Graph Spectral Clustering

Catherine Huang, Tynan Sigg, Clark Wang, Jonathan Wang

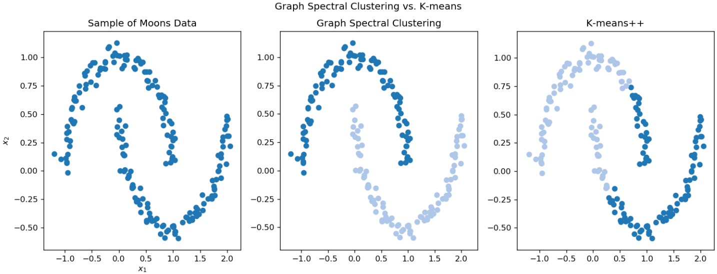
Lecture Slides

February 22, 2021

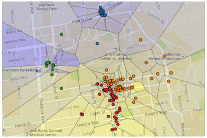
Motivating Questions

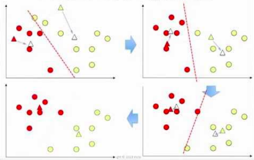
- Why does **graph spectral clustering** work on the Two Moons data set, while canonical **k-means** fails?
- What is the graph spectral clustering algorithm?
  - How do we go from data to a **similarity graph**?
  - What is the **Laplacian** of a graph, and how do we find it?

Two Moons Dataset



K-means

- First seen in CS 61A - Bear Maps
  - 

- **Centroid-based** clustering algorithm
  - 

K-means

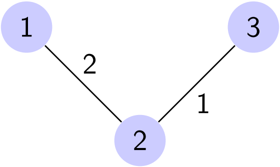
- Pseudocode:
  - Initialize k points $c_i \in \mathbb{R}^d$ with $i = 1, \ldots, k$ to be the centroids of each cluster.
  - Assign each data point to the closest centroid. The points that correspond to the same centroid form the k clusters.
  - Re-assign each centroid to be the center of its current cluster (using the updated assignments of data).
  - Continue to iterate steps 2 and 3 until updating the centroids in step 3 no longer changes the cluster assignments in step 2.

K-means++

- Vanilla k-means isn't perfect
- Solution: better way to initialize centroids that are more spread apart (**k-means++**)
- Pseudocode:
  - Choose one centroid c_1, chosen uniformly at random from X.
  - Choose a new centroid c_i, chosen with new probability $\frac{D(x)^2}{\sum_{x \in X} D(x)^2}$, where $D(x)$ is the shortest distance from data point x to its closest centroid.
  - Repeat the previous step, until k centroids have been chosen.
  - Continue with the original k-means algorithm.

Graph Terminology

- **Graph** - defined by a set of **vertices** and a set of **edges**, where each edge connects exactly two vertices
- **Weighted Graph** - in weighted graphs, each edge is associated with a weight $w : E \to \mathbb{R}$
- **Undirected Graph** - the weights of edges in both directions are equal



Matrix Terminology

- **Degree Matrix** - Degree of a vertex $v_i \in V$ is $d_i = \sum_{j=1}^{n} w_{ij}$, the sum of the weights of all edges incident to vertex v_i. The diagonal entries of degree matrix D are degrees $d_1, ..., d_n$.
- **Weighted Adjacency Matrix** - The adjacency matrix of a weighted graph $G = (V, E)$ is the matrix $W = [w_{ij}]_{i,j \in V}$, where each w_{ij} is the weight of edge (i, j).
    - Sometimes referred to as the **similarity matrix** when working over a similarity graph
- **Laplacian** - $L_s := D - W$
- **Normalized Laplacian** - $L_n := I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} = D^{-\frac{1}{2}} L_s D^{-\frac{1}{2}}$

Desired Properties in a Similarity Metric

- Datapoints that are more similar have a higher value
- Datapoints that are less similar have a lower value

What functions accomplish this? How do we go from having some data to a similarity graph?

Going from Data to Similarity Graph

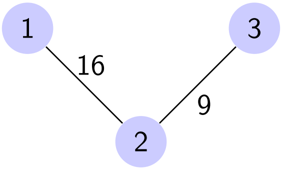
Examples of **similarity kernels**:

- **Gaussian kernel (RBF)**
  - $s(x_i, x_j) = \exp(-||x_i - x_j||^2/2\sigma^2)$
- **Exponential kernel**
  - $s(x_i, x_j) = \exp(-||x_i - x_j||/\sigma)$

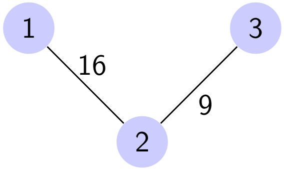
Similarity graph: each edge weight represents the similarity between pairs of vertices calculated from one of these similarity kernels.

Normalized Spectral Clustering: Walkthrough

Suppose we're given this similarity graph from our data of size 3.



Computing the Laplacian



$$D = \begin{bmatrix} 16 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 9 \end{bmatrix} \quad W = \begin{bmatrix} 0 & 16 & 0 \\ 16 & 0 & 9 \\ 0 & 9 & 0 \end{bmatrix}$$

$$L := D - W = \begin{bmatrix} 16 & -16 & 0 \\ -16 & 25 & -9 \\ 0 & -9 & 9 \end{bmatrix}$$

Computing the Normalized Laplacian

$$D = \begin{bmatrix} 16 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 9 \end{bmatrix} \qquad D^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{5} & 0 \\ 0 & 0 & \frac{1}{3} \end{bmatrix}$$

$$\begin{aligned} L_n &= D^{-\frac{1}{2}} L D^{-\frac{1}{2}} \\ &= D^{-\frac{1}{2}} \begin{bmatrix} 16 & -16 & 0 \\ -16 & 25 & -9 \\ 0 & -9 & 9 \end{bmatrix} D^{-\frac{1}{2}} \\ &= \begin{bmatrix} 1 & -\frac{4}{5} & 0 \\ -\frac{4}{5} & 1 & -\frac{3}{5} \\ 0 & -\frac{3}{5} & 1 \end{bmatrix} \end{aligned}$$

Computing the H matrix

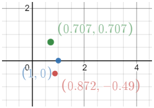
$$L_n = QDQ^{-1} = \begin{bmatrix} \frac{4}{3} & -\frac{3}{4} & \frac{4}{3} \\ \frac{5}{3} & 0 & -\frac{5}{3} \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} \frac{4}{3} & -\frac{3}{4} & \frac{4}{3} \\ \frac{5}{3} & 0 & -\frac{5}{3} \\ 1 & 1 & 1 \end{bmatrix}^{-1}$$

$$H_{temp} = \begin{bmatrix} | & | \\ v_1 & v_2 \\ | & | \end{bmatrix} = \begin{bmatrix} \frac{4}{3} & -\frac{3}{4} \\ \frac{5}{3} & 0 \\ 1 & 1 \end{bmatrix}$$

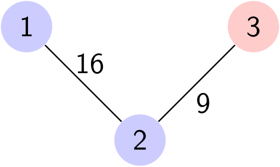
Normalize $H_{temp} \implies H = \begin{bmatrix} \frac{16}{\sqrt{337}} & -\frac{9}{\sqrt{337}} \\ 1 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$

Final Result

Perform k-means on the rows of $H = \begin{bmatrix} \frac{16}{\sqrt{337}} & -\frac{9}{\sqrt{337}} \\ 1 & 0 \\ \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix}$



Resulting clusters:

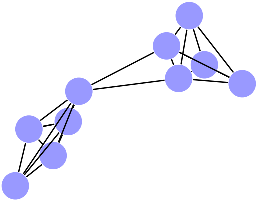


Normalized Spectral Clustering: Complete Algorithm

- Pseudocode
  - Construct a similarity graph $S \in \mathbb{R}^{n \times n}$ according to some similarity metric (e.g. the Gaussian or exponential kernels)
  - Compute $L_n = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$, the symmetric normalized graph Laplacian of S.
  - Compute $v_{1 \dots k}$, the eigenvectors of L_n corresponding to its k smallest eigenvalues.
  - Construct the matrix $H = [v_1; \dots; v_k] \in \mathbb{R}^{n \times k}$ having the first k eigenvectors of L_n as the columns.
  - Run k-means on the rows of H to cluster the data. Each data point is assigned to the cluster of its corresponding row in H.

Recap of Normalized Spectral Clustering

Suppose we're given a similarity graph or we created a similarity graph from our data.

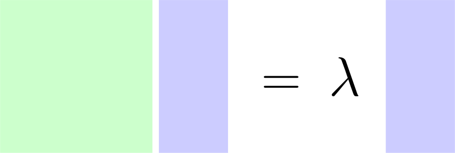


Recap of Normalized Spectral Clustering

From the similarity graph, we can construct the normalized Laplacian

Recap of Normalized Spectral Clustering

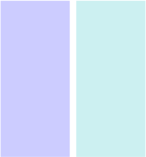
We take the eigenvectors corresponding to the smallest *k* eigenvalues of the Laplacian, where *k* corresponds to the number of clusters.



The eigenvector is purple.

Recap of Normalized Spectral Clustering

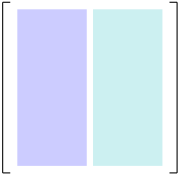
We specifically want the k eigenvectors corresponding to the k smallest eigenvalues



Recap of Normalized Spectral Clustering

We then concatenate our eigenvectors to form the matrix H_{temp}. Then normalize the rows to have norm 1 to get the matrix H.

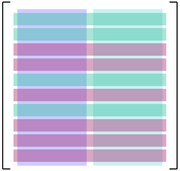
$$H =$$



Recap of Normalized Spectral Clustering

Running k-means on the rows of H, we form our clusters.

$$H =$$



Recap of Normalized Spectral Clustering

Since each row of the matrix H corresponds to a data point, we get our clusters!