Lecture 4: Graph partitioning, basic linear algebra

We introduce two natural objectives for *graph partitioning*. Then we will see how these objectives have eigenvalue problems as a natural *relaxation*. We then review some basic linear algebra, and introduce the notions of the adjacency matrix and the Laplacian of a graph.

Graph partitioning – the objectives

Dividing a graph into two sets of vertices with *not much interaction* between them is a fundamental problem. Applications include community finding in social networks, designing divide and conquer algorithms, chip design, finding bottlenecks in communication networks, etc.

Broadly, we would like to divide the graph into two parts, such that each part is not too small, and the number of edges that go between the parts is small. There are many ways to formalize this, so we will motivate one by an application.

Routing with small congestion. Suppose we have an n vertex graph (V, E), and the goal is to design a *communication path* for every pair of vertices, i.e., for every $u, v \in V$, we want to construct a path P_{uv} that starts from u, ends at v, and consists of edges in the graph. The goal is to ensure that no edge is used in too many paths. Formally, the *congestion* of an edge e is defined to be the number of pairs (u, v) such that P_{uv} contains e. The aim is to construct a set of paths P_{uv} so as to minimize the maximum congestion.

Given a graph, is there a way to say that *no matter how the paths are chosen*, the max congestion has to be large? A little thought reveals that this is true if there are two large sets of vertices with too few edges between them – i.e., a *communication bottleneck* in the graph. Formally, suppose we divide the graph into S and $V \setminus S$, and find that there are only K edges that go across. Then, any path P_{uv} with $u \in S$ and $v \in V \setminus S$ must use one of the K edges. Since there are $|S||V \setminus S|$ such pairs, the max congestion, no matter how P_{uv}'s look like, is at least

$$\frac{|S||V \setminus S|}{K}.$$

The inverse of this quantity is defined as the *sparsity* of the cut $(S, V \setminus S)$:

$$\sigma(S) := \frac{|E(S, V \setminus S)|}{|S||V \setminus S|}.$$

In the routing application, we would like to find the S that minimizes $\sigma(S)$, as that gives the best lower bound for the congestion. This is referred to as the *sparsest cut* problem, $\min_{S \subseteq V} \sigma(S)$.

A related quantity is the *expansion* of a set of vertices S. It is defined as

$$\phi(S) := \frac{|E(S, V \setminus S)|}{\sum_{u \in S} \deg(u)},$$

where $\deg(u)$ denotes the degree of the vertex u. Expansion measures the fraction of the edges incident on the set S that leave the set S. In some applications, expansion turns out to be a more natural measure than

Graph partitioning – the objectives continued on next page...

sparsity.

However for regular graphs (all vertices have degree d, for some d), the expansion and sparsity are closely related. For sets of size $\leq n/2$, we have

$$\frac{\sigma(S)}{2} \leq \frac{d}{n}\phi(S) \leq \sigma(S).$$

The *least expanding set* is another useful parameter of a graph. I.e., we wish to find $\min_{S\subset V, |S|\leq n/2} \phi(S)$.[1] This quantity is often called the *expansion of the graph*, denoted $\Phi(G)$.

The hardness. Both the formulations above, finding the sparsest cut, and finding the expansion of a graph are known to be NP-hard. However, thier approximability is not completely settled – it is one of the big open problems in theory research. The best known algorithms have an approximation factor $O(\sqrt{\log n})$, obtained via semidefinite programming.

Remark 1. *Note that without the denominator, minimizing $E(S, V \setminus S)$ is simply the Min-Cut problem, which can be solved in polynomial time.*

Examples. It is instructive to work out the expansion and sparsest cut in the following graphs:

1. A cycle on n vertices
2. An $\sqrt{n} \times \sqrt{n}$ grid
3. A three dimensional grid ($n^{1/3} \times n^{1/3} \times n^{1/3}$)
4. A complete graph on n vertices

A relaxation for the sparsest cut

Assumption. For the rest of this lecture, let us restrict ourselves to d-regular graphs, for some parameter d. This will make all our arguments clean.

Let us begin by writing the sparsest cut problem as a 0/1 optimization problem. We observe that

$$\min_{S\subseteq V} \frac{E(S, V\setminus S)}{|S||V\setminus S|} = \min_{x\in\{0,1\}^n} \frac{\sum_{\{ij\}\in E} |x_i - x_j|}{\sum_{i,j} |x_i - x_j|}.$$

The sum in the numerator is over edges, and the one in the denominator is over all pairs i, j. To see the equality, note that there is a one-one correspondence between $S \subseteq V$ and $x \in \{0,1\}^n$, and that $|x_i - x_j|$ is zero if i, j are assigned the same value (0 or 1) and 1 otherwise.

Since $|x_i - x_j|$ takes only 0/1 values, we can write our minimization problem as

$$\min_{x\in\{0,1\}^n} \frac{\sum_{\{ij\}\in E} (x_i - x_j)^2}{\sum_{i,j} (x_i - x_j)^2}.$$

[1] Why the condition $|S| \leq n/2$? Because the numerator remains the same whether we pick S or its complement, and it only makes sense to use the smaller side.

Consider the *continuous relaxation* of the problem, in which we replace $x \in \{0,1\}^n$ by $x \in \mathbb{R}^n$. Let us call this quantity $\lambda(G)$. (By definition, $\lambda(G) \leq \min_S \sigma(S)$.)

$$\lambda(G) := \min_{x \in \mathbb{R}^n} \frac{\sum_{\{ij\} \in E} (x_i - x_j)^2}{\sum_{i,j} (x_i - x_j)^2}.$$

Now, note that both the numerator and the denominator of this expression are shift invariant, i.e., do not change if every x_i is replaced with $x_i + c$, for some c. Thus we can shift the x_i such that they are *centered*, i.e., $\sum_i x_i = 0$. (Formally, think of picking $c = \frac{-(x_1 + x_2 + \cdots + x_n)}{n}$.) Doing this gives a clean form for the denominator:

$$\sum_{i,j} (x_i - x_j)^2 = \sum_{i,j} x_i^2 + x_j^2 - 2x_i x_j \tag{1}$$

$$= \sum_{i,j} x_i^2 + x_j^2 - \sum_i \sum_j 2x_i x_j \tag{2}$$

$$= n \cdot \Big(\sum_i x_i^2 + \sum_j x_j^2 \Big) - 2 \sum_i x_i \Big(\sum_j x_j \Big) \tag{3}$$

$$= (2n) \sum_i x_i^2. \tag{4}$$

The last equality is because $\sum_j x_j = 0$. Thus we can rewrite

$$\lambda(G) = \frac{1}{2n} \cdot \min_{x \in \mathbb{R}^n, \sum_i x_i = 0} \frac{\sum_{\{i,j\} \in E} (x_i - x_j)^2}{\sum_i x_i^2}.$$

This is essentially the problem of minimizing, subject to $\sum_i x_i = 0$, the ratio $\frac{x^T M x}{x^T x}$, for some matrix M, which is an eigenvalue problem! (See below for a quick linear algebra recap.) Thus a natural relaxation of the sparsest cut problem gives an eigenvalue problem. A fundamental inequality (called Cheeger's inequality, which we will see next class) in spectral graph theory says that the relaxation indeed leads to a good way to find an S with a small value of $\sigma(S)$.

Review of Basic Linear Algebra

We will throughout deal with matrices that are real and symmetric, i.e., $n \times n$ matrices M with $M^T = M$. Every such matrix has n real eigenvalues, and n orthogonal eigenvectors. Let us denote by $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ the eigenvalues and $v_1, v_2, \ldots, v_n$ the corresponding eigenvectors (which we may assume are of unit norm).

Then we have $\langle v_i, v_j \rangle = 0$. The vectors v_i thus form an orthonormal basis for $\mathbb{R}^n$. Further, we have that we can write

$$M = \sum_i \lambda_i v_i v_i^T.$$

Since v_i form an orthonormal basis, any vector x of unit length can be written as $\sum_i \alpha_i v_i$, for some α_i (in fact, $\alpha_i = \langle x, v_i \rangle$), where $\sum_i \alpha_i^2 = 1$. A useful consequence of the above is that

$$Mx = \sum_i \lambda_i \alpha_i v_i.$$

Review of Basic Linear Algebra continued on next page...

This can easily be verified, noting that $v_i^T v_j = 0$ for $i \neq j$. Thus if a vector x has components α_i in the basis $v_1, \ldots, v_n$, then Mx has components $\alpha_i \lambda_i$ (coefficients get multiplied by the corresponding eigenvalue). Furthermore, we can see that $x^T Mx$ (a scalar) is

$$x^T Mx = \sum_i \lambda_i \alpha_i^2.$$

Since we are working with a unit vector x, i.e. $\sum_i \alpha_i^2 = 1$, we can think of $x^T Mx$ as a weighted average of the eigenvalues. This immediately leads us to an alternate definition of eigenvalues, which is quite useful.

Eigenvalues of a real symmetric matrix. Let M be an $n \times n$ real, symmetric matrix. Then we have

$$\lambda_1 = \min_{x \in \mathbb{R}^n} \frac{x^T Mx}{x^T x}.$$

Further, the vector x at which the minimum is attained is the corresponding eigenvector, say v_1. Then

$$\lambda_2 = \min_{\substack{x \in \mathbb{R}^n \\ x \perp v_1}} \frac{x^T Mx}{x^T x}.$$

Again, the vector x that attains this minimum is the corresponding eigenvector v_2. In general, we have

$$\lambda_{k+1} = \min_{\substack{x \in \mathbb{R}^n \\ x \perp \mathrm{span}(v_1, v_2, \ldots, v_k)}} \frac{x^T Mx}{x^T x}.$$

Non-symmetric matrices. For non-symmetric (and non-square) matrices, a useful analog of eigenvalues are the *singular values*. We will encounter them at a later point in the course.

For more links on Linear Algebra, please refer to the excellent texbook by Gilbert Strang. His video lectures on MIT's OpenCourseWare are also a great resource.

Linear Algebra of Graphs

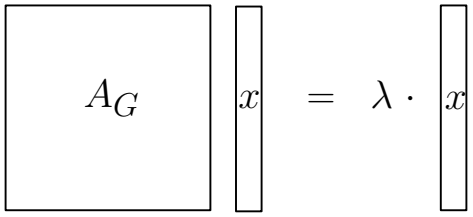
Let us now see some elementary connections between graphs and properties of matrices associated with them. The first matrix we associate with an n vertex graph is its so-called *adjacency matrix*:

The adjacency matrix of a graph G, denoted A_G is an $n \times n$ matrix whose ijth entry is 1 if $\{ij\}$ is an edge, and 0 otherwise.
Note that for an undirected graph, the adjacency matrix is symmetric. If the graph is d-regular, then every row and every column have precisely d non-zero entries. Thus scaling the matrix down by a factor d gives what is called a *doubly stochastic* matrix. We will see applications of it a couple of lectures later.

Let us try to understand a little about the eigenvalues of A_G. First, what is the largest eigenvalue of A_G?

Lemma 1. *For a d-regular graph G, we have $\lambda_{\max}(A_G) = d$.*

Proof. First, let us see why it should be at most d. Let λ be an eigenvalue. Thus by definition, we have $A_G x = \lambda x$. Pictorially, we have Now, suppose the largest coordinate in x, in magnitude, is x_i, for some i.



Then the equality above implies that

$$\lambda x_i = \sum_j A_{ij} x_j.$$

Taking absolute values, we have $|\lambda||x_i| \leq \sum_j |A_{ij}||x_j| \leq \sum_j |A_{ij}||x_i| \leq d|x_i|$ (where we used the fact that $|x_j| \leq |x_i|$ for all j, and that there are precisely d 1's in the ith row of A).

This implies that $|\lambda| \leq d$, or $-d \leq \lambda \leq d$. This is true for every eigenvalue, and so $\lambda_{\max} \leq d$.

Now, it is easy to see that we can arrange for equality in the above: suppose we set all x_i's equal – say to 1. Then we have $Ax = dx$. Thus $\lambda_{\max} = d$. □

In the next class, we will see that in any connected graph, there is exactly one eigenvalue which is d. Furthermore, the multiplicity of the eigenvalue d tells us precisely the number of connected components in a graph!

This is one of the first of many connections between the eigenvalues of a graph and its combinatorial properties (such as connectivity, bipartiteness, etc.) we will see in the course.