

Computational Data Analysis

Machine Learning

Yao Xie, Ph.D.

Associate Professor

Harold R. and Mary Anne Nash Early Career Professor
H. Milton Stewart School of Industrial and Systems
Engineering

Spectral Clustering



General formulation of clustering

- Given m data points, $\{x^1, x^2, \dots, x^m\} \in R^n$
- Find k cluster centers, $\{c^1, c^2, \dots, c^k\} \in R^n$
- And assign each data point i to one cluster, $\pi(i) \in \{1, \dots, k\}$
- Such that the sum of the squared distances from each data point to its respective cluster center is minimized

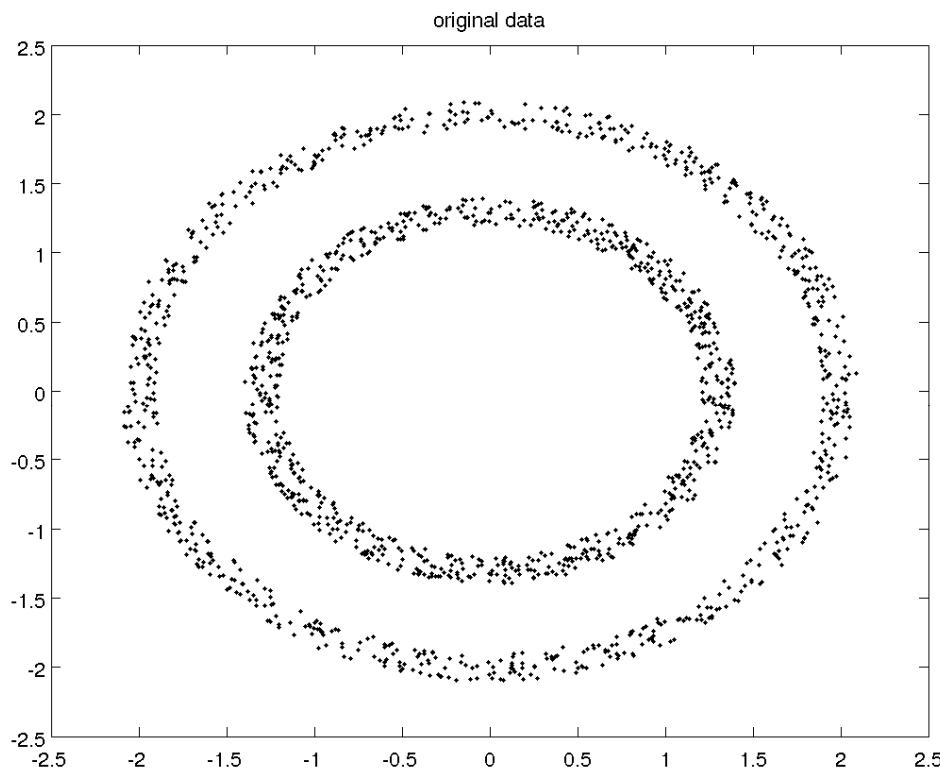
$$\min_{c,\pi} \sum_{i=1}^m d(x^i, c^{\pi(i)})^2$$



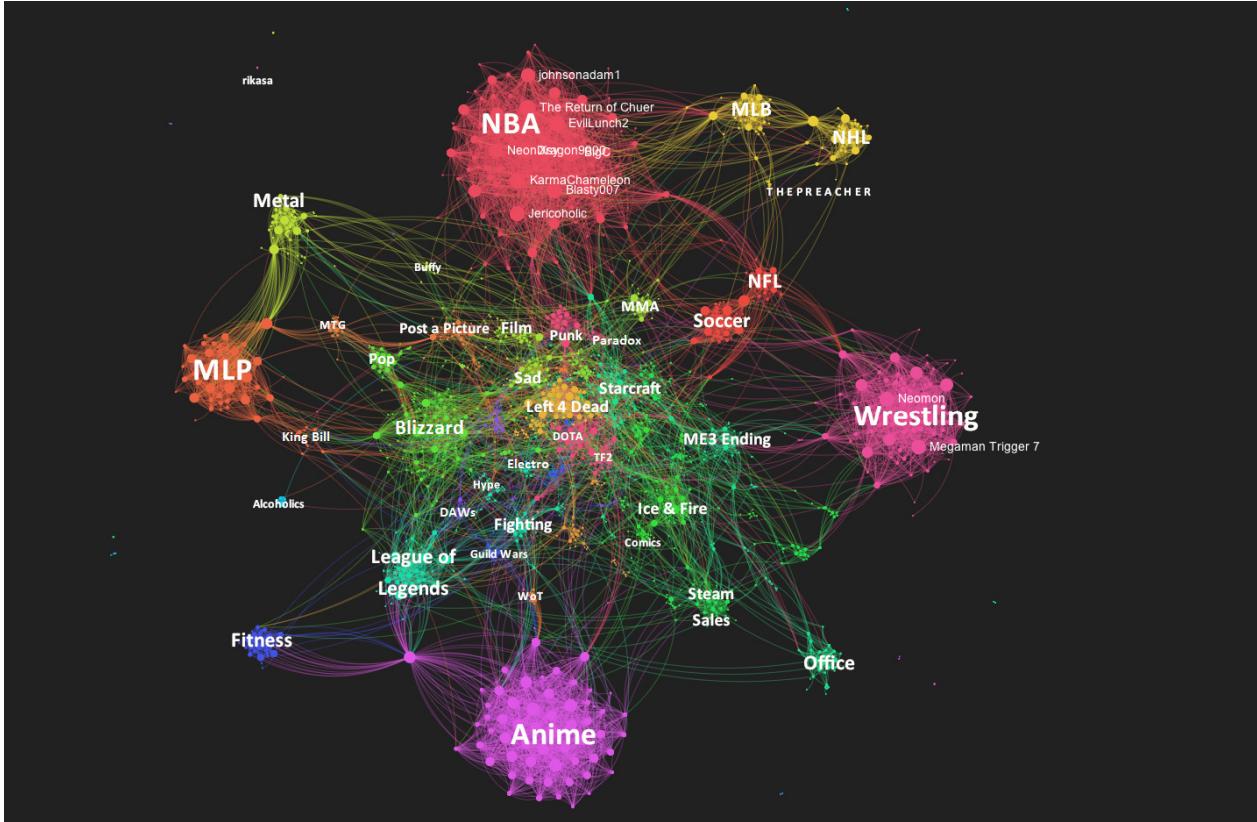
K-means algorithm

- Initialize k cluster centers, $\{c^1, c^2, \dots, c^k\}$, randomly
- Do
 - Decide the cluster memberships of each data point, x^i , by assigning it to the nearest cluster center (**cluster assignment**)
$$\pi(i) = \operatorname{argmin}_{j=1,\dots,k} \|x^i - c^j\|^2$$
 - Adjust the cluster centers (**center adjustment**)
$$c^j = \frac{1}{|\{i: \pi(i) = j\}|} \sum_{i: \pi(i)=j} x^i$$
- While any cluster center has been changed

How about this dataset? (Run `test_tworings.m`)



How about clustering nodes in social networks

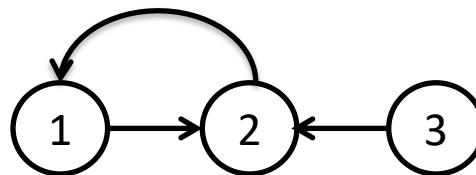
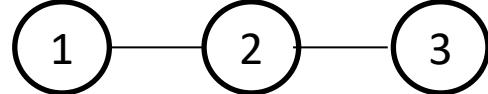


Representing graph using matrices

- Adjacency matrix
- Unweighted graph (either directed or undirected)

$$A_{ij} = \begin{cases} 1, & \text{if there is an edge between } i \text{ and } j \\ 0, & \text{otherwise.} \end{cases}$$

- E.g. in both cases, the adjacency matrix is



$$A = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

- Graph Laplacian L is positive semi-definite (meaning all the eigenvalues are non-negative).

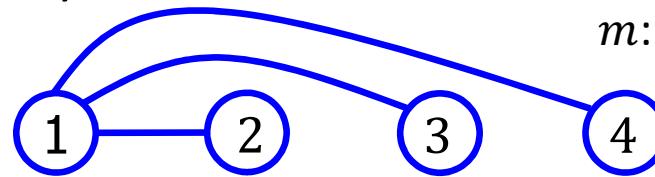
Spectral clustering algorithm

- Step 1: represent graph as adjacency matrix $A \in R^{m \times m}$



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$



$$A = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

m : number of nodes

- Step 2: form a special matrix $L = D - A$, the graph Laplacian

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

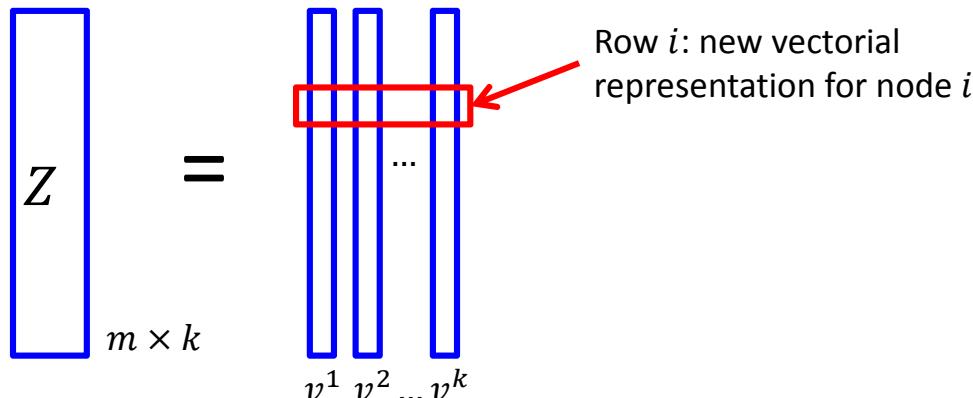
$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

Spectral clustering algorithm (cont.)

- Step 3: compute k eigenvectors, v^1, v^2, \dots, v^k , of L corresponding to the k **smallest** eigenvalues ($k \ll m$)

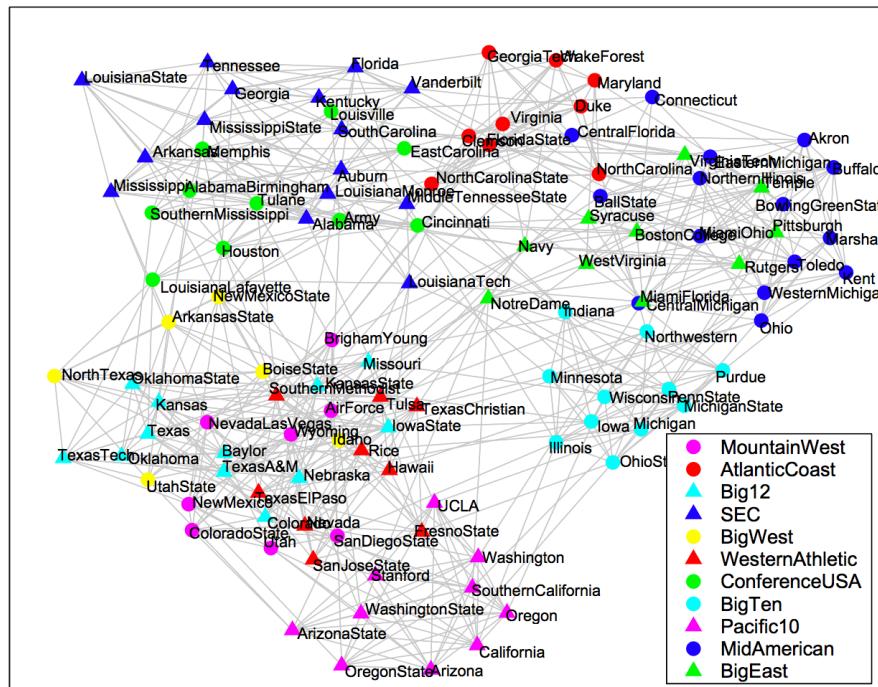
$$Lv^1 = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} v^1 \stackrel{?}{=} \lambda_1 v^1$$

- Step 4: run kmeans algorithm on $Z = (v^1, v^2, \dots, v^k)$ by treating each row as a new data point



Run demo test_football.m

The image shows the top portion of a college football website. At the top, there's a navigation bar with links like 'PLAY FANTASY', 'The Most Award Winning Fantasy game with real time scoring, top expert analysis, custom settings, and more.', and 'PLAY NOW'. Below the navigation is a 'NCAA FB Home' dropdown menu. The main content area features a banner for the Mayweather vs. Maidana fight, with text indicating it's a pay-per-view event on Saturday, September 13, at 8pm/5pm ET, with options to buy tickets or get more info. Below the banner is a section titled 'COLLEGE FOOTBALL SCHEDULES' with a 'FBS FCS' dropdown. A navigation bar below that includes links for weeks 1 through 16. To the right of the main content is a vertical sidebar with a 'Buy College Football Tickets' button and a large 'T-Mobile' logo. The bottom half of the image shows a pink mobile advertisement for T-Mobile. It features the slogan 'BRING YOUR OWN PHONE TO T-MOBILE' and a 'SWITCH NOW' button. There's also small text about device requirements and service plans. The bottom right corner of the ad contains the CBSSPORTS.COM SHOP logo.



Graph Laplacian

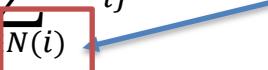
- Graph Laplacian $L \in R^{m \times m}$ is a matrix representation of graph
- Capture information on many graph properties
- Computation $L = D - A$
 - Start with (weighted) adjacency matrix A

$$A_{ij} = \begin{cases} w_{ij} > 0, & \text{if node } i \text{ and } j \text{ are neighbors} \\ 0, & \text{if } i \text{ and } j \text{ are not direct neighbors} \end{cases}$$

- Diagonal degree matrix

$$D_{ii} = \sum_{j \in N(i)} w_{ij}$$

Neighboring node of i



Graph Laplacian Example



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

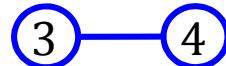
$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Property I of Graph Laplacian

- $L = D - A$

- The multiplicity of the eigenvalue 0 corresponds to the number of connected components in the graph

- Example



$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$Lv_1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$Lv_2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

Property II of Graph Laplacian

- $L = D - A$
- The eigenvectors with eigenvalue 0 contains cluster assignment information

- Example

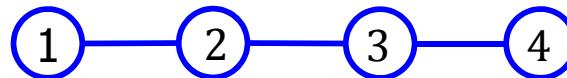


$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$Lv^1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$Lv^2 = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

What if the graph has only 1 component



$$A = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

Special eigenvector with all 1's

- $L = D - A$
- The smallest eigenvalue of L is 0, corresponding a constant eigenvector $\frac{1}{\sqrt{m}} \mathbf{1}$
- Example



$$L = \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$

$$\frac{1}{\sqrt{4}} L \mathbf{1} = \frac{1}{\sqrt{m}} \begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \frac{1}{\sqrt{m}} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = 0 \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

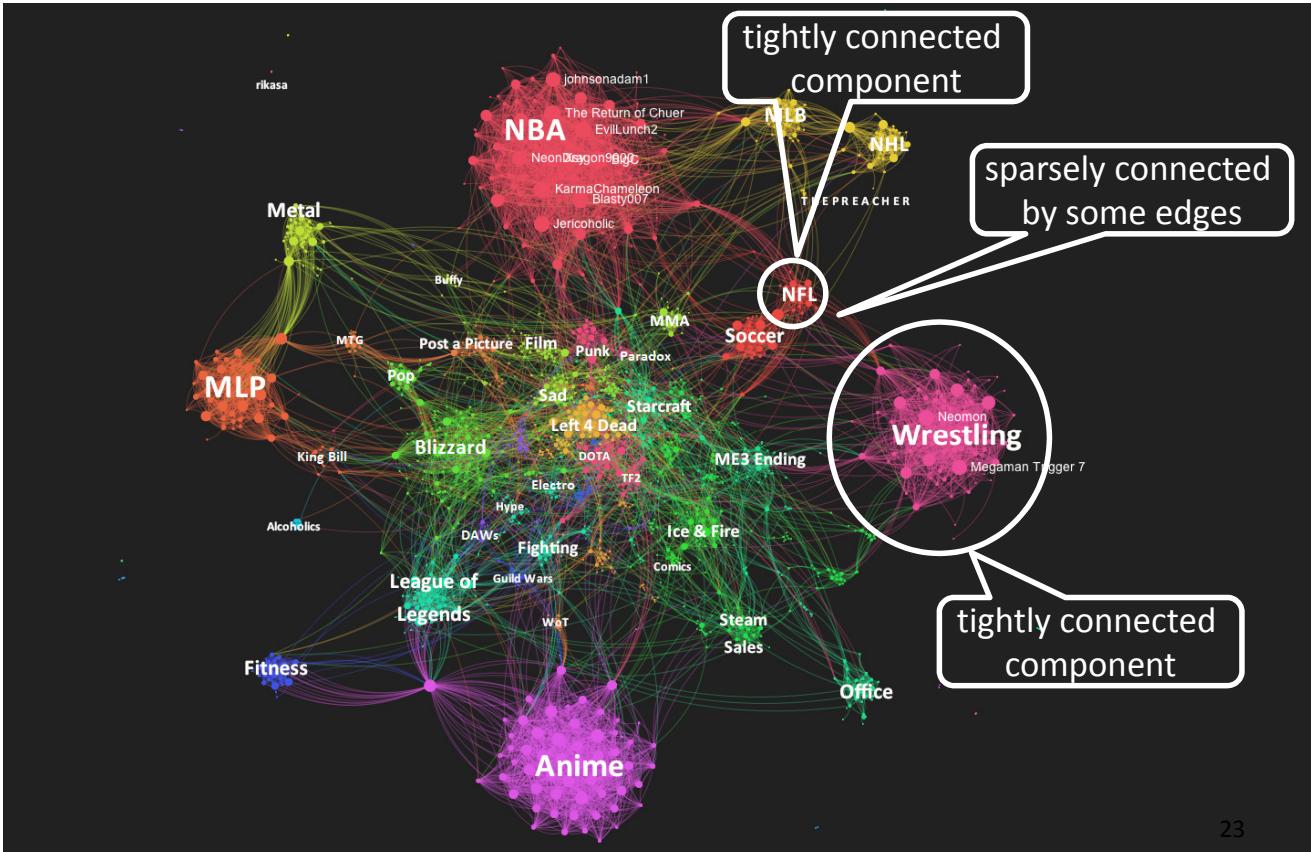
What if the graph has k components

- If a graph has k connected components (or k clusters)
- The graph Laplacian has k blocks

$$L = \begin{pmatrix} L_1 & 0 & 0 & 0 \\ 0 & L_2 & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & L_k \end{pmatrix}$$

- The graph Laplacian has k eigenvectors with zero eigenvalues
 - Eigenvector 1 is constant in block 1, but 0 in other blocks;
eigenvector 2 is constant in block 2, but 0 in other blocks;
- ...

Real world not perfectly block



High level idea of spectral clustering

- Examine the properties of graph Laplacian for the perfect cases
 - The number of 0 eigenvalues corresponds to the number of connected components
 - Eigenvectors correspond to cluster assignment
- Then use the intuition from perfect cases to design algorithms for the imperfect case.
 - Eigenvectors no longer correspond exactly cluster indicator
 - Perform post processing to obtain cluster assignment

In general (imperfect case)

- If a graph has k **tightly** connected components (or k clusters) with **sparsely** connected edges
- The graph Laplacian has **approximately** k blocks
- The graph Laplacian has k eigenvectors with **small** eigenvalues
- Eigenvector 1 is **approximately** constant in block 1, but 0 in other blocks; eigenvector 2 ...

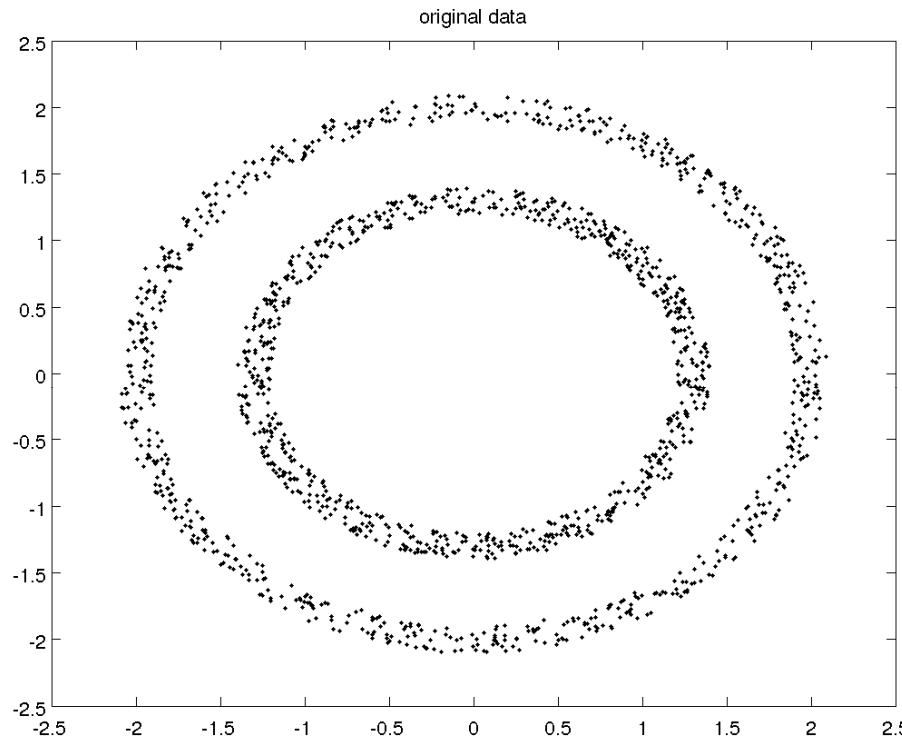
Summary of spectral clustering

- Step 1: represent graph as adjacency matrix $A \in R^{m \times m}$
- Step 2: form a special matrix $L = D - A$, the graph Laplacian
- Step 3: compute k eigenvectors, v^1, v^2, \dots, v^k , of L corresponding to the k **smallest** eigenvalues ($k \ll m$)
- Step 4: run kmeans algorithm on $Z = (v^1, v^2, \dots, v^k)$ by treating each row as a new data point

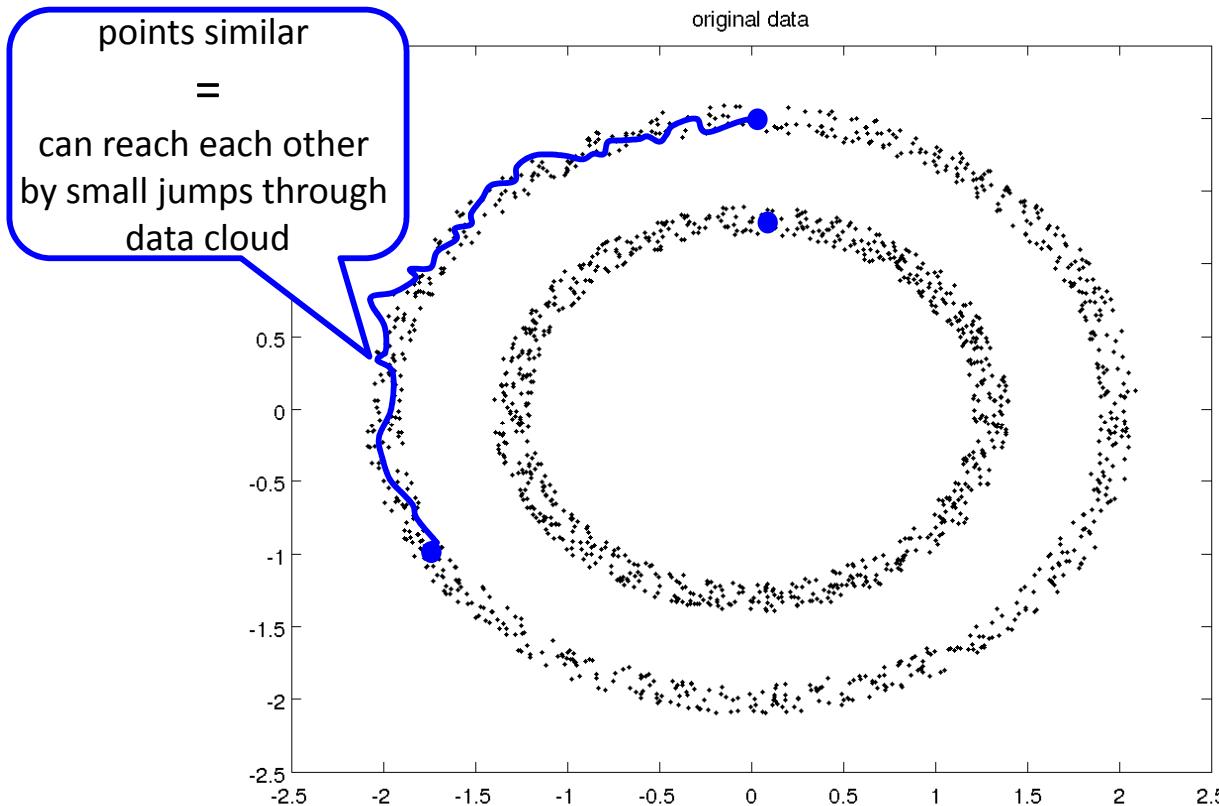
Questions

- Similarity in spectral clustering is based on
 - Euclidean distance
 - Connectivity
- How to pick the number eigenvectors?
 - Random
 - Look at the eigengap

How about this dataset?



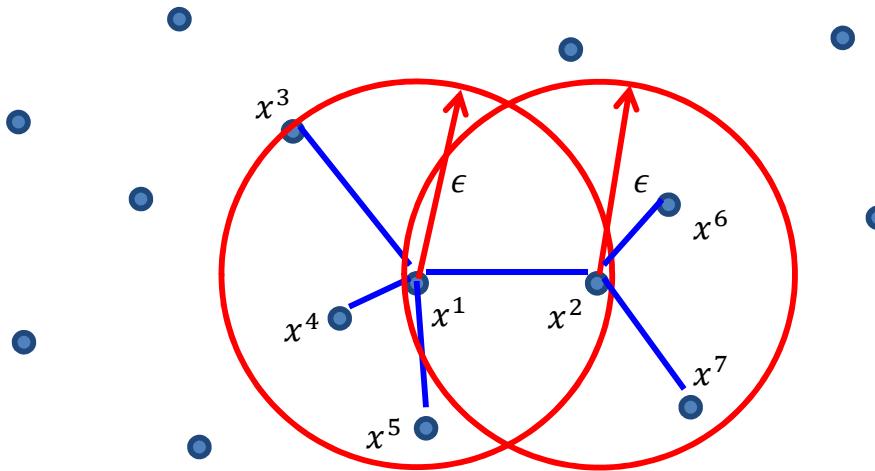
What's a reasonable similarity measure?



Nearest neighbor graph

- Given m data points, threshold ϵ , construct matrix $A \in R^{m \times m}$

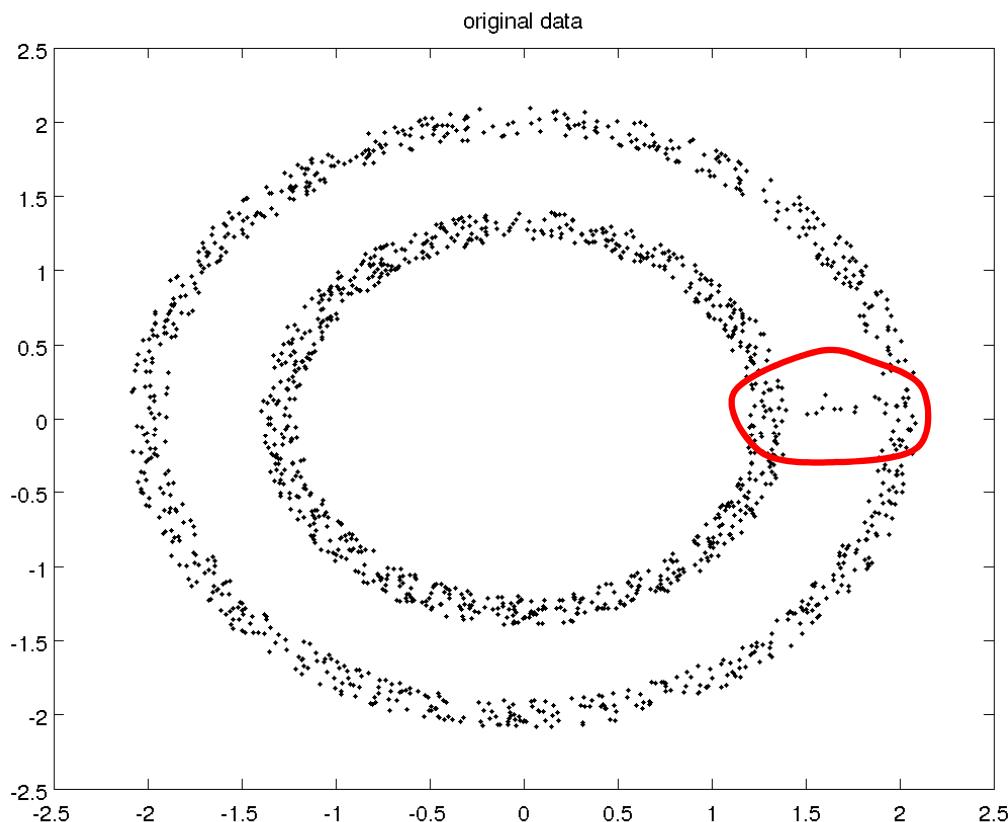
$$A^{ij} = \begin{cases} 1, & \text{if } \|x^i - x^j\| \leq \epsilon \\ 0, & \text{otherwise} \end{cases}$$



Spectral clustering for vector data

- Given m nodes, $\{x^1, x^2, \dots, x^m\} \in R^n$
- Step 1: build an adjacency matrix A using nearest neighbors
- Step 2: represent graph as adjacency matrix $A \in R^{m \times m}$
- Step 3: form a special matrix $L = D - A$, the graph Laplacian
- Step 4: compute k eigenvectors, v^1, v^2, \dots, v^k , of L corresponding to the k **smallest** eigenvalues ($k \ll m$)
- Step 5: run kmeans algorithm on $Z = (v^1, v^2, \dots, v^k)$ by treating each row as a new data point

What happens by adding more data points?



Variants of spectral clustering

- Given m data points (nodes), $\{x^1, x^2, \dots, x^m\} \in R^n$
- Build an adjacency matrix A using **kernel functions** (if the input is already a graph, skip this step)
- Compute $B = D^{-1/2} A D^{-1/2}$, where D is the degree matrix
- Compute k eigenvectors, v^1, v^2, \dots, v^k , of B corresponding to the k **largest** eigenvalues
- Use $z^1 = (v_1^1, v_1^2, \dots, v_1^k), z^2 = (v_2^1, v_2^2, \dots, v_2^k) \dots$ as the new coordinates for data point 1, 2, ..., and then run kmeans on these new coordinates

