
University of Michigan–Ann Arbor

Department of Electrical Engineering and Computer Science
EECS 498 004 Advanced Graph Algorithms, Fall 2021

Lecture 8: λ_n and The Power Method

September 22, 2021

Instructor: Thatchaphol Saranurak

Scribe: Peter Ly

We will mainly talk about two things for this lecture:

1. $\lambda_n(\mathbf{N})$ for finding max cuts (similar to $\lambda_2(\mathbf{N})$ for finding sparse cuts)
2. The power method: a fast algorithm for finding approximate eigenvectors.

1 $\lambda_n(\mathbf{N})$ vs Max Cut

We saw from Cheeger's Inequality that $\lambda_2(\mathbf{N}_G)$ corresponds to sparsest cut of the graph G . That is, $\lambda_2(\mathbf{N}_G)$ is small if and only if G has a sparse cut. In the extreme case where $\lambda_2(\mathbf{N}_G) = 0$, then we have that G is not connected.

There is an analogous result for $\lambda_n(\mathbf{N}_G)$ and the size of the maximum cut of G . $\lambda_n(\mathbf{N}_G)$ is large if and only if the size of the maximum cut is large - in other words, the graph is close to being bipartite. In the extreme case where $\lambda_n(\mathbf{N}_G) = 2$, then we have that G contains a connected component which is bipartite. We will show the proof of the extreme case and mention what happens if $\lambda_n(\mathbf{N}_G)$ is close to 2.

Lemma 1.1. *We have*

1. $\lambda_n(\mathbf{N}_G) \leq 2$. So all eigenvalues of \mathbf{N}_G are in the range of $[0, 2]$.
2. $\lambda_n(\mathbf{N}_G) = 2$ if and only if G contains a connected component which is bipartite.

Proof. Recall

$$\lambda_n(\mathbf{N}_G) = \max_{\mathbf{x} \neq 0} \frac{\mathbf{x}^\top \mathbf{N}_G \mathbf{x}}{\mathbf{x}^\top \mathbf{x}} = \max_{\mathbf{x} \neq 0} \frac{\mathbf{x}^\top \mathbf{L}_G \mathbf{x}}{\mathbf{x}^\top \mathbf{D} \mathbf{x}}$$

1. To prove $\lambda_n(\mathbf{N}_G) \leq 2$, observe that

$$2\mathbf{x}^\top \mathbf{D}\mathbf{x} - \mathbf{x}^\top \mathbf{L}_G \mathbf{x} = \sum_u 2\deg(u)x_u^2 - \sum_{uv \in E} w_{uv}(x_u - x_v)^2 = \sum_{uv \in E} w_{uv}(x_u + x_v)^2$$

So

$$2 - \lambda_n(\mathbf{N}_G) = 2 - \max_{\mathbf{x} \neq 0} \frac{\mathbf{x}^\top \mathbf{L}_G \mathbf{x}}{\mathbf{x}^\top \mathbf{D} \mathbf{x}} = \min_{\mathbf{x} \neq 0} \frac{2\mathbf{x}^\top \mathbf{D}\mathbf{x} - \mathbf{x}^\top \mathbf{L}_G \mathbf{x}}{\mathbf{x}^\top \mathbf{D} \mathbf{x}} = \min_{\mathbf{x} \neq 0} \frac{\sum_{uv \in E} w_{uv}(x_u + x_v)^2}{\mathbf{x}^\top \mathbf{D} \mathbf{x}} \geq 0.$$

2. (\Leftarrow) Now, suppose there is a bipartite component C where $V(C) = L \cup R$ (see Figure 1 for the image to keep in mind). Define a vector \mathbf{x} where

$$x_u = \begin{cases} 1 & u \in L \\ -1 & u \in R \\ 0 & u \notin V(C) \end{cases}$$

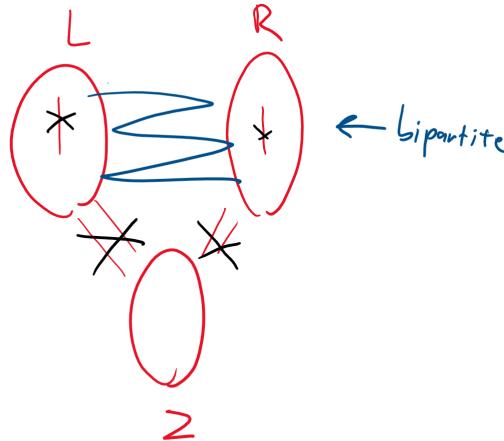
We have (from the proof of (1) in the lemma) that

$$2 - \lambda_n(\mathbf{N}_G) = \min_{\mathbf{x} \neq 0} \frac{\sum_{uv \in E} w_{uv}(x_u + x_v)^2}{\mathbf{x}^\top \mathbf{D} \mathbf{x}} = 0$$

So $\lambda_n(\mathbf{N}_G) = 2$.

(\Rightarrow) Suppose $\lambda_n(\mathbf{N}_G) = 2$. Then there is some non-zero \mathbf{x} such that $\sum_{uv \in E} w_{uv}(x_u + x_v)^2 = 0$. This happens if and only if $x_u = -x_v$ for all edges $(u, v) \in E$. Define $L = \{u \mid x_u < 0\}$, $R = \{u \mid x_u > 0\}$, and $Z = \{u \mid x_u = 0\}$ (see Figure 1 for an example of how the components relate). Since $\mathbf{x} \neq 0$, we have that $L \cup R$ is non-empty. Further, there are no edges between $L \cup R$ and Z . It follows that $L \cup R$ is a bipartite component of G .

Figure 1: $L \cup R$ form a bipartite component of G . There are no edges between $L \cup R$ and Z . Z may have edges inside of it, but those edges contribute 0 to the sum.



□

When we have that G is connected, then the only component is G itself, and Lemma 1.1 tells us that $\lambda_n(\mathbf{N}_G) = 2$ if and only if G is bipartite i.e. there is a cut that cuts all edges.

If we instead have that $\lambda_n(\mathbf{N}_G)$ is close to 2, then we can show that G is “close to being bipartite” - there is a cut which cuts almost every edge of G . This result was shown by Trevisan - more precisely, Trevisan showed a Cheeger’s type inequality for λ_n :

$$\frac{2 - \lambda_n}{2} \leq \beta(G) \leq \sqrt{2(2 - \lambda_n)}$$

where $\beta(G)$ measures bipartiteness of G (see [his lecture notes¹](#)).

¹<https://lucatrevisan.github.io/teaching/expanders2016/lecture05.pdf>

2 The Power Method

Last time, we showed that we can construct a sparse cut S of G satisfying $\Phi_G(S) \leq \sqrt{2\lambda_2(\mathbf{N}_G)} \leq 2\sqrt{\Phi(G)}$ by using Fiedler's algorithm. The only slow step for constructing such an S was computing the second eigenvector \mathbf{v}_2 of \mathbf{N}_G . Now, we will show how to compute an "approximate second smallest eigenvector" in near-linear time. We can then use the approximate second smallest eigenvector in Fiedler's algorithm to obtain a sparse cut satisfying $\Phi_G(S) \leq \sqrt{2(\lambda_2(\mathbf{N}_G) + \epsilon)}$.

We will begin by presenting how to find an approximation to the largest eigenvector and second largest eigenvector, and then we will show how to use the techniques for computing the largest and second largest eigenvector to compute an approximate second smallest eigenvector. The technique we will use to compute these approximations is called the Power Method.

2.1 Largest Eigenvalue of \mathbf{M}

This algorithm gives an approximation to the largest eigenvalue (and, in fact, its corresponding eigenvector).

POWER

Input: a PSD matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ and parameter k

1. Sample uniformly $\mathbf{x} \in \{-1, 1\}^n$
2. Return $\mathbf{y} = \mathbf{M}^k \mathbf{x}$.

POWER has runtime $O(k(n + \text{nnz}(\mathbf{M}))$ where $\text{nnz}(\mathbf{M})$ is the number of non-zero entries of \mathbf{M} .

Let $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ be eigenvalues of \mathbf{M} (we have that all eigenvalues are non-negative because \mathbf{M} is PSD).

Theorem 2.1 (Performance Guarantee of POWER). *For any $\epsilon > 0$, with probability at least $3/16$, the algorithm POWER returns \mathbf{y} such that*

$$\frac{\mathbf{y}^\top \mathbf{M} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \geq (1 - \epsilon)\lambda_1 \cdot \frac{1}{1 + 4n(1 - \epsilon)^{2k}}.$$

If we take $k = O(\log(n)/\epsilon)$, then we have that $\frac{\mathbf{y}^\top \mathbf{M} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \geq (1 - O(\epsilon))\lambda_1$ - that is we have a multiplicative approximation for the first eigenvector with constant probability. We can amplify this probability to at least $1 - \frac{1}{\text{poly}(n)}$ by repeating POWER independently $O(\log n)$ times and taking the vector \mathbf{y}^* which maximizes $\frac{\mathbf{y}^\top \mathbf{M} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}}$.

Analysis of the POWER algorithm. Let $\mathbf{v}_1, \dots, \mathbf{v}_n$ be eigenvectors of \mathbf{M} . We can write

$$\mathbf{M} = \sum_{i=1}^n \lambda_i \mathbf{v}_i \mathbf{v}_i^\top \text{ and } \mathbf{M}^k = \sum_{i=1}^n \lambda_i^k \mathbf{v}_i \mathbf{v}_i^\top$$

by spectral decomposition. We can then write

$$\mathbf{x} = \sum_i a_i \mathbf{v}_i$$

where $a_i = \langle \mathbf{x}, \mathbf{v}_i \rangle$ i.e. express \mathbf{x} in the vector space spanned by the eigenvectors of \mathbf{M} . We can then write

$$\mathbf{y} = \mathbf{M}^k \mathbf{x} = \sum_i a_i \lambda_i^k \mathbf{v}_i.$$

Let us first see on a high level why the power method works.

Intuition. Suppose $\lambda_1 = 1$ and all $\lambda_2, \dots, \lambda_n \leq 1/2$. Then we can write

$$\mathbf{y} = a_1 \mathbf{v}_1 + \sum_{i \geq 2} a_i \lambda_i^k \mathbf{v}_i.$$

When $k = \Theta(\log n)$, then each $\lambda_i^k \leq \frac{1}{\text{poly}(n)}$ and the second term becomes vanishingly small. Note that here we need \mathbf{M} to be PSD to guarantee that each λ_i^k was non-negative so that each term with λ_i^k for $i \geq 2$ would actually go to zero - otherwise we might have a negative eigenvalue < -1 and some terms of the sum might not go to zero. Now suppose we can show that $|a_1| = |\langle \mathbf{x}, \mathbf{v}_1 \rangle|$ is not very small, then we get $\mathbf{y} \approx \pm \mathbf{v}_1$, so $\frac{\mathbf{y}^\top \mathbf{M} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \approx \lambda_1$.

Making it formal. There were two points we were informal about in our intuition:

1. Why should $|a_1| = |\langle \mathbf{x}, \mathbf{v}_1 \rangle|$ be not too small?

This is very natural to expect - essentially we are projecting \mathbf{v}_1 in a random direction - the magnitude of this projection is likely to be comparable to \mathbf{v}_1 . We prove this formally in the exercise session. To be exact, we show

Lemma 2.2. *For any vector $\mathbf{v} \in \mathbb{R}^n$ where $\|\mathbf{v}\| = 1$, sample $\mathbf{x} \in \{-1, 1\}^n$. Then*

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

In particular, with good probability $|a_1|$ is not too small.

2. There is no reason for there to be a large gap between λ_1 and $\lambda_2, \dots, \lambda_n$. What if λ_2 is close to λ_1 ? We will show that even if we do not have such a gap, we still obtain a large value for $\frac{\mathbf{y}^\top \mathbf{M} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}}$.

Fix $\varepsilon > 0$, and define ℓ so that λ_i can be classified as follows: $\lambda_1, \dots, \lambda_\ell \geq (1 - \varepsilon)\lambda_1$ and $(1 - \varepsilon)\lambda_1 > \lambda_{\ell+1}, \dots, \lambda_n$.

We have that $\mathbf{y} = \sum_i a_i \lambda_i^k \mathbf{v}_i$ (shown above) and

$$\mathbf{y}^\top \mathbf{y} = \sum_i a_i^2 \lambda_i^{2k} \text{ and } \mathbf{y}^\top \mathbf{M} \mathbf{y} = \sum_i a_i^2 \lambda_i^{2k+1}$$

from the orthogonality of the eigenvectors from the spectral decomposition of \mathbf{M} .

Our goal is to show that $\frac{\mathbf{y}^\top \mathbf{M} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}}$ is big i.e. show a lower bound on the numerator and an upper bound on the denominator.

By truncating terms in $\mathbf{y}^\top \mathbf{M} \mathbf{y}$ and then applying the bound $\lambda_\ell \geq (1 - \varepsilon)\lambda_1$, we obtain the following lower bound on $\mathbf{y}^\top \mathbf{M} \mathbf{y}$:

$$\mathbf{y}^\top \mathbf{M} \mathbf{y} \geq \sum_{i=1}^{\ell} a_i^2 \lambda_i^{2k+1} \geq (1 - \varepsilon)\lambda_1 \cdot \sum_{i=1}^{\ell} a_i^2 \lambda_i^{2k}.$$

Recall that

$$\mathbf{y}^\top \mathbf{y} = \sum_{i=1}^{\ell} a_i^2 \lambda_i^{2k} + \sum_{i=\ell+1}^n a_i^2 \lambda_i^{2k}$$

We can upper bound the $\sum_{i=\ell+1}^n a_i^2 \lambda_i^{2k}$ term (and therefore $\mathbf{y}^\top \mathbf{y}$) as follows:

$$\begin{aligned} \sum_{i=\ell+1}^n a_i^2 \lambda_i^{2k} &< (1-\epsilon)^{2k} \lambda_1^{2k} \sum_{i=\ell+1}^n a_i^2 && (\ell \text{ defined so } (1-\epsilon)\lambda_1 > \lambda_{\ell+1}, \dots, \lambda_n) \\ &\leq (1-\epsilon)^{2k} \lambda_1^{2k} \|\mathbf{x}\|^2 \\ &\leq (1-\epsilon)^{2k} a_1^2 \lambda_1^{2k} \frac{\|\mathbf{x}\|^2}{a_1^2} \\ &\leq (1-\epsilon)^{2k} \sum_{i=1}^{\ell} a_i^2 \lambda_i^{2k} + 4n && (\text{using } a_1^2 = |\langle \mathbf{x}, \mathbf{v}_1 \rangle|^2 \geq \frac{1}{4}) \end{aligned}$$

So we can upper bound $\mathbf{y}^\top \mathbf{y}$ by

$$\mathbf{y}^\top \mathbf{y} \leq \sum_{i=1}^{\ell} a_i^2 \lambda_i^{2k} \cdot (1 + 4n(1-\epsilon)^{2k})$$

Putting the bounds together, we have with probability at least 3/16

$$\frac{\mathbf{y}^\top \mathbf{M} \mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \geq (1-\epsilon)\lambda_1 \cdot \frac{1}{1 + 4n(1-\epsilon)^{2k}}$$

2.2 Second Largest Eigenvalue of \mathbf{M}

What if we want the second largest eigenvalue of \mathbf{M} ? Just run the same algorithm after translation so that the vector we pick is in the space orthogonal to the first eigenvector \mathbf{v}_1 .

POWER2

Input: a PSD matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$, vector \mathbf{v}_1 , and parameter k

1. Sample uniformly $\mathbf{x}' \in \{-1, 1\}^n$
2. $\mathbf{x} \leftarrow \mathbf{x}' - \mathbf{v}_1 \langle \mathbf{x}', \mathbf{v}_1 \rangle$ (and so $\mathbf{x} \perp \mathbf{v}_1$)
3. Return $\mathbf{y} = \mathbf{M}^k \mathbf{x}$.

We can write

$$\mathbf{x} = \sum_{i=2}^n a_i \mathbf{v}_i$$

where $a_i = \langle \mathbf{x}, \mathbf{v}_i \rangle$ as above. We only need to sum for $i = 2$ to n because $\mathbf{x} \perp \mathbf{v}_1$ by step 2 of POWER2. We also have that $a_2 \geq 1/2$ with probability at least 3/16. Performing the same analysis as above for

$$\mathbf{y} = \sum_{i=2}^n a_i \lambda_i^k \mathbf{v}_i$$

and we have

$$\frac{\mathbf{y}^\top \mathbf{M}\mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \geq (1 - \epsilon)\lambda_2 \cdot \frac{1}{1 + 4n(1 - \epsilon)^{2k}}$$

with probability at least 3/16.

Theorem 2.3. Suppose that \mathbf{v}_1 is the first eigenvector of \mathbf{M} . For any $\epsilon > 0$, with probability $\geq 3/16$, the algorithm POWER2 returns $\mathbf{y} \perp \mathbf{v}_1$ such that

$$\frac{\mathbf{y}^\top \mathbf{M}\mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \geq (1 - \epsilon)\lambda_2 \cdot \frac{1}{1 + 4n(1 - \epsilon)^{2k}}.$$

2.3 Second Smallest Eigenvalue of \mathbf{N}_G

Now that we have an algorithm for the second largest eigenvalue of a PSD matrix, we can show the algorithm to get the second smallest eigenvalue of \mathbf{N}_G to use it for Cheeger's cut. Recall that all eigenvalues of \mathbf{N}_G are in $[0, 2]$. If we instead work with the matrix

$$\mathbf{M} = 2\mathbf{I} - \mathbf{N}_G = \mathbf{I} + \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}.$$

we can get the second smallest eigenvalue of \mathbf{N}_G by finding the second largest eigenvalue of \mathbf{M} . If $0 = \lambda_1 \leq \dots \leq \lambda_n \leq 2$ are eigenvalues of \mathbf{N}_G , then the eigenvalues of \mathbf{M} are

$$2 = 2 - \lambda_1 \geq 2 - \lambda_2 \geq \dots \geq 2 - \lambda_n \geq 0$$

and so \mathbf{M} is PSD and we can apply POWER2 to \mathbf{M} .

To apply POWER2, we need the eigenvector \mathbf{v}_1 corresponding to the largest eigenvalue $2 - \lambda_1$ of \mathbf{M} . The eigenvector is the same as λ_1 of \mathbf{N}_G which is $\mathbf{v}_1 = \mathbf{d}^{1/2}$. By running POWER2 on \mathbf{M} with \mathbf{v}_1 , we get $\mathbf{y} \perp \mathbf{d}^{1/2}$ where $\mathbf{y}^\top \mathbf{M}\mathbf{y} \geq (2 - \lambda_2 - \epsilon) \cdot \mathbf{y}^\top \mathbf{y}$.

However

$$\begin{aligned} \mathbf{y}^\top (2\mathbf{I} - \mathbf{N}_G)\mathbf{y} &\geq (2 - \lambda_2 - \epsilon) \cdot \mathbf{y}^\top \mathbf{y} \\ \iff (\lambda_2 + \epsilon)\mathbf{y}^\top \mathbf{y} &\geq \mathbf{y}^\top \mathbf{N}_G \mathbf{y} \end{aligned}$$

so we have that

$$\frac{\mathbf{y}^\top \mathbf{N}_G \mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \leq (\lambda_2 + \epsilon)$$

in time $\tilde{O}(m/\epsilon)$.

Given \mathbf{y} , Fiedler's algorithm computes a cut S where $\Phi_G(S) \leq \sqrt{2(\lambda_2 + \epsilon)}$ in time $O(m+n \log n)$.

However, using $\mathbf{M} = 2(\mathbf{I} - \mathbf{N}_G)$ gives an *additive* approximation, rather than a multiplicative approximation. If we want a $(1 + \epsilon')$ multiplicative approximation, then we need to choose $\epsilon = \epsilon' \lambda_2$. Choosing such ϵ gives a time complexity of $\tilde{O}(m/(\epsilon' \lambda_2)) \leq \tilde{O}(m/(\epsilon' \Phi(G)^2))$ because $\Phi(G) \leq \sqrt{2\lambda_2}$ - when $\Phi(G)$ is small, obtaining a $(1 + \epsilon')$ multiplicative approximation is slow.

Question 2.4 (Open Problem). Given a graph G with a promise that either $\Phi(G) \geq 0.1$ or $\Phi(G) \leq 1/\log^{100} n$, can we decide which case it is in $\tilde{O}(m)$ **deterministically**?

We presented the Fiedler's algorithm which runs in time $\tilde{O}(m)$, but the algorithm is not deterministic because we sample a vector on the hypercube uniformly at random. The current best known deterministic algorithm runs in time $O(m^{1.01})$ [CGLNPS'20]². If you can solve this problem, you will likely have a successful Ph.D.; Most fast, deterministic algorithms have this problem as a bottleneck, so if you can solve it, then it should lead to several improvements in existing algorithms!

2.4 Let's see examples and get more intuition

Let's interpret what the power method does.

Suppose that the graph G is d -regular. Then we have

$$\mathbf{M} = 2\mathbf{I} - \mathbf{N}_G = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2} = \mathbf{I} + \frac{\mathbf{A}}{d}.$$

Let's consider the matrix

$$\mathbf{M}' = \frac{\mathbf{M}}{2} = \frac{\mathbf{I}}{2} + \frac{\mathbf{A}}{2d}$$

This matrix is easier to work with because its eigenvalues are between $[0, 1]$. It is just a scaled version of \mathbf{M} .

Consider

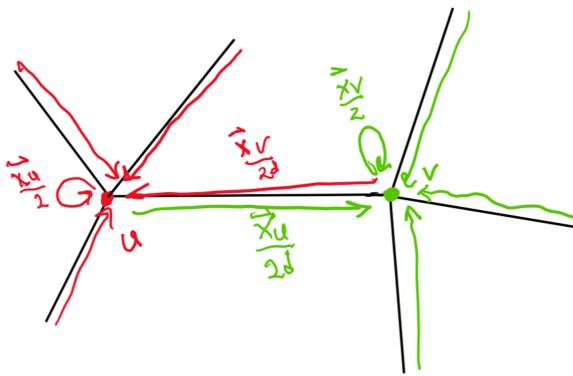
$$\mathbf{x}' = \mathbf{M}'\mathbf{x} = \left(\frac{\mathbf{I}}{2} + \frac{\mathbf{A}}{2d}\right)\mathbf{x}$$

which represents (up to scaling) one matrix multiplication in the power method. We can write

$$\mathbf{x}'_u = \frac{\mathbf{x}_u}{2} + \sum_{(u,v) \in E} \frac{\mathbf{x}_v}{2d}.$$

In total, there are k rounds in the power method and by the equation, in each round, each node u updates its value as a weighted average between its one value and its neighbors' values. This looks like a diffusion process like in Figure 2.

Figure 2: u and v are both updating their values. Because (u, v) are connected, they send some of their value to each other in addition to receiving value from their other neighbors. They also retain half of their value.

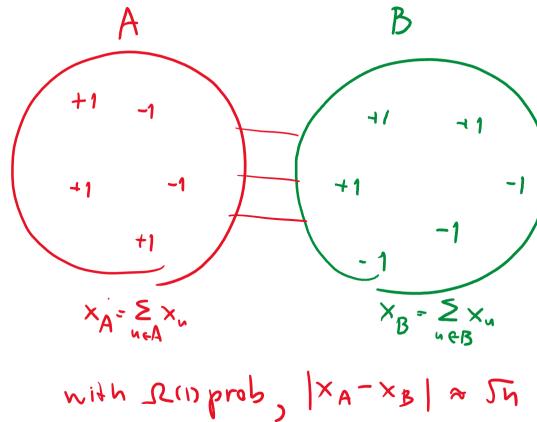


²<https://arxiv.org/pdf/1910.08025.pdf>

See [this video](#)³ about the algorithm in action for cycles and hypercubes.

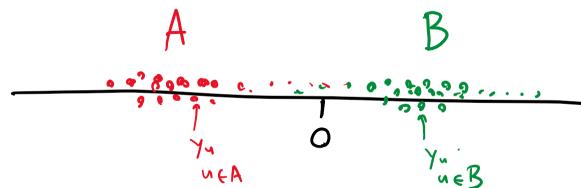
So, intuitively, why should this process give sparse cuts? Suppose there is a balanced sparse cut (A, B) (as in Figure 3) but A and B are themselves well-connected respectively. Denote $x_A = \sum_{u \in A} x_u$, $x_B = \sum_{u \in B} x_u$. In the beginning, since \mathbf{x} is chosen uniformly from the hypercube, in expectation x_A and x_B are 0. Wlog we assume that $x_A \approx -\sqrt{n} < 0$ and $x_B \approx \sqrt{n} > 0$ (since $|x_A| = |x_B| = \Theta(\sqrt{n})$ with $\Omega(1)$ probability.) Then, after updating $\mathbf{x} = (\mathbf{M}')^k \mathbf{x}$, the sums won't change too much, i.e. $x_A \approx -\sqrt{n}$ and $x_B \approx \sqrt{n}$ since there are few edges connecting between A and B whence tiny amount of weight got sent between A and B . What's more, since the process allow vertices to average their value with their neighbors, and vertices in A will mainly average with vertices within A , whence the value of vertices in A will be close to each other and most of them will be negative since we started with $x_A < 0$. Similarly, the values of vertices in B will still be close to each other and most of them will be positive.

Figure 3: (A, B) is a balanced sparse cut. Each vertex was assigned to -1 or 1 uniformly.



Fiedler's algorithm just sorts vertices according to $\mathbf{y} = (\mathbf{M}')^k \mathbf{x}$ (as shown in Figure 4) and sweeps through the order for the sparsest cut among the nested cuts. One of these cuts should be close to (A, B) so it should be sparse. This should also hold in the general case.

Figure 4: After computing $\mathbf{y} = (\mathbf{M}')^k \mathbf{x}$, Fiedler's algorithm sorts vertices according to \mathbf{y} and sweeps across the nested cuts for the sparsest. Though there are some vertices in B that end with value less than 0 and vertices in A that end with value more than 0, there should be a cut such that there are not so many vertices on the wrong side - this cut should be close to (A, B) .

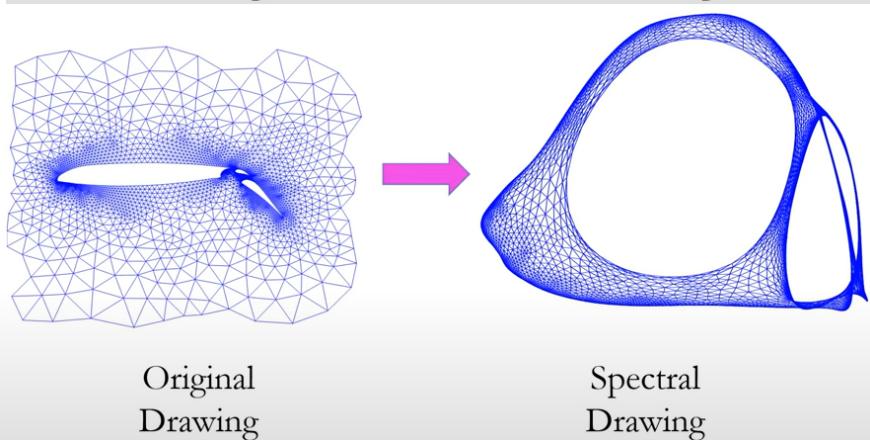
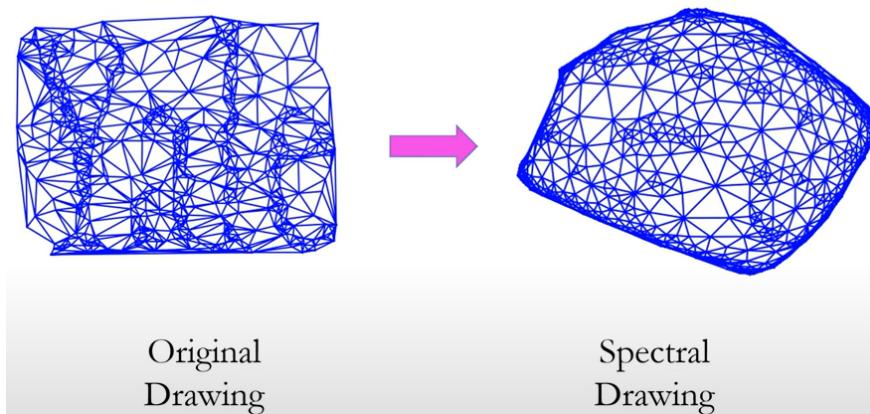
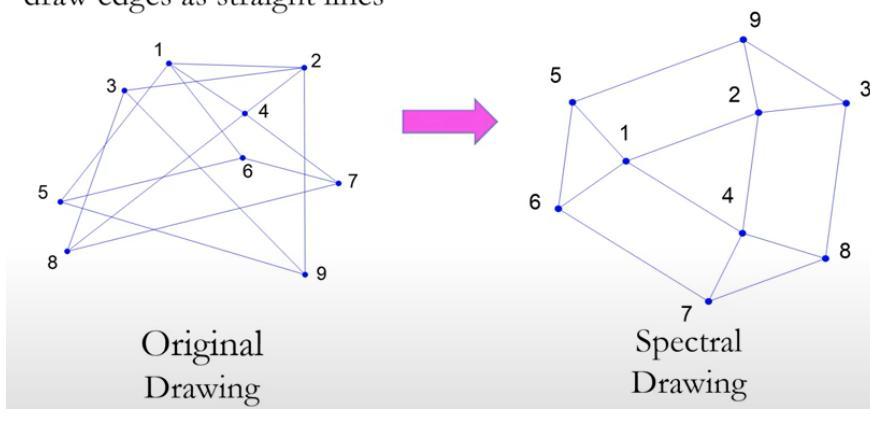


³<https://www.youtube.com/watch?v=cbYcn8o0Jfg>

3 Digression: Spectral Graph Drawing

Let's look at the magic of \mathbf{v}_2 and \mathbf{v}_3 of L_G .

Plot vertex a at $(v_2(a), v_3(a))$
draw edges as straight lines



Automatically, the second and third eigenvectors tells us a lot about the topology of the graph.

⁴Image from <https://www.youtube.com/watch?v=CDMQR422LGM>

4 Additional Discussion in Exercise Session

4.1 Removing the $\frac{1}{\Phi(G)^2}$ Dependency from the Power Method

We obtained an additive approximation when we worked with $\mathbf{M} = 2\mathbf{I} - \mathbf{N}_G$ because we were approximating eigenvalues of the form $2 - \lambda_i$. The approximation is multiplicative for $2 - \lambda_i$, but when we view it to an approximation for λ_2 , we obtain an additive approximation.

If we instead work with the inverse \mathbf{N}_G^{-1} , the eigenvalues should be of the form $1/\lambda_i$, but the first eigenvalue of \mathbf{N}_G is $\lambda_1 = 0$ so \mathbf{N}_G does not have an inverse. Instead, we will work with the **pseudo-inverse** \mathbf{N}_G^\dagger . \mathbf{N}_G^\dagger is defined as

$$\mathbf{N}_G^\dagger = \sum_{i:\lambda_i \neq 0} \frac{1}{\lambda_i} v_i v_i^\top$$

where the v_i are from the spectral decomposition of \mathbf{N}_G^\dagger and \mathbf{N}_G^\dagger has eigenvalues

$$0, \frac{1}{\lambda_2} \geq \frac{1}{\lambda_3} \geq \cdots \geq \frac{1}{\lambda_n}.$$

We can directly use \mathbf{N}_G^\dagger to get a multiplicative approximation for eigenvalues of the form $\frac{1}{\lambda_i}$ which can then be used to obtain a multiplicative approximation for λ_i .

Exercise 4.1. Show that $\mathbf{y} = (\mathbf{N}_G^\dagger)^k \mathbf{x}$ satisfies $\frac{\mathbf{y}^\top \mathbf{N}_G \mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \leq (1 + \varepsilon) \lambda_2$ for all $\varepsilon > 0$ and $k = O(\log \frac{n}{\varepsilon} / \varepsilon)$.

Proof. Similar to the analysis of POWER we upper bound $\mathbf{y}^\top \mathbf{N}_G \mathbf{y}$ and lower bound $\mathbf{y}^\top \mathbf{y}$. As before, we express \mathbf{x} in the eigenvectors of $(\mathbf{N}_G^\dagger)^\dagger$ so $\mathbf{x} = \sum_{i:\lambda_i \neq 0} a_i \mathbf{v}_i$ where $a_i = \langle \mathbf{v}_i, \mathbf{x} \rangle$.

Observe that $\mathbf{y} = \sum_{i:\lambda_i \neq 0} a_i \frac{1}{\lambda_i^k} \mathbf{v}_i$, $\mathbf{y}^\top \mathbf{N}_G \mathbf{y} = \sum_{i:\lambda_i \neq 0} a_i^2 \frac{1}{\lambda_i^{2k-1}}$, and $\mathbf{y}^\top \mathbf{y} = \sum_{i:\lambda_i \neq 0} a_i^2 \frac{1}{\lambda_i^{2k}}$.

Fix $\varepsilon' = \varepsilon/2$. Again, we define ℓ to separate the eigenvalues such that $\lambda_2 \leq \cdots \leq \lambda_\ell \leq (1 + \varepsilon')\lambda_2$ and $(1 + \varepsilon')\lambda_2 < \lambda_{\ell+1} \leq \cdots \leq \lambda_n$.

Then we obtain the following bounds:

$$\begin{aligned}
\mathbf{y}^\top \mathbf{N}_G \mathbf{y} &= \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} a_i^2 \frac{1}{\lambda_i^{2k-1}} + \sum_{\substack{i: \lambda_i \neq 0 \\ i > \ell}} a_i^2 \frac{1}{\lambda_i^{2k-1}} \\
&\leq \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} a_i^2 \frac{1}{\lambda_i^{2k-1}} + \frac{1}{(1 + \varepsilon')^{2k-1} \lambda_2^{2k-1}} \sum_{\substack{i: \lambda_i \neq 0 \\ i > \ell}} a_i^2 \\
&\leq \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} a_i^2 \frac{1}{\lambda_i^{2k-1}} + \frac{1}{(1 + \varepsilon')^{2k-1} \lambda_2^{2k-1}} \|\mathbf{x}\|^2 \\
&= \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} a_i^2 \frac{1}{\lambda_i^{2k-1}} + \frac{1}{(1 + \varepsilon')^{2k-1} \lambda_2^{2k-1}} a_2^2 \frac{\|\mathbf{x}\|^2}{a_2^2} \\
&= \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} a_i^2 \frac{1}{\lambda_i^{2k-1}} + \frac{4n}{(1 + \varepsilon')^{2k-1}} \frac{a_2^2}{\lambda_2^{2k-1}} \quad (\text{using } a_2^2 = |\langle \mathbf{x}, \mathbf{v}_2 \rangle|^2 \geq \frac{1}{4}) \\
&\leq \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} a_i^2 \frac{1}{\lambda_i^{2k-1}} + \frac{4n}{(1 + \varepsilon')^{2k-1}} \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} \frac{a_i^2}{\lambda_i^{2k-1}} \\
&= \left(1 + \frac{4n}{(1 + \varepsilon')^{2k-1}}\right) \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} \frac{a_i^2}{\lambda_i^{2k-1}} \\
\\
\mathbf{y}^\top \mathbf{y} &= \sum_{i: \lambda_i \neq 0} a_i^2 \frac{1}{\lambda_i^{2k}} \\
&= \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} a_i^2 \frac{1}{\lambda_i^{2k}} + \sum_{\substack{i: \lambda_i \neq 0 \\ i > \ell}} a_i^2 \frac{1}{\lambda_i^{2k}} \\
&\geq \frac{1}{(1 + \varepsilon') \lambda_2} \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} a_i^2 \frac{1}{\lambda_i^{2k-1}}
\end{aligned}$$

So

$$\frac{\mathbf{y}^\top \mathbf{N}_G \mathbf{y}}{\mathbf{y}^\top \mathbf{y}} \leq \frac{\left(1 + \frac{4n}{(1 + \varepsilon')^{2k-1}}\right) \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} \frac{a_i^2}{\lambda_i^{2k-1}}}{\frac{1}{(1 + \varepsilon') \lambda_2} \sum_{\substack{i: \lambda_i \neq 0 \\ i \leq \ell}} a_i^2 \frac{1}{\lambda_i^{2k-1}}} = \left(1 + \frac{4n}{(1 + \varepsilon')^{2k-1}}\right) (1 + \varepsilon') \lambda_2.$$

When we take $k = O(\log(\frac{n}{\varepsilon'})/\varepsilon')$, we have that

$$\left(1 + \frac{4n}{(1 + \varepsilon')^{2k-1}}\right) (1 + \varepsilon') \lambda_2 \leq (1 + 2\varepsilon') \lambda_2 = (1 + \varepsilon) \lambda_2.$$

□

However, to directly use \mathbf{N}_G^\dagger to get a multiplicative approximation for eigenvalues of the form $\frac{1}{\lambda_i}$, we would need to compute $(\mathbf{N}_G^\dagger)^k \mathbf{x}$. Computing \mathbf{N}_G^\dagger itself can be expensive - \mathbf{N}_G^\dagger can be dense, even if G is sparse. Observe that $\mathbf{N}_G^\dagger \mathbf{x}$ is the same as finding \mathbf{v} where

$$\mathbf{N}_G \mathbf{v} = \mathbf{x}.$$

This is a Laplacian linear system. Laplacian linear systems can be solved in near-linear time⁵! (I hope I will have time to cover this).

4.2 Proof of the Paley-Zygmund inequality

Lemma 4.2 (Paley-Zygmund inequality). *If Z is a non-negative random variable with finite variance, then, for every $0 \leq \delta \leq 1$,*

$$\Pr(Z \geq \delta \cdot \mathbb{E}[Z]) \geq (1 - \delta)^2 \cdot \frac{(\mathbb{E}[Z])^2}{\mathbb{E}[Z^2]}$$

Proof. $\mathbb{E}[Z] = \mathbb{E}[Z \cdot \mathbf{1}_{\{Z \geq 0\}}]$ where $\mathbf{1}_{\{Z \geq 0\}}$ is the indicator random variable taking value 1 when $Z \geq 0$ and value 0 otherwise. $\mathbb{E}[Z \cdot \mathbf{1}_{\{Z \geq 0\}}] = \mathbb{E}[Z \cdot \mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}} + Z \cdot \mathbf{1}_{\{Z < \delta \cdot \mathbb{E}[Z]\}}]$. We have $\mathbb{E}[Z \cdot \mathbf{1}_{\{Z < \delta \cdot \mathbb{E}[Z]\}}] \leq \delta \cdot \mathbb{E}[Z]$, so $\mathbb{E}[Z \cdot \mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}}] \geq (1 - \delta) \mathbb{E}[Z]$.

We can define the inner product on the set of random variables by the expectation of their product i.e. $\langle X, Y \rangle = \mathbb{E}[XY]$. Then by the Cauchy-Schwarz inequality, we have

$$\begin{aligned} \langle Z, \mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}} \rangle^2 &\leq \langle Z, Z \rangle \cdot \langle \mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}}, \mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}} \rangle \\ \iff \mathbb{E}[Z \cdot \mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}}]^2 &\leq \mathbb{E}[Z^2] \cdot \mathbb{E}[\mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}}^2] \\ \iff \mathbb{E}[Z \cdot \mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}}] &\leq \sqrt{\mathbb{E}[Z^2]} \cdot \sqrt{\mathbb{E}[\mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}}^2]} \end{aligned}$$

Then, $\mathbb{E}[\mathbf{1}_{\{Z \geq \delta \cdot \mathbb{E}[Z]\}}] = \Pr[Z \geq \delta \cdot \mathbb{E}[Z]]$ and by rearrangement and substitution, we obtain the inequality. \square

4.3 Inner Product between a Vector with a Random Point from the Hypercube

We mentioned earlier a lemma about the inner product between a vector and a random point from the hypercube in order to prove the performance of POWER. This is a proof of the lemma.

Lemma 4.3. *For any vector $\mathbf{v} \in \mathbb{R}^n$ where $\|\mathbf{v}\| = 1$. Sample $\mathbf{x} \in \{-1, 1\}^n$. Then*

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

Remark 4.4. The proof of Lemma 4.3 works even if $\mathbf{x} \sim \{-1, 1\}^n$ is selected according to a 4-wise independent distribution. This means that the algorithm can be derandomized in polynomial time.

⁵See the Spielman-Teng '04 paper

Proof. From Luca's lecture note⁶. We apply the Paley-Zygmund inequality to the random variable $\langle \mathbf{x}, \mathbf{v} \rangle^2$.

Define Z to be the random variable $\langle \mathbf{x}, \mathbf{v} \rangle = \sum_i x_i v_i$. We compute its first, second, and fourth moments to be:

$$\mathbb{E}[Z] = 0.$$

$$\mathbb{E}[Z^2] = 1.$$

$$\mathbb{E}[Z^4] = 3 \left(\sum_i v_i^2 \right) - 2 \sum_i v_i^4 \leq 3.$$

Then observe that $\text{Var}(Z) = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2 = 1$. Having the first, second, and fourth moment of Z , we can apply the Paley-Zygmund inequality to Z^2 and $\delta = 1/4$, so we obtain

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

□

⁶<https://lucatrevisan.github.io/teaching/expanders2016/lecture08.pdf>