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

Lecture 13: Spectral Graph Theory

Lecturer: Shayan Oveis Gharan *11/14/18*

13.1 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 13.1.

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

The theorem shows a very nice connection between a combinatorial property of a graph and an algebraic property of its matrices. Namely, we can test whether a given graph is connected without running any graph search algorithm, include BFS or DFS. All we need to do is to test if the second eigenvalue of Laplacian matrix is nonzero. In the future lectures we see efficient algorithms to approximately compute the second eigenvalue of the Laplacian matrix.

Theorem 13.2. *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 13.3. *For any graph G,*

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

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

This theorem specifies that the determinant of the Laplacian of G with $\mathbf{1}/n$ 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.

13.1.1 Laplacian Matrix

Definition 13.4 (Laplacian Matrix)**.** *Given $G = (V, E)$ a weighted graph G, where every edge (i, j) has weight $w_{i,j}$. Consider the weighted adjacency matrix where for every edge (i, j)*

$$A_{i,j} = A_{j,i} = w_{i,j},$$

and the rest of the entries are zero. Also, let D be the diagonal matrix of vertex (weighted) degrees; that is for each vertex i,

$$D_{i,i} = d_w(i) = \sum_{j \sim i} w_{i,j}$$

The Laplacian matrix of G

$$L_G = D - A.$$

In other words, it has degrees on the diagonal and the off-diagonal entries $L_{i,j}$ is $-w_{i,j}$ if there is an edge and 0 otherwise.

For unweighted graph G, the quadratic form of L_G is

$$\begin{aligned} x^T L_G x &= x^T (D - A) x \\ &= \sum_i d(i) x_i^2 - \sum_{i \sim j} x_i x_j \\ &= \sum_i d(i) x_i^2 - 2 \sum_{i \sim j : i < j} x_i x_j = \sum_{i,j \in G} (x_i - x_j)^2 \end{aligned}$$

Similarly for a weighted graph:

$$x^T L_G x = \sum_{i,j \in G} w_{i,j} (x_i - x_j)^2 \geq 0 \tag{13.1}$$

Then the following fact is immediate.

Fact 13.5. *For any (weighted) graph G, L_G is a PSD matrix, $L_G \succeq 0$*

Proof. By variational characterization of eigenvalues (see Lecture 8, section 2), L_G's minimum eigenvalue is

$$\lambda_{min}(L_G) = \min_x \frac{x^T L_G x}{x^T x} = \min_x \frac{\sum_{i \sim j} (x_i - x_j)^2}{x^T x} \geq 0.$$

where we used that the numerator is always nonnegative. □

Fact 13.6. $\lambda_{min}(L_G) = 0$

Proof. Let $x = \vec{1}$ be the all ones vectors. Then,

$$\sum_{i \sim j} (x_i - x_j)^2 = \sum_{i \sim j} (1 - 1)^2 = 0.$$

Since by Fact 13.5, $y^T L_G y \geq 0$ for any vector y, by variational characterization of eigenvalues, the smallest eigenvalue of L_G is zero. □

In the rest of this section we prove Theorem 13.1. The following two claims imply what we need.

Claim 13.7. *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 $x^T L x = 0$, 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 13.8. *$\lambda_2 = 0 \implies$ G is disconnected*

Proof. First, by the quadratic form,

$$\lambda_2 = 0 \implies \exists x \perp \mathbf{1} \; s.t.$$
$$x^T L x = \sum_{i \sim j} (x_i - x_j)^2 = 0 \iff (x_i - x_j)^2 = 0 \; \forall i \sim j$$

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. □

13.2 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.

13.2.1 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 13.9 (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 13.10 (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 13.11 (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.

13.2.2 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 13.12.

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

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.

13.3 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 13.13 (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.

13.3.1 Cheeger's Inequality

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

Theorem 13.14 (Cheeger's Inequality)**.** *For any graph G,*

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

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 13.7 and Claim 13.8.

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, $\phi(G) \leq \sqrt{2\lambda_2}$ is constructive, and it shows that the spectral partitioning algorithm always returns a set S such that $\text{vol}(S) \leq \text{vol}(V)/2$ and

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

Both sides of Cheeger's inequality are tight. The right side is tight for a cycle and the left side is tight for the hypercube. We will talk more about Cheeger's inequality in the next lecture.