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.