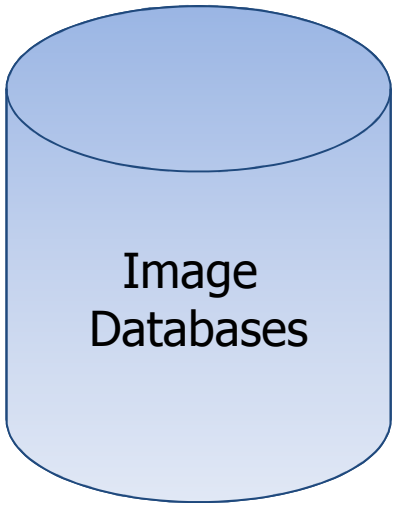
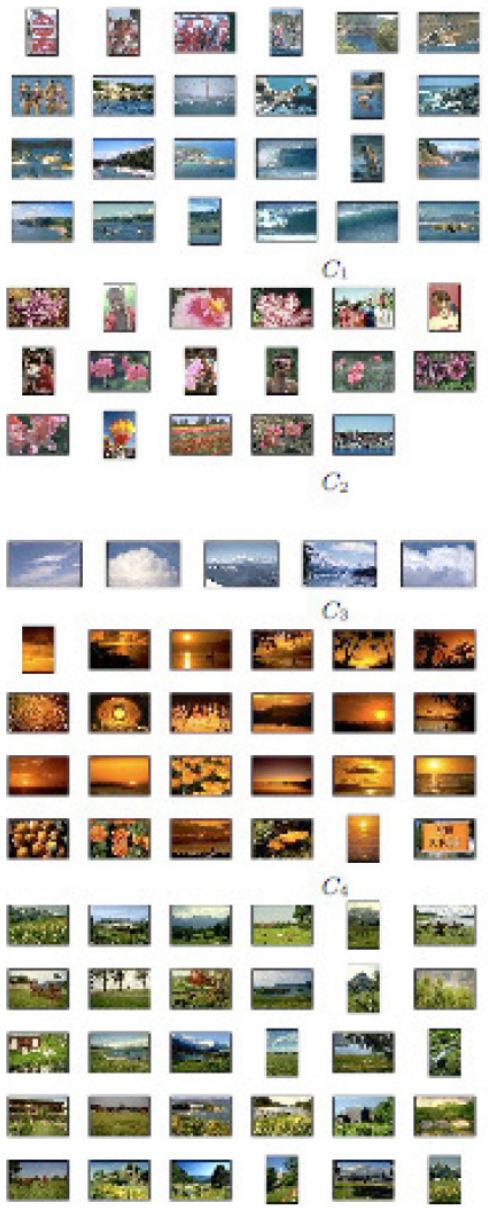
Clustering Nodes in Graphs

Le Song

Machine Learning
CS 7641,CSE/ISYE 6740, Fall 2016

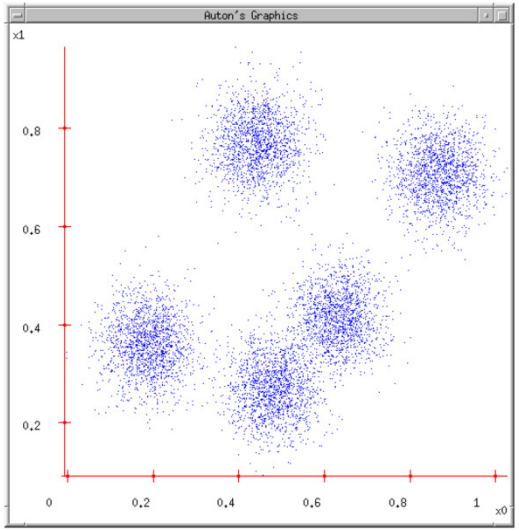
Clustering images



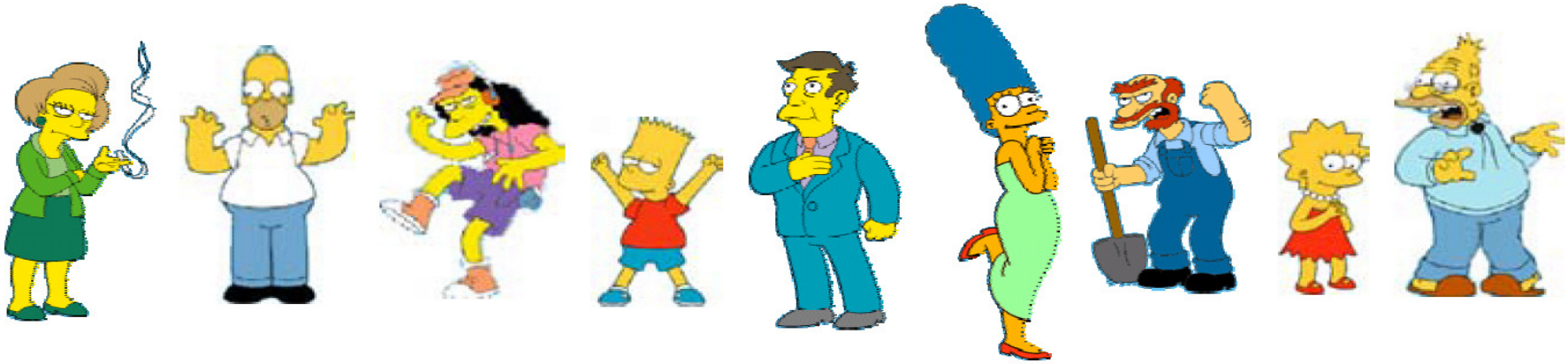


Goal of clustering:

Divide object into groups, and objects within a group are more similar than those outside the group

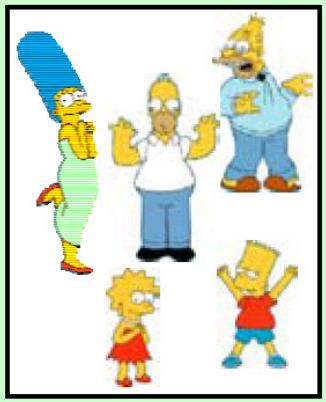


Clustering is a subjective task



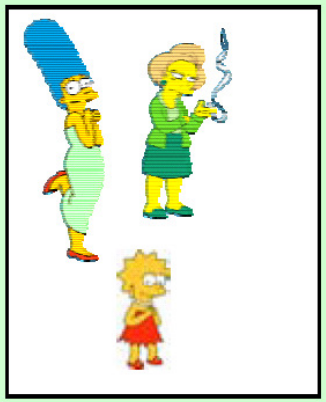
What is consider similar/dissimilar?

Clustering is subjective



Simpson's Family

School Employees



Females

Males

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)

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

  - Adjust the cluster centers (center adjustment)

$$c^j = argmin_{v \in R^n} \sum_{i:\pi(i)=j} d\left(x^i, v\right)^2$$

- While any cluster center has been changed

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)

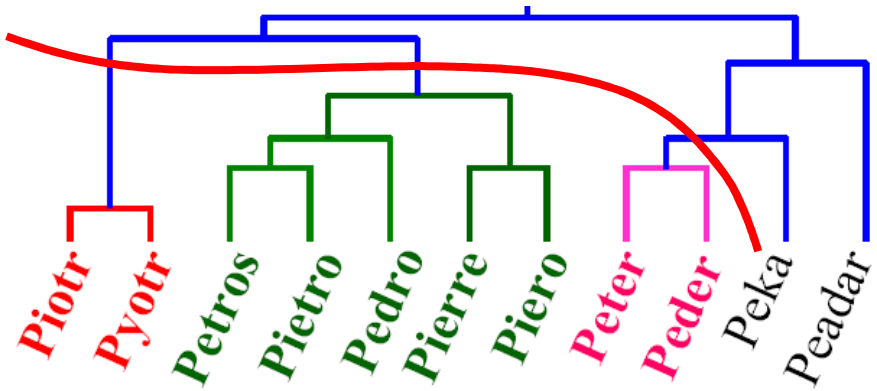
$$c^j = \frac{1}{|\{i: \pi(i) = j\}|} \sum_{i:\pi(i)=j} x^i$$

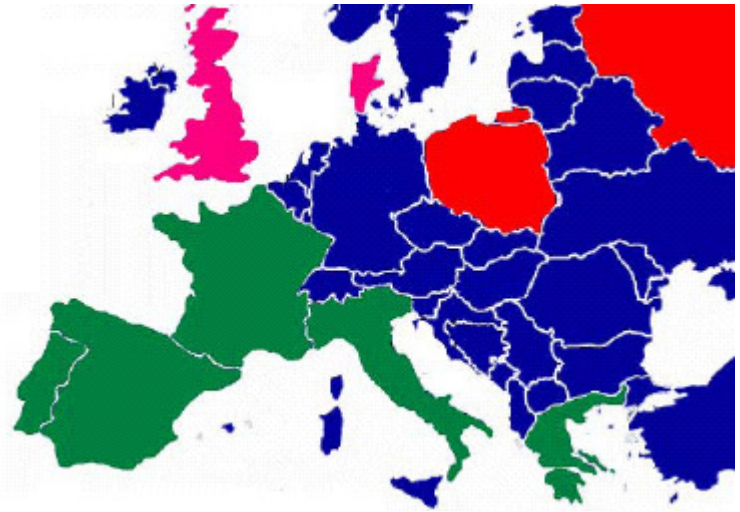
- While any cluster center has been changed

Hierarchical clustering

- Organize data in a hierarchical fashion (dendrogram)

- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.





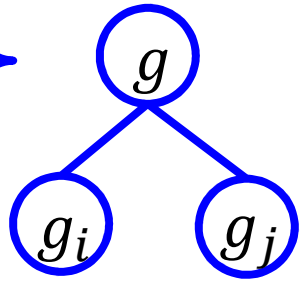
Bottom up hierarchical clustering

- Assign each data point to its own cluster, $g_1 = \{x_1\}, g_2 = \{x_2\}, \ldots, g_m = \{x_m\}$, and let $G = \{g_1, g_2, \ldots, g_m\}$

$$D(g_i, g_j) = \min_{x \in g_i, y \in g_j} d(x, y)$$

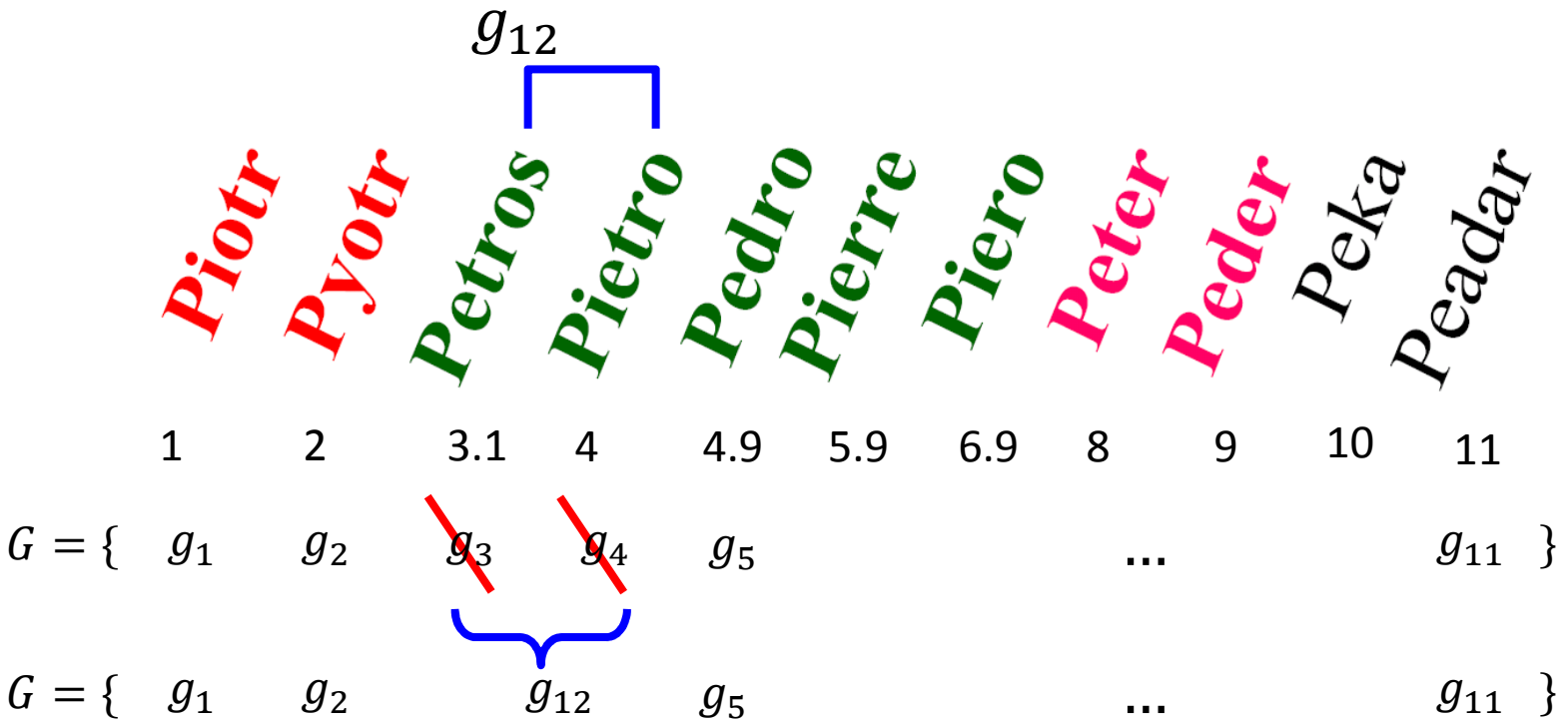
- Do
  - Find two clusters to merge: $i, j = argmin_{1 \le i,j \le |G|} \; D(g_i, g_j)$
  - Merge the two clusters to a new cluster: $g \leftarrow g_i \cup g_j$
  - Remove the merged clusters: $G \leftarrow G \setminus g_i, G \leftarrow G \setminus g_j$
  - Add the new cluster: $G \leftarrow G \cup \{g\}$

keep track of relations

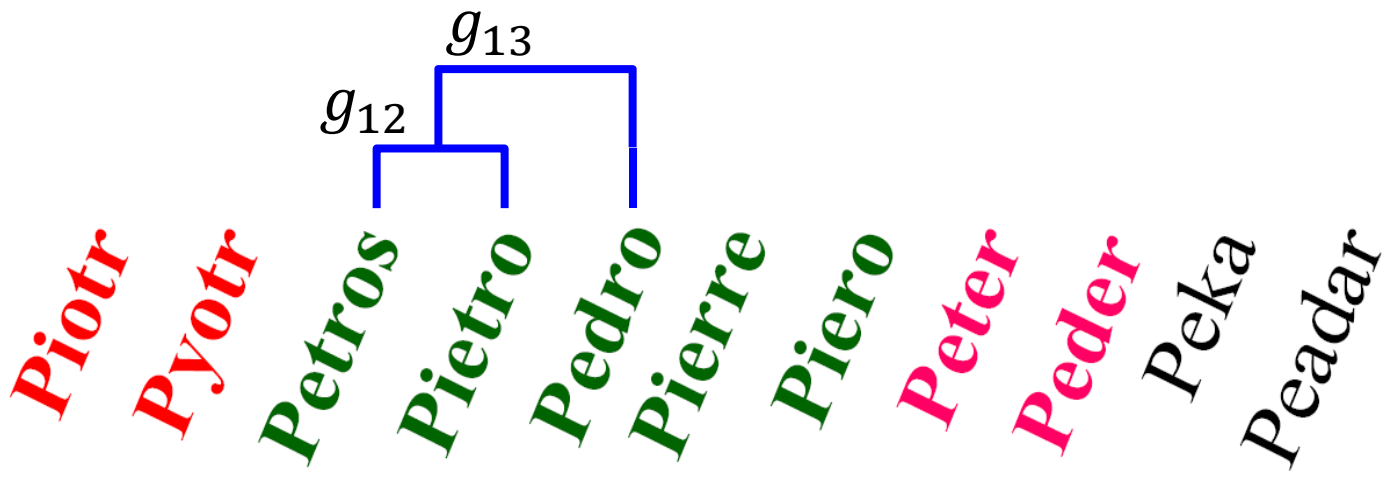


- While $|G| > 1$

Hierarchical clustering: step-2



Hierarchical clustering: step-3

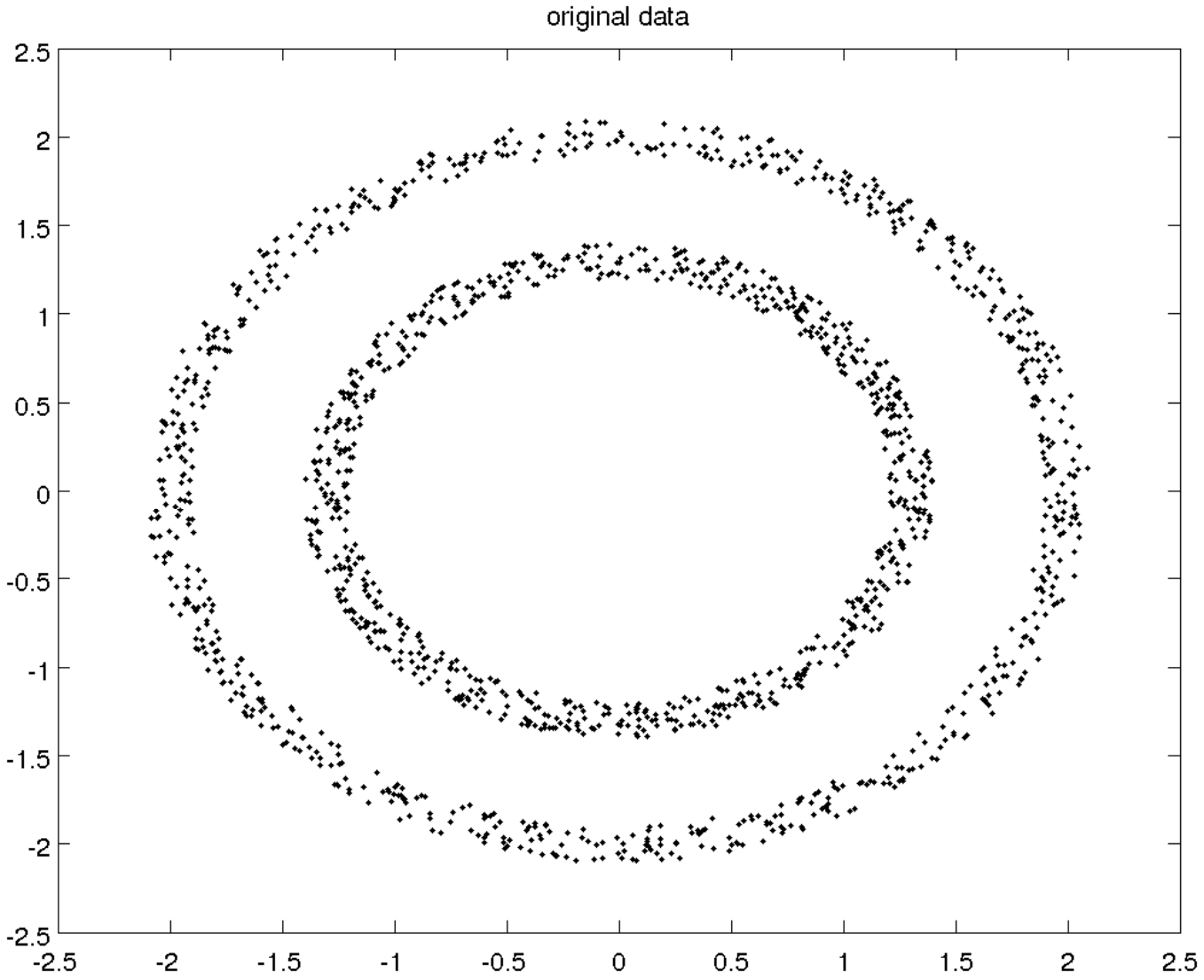


$G = \{ \; g_1 \quad g_2 \quad g_3 \quad g_4 \quad g_5 \quad \ldots \quad g_{11} \; \}$

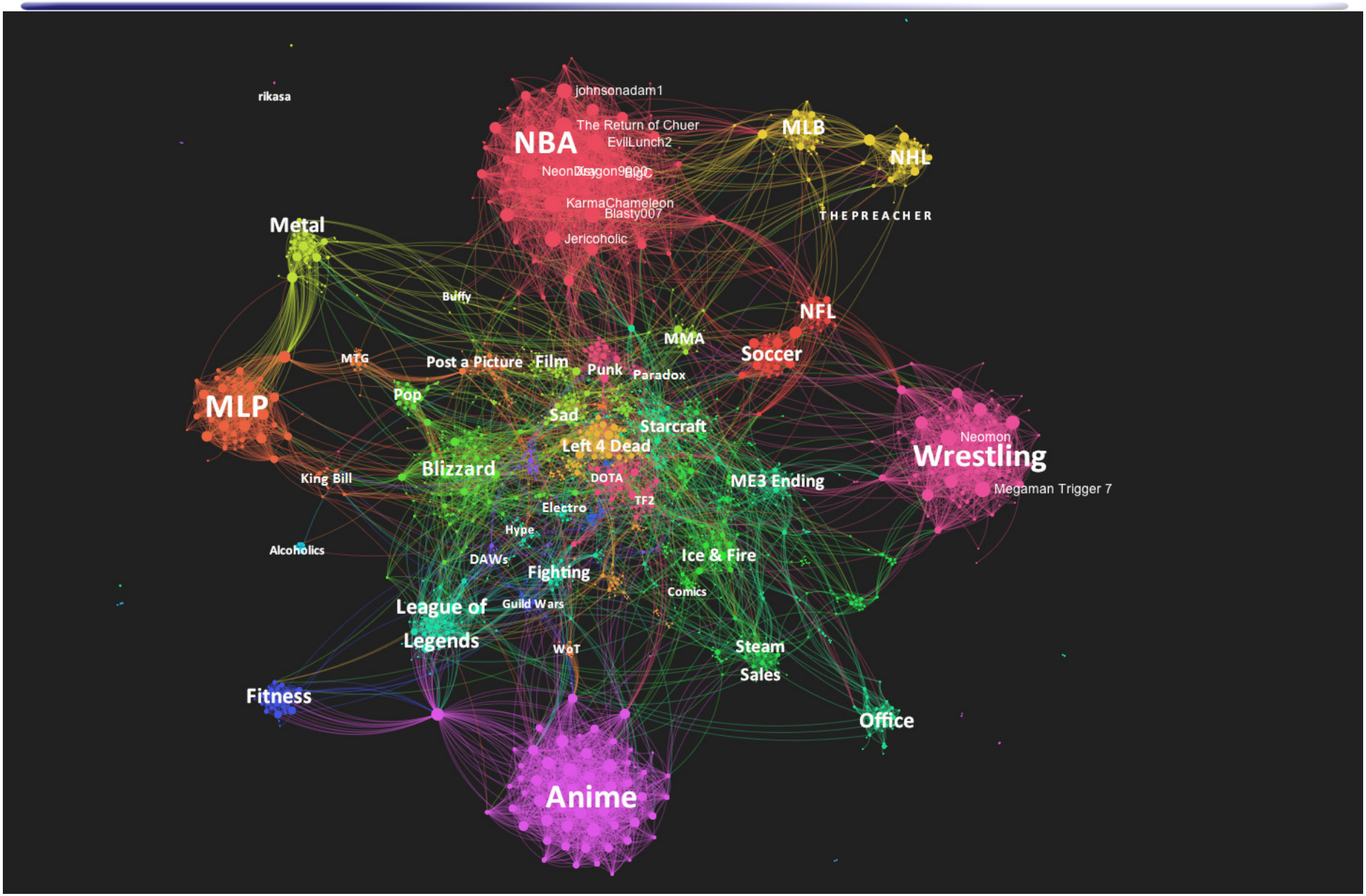
$G = \{ \; g_1 \quad g_2 \quad g_{12} \quad g_5 \quad \ldots \quad g_{11} \; \}$

$G = \{ \; g_1 \quad g_2 \quad g_{13} \quad \ldots \quad g_{11} \; \}$

How about this dataset? (Run test_tworings.m)



How about clustering nodes in social networks

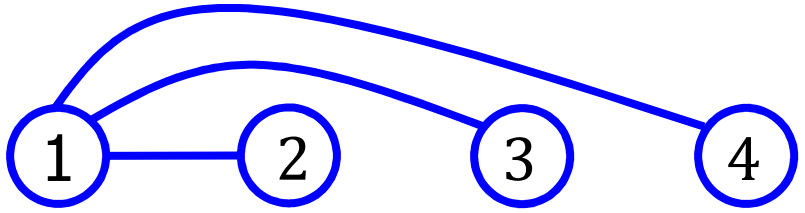


Spectral clustering algorithm

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

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

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



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

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

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

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

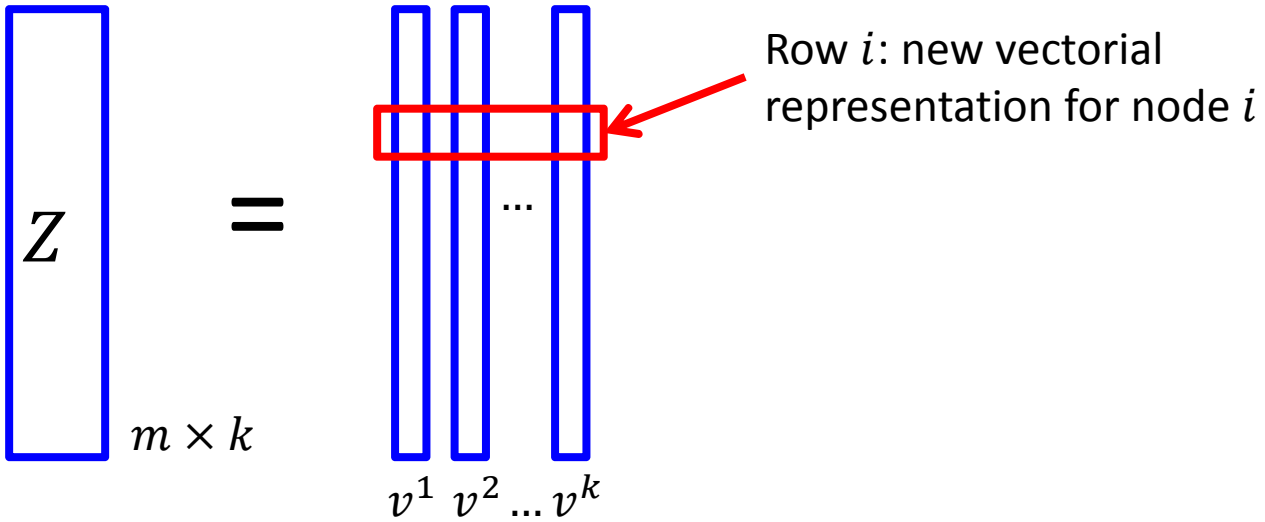
$$L = \begin{pmatrix} 3 & -1 & -1 & -1 \\ -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ -1 & 0 & 0 & 1 \end{pmatrix}$$

Spectral clustering algorithm (cont.)

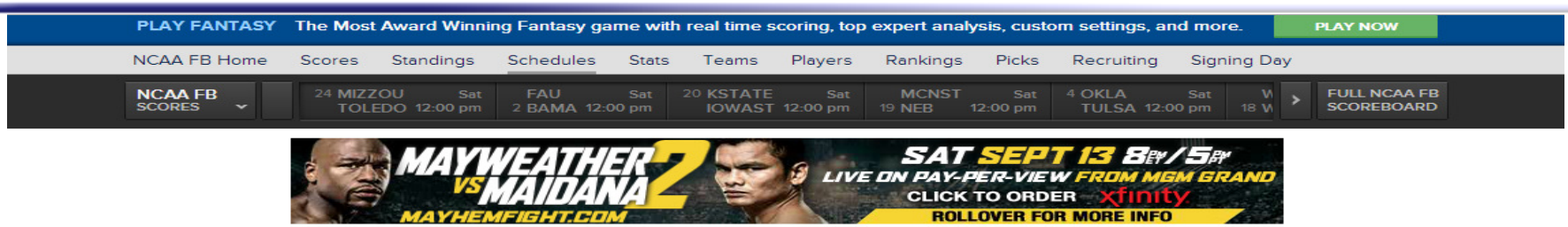
- Step 3: compute k eigenvectors, $v^1, v^2, \ldots, v^k$, of L corresponding to the k smallest eigenvalues ($k \ll m$)

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

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



Run demo test_football.m



COLLEGE FOOTBALL SCHEDULES

FBS | FCS

By Week: 1 · 2 · 3 · 4 · 5 · 6 · 7 · 8 · 9 · 10 · 11 · 12 · 13 · 14 · 15 · 16

Buy College Football Tickets

WEEK 1

SATURDAY, AUG. 23

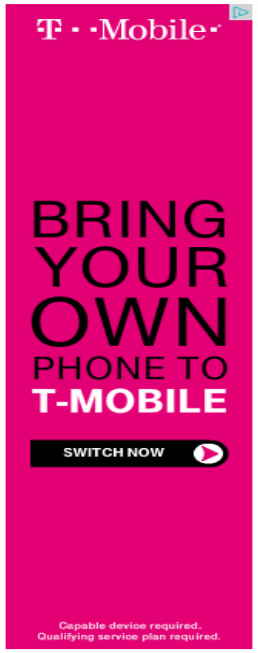
GAME	TIME/SCORE	TV	LOCATION/TICKETS
Sam Houston St. at E. Washington	Eastern Washington 56-35	ESPN	Woodward Stadium

WEDNESDAY, AUG. 27

GAME	TIME/SCORE	TV	LOCATION/TICKETS
Abil Chr. at Georgia State	Georgia State 38-37	ESPNU	Georgia Dome

THURSDAY, AUG. 28

GAME	TIME/SCORE	TV	LOCATION/TICKETS
Texas A&M at South Carolina	Texas A&M 52-28	SEC Network	Williams-Brice Stadium
E. Illinois at Minnesota	Minnesota 42-20	Big Ten Network	TCF Bank Stadium
Presbyterian at Northern Illinois	Northern Illinois 55-3		Huskie Stadium
Missouri St. at Northwestern St.	Missouri State 34-27		Turpin Stadium
Bryant at Stony Brook	Bryant 13-7		
Wake Forest at La.-Monroe	Louisiana-Monroe 17-10	ESPNU	Malone Stadium
Chattanooga at C. Michigan	Central Michigan 20-16		Kelly/Shorts Stadium
Howard at Akron	Akron 41-0		InfoCision Stadium - Summa Field
Charlotte at Campbell	Charlotte 33-9		Barker-Lane Stadium
Reinhardt at Mercer	Mercer 45-42		Moye Complex
E. Kentucky at Robert Morris	Eastern Kentucky 29-10		Joe Walton Stadium
Point U at Charleston So.	Charleston Southern 61-9		CSU Field
Missouri Baptist at SE Missouri St.	Southeast Missouri State 77-0		Houck Stadium
Idaho State at Utah	Utah 56-14	PAC-12 Network	Rice Eccles Stadium
Valparaiso at W. Illinois	Western Illinois 45-6		Hanson Field
Boise St. at Ole Miss	Ole Miss 35-13	ESPN	Georgia Dome
Kentucky Chr. at Tenn. Tech	Tennessee Tech 33-7		Tucker Stadium



CBSSPORTS.COM SHOP

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

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

  - Diagonal degree matrix $D = diag(A1)$

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

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

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

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

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

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

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

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

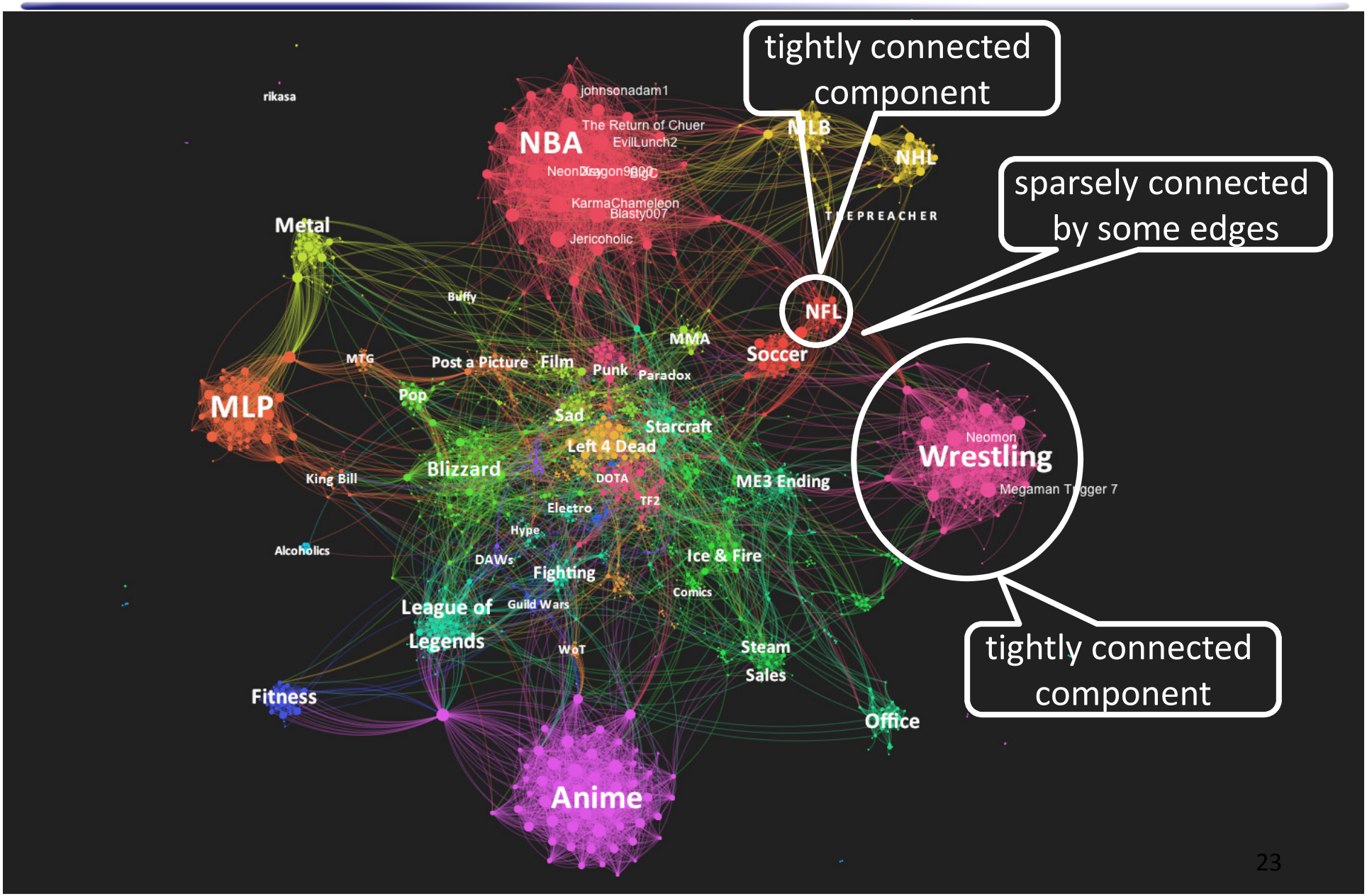
What if the graph has k components

- 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 …

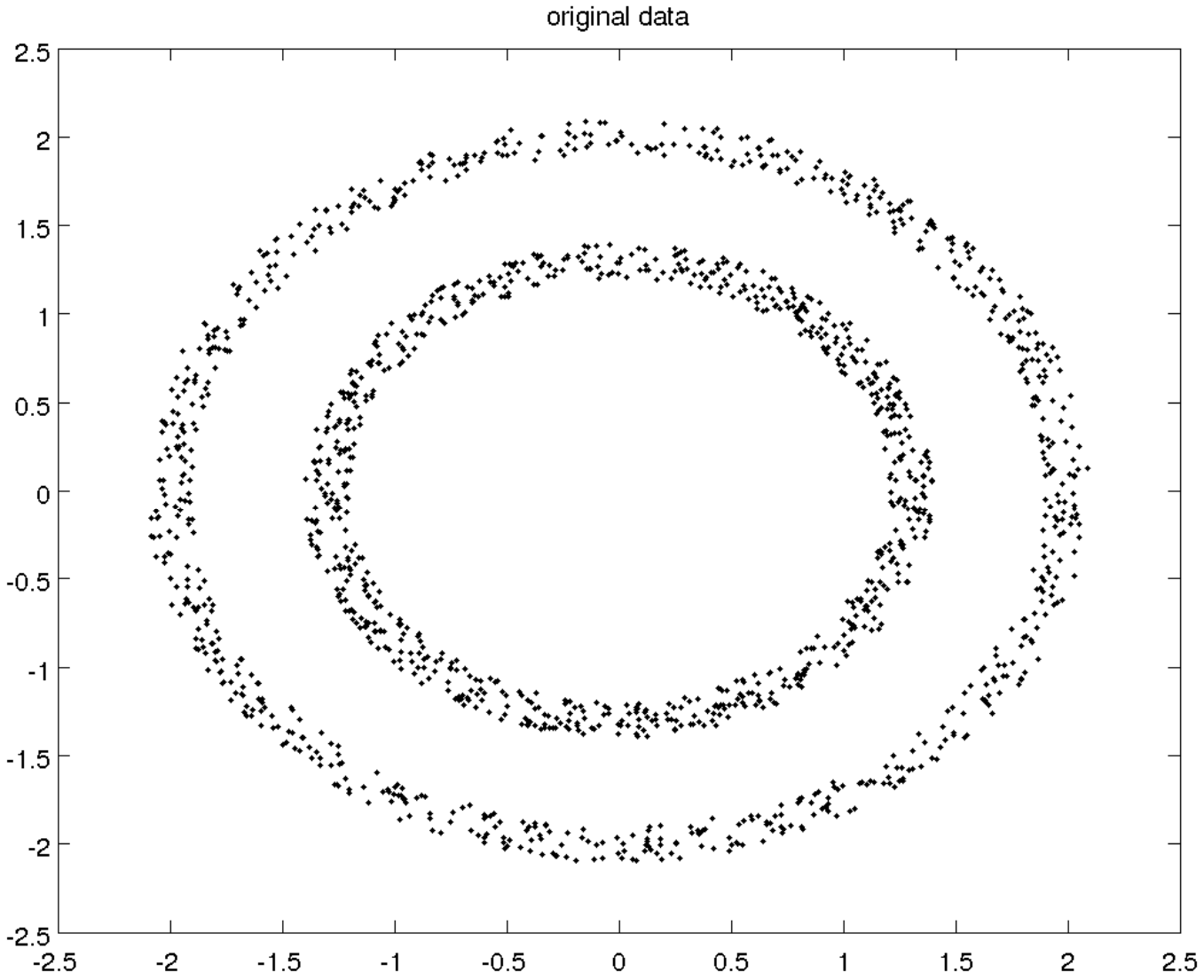
Summary of spectral clustering

- 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, $v^1, v^2, \ldots, v^k$, 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

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?

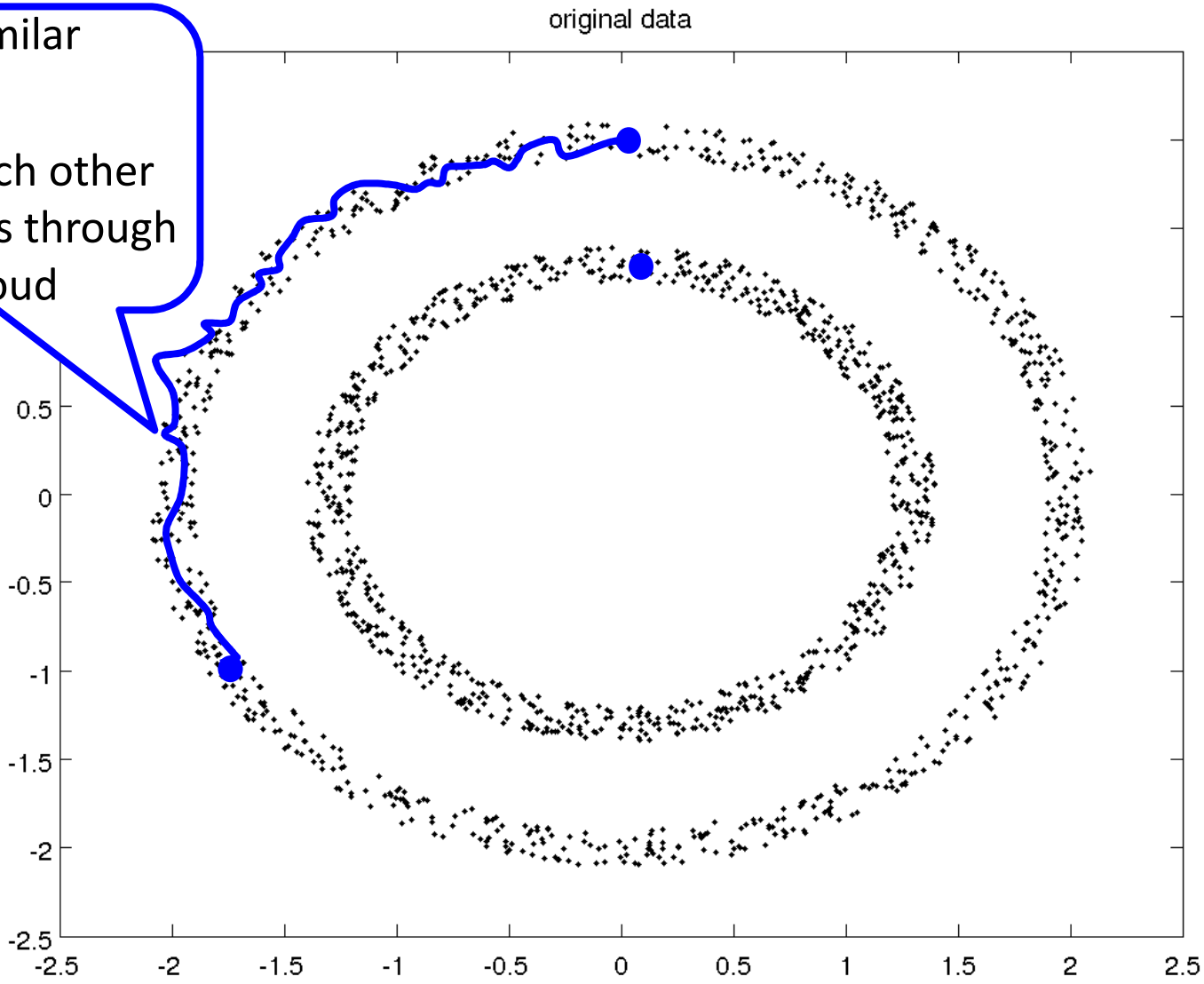


What's a reasonable similarity measure?

points similar

=

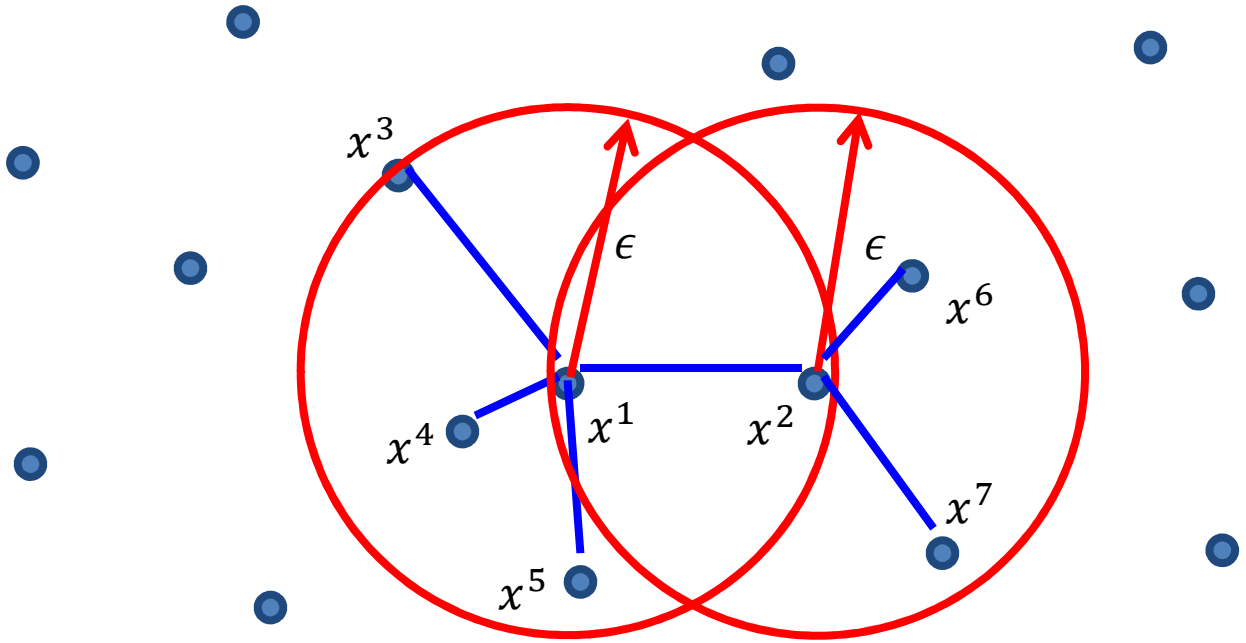
can reach each other by small jumps through data cloud



Nearest neighbor graph

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

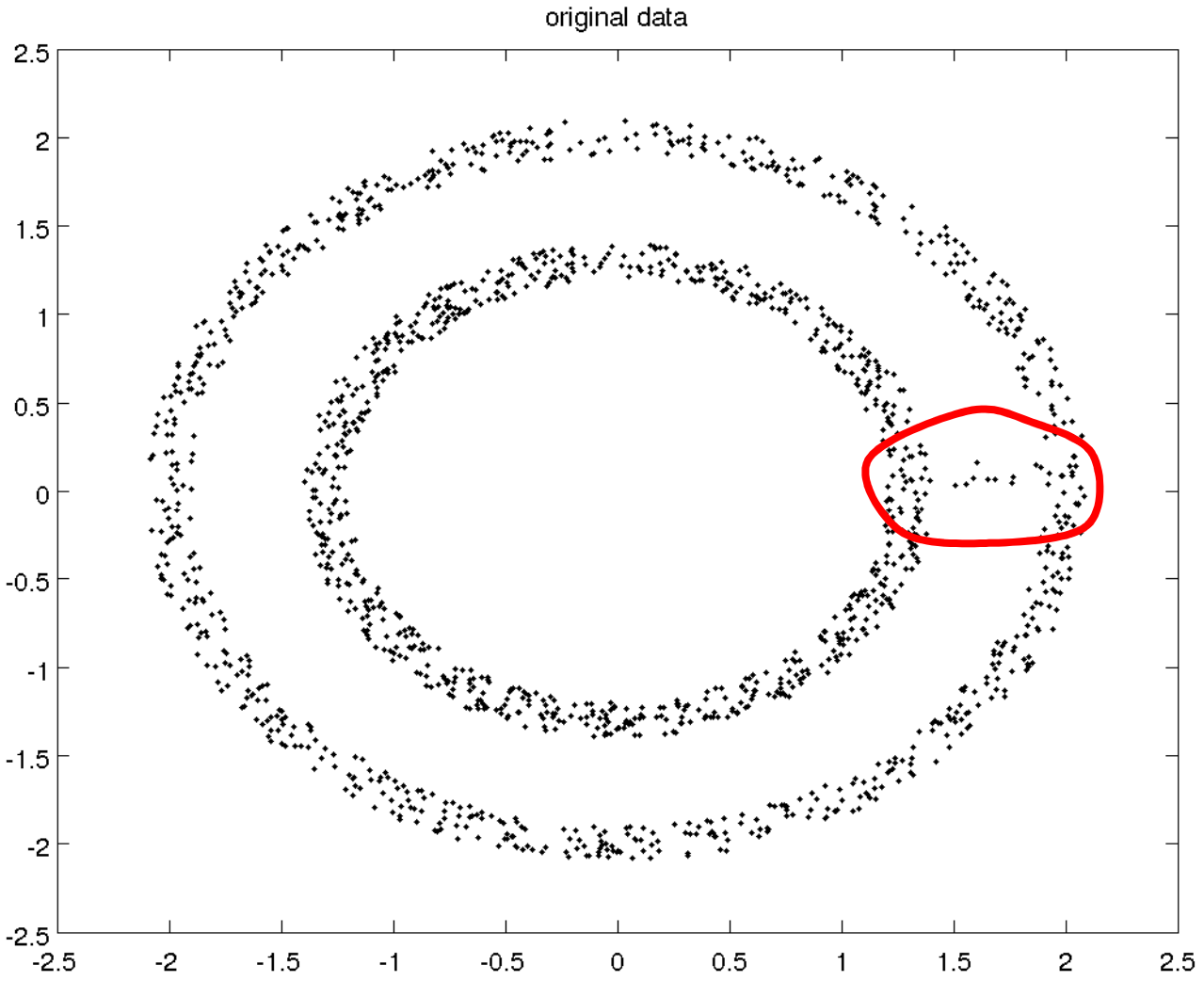
$$A^{ij} = \begin{cases} 1, \text{if } \left\| x^i - x^j \right\| \leq \epsilon \\ 0, \qquad \text{otherwise} \end{cases}$$



Spectral clustering for vectorial 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 (Ng et al.)

- Given m data points (nodes), $\{x^1, x^2, \ldots x^m\} \in R^n$
- 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 $D = diag(A1)$
- Compute k eigenvectors, $v^1, v^2, \ldots, v^k$, of B corresponding to the k largest eigenvalues
- Use $z^1 = \left(v_1^1, v_1^2, \ldots, v_1^k\right), z^2 = \left(v_2^1, v_2^2, \ldots, v_2^k\right)$ … as the new coordinates for data point 1, 2, …, and then run kmeans on these new coordinates