. The proof for the others is similar. Let λ_1 and λ_2 be the largest and second-largest roots of $r(x)$. Each polynomial $p_i(x)$ has exactly one root between λ_1 and λ_2, and is positive at all $x > \lambda_1$. Now, let μ be the largest root of $p_{\emptyset}(x)$. We can see that μ must lie between λ_1 and λ_2. We also know that

$$\sum_i p_i(\mu) = 0.$$

If $p_i(\mu) = 0$ for some i, then we are done. If not, there is an i for which $p_i(\mu) > 0$. As p_i only has one root larger than λ_2, and it is eventually positive, the largest root of p_i must be less than μ. □

Our polynomials do not all have a common interlacing. However, they satisfy a property that is just as useful: they form an interlacing family. We say that a set of polynomials $p_1, \ldots, p_m$ forms an interlacing family if there is a rooted tree T in which

a. every leaf is labeled by some polynomial p_i,

b. every internal vertex is labeled by a nonzero, nonnegative combination of its children, and

c. all siblings have a common interlacing.

The last condition guarantees that every internal vertex is labeled by a real rooted polynomial. Note that the same label is allowed to appear at many leaves.

Lemma 44.3.2. *Let $p_1, \ldots, p_m$ be an interlacing family, let T be the tree witnessing this, and let $p_{\emptyset}$ be the polynomial labeling the root of the tree. Then, for every j there exists an i for which the jth largest root of p_i is at most the jth largest root of $p_{\emptyset}$.*

Proof. By Lemma 44.3.1, there is a child of the root whose label has a jth largest root that is smaller than the jth largest root of $p_{\emptyset}$. If that child is not a leaf, then we can proceed down the tree until we reach a leaf, at each step finding a node labeled by a polynomial whose jth largest root is at most the jth largest root of the previous polynomial. □

Our construction of permutations by sequences of random swaps provides the required interlacing family.

Theorem 44.3.3. *For permutation matrices $\Pi_1, \ldots, \Pi_d$, let*

$$p_{\Pi_1,\ldots,\Pi_d}(x) = \chi_x(\Pi_1 \boldsymbol{M} \Pi_1^T + \cdots + \Pi_d \boldsymbol{M} \Pi_d^T).$$

These polynomials form an interlacing family.

We will finish this lecture by proving that the second-largest root of

$$\mathbb{E}\left[p_{\Pi_1,\ldots,\Pi_d}(x)\right]$$

is at most $2\sqrt{d-1}$. This implies that there is a d-regular multigraph on n vertices in our family with second-largest adjacency eigenvalue at most $2\sqrt{d-1}$.

44.4 Root Bounds for Finite Free Convolutions

Recall from the last lecture that for n-dimensional symmetric matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ with uniform row sums a and b and characteristic polynomials $(x-a)p(x)$ and $(x-b)q(x)$,

$$\mathbb{E}_{\Pi}\left[\chi_x(\boldsymbol{A}+\Pi\boldsymbol{B}\Pi^T)\right] = (x-(a+b))p(x) \boxplus_{n-1} q(x).$$

This formula extends to sums of many such matrices. It is easy to show that

$$\chi_x(\boldsymbol{M}) = (x-1)^{n/2}(x+1)^{n/2} = (x-1)p(x), \quad \text{where } p(x) \overset{\text{def}}{=} (x-1)^{n/2-1}(x+1)^{n/2}.$$

So,

$$p_{\emptyset}(x) \overset{\text{def}}{=} \mathbb{E}\left[p_{\Pi_1,\ldots,\Pi_d}(x)\right] = (x-d)\left(p(x) \boxplus_{n-1} p(x) \boxplus_{n-1} p(x) \boxplus_{n-1} \cdots \boxplus_{n-1} p(x)\right),$$

where $p(x)$ appears d times above.

We would like to prove a bound on the largest root of this polynomial in terms of the largest roots of $p(x)$. This effort turns out not to be productive. To see why, consider matrices $\boldsymbol{A} = a\boldsymbol{I}$ and $\boldsymbol{B} = b\boldsymbol{I}$. It is clear that $\boldsymbol{A}+\Pi\boldsymbol{B}\Pi^T = (a+b)\boldsymbol{I}$ for every Π. This tells us that

$$(x-a)^n \boxplus (x-b)^n = (x-(a+b))^n.$$

So, the largest roots can add. This means that if we are going to obtain useful bounds on the roots of the sum, we are going to need to exploit facts about the distribution of the roots of $p(x)$. As in Lecture ??, we will use the barrier functions, just scaled a little differently.

For,

$$p(x) = \prod_{i=1}^{n}(x-\lambda_i),$$

define the Cauchy transform of p at x to be

$$\mathcal{G}_p\left(x\right) = \frac{1}{d}\sum_{i=1}^{d}\frac{1}{x-\lambda_i} = \frac{1}{d}\frac{p'(x)}{p(x)}.$$

For those who are used to Cauchy transforms, I remark that this is the Cauchy transform of the uniform distribution on the roots of $p(x)$. As we will be interested in upper bounds on the Cauchy transform, we will want a number u so that for all $x > u$, $\mathcal{G}_p\left(x\right)$ is less than some specified value. That is, we want the *inverse Cauchy transform*, which we define to be

$$\mathcal{K}_p\left(w\right) = \max\left\{x : \mathcal{G}_p\left(x\right) = w\right\}.$$

For a real rooted polynomial p, and thus for real $\lambda_1, \ldots, \lambda_d$, it is the value of x that is larger than all the λ_i for which $\mathcal{G}_p\left(x\right) = w$. For $w = \infty$, it is the largest root of p. But, it is larger for finite w.

We will prove the following bound on the Cauchy transforms.

Theorem 44.4.1. *For degree n polynomials p and q and for $w > 0$,*

$$\mathcal{K}_{p \boxplus_n q}\left(w\right) \leq \mathcal{K}_p\left(w\right) + \mathcal{K}_q\left(w\right) - 1/w.$$

For $w = \infty$, this says that the largest root of $p \boxplus_n q$ is at most the sum of the largest roots of p and q. But, this is obvious.

To explain the $1/w$ term in the above expression, consider $q(x) = x^n$. As this is the characteristic polynomial of the all-zero matrix, $p \boxplus_n q = p(x)$. We have

$$\mathcal{G}_q\left(x\right) = \frac{1}{n}\frac{nx^{n-1}}{x^n} = \frac{1}{x}.$$

So,

$$\mathcal{K}_q\left(w\right) = \max\left\{x : 1/x = w\right\} = 1/w.$$

Thus,

$$\mathcal{K}_q\left(w\right) - 1/w = 0.$$

I will defer the proof of this theorme to next lecture (or maybe the paper [MSS15a]), and now just show how we use it.

44.5 The Calculation

For $p(x) = (x-1)^{n/2-1}(x+1)^{n/2}$,

$$\mathcal{G}_p\left(x\right) = \frac{1}{n-1}\left(\frac{n/2-1}{x-1} + \frac{n/2}{x+1}\right) \leq \frac{1}{n}\left(\frac{n/2}{x-1} + \frac{n/2}{x+1}\right),$$

for $x \geq 1$. This latter expression is simple to evaluate. It is

$$\frac{x}{x^2-1} = \mathcal{G}_{\chi(\boldsymbol{M})}\left(x\right).$$

We also see that

$$\mathcal{K}_p\left(w\right) \leq \mathcal{K}_{\chi(\boldsymbol{M})}\left(w\right),$$

for all $w \geq 0$.

Theorem 44.4.1 tells us that

$$\mathcal{K}_{p \boxplus_{n-1} \cdots \boxplus p}\left(w\right) \leq d\mathcal{K}_p\left(w\right) - \frac{d-1}{w}.$$

Using the above inequality, we see that this is at most

$$d\mathcal{K}_{\chi(\boldsymbol{M})}\left(w\right) - \frac{d-1}{w}.$$

As this is an upper bound on the largest root of $p \boxplus_{n-1} \cdots \boxplus_{n-1} p$, we wish to set w to minimize this expression. As,

$$\mathcal{G}_{\chi(\boldsymbol{M})}(x) = \frac{x}{x^2-1},$$

we have

$$\mathcal{K}_{\chi(\boldsymbol{M})}(w) = x \quad \text{if and only if} \quad w = \frac{x}{x^2-1}.$$

So,

$$d\mathcal{K}_{\chi(\boldsymbol{M})}(w) - \frac{d-1}{w}. \leq dx - d - 1\frac{x^2-1}{x}.$$

The choice of x that minimizes this is $\sqrt{d-1}$, at which point it becomes

$$d\sqrt{d-1} - \frac{(d-1)(d-2)}{\sqrt{d-1}} = d\sqrt{d-1} - (d-2)\sqrt{d-1} = 2\sqrt{d-1}.$$

44.6 Some explanation of Theorem 44.4.1

I will now have time to go through the proof of Theorem 44.4.1. So, I'll just tell you a little about it. We begin by transforming statements about the inverse Cauchy transform into statements about the roots of polynomials.

As $\mathcal{G}_p(x) = \frac{1}{d}\frac{p'(x)}{p(x)}$,

$$\mathcal{G}_p(x) = w \quad \iff \quad p(x) - \frac{1}{wd}p'(x) = 0.$$

This tells us that

$$\mathcal{K}_p(w) = \mathrm{maxroot}\left(p(x) - p'(x)/wd\right) = \mathrm{maxroot}\left((1-(1/wd)\partial_x)p\right).$$

As this sort of operator appears a lot in the proof, we give it a name:

$$U_\alpha = 1 - \alpha\partial_x.$$

In this notation, Theorem 44.4.1 becomes

$$\mathrm{maxroot}\left(U_\alpha(p \boxplus_n q)\right) \leq \mathrm{maxroot}\left(U_\alpha p\right) + \mathrm{maxroot}\left(U_\alpha p\right) - n\alpha. \tag{44.1}$$

We, of course, also need to exploit an expression for the finite free convolution. Last lecture, we proved that if

$$p(x) = \sum_{i=0}^{n} x^{n-i}(-1)^i a_i \quad \text{and} \quad q(x) = \sum_{i=0}^{n} x^{n-i}(-1)^i b_i.$$

Then,

$$p(x) \boxplus_n q(x) = \sum_{k=0}^{n} x^{n-k}(-1)^k \sum_{i+j=k} \frac{(n-i)!(n-j)!}{n!(n-i-j)!} a_i b_j. \tag{44.2}$$

From this, one can derive a formula that plays better with derivatives:

$$p(x) \boxplus_n q(x) = \frac{1}{n!} \sum_{i=0}^{n} (n-i)! b_i p^{(i)}(x).$$

This equation allows us to understand what happens when p and q have different degrees.

Lemma 44.6.1. *If $p(x)$ has degree n and $q(x) = x^{n-1}$, then*

$$p(x) \boxplus_n q(x) = \partial_x p(x).$$

For the special case of $q(x) = x^{n-1}$, we have

$$U_\alpha q(x) = x^{n-1} - \alpha(n-1)x^{n-2},$$

so

$$\operatorname{maxroot}\left(U_\alpha q(x)\right) = \alpha(n-1).$$

So, in this case (44.1) says

$$\operatorname{maxroot}\left(U_\alpha \partial_x p\right) \leq \operatorname{maxroot}\left(U_\alpha p\right) + \operatorname{maxroot}\left(U_\alpha q\right) - n\alpha = \operatorname{maxroot}\left(U_\alpha p\right) - \alpha.$$

The proof of Theorem 44.4.1 has two major ingredients. We begin by proving the above inequality. We then show that the extreme case for the inequality is when $q(x) = (x-b)^n$ for some b. To do this, we consider an arbitrary real rooted polynomial q, and then modify it to make two of its roots the same. This leads to an induction on degree, which is essentially handled by the following result.

Lemma 44.6.2. *If $p(x)$ has degree n and the degree of $q(x)$ is less than n, then*

$$p \boxplus_n q = \frac{1}{n} (\partial_x p) \boxplus_{n-1} q.$$

The whose proof is fairly straightforward, and only requires 2 pages.

44.7 Some thoughts

I would like to reflect on the fundamental difference between considering expected characteristic polynomials and the distributions of the roots of random polynomials. Let $\boldsymbol{A}$ be a symmetric matrix of dimension $3k$ with k eigenvalues that are 1, 0, and -1. If you consider $\boldsymbol{A} + \Pi \boldsymbol{A} \Pi^T$ for a random Π, the resulting matrix will almost definitely have a root at 2 and a root at -2. In fact, the chance that it does not is exponentially small in k. However, all the roots of the expected characteristic polynomial of this matrix are strictly bounded away from 2. You could verify this by computing the Cauchy transform of this polynomial.

In our case, we considered a matrix $\boldsymbol{A}$ with k eigenvalues of 1 and k eigenvalues of -1. If we consider $\boldsymbol{A} + \Pi \boldsymbol{A} \Pi^T$, it will almost definitely have roots at 2 and -2, and in fact the expected characteristic polynomial has roots that are very close to this. But, if we consider

$$\boldsymbol{A} + \Pi_1 \boldsymbol{A} \Pi_1^T + \Pi_2 \boldsymbol{A} \Pi_2^T,$$

even though it almost definitely has roots at 3 and -3, the largest root of the expected characteristic polynomial is at most $2\sqrt{2} < 3$.

I should finish by saying that Theorem 44.4.1 is inspired by a theorem of Voiculescu that holds in the infinite dimensional case. In this limit, the inequality becomes an equality.

Chapter 45

Matching Polynomials of Graphs

Lecture 25 from December 7, 2015

45.1 Overview

The coefficients of the matching polynomial of a graph count the numbers of matchings of various sizes in that graph. It was first defined by Heilmann and Lieb [HL72], who proved that it has some amazing properties, including that it is real rooted. They also proved that all root of the matching polynomial of a graph of maximum degree d are at most $2\sqrt{d-1}$. In the next lecture, we will use this fact to derive the existence of Ramanujan graphs.

Our proofs today come from a different approach to the matching polynomial that appears in the work of Godsil [God93, God81]. My hope is that someone can exploit Godsil's approach to connect the $2\sqrt{d-1}$ bound from today's lecture with that from last lecture. In today's lecture, $2\sqrt{d-1}$ appears as an upper bound on the spectral radius of a d-ary tree. Infinite d-ary trees appear as the graphs of free groups in free probability. I feel like there must be a formal relation between these that I am missing.

45.2 The Matching Polynomial

A matching in a graph $G = (V, E)$ is a subgraph of G in which every vertex has degree 1. We say that a matching has size k if it has k edges. We let

$$m_k(G)$$

denote the number of matchings in G of size k. Throughout this lecture, we let $|V| = n$. Observe that $m_1(G)$ is the number of edges in G, and that $m_{n/2}(G)$ is the number of perfect matchings in G. By convention we set $m_0(G) = 1$, as the empty set is matching with no edges. Computing the number of perfect matchings is a $\#P$-hard problem. This means that it is much harder than solving NP-hard problems, so you shouldn't expect to do it quickly on large graphs.

The *matching polynomial of* G, written $\mu_x [G]$, is

$$\mu_x [G] \overset{\text{def}}{=} \sum_{k=0}^{n/2} x^{n-2k}(-1)^k m_k(G).$$

Our convention that $m_0(G) = 1$ ensures that this is a polynomial of degree n.

This is a fundamental example of a polynomial that is defined so that its coefficients count something. When the "something" is interesting, the polynomial usually is as well.

45.3 Properties of the Matching Polynomial

We begin by establishing some fundamental properties of the matching polynomial. For graphs G and H on different vertex sets, we write $G \cup H$ for their disjoint union.

Lemma 45.3.1. *Let G and H be graphs on different vertex sets. Then,*

$$\mu_x [G \cup H] = \mu_x [G] \, \mu_x [H] .$$

Proof. Every matching in $G \cup H$ is the union of a matchings in G and a matching in H. Thus,

$$m_k(G \cup H) = \sum_{j=0}^{k} \mu_j(G)\mu_{k-j}(H).$$

The lemma follows. □

For a a vertex of $G = (V, E)$, we write $G - a$ for the graph $G(V - \{a\})$. This notation will prove very useful when reasoning about matching polynmomials. Fix a vertex a of G, and divide the matchings in G into two classes: those that involve vertex a and those that do not. The number of matchings of size k that do not involve a is $m_k(G - a)$. On the other hand, those that do involve a connect a to one of its neighbors. To count these, we enumerate the neighbors b of a. A matching of size k that includes edge (a, b) can be written as the union of (a, b) and a matching of size $k - 1$ in $G - a - b$. So, the number of matchings that involve a is

$$\sum_{b \sim a} m_{k-1}(G - a - b).$$

So,

$$m_k(G) = m_k(G - a) + \sum_{b \sim a} m_{k-1}(G - a - b).$$

To turn this into a recurrence for $\mu_x [G]$, write

$$x^{n-2k}(-1)^k m_k(G) = x \cdot x^{n-1-2k}(-1)^k m_k(G - a) - x^{n-2-2(k-1)}(-1)^{k-1} m_{k-1}(G - a - b).$$

This establishes the following formula.

Lemma 45.3.2.

$$\mu_x[G] = x\mu_x[G-a] - \sum_{b \sim a} \mu_x[G-a-b].$$

The matching polynomials of trees are very special—they are exactly the same as the characteristic polynomial of the adjacency matrix.

Theorem 45.3.3. *Let G be a tree. Then*

$$\mu_x[G] = \chi_x(\boldsymbol{A}_G).$$

Proof. Expand

$$\chi_x(\boldsymbol{A}_G) = \det(x\boldsymbol{I} - \boldsymbol{A}_G)$$

by summing over permutations. We obtain

$$\sum_{\pi \in S_n} (-1)^{\mathsf{sgn}(\pi)} x^{|\{a:\pi(a)=a\}|} \prod_{a:\pi(a)\neq a} (-\boldsymbol{A}_G(a,\pi(a))).$$

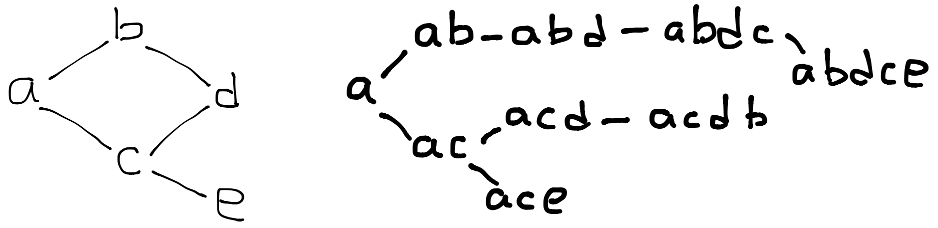
We will prove that the only permutations that contribute to this sum are those for which $\pi(\pi(a)) = a$ for every a. And, these correspond to matchings.

If π is a permutation for which there is an a so that $\pi(\pi(a)) \neq a$, then there are $a = a_1, \dots, a_k$ with $k > 2$ so that $\pi(a_i) = a_{i+1}$ for $1 \leq i < k$, and $\pi(a_k) = a_1$. For this term to contribute, it must be the case that $\boldsymbol{A}_G(a_i, a_{i+1}) = 1$ for all i, and that $\boldsymbol{A}_G(a_k, a_1) = 1$. For $k > 2$, this would be a cycle of length k in G. However, G is a tree and so cannot have a cycle.

So, the only permutations that contribute are the *involutions*: the permutations π that are their own inverse. An involution has only fixed points and cycles of length 2. Each cycle of length 2 that contributes a nonzero term corresponds to an edge in the graph. Thus, the number of permutations with k cycles of length 2 is equal to the number of matchings with k edges. As the sign of an involution with k cycles of length 2 is $(-1)^k$, the coefficient of x^{n-2k} is $(-1)^k m_k(G)$. □

45.4 The Path Tree

Godsil proves that the matching polynomial of a graph is real rooted by proving that it divides the matching polynomial of a tree. As the matching polynomial of a tree is the same as the characteristic polynomial of its adjacency matrix, it is real rooted. Thus, the matching polynomial of the graph is as well. The tree that Godsil uses is the *path tree of* G starting at a vertex of G. For a a vertex of G, the path tree of G starting at a, written $T_a(G)$ is a tree whose vertices correspond to paths in G that start at a and do not contain any vertex twice. One path is connected to another if one extends the other by one vertex. For example, here is a graph and its path tree starting at a.



When G is a tree, $T_a(G)$ is isomorphic to G.

Godsil's proof begins by deriving a somewhat strange equality. Since I haven't yet found a better proof, I'll take this route too.

Theorem 45.4.1. *For every graph G and vertex a of G,*

$$\frac{\mu_x\left[G-a\right]}{\mu_x\left[G\right]}=\frac{\mu_x\left[T_a(G)-a\right]}{\mu_x\left[T_a(G)\right]}.$$

The term on the upper-right hand side is a little odd. It is a forrest obtained by removing the root of the tree $T_a(G)$. We may write it as a disjoint union of trees as

$$T_a(G)-a=\bigcup_{b\sim a}T_b(G-a).$$

Proof. If G is a tree, then the left and right sides are identical, and so the inequality holds. As the only graphs on less than 3 vertices are trees, the theorem holds for all graphs on at most 2 vertices. We will now prove it by induction on the number of vertices.

We may use Lemma 45.3.2 to expand the reciprocal of the left-hand side:

$$\frac{\mu_x\left[G\right]}{\mu_x\left[G-a\right]}=\frac{x\mu_x\left[G-a\right]-\sum_{b\sim a}\mu_x\left[G-a-b\right]}{\mu_x\left[G-a\right]}=x-\sum_{b\sim a}\frac{\mu_x\left[G-a-b\right]}{\mu_x\left[G-a\right]}.$$

By applying the inductive hypothesis to $G-a$, we see that this equals

$$x-\sum_{b\sim a}\frac{\mu_x\left[T_b(G-a)-b\right]}{\mu_x\left[T_b(G-a)\right]}. \tag{45.1}$$

To simplify this expression, we examine these graphs carefully. By the observtion we made before the proof,

$$T_b(G-a)-b=\bigcup_{c\sim b,c\neq a}T_c(G-a-b).$$

Similarly,

$$T_a(G)-a=\bigcup_{c\sim a}T_c(G-a),$$

which implies

$$\mu_x\left[T_a(G)-a\right]=\prod_{c\sim a}\mu_x\left[T_c(G-a)\right].$$

Let ab be the vertex in $T_a(G)$ corresponding to the path from a to b. We also have

$$\begin{aligned} T_a(G) - a - ab &= \left(\bigcup_{c \sim a, c \neq b} T_c(G-a) \right) \cup \left(\bigcup_{c \sim b, c \neq a} T_c(G-a-b) \right) \\ &= \left(\bigcup_{c \sim a, c \neq b} T_c(G-a) \right) \cup (T_b(G-a)-b). \end{aligned}$$

which implies

$$\mu_x [T_a(G) - a - ab] = \left(\prod_{c \sim a, c \neq b} \mu_x [T_c(G-a)] \right) \mu_x [T_b(G-a) - b].$$

Thus,

$$\begin{aligned} \frac{\mu_x [T_a(G) - a - ab]}{\mu_x [T_a(G) - a]} &= \frac{\left(\prod_{c \sim a, c \neq b} \mu_x [T_c(G-a)] \right) \mu_x [T_b(G-a) - b]}{\prod_{c \sim a} \mu_x [T_c(G-a)]} \\ &= \frac{\mu_x [T_b(G-a) - b]}{\mu_x [T_b(G-a)]}. \end{aligned}$$

Plugging this in to (45.1), we obtain

$$\begin{aligned} \frac{\mu_x [G]}{\mu_x [G-a]} &= x - \sum_{b \sim a} \frac{\mu_x [T_a(G) - a - ab]}{\mu_x [T_a(G) - a]} \\ &= \frac{x\mu_x [T_a(G) - a] - \sum_{b \sim a} \mu_x [T_a(G) - a - ab]}{\mu_x [T_a(G) - a]} \\ &= \frac{\mu_x [T_a(G)]}{\mu_x [T_a(G) - a]}. \end{aligned}$$

Be obtain the equality claimed in the theorem by taking the reciprocals of both sides. □

Theorem 45.4.2. *For every vertex a of G, the polynomial $\mu_x [G]$ divides the polynomial $\mu_x [T_a(G)]$.*

Proof. We again prove this by induction on the number of vertices in G, using as our base case graphs with at most 2 vertices. We then know, by induction, that for $b \sim a$,

$$\mu_x [G-a] \quad \text{divides} \quad \mu_x [T_b(G-a)].$$

As

$$T_a(G) - a = \cup_{b \sim a} T_b(G-a),$$

$$\mu_x [T_b(G-a)] \quad \text{divides} \quad \mu_x [T_a(G) - a].$$

Thus,

$$\mu_x [G-a] \quad \text{divides} \quad \mu_x [T_a(G) - a],$$

and so

$$\frac{\mu_x\left[T_a(G)-a\right]}{\mu_x\left[G-a\right]}$$

is a polynomial in x. To finish the proof, we apply Theorem 45.4.1, which implies

$$\mu_x\left[T_a(G)\right] = \mu_x\left[T_a(G)-a\right]\frac{\mu_x\left[G\right]}{\mu_x\left[G-a\right]} = \mu_x\left[G\right]\frac{\mu_x\left[T_a(G)-a\right]}{\mu_x\left[G-a\right]}.$$

□

45.5 Root bounds

If every vertex of G has degree at most d, then the same is true of $T_a(G)$. We will show that the norm of the adjacency matrix of a tree in which every vertex has degree at most d is at most $2\sqrt{d-1}$. Thus, all of the roots of the matching polynomial of a graph of maximum degree d are at most $2\sqrt{d-1}$.

Theorem 45.5.1. *Let T be a tree in which every vertex has degree at most d. Then, all eigenvalues of $\chi_x(\boldsymbol{A}_T)$ have absolute value at most $2\sqrt{d-1}$.*

Proof. Let $\boldsymbol{A}$ be the adjacency matrix of T. Choose some vertex to be the root of the tree, and define its height to be 0. For every other vertex a, define its height, $h(a)$, to be its distance to the root. Define $\boldsymbol{D}$ to be the diagonal matrix with

$$\boldsymbol{D}(a,a) = \left(\sqrt{d-1}\right)^{h(a)}.$$

Recall that the eigenvalues of $\boldsymbol{A}$ are the same as the eigenvalues of $\boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^{-1}$. We will use the fact that all eigenvalues of a nonnegative matrix are upper bounded in absolute value by its maximum row sum.

So, we need to prove that all row sums of $\boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^{-1}$ are at most $2\sqrt{d-1}$. There are three types of vertices to consider. First, the row of the root has up to d entries that are all $1/\sqrt{d-1}$. For $d \geq 2$, $d/\sqrt{d-1} \leq 2\sqrt{d-1}$. The intermediate vertices have one entry in their row that equals $\sqrt{d-1}$, and up to $d-1$ entries that are equal to $1/\sqrt{d-1}$, for a total of $2\sqrt{d-1}$. Finally, every leaf only has one nonzero entry in its row, and that entry equals $\sqrt{d-1}$. □

When combined with Theorem 45.4.2, this tells us that that matching polynomial of a graph with all degrees at most d has all of its roots bounded in absolute value by $2\sqrt{d-1}$.

Chapter 46

Bipartite Ramanujan Graphs of Every Degree

Lecture 26 from December 9, 2015

These notes are still very rough.

46.1 Overview

In today's lecture, we will prove the existence of infinite families of bipartite Ramanujan of every degree. We do this by proving (half) a conjecture of Bilu and Linial [BL06] that every bipartite Ramanujan graph has a 2-lift that is also Ramanujan.

Today's theorem comes from [MSS15b], and the proof is informed by the techniques of [HPS15]. We will use theorems about the matching polynomials of graphs that we proved last lecture.

46.2 2-Lifts

We saw 2-lifts of graphs in Problem 4 from Problem Set 2:

> We define a *signed adjacency matrix* of G to be a symmetric matrix $\boldsymbol{S}$ with the same nonzero pattern as the adjacency matrix $\boldsymbol{A}$, but such that each nonzero entry is either 1 or -1.
>
> We will use it to define a graph $G^{\boldsymbol{S}}$. Like the double-cover, the graph $G^{\boldsymbol{S}}$ will have two vertices for every vertex of G and two edges for every edge of G. For each edge $(u, v) \in E$, if $\boldsymbol{S}(u, v) = -1$ then $G^{\boldsymbol{S}}$ has the two edges
>
> $$(u_1, v_2) \quad \text{and} \quad (v_1, u_2),$$

just like the double-cover. If $\boldsymbol{S}(u,v) = 1$, then $G^{\boldsymbol{S}}$ has the two edges

$$(u_1, v_1) \quad \text{and} \quad (v_2, u_2).$$

You should check that $G^{-\boldsymbol{A}}$ is the double-cover of G and that $G^{\boldsymbol{A}}$ consists of two disjoint copies of G.

Prove that the eigenvalues of the adjacency matrix of $G^{\boldsymbol{S}}$ are the union of the eigenvalues of $\boldsymbol{A}$ and the eigenvalues of $\boldsymbol{S}$.

The graphs $G^{\boldsymbol{S}}$ that we form this way are called 2-lifts of G.

For your convenience, I now recall the solution to this problem.

Let $\boldsymbol{A}_+$ be the matrix with entries

$$\boldsymbol{A}_+(u,v) = \begin{cases} 1 & \text{if } \boldsymbol{S}(u,v) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

Let $\boldsymbol{A}_- = -(\boldsymbol{S} - \boldsymbol{A}_+)$. So,

$$\boldsymbol{S} = \boldsymbol{A}_+ - \boldsymbol{A}_- \quad \text{and} \quad \boldsymbol{A} = \boldsymbol{A}_+ + \boldsymbol{A}_-.$$

The adjacency matrix of $G^{\boldsymbol{S}}$ can be expressed in terms of these matrices as

$$\boldsymbol{A}^{\boldsymbol{S}} \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{A}_+ & \boldsymbol{A}_- \\ \boldsymbol{A}_- & \boldsymbol{A}_+ \end{pmatrix}.$$

Let $\boldsymbol{\psi}_1, \dots, \boldsymbol{\psi}_n$ be an orthonormal basis of eigenvectors of $\boldsymbol{A}$ of eigenvalues $\lambda_1, \dots, \lambda_n$, and let $\boldsymbol{\phi}_1, \dots, \boldsymbol{\phi}_n$ be an orthonormal basis of eigenvectors of $\boldsymbol{S}$ of eigenvalues $\mu_1, \dots, \mu_n$. We will prove that the vectors

$$\boldsymbol{\psi}_i^+ \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{\psi}_i \\ \boldsymbol{\psi}_i \end{pmatrix} \quad \text{and} \quad \boldsymbol{\phi}_i^- \overset{\text{def}}{=} \begin{pmatrix} \boldsymbol{\phi}_i \\ -\boldsymbol{\phi}_i \end{pmatrix}$$

are an orthogonal basis of $2n$ eigenvectors of $\boldsymbol{A}^{\boldsymbol{S}}$ with eigenvalues $\lambda_1, \dots, \lambda_n$ and $\mu_1, \dots, \mu_n$.

For $i \neq j$, it is immediately clear that $\boldsymbol{\psi}_i^+$ and $\boldsymbol{\psi}_j^+$ are orthogonal, and that $\boldsymbol{\phi}_i^+$ and $\boldsymbol{\phi}_j^+$ are orthogonal. Also, for every i and j,

$$(\boldsymbol{\psi}_i^+)^T \boldsymbol{\phi}_j^- = \begin{pmatrix} \boldsymbol{\psi}_i \\ \boldsymbol{\psi}_i \end{pmatrix}^T \begin{pmatrix} \boldsymbol{\phi}_i \\ -\boldsymbol{\phi}_i \end{pmatrix} = \boldsymbol{\psi}_i^T \boldsymbol{\phi}_j - \boldsymbol{\psi}_i^T \boldsymbol{\phi}_j = 0.$$

To show that these are the eigenvectors with the claimed eigenvalues, compute

$$\boldsymbol{A}^{\boldsymbol{S}} \boldsymbol{\psi}_i^+ = \begin{pmatrix} \boldsymbol{A}_+ & \boldsymbol{A}_- \\ \boldsymbol{A}_- & \boldsymbol{A}_+ \end{pmatrix} \begin{pmatrix} \boldsymbol{\psi}_i \\ \boldsymbol{\psi}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_+ \boldsymbol{\psi}_i + \boldsymbol{A}_- \boldsymbol{\psi}_i \\ \boldsymbol{A}_- \boldsymbol{\psi}_i + \boldsymbol{A}_+ \boldsymbol{\psi}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{A} \boldsymbol{\psi}_i \\ \boldsymbol{A} \boldsymbol{\psi}_i \end{pmatrix} = \lambda_i \begin{pmatrix} \boldsymbol{\psi}_i \\ \boldsymbol{\psi}_i \end{pmatrix},$$

and

$$\boldsymbol{A}^{\boldsymbol{S}} \boldsymbol{\phi}_i^- = \begin{pmatrix} \boldsymbol{A}_+ & \boldsymbol{A}_- \\ \boldsymbol{A}_- & \boldsymbol{A}_+ \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi}_i \\ -\boldsymbol{\phi}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_+ \boldsymbol{\phi}_i - \boldsymbol{A}_- \boldsymbol{\phi}_i \\ -\boldsymbol{A}_- \boldsymbol{\phi}_i + \boldsymbol{A}_+ \boldsymbol{\phi}_i \end{pmatrix} = \begin{pmatrix} \boldsymbol{S} \boldsymbol{\phi}_i \\ \boldsymbol{S} \boldsymbol{\phi}_i \end{pmatrix} = \mu_i \begin{pmatrix} \boldsymbol{\phi}_i \\ \boldsymbol{\phi}_i \end{pmatrix}.$$

Bilu and Linial [BL06] conjectured that every d-regular graph G has a signed adjacency matrix $\boldsymbol{S}$ so that $\|\boldsymbol{S}\| \leq 2\sqrt{d-1}$. This would give a simple procedure for constructing infinite families of Ramanujan graphs. We would begin with any small d-regular Ramanujan graph, such as the complete graph on $d+1$ vertices. Then, given any d-regular Ramanujan graph we could construct a new Ramanujan graph on twice as many vertices by using $G^{\boldsymbol{S}}$ where $\|\boldsymbol{S}\| \leq 2\sqrt{d-1}$.

We will prove something close to their conjecture.

Theorem 46.2.1. *Every d-regular graph G has a signed adjacency matrix $\boldsymbol{S}$ for which the maximum eigenvalue of $\boldsymbol{S}$ is at most $2\sqrt{d-1}$.*

We can use this theorem to build infinite families of bipartite Ramanujan graphs, because their eigenvalues are symmetric about the origin. Thus, if $\mu_2 \leq 2\sqrt{d-1}$, then we know that $|\mu_i| \leq 2\sqrt{d-1}$ for all $1 < i < n$. Note that the 2-lift of a bipartite graph is also a bipartite graph.

46.3 Random 2-Lifts

We will prove Theorem 46.2.1 by considering a random 2-lift, and then applying the method of interlacing polynomials. In particular, we consider

$$\mathbb{E}\left[\chi_x(\boldsymbol{S})\right]. \tag{46.1}$$

Godsil and Gutman [GG81] proved that this is equal to the matching polynomial of G!

Lemma 46.3.1. *Let G be a graph and let $\boldsymbol{S}$ be a uniform random signed adjacency matrix of G. Then,*

$$\mathbb{E}\left[\chi_x(\boldsymbol{S})\right] = \mu_x\left[G\right].$$

Proof. Expand the expected characteristic polynomial as

$$\begin{aligned}
\mathbb{E}\left[\chi_x(\boldsymbol{S})\right] &= \mathbb{E}\left[\det(x\boldsymbol{I} - \boldsymbol{S})\right] \\
&= \mathbb{E}\left[\sum_{\pi \in S_n} (-1)^{\mathsf{sgn}(\pi)} x^{|\{a:\pi(a)=a\}|} \prod_{a:\pi(a)\neq a} (\boldsymbol{S}(a,\pi(a))).\right] \\
&= \sum_{\pi \in S_n} (-1)^{\mathsf{sgn}(\pi)} x^{|\{a:\pi(a)=a\}|} \mathbb{E}\left[\prod_{a:\pi(a)\neq a} (\boldsymbol{S}(a,\pi(a)))\right].
\end{aligned}$$

As $\mathbb{E}\left[\boldsymbol{S}(a,\pi(a))\right] = 0$ for every a so that $\pi(a) \neq a$, the only way we can get a nonzero contribution from a permutation π is if for all a so that $\pi(a) \neq a$,

a. $(a,\pi(a)) \in E$, and

b. $\pi(\pi(a)) = a$.

The latter condition guarantees that whenever $\boldsymbol{S}(a, \pi(a))$ appears in the product, $\boldsymbol{S}(\pi(a), a)$ does as well. As these entries are constrained to be the same, their product is 1.

Thus, the only permutations that count are the involutions. As we saw last lecture, these correspond exactly to the matchings in the graph. □

Thus, we know that the largest root of (46.1) is at most $2\sqrt{d-1}$. So, all we need to do is to show that there is some signed adjacency matrix whose largest eigenvalue is at most this bound. We do this via the method of interlacing polynomials.

To this end, choose an ordering on the m edges of the graph. We can now associate each $\boldsymbol{S}$ with a vector $\sigma \in \{\pm 1\}^m$. Define

$$p_\sigma = \chi_x(\boldsymbol{S}).$$

The expected polynomial is the average of all these polynomials.

To form an interlacing family, we will form a tree that has the polynomials p_σ at the leaves. The intermediate nodes will correspond to choices of the first couple signs. That is, for $k < m$ and $\sigma \in \{\pm 1\}^k$ we define

$$p_\sigma(x) \overset{\text{def}}{=} \mathbb{E}_{\rho \in \{\pm 1\}^{n-k}} \left[p_{\sigma,\rho}(x) \right].$$

So, $p_\emptyset$ is the polynomial at the root of the tree. It remains to show that all pairs of siblings in the tree have a common interlacing.

Polynomials indexed by σ and τ are siblings if σ and τ have the same length, and only differ in their last index. To show that they have a common interlacing, we recall a few results from Lecture 22.

Lemma 46.3.2. *[Lemma 22.3.3] Let $\boldsymbol{A}$ be an n-dimensional symmetric matrix and let $\boldsymbol{v}$ be a vector. Let*

$$p_t(x) = \chi_x(\boldsymbol{A} + t\boldsymbol{v}\boldsymbol{v}^T).$$

Then there is a degree $n-1$ polynomial $q(x)$ so that

$$p_t(x) = \chi_x(\boldsymbol{A}) - tq(x).$$

Lemma 46.3.3. *[Lemma 22.3.2] Let p and q be polynomials of degree n and $n-1$, and let $p_t(x) = p(x) - tq(x)$. If p_t is real rooted for all $t \in \mathbb{R}$, then p and q interlace.*

Lemma 46.3.4. *[Lemma 22.3.1] Let p and q be polynomials of degree n and $n-1$ that interlace and have positive leading coefficients. For every $t > 0$, define $p_t(x) = p(x) - tq(x)$. Then, $p_t(x)$ is real rooted and*

$$p(x) \rightarrow p_t(x).$$

Lemma 46.3.5. *Let $p_0(x)$ and $p_1(x)$ be two degree n monic polynomials for which there is a third polynomial $r(x)$ that has the same degree as p_0 and p_1 and so that*

$$p_0(x) \rightarrow r(x) \quad \text{and} \quad p_1(x) \rightarrow r(x).$$

Then for all $0 \leq s \leq 1$,

$$p_s(x) \overset{\text{def}}{=} s p_1(x) + (1-s) p_0(x)$$

is a real rooted polynomial.

Theorem 46.3.6. *Let $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k$ be independently distributed random n-dimensional vectors and let $\boldsymbol{A}$ be a symmetric n-dimensional matrix. Then, the polynomial*

$$\mathbb{E}\left[\chi_x\left(\boldsymbol{A} + \sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i^T\right)\right]$$

is real rooted. Moreover, for every vector $\boldsymbol{u}$ in the support of $\boldsymbol{v}_k$, all the polynomials

$$\mathbb{E}\left[\chi_x\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{u}^T + \sum_{i=1}^{k-1} \boldsymbol{v}_i \boldsymbol{v}_i^T\right)\right]$$

have a common interlacing.

Proof. We prove this by induction on k. Assuming that we have proved it for k, we now prove it for $k+1$. Let $\boldsymbol{u}$ be any vector and let $t \in \mathbb{R}$. Define

$$p_t(x) = \mathbb{E}\left[\chi_x\left(\boldsymbol{A} + t\boldsymbol{u}\boldsymbol{u}^T + \sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i^T\right)\right].$$

By Lemma 46.3.2, we can express this polynomial in the form

$$p_t(x) = p_0(x) - tq(x),$$

where q has degree $n-1$. By induction, we know that $p_t(x)$ is real rooted for all t. Thus, Lemma 46.3.3 implies that $q(x)$ interlaces $p_0(x)$, and Lemma 46.3.4 tells us that for $t > 0$

$$p_0(x) \rightarrow p_t(x).$$

So, we may conclude that for every vector $\boldsymbol{u}$,

$$\mathbb{E}\left[\chi_x\left(\boldsymbol{A} + \sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i^T\right)\right] \rightarrow \mathbb{E}\left[\chi_x\left(\boldsymbol{A} + \boldsymbol{u}\boldsymbol{u}^T + \sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i^T\right)\right].$$

We now apply this result with each $\boldsymbol{u}$ from the support of $\boldsymbol{v}_{k+1}$ to conclude (via Lemma) that

$$\mathbb{E}\left[\chi_x\left(\boldsymbol{A} + \sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i^T\right)\right] \rightarrow \mathbb{E}\left[\chi_x\left(\boldsymbol{A} + \boldsymbol{v}_{k+1}\boldsymbol{v}_{k+1}^T + \sum_{i=1}^{k} \boldsymbol{v}_i \boldsymbol{v}_i^T\right)\right],$$

and that the latter polynomial is real rooted. □

To apply this theorem to the matrices $\boldsymbol{S}$, we must write them as a sum of outer products of random vectors. While we cannot do this, we can do something just as good. For each edge (a,b) of G, let $\boldsymbol{v}_{a,b}$ be the random vector that is $\boldsymbol{\delta}_a - \boldsymbol{\delta}_b$ with probability $1/2$ and $\boldsymbol{\delta}_a + \boldsymbol{\delta}_b$ with probability $1/2$. The random matrix $\boldsymbol{S}$ is distributed according to

$$\sum_{(a,b)\in E} \boldsymbol{v}_{a,b}\boldsymbol{v}_{a,b}^T - d\boldsymbol{I}.$$

Subtracting $d\boldsymbol{I}$ shifts the roots by d, and so does not impact any results we have proved about interlacing or real rootedness.

Bibliography

[ABN$^+$92] Noga Alon, Jehoshua Bruck, Joseph Naor, Moni Naor, and Ron M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Transactions on Information Theory*, 38(2):509–516, March 1992. 61, 240

[ABN08] I. Abraham, Y. Bartal, and O. Neiman. Nearly tight low stretch spanning trees. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 781–790, Oct. 2008. 282

[AC88] Noga Alon and Fan Chung. Explicit construction of linear sized tolerant networks. *Discrete Mathematics*, 72:15–19, 1988. 209

[AH77a] Kenneth Appel and Wolfgang Haken. Every planar map is four colorable part i. discharging. *Illinois Journal of Mathematics*, 21:429–490, 1977. 152

[AH77b] Kenneth Appel and Wolfgang Haken. Every planar map is four colorable part ii. reducibility. *Illinois Journal of Mathematics*, 21:491–567, 1977. 152

[AKPW95] Noga Alon, Richard M. Karp, David Peleg, and Douglas West. A graph-theoretic game and its application to the k-server problem. *SIAM Journal on Computing*, 24(1):78–100, February 1995. 279

[AKV02] Noga Alon, Michael Krivelevich, and Van H. Vu. On the concentration of eigenvalues of random symmetric matrices. *Israel Journal of Mathematics*, 131(1):259–267, 2002. 180

[Alo86] N. Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986. 164

[AM85] Noga Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *J. Comb. Theory, Ser. B*, 38(1):73–88, 1985. 164

[AN12] Ittai Abraham and Ofer Neiman. Using petal-decompositions to build a low stretch spanning tree. In *Proceedings of the 44th Annual ACM Symposium on the Theory of Computing (STOC '12)*, pages 395–406, 2012. 279

[AR94] Noga Alon and Yuval Roichman. Random cayley graphs and expanders. *Random Structures & Algorithms*, 5(2):271–284, 1994. 59

[AS06] A. Ashikhmin and V. Skachek. Decoding of expander codes at rates close to capacity. *IEEE Transactions on Information Theory*, 52(12):5475–5485, Dec. 2006. 230

[AW02] R. Ahlswede and A. Winter. Strong converse for identification via quantum channels. *Information Theory, IEEE Transactions on*, 48(3):569–579, 2002. 247

[AZLO15] Zeyuan Allen-Zhu, Zhenyu Liao, and Lorenzo Orecchia. Spectral sparsification and regret minimization beyond matrix multiplicative updates. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 237–245. ACM, 2015. 259

[Bab79] László Babai. Spectra of cayley graphs. *Journal of Combinatorial Theory, Series B*, pages 180–189, 1979. 62

[Bab80] László Babai. On the complexity of canonical labeling of strongly regular graphs. *SIAM Journal on Computing*, 9(1):212–216, 1980. 309

[Bab81] László Babai. Moderately exponential bound for graph isomorphism. In *Fundamentals of Computation Theory*, number 117 in Lecture Notes in Math, pages 34–50. Springer-Verlag, Berlin-Heidelberg-New York, 1981. 295

[Bab16] László Babai. Graph isomorphism in quasipolynomial time. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 684–697. ACM, 2016. 295, 312

[Bar82] Earl R. Barnes. An algorithm for partitioning the nodes of a graph. *SIAM Journal on Algebraic and Discrete Methods*, 3(4):541–550, 1982. 192

[BCS$^+$13] László Babai, Xi Chen, Xiaorui Sun, Shang-Hua Teng, and John Wilmes. Faster canonical forms for strongly regular graphs. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 157–166. IEEE, 2013. 312

[BGM82] László Babai, D Yu Grigoryev, and David M Mount. Isomorphism of graphs with bounded eigenvalue multiplicity. In *Proceedings of the fourteenth annual ACM symposium on Theory of computing*, pages 310–324. ACM, 1982. 14, 295

[BH01] Erik Boman and B. Hendrickson. On spanning tree preconditioners. Manuscript, Sandia National Lab., 2001. 280

[BL83] László Babai and Eugene M Luks. Canonical labeling of graphs. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, pages 171–183. ACM, 1983. 295

[BL06] Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap*. *Combinatorica*, 26(5):495–519, 2006. 314, 315, 358, 360

[BLM15] Charles Bordenave, Marc Lelarge, and Laurent Massoulié. Non-backtracking spectrum of random graphs: community detection and non-regular ramanujan graphs. *arXiv preprint arXiv:1501.06087*, 2015. 177

[BLR10] P. Biswal, J. Lee, and S. Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *Journal of the ACM*, 2010. to appear. 198

[BMS93] Richard Beigel, Grigorii Margulis, and Daniel A. Spielman. Fault diagnosis in a small constant number of parallel testing rounds. In *SPAA '93: Proceedings of the fifth annual ACM symposium on Parallel algorithms and architectures*, pages 21–29, New York, NY, USA, 1993. ACM. 209

[Bol86] Béla Bollobás. *Combinatorics: set systems, hypergraphs, families of vectors, and combinatorial probability*. Cambridge University Press, 1986. 51

[BR97] R. B. Bapat and T. E. S. Raghavan. *Nonnegative Matrices and Applications*. Number 64 in Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1997. 36

[BRSW12] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-source dispersers for $n^{o(1)}$ entropy, and ramsey graphs beating the frankl-wilson construction. *Annals of Mathematics*, pages 1483–1543, 2012. 155

[BSS12] Joshua Batson, Daniel A Spielman, and Nikhil Srivastava. Twice-Ramanujan sparsifiers. *SIAM Journal on Computing*, 41(6):1704–1721, 2012. 210, 252

[BSS14] Joshua Batson, Daniel A Spielman, and Nikhil Srivastava. Twice-ramanujan sparsifiers. *SIAM Review*, 56(2):315–334, 2014. 252

[BW13] László Babai and John Wilmes. Quasipolynomial-time canonical form for steiner designs. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 261–270. ACM, 2013. 312

[BZ02] A. Barg and G. Zemor. Error exponents of expander codes. *IEEE Transactions on Information Theory*, 48(6):1725–1729, Jun 2002. 230

[BZ05] A. Barg and G. Zemor. Concatenated codes: serial and parallel. *IEEE Transactions on Information Theory*, 51(5):1625–1634, May 2005. 230

[BZ06] A. Barg and G. Zemor. Distance properties of expander codes. *IEEE Transactions on Information Theory*, 52(1):78–90, Jan. 2006. 230

[BZ13] Nick Bridle and Xiaojin Zhu. p-voltages: Laplacian regularization for semi-supervised learning on high-dimensional data. In *Eleventh Workshop on Mining and Learning with Graphs (MLG2013)*, 2013. 146

[CCL$^+$15] Dehua Cheng, Yu Cheng, Yan Liu, Richard Peng, and Shang-Hua Teng. Efficient sampling for gaussian graphical models via spectral sparsification. In Peter Grünwald, Elad Hazan, and Satyen Kale, editors, *Proceedings of The 28th Conference on Learning Theory*, volume 40 of *Proceedings of Machine Learning Research*, pages 364–390, Paris, France, 03–06 Jul 2015. PMLR. 119

[CFM94] F. R. K. Chung, V. Faber, and T. A. Manteuffel. On the diameter of a graph from eigenvalues associated with its Laplacian. *SIAM Journal on Discrete Mathematics*, 7:443–457, 1994. 274

[CGP$^+$18] Timothy Chu, Yu Gao, Richard Peng, Sushant Sachdeva, Saurabh Sawlani, and Junxing Wang. Graph sparsification, spectral sketches, and faster resistance computation, via short cycle decompositions. *arXiv preprint arXiv:1805.12051*, 2018. 251

[CH91] Joel E Cohen and Paul Horowitz. Paradoxical behaviour of mechanical and electrical networks. 1991. 138

[Che70] J. Cheeger. A lower bound for smallest eigenvalue of the Laplacian. In *Problems in Analysis*, pages 195–199, Princeton University Press, 1970. 164

[Chi92] Patrick Chiu. Cubic Ramanujan graphs. *Combinatorica*, 12(3):275–285, 1992. 314

[Chu97] F. R. K. Chung. *Spectral Graph Theory.* American Mathematical Society, 1997. 161

[CKM$^+$14] Michael B. Cohen, Rasmus Kyng, Gary L. Miller, Jakub W. Pachocki, Richard Peng, Anup B. Rao, and Shen Chen Xu. Solving sdd linear systems in nearly mlog1/2n time. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 343–352, New York, NY, USA, 2014. ACM. 281, 294

[CLM$^+$15] Michael B Cohen, Yin Tat Lee, Cameron Musco, Christopher Musco, Richard Peng, and Aaron Sidford. Uniform sampling for matrix approximation. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science*, pages 181–190. ACM, 2015. 251

[dCSHS11] Marcel K. de Carli Silva, Nicholas J. A. Harvey, and Cristiane M. Sato. Sparse sums of positive semidefinite matrices. *CoRR*, abs/1107.0088, 2011. 259

[DH72] W. E. Donath and A. J. Hoffman. Algorithms for partitioning graphs and computer logic based on eigenvectors of connection matrices. *IBM Technical Disclosure Bulletin*, 15(3):938–944, 1972. 192

[DH73] W. E. Donath and A. J. Hoffman. Lower bounds for the partitioning of graphs. *IBM Journal of Research and Development*, 17(5):420–425, September 1973. 192

[DK70] Chandler Davis and William Morton Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970. 180

[DKMZ11] Aurelien Decelle, Florent Krzakala, Cristopher Moore, and Lenka Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Physical Review E*, 84(6):066106, 2011. 177

[Dod84] Jozef Dodziuk. Difference equations, isoperimetric inequality and transience of certain random walks. *Transactions of the American Mathematical Society*, 284(2):787–794, 1984. 164

[DS91] Persi Diaconis and Daniel Stroock. Geometric bounds for eigenvalues of Markov chains. *The Annals of Applied Probability*, 1(1):36–61, 1991. 39

[Duf47] R. J. Duffin. Nonlinear networks. IIa. *Bull. Amer. Math. Soc*, 53:963–971, 1947. 143

[dV90] Colin de Verdière. Sur un nouvel invariant des graphes et un critère de planarité. *J. Combin. Theory Ser. B*, 50:11–21, 1990. 200

[EEST08] Michael Elkin, Yuval Emek, Daniel A. Spielman, and Shang-Hua Teng. Lower-stretch spanning trees. *SIAM Journal on Computing*, 32(2):608–628, 2008. 282

[Eli55] Peter Elias. Coding for noisy channels. *IRE Conv. Rec.*, 3:37–46, 1955. 220

[Erd47] Paul Erdös. Some remarks on the theory of graphs. *Bulletin of the American Mathematical Society*, 53(4):292–294, 1947. 155

[Fan49] Ky Fan. On a theorem of weyl concerning eigenvalues of linear transformations i. *Proceedings of the National Academy of Sciences of the United States of America*, 35(11):652, 1949. 30

[Fie73] M. Fiedler. Algebraic connectivity of graphs. *Czechoslovak Mathematical Journal*, 23(98):298–305, 1973. 16

[Fie75] M. Fiedler. A property of eigenvectors of nonnegative symmetric matrices and its applications to graph theory. *Czechoslovak Mathematical Journal*, 25(100):618–633, 1975. 187, 190

[FK81] Z. Füredi and J. Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981. 71

[FK98] Uriel Feige and Joe Kilian. Zero knowledge and the chromatic number. *Journal of Computer and System Sciences*, 57(2):187–199, 1998. 153

[Fri08] Joel Friedman. *A Proof of Alon's Second Eigenvalue Conjecture and Related Problems*. Number 910 in Memoirs of the American Mathematical Society. American Mathematical Society, 2008. 209, 314, 344

[Fro12] Georg Frobenius. Über matrizen aus nicht negativen elementen. 1912. 36

[Gal63] R. G. Gallager. *Low Density Parity-Check Codes*. MIT Press, Cambridge, MA, 1963. 229

[Gee12] Jim Geelen. On how to draw a graph. https://www.math.uwaterloo.ca/∼jfgeelen/Publications/tutte.pdf, 2012. 129

[GG81] C. D. Godsil and I. Gutman. On the matching polynomial of a graph. In L. Lovász and Vera T. Sós, editors, *Algebraic Methods in graph theory*, volume I of *Colloquia Mathematica Societatis János Bolyai, 25*, pages 241–249. János Bolyai Mathematical Society, 1981. 316, 322, 323, 360

[GGT06] Steven J Gortler, Craig Gotsman, and Dylan Thurston. Discrete one-forms on meshes and applications to 3d mesh parameterization. *Computer Aided Geometric Design*, 23(2):83–112, 2006. 129

[Gil98] David Gillman. A chernoff bound for random walks on expander graphs. *SIAM Journal on Computing*, 27(4):1203–1220, 1998. 244

[GLM99] S. Guattery, T. Leighton, and G. L. Miller. The path resistance method for bounding the smallest nontrivial eigenvalue of a Laplacian. *Combinatorics, Probability and Computing*, 8:441–460, 1999. 39

[GLSS18] Ankit Garg, Yin Tat Lee, Zhao Song, and Nikhil Srivastava. A matrix expander chernoff bound. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1102–1114. ACM, 2018. 244

[GN08] C.D. Godsil and M.W. Newman. Eigenvalue bounds for independent sets. *Journal of Combinatorial Theory, Series B*, 98(4):721 – 734, 2008. 153

[God81] C. D. Godsil. Matchings and walks in graphs. *Journal of Graph Theory*, 5(3):285–297, 1981. 322, 352

[God93] Chris Godsil. *Algebraic Combinatorics*. Chapman & Hall, 1993. 54, 322, 352

[Gol07] Oded Goldreich. *Foundations of cryptography: volume 1, basic tools*. Cambridge university press, 2007. 239

[Hal70] K. M. Hall. An r-dimensional quadratic placement algorithm. *Management Science*, 17:219–229, 1970. 29

[Har76] Sergiu Hart. A note on the edges of the n-cube. *Discrete Mathematics*, 14(2):157–163, 1976. 51

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. 239, 240

[HL72] Ole J Heilmann and Elliott H Lieb. Theory of monomer-dimer systems. *Communications in Mathematical Physics*, 25(3):190–232, 1972. 316, 322, 352

[HL09] Mark Herbster and Guy Lever. Predicting the labelling of a graph via minimum p-seminorm interpolation. In *Proceedings of the 2009 Conference on Learning Theory (COLT)*, 2009. 146

[Hof70] A. J. Hoffman. On eigenvalues and colorings of graphs. In *Graph Theory and its Applications*, pages 79–92. Academic Press, New York, 1970. 153

[HPS15] Chris Hall, Doron Puder, and William F Sawin. Ramanujan coverings of graphs. *arXiv preprint arXiv:1506.02335*, 2015. 314, 321, 329, 358

[IZ89] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *30th annual IEEE Symposium on Foundations of Computer Science*, pages 248–253, 1989. 239

[JL84] William B Johnson and Joram Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. *Contemporary mathematics*, 26(189-206):1, 1984. 118

[JL97] Bruce W Jordan and Ron Livné. Ramanujan local systems on graphs. *Topology*, 36(5):1007–1024, 1997. 314

[Kel06] Jonathan A. Kelner. Spectral partitioning, eigenvalue bounds, and circle packings for graphs of bounded genus. *SIAM J. Comput.*, 35(4):882–902, 2006. 198

[KLL16] Tsz Chiu Kwok, Lap Chi Lau, and Yin Tat Lee. Improved cheeger's inequality and analysis of local graph partitioning using vertex expansion and expansion profile. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms*, pages 1848–1861. Society for Industrial and Applied Mathematics, 2016. 169

[KLP12] Ioannis Koutis, Alex Levin, and Richard Peng. Improved spectral sparsification and numerical algorithms for sdd matrices. In *STACS'12 (29th Symposium on Theoretical Aspects of Computer Science)*, volume 14, pages 266–277. LIPIcs, 2012. 251

[KLP$^+$16] Rasmus Kyng, Yin Tat Lee, Richard Peng, Sushant Sachdeva, and Daniel A Spielman. Sparsified cholesky and multigrid solvers for connection laplacians. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 842–850. ACM, 2016. 281

[KLPT09] Jonathan A. Kelner, James Lee, Gregory Price, and Shang-Hua Teng. Higher eigenvalues of graphs. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science*, 2009. 198

[KMP10] I. Koutis, G.L. Miller, and R. Peng. Approaching optimality for solving sdd linear systems. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 235 –244, 2010. 283

[KMP11] I. Koutis, G.L. Miller, and R. Peng. A nearly-mlogn time solver for sdd linear systems. In *Foundations of Computer Science (FOCS), 2011 52nd Annual IEEE Symposium on*, pages 590–598, 2011. 281, 283

[Kou14] Ioannis Koutis. Simple parallel and distributed algorithms for spectral graph sparsification. In *Proceedings of the 26th ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '14, pages 61–66, New York, NY, USA, 2014. ACM. 251

[KS16] Rasmus Kyng and Sushant Sachdeva. Approximate gaussian elimination for laplacians-fast, sparse, and simple. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 573–582. IEEE, 2016. 281

[Kur30] Casimir Kuratowski. Sur le probleme des courbes gauches en topologie. *Fundamenta mathematicae*, 15(1):271–283, 1930. 124

[LM82] F. Tom Leighton and Gary Miller. Certificates for graphs with distinct eigenvalues. Manuscript, 1982. 295

[LM00] Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000. 120

[Lov01] Làszlò Lovàsz. Steinitz representations of polyhedra and the Colin de Verdière number. *Journal of Combinatorial Theory, Series B*, 82(2):223 – 236, 2001. 15, 201

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988. 18, 61, 212, 240, 314

[LPS15] Yin Tat Lee, Richard Peng, and Daniel A. Spielman. Sparsified cholesky solvers for SDD linear systems. *CoRR*, abs/1506.08204, 2015. 251, 294

[LR97] John D. Lafferty and Daniel N. Rockmore. Spectral techniques for expander codes. In *STOC '97: Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 160–167, New York, NY, USA, 1997. ACM. 230

[LRT79] Richard J Lipton, Donald J Rose, and Robert Endre Tarjan. Generalized nested dissection. *SIAM journal on numerical analysis*, 16(2):346–358, 1979. 262

[LS88] Gregory F. Lawler and Alan D. Sokal. Bounds on the l^2 spectrum for Markov chains and Markov processes: A generalization of Cheeger's inequality. *Transactions of the American Mathematical Society*, 309(2):557–580, 1988. 164

[LS90] L. Lovàsz and M. Simonovits. The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume. In IEEE, editor, *Proceedings: 31st Annual Symposium on Foundations of Computer Science: October 22–24, 1990, St. Louis, Missouri*, volume 1, pages 346–354, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 1990. IEEE Computer Society Press. 130

[LS98] Làszlò Lovàsz and Alexander Schrijver. A borsuk theorem for antipodal links and a spectral characterization of linklessly embeddable graphs. *Proceedings of the American Mathematical Society*, 126(5):1275–1285, 1998. 200

[LS15] Yin Tat Lee and He Sun. Constructing linear-sized spectral sparsification in almost-linear time. *arXiv preprint arXiv:1508.03261*, 2015. 259

[LSY18] Yang P Liu, Sushant Sachdeva, and Zejun Yu. Short cycles via low-diameter decompositions. *arXiv preprint arXiv:1810.05143*, 2018. 251

[LV99] László Lovász and Katalin Vesztergombi. Geometric representations of graphs. *Paul Erdos and his Mathematics*, 1999. 129

[Mar88] G. A. Margulis. Explicit group theoretical constructions of combinatorial schemes and their application to the design of expanders and concentrators. *Problems of Information Transmission*, 24(1):39–46, July 1988. 18, 61, 212, 240, 314

[Mas14] Laurent Massoulié. Community detection thresholds and the weak ramanujan property. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 694–703. ACM, 2014. 177

[McS01] F. McSherry. Spectral partitioning of random graphs. In *FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science*, page 529, 2001. 177

[Mil51] William Millar. Cxvi. some general theorems for non-linear systems possessing resistance. *Philosophical Magazine*, 42(333):1150–1160, 1951. 146

[MNS08] Steven J. Miller, Tim Novikoff, and Anthony Sabelli. The distribution of the largest nontrivial eigenvalues in families of random regular graphs. *Experiment. Math.*, 17(2):231–244, 2008. 314

[MNS14] Elchanan Mossel, Joe Neeman, and Allan Sly. Belief propagation, robust reconstruction and optimal recovery of block models. In *Proceedings of The 27th Conference on Learning Theory*, pages 356–370, 2014. 177

[Mor94] M. Morgenstern. Existance and explicit constructions of $q + 1$ regular Ramanujan graphs for every prime power q. *Journal of Combinatorial Theory, Series B*, 62:44–62, 1994. 314

[MSS14] Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Ramanujan graphs and the solution of the Kadison-Singer problem. In *Proceedings of the International Congress of Mathematicians*, 2014. 259, 321

[MSS15a] A. W. Marcus, D. A. Spielman, and N. Srivastava. Finite free convolutions of polynomials. *arXiv preprint arXiv:1504.00350*, April 2015. 336, 344, 348

[MSS15b] Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families I: Bipartite Ramanujan graphs of all degrees. *Ann. of Math.*, 182-1:307–325, 2015. 314, 358

[MSS15c] Adam W. Marcus, Daniel A. Spielman, and Nikhil Srivastava. Interlacing families II: Mixed characteristic polynomials and the Kadison-Singer problem. *Ann. of Math.*, 182-1:327–350, 2015. 259

[MSS15d] Adam W Marcus, Nikhil Srivastava, and Daniel A Spielman. Interlacing families IV: Bipartite Ramanujan graphs of all sizes. *arXiv preprint arXiv:1505.08010*, 2015. appeared in Proceedings of the 56th IEEE Symposium on Foundations of Computer Science. 321, 329, 336, 344

[Nil91] A. Nilli. On the second eigenvalue of a graph. *Discrete Math*, 91:207–210, 1991. 212

[Obr63] Nikola Obrechkoff. *Verteilung und berechnung der Nullstellen reeller Polynome*. VEB Deutscher Verlag der Wissenschaften, Berlin, 1963. 320, 331

[Per07] Oskar Perron. Zur theorie der matrices. *Mathematische Annalen*, 64(2):248–263, 1907. 36

[Piz90] Arnold K Pizer. Ramanujan graphs and Hecke operators. *Bulletin of the AMS*, 23(1), 1990. 314

[PP03] Claude M Penchina and Leora J Penchina. The braess paradox in mechanical, traffic, and other networks. *American Journal of Physics*, 71:479, 2003. 138

[PS14] Richard Peng and Daniel A. Spielman. An efficient parallel solver for SDD linear systems. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 333–342, 2014. 281, 288

[PSL90] A. Pothen, H. D. Simon, and K.-P. Liou. Partitioning sparse matrices with eigenvectors of graphs. *SIAM J. Matrix Anal. Appl.*, 11:430–452, 1990. 192

[Rud99] M. Rudelson. Random vectors in the isotropic position,. *Journal of Functional Analysis*, 164(1):60 – 72, 1999. 246, 247

[RV07] Mark Rudelson and Roman Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *J. ACM*, 54(4):21, 2007. 247

[RVW02] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, 2002. 231, 238

[Sen06] Eugene Seneta. *Non-negative matrices and Markov chains.* Springer Science & Business Media, 2006. 36

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. *Bell system technical journal*, 27(3):379–423, 1948. 220

[Sim91] Horst D. Simon. Partitioning of unstructured problems for parallel processing. *Computing Systems in Engineering*, 2:135–148, 1991. 192

[SJ89] Alistair Sinclair and Mark Jerrum. Approximate counting, uniform generation and rapidly mixing Markov chains. *Information and Computation*, 82(1):93–133, July 1989. 39, 164

[Spi96a] D.A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, Nov 1996. 230

[Spi96b] Daniel A. Spielman. Faster isomorphism testing of strongly regular graphs. In *STOC '96: Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 576–584, New York, NY, USA, 1996. ACM. 312

[SS96] M. Sipser and D.A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, Nov 1996. 230

[SS11] D.A. Spielman and N. Srivastava. Graph sparsification by effective resistances. *SIAM Journal on Computing*, 40(6):1913–1926, 2011. 246, 251

[ST04] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *Proceedings of the thirty-sixth annual ACM Symposium on Theory of Computing*, pages 81–90, 2004. Full version available at `http://arxiv.org/abs/cs.DS/0310051`. 169

[ST07] Daniel A. Spielman and Shang-Hua Teng. Spectral partitioning works: Planar graphs and finite element meshes. *Linear Algebra and its Applications*, 421:284–305, 2007. 192, 198

[ST13] Daniel A Spielman and Shang-Hua Teng. A local clustering algorithm for massive graphs and its application to nearly linear time graph partitioning. *SIAM Journal on Computing*, 42(1):1–26, 2013. 169

[ST14] Daniel A. Spielman and Shang-Hua Teng. Nearly-linear time algorithms for preconditioning and solving symmetric, diagonally dominant linear systems. *SIAM. J. Matrix Anal. & Appl.*, 35:835–885, 2014. 269, 281, 283

[SW09] Daniel A. Spielman and Jaeoh Woo. A note on preconditioning by low-stretch spanning trees. *CoRR*, abs/0903.2816, 2009. Available at `http://arxiv.org/abs/0903.2816`. 281

[SW15] Xiaorui Sun and John Wilmes. Faster canonical forms for primitive coherent configurations. In *Proceedings of the forty-seventh annual ACM symposium on Theory of computing*, pages 693–702. ACM, 2015. 312

[Tan81] R. Michael Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, September 1981. 229

[Tan84] R. Michael Tanner. Explicit concentrators from generalized n-gons. *SIAM Journal Alg. Disc. Meth.*, 5(3):287–293, September 1984. 211

[Tre09] Luca Trevisan. Max cut and the smallest eigenvalue. In *STOC '09: Proceedings of the 41st annual ACM symposium on Theory of computing*, pages 263–272, 2009. 17

[Tre11] Luca Trevisan. Lecture 4 from cs359g: Graph partitioning and expanders, stanford university, January 2011. available at `http://theory.stanford.edu/ trevisan/cs359g/lecture04.pdf`. 164

[Tro12] Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012. 183, 246, 247

[Tut63] W. T. Tutte. How to draw a graph. *Proc. London Mathematical Society*, 13:743–768, 1963. 17, 122

[Vai90] Pravin M. Vaidya. Solving linear equations with symmetric diagonally dominant matrices by constructing good preconditioners. Unpublished manuscript UIUC 1990. A talk based on the manuscript was presented at the IMA Workshop on Graph Theory and Sparse Matrix Computation, October 1991, Minneapolis., 1990. 278, 283

[Val79] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979. 323

[van95] Hein van der Holst. A short proof of the planarity characterization of Colin de Verdière. *Journal of Combinatorial Theory, Series B*, 65(2):269 – 272, 1995. 203

[Var85] N. Th. Varopoulos. Isoperimetric inequalities and Markov chains. *Journal of Functional Analysis*, 63(2):215 – 239, 1985. 164

[Ver10] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010. 183

[Vis12] Nisheeth K. Vishnoi. $Lx = b$, 2012. available at `http://research.microsoft.com/en-us/um/people/nvishno/Site/Lxb-Web.pdf`. 268

[Voi97] Dan V Voiculescu. *Free probability theory.* American Mathematical Society, 1997. 336

[Vu07] Van Vu. Spectral norm of random matrices. *Combinatorica*, 27(6):721–736, 2007. 68, 71, 179

[Vu14] Van Vu. A simple svd algorithm for finding hidden partitions. *arXiv preprint arXiv:1404.3918*, 2014. 177

[Wal22] JL Walsh. On the location of the roots of certain types of polynomials. *Transactions of the American Mathematical Society*, 24(3):163–180, 1922. 337

[Wig58] Eugene P Wigner. On the distribution of the roots of certain symmetric matrices. *Ann. Math*, 67(2):325–327, 1958. 63

[Wil67] Herbert S. Wilf. The eigenvalues of a graph and its chromatic number. *J. London math. Soc.*, 42:330–332, 1967. 32, 35

[Zem01] G. Zemor. On expander codes. *IEEE Transactions on Information Theory*, 47(2):835–837, Feb 2001. 230

[ZKT85] V. M. Zemlyachenko, N. M. Kornienko, and R. I. Tyshkevich. Graph isomorphism problem. *Journal of Soviet Mathematics*, 29:1426–1481, 1985. 295

Index