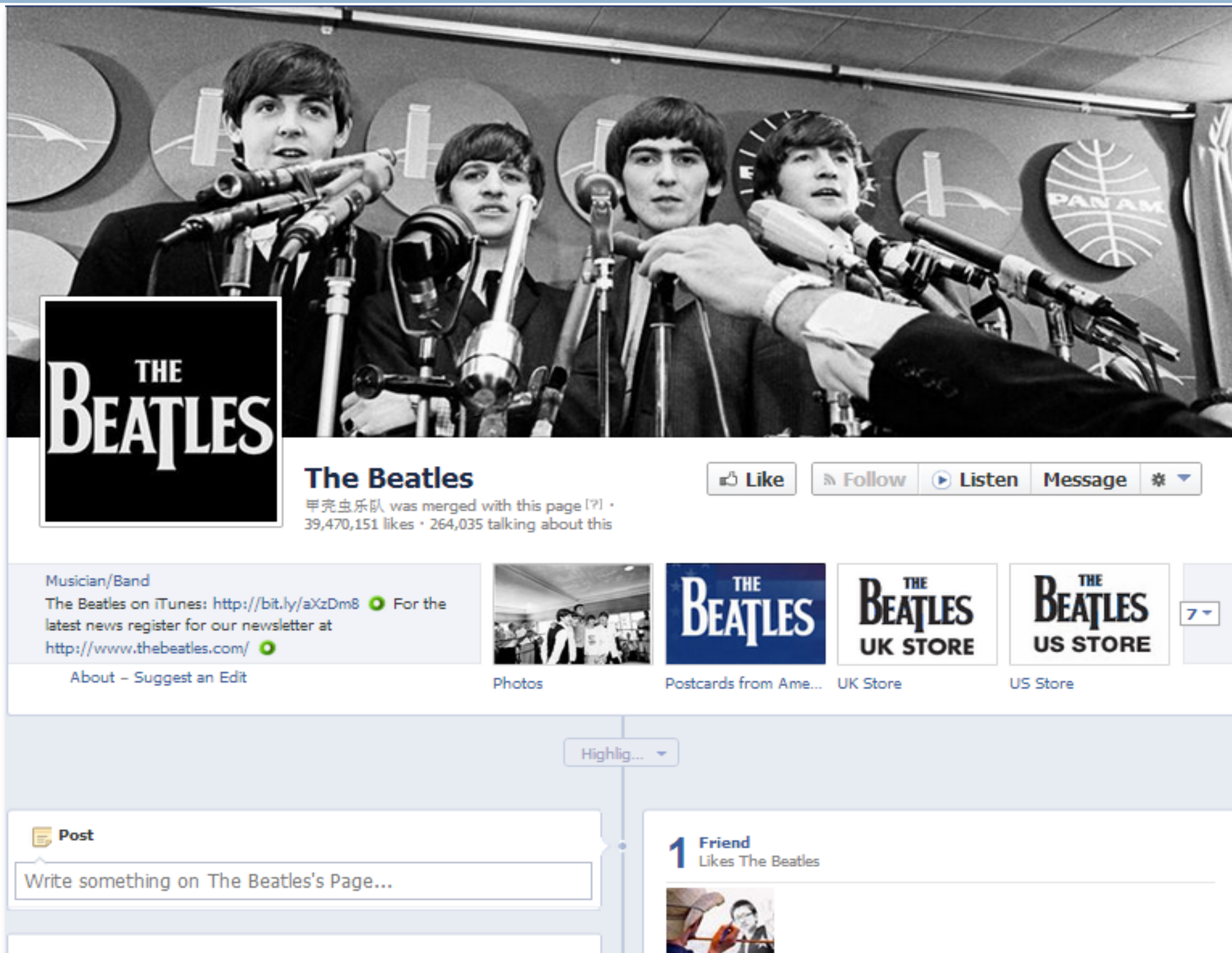
Graph as Linear algebra

Spectral Clustering and Page Rank

INTRODUCTION

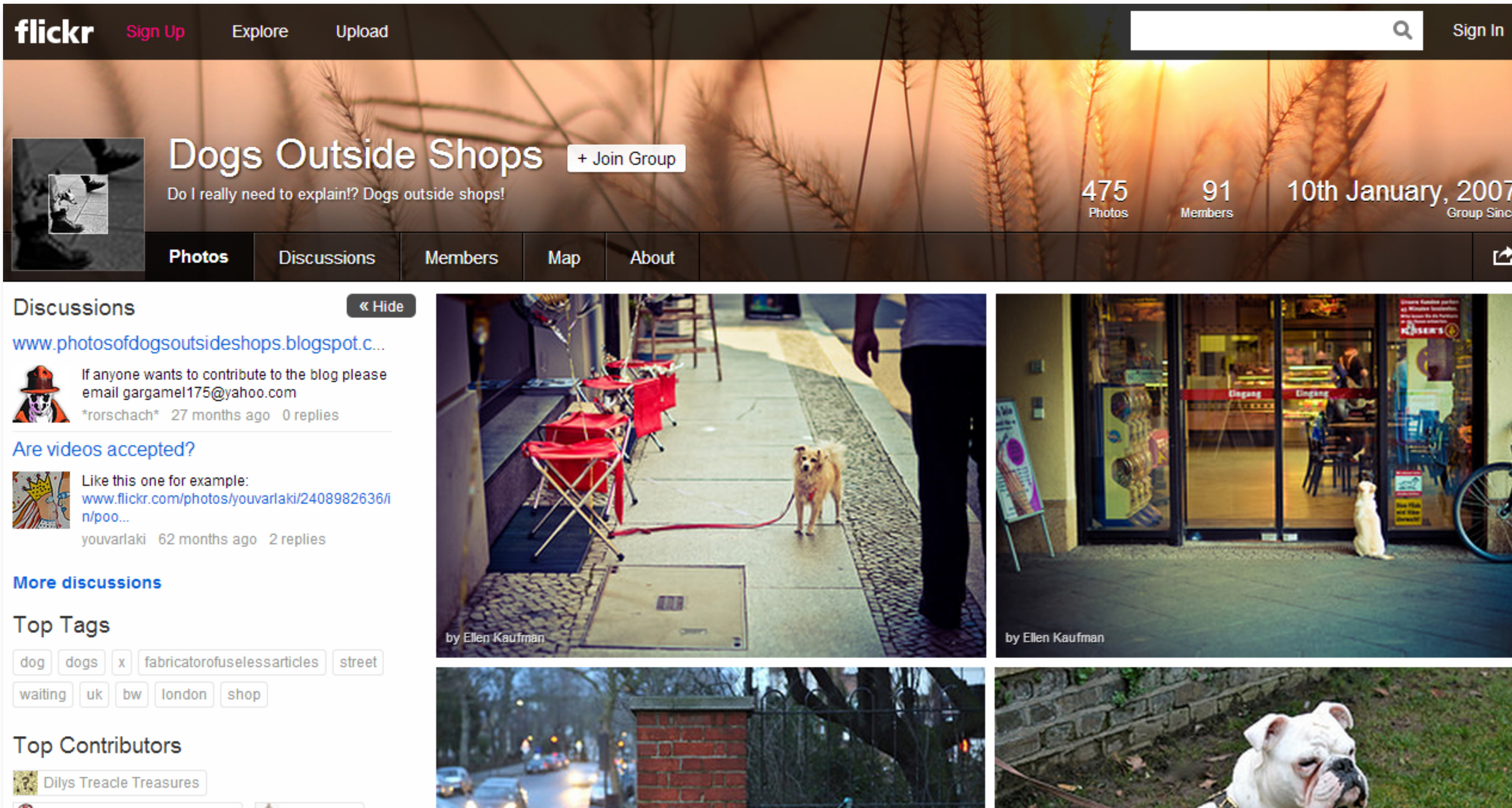
-BY HONG HANDE

Facebook Group



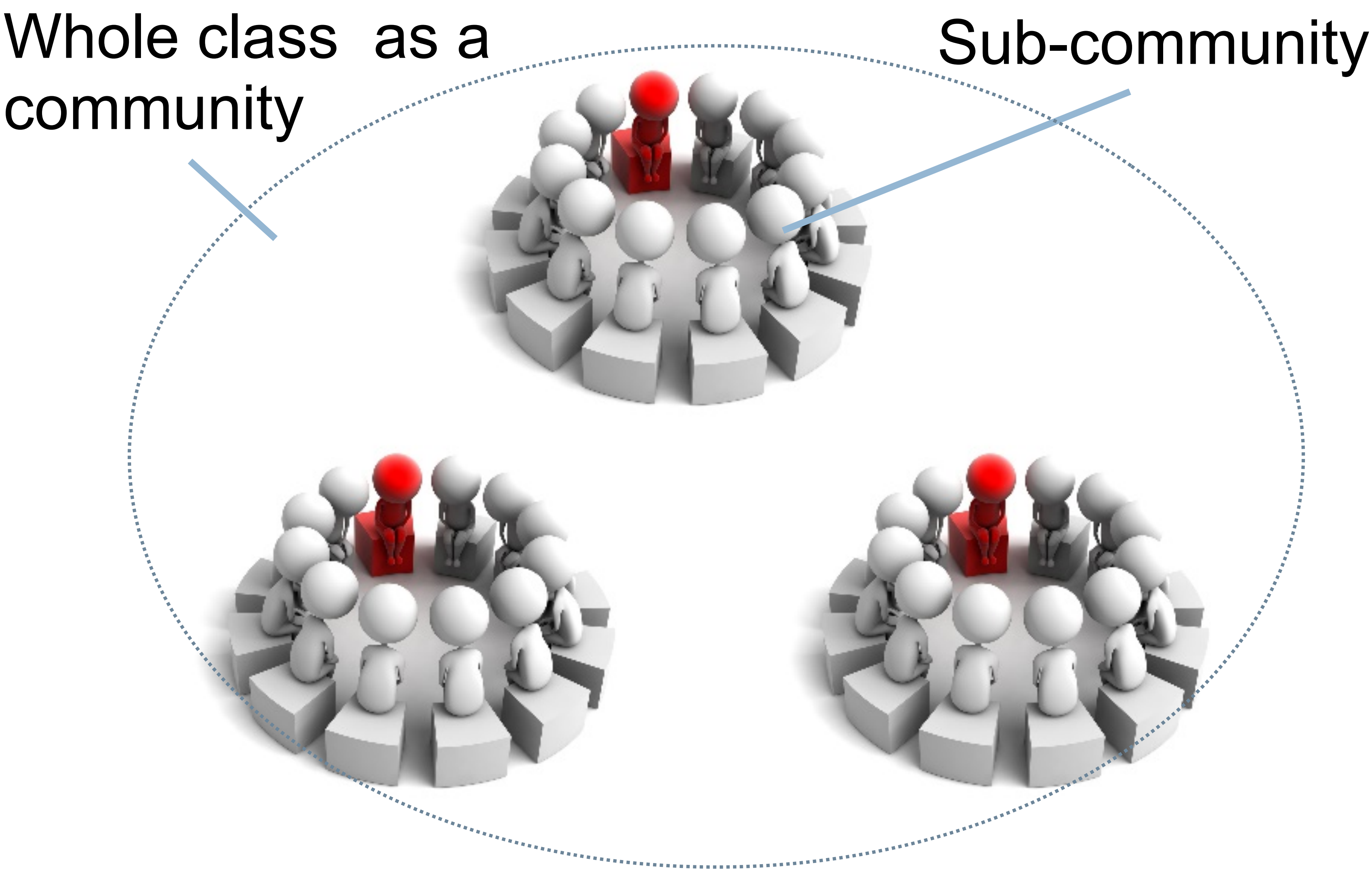
https://www.facebook.com/thebeatles?rf=111113312246958

Flickr group



http://www.flickr.com/groups/49246928@N00/pool/with/417646359/#photo_417646359

Math UA-Linear Algebra



Graph construction from web data(1)

Webpage www.x.com

href = “www.y.com”

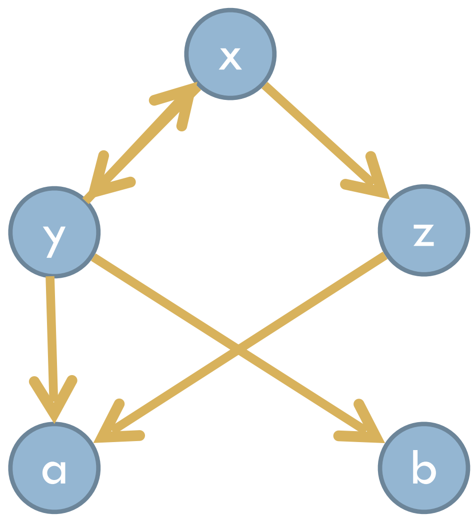
href = “www.z.com”

Webpage www.y.com

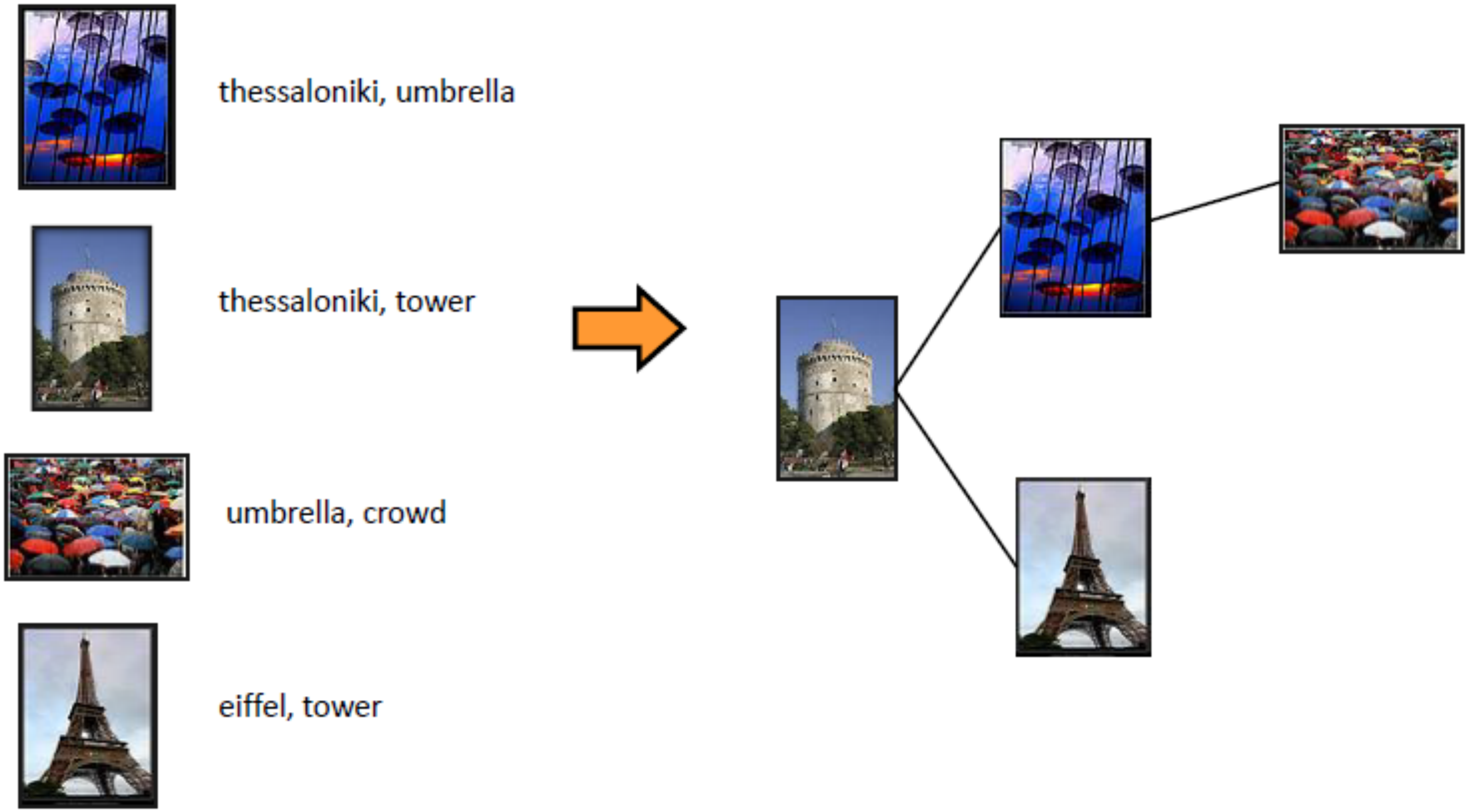
href = “www.x.com”
href = “www.a.com”
href = “www.b.com”

Webpage www.z.com

href = “www.a.com”



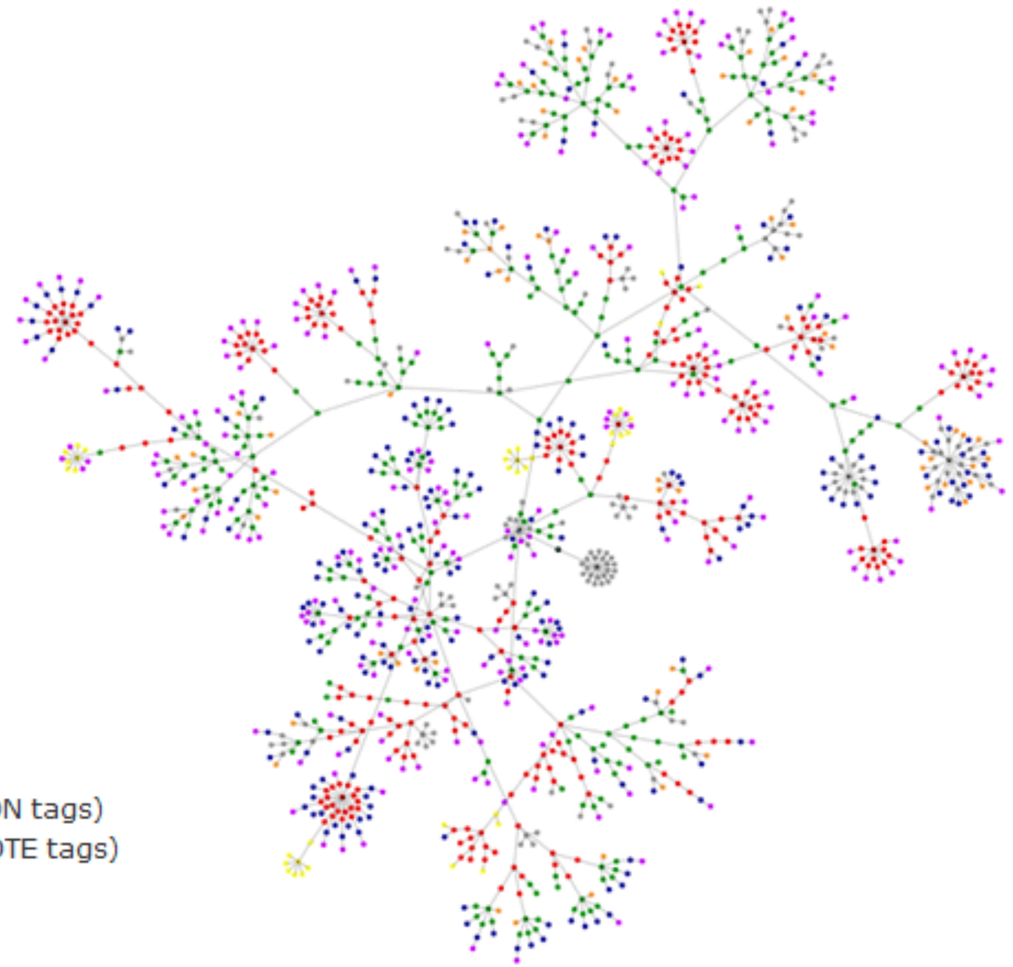
Graph construction from web data(2)



Web pages as a graph

Cnn.com

Lots of links, lots of images. (1316 tags)



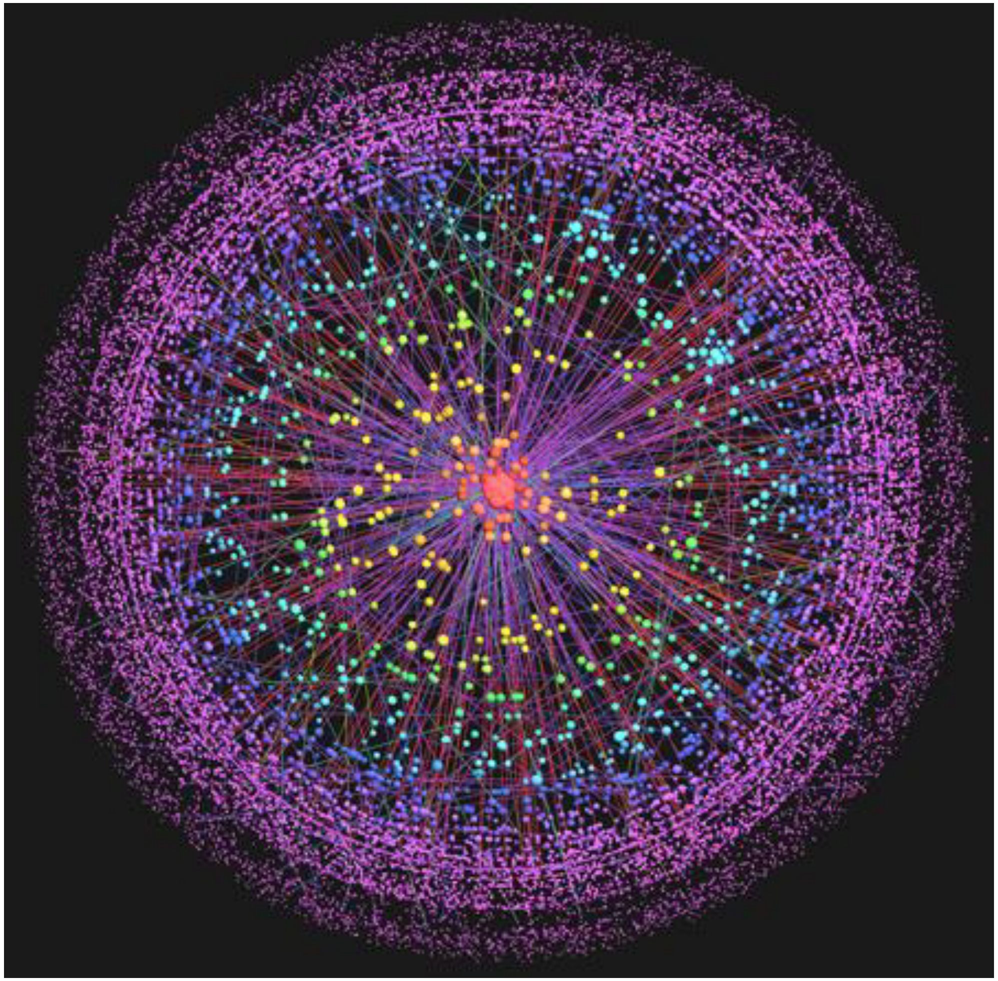
blue: for links (the A tag)
red: for tables (TABLE, TR and TD tags)
green: for the DIV tag
violet: for images (the IMG tag)
yellow: for forms (FORM, INPUT, TEXTAREA, SELECT and OPTION tags)
orange: for linebreaks and blockquotes (BR, P, and BLOCKQUOTE tags)
black: the HTML tag, the root node
gray: all other tags

http://www.aharef.info/2006/05/websites_as_graphs.htm

Internet as a graph

nodes = service providers
edges = connections

hierarchical structure



S. Carmi,S. Havlin, S. Kirkpatrick, Y. Shavitt, E. Shir. A model of Internet topology using k-shell decomposition. PNAS 104 (27), pp. 11150-11154, 2007

Emerging structures

- Graph (from web, daily life) present certain structural characteristics

- Group of nodes interacting with each other

  → Dense inter-connections

  → functional/topical associations

Community

a.k.a. group, subgroup, module, cluster

Community Types

- Explicit
  - The result of conscious human decision
- Implicit
  - Emerging from the interactions & activities of users
  - Need special methods to be discovered

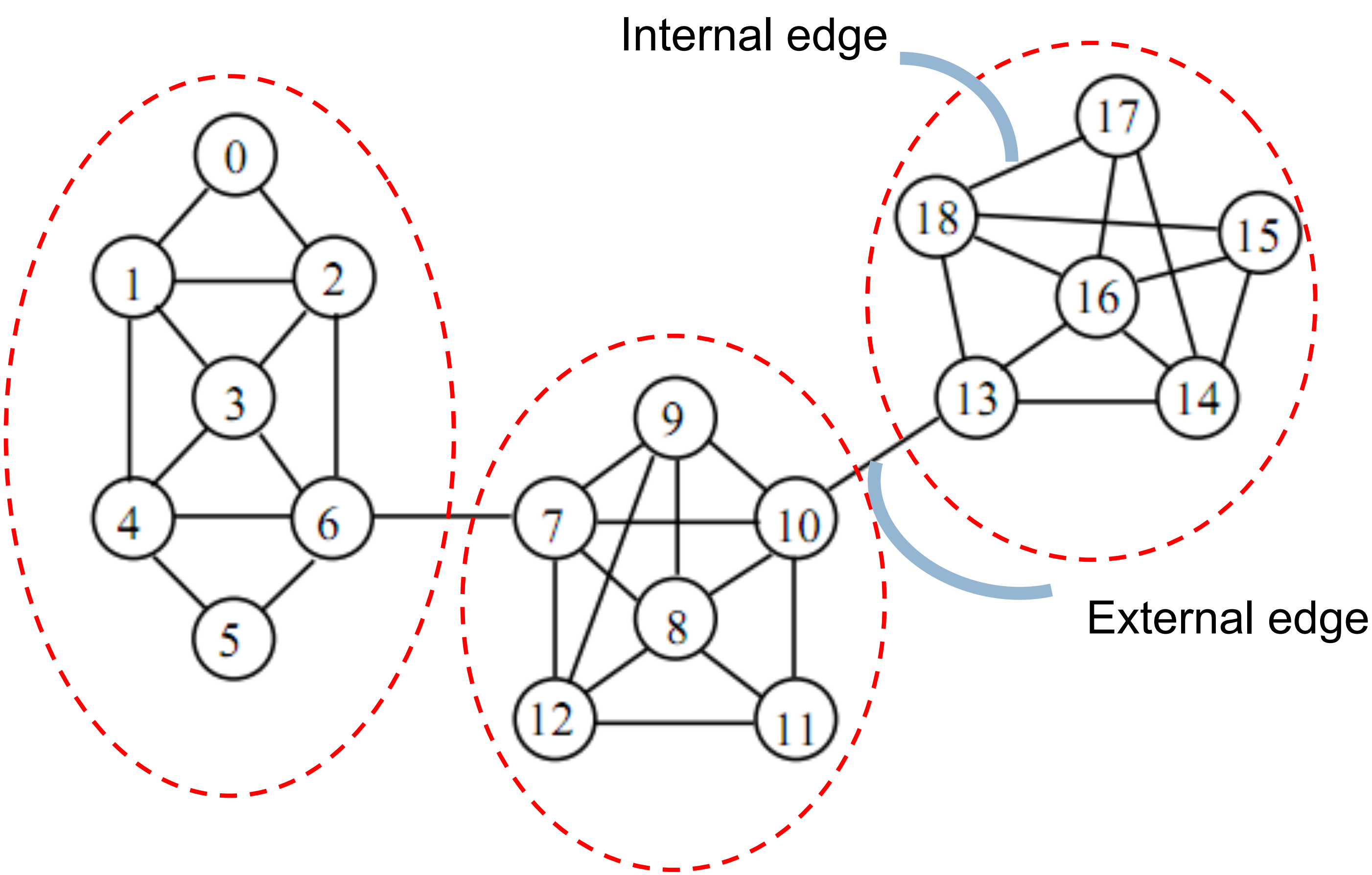
Defining Communities

- Often communities are defined with respect to a graph, G = (V,E) representing a set of objects (V) and their relations (E).

- Even if such graph is not explicit in the raw data, it is usually possible to construct, e.g. feature → vectors → distances → graph

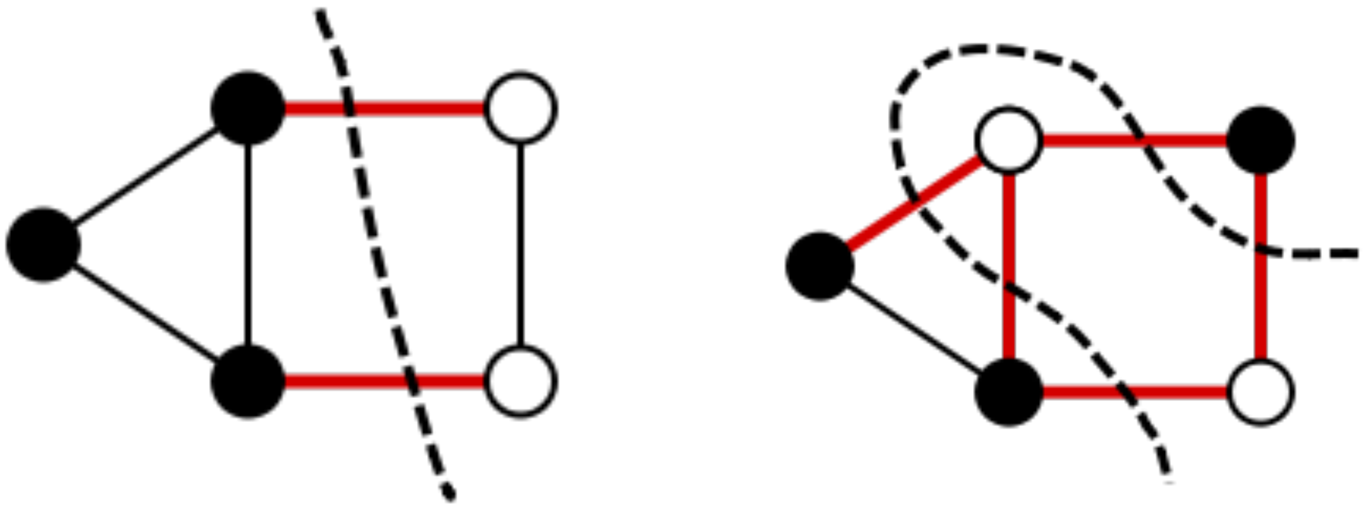
Communities and graphs

- Given a graph, a community is defined as a set of nodes that are more densely connected to each other than to the rest of the network nodes



Graph cuts

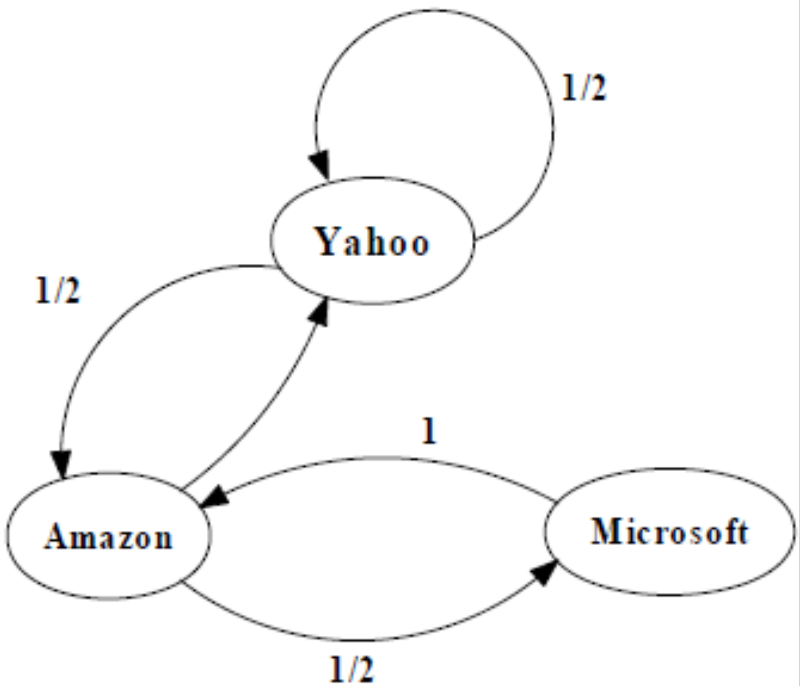
- A cut is a partition of the vertices of a graph into two disjoint subsets.



- The cut-set of the cut is the set of edges whose end points are in different subsets of the partition.

PAGE RANK

An example of Simplified PageRank



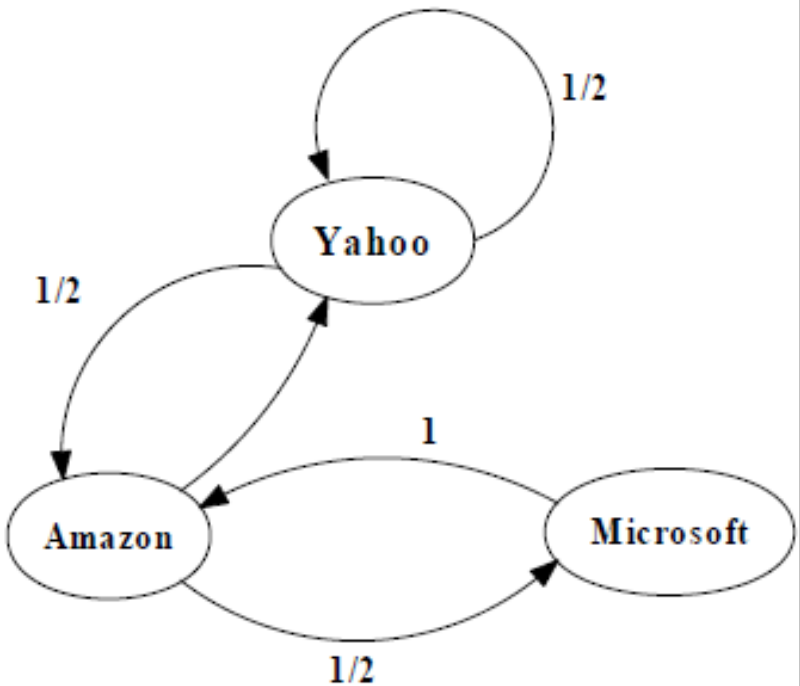
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

PageRank Calculation: first iteration

An example of Simplified PageRank



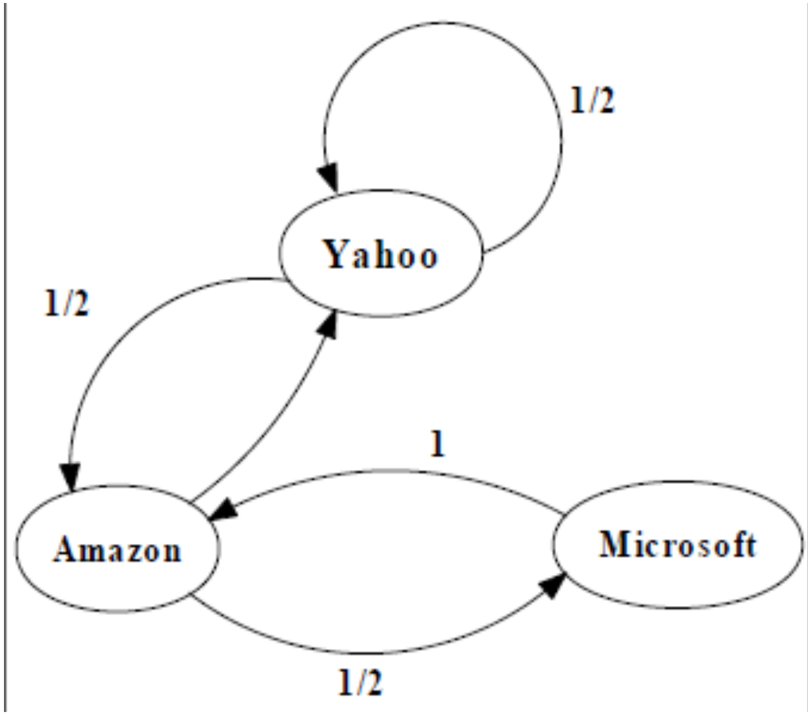
$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}$$

PageRank Calculation: second iteration

An example of Simplified PageRank



$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 17/48 \\ 11/48 \end{bmatrix} \ldots \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}$$

Convergence after some iterations

Converge to eigenvectors!

- Simplest method for computing one eigenvalue-eigenvector pair is *power iteration*, which repeatedly multiplies matrix times initial starting vector
- Assume A has unique eigenvalue of maximum modulus, say λ_1, with corresponding eigenvector v_1
- Then, starting from nonzero vector x_0, iteration scheme

$$x_k = Ax_{k-1}$$

converges to multiple of eigenvector v_1 corresponding to *dominant* eigenvalue λ_1

Convergence of Power iteration

- To see why power iteration converges to dominant eigenvector, express starting vector x_0 as linear combination

$$x_0 = \sum_{i=1}^{n} \alpha_i v_i$$

where v_i are eigenvectors of A

- Then

$$x_k = Ax_{k-1} = A^2 x_{k-2} = \cdots = A^k x_0 =$$

$$\sum_{i=1}^{n} \lambda_i^k \alpha_i v_i = \lambda_1^k \left(\alpha_1 v_1 + \sum_{i=2}^{n} (\lambda_i/\lambda_1)^k \alpha_i v_i \right)$$

- Since $|\lambda_i/\lambda_1| < 1$ for $i > 1$, successively higher powers go to zero, leaving only component corresponding to v_1

$X = \alpha_1 V_1 + \cdots + \alpha_n V_n$

$\Rightarrow A^k x = \alpha_1 \lambda_1^k V_1 + \cdots + \alpha_n \lambda_n^k V_n$

largest eigen value.

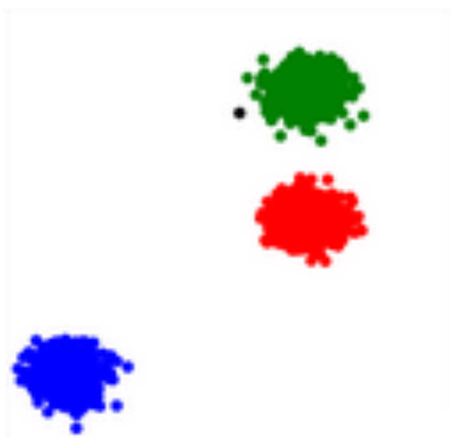
$\alpha_1 \lambda_1^k$ growth much faster

SPECTRAL CLUSTERING

Motivation

Motivation

- Two kinds of clusters

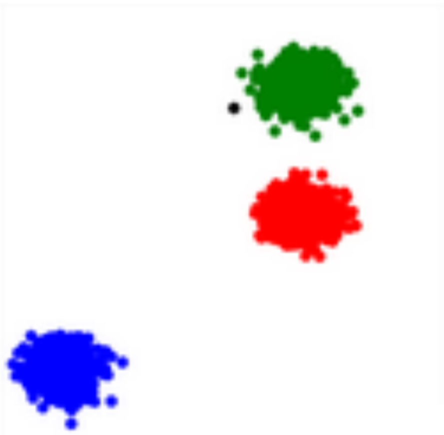


convex shaped

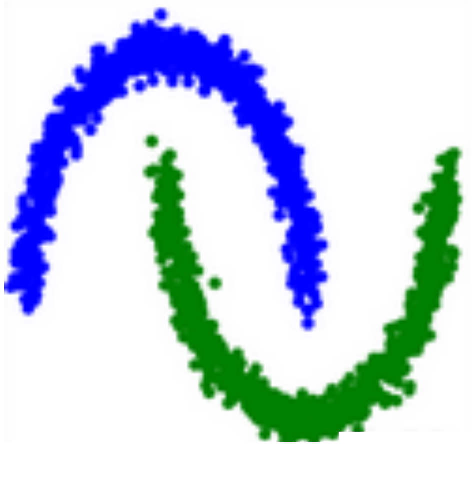
non-convex shaped

Motivation

- Two kinds of clusters
  - convex shaped, compact ⇒ k-means



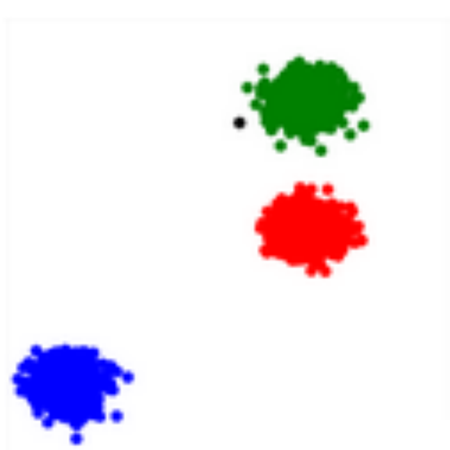
convex shaped



non-convex shaped

Motivation

- Two kinds of clusters
  - convex shaped, compact ⟹ k-means
  - non-convex shaped, connected ⟹ spectral clustering



convex shaped

non-convex shaped

Key Idea

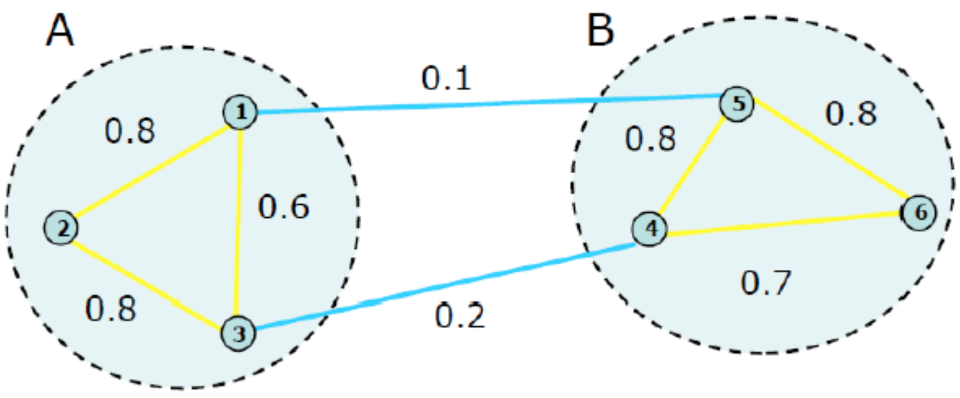
- Project the data points into a new space
- Clusters can be trivially detected in the new space

Key Idea

- Project the data points into a new space
- Clusters can be trivially detected in the new space

- Next, we will cover
  - How to find the new space
  - How to represent data points in the space

Matrix Representations of Graphs



Matrix Representations of Graphs

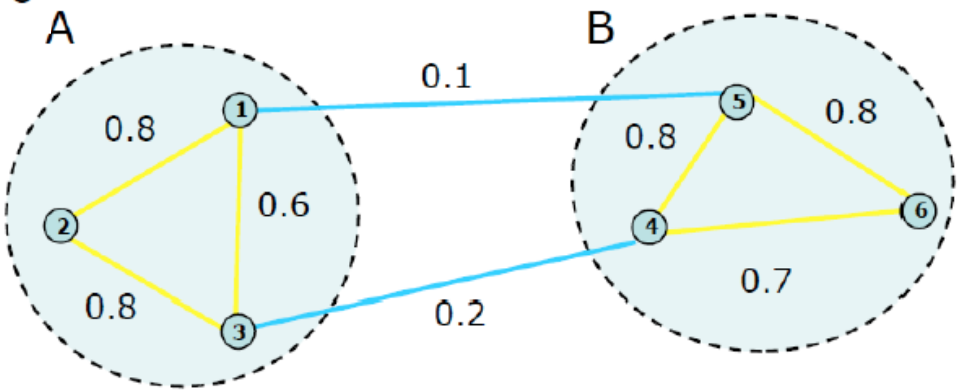
- Adjacency matrix *W*

  $W = (w_{ij})\ i, j = 1, \dots, n \quad w_{ij} \geq 0$

- Degree d_i of a node *i*

  $d_i = \sum_{j=1}^{n} w_{ij}$

- Degree matrix *D*



Diagonal matrix with the degrees $d_1, \dots, d_n$ on the diagonal.

Matrix Representations of Graphs

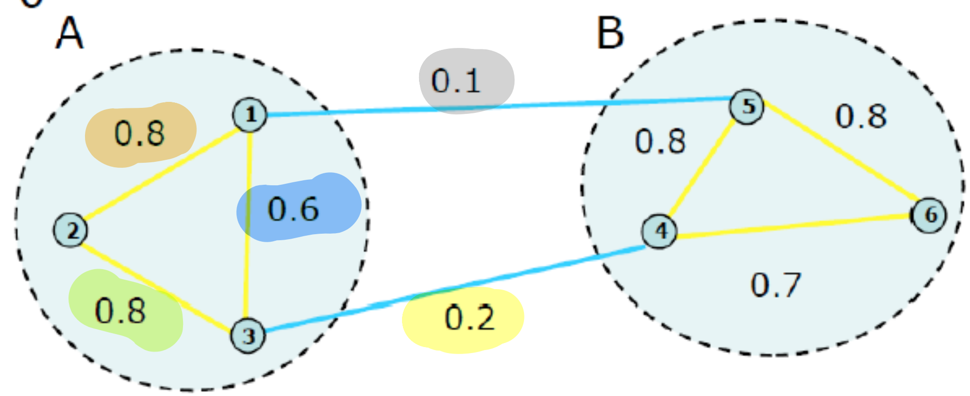
- Adjacency matrix W Symmetric

  $W = (w_{ij})\ i,j = 1, \dots, n \quad w_{ij} \geq 0$

- Degree d_i of a node i

  $d_i = \sum_{j=1}^{n} w_{ij}$

- Degree matrix D



Diagonal matrix with the degrees $d_1, \dots, d_n$ on the diagonal

$$W = \begin{pmatrix} 0 & 0.8 & 0.6 & 0 & 0.1 & 0 \\ 0.8 & 0 & 0.8 & 0 & 0 & 0 \\ 0.6 & 0.8 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0.8 & 0.7 \\ 0.1 & 0 & 0 & 0.8 & 0 & 0.8 \\ 0 & 0 & 0 & 0.7 & 0.8 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 1.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.5 \end{pmatrix}$$

Matrix Representations of Graphs

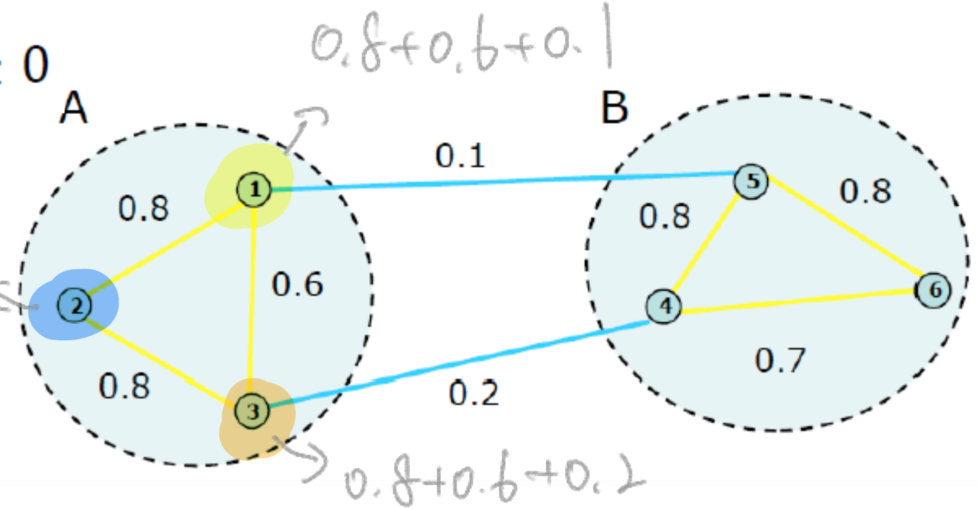
- Adjacency matrix *W*

$$W = (w_{ij})\ i,j = 1,\dots,n \quad w_{ij} \geq 0$$

- Degree d_i of a node *i*

$$d_i = \sum_{j=1}^{n} w_{ij}$$

- Degree matrix *D*



Diagonal matrix with the degrees $d_1, \dots, d_n$ on the diagonal

$$W = \begin{pmatrix} 0 & 0.8 & 0.6 & 0 & 0.1 & 0 \\ 0.8 & 0 & 0.8 & 0 & 0 & 0 \\ 0.6 & 0.8 & 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0.8 & 0.7 \\ 0.1 & 0 & 0 & 0.8 & 0 & 0.8 \\ 0 & 0 & 0 & 0.7 & 0.8 & 0 \end{pmatrix} \qquad D = \begin{pmatrix} 1.5 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1.6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1.6 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.7 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1.7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1.5 \end{pmatrix}$$

Graph Laplacian

- Graph Laplacian

D : Degree matrix
W : Adjacency matrix

$$d_i = \sum_{j=1}^{n} w_{ij}$$

$$L = D - W$$

diag ↗ ↖ symmetric.

← L is a symmetric matrix

⇕

$x^T L x$ is a quadratic function

Graph Laplacian

$$D : Degree\ matrix$$
$$W : Adjacency\ matrix$$
$$d_i = \sum_{j=1}^{n} w_{ij}$$

- Graph Laplacian

$$L = D - W$$

- Next, we will see some properties of *L*, which would be used for spectral clustering

- We will work closely with linear algebra, especially eigenvalues and eigenvectors

Properties of Graph Laplacian (1)

D : Degree matrix
W : Adjacency matrix

$$d_i = \sum_{j=1}^{n} w_{ij} \qquad (1)$$

$$L = D - W \qquad (2)$$

For any vector $f \in \mathbb{R}^n$ we have

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

L is P.S.D

because

$f^T L f$ always larger than 0.

$w_{ij} = 0$

$i = j$

f_i and f_j can be different

$w_{ij} = \square$

i, j connection

we want f_i and f_j are similar

Properties of Graph Laplacian (1)

$D : Degree\ matrix$
$W : Adjacency\ matrix$

$$d_i = \sum_{j=1}^{n} w_{ij} \quad (1)$$

$$L = D - W \quad (2)$$

For any vector $f \in \mathbb{R}^n$ we have

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.$$

Proof:

$$f^T L f = f^T D f - f^T W f$$ apply Equation 2

$$= (f_1, f_2, ..., f_n) \begin{pmatrix} d_{11} & ... & 0 \\ ... & d_{ii} & ... \\ 0 & ... & d_{nn} \end{pmatrix} \begin{pmatrix} f_1 \\ ... \\ f_n \end{pmatrix} - (f_1, f_2, ..., f_n) \begin{pmatrix} w_{11} & ... & w_{1n} \\ ... & w_{ij} & ... \\ w_{n1} & ... & w_{nn} \end{pmatrix} \begin{pmatrix} f_1 \\ ... \\ f_n \end{pmatrix}$$

$$= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

Properties of Graph Laplacian (1)

D : *Degree matrix*
W : *Adjacency matrix*

$$d_i = \sum_{j=1}^{n} w_{ij} \qquad (1)$$

$$L = D - W \qquad (2)$$

For any vector $f \in \mathbb{R}^n$ we have

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.$$

Proof:

$$f^T L f = f^T D f - f^T W f$$

apply Equation 2

$$= (f_1, f_2, ..., f_n) \begin{pmatrix} d_{11} & ... & 0 \\ ... & d_{ii} & ... \\ 0 & ... & d_{nn} \end{pmatrix} \begin{pmatrix} f_1 \\ ... \\ f_n \end{pmatrix} - (f_1, f_2, ..., f_n) \begin{pmatrix} w_{11} & ... & w_{1n} \\ ... & w_{ij} & ... \\ w_{n1} & ... & w_{nn} \end{pmatrix} \begin{pmatrix} f_1 \\ ... \\ f_n \end{pmatrix}$$

$$= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2} \left(\underline{\sum_{i=1}^{n} d_i f_i^2} - 2 \sum_{i,j=1}^{n} f_i f_j w_{ij} + \underline{\sum_{j=1}^{n} d_j f_j^2} \right)$$

Properties of Graph Laplacian (1)

D : *Degree matrix*
W : *Adjacency matrix*

$$d_i = \sum_{j=1}^{n} w_{ij} \quad (1)$$

$$L = D - W \quad (2)$$

For any vector $f \in \mathbb{R}^n$ we have

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij}(f_i - f_j)^2.$$

Proof:

$$f^T L f = f^T D f - f^T W f$$ apply Equation 2

$$= (f_1, f_2, ..., f_n) \begin{pmatrix} d_{11} & ... & 0 \\ ... & d_{ii} & ... \\ 0 & ... & d_{nn} \end{pmatrix} \begin{pmatrix} f_1 \\ ... \\ f_n \end{pmatrix} - (f_1, f_2, ..., f_n) \begin{pmatrix} w_{11} & ... & w_{1n} \\ ... & w_{ij} & ... \\ w_{n1} & ... & w_{nn} \end{pmatrix} \begin{pmatrix} f_1 \\ ... \\ f_n \end{pmatrix}$$

$$= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2} \left(\underline{\sum_{i=1}^{n} d_i f_i^2} - 2 \sum_{i,j=1}^{n} f_i f_j w_{ij} + \underline{\sum_{j=1}^{n} d_j f_j^2} \right)$$

$$= \frac{1}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} f_i^2 - 2 \sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} \sum_{i=1}^{n} w_{ij} f_j^2 \right)$$ apply Equation 1

Properties of Graph Laplacian (1)

D : *Degree matrix*
W : *Adjacency matrix*

$$d_i = \sum_{j=1}^{n} w_{ij} \quad (1)$$

$$L = D - W \quad (2)$$

For any vector $f \in \mathbb{R}^n$ we have

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2.$$

Proof:

$$f^T L f = f^T D f - f^T W f$$ apply Equation 2

$$= (f_1, f_2, ..., f_n) \begin{pmatrix} d_{11} & ... & 0 \\ ... & d_{ii} & ... \\ 0 & ... & d_{nn} \end{pmatrix} \begin{pmatrix} f_1 \\ ... \\ f_n \end{pmatrix} - (f_1, f_2, ..., f_n) \begin{pmatrix} w_{11} & ... & w_{1n} \\ ... & w_{ij} & ... \\ w_{n1} & ... & w_{nn} \end{pmatrix} \begin{pmatrix} f_1 \\ ... \\ f_n \end{pmatrix}$$

$$= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} f_i f_j w_{ij}$$

$$= \frac{1}{2} \left(\underline{\sum_{i=1}^{n} d_i f_i^2} - 2 \sum_{i,j=1}^{n} f_i f_j w_{ij} + \underline{\sum_{j=1}^{n} d_j f_j^2} \right)$$

$$= \frac{1}{2} \left(\sum_{i=1}^{n} \sum_{j=1}^{n} w_{ij} f_i^2 - 2 \sum_{i,j=1}^{n} f_i f_j w_{ij} + \sum_{j=1}^{n} \sum_{i=1}^{n} w_{ij} f_j^2 \right)$$ apply Equation 1

$$= \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

Properties of Graph Laplacian (2)

The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector $\mathbb{1}$

D : Degree matrix
W : Adjacency matrix

$$d_i = \sum_{j=1}^{n} w_{ij} \qquad (1)$$

$$L = D - W \qquad (2)$$

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

if f is all one vector

$$\Rightarrow f^T L f = 0$$

Properties of Graph Laplacian (2)

The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector $\mathbb{1}$

D : *Degree matrix*
W : *Adjacency matrix*

$$d_i = \sum_{j=1}^{n} w_{ij} \quad (1)$$

$$L = D - W \quad (2)$$

Let λ be an eigenvalue of L, and v be the corresponding eigenvector, then $Lv = \lambda v$.

Properties of Graph Laplacian (2)

D : *Degree matrix*
W : *Adjacency matrix*

$$d_i = \sum_{j=1}^{n} w_{ij} \quad (1)$$

$$L = D - W \quad (2)$$

The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector $\mathbb{1}$

Let λ be an eigenvalue of L, and v be the corresponding eigenvector, then $Lv = \lambda v$.

Proof:

From Property 1, $f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2 \geqslant 0 \; \forall f$,

then suppose $Lv = \lambda v$, we have $v^T L v = v^T \lambda v = \lambda \sum_{i=1}^{n} v_i^2 \geqslant 0$.

Thus the smallest eigenvalue is 0.

Properties of Graph Laplacian (2)

The smallest eigenvalue of L is 0, the corresponding eigenvector is the constant one vector $\mathbb{1}$

$D: Degree\ matrix$
$W: Adjacency\ matrix$

$$d_i = \sum_{j=1}^{n} w_{ij} \quad (1)$$

$$L = D - W \quad (2)$$

Let λ be an eigenvalue of L, and v be the corresponding eigenvector, then $Lv = \lambda v$.

Proof:

From Property 1, $f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2 \geqslant 0 \; \forall f$,

then suppose $Lv = \lambda v$, we have $v^T L v = v^T \lambda v = \lambda \sum_{i=1}^{n} v_i^2 \geqslant 0$.

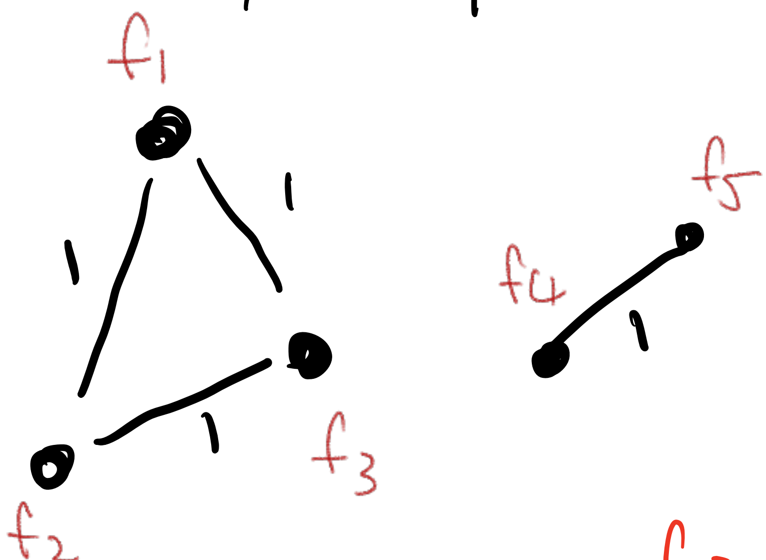
Thus the smallest eigenvalue is 0.

$$L \cdot \mathbb{1} = (D - W)\mathbb{1} = D\mathbb{1} - W\mathbb{1} = \left(d_i - \sum_{j=1}^{n} w_{ij} \right)_i = \mathbb{0} = 0 \cdot \mathbb{1}$$

Thus the corresponding eigenvector is the constant vector.

We Have Done So Many Works…

If My Graph is



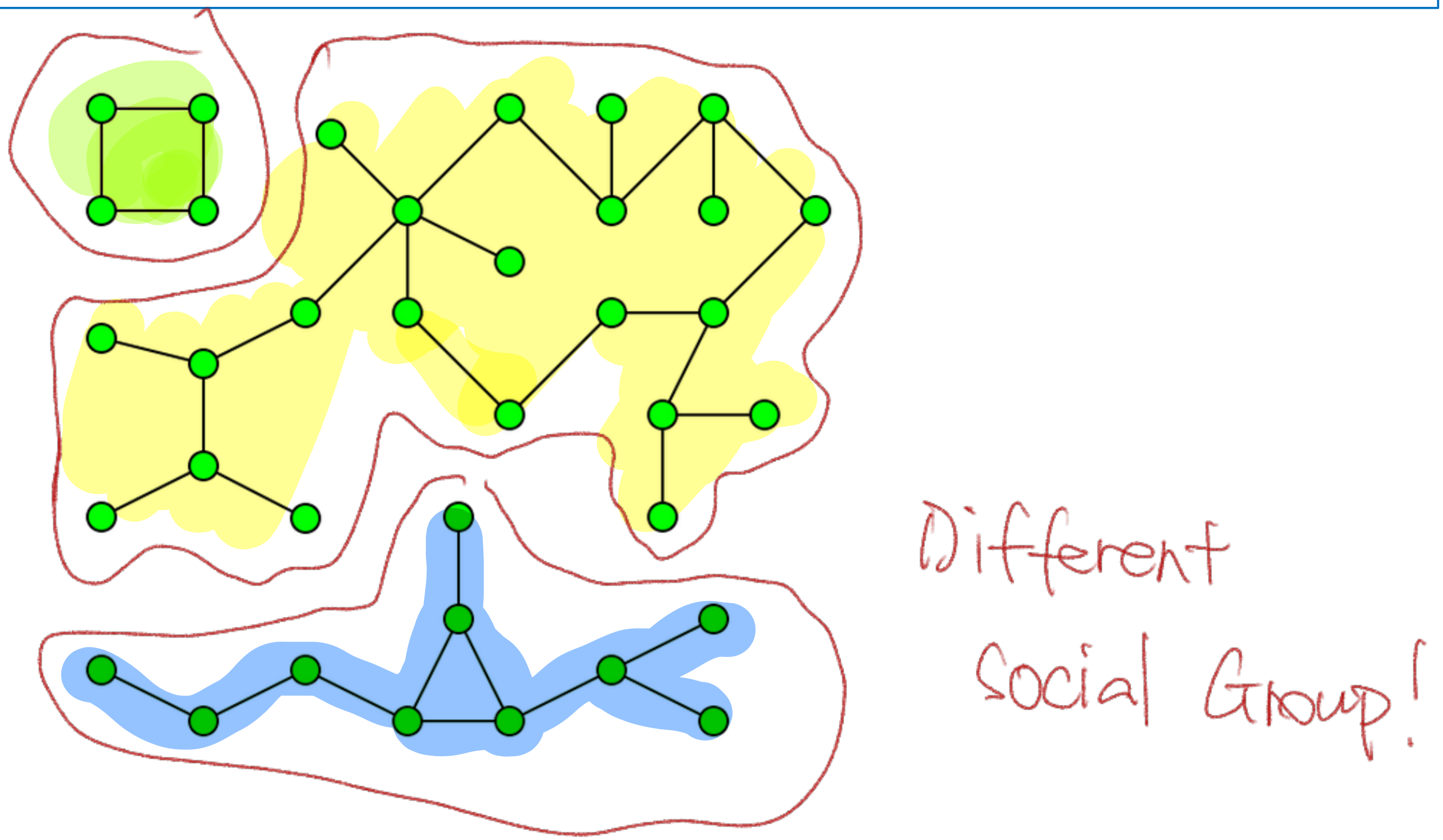
is also a 0-th eigenvector

$f_1 = f_2 = f_3 = 1$ — $f_4 = f_5 = 0$

$$f^T L f = (f_1 - f_2)^2 + (f_2 - f_3)^2 + (f_3 - f_1)^2 + (f_4 - f_5)^2$$

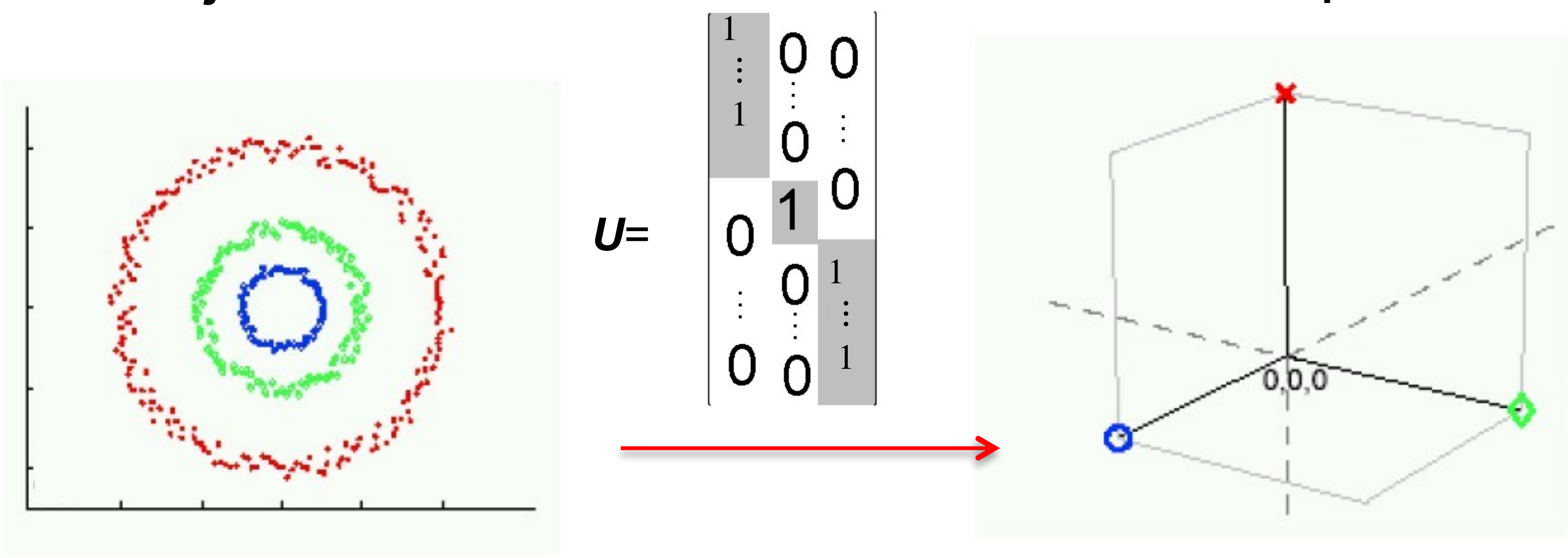
Number of Connected Components & Eigenvalues of *L*

a connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph



Why Spectral Clustering Works?(2)

- Consider an ideal case
  - Let the three eigenvectors be three columns of a matrix ***U***.
  - Project the rows in ***U*** to a 3-dimensional space.



We Have Done So Many Works…

- Transform the graph to Laplacian L

We Have Done So Many Works…

- Transform the graph to Laplacian *L*

- Study the properties of *L*, basically the eigenvalues and eigenvectors

We Have Done So Many Works…

- Transform the graph to Laplacian *L*
- Study the properties of *L*, basically the eigenvalues and eigenvectors
- Finally, we can see the relationship between the graph and the eigenvalues!

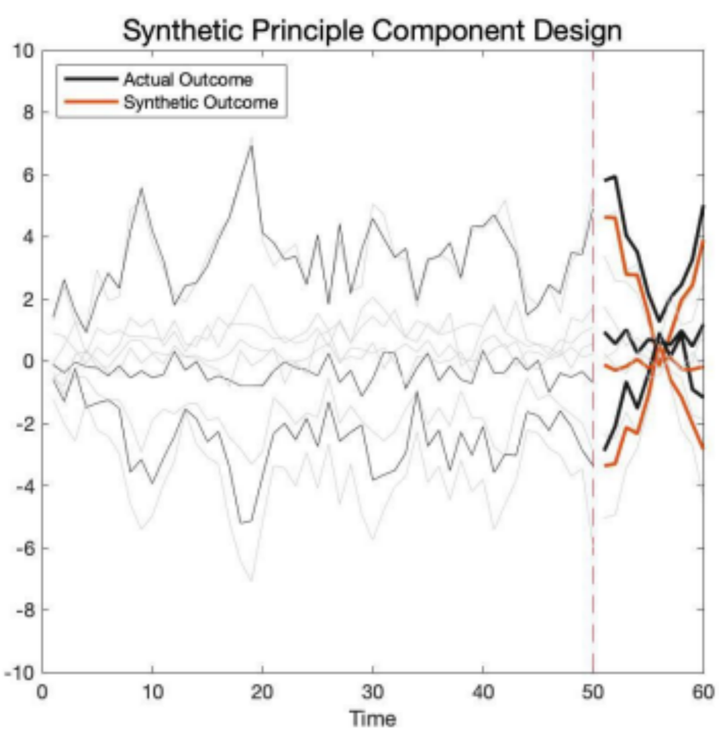
Applications: Social Media

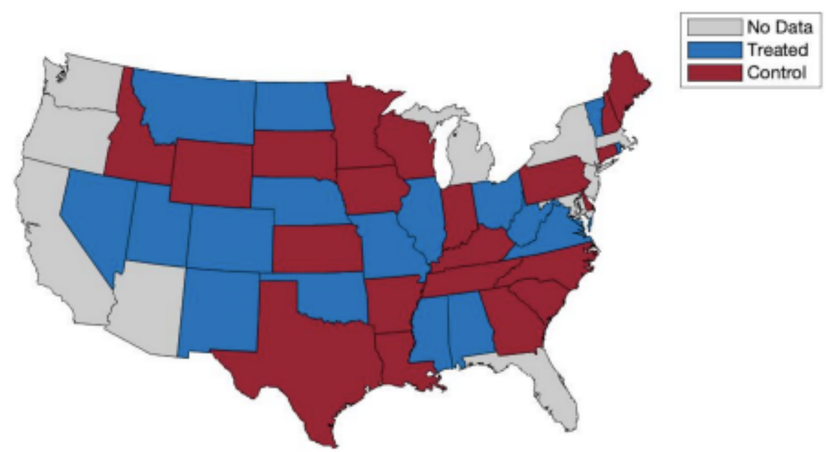
NEXT
NUS - Extreme - Tsinghua

http://next.comp.nus.edu.sg

Smallest eigenvectors means…

~~Largest~~ Smallest eigenvectors separate data to two distance class, so ~~smallest~~ largest! eigenvectors will separate data to similar groups.
Consider if you want to test a vaccine or a marketing policy….



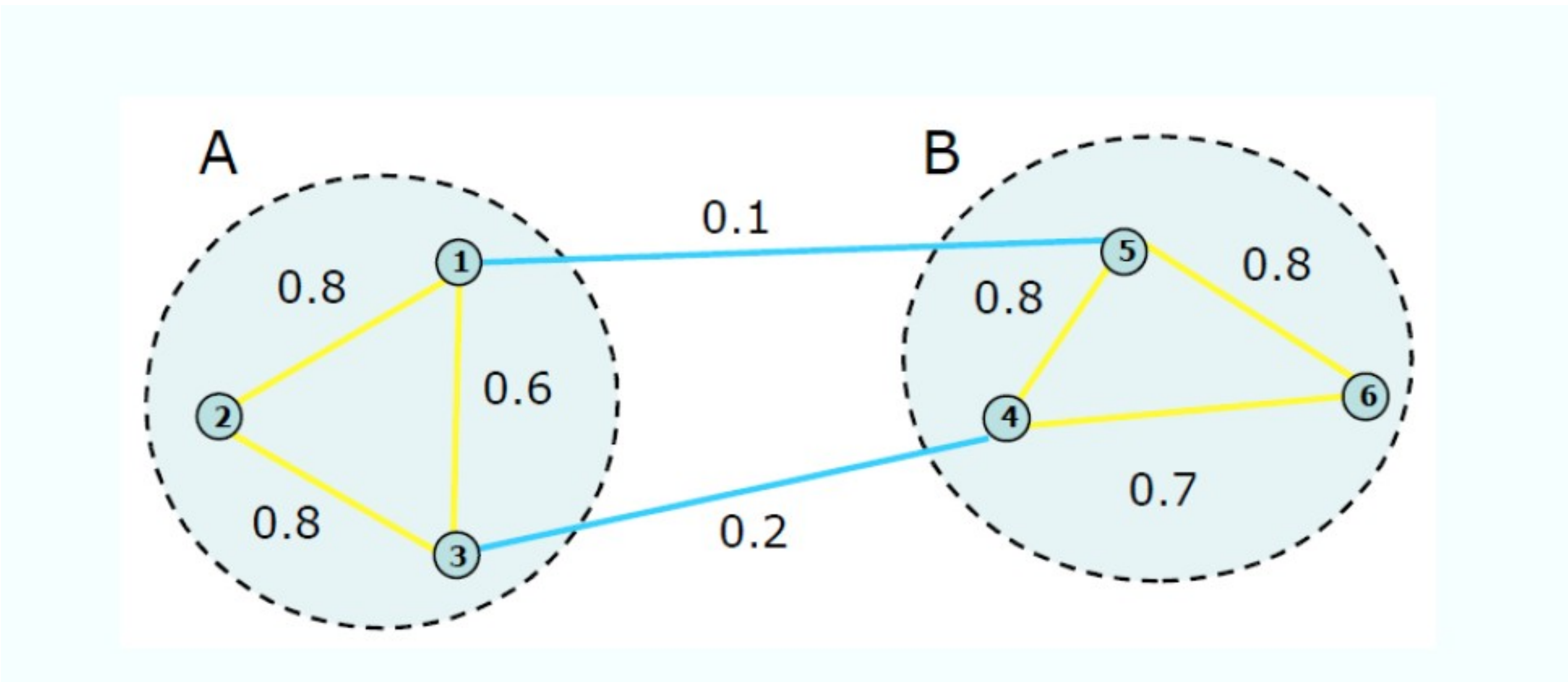


(a) Treatment Selected when T = 25

This is my paper! https://arxiv.org/pdf/2211.15241.pdf

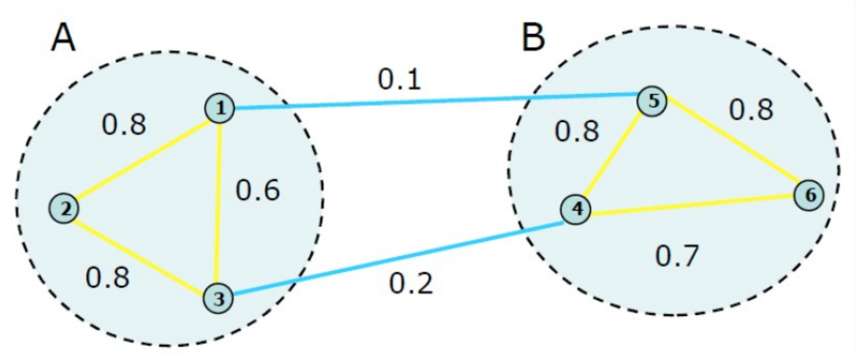
Example(1)

- Now let's go through an example.
- n = 6, k=2



Example(2)

- Step 1: Weighted adjacency matrix ***W*** and degree matrix ***D***



	X_1	X_2	X_3	X_4	X_5	X_6
X_1	0	0.8	0.6	0	0.1	0
X_2	0.8	0	0.8	0	0	0
X_3	0.6	0.8	0	0.2	0	0
X_4	0	0	0.2	0	0.8	0.7
X_5	0.1	0	0	0.8	0	0.8
X_6	0	0	0	0.7	0.8	0

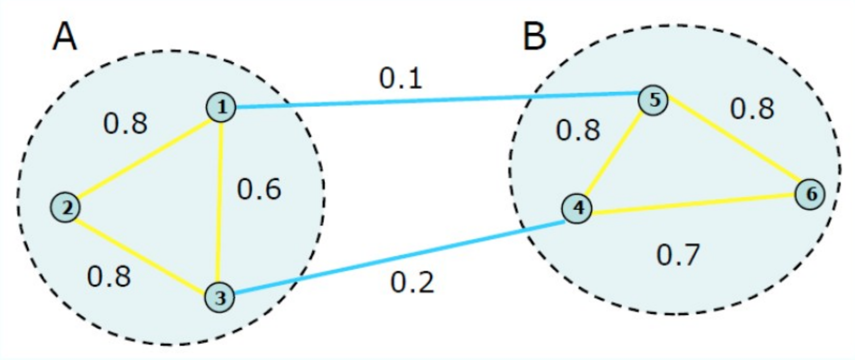
Adjacency Matrix ***W***

	X_1	X_2	X_3	X_4	X_5	X_6
X_1	1.5	0	0	0	0	0
X_2	0	1.6	0	0	0	0
X_3	0	0	1.6	0	0	0
X_4	0	0	0	1.7	0	0
X_5	0	0	0	0	1.7	0
X_6	0	0	0	0	0	1.5

Degree Matrix ***D***

Example(3)

- Step 2: Laplacian matrix
  - ***L*=*D*-*W***



	X_1	X_2	X_3	X_4	X_5	X_6
X_1	1.5	-0.8	-0.6	0	-0.1	0
X_2	-0.8	1.6	-0.8	0	0	0
X_3	-0.6	-0.8	1.6	-0.2	0	0
X_4	0	0	-0.2	1.7	-0.8	-0.7
X_5	-0.1	0	0	-0.8	1.7	-0.8
X_6	0	0	0	-0.7	-0.8	1.5

Laplacian Matrix ***L***

Example(4)

- Step 3: Eigen-decomposition
  - Eigenvalues

0
0.18
2.08
2.28
2.46
2.57

  - Eigenvectors

-0.4082	0.4084	...
-0.4082	0.4418	...
-0.4082	0.3713	...
-0.4082	-0.3713	...
-0.4082	-0.4050	...
-0.4082	-0.4452	...

	x_1	x_2	x_3	x_4	x_5	x_6
x_1	1.5	-0.8	-0.6	0	-0.1	0
x_2	-0.8	1.6	-0.8	0	0	0
x_3	-0.6	-0.8	1.6	-0.2	0	0
x_4	0	0	-0.2	1.7	-0.8	-0.7
x_5	-0.1	0	0	-0.8	1.7	-0.8
x_6	0	0	0	-0.7	-0.8	1.5

Laplacian Matrix ***L***

Example(5)

- Step 3: Eigen-decomposition
  - Eigenvalues

0
0.18
2.08
2.28
2.46
2.57

	X_1	X_2	X_3	X_4	X_5	X_6
X_1	1.5	-0.8	-0.6	0	-0.1	0
X_2	-0.8	1.6	-0.8	0	0	0
X_3	-0.6	-0.8	1.6	-0.2	0	0
X_4	0	0	-0.2	1.7	-0.8	-0.7
X_5	-0.1	0	0	-0.8	1.7	-0.8
X_6	0	0	0	-0.7	-0.8	1.5

  - Eigenvectors=

-0.4082	0.4084	...
-0.4082	0.4418	...
-0.4082	0.3713	...
-0.4082	-0.3713	...
-0.4082	-0.4050	...
-0.4082	-0.4452	...

U

Example(6)

- Step 4: Embedding
  - ***U***=

-0.4082	0.4084
-0.4082	0.4418
-0.4082	0.3713
-0.4082	-0.3713
-0.4082	-0.4050
-0.4082	-0.4452

Example(6)

- Step 4: Embedding
  - ***U***=

-0.4082	0.4084
-0.4082	0.4418
-0.4082	0.3713
-0.4082	-0.3713
-0.4082	-0.4050
-0.4082	-0.4452

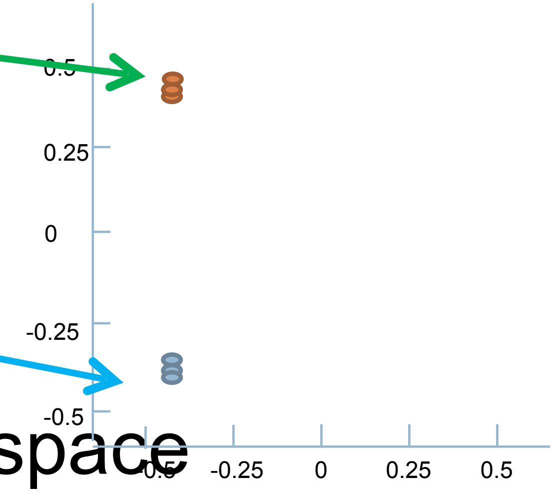
Each row represents a data point

Example(7)

- Step 4: Embedding
  - ***U***=

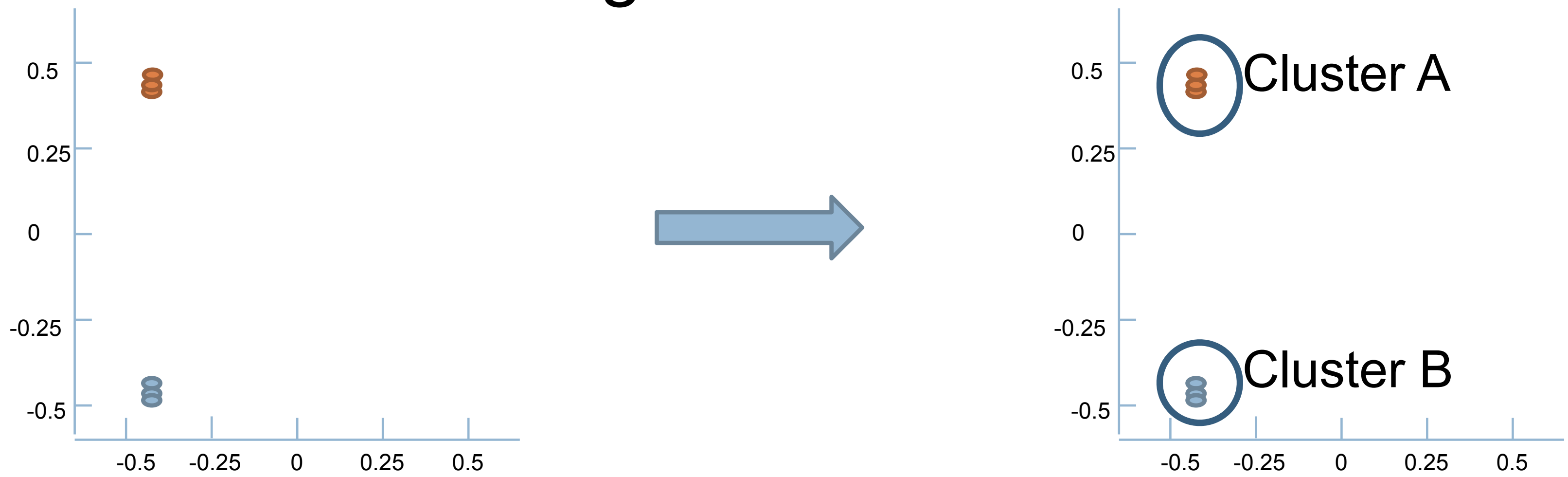
-0.4082	0.4084
-0.4082	0.4418
-0.4082	0.3713
-0.4082	-0.3713
-0.4082	-0.4050
-0.4082	-0.4452

  - Map it to a two-dimensional space



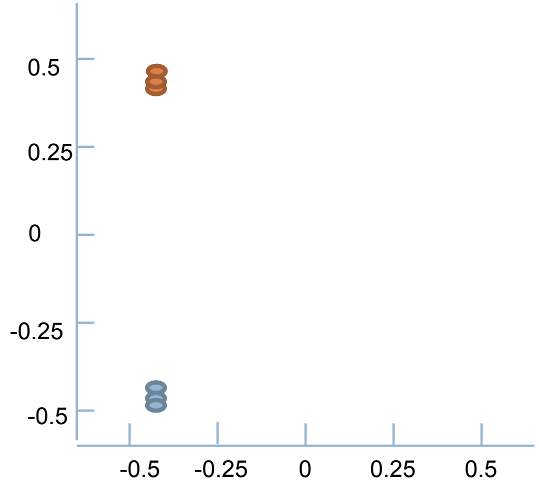
Example(8)

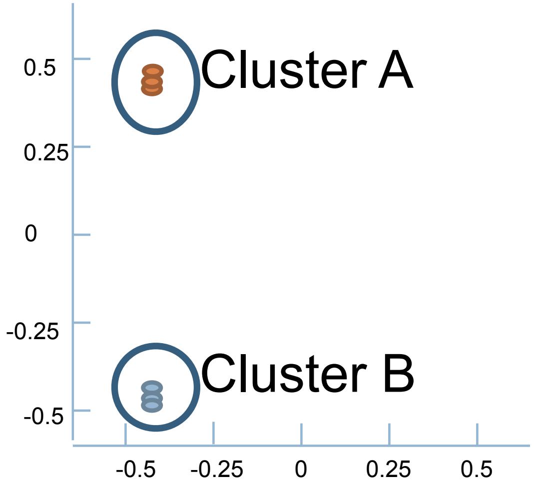
- Step 5: Clustering
  - K-means clustering

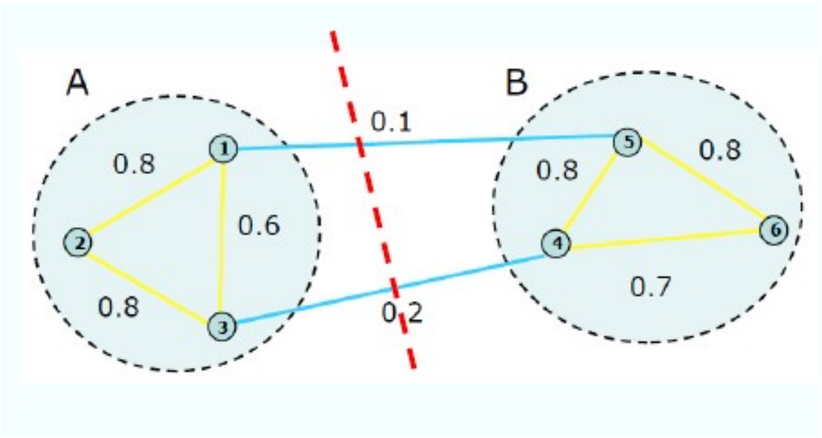


Example(8)

- Step 5: Clustering
  - K-means clustering







Number of Connected Components & Eigenvalues of *L*

a connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph

If an eigenvalue v has multiplicity k, then there are k linear independent eigenvectors corresponding to v

Number of Connected Components & Eigenvalues of L

a connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph

If an eigenvalue v has multiplicity k, then there are k linear independent eigenvectors corresponding to v

indicator vector: $\mathbb{1}_A = (f_1, \dots, f_n)' \in \mathbb{R}^n \quad f_i \in \{0,1\}$

Number of Connected Components & Eigenvalues of L

a connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph

If an eigenvalue v has multiplicity k, then there are k linear independent eigenvectors corresponding to v

indicator vector: $\mathbb{1}_A = (f_1, \ldots, f_n)' \in \mathbb{R}^n \quad f_i \in \{0,1\}$

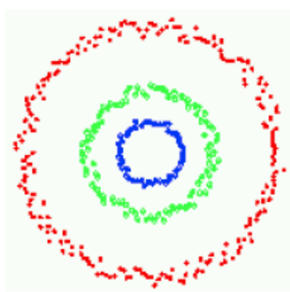
Proposition 2 (Number of connected components and the spectrum of L) Let G be an undirected graph with non-negative weights. The multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \ldots, \mathbb{1}_{A_k}$ of those components.

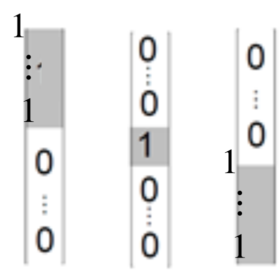
Number of Connected Components & Eigenvalues of L

a connected component of an undirected graph is a subgraph in which any two vertices are connected to each other by paths, and which is connected to no additional vertices in the supergraph

If an eigenvalue v has multiplicity k, then there are k linear independent eigenvectors corresponding to v

indicator vector: $\mathbb{1}_A = (f_1, \dots, f_n)' \in \mathbb{R}^n \quad f_i \in \{0,1\}$





eigenvectors corresponding to eigenvalue 0

Proposition 2 (Number of connected components and the spectrum of L) Let G be an undirected graph with non-negative weights. The multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \dots, A_k$ in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \dots, \mathbb{1}_{A_k}$ of those components.

Proof of Proposition 2

$D : Degree\ matrix$
$W : Adjacency\ matrix$

$$d_i = \sum_{j=1}^{n} w_{ij}$$

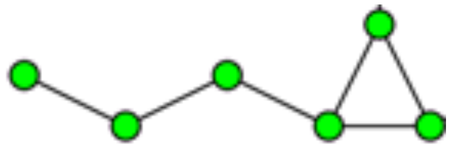
$$L = D - W$$

$$Lv = \lambda v.$$

Proof of Proposition 2

When $k = 1$: 1 connected component

Suppose $L \cdot f = 0 \cdot f$. Then we have



$$f^T L f = f^T \cdot 0 \cdot f = 0.$$

D : *Degree matrix*

W : *Adjacency matrix*

$$d_i = \sum_{j=1}^{n} w_{ij}$$

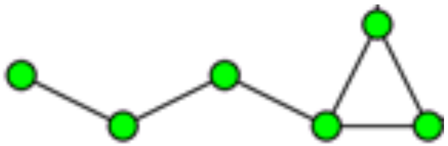
$$L = D - W$$

$$Lv = \lambda v.$$

Proof of Proposition 2

When $k = 1$: 1 connected component

Suppose $L \cdot f = 0 \cdot f$. Then we have



$$f^T L f = f^T \cdot 0 \cdot f = 0.$$

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

D : *Degree matrix*

W : *Adjacency matrix*

$$d_i = \sum_{j=1}^{n} w_{ij}$$

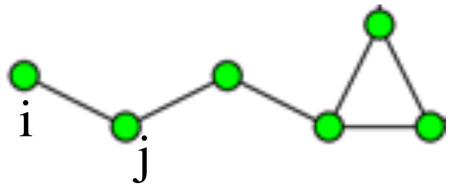
$$L = D - W$$

$$Lv = \lambda v.$$

Proof of Proposition 2

When $k = 1$: 1 connected component

Suppose $L \cdot f = 0 \cdot f$. Then we have



$$f^T L f = f^T \cdot 0 \cdot f = 0.$$

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

D : *Degree matrix*

W : *Adjacency matrix*

$$d_i = \sum_{j=1}^{n} w_{ij}$$

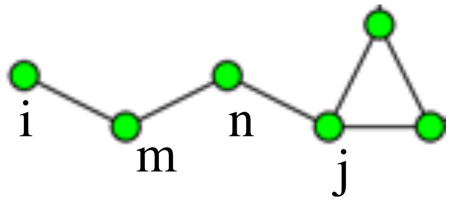
$$L = D - W$$

$$Lv = \lambda v.$$

Proof of Proposition 2

When $k = 1$: 1 connected component

Suppose $L \cdot f = 0 \cdot f$. Then we have



$$f^T L f = f^T \cdot 0 \cdot f = 0.$$

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

D : *Degree matrix*

W : *Adjacency matrix*

$$d_i = \sum_{j=1}^{n} w_{ij}$$

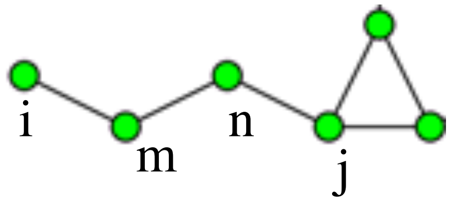
$$L = D - W$$

$$Lv = \lambda v.$$

Proof of Proposition 2

When $k = 1$: 1 connected component

Suppose $L \cdot f = 0 \cdot f$. Then we have



$$f^T L f = f^T \cdot 0 \cdot f = 0.$$

$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

$$f = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

D : *Degree matrix*

W : *Adjacency matrix*

$d_i = \sum_{j=1}^{n} w_{ij}$

$L = D - W$

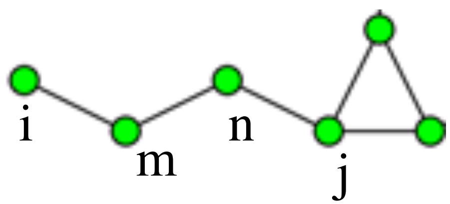
$Lv = \lambda v.$

Proof of Proposition 2

D : *Degree matrix*
W : *Adjacency matrix*
$d_i = \sum_{j=1}^{n} w_{ij}$
$L = D - W$
$Lv = \lambda v.$

When $k = 1$: 1 connected component

Suppose $L \cdot f = 0 \cdot f$. Then we have



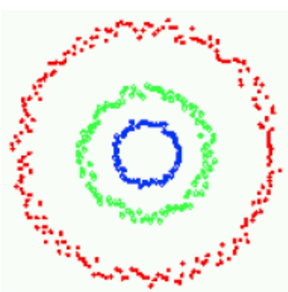
$$f^T L f = f^T \cdot 0 \cdot f = 0.$$

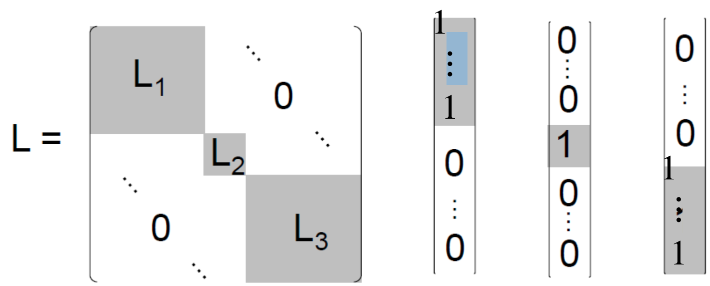
$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

$$f = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

When $k > 1$: several connected components

We assume that the vertices are ordered according to the connected components they belong to. In this case, the adjacency matrix W has a block diagonal form, and the same is true for the matrix L:



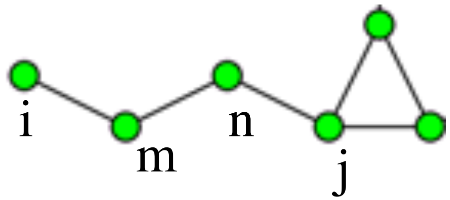


Proof of Proposition 2

D : *Degree matrix*
W : *Adjacency matrix*
$$d_i = \sum_{j=1}^{n} w_{ij}$$
$$L = D - W$$
$$Lv = \lambda v.$$

When $k = 1$: 1 connected component

Suppose $L \cdot f = 0 \cdot f$. Then we have



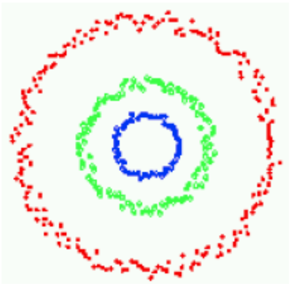
$$f^T L f = f^T \cdot 0 \cdot f = 0.$$

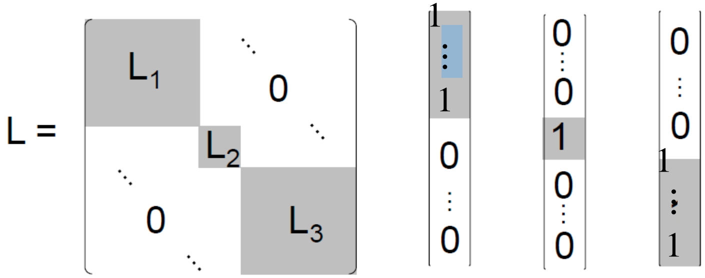
$$f^T L f = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} (f_i - f_j)^2$$

$$f = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

When $k > 1$: several connected components

We assume that the vertices are ordered according to the connected components they belong to. In this case, the adjacency matrix W has a block diagonal form, and the same is true for the matrix L:





Eigenvalues of L is the union of the eigenvalues of L_i, while the eigenvectors is given by v_i filled with 0s.

Spectral Clustering Algorithm

- Input: Graph $S \in \mathbb{R}^{n \times n}$, number k of clusters to form
  - Compute adjacency matrix W and degree matrix D
  - Laplacian $L = D - W$

Spectral Clustering Algorithm

- Input: Graph $S \in \mathbb{R}^{n \times n}$, number k of clusters to form
  - Compute adjacency matrix W and degree matrix D
  - Laplacian $L = D - W$
  - Compute the first *k* eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ of L
  - Let $\boldsymbol{U} \in \mathbb{R}^{N \times k}$ contain the vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ as columns

Spectral Clustering Algorithm

- Input: Graph $S \in \mathbb{R}^{n \times n}$, number k of clusters to form
  - Compute adjacency matrix W and degree matrix D
  - Laplacian L = D – W
  - Compute the first *k* eigenvectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k$ of L
  - Let $\boldsymbol{U} \in \mathbb{R}^{N \times k}$ contain the vectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k$ as columns

New space found!

Spectral Clustering Algorithm

- Input: Graph $S \in \mathbb{R}^{n \times n}$, number k of clusters to form
  - Compute adjacency matrix W and degree matrix D
  - Laplacian L = D – W
  - Compute the first *k* eigenvectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k$ of L
  - Let $\boldsymbol{U} \in \mathbb{R}^{N \times k}$ contain the vectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k$ as columns
  - Let $\boldsymbol{y}_i \in \mathbb{R}^{k}$ be the vector corresponding to the *i*-th row of U
  - Cluster the points $(\boldsymbol{y}_i)\ i = 1, \dots, N$ into *k* clusters using k-means

Spectral Clustering Algorithm

- Input: Graph $S \in \mathbb{R}^{n \times n}$, number k of clusters to form
  - Compute adjacency matrix W and degree matrix D
  - Laplacian $L = D - W$
  - Compute the first *k* eigenvectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k$ of L
  - Let $\boldsymbol{U} \in \mathbb{R}^{N \times k}$ contain the vectors $\boldsymbol{u}_1, \dots, \boldsymbol{u}_k$ as columns
  - Let $\boldsymbol{y}_i \in \mathbb{R}^k$ be the vector corresponding to the *i*-th row of U
  - Cluster the points $(\boldsymbol{y}_i)\ i = 1, \dots, N$ into *k* clusters using k-means

Representing data in the new space!

Spectral Clustering Algorithm

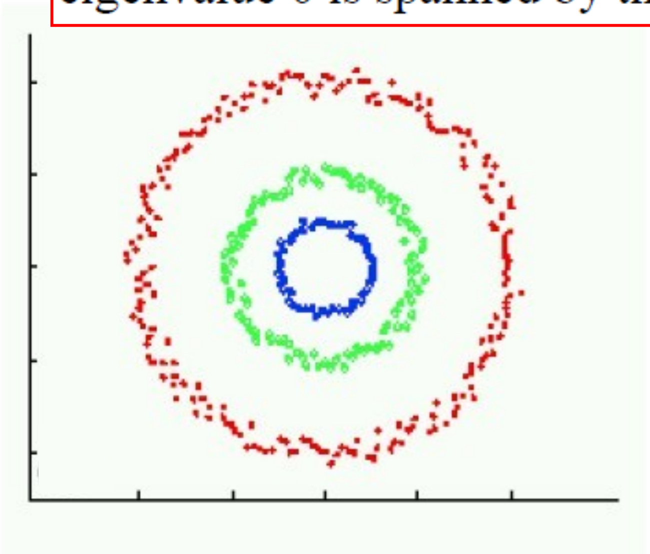
- Input: Graph $S \in \mathbb{R}^{n \times n}$, number k of clusters to form
  - Compute adjacency matrix W and degree matrix D
  - Laplacian $L = D - W$
  - Compute the first *k* eigenvectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ of L
  - Let $\boldsymbol{U} \in \mathbb{R}^{N \times k}$ contain the vectors $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_k$ as columns
  - Let $\boldsymbol{y}_i \in \mathbb{R}^{k}$ be the vector corresponding to the *i*-th row of U
  - Cluster the points $(\boldsymbol{y}_i)_{\ i = 1, \ldots, N}$ into *k* clusters using k-means

Time Complexity: O(n³)

Why Spectral Clustering Works?(1)

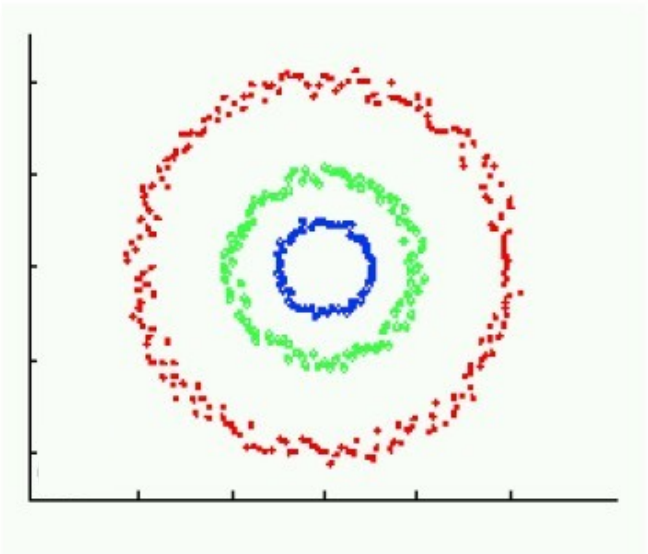
- Consider an ideal case
  - There are no similarities between any nodes in different connected components
  - This conforms to Proposition 2:

Proposition 2 (Number of connected components and the spectrum of L) Let G be an undirected graph with non-negative weights. The multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \dots, A_k$ in the graph. The eigenspace of eigenvalue 0 is spanned by the indicator vectors $\mathbb{1}_{A_1}, \dots, \mathbb{1}_{A_k}$ of those components.



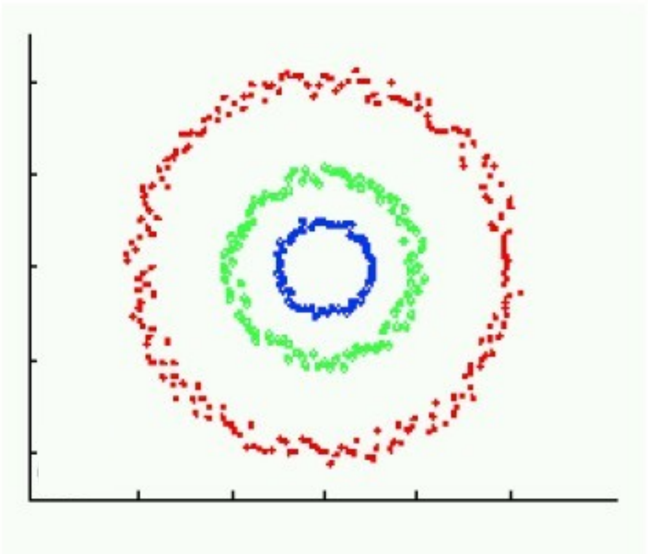
Why Spectral Clustering Works?(1)

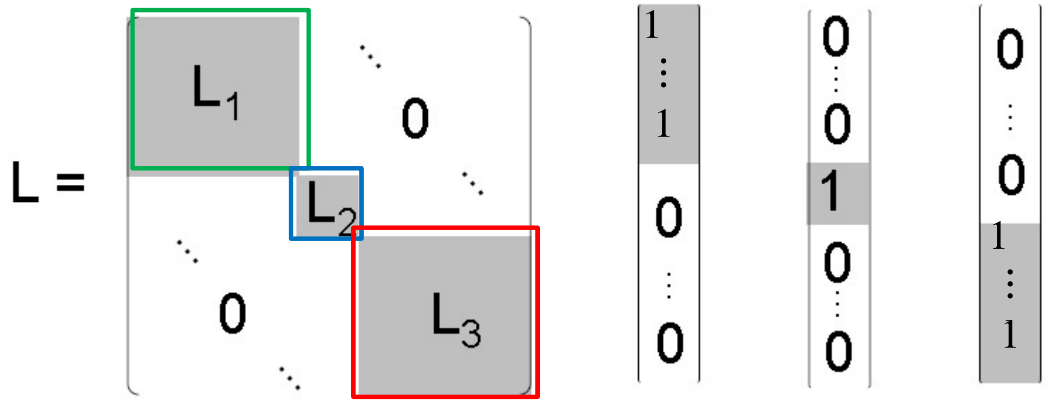
- Consider an ideal case
  - There are no similarities between any nodes in different connected components
  - Compute the weighted adjacency matrix ***W*** and degree matrix ***D***.
  - ***L*** = ***D*** - ***W***; compute ***L***'s 3 eigenvectors of eigenvalue 0.



Why Spectral Clustering Works?(1)

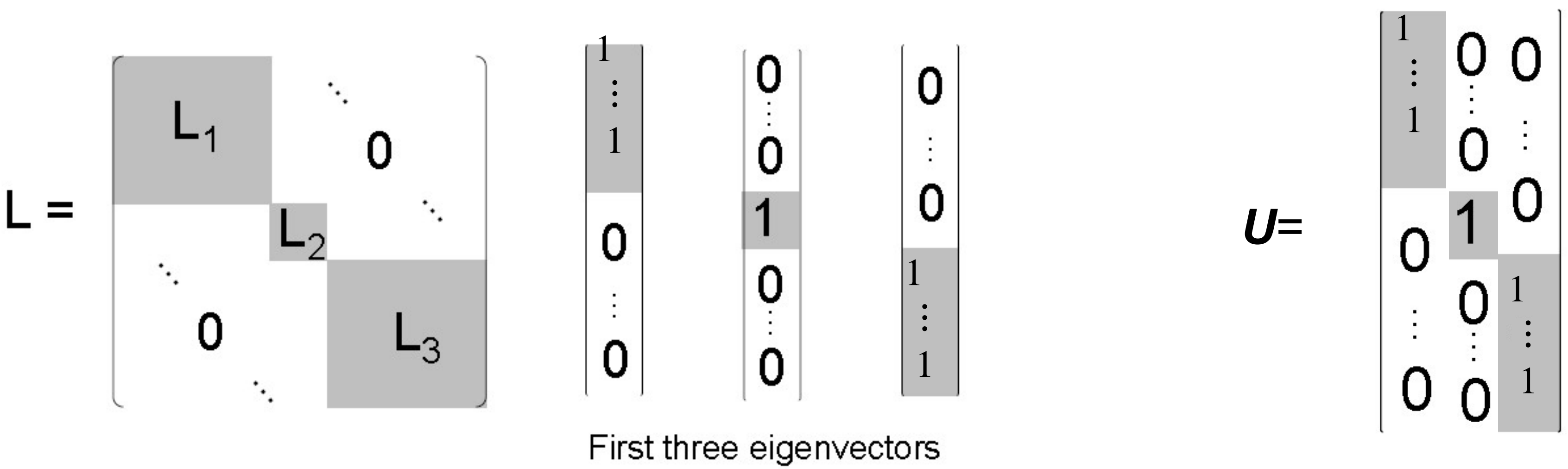
- Consider an ideal case
  - There are no similarities between any nodes in different connected components
  - Compute the weighted adjacency matrix ***W*** and degree matrix ***D***.
  - ***L*** = ***D*** - ***W***; compute ***L***'s 3 eigenvectors of eigenvalue 0.





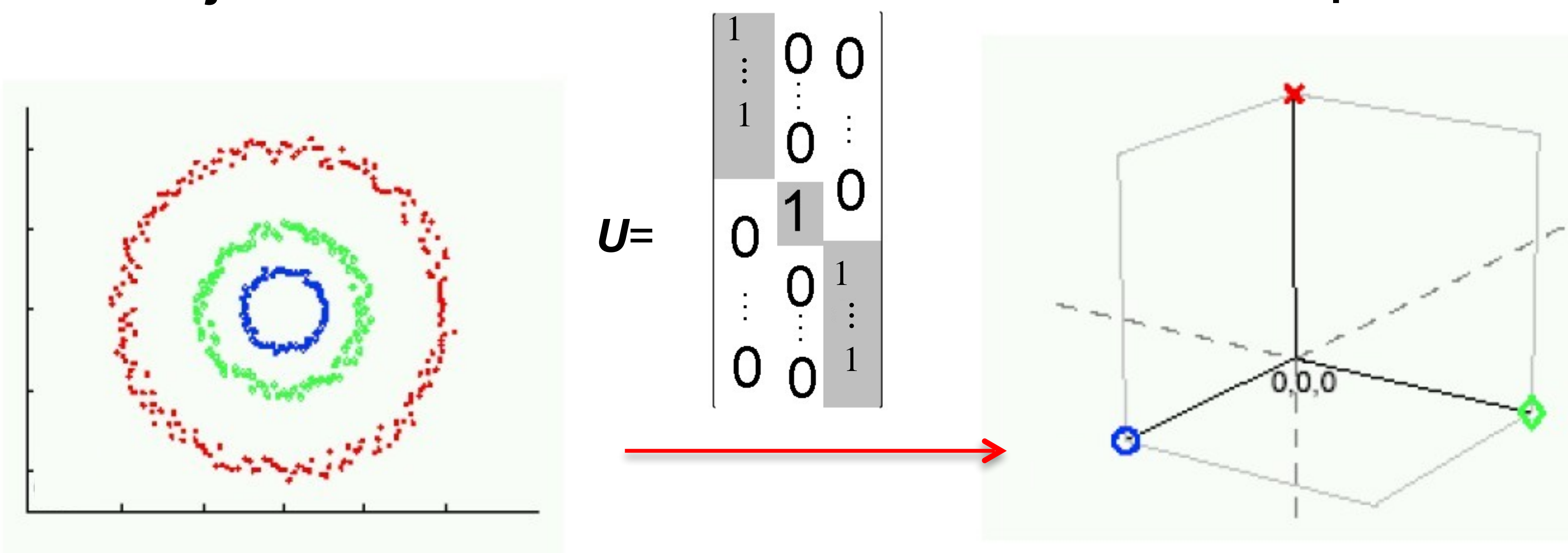
Why Spectral Clustering Works?(2)

- Consider an ideal case
  - Let the three eigenvectors be three columns of matrix ***U***.



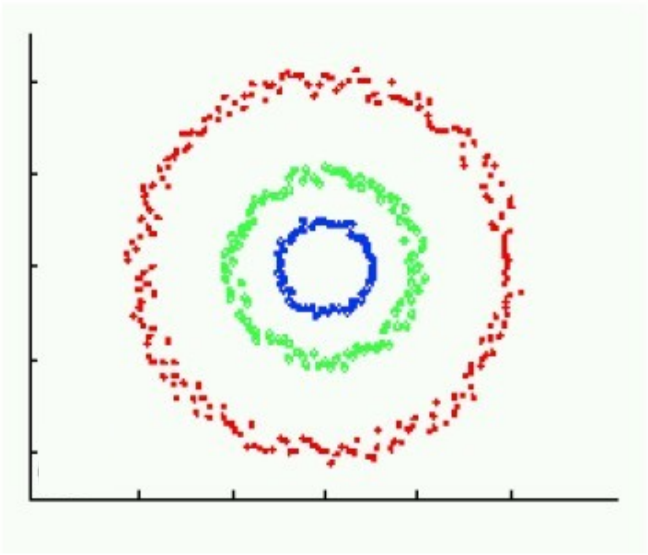
Why Spectral Clustering Works?(2)

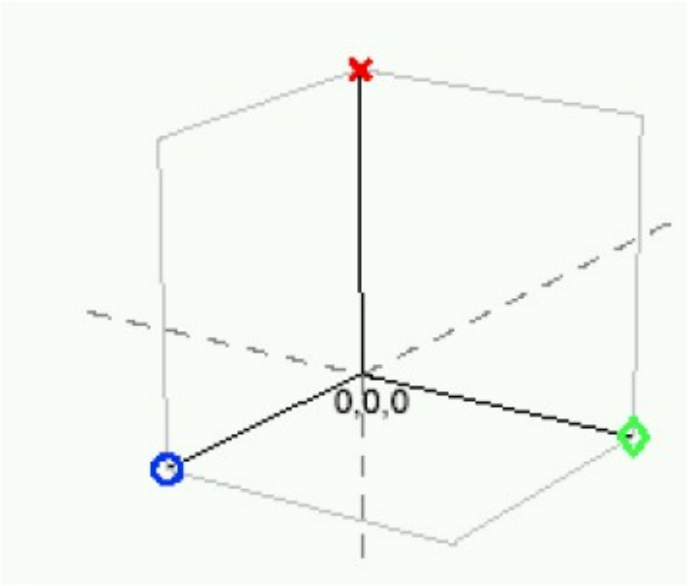
- Consider an ideal case
  - Let the three eigenvectors be three columns of a matrix ***U***.
  - Project the rows in ***U*** to a 3-dimensional space.



Why Spectral Clustering Works?(3)

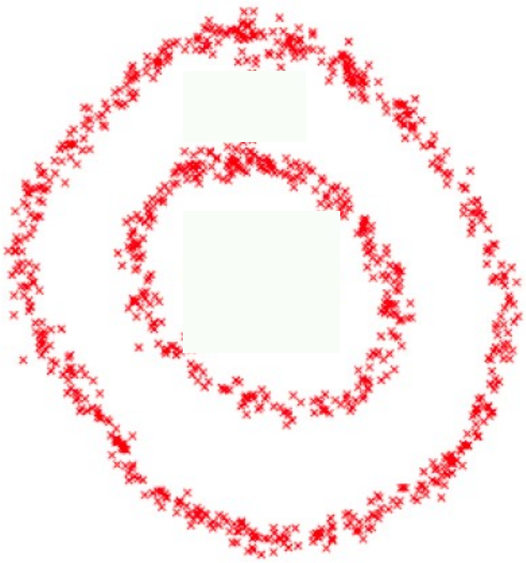
- Consider an ideal case
  - Now we use K-Means in this space, we can have very good results.
  - # of 0 eigenvalues = # of connected components





Why Spectral Clustering Works?(4)

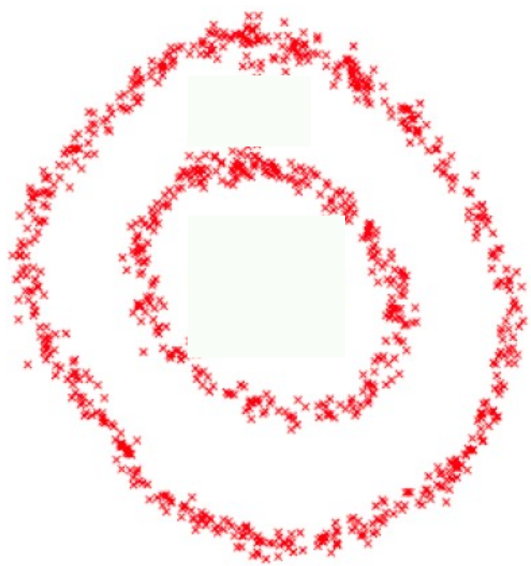
- What if not the ideal case?
  - We need to introduce Perturbation Theory.



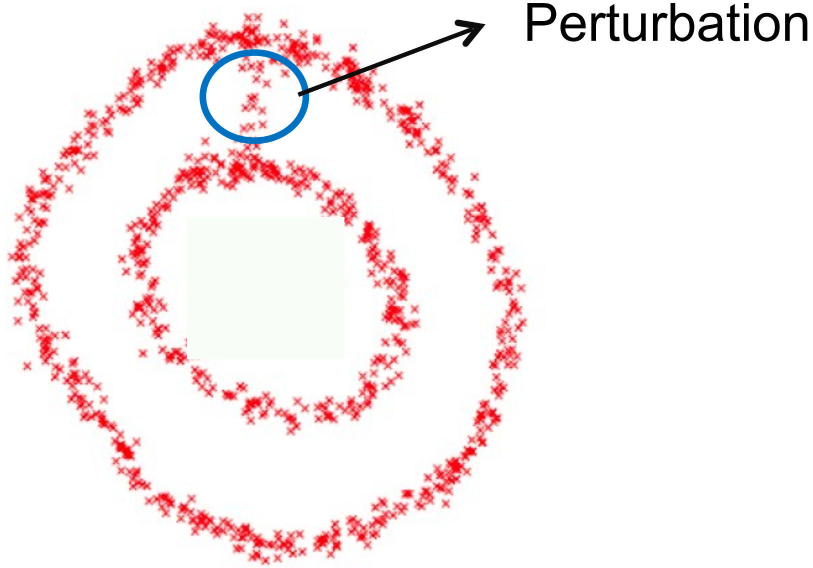
Ideal Case

Why Spectral Clustering Works?(4)

- What if not the ideal case?
  - We need to introduce Perturbation Theory.
    - Perturbation is like noise.



Ideal Case



Nearly ideal Case

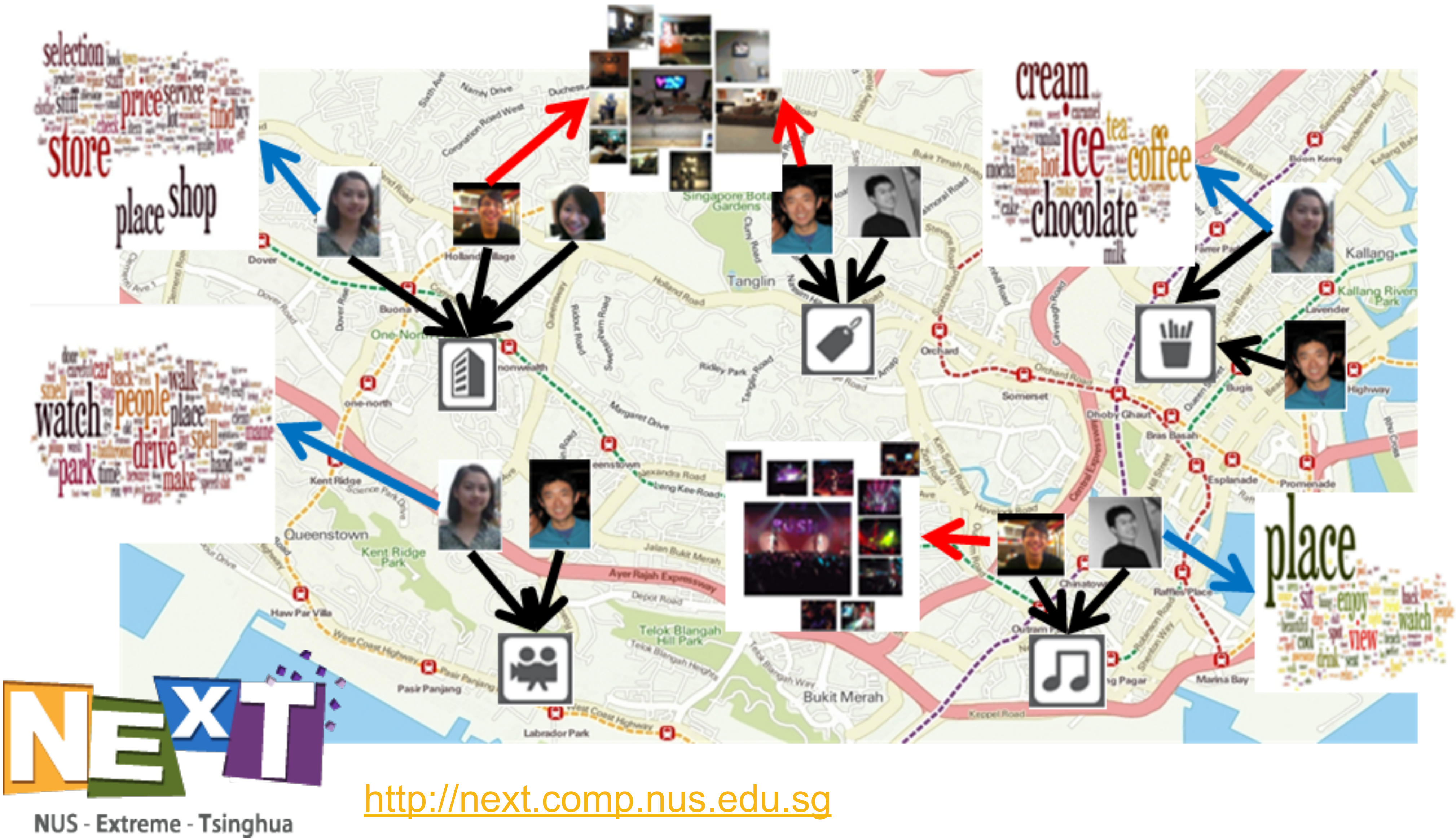
Why Spectral Clustering Works?(5)

- What if not the ideal case?
  - Perturbation Theory will not be formally discussed here.
  - References will be offered on IVLE.

Why Spectral Clustering Works?(5)

- What if not the ideal case?
  - Perturbation Theory will not be formally discussed here.
  - What you need to know is:
    - For ideal case, the between-cluster similarity is 0.
    - The first *k* eigenvectors of Laplacian matrix ***L*** are indicators of clusters.
    - For real case, ***L'*** = ***L*** + ***H***, where ***H*** is the perturbation.
    - Perturbation theory tells us the eigenvectors generated from ***L'*** will be very close to the ideal vectors from *L*, bounded by a small value.

Applications: Social Media



http://next.comp.nus.edu.sg