CS 466 / 666 : Algorithm Design and Analysis . Spring 2018 , Waterloo.

Lecture 12: Cheeger's inequality

We will study Cheeger's inequality which relates the second eigenvalue of the Laplacian matrix to the (combinatorial) expansion of a graph. We will also discuss some recent generalizations.

Graph expansion

Recall that $\lambda_2 = 0$ if and only if the graph is disconnected.

Cheeger's inequality will show that λ_2 is "small" if and only if the graph G is "close" to disconnected.

First, let us make precise what it means for a graph to be close to disconnected.

There are different definitions to measure how well a graph is connected.

The <u>expansion</u> of a graph is defined as $\Phi(G) := \min_{S \subseteq V, |S| \leq |V|/2} \Phi(S)$, where $\Phi(S) := \frac{|\delta(S)|}{|S|}$,

the ratio of the number of edges cut to the number of vertices in the set.

The <u>conductance</u> of a graph is defined as $\phi(G) := \min_{S \subseteq V, \mathrm{vol}(S) \leq |E|} \phi(S)$, where $\phi(S) := \frac{|\delta(S)|}{\mathrm{vol}(S)}$ and $\mathrm{vol}(S) := \sum_{v \in S} \deg(v)$, the ratio of the number of edges cut to the total degree in the set.

These definitions are basically equivalent when the graphs are d-regular ($\Phi(S) = d\phi(S)$).

In non-regular graphs, we will relate the graph conductance to the second eigenvalue.

We say a graph is an <u>expander graph</u> if $\phi(G)$ is large (e.g. $\phi(G) \geq 0.1$), and we say $S \subseteq V$ a <u>sparse cut</u> if $\phi(S)$ is small. Note that $0 \leq \phi(S) \leq 1$ for every $S \subseteq V$.

Both concepts are very useful. As we have seen, sparse expander graphs are "magical" and have algorithmic applications, and we may also see that they can be used in derandomization.

Finding a sparse cut is useful in designing divide-and-conquer algorithms, and have applications in image segmentation, data clustering, community detection in social networks, VLSI design, etc.

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 $\mathcal{L}$ (the eigenvector corresponding to the second largest eigenvalue)

② Sort the vertices so that $x_1 \geq x_2 \geq \ldots \geq x_n$ (where $n = |V|$ is the number of vertices)

③ Let $S_i = \begin{cases} \{1, \ldots, i\} & \text{if } i \leq n/2 \\ \{\ldots\} & \text{if } i > n/2 \end{cases}$

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

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 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.

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 15 years ago.

The proof of Cheeger's inequality will provide some performance guarantee of this algorithm.

Normalized matrices

To state Cheeger's inequality nicely, we will use the "normalized" Laplacian matrix, which allows us to remove the dependency on the maximum degree of the graph.

Given an adjacency matrix A, let $\mathcal{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ be the normalized adjacency matrix, and let $\mathcal{L} = I - \mathcal{A}$ be the normalized Laplacian matrix., where D is the diagonal matrix whose i-th entry is the degree of vertex i. Note that $\mathcal{L} = I - \mathcal{A} = D^{-\frac{1}{2}}(D-A)D^{-\frac{1}{2}} = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$.

Let $\alpha_1 \ge \alpha_2 \ge \dots \ge \alpha_n$ be the eigenvalues of $\mathcal{A}$, and let $\lambda_1 \le \lambda_2 \le \dots \le \lambda_n$ be the eigenvalues of $\mathcal{L}$.

Claim $1 = \alpha_1 \ge \alpha_n \ge -1$ and $0 = \lambda_1 \le \lambda_n \le 2$.

Proof We prove the result for normalized adjacency, and the result for normalized Laplacian follows easily.

Note that 0 is an eigenvalue for $\mathcal{L}$, as $\mathcal{L}(D^{\frac{1}{2}}\vec{1}) = (D^{-\frac{1}{2}} L D^{-\frac{1}{2}})(D^{\frac{1}{2}}\vec{1}) = D^{-\frac{1}{2}} L \vec{1} = 0$

To prove $\lambda_1 = 0$, we will show that $\mathcal{L}$ is a positive semidefinite matrix.

To see it, observe that $x^T \mathcal{L} x = x^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} x = \sum_{e \in E} x^T D^{-\frac{1}{2}} L_e D^{-\frac{1}{2}} x = \sum_{e=ij \in E} \left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2 \ge 0$,

where $L_e = b_e b_e^T$ that we defined last time (i.e. $L_e = \begin{array}{c|cc} & i & j \\ \hline i & 1 & -1 \\ j & -1 & 1 \end{array}$ for $e = ij$).

This implies that $I - \mathcal{A} \succcurlyeq 0$, and thus $\alpha_1 \le 1$.

where $L_e = b_e b_e^T$ that we defined last time (i.e. $L_e = \begin{array}{c} i \\ j \end{array}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$ for $e = ij$).

This implies that $I - \mathcal{A} \succcurlyeq 0$, and thus $\alpha_1 \leq 1$.

Also, we can write $x^T(I + \mathcal{A})x = x^T \mathcal{L} x + 2x^T \mathcal{A} x = \sum_{e=ij \in E} \left(\left(\frac{x_i}{\sqrt{d_i}} - \frac{x_j}{\sqrt{d_j}} \right)^2 + \frac{2 x_i x_j}{\sqrt{d_i d_j}} \right) = \sum_{e=ij\in E} \left(\frac{x_i}{\sqrt{d_i}} + \frac{x_j}{\sqrt{d_j}} \right)^2 \geq 0$,

and this implies that $I + \mathcal{A} \succcurlyeq 0$, and thus $\alpha_n \geq -1$, and hence $\lambda_n = 1 - \alpha_n \leq 2$. $\square$

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

For simplicity, we assume the graph is d-regular, in which case $\mathcal{L} = \frac{1}{d} L$. The general case is similar.

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

So, naturally we will prove the easy direction first.

(assuming d-regular)

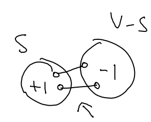
One nice thing about the Laplacian matrix is that we know that the first eigenvector is the all-one vector (unlike for adjacency matrix), and by the characterization of λ_2 using Rayleigh quotient, we have

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

So, 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 $\lambda_2 \leq \frac{\sum_{ij \in E} (x_i - x_j)^2}{d \sum_{i \in V} x_i^2} = \frac{4|\delta(S)|}{d|V|} = \frac{2|\delta(S)|}{d|S|} = 2\phi(S)$.

each edge in $\delta(S)$ contributes 4.

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 $\lambda_2 \leq \frac{\sum_{ij \in E} (x_i - x_j)^2}{d \sum_{i \in V} x_i^2} = \frac{|\delta(S)| \cdot \left(\frac{1}{|S|} + \frac{1}{|V-S|} \right)^2}{d \left(|S| \cdot \frac{1}{|S|^2} + |V-S| \cdot \frac{1}{|V-S|^2} \right)} = \frac{|\delta(S)| \cdot |V|}{d \cdot |S| \cdot |V-S|} \leq 2\phi(S)$

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.

This direction is useful in deterministic construction of expander graphs.

<u>The Hard Direction: Intuition</u>

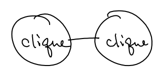
In the minimization problem $\min \frac{\sum_{ij \in E} (x_i - x_j)^2}{\ldots}$, if we can only search for "binary" solutions,

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 $(x_i - x_j)^2 / (x_i^2 + x_j^2)$.

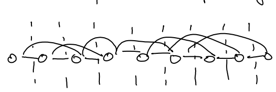
In this case, it is not good to "split" the vertices in a clique, because there are so many edges within it So, we would expect that the values in each clique are very similar, while the two cliques would have different values so that $x \perp \vec{1}$. Hence, we expect that the minimizer would look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

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 $|S| \leq |V|/2$.

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: $y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ 0 & \text{if } x_i < 0 \end{cases}$.

Denote the Rayleigh quotient by $R(x) := x^T \mathcal{L} x / x^T x$.

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

proof $(\mathcal{L} y)_i = y_i - \sum_{j \in N(i)} \frac{y_j}{d} \leq x_i - \sum_{j \in N(i)} \frac{x_j}{d} = (\mathcal{L} x)_i = \lambda_2 \cdot x_i$, for all i with $y_i > 0$.

Therefore, $y^T \mathcal{L} y = \sum_{i \in V} y_i \cdot (\mathcal{L} y)_i \leq \sum_{i: y(i) > 0} \lambda_2 x_i^2 = \sum_i \lambda_2 y_i^2$, 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 $S \subseteq \mathrm{supp}(y)$ such that $\phi(S) \leq \sqrt{2R(y)}$, where $\mathrm{supp}(y) = \{ i \mid y(i) \neq 0 \}$.

Proof We can assume that $0 \leq y_i \leq 1$ 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 $S_t \subseteq \mathrm{supp}(y)$ by construction.

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

$$\begin{aligned}
E_t(|\delta(S_t)|) &= \sum_{ij \in E} [\Pr(\text{the edge } ij \text{ is cut})] \quad \text{by linearity of expectation} \\
&= \sum_{ij \in E} [\Pr(y_i^2 < t \leq y_j^2)] \\
&= \sum_{ij \in E} |y_i^2 - y_j^2| \\
&= \sum_{ij \in E} |y_i - y_j| \, |y_i + y_j| \\
&\leq \sqrt{\sum_{ij \in E} (y_i - y_j)^2} \sqrt{\sum_{ij \in E} (y_i + y_j)^2} \quad \text{by Cauchy-Schwarz } \langle a, b \rangle \leq \|a\| \cdot \|b\| \\
&\leq \sqrt{\sum_{ij \in E} (y_i - y_j)^2} \sqrt{2 \sum_{ij \in E} (y_i^2 + y_j^2)} \\
&= \sqrt{\sum_{ij \in E} (y_i - y_j)^2} \sqrt{2d \sum_{i \in V} y_i^2} \\
&= \sqrt{2R(y)} \left(d \sum_{i \in V} y_i^2 \right)
\end{aligned}$$

$$E[|S_t|] = \sum_{i \in V} \Pr[y_i^2 \geq t] = \sum_{i \in V} y_i^2$$

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

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

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

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 has searched for.

Discussions

① The proof can be generalized to weighted non-regular graphs, with suitable modifications.

② Both sides of Cheeger's inequality are tight, even the constants are tight.

To see an example where the hard direction is (almost) tight, consider a cycle of length n.

One can compute the spectrum of the cycle exactly, but we won't do it here.

Recall that $\lambda_2 = \min_{x \perp \vec{1}} \frac{x^T \mathcal{L} x}{x^T x}$, so to give an upper bound on λ_2, we just need to demonstrate one vector.

Consider $x = (1, 1-\frac{1}{n}, 1-\frac{2}{n}, \ldots, \frac{1}{n}, 0, -\frac{1}{n}, \ldots, -1+\frac{1}{n}, -1, -1+\frac{1}{n}, \ldots, 0, \frac{1}{n}, \ldots, 1)$.

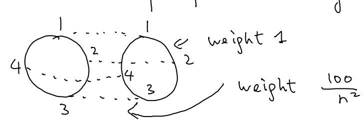
Then $\lambda_2 \leq \frac{\sum_{i \sim j} (x_i - x_j)^2}{2 \sum_i x_i^2} = O\left(\frac{n(\frac{1}{n})^2}{n}\right) = O(\frac{1}{n^2})$.

On the other hand, it is easy to verify that the expansion of a cycle is $\Omega(\frac{1}{n})$.

Therefore, in this example, $\phi(G) = \Omega(\sqrt{\lambda_2})$.

You may think that it is an artificial example in which the second eigenvalue clearly underestimates the expansion, but let's consider the following related example.

Two cycles of length n, and there is a perfect matching between the two cycles, where each edge has weight $100/n^2$.



Clearly, the optimal sparse cut is the perfect matching, with $\phi(G) = O(\frac{1}{n^2})$.

On the other hand, one can show that the second eigenvector would still be the same as the cycle example (with two nodes in the perfect matching identified as one node).

Therefore, λ_2 is still $O(\frac{1}{n^2})$ and the value is correct, but the optimal cut is lost and every threshold cut is bad.

These examples show how Cheeger's inequality got cheated, both in terms of the value and in terms of the actual cut returned.

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. There are some recent generalization that addressed this question.

④ The second eigenvalue is closely related to the mixing time of random walks, and so Cheeger's inequality provides a combinatorial approach to bound the mixing time, which we will see in next lecture.

⑤ The proof can be modified to show that any vector with Rayleigh quotient $\approx \lambda_2$ could be used to produce a sparse cut of conductance $O(\sqrt{\lambda_2})$, i.e. it doesn't have to be an eigenvector. This point is useful in some other proofs, but we probably won't see in this course.

References Course notes of Dan Spielman and Luca Trevisan.