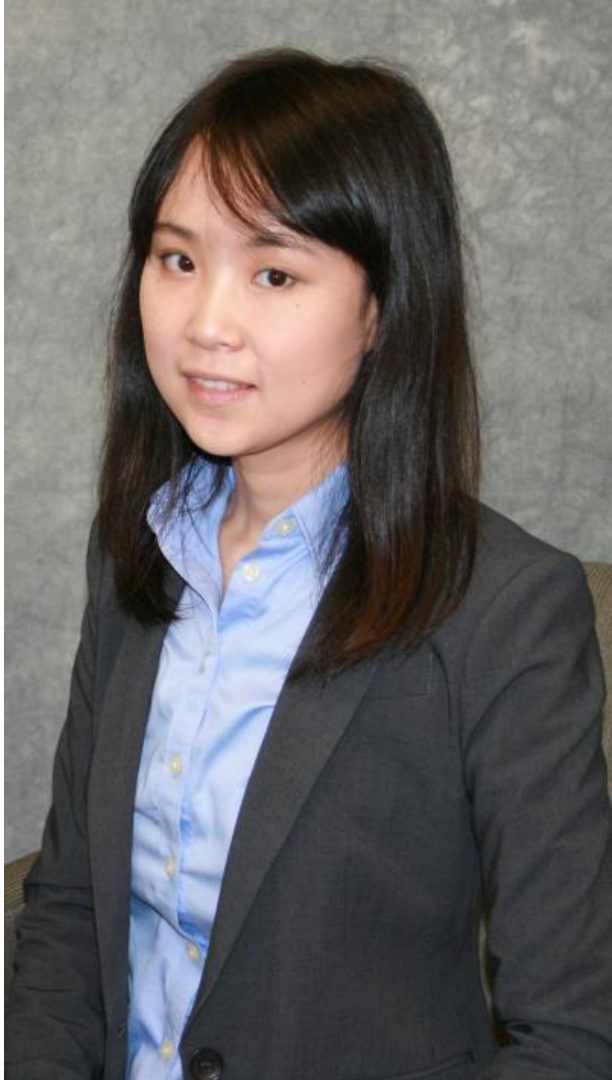
Computational Data Analysis

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

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

K-means algorithm

- Initialize k cluster centers, $\{c^1, c^2, \ldots, c^k\}$, randomly
- Do
  - Decide the cluster memberships of each data point, x^i, by assigning it to the nearest cluster center (cluster assignment)

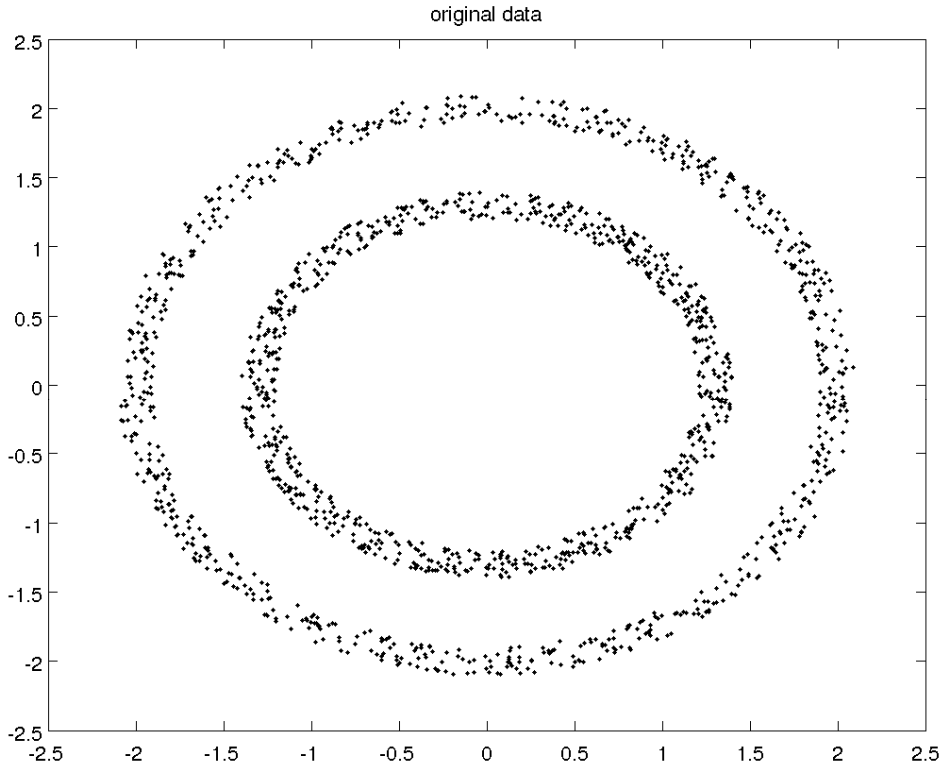
$$\pi(i) = argmin_{j=1,\ldots,k} \left\| x^i - c^j \right\|^2$$

  - Adjust the cluster centers (center adjustment)

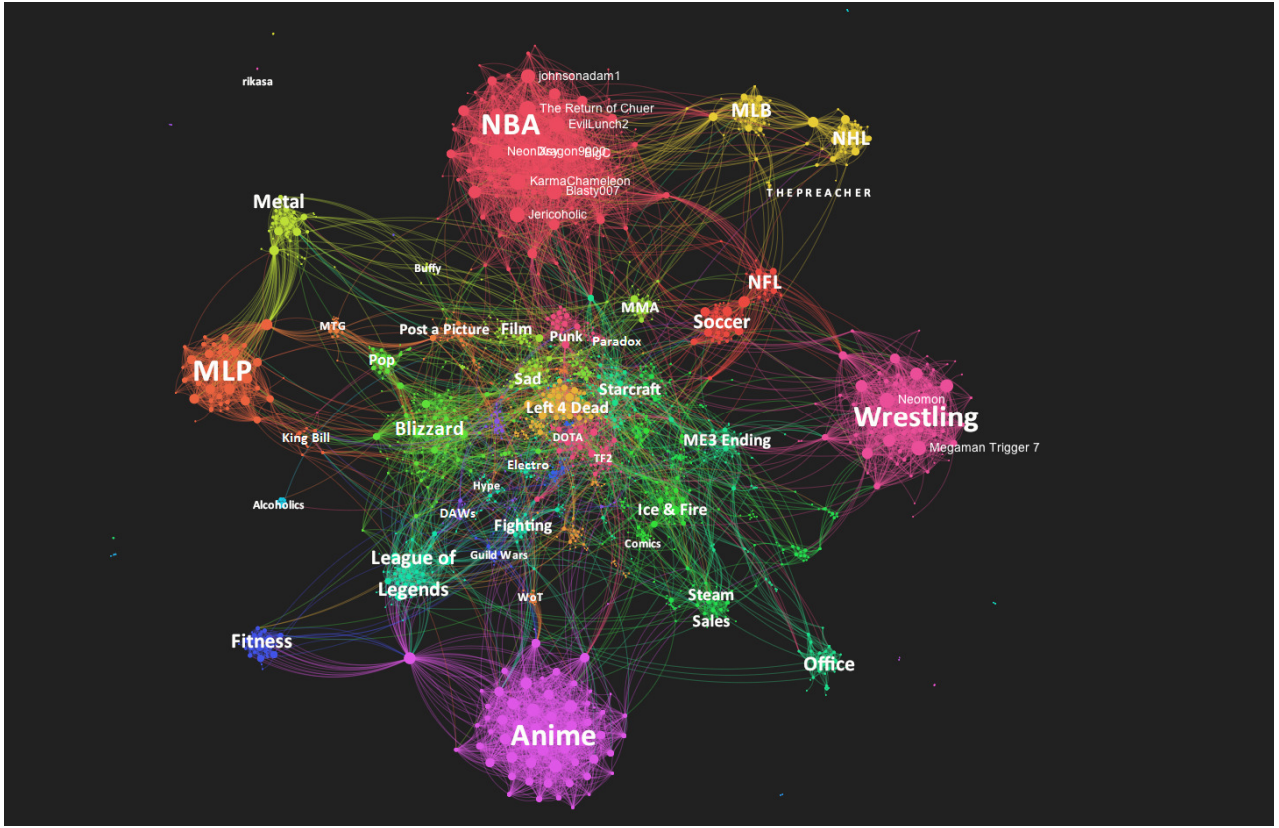
$$c^j = \frac{1}{|\{i : \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$$

- While any cluster center has been changed

How about this dataset? (Run test_tworings.m)



How about clustering nodes in social networks



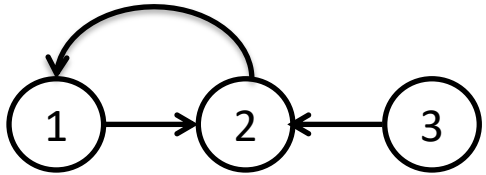
Representing graph using matrices

- Adjacency matrix for unweighted graph

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

- Directed vs. undirected graph
- Weighted vs. unweighted graph

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



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

$$A = \begin{bmatrix} 0 & 1.2 & 0 \\ 1.2 & 0 & 0.3 \\ 0 & 0.3 & 0 \end{bmatrix}$$

Graph Laplacian (undirected graph)

- For a graph with m nodes, adjacency matrix $A \in R^{m \times m}$
- Vertex degree $d_i = \sum A_{ij}$ (for unweighted graph: the number of neighboring nodes)
- Degree matrix

$$D = diag\{d_1, d_2, \dots, d_m\}$$

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

$$L = D - A$$

- measuring to what extent a graph differs at one vertex from its values at nearby vertices.

https://mathworld.wolfram.com/LaplacianMatrix.html

Graph Laplacian Example

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

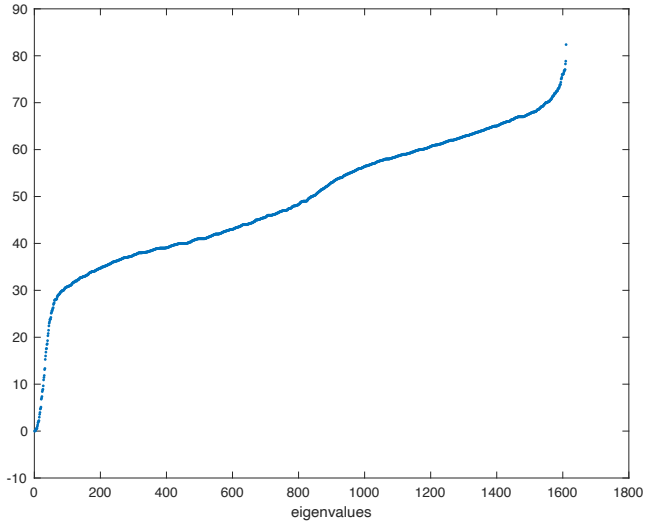
$$D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

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

Eigenvalue problem

- Given a symmetric matrix $C \in R^{n \times n}$
  - Find a vector $u \in R^n$ and $\|u\| = 1$
  - Such that
    $$Cu = \lambda u$$

- There will be multiple solution: $u^1, u^2, \ldots u^n$ (called the **eigenvectors**) with different $\lambda_1, \lambda_2, \ldots \lambda_n$ (called the **eigenvalues**.)
  - Eigenvectors are ortho-normal: ${u^i}^\top u^i = 1, {u^i}^\top u^j = 0$
  - Eigenvalues are called spectrum



Eigenvalues of graph Laplacian of the ”two-ring” data

Eigendecomposition

- Given a symmetric matrix $C \in R^{n \times n}$
- Eigendecomposition

$$C = U \Lambda U^T$$

- Example: $C = \begin{bmatrix} 5 & 3 & 4 \\ 3 & 2 & 3 \\ 4 & 3 & 5 \end{bmatrix}$, $C = U \Lambda U^T$

$$U = \begin{bmatrix} 0.3015 & 0.7071 & 0.6396 \\ -0.9045 & 0 & 0.4264 \\ 0.3015 & -0.7071 & 0.6396 \end{bmatrix}, \Lambda = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 11 \end{bmatrix}$$

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

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

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

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

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 $\frac{1}{\sqrt{m}}1$
- Example

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

$$\frac{1}{\sqrt{4}}L1 = \frac{1}{\sqrt{m}}\begin{pmatrix} 1 & -1 & 0 & 0 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = 0\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

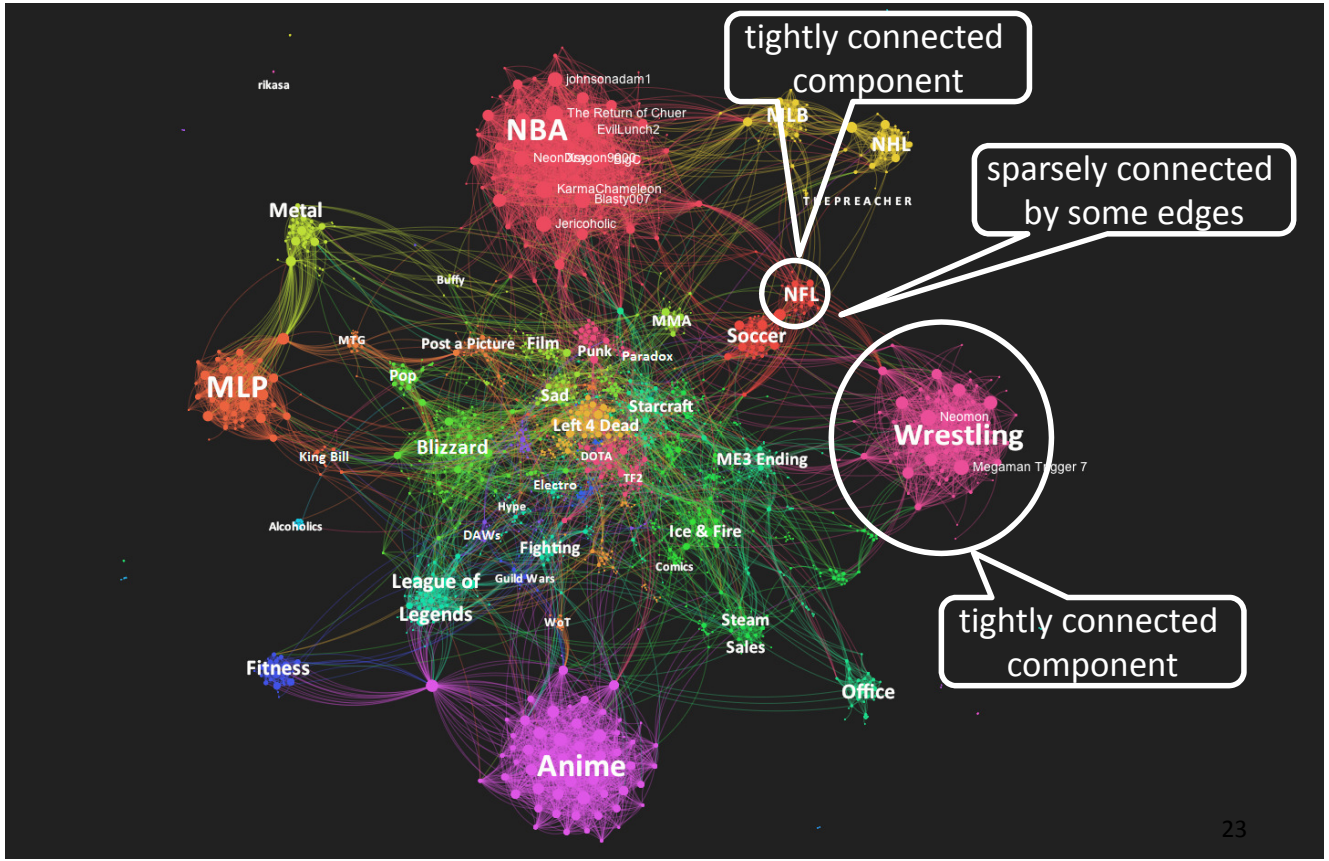
What if the graph has k components

- 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 Laplancian 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 not 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 Laplancian has k eigenvectors with small eigenvalues
- Eigenvector 1 is approximately constant in block 1, but 0 in other blocks; eigenvector 2 …

Ideas of spectral clustering

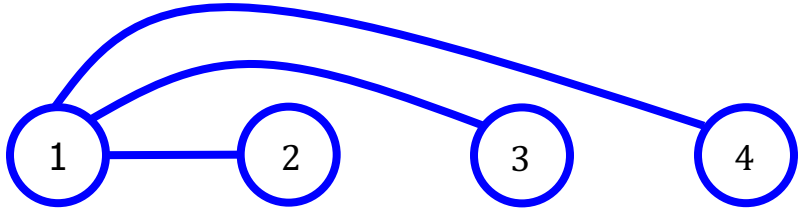
- Step 1: represent graph as adjacency matrix $A \in R^{m \times m}$
- Step 2: form a special matrix $L = D - A$, the graph Laplacian
- Step 3: compute k eigenvectors, $v^1, v^2, \ldots, v^k$, of L corresponding to the k smallest eigenvalues ($k \ll m$)
- Step 4: run kmeans algorithm on $Z = \left(v^1,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

Spectral clustering algorithm

Step 1: represent graph as adjacency matrix $A \in R^{m \times m}$, m: number of nodes

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

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



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

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

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

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

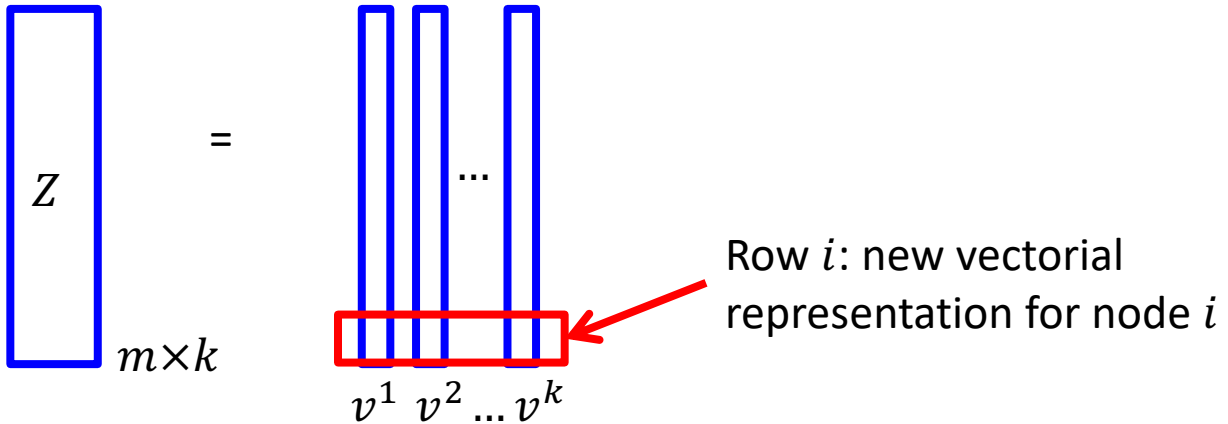
$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

Spectral clustering algorithm (cont.)

Step 3: Perform eigendecomposition of **graph Laplacian** L, compute k eigenvectors, $v^1, v^2, \ldots, v^k$, corresponding to the k smallest eigenvalues ($k \ll m$)

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

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

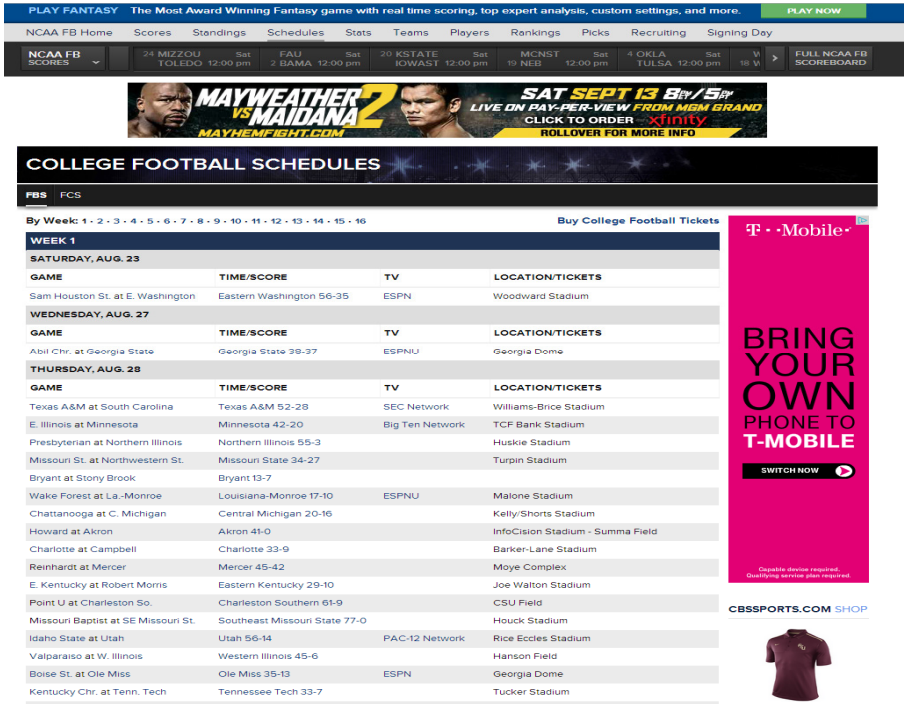


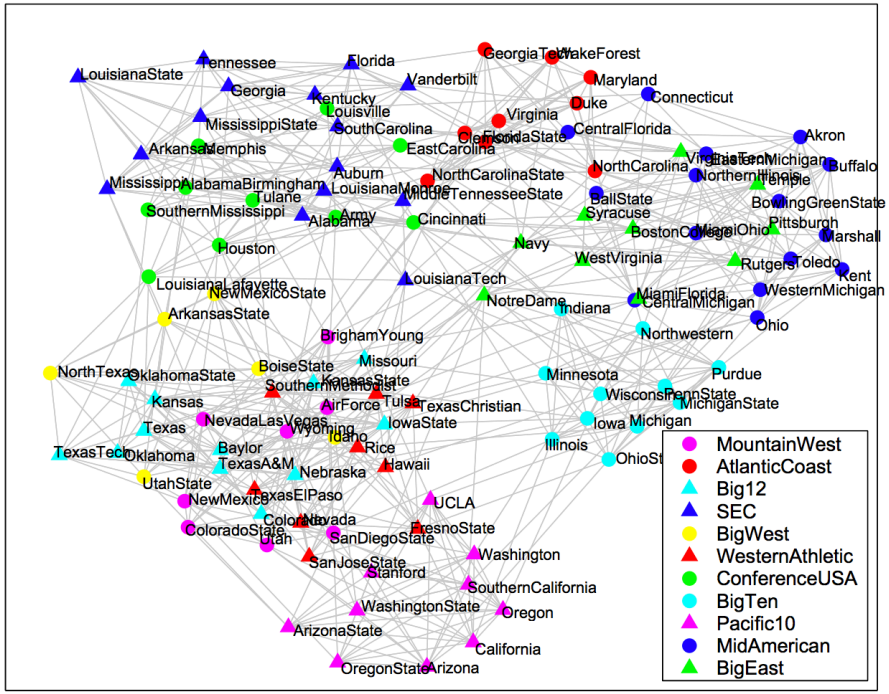
Questions

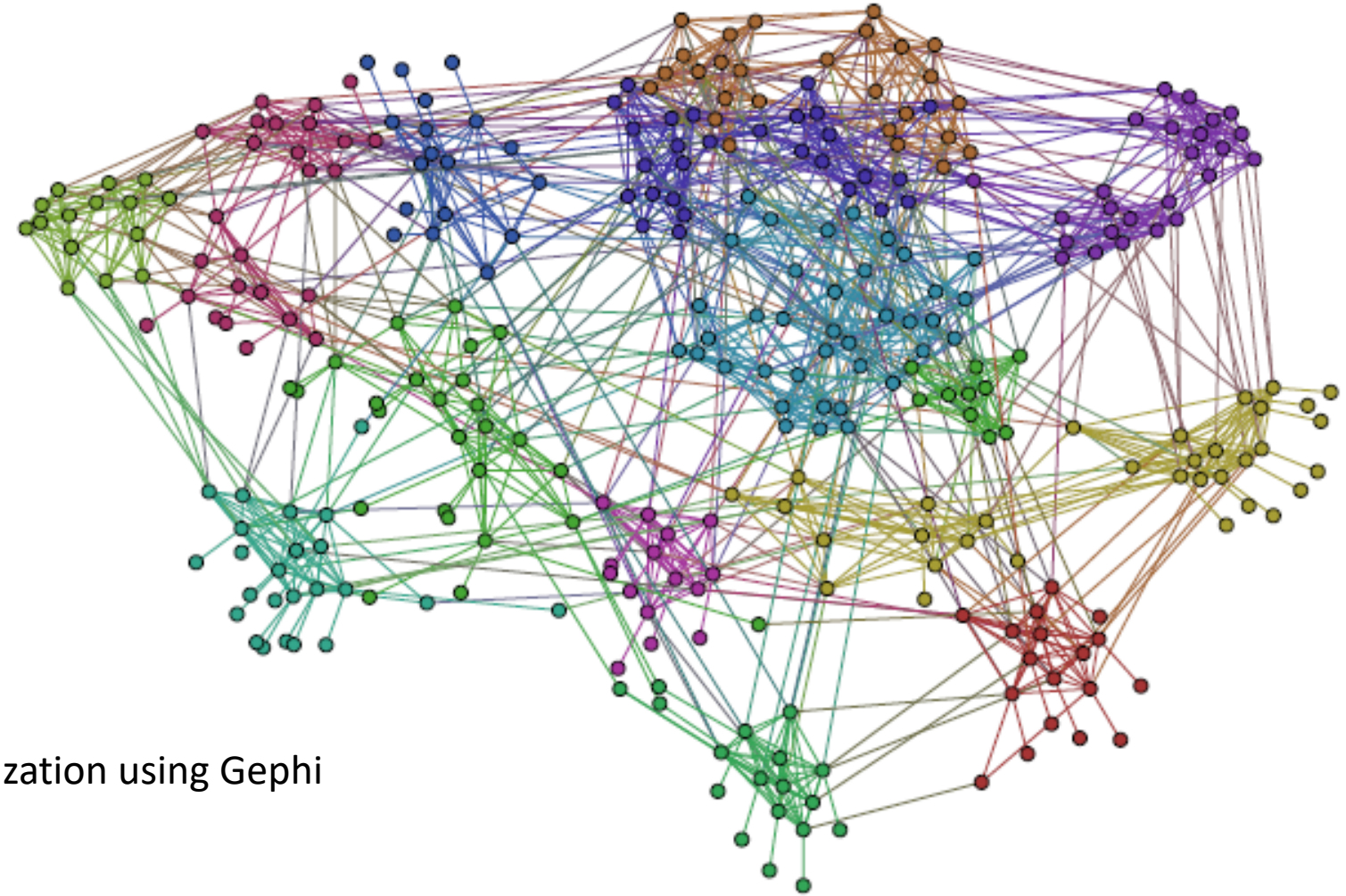
- Similarity in spectral clustering is based on
  - Euclidean distance
  - Connectivity

- How to pick the number eigenvectors?
  - Random
  - Look at the eigengap

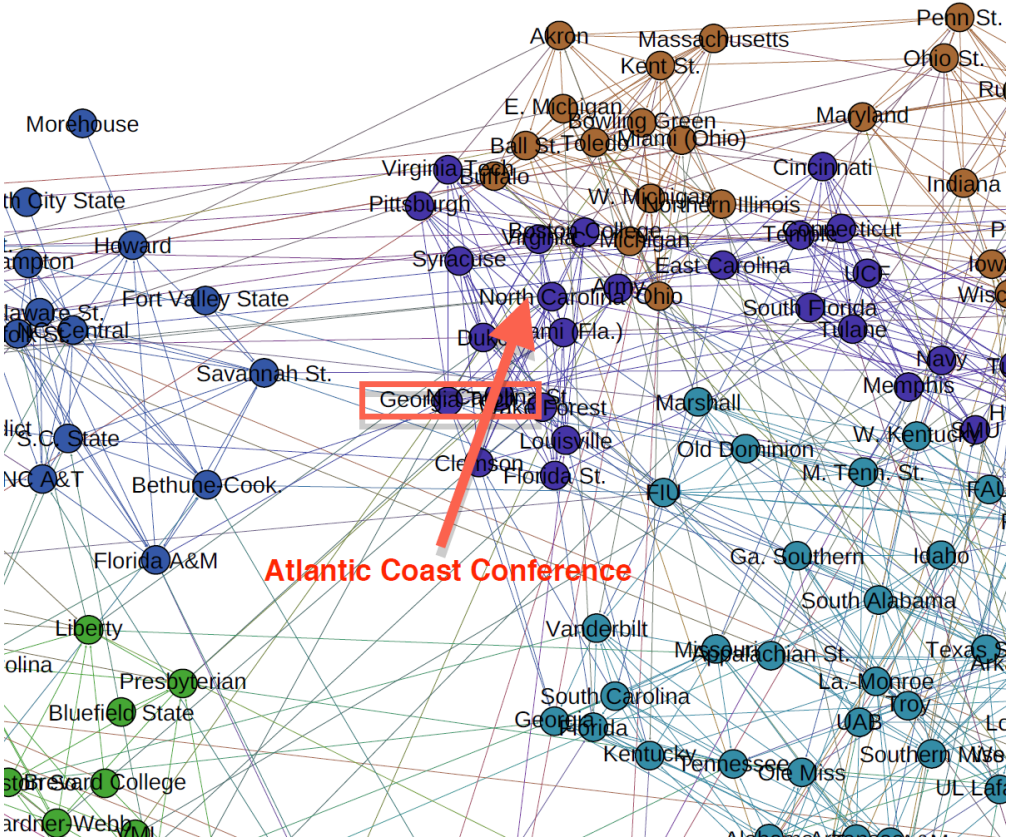
Run demo test_football.m



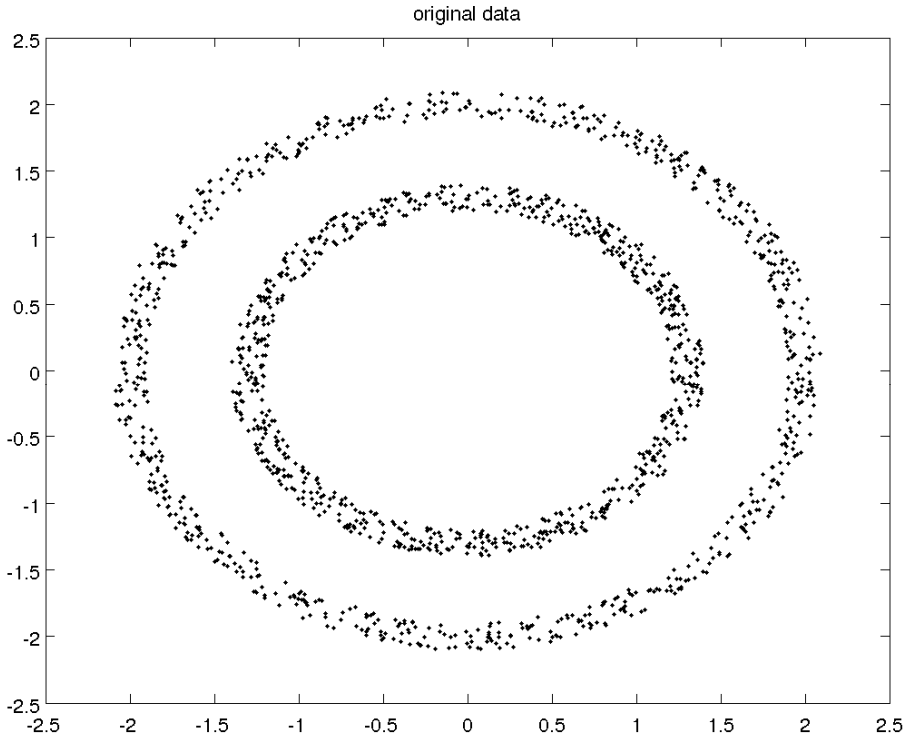




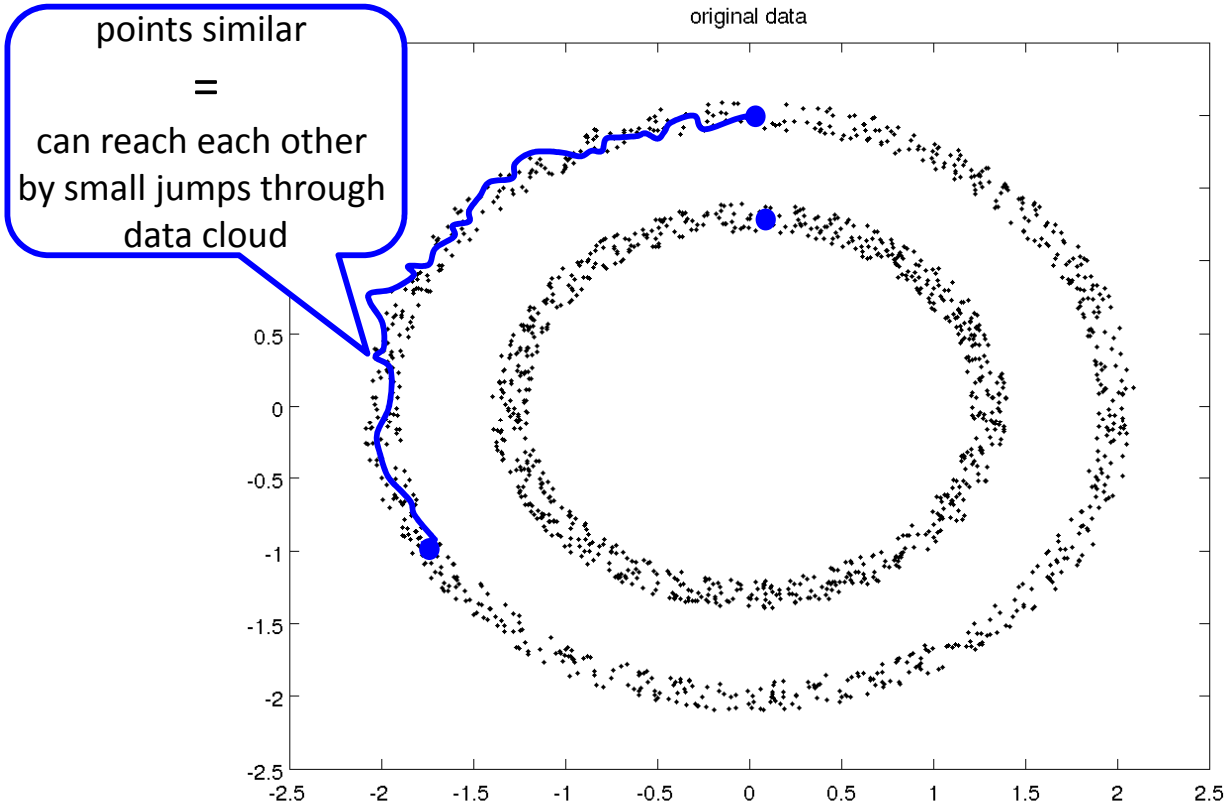
Visualization using Gephi


Akron
Massachusetts
Kent St.
Penn St.
Ohio St.
E. Michigan
Bowling Green
Maryland
Morehouse
Ball St.
Toledo
Miami (Ohio)
Virginia Tech
Buffalo
Cincinnati
Indiana
Pittsburgh
W. Michigan
Northern Illinois
Howard
Boston College
Virginia
Michigan
Connecticut
Syracuse
East Carolina
UCF
Fort Valley State
North Carolina
Army
Ohio
South Florida
Tulane
Savannah St.
Navy
Memphis
Georgia
Wake Forest
Marshall
S.C. State
Louisville
W. Kentucky
Old Dominion
Clemson
Florida St.
M. Tenn. St.
Bethune-Cookman
FIU
Florida A&M
Atlantic Coast Conference
Ga. Southern
Idaho
South Alabama
Liberty
Vanderbilt
Presbyterian
Appalachian St.
La.-Monroe
Bluefield State
South Carolina
Troy
Georgia
Florida
UAB
Kentucky
Tennessee
Ole Miss
Southern Miss
Brevard College
Gardner-Webb
UL Lafayette

How about this dataset?



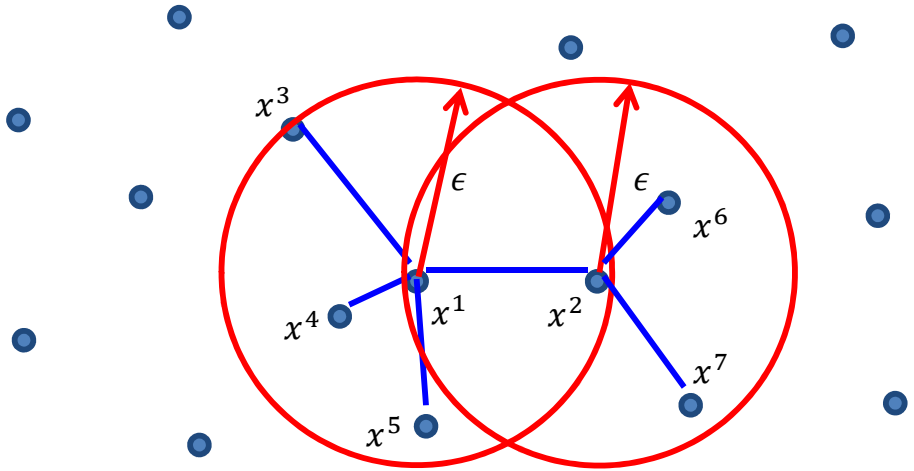
What's a reasonable similarity measure?



Nearest neighbor graph

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

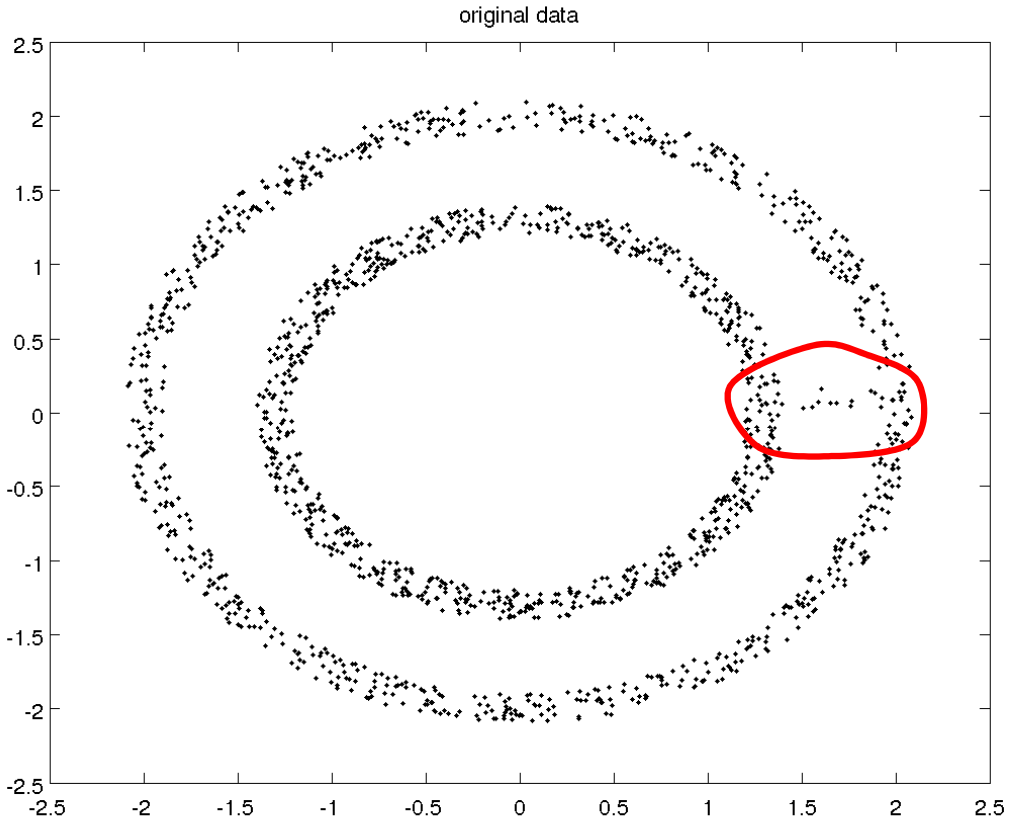
$$A^{ij} = \begin{cases} 1, \text{if } \|x^i - x^j\| \leq \epsilon \\ 0, \qquad \text{otherwise} \end{cases}$$



Spectral clustering for vector data

- Given m nodes, $\{x^1, x^2, \ldots x^m\} \in R^n$
- 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, $v^1, v^2, \ldots, v^k$, of L corresponding to the k smallest eigenvalues ($k \ll m$)
- Step 5: run kmeans algorithm on $Z = \left(v^1, v^2, \ldots v^k\right)$ 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 $B = D^{-1/2}AD^{-1/2}$, 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 $z^1 = \left(v_1^1, v_1^2, \dots, v_1^k\right), z^2 = \left(v_2^1, v_2^2, \dots, v_2^k\right)$ … as the new coordinates for data point 1, 2, …, and then run kmeans on these new coordinates


SUMMARY