ITCSC-INC Winter School, 2013, CUHK

Jan 14: Spectral Graph Theory

What is the relation between the eigenvalues of the adjacency matrix of a graph and its combinatorial properties? Surprinsingly, eigenvalues and eigenvectors reveal much information about how to partition the graph. This connection is used in designing some of most efficient heuristics for graph partitioning problems, with applications in different areas of computer science.

Today, we will go through the basics of spectral graph theory, prove the Cheeger's inequality relating the second eigenvalue to the graph expansion, and discuss some recent developments relating other eigenvalues to graph partitioning problems.

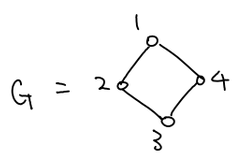
Introduction

Spectral graph theory studies the relations between the eigenvalues of a graph and its combinatorial properties. This allows us to use ideas from linear algebra to analyze combinatorial problems. Surprisingly this approach is very powerful, and it can solve combinatorial problems that no combinatorial methods are known yet.

Most of spectral graph theory is about undirected graphs, the objects that we study today.

Given an undirected graph $G=(V,E)$, we consider the adjacency matrix $A(G)$ of G.

e.g.



$$A(G) = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix} \end{array}$$

Note that $A(G)$ is symmetric, since G is undirected.

Given the adjacency matrix A, we can compute its eigenvalues and eigenvectors.

Recall that a vector $x \neq 0$ is an <u>eigenvector</u> of A if $Ax = \lambda x$ for some scalar λ, which we call the <u>eigenvalue</u> corresponding to the eigenvector x.

As we will explain later, since A is real and symmetric, it has a set of $|V|$ orthonormal eigenvectors $x_1, x_2, \ldots, x_{|V|}$, with corresponding eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_{|V|}$, all of them are real numbers.

Sort the eigenvalues so that $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \ldots \geq \lambda_{|V|}$. We call it the <u>spectrum</u> of G.

In the 4-cycle above, its spectrum is $[2, 0, 0, -2]$.

As I will try to convince you later, the spectrum reveals much information about the graph.

Some areas that the spectrum plays an important role includes graph partitioning, analysis of random walk, and solving linear equations.

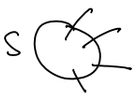
Each of these areas has a rich literature and has many applications to other areas.

Today we will focus on graph partitioning.

Graph Partitioning

Broadly speaking, graph partitioning problems ask how to divide the vertices into two or more groups, so as to minimize or maximize a certain objective function on the "crossing edges".

Given $S \subseteq V$, we define $\delta(S)$ to be the set of edges with one endpoint in S and another endpoint in $V-S$.



The followings are some well studied graph partitioning problems:

(i) minimum cut : find $\phi \neq S \neq V$ that minimizes $|\delta(S)|$; in order words, remove the minimum

number of edges to disconnect the graph.

(2) maximum cut : find an $S \subseteq V$ that maximizes $|\delta(S)|$.

(3) minimum bisection : find S that minimizes $|\delta(S)|$ with $|S| = |V|/2$.

(4) graph expansion : find S with $|S| \leq |V|/2$ that minimizes $|\delta(S)|/|S|$.

(5) minimum conductance : find S with $\sum_{v \in S} \deg(v) \leq |E|/2$ that minimizes $|\delta(S)| / \sum_{v \in S} \deg(v)$.

(6) sparsest cut : find S that minimizes $|\delta(S)|/(|S| \cdot |V - S|)$.

(7) multiway partitioning : partition V into $\{S_1, S_2, \ldots, S_k\}$ and minimize the maximum conductance.

Except (1), all other problems are NP-hard.

Our goal is to find good approximation algorithms for these problems, with provable performance guarantees on their output solutions.

For simplicity, we assume the input graph is <u>d-regular</u>. All the results can be generalized to arbitrary weighted graphs with suitable modifications, that we will mention later.

Observe that (4) and (6) are within a factor of $|V|/2$ to a factor of $|V|$ of each other.

When the graph is d-regular, (4) and (5) are exactly a factor of d of each other.

Also, if we can solve (4), then we can somewhat solve (3).

Therefore, for (3)-(6), we only consider the minimum conductance problem (5), and it will be the main problem that we focus on. At the end, we will also mention (2) and (7).

Okay, let $\phi(S) := \frac{|\delta(S)|}{d|S|}$ be the conductance of a subset S, and let

$$\phi(G) := \min_{|S| \leq |V|/2} \phi(S)$$ be the conductance of the graph.

will explain later

We call a set S a <u>sparse cut</u> if $\phi(S)$ is small.

We call a set S a sparse cut if $\phi(S)$ is small.

Approximating $\phi(G)$ is a fundamental problem in theoretical computer science, and it has various applications in different areas of computer science, including image segmentation [1], clustering, community detection in social networks, etc.

In some applications, we would like to find the "ground truth" under some noise.

In this sense, the topic today fits in the winter school's them "dealing with noise".

The Spectral Partitioning Algorithm

A popular heuristic in finding a sparse cut in practice is the following spectral partitioning algorithm.

---

(1) Compute the second eigenvector x of A (the eigenvector corresponding to the second largest eigenvalue)

(2) Sort the vertices so that $x(1) \geq x(2) \geq \dots \geq x(n)$ (where $n = |V|$ is the number of vertices)

(3) Let $S_i = \begin{cases} \{1, \dots, i\} & \text{if } i \leq n/2 \\ \{i+1, \dots, n\} & \text{if } i > n/2 \end{cases}$.

Return $\min_i \{\phi(S_i)\}$.

That's the algorithm.

First, there is an almost linear time algorithm (in terms of number of edges) to compute (approximately) the second eigenvector of the adjacency matrix. It is known as the "power method", which we won't discuss today. So, the whole algorithm can be implemented in near linear time, quite easily especially if you use some mathematical software (e.g. MATLAB). This is one reason that this heuristic is popular.

that this heuristic is popular.

Another reason is that it performs very well in various applications, especially in image segmentation and clustering, and it was considered a breakthough in image segmentation about ten years ago [1]. While researchers have some intuitions about why this algorithm would work well, it has been an open problem to explain this rigorously. We will come back to that.

Let's see a demostration program before we move on. (http://www.cse.cuhk.edu.hk/~chi/fyp2.html)

Overview

Our goal is to analyze the performance of the spectral partitioning algorithm, and more generally to understand deeper the connection between eigenvalues and graph partitioning.

Today's agenda:

0. Review of linear algebra
1. Basic spectral graph theory
2. Cheeger's inequality: informally, $\phi(G)$ is small "if and only if" λ_2 is close to λ_1.
   We will do the full proof, and mention its importance in theoretical computer science.
3. Recent developments through higher eigenvalues:
   - maximum cut and λ_n (Trevisan, 2009 [2])
   - small set expansion, multiway partitioning and λ_k (Arora, Barak, Steurer 2010 [3], Lee, Oveis Gharan, Trevisan 2012 [4a], Louis, Raghavendra, Tetali, Vempala 2012 [4b])
   - minimum conductance and λ_k (Kwok, Lau, Lee, Oveis Gharan, Trevisan [5])

   We will just see the algorithm and some intuitions, to give you an idea of what research is like.

Linear Algebra

We will work throughout with vectors over reals.

Linear independence: A set of vectors $x_1, x_2, \ldots, x_k$ are <u>linear independent</u> if $c_1x_1 + c_2x_2 + \ldots + c_kx_k = 0$ implies $c_1 = c_2 = \ldots = c_k = 0$ where $c_1, c_2, \ldots, c_k \in \mathbb{R}$; otherwise they are <u>linearly dependent</u>.

Nullspace: The <u>nullspace</u> of a matrix M is defined as $\text{nullspace}(M) := \{ x \mid Mx = 0 \}$. Its <u>dimension</u> (the maximum number of linearly independent vectors in $\text{nullspace}(M)$) is denoted by $\text{null}(M)$.

Determinant: The <u>determinant</u> of an $n \times n$ matrix M, denoted by $\det(M)$, is defined as:

$$\det(M) = \sum_{\sigma \in S^n} \text{sgn}(\sigma) \prod_{i=1}^{n} M(i, \sigma(i)),$$

where $\sigma : [n] \to [n]$ is a permutation of the indices, and $\text{sgn}(\sigma) = +1$ if σ is even and $\text{sgn}(\sigma) = -1$ if σ is odd, i.e. $\text{sgn}(\sigma) = (-1)^{\text{inv}(\sigma)}$ where $\text{inv}(\sigma) = |\{(i,j) \mid i < j \text{ and } \sigma(i) > \sigma(j)\}|$.

Note that $\det(M)$ is a degree n multivariate polynomial of its entries.

Also, $\det(M)$ can be computed in polynomial time by Gaussian elimination.

Eigenvectors and eigenvalues: To compute the eigenvalues of a matrix, we can solve the equation $Mx = \lambda x$.

Note that $Mx = \lambda x \iff (M - \lambda I)x = 0 \iff \text{nullspace}(M - \lambda I) \neq 0 \iff \det(M - \lambda I) = 0$.

Recall that $\det(M - \lambda I)$ is a degree n polynomial of λ (assuming entries of M are constants), and this is called the <u>characteristic polynomial</u> of M.

From this, we see that there are at most n distinct eigenvalues of M.

Any root of this characteristic polynomial is an eigenvalue, and any vector x in

nullspace $(M - \lambda I)$ is an eigenvector corresponding to λ.

The characteristic polynomial can be computed in polynomial time.

Note that a matrix M may not have any real eigenvalue (e.g. a rotation matrix).

Orthogonality: Given two vectors $x, y \in \mathbb{R}^n$, the <u>inner product</u> $\langle x, y \rangle := \sum_{i=1}^{n} x(i)\, y(i)$.

The <u>norm</u> of a vector x is $\|x\| := \sqrt{\langle x, x \rangle}$.

Two vectors x and y are <u>orthogonal</u> if $\langle x, y \rangle = 0$.

A set S of vectors are orthogonal if $\langle x, y \rangle = 0$ for any pair $x, y \in S$.

A set S of vectors are <u>orthonormal</u> if they are orthogonal and $\|x\| = 1 \ \forall\, x \in S$.

A set S of vectors is called a <u>basis</u> of a vector space V if they are linear independent and every vector in V can be written as a linear combination of the vectors in S.

Observe that any set of n orthogonal vectors form a basis, and any basis can be converted into an orthonormal basis (by the Gram-Schmidt process).

<u>Spectral theorem for real symmetric matrix</u> Let M be an $n \times n$ real symmetric matrix.

Then (1) there is an <u>orthonormal basis of eigenvectors</u> of M, and

(2) all eigenvalues are real.

It is good to know the proof, but it is not essential to understand the material follows.

So, we will skip the proof, and the interested reader can read the notes in [6].

Multiplicity: The spectral theorem says that the characteristic polynomial of M can be written as $\prod_{i=1}^{n} (\lambda - \lambda_i)$ where λ_i are real numbers.

$i=1$ (...) ... [illegible]

It is possible that some values appear more than once.

If some eigenvalue appears k times, then we say it has ~~multiplicity~~ k.

The multiplicity of λ is equal to the dimension of nullspace$(M-\lambda I)$.

Trace = Sum of Eigenvalues

For a matrix M, the trace of M, denoted by trace(M), is defined as $\sum_{i=1}^{n} M(i,i)$, i.e. the sum of diagonal entries of M.

Fact Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be the eigenvalues of M. Then $\text{trace}(M) = \sum_{i=1}^{n} \lambda_i$.

proof Consider $\det(M-\lambda I) = (\lambda-\lambda_1)(\lambda-\lambda_2)\ldots(\lambda-\lambda_n)$.

The coefficient of λ^{n-1} is $-\sum_{i=1}^{n} \lambda_i$.

On the other hand, $\det(\lambda I - M) = \det \begin{pmatrix} \lambda-M(1,1) & -M(1,2) & \cdots & -M(1,n) \\ -M(2,1) & \lambda-M(2,2) & \cdots & -M(2,n) \\ \vdots & \vdots & \ddots & \vdots \\ -M(n,1) & -M(n,2) & \cdots & \lambda-M(n,n) \end{pmatrix}$

By the definition of the determinant, the coefficient of λ only comes from the term $(\lambda-M(1,1))(\lambda-M(2,2))\ldots(\lambda-M(n,n))$, which is equal to $-\sum_{i=1}^{n} M(i,i)$.

So, $\text{trace}(M) = \sum_{i=1}^{n} M(i,i) = -$ coefficient of λ^{n-1} in $\det(\lambda I - M) = \sum_{i=1}^{n} \lambda_i$. □

Basic Spectral Graph Theory

By the spectral theorem for real symmetric matrix, the adjacency matrix A of an undirected graph has $|V|$ real eigenvalues $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$, with corresponding eigenvectors $x_1, x_2, \ldots, x_n$ which form an orthonormal basis of $\mathbb{R}^n$.

(We will reserve $\lambda_1, \ldots, \lambda_n$ to be the eigenvalues of the Laplacian matrix to be introduced later.)

It is good to compute the eigenvalues of some simple graphs, but we don't have time...

Exercise 1 What is the spectrum of the complete graph with n vertices?

(Ans: $n-1$ of multiplicity 1, -1 of multiplicity $n-1$.)

Exercise 2 What is the spectrum of the complete bipartite graph with m vertices on one side and n vertices on the other side?

(Ans: $\sqrt{nm}$ of multiplicity 1, 0 of multiplicity $n+m-2$, $-\sqrt{nm}$ of multiplicity 1.)

To study the eigenvalues of the adjacency matrix, let's start from the two extremes.

Recall that we assume the graph is d-regular.

It is easy to check that the all-one vector $\vec{1}$ is an eigenvector of A with eigenvalue d, because $(A\vec{1})(i) = \sum_{j \in N(i)} A(i,j) = \sum_{j \in N(i)} 1 = d$ for all $1 \le i \le n$.

Is there a larger eigenvalue than d?

Let x be any eigenvector. Let i be an entry with maximum value, i.e. $x(i) \ge x(j)$ $\forall j$.

Then $(Ax)(i) = \sum_{j \in N(i)} x(j) \le \sum_{j \in N(i)} x(i) = d \cdot x(i)$. $-(*)$

So, any eigenvalue of A is at most d.

What is the multiplicity of d?

Look at $(*)$, if $(Ax)(i) = d \cdot x(i)$, then the inequality must be tight, and thus $x(j) = x(i)$ for all $j \in N(i)$.

Now, if the graph is connected, we can repeat this argument to force every entry of x to be the same.

So, if G is d-regular and connected, then the only eigenvectors with eigenvalue d are the constant vectors, and thus the eigenvalue is of multiplicity one.

On the other hand, if G is disconnected, then $A = \begin{array}{c} \\ X \\ Y \end{array}\begin{array}{c} \begin{array}{cc} X & Y \end{array} \\ \left(\begin{array}{c|c} & 0 \\ \hline 0 & \end{array}\right) \end{array}$, then the vectors $\begin{bmatrix} 1 \\ 0 \end{bmatrix}\}X$ and $\begin{bmatrix} 0 \\ 1 \end{bmatrix}\}Y$ are both eigenvectors with eigenvalue d. Since they are linearly independent, the nullspace$(A-dI)$ is of dimension at least two, and thus d is of multiplicity two.

This is a summary of what we have discussed.

Fact (connectedness) Let G be a d-regular graph and $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n$ be the eigenvalues of its adjacency matrix. Then (1) $\alpha_1 \leq d$.

(2) $\alpha_1 = \alpha_2$ if and only if G is disconnected.

This is the spectral characterization of connectedness. We can say a bit more.

Exercise (number of connected components) Let G be a d-regular graph. Then $\alpha_k = d$ if and only if G has at least k connected components.

Okay, let's consider the other end of the spectrum.

How small can an eigenvalue be? Again we assume the graph is d-regular.

Actually, almost the same argument as above can answer the question.

Let x be an eigenvector. Let $x(i)$ be an entry with maximum absolute value, without loss of generality we assume $x(i)=1$.

Consider $(Ax)(i) = \sum_{j \in N(i)} A(i,j)x(j) = \sum_{j \in N(i)} x(j) \geq \sum_{j \in N(i)} -1 = -d$. (because $|x(j)| \leq 1$)

Therefore, the smallest eigenvalue is at least $-d$.

when $-d$ is an eigenvalue?

Then the inequality must achieve as equality. This means that $x(j) = -1$ for all $j \in N(i)$.

By the same argument, all neighbors of $j \in N(i)$ must have value $+1$, and so on.

If G is connected, then all the neighbors of a vertex i with $x(i)=+1$ must have value -1, and vice versa. This implies that G is a bipartite graph.

Summarizing the above discussion, we have the following fact.

Fact (Bipartiteness) Let G be a d-regular graph.

Then $\lambda_n = -d$ if and only if G has a component which is a bipartite graph.

This is the spectral characterization of bipartiteness.

Before we move on, let me take this opportunity to introduce other commonly used matrices:

- diagonal degree matrix: D, where $D(i,i) = \deg(i)$.
- normalized adjacency matrix: $\mathcal{A} := D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$. All eigenvalues are between -1 and $+1$.
- Laplacian matrix: $L := D - A$.
- normalized Laplacian matrix: $\mathcal{L} := D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - \mathcal{A}$. All eigenvalues are between 0 and 2.

(need to think...)

If G is d-regular, these matrices are essentially the same:

$$\alpha_i \text{ of } A \iff \frac{\alpha_i}{d} \text{ of } \mathcal{A} \iff 1 - \frac{\alpha_i}{d} \text{ of } \mathcal{L} \iff d - \alpha_i \text{ of } L.$$

For general graphs, (normalized) Laplacian matrices are more often used, for the reason explained soon.

Rayleigh Quotient

The main tool in connecting eigenvalues and eigenvectors to optimization problem is the Rayleigh quotient, which is defined as $R(x) := \frac{x^T M x}{x^T x} = \frac{\sum_{i,j} M(i,j)\, x(i)\, x(j)}{\sum_i x^2(i)}$.

Let M be a real symmetric matrix with eigenvalues $\lambda_1 \le \lambda_2 \le \dots \le \lambda_n$ with corresponding orthonormal eigenvectors $x_1, x_2, \dots, x_n$.

Claim $\lambda_1 = \min_x \frac{x^T M x}{x^T x}$.

proof Let $x \in \mathbb{R}^n$. We can write $x = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$, as x_i form a basis.

Then $x^T M x = (c_1 x_1 + \dots + c_n x_n)^T M (c_1 x_1 + \dots + c_n x_n)$

$= (c_1 x_1 + \dots + c_n x_n)^T (c_1 \lambda_1 x_1 + \dots + c_n \lambda_n x_n)$ since x_i is an eigenvector with eigenvalue λ_i

$= c_1^2 \lambda_1 + c_2^2 \lambda_2 + \dots + c_n^2 \lambda_n$ since $\langle x_i, x_j \rangle = 0$ for $i \neq j$ and $\langle x_i, x_i \rangle = 1 \ \forall i$

Similarly, $x^T x = (c_1 x_1 + \dots + c_n x_n)^T (c_1 x_1 + \dots + c_n x_n) = c_1^2 + \dots + c_n^2$.

So, $\frac{x^T M x}{x^T x} = \frac{c_1^2 \lambda_1 + \dots + c_n^2 \lambda_n}{c_1^2 + \dots + c_n^2} \le \frac{\lambda_1 (c_1^2 + \dots + c_n^2)}{c_1^2 + \dots + c_n^2} = \lambda_1$. □

The advantage of this characterization is that it can be extended to other eigenvalues.

Let T_k be the set of vectors that are orthogonal to $x_1, x_2, \dots, x_{k-1}$.

Claim $\lambda_k = \min_{x \in T_k} \frac{x^T M x}{x^T x}$.

proof Let $x \in T_k$. Write $x = c_1 x_1 + \dots + c_n x_n$.

Observe that $c_i = \langle x, x_i \rangle$. Since $x \in T_k$, $c_1 = c_2 = \dots = c_{k-1} = 0$.

Then $\frac{x^T M x}{x^T x} = \frac{\sum_{i=k}^{n} c_i^2 \lambda_k}{\sum_{i=k}^{n} c_i^2} \le \lambda_k$. □

The above result gives a characterization of λ_k, but it requires the knowledge of previous eigenvectors.

The following result is easier to use.

Courant-Fischer Theorem $\lambda_k = \min_{\substack{S \subseteq \mathbb{R}^n \\ \dim(S)=k}} \max_{x \in S} \frac{x^T M x}{x^T x}$

not so hard (see [77])

We skip the proof, but mention that to give an upper bound on λ_k, it is enough to find a k-dimensional subspace S such that $R(x) \leq \lambda_k$ for all $x \in S$.

Laplacian Matrix

We will use the above claim to study the second eigenvalue, i.e. $\lambda_2 = \min_{x \perp x_1} \frac{x^T M x}{x^T x}$.

Before that, let's see why we prefer Laplacian matrices over adjacency matrices.

There are at least two reasons:

① For any graph (not only d-regular graph), the all-one vector $\vec{1}$ is an eigenvector of L with eigenvalue 0 (the smallest eigenvalue of L). ← need to think...

Then, $\lambda_2 = \min_{x \perp \vec{1}} \frac{x^T L x}{x^T x}$. So we know we are looking for minimizer over all vectors x with $\sum x(i) = 0$.

② $x^T L x$ has a very nice quadratic form: $x^T L x = \sum_{ij \in E} (x(i) - x(j))^2$.

There is a good way to see it, although you can directly check it.

Let $e = ij$ be an edge. Let $L_e = \begin{array}{c} \\ i \\ j \end{array} \begin{array}{c} \begin{array}{cc} i & j \end{array} \\ \begin{bmatrix} +1 & -1 \\ -1 & +1 \end{bmatrix} \end{array}$ be the Laplacian of an edge.

Then $L = \sum_{e \in E} L_e$. Then $x^T L x = x^T \left(\sum_{e \in E} L_e \right) x = \sum_{e \in E} x^T L_e x = \sum_{ij \in E} (x(i) - x(j))^2$.

Cheeger's Inequality

Let $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2$ be the eigenvalues of the normalized Laplacian matrix $\mathcal{L}$.

Recall that in d-regular graphs, $\lambda_1 = 1 - \frac{\alpha_1}{d}$ and $\lambda_2 = 1 - \frac{\alpha_2}{d}$.

By the fact of connectedness, $\lambda_1 = \lambda_2 = 0$ if the graph is disconnected.

By the fact of connectedness, $\lambda_1 = \lambda_2 = 0$ if the graph is disconnected.

This spectral characterization may not look so useful, as it is so easy to check whether a graph is disconnected (e.g. BFS, DFS).

The real power of this spectral characterization is that it can be generalized "continuously":

- λ_2 is small $\Rightarrow$ the graph is closed to being disconnected, i.e. there is a sparse cut
- λ_2 is large $\Rightarrow$ the graph is far from being disconnected, i.e. there is no sparse cut (or the graph is an "expander").

Cheeger's inequality: $\frac{1}{2}\lambda_2 \le \phi(G) \le \sqrt{2\lambda_2}$, where λ_2 is the second eigenvalue of $\mathcal{L}$.

The first inequality is called the "easy" direction, and the second inequality is called the "hard" direction.

So, naturally we prove the easy direction first.

Recall $\lambda_2 = \min_{x \perp \vec{1}} \frac{x^T \mathcal{L} x}{x^T x} = \min_{x \perp \vec{1}} \frac{x^T L x}{d x^T x} = \min_{x \perp \vec{1}} \frac{\sum_{ij \in E} (x(i) - x(j))^2}{d \sum_{i \in V} x(i)^2}$.

To upper bound λ_2, we just need to find a vector $x \perp \vec{1}$ and compute its Rayleigh quotient.

To get some intuition, let say $\phi(G) = \phi(S)$ and $|S| = n/2$.

We consider the "binary" solution: $x(i) = \begin{cases} +1 & \text{if } i \in S \\ -1 & \text{if } i \notin S \end{cases}$.

Since $|S| = n/2$, $\sum_{i \in V} x(i) = 0$, and thus $x \perp \vec{1}$.

Then $\lambda_2 \le \frac{\sum_{ij \in E} (x(i) - x(j))^2}{d \sum_{i \in V} x(i)^2} = \frac{4 |\delta(S)|}{d|V|} = \frac{2|\delta(S)|}{d|S|} = 2\phi(S)$.

For general S, we consider the binary solution: $x(i) = \begin{cases} +\frac{1}{|S|} & \text{if } i \in S \\ -\frac{1}{|V-S|} & \text{if } i \notin S \end{cases}$.

Then $x \perp \vec{1}$, and $\lambda_2 \le \frac{\sum_{ij \in E} (x(i) - x(j))^2}{d \sum_{i \in V} x(i)^2} = \frac{|\delta(S)| \cdot \left(\frac{1}{|S|} + \frac{1}{|V-S|}\right)^2}{d\left(|S| \cdot \frac{1}{|S|^2} + |V-S| \cdot \frac{1}{|V-S|^2}\right)} = \frac{|\delta(S)| \cdot |V|}{d \cdot |S| \cdot |V-S|} \le 2\phi(S)$.

This proves the easy direction.

This proves the easy direction.

To summarize, if there is a sparse cut, then λ_2 is small.

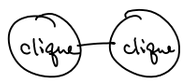
A consequence is that if λ_2 is large, then we know that G has no sparse cut.

The Hard Direction : Intuition

In the minimization problem $\min_{x \perp \vec{1}} \frac{\sum_{ij \in E} (x(i)-x(j))^2}{d \sum_{i \in V} x(i)^2}$, if we can only search for "binary" solutions,

then we are essentially optimizing over the conductances.

Unfortunately, we are optimizing over a much larger domain (otherwise the problem is not efficiently solvable), and there could be some very non-binary solutions (very "smooth" vector), for which it is not clear how to find a sparse cut from it.

To get some feeling, suppose we are given a graph like 

.

Observe that the optimizer tries to minimize the average $(x(i)-x(j))^2 / (x(i)^2 + x(j)^2)$.

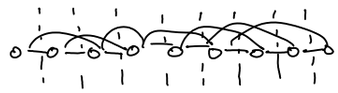
In this case, it is not good to "split" the vertices in a clique, because there are so many edges within it. So, we would expect that the values in each clique are very similar, while the two cliques would have different values so that $x \perp \vec{1}$. Hence, we expect that the minimizer would look very similar to a binary vector, and we can easily a good cut with $\phi(S) \approx \lambda_2$.

Now, suppose we are given a graph like  , then the minimizer can do much better by making each edge very short, while the values decrease smoothly from +1 to -1, in which case $\lambda_2 << \phi(G)$.

The key of Cheeger's inequality is to show that λ_2 cannot be much smaller than $\phi(G)$.

In other words, if λ_2 is small, then we can extract a somewhat sparse cut from the eigenvector.

We can think of the optimizer "embeds" the graph into a line, while most edges are short.



Then it should be the case that some threshold gives a sparse cut.

The Hard Direction : Proof

The first step is to preprocess the second eigenvector so that at most half the entries are nonzero.

This would guarantee that the output set S satisfies $|S| \le |V|/2$.

This step is simple. Without loss of generality we assume there are fewer positive entries in x than negative entries.

Consider the following vector y: $y(i) = \begin{cases} x(i) & \text{if } x(i) \ge 0 \\ 0 & \text{if } x(i) < 0 \end{cases}$.

Claim $R(y) \le R(x)$.

proof $(\mathcal{L}y)(i) = y(i) - \sum_{j \in N(i)} \frac{y(j)}{d} \le x(i) - \sum_{j \in N(i)} \frac{x(j)}{d} = (\mathcal{L}x)(i) = \lambda_2 \cdot x(i) \quad \forall\, i$ with $y(i) > 0$.

Therefore, $y^T \mathcal{L} y = \sum_{i \in V} y(i) \cdot (\mathcal{L}y)(i) \le \sum_{i : y(i) > 0} \lambda_2 x(i)^2 = \sum_i \lambda_2 y(i)^2$, proving the claim. $\square$

There is a very elegant argument to make the above intuition precise: just pick a random threshold.

Lemma Given any y, there exists a subset $S \subseteq \text{supp}(y)$ such that $\phi(S) \le \sqrt{2R(y)}$, where

$$\text{supp}(y) = \{ i \mid y(i) \neq 0 \}.$$

proof We can assume that $-1 \le y_i \le +1$ for all i, by scaling y if necessary.

Let $t \in (0,1]$ be chosen uniformly at random.

Let $S_t = \{ i \mid y_i^2 \geq t \}$. Then $S_t \subseteq \text{supp}(y)$ by construction.

We analyze the expected value of $|\delta(S_t)|$ and the expected value of $|S_t|$.

$$E(|\delta(S_t)|) = \sum_{ij \in E} \left[\Pr(\text{the edge } ij \text{ is cut}) \right] \quad \text{by linearity of expectation}$$

$$= \sum_{ij \in E} \left[\Pr\left(y_{(i)}^2 < t \leq y_{(j)}^2 \right) \right]$$

$$= \sum_{ij \in E} |y_{(j)}^2 - y_{(i)}^2|$$

$$= \sum_{ij \in E} |y_{(i)} - y_{(j)}| \, |y_{(i)} + y_{(j)}|$$

$$\leq \sqrt{\sum_{ij \in E} (y_{(i)} - y_{(j)})^2} \sqrt{\sum_{ij \in E} (y_{(i)} + y_{(j)})^2} \quad \text{by Cauchy-Schwarz } \langle a,b \rangle \leq \|a\| \cdot \|b\|$$

$$\leq \sqrt{\sum_{ij \in E} (y_{(i)} - y_{(j)})^2} \sqrt{2 \sum_{ij \in E} (y_{(i)}^2 + y_{(j)}^2)}$$

$$= \sqrt{\sum_{ij \in E} (y_{(i)} - y_{(j)})^2} \sqrt{2d \sum_{i \in V} y_{(i)}^2}$$

$$= \sqrt{2R(y)} \cdot \left(d \sum_{i \in V} y_{(i)}^2 \right).$$

$$E[|S_t|] = \sum_{i \in V} \Pr[\, y_{(i)}^2 \geq t \,] = \sum_{i \in V} y_{(i)}^2$$

Therefore, $\dfrac{E[|\delta(S_t)|]}{E[d|S_t|]} \leq \sqrt{2R(y)}$.

This means that $E[\, |\delta(S_t)| - \sqrt{2R(y)} \cdot d \cdot |S_t| \,] \leq 0$.

Hence, there exists t such that $\dfrac{|\delta(S_t)|}{d \cdot |S_t|} \leq \sqrt{2R(y)}$. □

Combining the claim and the lemma proves Cheeger's inequality!

And the proof shows that the spectral partitioning algorithm achieves the performance guarantee, because the output set S_t is a "threshold" set that the algorithm searches.

Discussions

① For general weighted graphs, the same result holds (using $\mathcal{L} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ and $\phi(S) = \frac{|\delta(S)|}{\sum_{v \in S} \deg(v)}$), and the proof is almost the same.

② Cheeger's inequality is tight on both sides, i.e. there are graphs for which the inequality achieves as equality. You can check that if G is a cycle of n vertices, then $\phi(G) = \frac{2}{n}$ but $\lambda_2 = O(\frac{1}{n^2})$.

③ Cheeger's inequality gives an $O(\frac{1}{\sqrt{\lambda_2}})$-approximation algorithm for computing $\phi(G)$. When λ_2 is large, then it is a pretty good approximation. But λ_2 could be as small as $O(\frac{1}{n^2})$, and so it could be an $\Omega(n)$-approximation. It doesn't quite explain the good performance in practice. We will come back to this question later.

④ The second eigenvalue is very closely related to the convergence rate of a random walk: the convergence rate is fast if and only if λ_2 is bounded away from λ_1.

In many applications, we would like to show that the convergence rate is fast, and then we would have an efficient algorithm to do a uniform sampling by the Markov chain Monte Carlo method. It is very difficult to lower bound λ_2 directly, as those graphs coming from the sampling problems are complicated. It turns out that Cheeger's inequality provides a way to do it, by lower bounding $\phi(G)$. It is also not so easy to lower bound $\phi(G)$, but it turns out that it is more manageable, and many analyses of convergence rate of random walk are based on Cheeger's inequality.

⑤ Another important applications of Cheeger's inequality is to construct expander graphs, which are graphs with linear number of edges but have no sparse cut. These graphs are useful in almost every branch of theoretical computer science, from proving lower bound, to derandomization, to designing algorithms, see [8]. In this scenario, we have the freedom to construct the graph and we want to lower bound $\phi(G)$. It turns out in this situation it is very difficult to directly lower bound $\phi(G)$, but to lower bound λ_2 and use Cheeger's inequality. Again, this is still very difficult to lower bound λ_2, but since the graph can be chosen by us, there are some sophisticated mathematical constructions for which λ_2 can be bounded by some deep mathematics.

Last Eigenvalue

Now we survey the recent developments relating other eigenvalues to graph partitioning problems. In the following we will just state the results and discuss some main ideas.

Recall that Cheeger's inequality is a robust generalization of the basic spectral characterization of connectedness. Can we generalize the spectral characterization of bipartiteness? It turns out that the proof in Cheeger's inequality can be adapted in this setting.

We define $$\beta(G) = \min_{y \in \{-1,0,+1\}^V} \frac{\sum_{ij \in E} |y(i)+y(j)|}{d \sum_{i \in V} |y(i)|} = \min_{S \subseteq V} \frac{2|\#\text{ edges within } L| + 2|\#\text{ edges within } R| + |\delta(S)|}{d|S|}$$ $$(L,R) \text{ partition of } S$$

This is called the bipartiteness ratio of G. Note that $\beta(G)$ is small if and only if G contains a subset $S \subseteq V$ which is close to a bipartite component, with most edges in S

crossing L and R. 

(edges within are counted twice and edges going out S are counted once)

Note that we still assume the graph is d-regular for simplicity.

Consider the matrix $I + \mathcal{A} = I + \frac{1}{d}A$. Let $2 = \alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_n \geq 0$ be its eigenvalues.

Recall that $\alpha_n = 0$ if and only if G has a bipartite component.

Trevisan [2] proved the following generalization.

Theorem $\frac{1}{2}\alpha_n \leq \beta(G) \leq \sqrt{2\alpha_n}$

The proof is very similar to the proof of Cheeger's inequality ("randomized rounding"), and it is left in the problem set.

Applying this theorem recursively, Trevisan obtained an approximation algorithm for the maximum cut problem with worst case approximation ratio 0.531.

Note that getting a 0.5-approximation is trivial, and there is a 0.878-approximation algorithm by semidefinite programming. The spectral algorithm is the only known alternative method to do better than 0.5.

The k-th Eigenvalue

Let $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2$ be the eigenvalues of the normalized Laplacian matrix.

Recall the spectral characterization that $\lambda_k = 0$ if and only if G has k connected components.

It turns out that there are two meaningful ways to generalize this basic fact.

① Small sparse cut: If λ_k is small, then there is a sparse cut S with $|S| \approx |V|/k$.

② Many sparse cuts: If λ_k is small, then there are k disjoint sparse cuts.

It may appear that ② is more general than ①, but the results obtained are incomparable as we will explain soon.

The informal intuition is that each eigenvector defines a sparse cut, and since the eigenvectors are orthogonal, the sparse cuts should look quite different, and thus cut the graph into pieces. Proving this intuition formally is another story.

Small Sparse Cut

Arora, Barak, Steurer [3] proved:

Theorem For $k = \Omega(n^{\varepsilon})$ for $\varepsilon \in (0,1)$, there is a set S with $\phi(S) = O(\sqrt{\lambda_k})$ and $|S| \approx n/k$.

The proof has some nice new ideas. Consider the matrix $W := \frac{1}{2}I + \frac{1}{2}A = \frac{1}{2}I + \frac{1}{2d}A$ (d-regular).

Let $1 \geq \alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_n \geq 0$ be the eigenvalues of this "lazy random walk matrix".

So the condition that $\lambda_k \approx 0$ is translated to $\alpha_k \approx 1$.

Consider $\mathrm{Tr}(W^k)$.

Recall that $\mathrm{Tr}(W^k)$ = sum of eigenvalues of $W^k = \sum_i \lambda_i^k$ = "large" by our assumption.

On the other hand, $\mathrm{Tr}(W^k)$ = sum of diagonal entries of W^k = sum of "returning probabilities" after k steps of random walk.

If every small set has large conductance, then we expect that $\mathrm{Tr}(W^k)$ is small, contradicting to our assumption. So there must exist a small sparse cut.

Many Sparse Cuts

Two groups [4a, 4b] proved essentially the same results.

Theorem Let $\phi_k(G) = \min_{S_1,\dots,S_k} \max_i \phi(S_i)$. Then $\frac{1}{2}\lambda_k \leq \phi_k(G) \leq O(k^2)\sqrt{\lambda_k}$.

disjoint

The first inequality is the easy direction, and is left to you in the problem set.

Both use the spectral embedding: Let $x_1, \ldots, x_k$ be the first k eigenvectors. Map each vertex i to a k-dimensional point $(x_1(i), x_2(i), \ldots, x_k(i))$.

Since $x_1, \ldots, x_k$ are orthonormal, it can be proved that the points are "well-spread out".

The algorithm in [4b] is very simple: just pick k random directions, each point is put to the cluster defined by its closest diraction.

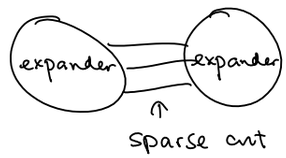
Improved Cheeger's Inequality

Finally, I would like to mention our new result on analysis of spectral partitioning through higher eigenvalues.

Let $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n \leq 2$ be the eigenvalues. Suppose λ_2 is small.

If λ_3 is small, then we know that there will be three disjoint sparse cuts.

What if λ_3 is large? Then we know that there is a good way to cut the graph into two pieces, but there is no good way to cut the graph into three pieces.

Then we expect the graph to look like 

, and the second eigenvector should look like a "binary" vector,

and thus $\lambda_2 \approx \phi(G)$, and thus λ_2 is a better approximation to $\phi(G)$ when λ_3 is large.

In general, we prove:

Theorem [5] $\quad \phi(G) \leq O(k) \dfrac{\lambda_2}{\sqrt{\lambda_k}}$.

The proof is to show that when λ_k is large, then the second eigenvector looks like a k-step function,

The proof is to show that when λ_k is large, then the second eigenvector looks like a k-step function, and thus λ_2 cannot be much smaller than $\phi(G)$, i.e. Cheeger's inequality can't be tight.

It shows that spectral partitioning is a constant factor approximation when λ_k is large for a small k.

In image segmentation and clustering, there are usually only a few outstanding sparse cuts, thus λ_k is large. Thus, the result shows that spectral partitioning actually gives a good approximation in those instances. It gives some theoretical justification of the empirical performance of spectral partitioning. The result can be generalized to the multiway partitioning problem.

Concluding Remarks

It is surprising that many results in spectral graph theory are only discovered recently.

It is an active research area, and I haven't mentioned some recent results, most notably the eigenspace enumeration approach [3] and analyzing semidefinite programming by higher eigenvalues.

Also, eigenvalues are used to design faster algorithms for solving linear equations.

It seems that much more can be done in this direction, hence the title.

For reference I recommend the course notes by Daniel Spielman [9].

References

[1] Shi, Malik. Normalized Cuts and Image Segmentation, 2000.

[2] Trevisan. Max cut and the smallest eigenvalue. 2009.

[3] Arora, Barak, Steurer. Subexponential algorithms for unique games and related problems. 2010.

[4a] Lee, Oveis Gharan, Trevisan. Multi-way spectral partitioning and higher-order Cheeger inequalities. 2012.

[4b] Louis, Raghavendra, Tetali, Vempala. Many sparse cuts via higher eigenvalues. 2012

[5] Kwok, Lau, Lee, Oveis Gharan, Trevisan. Improved Cheeger's inequality: analysis of spectral partitioning algorithms through higher order spectral gap. 2013.

[6] http://www.cse.cuhk.edu.hk/~chi/csc5160/notes/L01.pdf

[7] http://www.cse.cuhk.edu.hk/~chi/csc5160/notes/L02.pdf

[8] Hoory, Linial, Wigderson. Expander graphs and their applications. 2006.

[9] Spielman. Spectral graph theory (course notes), 2012.