COMS 4995-8: Advanced Algorithms (Spring'21) Mar 11, 2021

Lecture 16: Cheeger Inequality

Instructor: *Alex Andoni* Scribes: *Shaoyu Liu, Chengrui Zhou*

1 Spectral Partitioning

1.1 Recap

Recall from last lecture we have:
1) $\mu_2 = 0$ if and only if graph G is disconnected.
2) We define the Laplacian and its normalized form for a graph G = (V,E):

$$L = D - A$$

$$\hat{L} = D^{-1/2} L D^{-1/2}$$

$$= D^{-1/2}(D - A)D^{-1/2} = I - \hat{A}$$

where L is the Laplacian matrix, $\hat{L}$ is the normalized Laplacian matrix, D is the degree matrix, A is the adjacency matrix and $\hat{A}$ is the normalzied adjacency matrix.

1.2 Conductance

Definition 1: A cut is defined as a separation of a graph G into two subsections S and $\bar{S}$.
Definition 2: for cut S,

$$\partial S = \{(i,j) \in E, i \in S, j \in \bar{S}\}$$

In words, ∂S is the number of edges crossing S and $\bar{S}$.

We want to develop a quantitative statement to show how close to disconnected a graph is. This motivates the idea of conductance.
Definition 3: Conductance: We define the conductance of a cut S to be:

$$\phi(S) = \frac{|\partial S|}{\text{vol}\, S} \quad \text{where} \quad \text{vol}(S) = \sum_{i \in S} d_i$$

We define the conductance of a graph G to be:

$$\phi(G) = \min_{S \subset V} \phi(S) \text{ s.t. } 0 < \text{vol}(S) \leq \frac{\text{vol}(V)}{2}$$

Alternatively, it can be written as:

$$\phi(G) = \min_{S: vol(S) \neq 0} \frac{|\partial S|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}}$$

2 Cheeger's Inequality

Definition 4: Cheeger's Inequality

$$\frac{\mu_2}{2} \leq \phi(G) \leq \sqrt{2\mu_2}$$

Cheeger's inequality allows us to bound the connectivity of a graph, and get an idea of how "connected" a graph is just from its Laplacian. The left-hand inequality is somewhat easier and more "intuitive" to prove, while the right-hand inequality involves a much more involved proof.

2.1 Proof of $\frac{\mu_2}{2} \leq \phi(G)$

Remember from last lecture,

$$\mu_2 = \min_{x \neq 0, x \perp v_1} \frac{x^T \hat{L} x}{x^T x}$$

where $\frac{x^T \hat{L} x}{x^T x}$ is R($\hat{L}$, x), the Rayleigh quotient for $\hat{L}$ and nonzero vector x.
Therefore, we need to construct some x, such that it satisfies the two conditions:
1) $\mathbf{e}^T D^{\frac{1}{2}} x = 0$;
2) $R(\hat{L}, x) \leq 2\phi(G)$
Remember the definition for $\phi(G)$: $\phi(G) = \min_{S \subset V} \phi(S)$ s.t. $0 < \text{vol}(S) \leq \frac{\text{vol}(V)}{2}$. Let $S^* = \arg\min_S \phi(S)$ where $S \subset V$ and $0 < \text{vol}(S) \leq \frac{\text{vol}(V)}{2}$, so $\phi(G) = \phi(S^*) = \frac{\partial S^*}{Vol(S^*)}$.

Note: $R(\hat{L}, x) = \frac{x^T \hat{L} x}{x^T x} = \frac{x^T D^{-\frac{1}{2}} L D^{-\frac{1}{2}} x}{x^T x}$.
Change notation, let $D^{-\frac{1}{2}} x = y$, then $x = D^{\frac{1}{2}} y$. As a result, we can write:

$$R(\hat{L}, x) = \frac{y^T L y}{y^T D y} \tag{1}$$

Remember we have shown that $y^T L y = \sum_{(i,j) \in E} (y_i - y_j)^2$ before, plug into equation (1), we have:

$$R(\hat{L}, x) = \frac{\sum_{(i,j) \in E} (y_i - y_j)^2}{y^T D y} \tag{2}$$

Now the problem is to find such a y where 1) and 2) above hold.

So define $\mathbf{y} = \mathbb{1}_{S^*}$ where $y_i = \begin{cases} 1 & \text{if } i \in S^* \\ 0 & \text{otherwise} \end{cases}$ Note by such definition, we immediately find that $y^T L y = \sum_{(i,j) \in E} (y_i - y_j)^2 = |\partial S^*|$, where the second equality follows from the definition of y_i, where only an edge crossing from S to $\bar{S}$ will contribute to the summation. However, such definition of $\mathbf{y}$ does not satisfy 1) because $\mathbf{e}^T D^{\frac{1}{2}} x = \mathbf{e}^T y \neq 0$.

To deal with this, we have to modify the definition of $\mathbf{y}$ slightly. Let $\mathbf{y}' = \mathbf{y} - \sigma \times \mathbf{e}$ where σ is a variable to be found to make $\mathbf{e}^T \mathbf{y} = 0$. Note that we still have $(y')^T L y' = |\partial S^*|$. Now the goal is to solve for an appropriate σ.

$\mathbf{e}^T\mathbf{y} = 0 \Rightarrow \mathbf{e}^T D(1_S - \sigma\mathbf{e}) = \sum_{i\in S^*} d_i(1-\sigma) + \sum_{i\notin S^*} d_i(-\sigma) = (1-\sigma)Vol(S) - \sigma(Vol(V) - Vol(S) \Rightarrow Vol(S) - \sigma Vol(V) \Rightarrow \sigma = \frac{\text{vol}(S)}{\text{vol}(V)}$.

Thus, we have found that $\sigma = \frac{\text{vol}(S)}{\text{vol}(V)}$ will make y' s.t. $e^T y' = 0$ and $(y')^T Ly' = |\partial S^*|$.

Next step, we will compute $(y')^T Dy'$.

$$\begin{aligned}
\left(y'\right)^T Dy' &= \sum_{i\in[n]} y_i' d_i y_i' \\
&= \sum_{i\in S^*} (1-\sigma)^2 d_i + \sum_{i\notin S^*} (-\sigma)^2 d_i \\
&= (1-\sigma)^2 \,\text{vol}\,(S^*) + \sigma^2\,(\text{vol}(V) - \text{vol}\,(S^*)) \\
&= (1 - \frac{\text{vol}(S^*)}{\text{vol}(V)})\,\text{vol}\,(S^*)
\end{aligned}$$

After that we check if such y' would make $R(\hat{L}, x) = \frac{y'^T L y'}{y'^T D y'} \leq 2\phi\,(S^*)$

$$R(\hat{L}, x) = \frac{(y')^T Ly'}{(y')^T Dy'} = \frac{|\partial S^*|}{(1-\sigma)\,\text{vol}\,(S^*)} = \frac{\phi\,(S^*)}{1-\sigma} \leq 2\phi\,(S^*) = 2\phi(G)$$

using the fact that $Vol(S) \leq \frac{1}{2} Vol(V)$ and $\phi(S^*) = \phi(G)$.

2.2 Proof of $\phi(G) \leq \sqrt{2\mu_2}$

Introduction behind the proof:

$$\mu_2 = \min_{x\in\mathbb{R}^n, x\neq 0, x\perp v_1} R(x)$$

x need not look like

$$x = D^{1/2}(\not{\Vdash} s - \sigma \times \mathbf{e})$$

We now use Spectral Partitioning Algorithm:
Find a set S with

$$\phi(s) \leqslant \sqrt{2\mu_2}\ .$$

Build S from 2nd eigenvector $V_2 = \underset{\substack{x\neq 0 \\ x\perp V_1}}{\text{argmin}} R(x) \quad \in \mathbb{R}^n$

Sort all nodes

$$i \in [n]$$

$$\pi_1, \pi_2, \ldots \pi_n \in [n]: \quad V_2\pi_i \leqslant V_2, \pi_{i+1}$$

Iterate i=1,2,...,n-1, consider

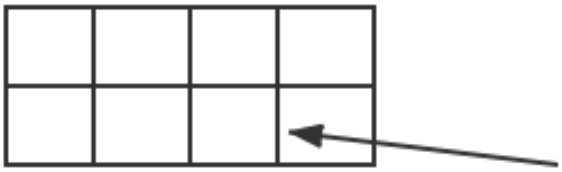
$$S_i = \{\pi_1, \pi_2, \ldots \pi_1\}$$

and compute

$$\phi(s_i) = \frac{\partial s}{\min\{vol(s), vol(s')\}}$$

2.3 Example

Now we give an example:



$$\text{Weight of edge } = \frac{1}{\text{difference in gray scale}}$$

Now what we want to do is to find a cut that minimizes weight of cut edges, and we can get

$$\phi(s) = \frac{\partial S}{\text{vol}(s)}.$$

Remark:Spectral Partitioning Algorithm runs in O(m + nlgn) time after having computed V_2.

3 Optimization

Now we are going to talk about:
——Linear Programming and the quality:
One of the main goals here will be to give a polynomial time algorithm following a program.
——Gradient Descent, Newton method (2^{nd} method)
——IPM(Interior Point Method)

3.1 General Optimization Problem

We usually want to maximize or minimize f(x), which is subject to $x \in F \subset \mathbb{R}^n$, called feasible set.And linear programming is a instance.

Problem 1: Computing $\phi(G) = \min_{s} \phi(S)$ var: $x \in \mathbb{R}^h, x_i \in \{0,1\}$, whether $i \in S$

Solution 1:
What we want is

$$\min f(x) = \min \frac{\sum_{ij \in E} (x_i - x_j)^2}{\sum x_i^2}$$

We can take absolute values certain like this, level of generality. Usually having these little squares, not optimal values leads to nicer optimization problems. And this is what x transpose x rays, which is a summation of x ray squared.
There are more constrains that it is divided by the volume. Where volume was summation, over degree I. So we have

$$\min f(x) = \min \frac{\sum_{ij \in E} (x_i - x_j)^2}{\sum_{i=1}^{n} d_i x_i} \qquad (1)$$

The volume has to be positive, so

$$\sum x_i \geqslant 1 \qquad (2)$$

There should be at least one node there. And everyone is as volume has to be at most half volume of the entire reference summation of J. And we get

$$\sum_{i=1}^{n} d_i x_i \leqslant \frac{1}{2} \sum_{i=1}^{n} d \qquad (3)$$

x_i is either 0 or 1, so we have

$$x_i (x_i - 1) = 0 \qquad (4)$$

Combine (1),(2),(3),(4), we get $\phi(G)$.
And combine (2),(3),(4), we get set F.
This is so called NP-hard problem. These are the main barriers to being able to solve these and put them in real time. And the ratio in (1) is actually not an issue because we can always intimidate dissidents. Anyway, this is just an example connecting us to this chapter of an opposition problem. We don't expect to solve all of condition problems in a little time, but we hope to solve some of them. And there are different classes of of optimization problems that are solvable. The most classic example is linear program.

3.2 Linear Programming

Linear Programming is basically optimization set of transition problems where f is linear:$f(x) = c^T \cdot x$.
F is a linear as well. F is defined as follows $Ax \geqslant b$ $A_i x_i \geqslant b_i$. We have the most general form of linear programming:

$$\min \left\{ \mathbf{c}^{\mathrm{T}} \mathbf{x} \mid Ax \geqslant b \wedge \mathbf{x} \geq 0 \right\}$$