Algebraic Connectivity and Spectral Clustering

Mattie Ji

APMA 2812G - Combinatorial Theory

December 6th, 2022

Note

Throughout this talk, when we say **"graph"**, we are always considering simple undirected graphs with more than 1 node.

Notion of Connectivity

Throughout this class, we have introduced many notions of connectivity on a graph G:

- $\delta(G)$ - the minimum degree of a graph
- $v(G)$ - the vertex connectivity of a graph
- $e(G)$ - the edge connectivity of a graph

In this talk, we will introduce a new notion of connectivity.

Algebraic Connectivity

Algebraic Connectivity

Let G be an unweighted graph, then the **algebraic connectivity** $a(G)$ of G is the second smallest eigenvalue of the graph Laplacian L of G (including multiplicity).

This is also called the "Fiedler value" of the graph.

Relation to Connectivity

Why is $a(G)$ called the **algebraic connectivity** of G?

Prop:

$a(G) \geq 0$, with the equality achieved if and only if G is disconnected.

The non-negativity of $a(G)$ follows from that fact that L is positive semi-definite.

Prop:

Let G_1 and G_2 be two graphs with the same vertex sets and $E(G_1) \subset E(G_2)$, then

$$a(G_1) \leq a(G_2)$$

Relation to other Connectivity

Theorem:

Let G be an unweighted non-complete graph, then we have the following inequality:

$$0 \leq a(G) \leq v(G) \leq e(G) \leq \delta(G)$$

Interestingly enough, when $G = K_n$:

$$v(G) = e(G) = n - 1 < n = a(G)$$

Why $a(G)$?

Why should we care about algebraic connectivity? What separates it from all the other notions of connectivity we care about?

One reason is that the idea of algebraic connectivity is really simple to generalize for weighted graphs:

Definition:

Let G be a weighted graph, the algebraic connectivity $a(G)$ of G is the second smallest eigenvalue of the graph Laplacian L.

Another reason is that - it turns out algebraic connectivity becomes really relevant in the realms of certain clustering problems.

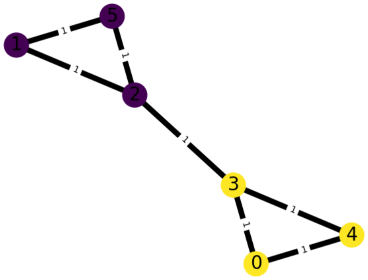
Definition:

Let $G = (V, E)$ be a weighted graph, and $S \subseteq V$, then the volume on S is

$$\text{vol}(S) = \sum_{v \in S} deg(v)$$

The Cut Ratio

Let G be a graph with adjacency matrix A, let C_1, C_2 be a partitioning of its vertices



We define the **cut-ratio** of C_1, C_2 as

$$[\sum_{v_1 \in C_1, v_2 \in C_2} A_{v_1, v_2}] \cdot (\frac{1}{vol(C_1)} + \frac{1}{vol(C_2)})$$

Question:

How can we find a partitioning of $V(G)$ such that the cut-ratio is minimized.

Trying to resolve this is actually NP-Hard, but we can find an approximate solution using the algebraic connectivity of G!

The Algorithm

Data: The graph G with vertex number $1, 2, ..., n$

Result: Clusters C_1, C_2 partitioning vertices of the graph

1. Compute the graph Laplacian L of G with ordering of vertices as $1, 2, ..., n$;
2. Find an eigenvector v of $a(G)$;
3. Let v_i be the i-th component of v, then the clusters are $C_1 = \{i : v_i \geq 0\}$ and $C_2 = C_1^c$

Algorithm 1: Spectral Clustering

Theorem:

If G is a connected graph, then the sub-graph generated by C_1 and C_2 with respect to G are both connected.

Consider the following graph:

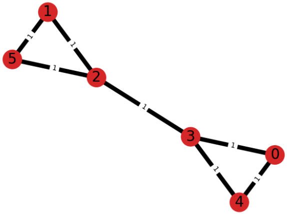


Figure: Sample Graph

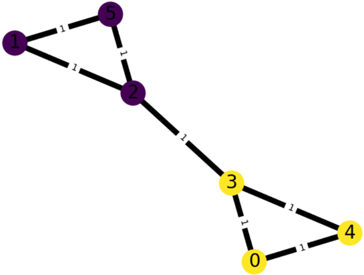


Figure: After Running Spectral Clustering

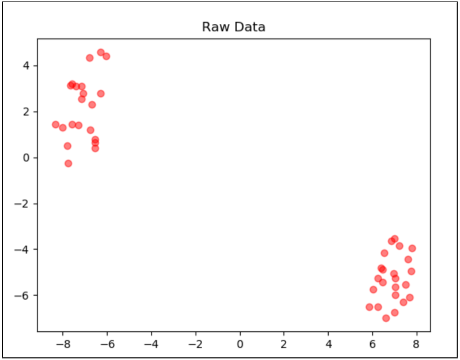
Spectral Clustering on Data Points

In practice, we aren't always going to be handed with a graph. Here's a common scenario

Question:

Given a list of data-points $p_1, ..., p_n \in \mathbb{R}^d$, how can we partition them in 2 different clusters?

For example:



Epsilon Neighborhoods

The idea here is to create a graph with vertices being the points listed. However, in general there's no one **correct** way to make these graphs, as it is largely dependent on the domain of the data.

There're a few common ways to create the graph:

Epsilon Neighborhoods:

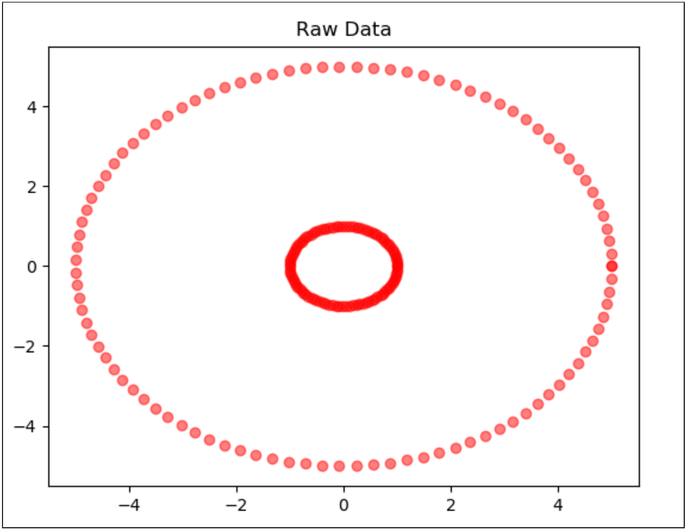
Let $\epsilon > 0$, for all $i \neq j$, we say that there's an edge of weight 1

$$p_i \to p_j \iff d(p_i, p_j) < \epsilon$$

This will create an undirected graph of uniform weight.

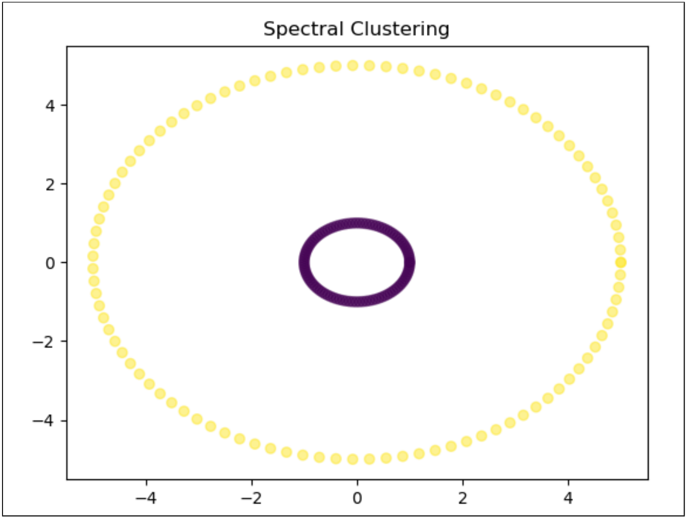
Examples of Clustering

Here's an example of spectral clustering using the **Epsilon Neighborhoods** with $\epsilon = 1$



Examples of Clustering

Here's an example of spectral clustering using the **Epsilon Neighborhoods** with $\epsilon = 1$



Another idea is to create a fully connected graph and assign weights based on how far they are:

Similarity Functions

A symmetric function $s : \mathbb{R}^d \times \mathbb{R}^d \to [0, 1]$ is a **similarity function**. The idea is that $s(x, y) = 1$ means x and y are really close, and $s(x, y) = 0$ means x and y are really far away.

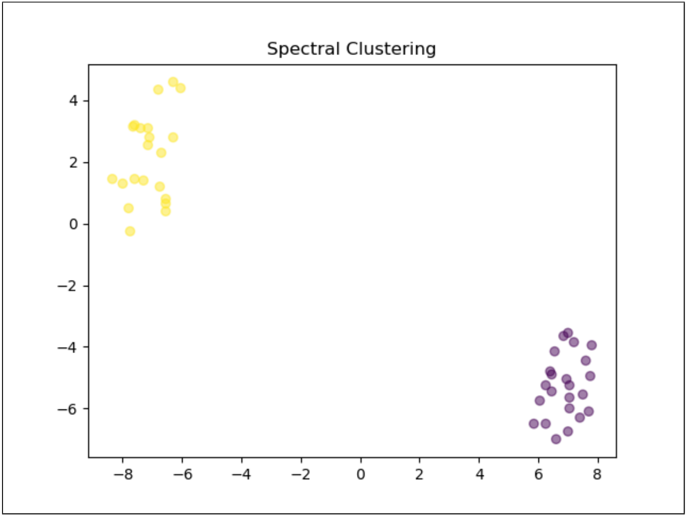
The idea is then, for all $i \neq j$, we assign an edge $p_i \to p_j$ of weight $s(p_i, p_j)$.

A common choice of similarity function is the **Gaussian similarity**:

$$s(x, y) = \text{exp}(\frac{-||x - y||_2^2}{2\sigma^2})$$

Examples of Clustering

Here's an example on our previous example using the with $\sigma = 1$



Further Questions

Question:

What if we want to split into k clusters rather than just 2?

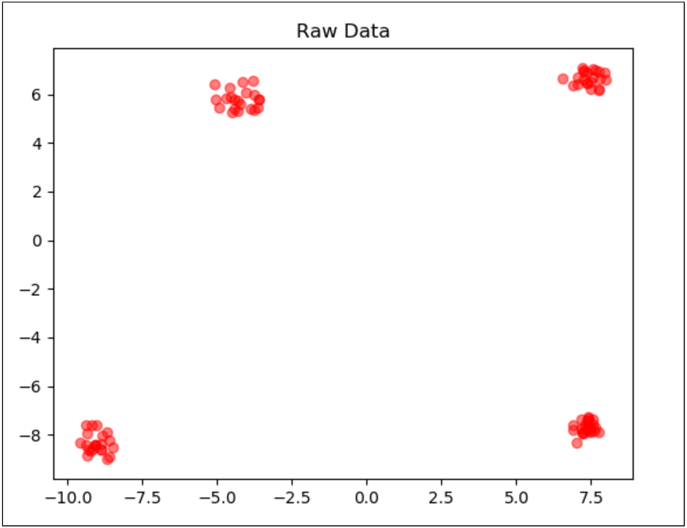
This is totally possible, and the process involves finding the k smallest eigenvalues of L and their respective eigenvectors, but it is beyond the scope of this talk.

If you are curious, the details of making k clusters are available on our GitHub:

- https://github.com/maroon-scorch/Algebraic-Connectivity

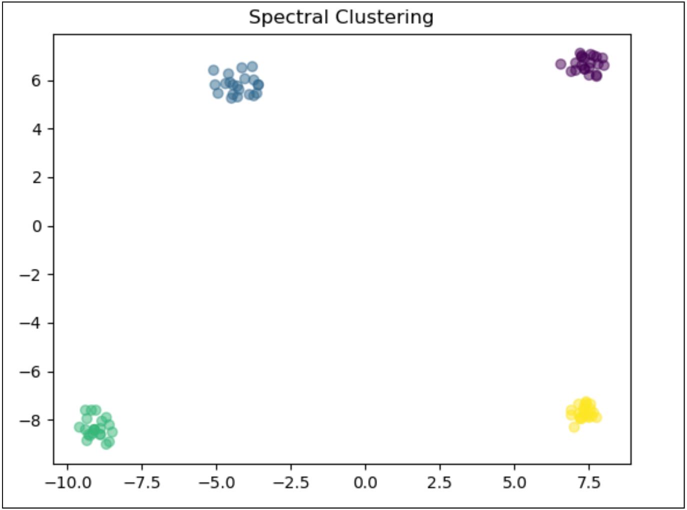
Examples of Clustering

Here's an example of spectral clustering using the with $\sigma = 1$ and $K = 4$



Examples of Clustering

Here's an example of spectral clustering using the with $\sigma = 1$ and $K = 4$



Acknowledgements

1 Fiedler, Miroslav. "Algebraic connectivity of graphs." Czechoslovak mathematical journal 23.2 (1973): 298-305.

2 Fiedler, Miroslav. "Laplacian of graphs and algebraic connectivity." Banach Center Publications 25.1 (1989): 57-70.

3 Strnadová-Neeley, Veronika. "Spectral Clustering" Seminar on Top Algorithms in Computational Science, 2010. https://sites.cs.ucsb.edu/~veronika/SpectralClustering.pdf

All figures and graphs in this presentation were generated by the code repository before.