CSE 521: Design and Analysis of Algorithms I	**Winter 2017**

Lecture 11: Spectral Graph Theory

Lecturer: Shayan Oveis Gharan *2/15/2017* *Scribe: Tomer Kaftan*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications.*

In this lecture we discuss Spectral Graph Theory, Conductance, Cheeger's Inequality, and Spectral Clustering.

11.1 Spectral Graph Theory

In the field of spectral graph theory we relate combinatorial properties of graphs to their algebraic properties. Combinatorial properties of a graph G can refer to whether it is connected, its diameter, its spanning trees, etc. Algebraic properties refer eigenvalues/eigenvectors of Adjacency matrix (A), Laplacian matrix (L), Random Walk matrix P, etc.

Here are some known results from Spectral Graph Theory are shown below:

Theorem 11.1.

$$\lambda_2(L) = 0 \iff G \textit{ is disconnected}$$

This theorem relates the value of the 2nd smallest eigenvalue of the Laplacian L of G to whether or not G is connected.

Theorem 11.2. *For all $k \geq 2$,*

$$Diam(G) \leq \frac{k \log n}{\lambda_k(\tilde{L})}$$

This theorem specifies a relationship between the diameter of G and the eigenvalues of its Normalized Laplacian matrix $\tilde{L}$. The Normalized Laplacian Matrix will be defined later in the lecture.

Theorem 11.3. *For any graph G,*

$$\det(L + \frac{\mathbf{1}\mathbf{1}^T}{n}) = \# \textit{ of spanning trees}$$

where $\mathbf{1}$ *is the all ones vector.*

This theorem specifies that the determinant of the Laplacian of G with $\mathbf{1}\mathbf{1}/n$ added to the value of every cell, is equal to the number of spanning trees of G. Note that in general a graph may have an exponential number of spanning trees, but since the determinant of any matrix can be computed efficiently, the above theorem gives an efficient algorithm to count the number of spanning trees of a given graph G.

Today we will specifically be focusing on the Laplacian and Normalized Laplacian, and how they relate to spectral clustering.

11.1.1 Laplacian Matrix

In the last lecture we defined the Laplacian matrix. Recall, $L = D - A$ where D is the Degrees matrix of G and A is the Adjacency matrix of A. The quadratic form of the Laplacian is:

$$x^T Lx = \sum_{i \sim j} (x_i - x_j)^2$$

Where $i \sim j$ means that there exists an edge in G between i and j.

This quadratic form implies that $L \succcurlyeq 0$ (meaning the Laplacian is PSD, or positive semi-definite). This form also implies that the all-ones vector 1 is an eigenvector of L, with the eigenvalue 0.

With these definitions in place, we attempt to prove the first theorem stated earlier in the lecture, proving each direction if the iff separately.

Claim 11.4. *G is disconnected $\implies \lambda_2 = 0$*

Proof. Since G is disconnected, we can split it into two sets S and $\overline{S}$ such that $|E(S, \overline{S})| = 0$. Let

$$x = \frac{\mathbf{1}^S}{|S|} - \frac{\mathbf{1}^{\overline{S}}}{|\overline{S}|}$$

where as usual $\mathbf{1}^S$ represents the indicator of S.

The quadratic form of L implies that $x^T Lx = 0$, as all neighboring vertices were assigned the same weight in x. We now show that x is orthogonal to the first eigenvector 1.

$$\langle x, \mathbf{1} \rangle = \sum x_i \mathbf{1}_i = \sum_{i \in S} \frac{1}{|S|} - \sum_{i \in \overline{S}} \frac{1}{\overline{S}} = 0$$

Therefore $\lambda_2 = 0$. □

Claim 11.5. *$\lambda_2 = 0 \implies G$ is disconnected*

Proof. First, by the quadratic form,

$$\lambda_2 = 0 \implies \exists x \perp \mathbf{1} \ s.t.$$

$$x^T Lx = \sum_{i \sim j} (x_i - x_j)^2 = 0 \iff (x_i - x_j)^2 = 0 \ \forall i \sim j$$

For the sake of contradiction suppose G is connected. Then, $(x_i - x_j)^2 = 0 \ \forall i \sim j$ holds true only if $x_i = x_j \ \forall i, j$. However, we know that:

$$x \perp 1 : \langle x, 1 \rangle = \sum_i x_i = 0$$

So x is not a constant vector, i.e., $\exists i, j : x_i \neq x_j$ Therefore we have a contradiction and G must be disconnected. □

11.2 Graph Partitioning

We will now talk about graph partitioning. What are we actually looking for by partitioning a graph? If a graph comes from data points, and edges represent their similarity, then we may be partitioning it to find clustered data. If the graph comes from a social network where edges represent friendships, then we may be partitioning it to find communities of friends.

11.2.1 Graph Partitioning Objectives

In Computer Science, whether or not a partitioning of a graph is a 'good' partitioning depends on the value of an objective function, and graph partitioning is an optimization problem intended to find a partition that maximizes or minimizes the objective. The appropriate objective function to use depends on where the graph came from and what we are trying to find.

Some examples of objective functions are given below:

Definition 11.6 (Diameter)**.** *The diameter of a graph refers to the maximum of the shortest paths between all pairs of vertices in the graph. When using this objective function, the goal is to find the partitioning where the partitions have the smallest possible diameters, which implies some level of 'closeness' between vertices in the partition.*

Definition 11.7 (Clustering Coefficient:)**.** *Count the number of triangles in G. The number of triangles in a graph is useful as an objective when trying to partition social networks, as within a community we expect friends of friends to also be friends.*

Definition 11.8 (Conductance)**.** *Given a graph $G = (V, E)$ with the vertices partitioned into $(S, \overline{S})$, the conductance of partition S is defined as:*

$$\phi(S) = \frac{|E(S, \overline{S})|}{vol(S)}$$

where the volume of a partition is defined as the sum of the degrees of vertices in that partition:

$$\text{vol}(S) = \sum_{i \in S} d(i)$$

The above definition naturally extends to weighted graphs. Say G is weighted, and the weight of the edges connecting i to j is $w_{i,j}$. Then,

$$\phi(S) = \frac{w(S, \overline{S}}{\text{vol}_w(S)} = \frac{\sum_{i \sim j : i \in S, j \notin S} w_{i,j}}{\sum_{i \in S} d_w(i)},$$

where $d_w(i) = \sum_{j \sim i} w_{i,j}$ is the weighted degree of i.

11.2.2 Conductance

Conductance has the property that $0 \leq \phi(S) \leq 1$, with value 0 when all edges of vertices of S stay within S, and a value of 1 when all edges of vertices of S go to vertices in $\overline{S}$. In general we want to find partitions with small conductances.

An intuitive reason why we would want this, going back to the community finding example, is that if a community has a small conductance there are many more friendships within a community than between members of the community and people from outside of the community.

As $S = V \implies \phi(S) = 0$, in order to find the best partitioning according to conductance we look for S with the smallest conductance such that $vol(S) \leq \frac{1}{2} vol(V)$, and define the conductance of the best partitioning of G to be:

Definition 11.9.

$$\phi(G) = \min_{\text{vol}(S) \leq vol(V)/2} \phi(S)$$

For a cycle of n vertices, the smallest conductance partitioning occurs when its left and right halves are separate partitions, and it has conductance $\phi(S) = \frac{2}{n}$.

For a complete graph with n vertices the best partitioning occurs when the graph's vertices are partitioned into two equal halves, and it has conductance $\phi(S) = \frac{1}{2}$. In an intuitive sense this means that unlike a cycle, a complete graph is not partitionable.

11.3 Approximating $\phi(G)$: Sparsest Cut Problem

There are three well known algorithms for finding a partition that has a conductance close to $\phi(G)$.

Spectral Partitioning finds a partition S such that $\phi(S) \leq 2\sqrt{\phi(G)}$

Linear Programming finds a partition S such that $\phi(S) \leq O(\log n)\phi(G)$

Semi-Definite Programming (SDP) finds a partition S such that $\phi(S) \leq O(\sqrt{\log n})\phi(G)$

Finding an algorithm with an approximation factor better than $O(\sqrt{\log n})$ remains one of the big open problems in the theory of algorithm design. Here we will only be discussing Spectral Partitioning, but the Linear Programming algorithm is covered in the Random Algorithms class.

We will shortly describe the Spectral Partitioning algorithm, but to do so we must first define the Normalized Laplacian Matrix.

Definition 11.10 (Normalized Laplacian Matrix)**.** *Given a graph G, the normalized Laplacian matrix is defined as follows:*

$$\tilde{L} = (D^{-\frac{1}{2}} L D^{-\frac{1}{2}})_{i,j} = \frac{L_{i,j}}{\sqrt{d_i d_j}} = I - D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$$

Recall that D is the Degrees matrix, and $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ is the Normalized Adjacency matrix.

The Normalized Adjacency matrix also has the property that:

$$\text{eig val } (D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) = \text{eig val } (P)$$

Where P is the Random Walk matrix. Additionally, the first eigenvector of $\tilde{L}$ is $D^{1/2}1$.

With these definitions in place, we are now ready to describe the Spectral Partitioning Algorithm.

Algorithm 1 Spectral Partitioning Algorithm

Let x be the 2nd smallest eigenvector of $\tilde{L}$
Sort the vertices (rename them) such that:

$$\frac{x_1}{\sqrt{d_1}} \leq \frac{x_2}{\sqrt{d_2}} \leq \cdots \leq \frac{x_n}{\sqrt{d_n}}$$

for $i = 1 \to n$ **do**
  Let $S = \{1, \ldots, i\}$.
  Compute $\phi(S)$.
end for
Return the minimum conductance among all sets S in the loop where $\text{vol}(S) \leq \text{vol}(V)/2$.

The sweep across the partitions in the algorithm can be computed in $O(|E|)$, by incrementally computing the conductance of each partition by updating the previously computed conductances with the edges of the newest vertex.

The challenging part of this algorithm is finding the 2nd smallest eigenvector. The next lecture will show that approximating the 2nd eigenvector is sufficient, turning this into an approximate algorithm for finding the partitioning that runs in near linear time.

Note that everything we described so far naturally extends to weighted graphs, and the above algorithm can also be used to find a sparse cut of a weighted graph.

11.3.1 Cheeger's Inequality

The following theorem is one of the fundamental inequalities in spectral graph theory.

Theorem 11.11 (Cheeger's Inequality)**.** *For any graph G,*

$$\lambda_2/2 \le \phi(G) \le \sqrt{2\lambda_2}$$

where λ_2 is the 2nd smallest eigenvalue of $\tilde{L}$.

Cheeger's inequality relates the combinatorial property of conductance to a spectral property, the 2nd smallest eigenvalue. Observe that in the extreme case where $\lambda_2 = 0$, we also have $\phi(G) = 0$ and vice versa. We have already seen the proof of this in Claim 11.4 and Claim 11.5.

A crucial fact about the above inequality is that it does not depend on the size of G, n. It implies that if G has a "small" 2nd eigenvalue, then it is partitionable, whereas if the 2nd eigenvalue is "large" the graph is similar to a complete graph and it is not partitionable.

The proof of the right side of Cheeger's inequality, $\phi(G) \le \sqrt{2\lambda_2}$ is constructive, and it shows that the spectral partitioning algorithm always returns a set S such that $\text{vol}(S) \le \text{vol}(V)/2$ and

$$\phi(S) \le \sqrt{2\lambda_2} \le \sqrt{4\phi(G)}.$$

We now discuss several consequences of the above theorem for a special family of graphs.

Definition 11.12 (Expander Graphs)**.** *Expander graphs are sparse highly connected graphs with large 2nd eigenvalues, i.e., $\lambda_2 \ge \Omega(1)$. So, the can be seen as a sparse complete graphs which have $\lambda_2 = 1$. It turns out that most of the graphs are expanders, because a random d-regular graph satisfies $\lambda_2 \ge 1 - \frac{2}{\sqrt{d}}$*

Expander graphs are the easiest instances to use for many of the optimization problems (see PS4 for applications to the max cut problem). They are frequently used in coding theory (see PS4) and in psuedorandom number generators. In Problem Set 4, we will discuss the expander mixing lemma, which states that Expander Graphs are approximately the same as random graph. In words, in a d-regular expander graphs, for every disjoint large sets S, T, $|E(S,T)|$ is very close to the expected number of edges between S, T in a random $G(n, d/n)$ graph.

Definition 11.13 (Planar Graphs)**.** *A planar graph is one where all vertices can be projected onto a plane with no crossing edges.*

We know a lot about planar graphs. For example, we know that they tend to be sparse with average degrees at most 5.

It turns out that the 2nd eigenvalue of $\tilde{L}$ of any planar graph is at most $O(1/n)$.

Theorem 11.14. *If G is a bounded degree planar graph, the*

$$\lambda_2 \leq O(\frac{1}{n}).$$

Using the Cheeger's inequality, we can show that for every bounded degree planar graph G, $\phi(G) \leq O(1/\sqrt{n})$. In fact, by repeatedly peeling off sets of small conductance in G, we can show that every planar graph with bounded degree has a sparse bisection, i.e., a set $S \subseteq V$ such that $\text{vol}(V)/3 \leq \text{vol}(S) \leq 2\,\text{vol}(V)/3$ and $\phi(S) \leq O(1/\sqrt{n})$. This means that it is very easy to break a bounded degree planar graph into two sets such that there is very small number of connections between the two. This makes planar graphs ideal candidate for divide and conquer algorithms. We can recursively solve our problem on the two sides $S, \overline{S}$ and then merge the solutions. Since there are only $O(\sqrt{n})$ edges between $S, \overline{S}$, the merge operation can be done very efficiently

Finally, we also note that we can generalize the task of partitioning a graph into two sets, into partitioning a graph into k sets. Doing so, we define the k-way conductance of a graph to be:

Definition 11.15 (k-way conductance)**.** *For an integer $k > 1$ and a graph G, let*

$$\phi_k(G) = \min_{disjoint S_1,\ldots,S_k} \max_{1 \leq i \leq k} \phi(S)$$

where the min is over all k disjoint sets $S_1, \ldots, S_k$ in G.

In other words, we are interested in finding k disjoint sets such that their maximum conductance is as small as possible. With this definition, $\phi(G)$ corresponds to the case $k = 2$.

It turns out that there is a natural generalization of Cheeger's inequality for larger values of k.

Theorem 11.16. *For any graph G and an integer $k > 2$,*

$$\begin{aligned} \lambda_k/2 \leq \phi_k(G) &\leq O(\sqrt{\lambda_k} \cdot k^2) \\ \phi_k(G) &\leq O(\sqrt{\log k \cdot \lambda_{2k}}). \end{aligned}$$

Note that the second inequality is stronger than the first one only when λ_{2k} is not much larger than λ_k. Similar to Cheeger's inequality, the proof of the right side of this inequality is constructive and provides an algorithm to k disjoint sets with small conductance.