Spectral Clustering

PPT by Brandon Fain

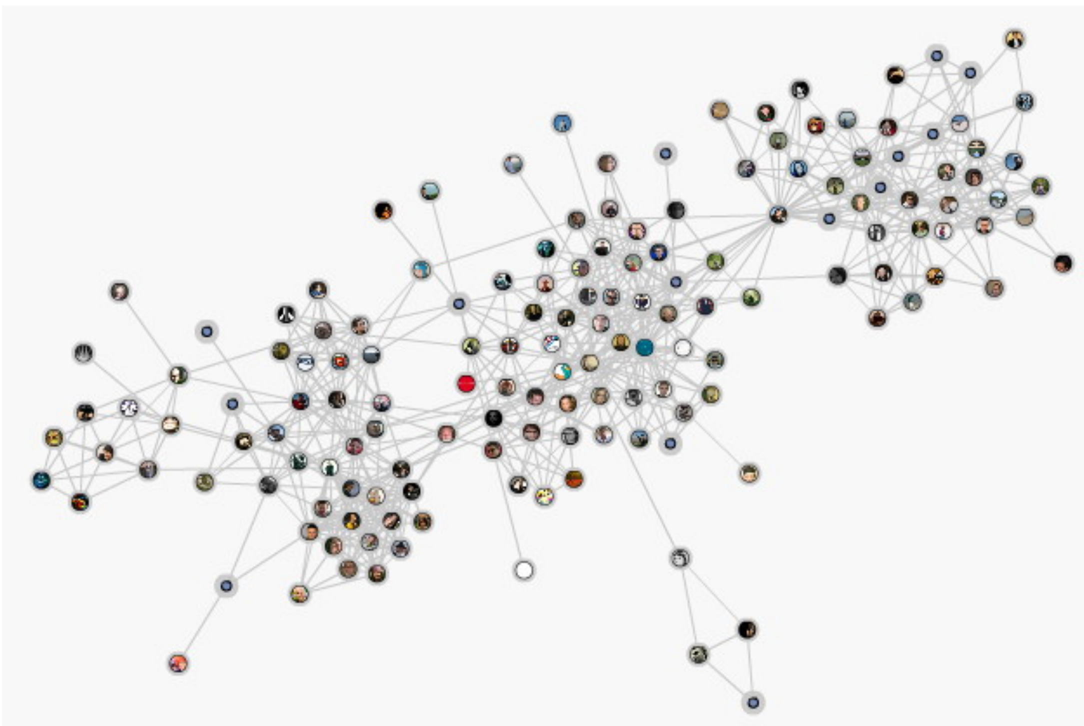
Outline

- Review
  - Community Detection Problem
  - Conductance
  - Graph Laplacian

- Spectral techniques to find low conductance cuts

- Spectral techniques for clustering and community detection

Motivating Problem: Community Detection



Given a social network, how do you find the strongly connected communities?

Corollary question: How would you suggest friends to a user?

Conductance

- Let $G = (V, E)$ be an undirected graph.
- $S \subseteq V$ denote a cut in the graph.
- Let $\delta(S) \coloneqq |\{(u, v) \in E : u \in S, v \notin S\}|$.
- Let $Vol(S) = \sum_{i \in S} d_i$, where d_i is the degree of node i.
- The **conductance** of S is

$$\phi(S) = \frac{\delta(S)}{\min(Vol(S), Vol(V - S))}.$$

- We want to find a low conductance cut: one with many more internal edges than cut edges.

Laplacian Matrix

- The **graph Laplacian** is defined as

$$L = D - A$$

where D is the diagonal matrix with $D_{ii} = d_i$ and $D_{ij} = 0$ for $i \neq j$, and A is the adjacency matrix.

- Recall $(Lv)_i = \sum_{j:(i,j)\in E} v_i - v_j$.
- Last time, we observed that the orthogonal eigenvectors corresponding to eigenvalues of 0 told us the connected components of the graph.
- Our intuition was that the eigenvectors for the smallest non-zero eigenvalues should tell us something about low conductance cuts.

Outline

- ~~Review~~
  - ~~Community Detection Problem~~
  - ~~Conductance~~
  - ~~Graph Laplacian~~

- Spectral techniques to find low conductance cuts

- Spectral techniques for clustering and community detection

Spectral Algorithm for Low Conductance Cut

- Let λ be the smallest *non-zero* eigenvalue of the graph Laplacian L with corresponding eigenvector $\vec{v}$.
- Sort the vertices i in non-decreasing order of v_i. For notational convenience, say that after sorting: $v_1 \leq v_2 \leq \cdots \leq v_n$.
- For i from 1 to n-1:
  - $S_i \leftarrow \{1, 2, \dots, i\}$
  - $C_i \leftarrow \phi(S_i)$
- Return S_i with minimum C_i.

Spectral Algorithm Analysis

- **Efficiency**. Note that the brute force algorithm for the problem considers 2^n cuts.
- Clearly, this algorithm considers $O(n)$ cuts.
- You need to calculate conductance at each step (potentially an $\Omega(m)$ calculation). Can you see how to avoid this?

- **Accuracy**. How "correct" is the algorithm?
- This is an NP-Complete problem, so this won't solve it exactly (i.e., no guarantee of minimum conductance cut). How close do we get?

Spectral Algorithm Analysis

- We will analyze the case of a **d-regular graph**, that is, one for which every vertex has degree exactly d.
  - (This makes the statement and proof easier, but a similar statement holds for non-regular graphs).
- Then the conductance can be rewritten as

$$\phi(S) = \frac{\delta(S)}{\mathrm{d} \cdot \min(|S|, |V - S|)}.$$

Suppose w.l.o.g. we just consider cuts where $|S| \leq |V - S|$. Define

$$\theta(S) = \frac{\delta(S)}{|S|}.$$

Then the minimum conductance cut of a graph also minimizes $\theta(S)$. Call this minimum $\theta(S)$ value Θ_G for a graph G.

Spectral Algorithm Analysis

- **Theorem (Cheeger's Inequality).** Let G be a d-regular connected graph with minimum conductance $\frac{\Theta_G}{d}$. Let *S* be the cut found by our spectral algorithm. Let λ_2 be the second smallest eigenvalue of the graph Laplacian of G. Then

$$\frac{\lambda_2}{2} \leq \Theta_G \leq \theta(S) \leq \sqrt{2d\lambda_2}.$$

- **Corrolary.**

$$\frac{\theta(S)}{\Theta_G} \leq \frac{\sqrt{2d\lambda_2}}{\Theta_G} \leq \frac{2\sqrt{2d}}{\sqrt{\lambda_2}}$$

This is a fairly pessimistic bound on typical performance in practice.

Spectral Algorithm Analysis

- Proving $\theta(S) \leq \sqrt{2d\lambda_2}$ is difficult, and we don't have the time.
- Proving $\frac{\lambda_2}{2} \leq \Theta_G \leq \theta(S)$ is relatively easy.
- Note that $\Theta_G \leq \theta(S)$ is by definition, so we really only need to prove $\frac{\lambda_2}{2} \leq \Theta_G$.
- **Proof Sketch.** Recall that for any eigenvector v with eigenvalue λ, $Lv = \lambda v$**.** Therefore

$$\frac{v^T L\, v}{v^T v} = \frac{v^T(\lambda\, v)}{v^T v} = \lambda$$

- **Proof Sketch** (continued). We have already seen that for a connected graph, the all 1 vector is an eigenvector for eigenvalue 0.
- The second smallest eigenvalue λ_2 has an eigenvector that is orthogonal to this all 1 vector, and in particular:
$$\lambda_2 = min_{v:v\cdot\vec{1}=0} \ \frac{v^T L\, v}{v^T v}.$$
- Consider a cut *S*, and define the vector $v_i = 1 - |S|/|V|$ for $i \in S$ and $-\ |S|/|V|$ otherwise. For every *S*, this vector is orthogonal to $\vec{1}$.
- Furthermore, if you work out the algebra,
$$\frac{v^T L\, v}{v^T v} = \frac{\delta(S)}{|S|\cdot|V-S|/|V|} = \frac{\delta(S)}{|S|\cdot(1-|S|/|V|)} \le 2\,\frac{\delta(S)}{|S|}$$
- Then λ_2 is at most $2\delta(S)/|S|$.
- Since this holds for any cut, it holds for the minimum cut, so $\lambda_2 \le 2\Theta_{\mathrm{G}}$.

Outline

- ~~Review~~
  - ~~Community Detection Problem~~
  - ~~Conductance~~
  - ~~Graph Laplacian~~

- ~~Spectral techniques to find low conductance cuts~~

- Spectral techniques for clustering and community detection

Further Questions

- What if you want to partition your data into more than 2 clusters?
- What if you want to detect the community of an individual, rather than just a good community globally in the graph?
- What if your data isn't actually a graph to begin with?
- We will conclude with some heuristic spectral approaches for these problems.

More Than 2 Clusters

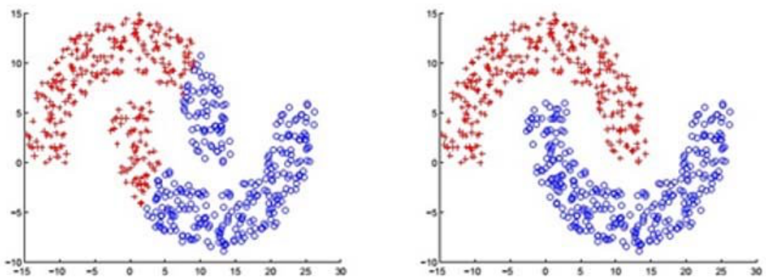
- Suppose we want to partition the data into k clusters. A common approach is as follows:
- Represent each vertex i as a length *m* vector, where:
  - The j'th component of the vector is the i'th entry in the eigenvector of the graph Laplacian corresponding to the j+1 smallest eigenvalue.
  - For example, suppose we set *m*=2, and <5, -1, 3, -2> is the eigenvector corresponding to the 2^{nd} smallest eigenvalue, and <-1, 5, 0, 0> is the eigenvector corresponding to the 3^{rd} smallest eigenvalue.
  - Then we would represent the first vertex as <5, -1>, the second as <-1, 5>, the third as <3, 0> and the fourth as <-2, 0>.
- Now, run a standard clustering algorithm (e.g., k-means) on these vectors.

Community Detection

- Suppose we have an individual i, and we know that she belongs to a community with between n_1 and n_2 individuals. We want to predict who those individuals are.
- One heuristic is as follows:
  - Represent each individual as a vector according to the eigenvectors corresponding to small (but non-zero) eigenvalues, exactly as in the last slide.
  - Let d(x,y) be a distance function on these vectors (e.g., standard Euclidean distance).
  - For n from n_1 to n_2:
    - Let S_i be the n individuals with minimum distance to i.
    - $C_i \leftarrow \phi(S_i)$
  - Return the S_i with minimum C_i

Non Graphical Data

- What if your data wasn't a graph to begin with? For example, if you wanted to cluster something like:



(a) K-means (b) Spectral Clustering

- Just create a graph by setting points that are sufficiently close to one another to be adjacent vertices.
- Then run your favorite spectral analysis.

Summary

- There are deep connections between the eigenvalues and eigenvectors of the graph Laplacian and the connectivity properties of a graph.

- For clustering problems where you care about *connectivity*, spectral clustering, exploiting these properties, is the standard approach.

- It is useful for minimum conductance cuts and community detection problems on graphs, but it can also be applied to non-graphical data.

- In your last lab homework, you will play around with spectral techniques on an email graph.