CPSC 8420 Advanced Machine Learning
Week 8: Spectral Clustering and Manifold Learning

Dr. Kai Liu

October 6, 2020

Motivation

All we algorithms have learnt by so far is based on (Euclidean) distance, such as Least Squares, PCA, LDA, *etc*.

- In some cases, distance based method can't achieve the result as we expect.
- Graph based algorithm may provide a different perspective and offer better results.

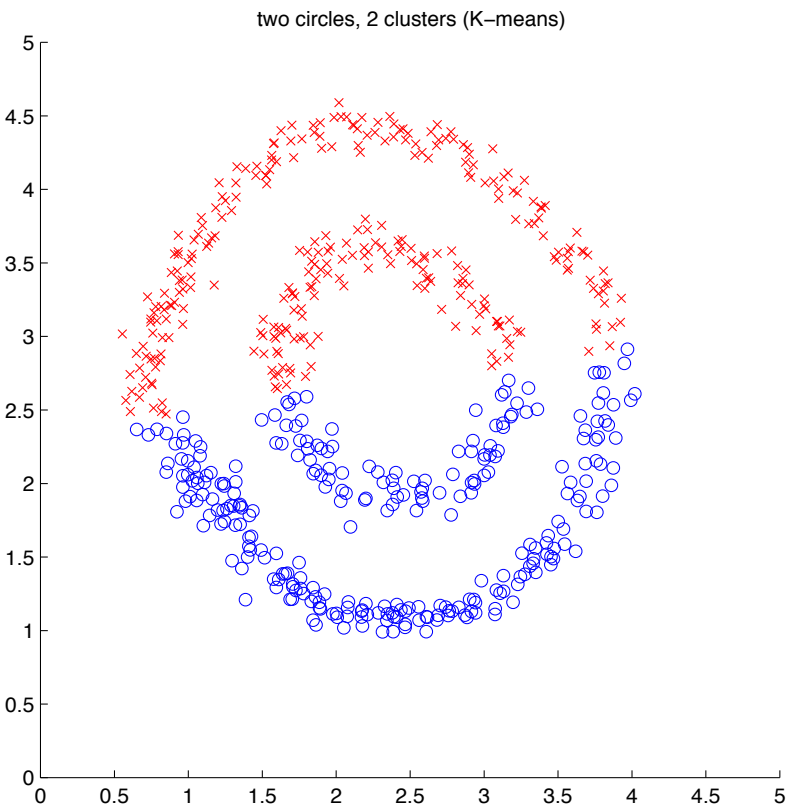
Learning Outcomes

Our goal for today's lecture is to understand:

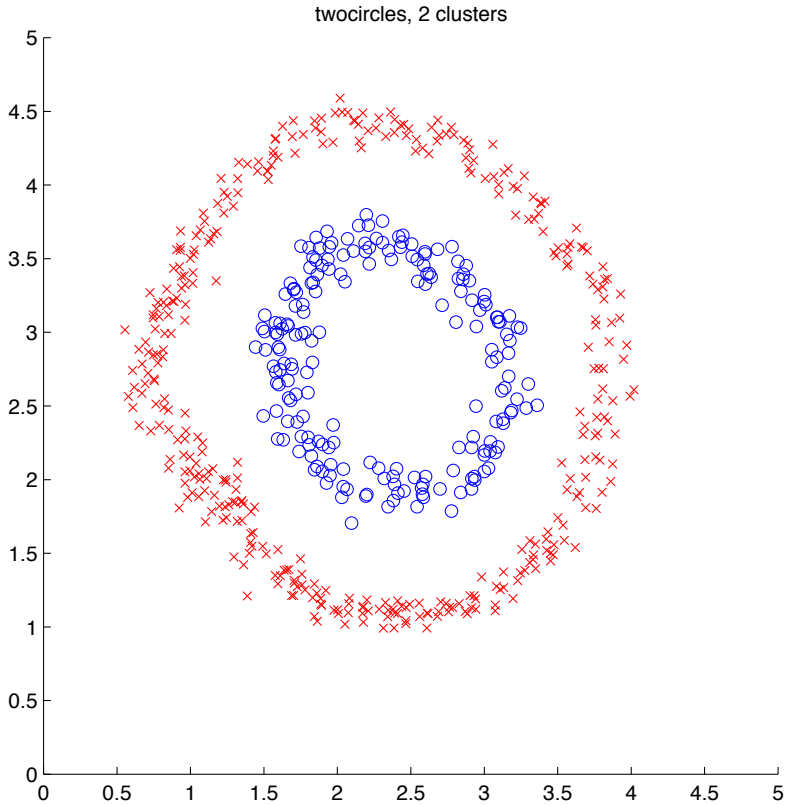
- Basic concept in graph such as Adjacency Matrix, Degree Matrix, Laplacian Matrix, *etc*.
- How to utilize RatioCut idea for clustering
- How to obtain the optimal solution

A Gentle Start

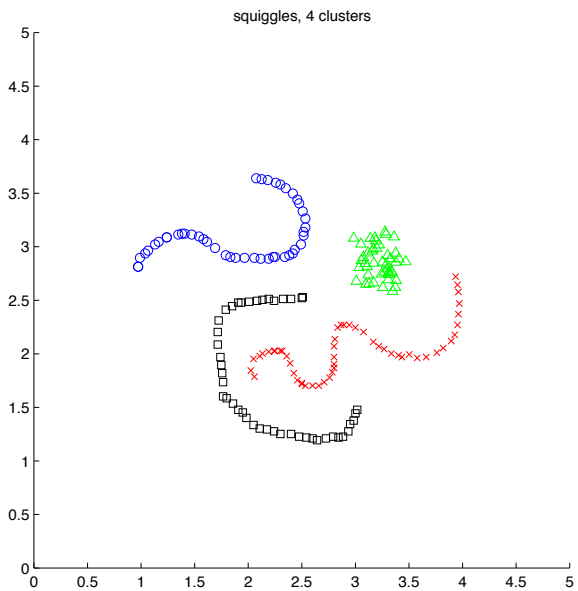
K-means

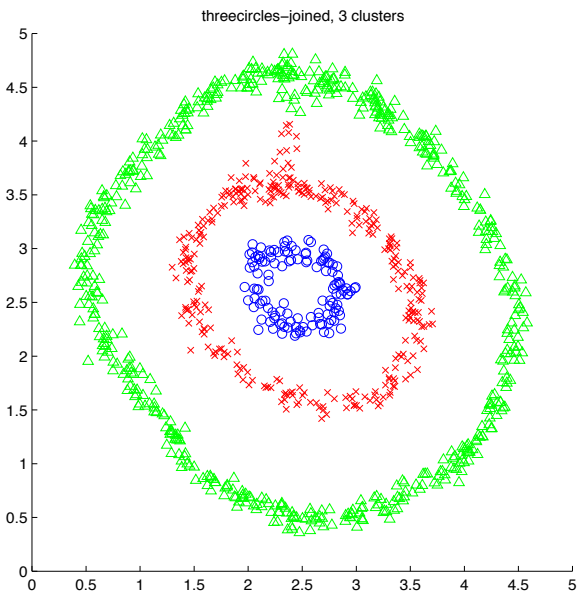


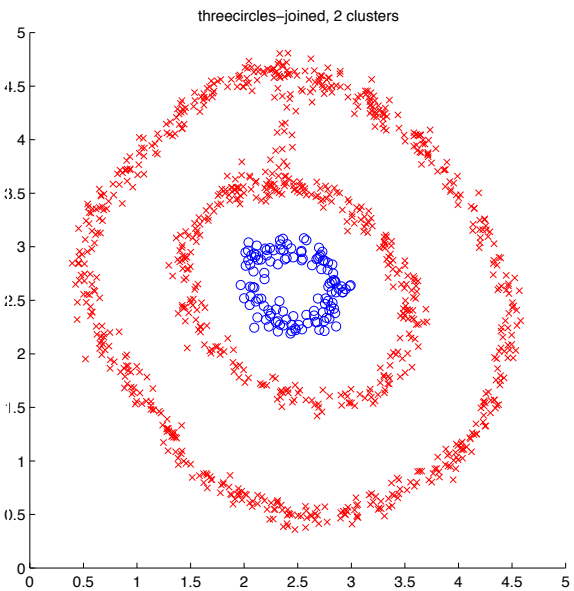
Spectral clustering



A Gentle Start

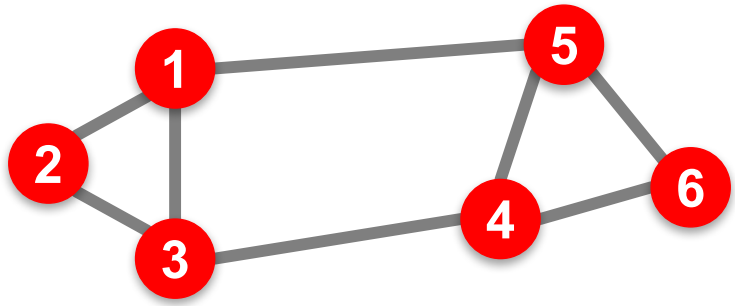




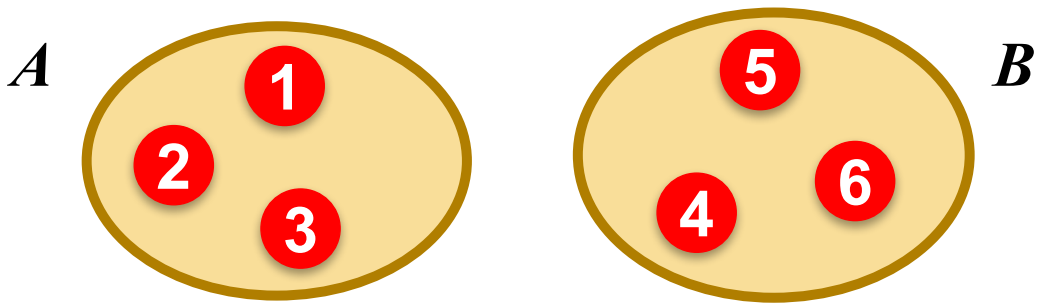


Basic Concept

- **Undirected graph $G(V, E)$:**



- **Bi-partitioning task:**
  - Divide vertices into two disjoint groups A, B

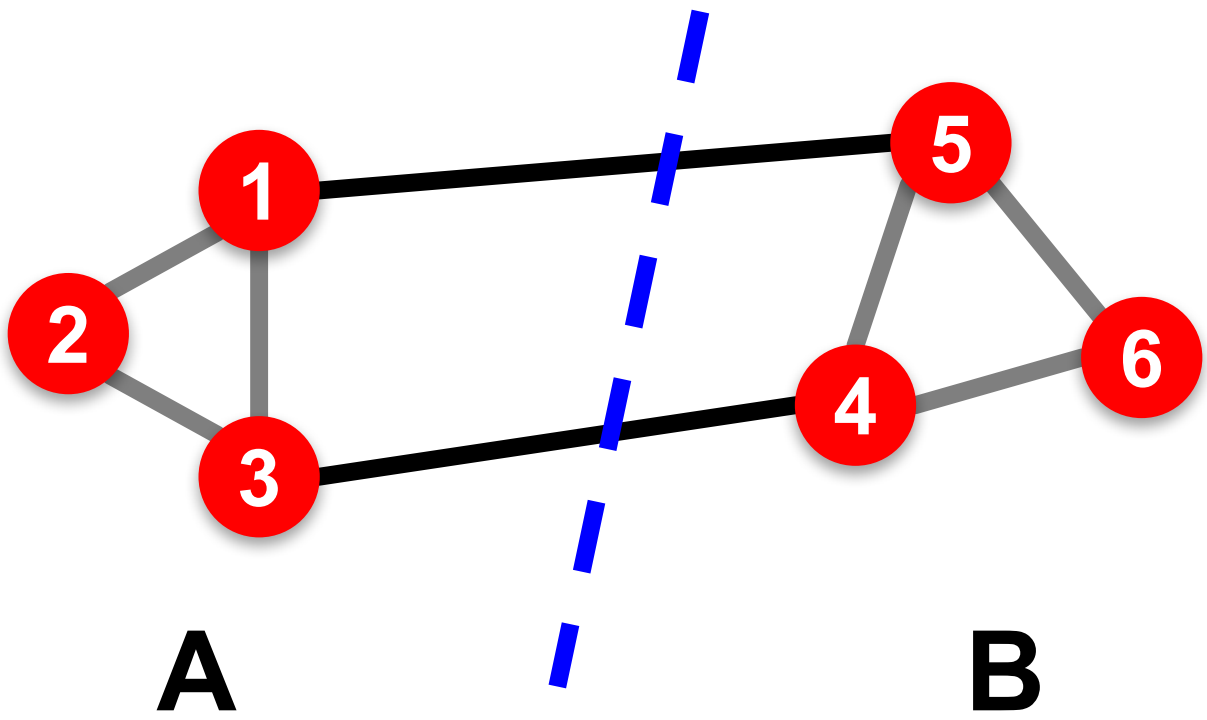


- **Questions:**
  - How can we define a "good" partition of G?
  - How can we efficiently identify such a partition?

Graph Partitioning

What makes a good partition?

1. Maximize the number of within-group connections
2. Minimize the number of between-group connections



Definition

Express partitioning objectives as a function of the 'edge cut' of the partition. For two not necessarily disjoint sets $A, B \subset V$, we define $W(A, B) = \sum_{i \in A, j \in B} w_{ij}$ and Cut as set of edges with only one vertex in a group $cut(A, B) = \sum_{i \in A, j \in B} w_{ij}$, and when there are k subsets, then

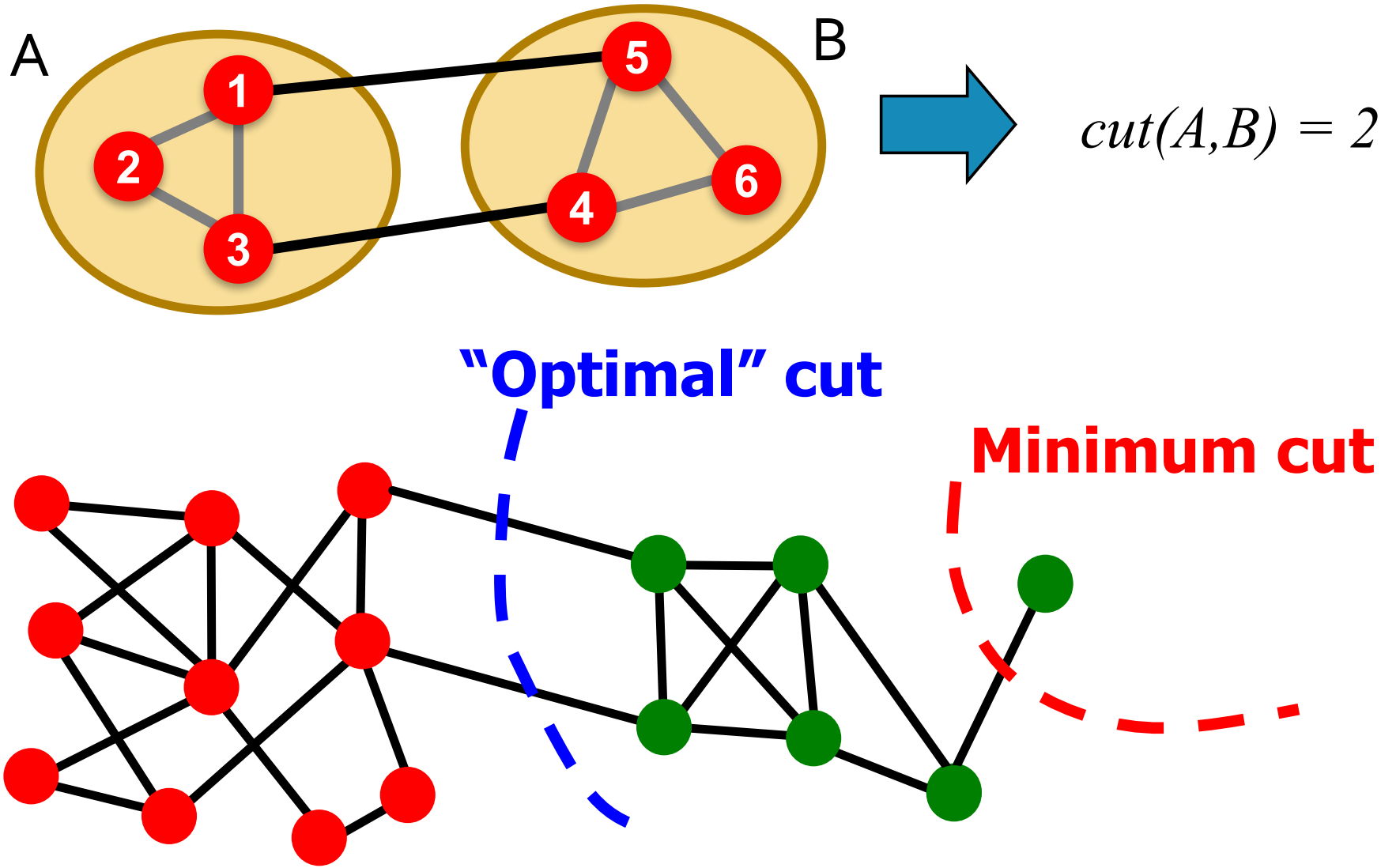
$cut(A_1, A_2, ...A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A}_i)$.

The degree of a vertex $v_i \in V$ is defined as: $d_i = \sum_{j=1}^{n} w_{ij}$. We consider two different ways of measuring the 'size' of a subset $A \subset V$:

$$|A| := \textit{the number of vertices in } A$$

$$vol(A) := \sum_{i \in A} d_i$$

Example



Adjacency Matrix

	1	2	3	4	5	6
1	0	1	1	0	1	0
2	1	0	1	0	0	0
3	1	1	0	1	0	0
4	0	0	1	0	1	1
5	1	0	0	1	0	1
6	0	0	0	1	1	0

Adjacency Matrix

There are several popular constructions to transform a given set $x_1, \ldots, x_n$ of data points with pairwise similarities s_{ij} into a graph.

- The ϵ-neighborhood graph: Here we connect all points whose pairwise distances are smaller than ϵ: $w_{ij} = \begin{cases} 0 & s_{ij} > \epsilon \\ \epsilon & s_{ij} \leq \epsilon \end{cases}$.

Adjacency Matrix

There are several popular constructions to transform a given set $x_1, \ldots, x_n$ of data points with pairwise similarities s_{ij} into a graph.

- The ϵ-neighborhood graph: Here we connect all points whose pairwise distances are smaller than ϵ: $w_{ij} = \begin{cases} 0 & s_{ij} > \epsilon \\ \epsilon & s_{ij} \leq \epsilon \end{cases}$.
- k-nearest neighbor graphs: Here the goal is to connect vertex v_i with vertex v_j if v_j is among the k-nearest neighbors of v_i: $w_{ij} = w_{ji} = \begin{cases} 0 & x_i \notin KNN(x_j) \text{ or } x_j \notin KNN(x_i) \\ exp(-\frac{||x_i - x_j||_2^2}{2\sigma^2}) & x_i \in KNN(x_j) \text{ and } x_j \in KNN(x_i) \end{cases}$.

Adjacency Matrix

There are several popular constructions to transform a given set $x_1, \ldots, x_n$ of data points with pairwise similarities s_{ij} into a graph.

- The ϵ-neighborhood graph: Here we connect all points whose pairwise distances are smaller than ϵ: $w_{ij} = \begin{cases} 0 & s_{ij} > \epsilon \\ \epsilon & s_{ij} \leq \epsilon \end{cases}$.
- k-nearest neighbor graphs: Here the goal is to connect vertex v_i with vertex v_j if v_j is among the k-nearest neighbors of v_i: $w_{ij} = w_{ji} = \begin{cases} 0 & x_i \notin KNN(x_j) \text{ or } x_j \notin KNN(x_i) \\ exp(-\frac{||x_i - x_j||_2^2}{2\sigma^2}) & x_i \in KNN(x_j) \text{ and } x_j \in KNN(x_i) \end{cases}$.
- The fully connected graph: Here we simply connect all points with positive similarity with each other, and we weight all edges by s_{ij}. An example for such a similarity function is the Gaussian similarity function: $w_{ij} = s_{ij} = exp(-\frac{||x_i - x_j||_2^2}{2\sigma^2})$, where the parameter σ controls the width of the neighborhoods.

Degree Matrix

	1	2	3	4	5	6
1	3	0	0	0	0	0
2	0	2	0	0	0	0
3	0	0	3	0	0	0
4	0	0	0	3	0	0
5	0	0	0	0	3	0
6	0	0	0	0	0	2

Laplacian Matrix

	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

$$L = D - A$$

Property of Laplacian Matrix

L is always symmetric and SPD. For arbitary vector f, we have:

$$f^T Lf = f^T Df - f^T Wf = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j$$

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

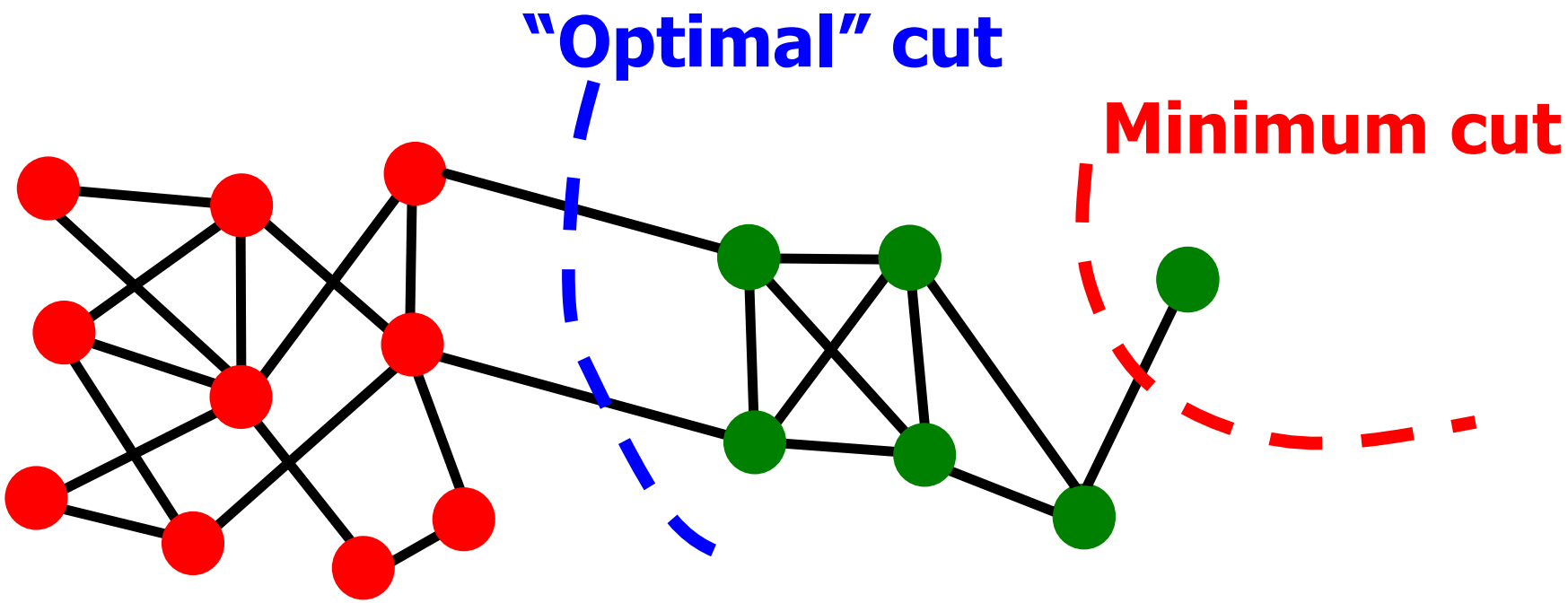
Then the multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \ldots, A_k$ in the graph.

$$\mathrm{L} = \begin{pmatrix} L_1 & \cdots & \cdots & \cdots \\ \cdots & L_2 & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \cdots & \cdots & L_n \end{pmatrix}$$

RatioCut

$$RatioCut(A_1, A_2, ...A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A}_i)}{|A_i|}$$

$$Cut(A_1, A_2, ...A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A}_i)$$



RatioCut

If we introduce indicator vector: $h_j \in \{h_1, h_2, ..h_k\}, j \in [1, k]$, for any vector $h_j \in R^n$, we define: $h_{ij} = \begin{cases} 0 & v_i \notin A_j \\ \frac{1}{\sqrt{|A_j|}} & v_i \in A_j \end{cases}$, then:

$$
\begin{aligned}
h_i^T L h_i &= \frac{1}{2} \sum_{m=1} \sum_{n=1} w_{mn} (h_{im} - h_{in})^2 \\
&= \frac{1}{2} (\sum_{m \in A_i, n \notin A_i} w_{mn} (\frac{1}{\sqrt{|A_i|}} - 0)^2 + \sum_{m \notin A_i, n \in A_i} w_{mn} (0 - \frac{1}{\sqrt{|A_i|}})^2 \\
&= \frac{1}{2} (\sum_{m \in A_i, n \notin A_i} w_{mn} \frac{1}{|A_i|} + \sum_{m \notin A_i, n \in A_i} w_{mn} \frac{1}{|A_i|} \\
&= \frac{1}{2} (cut(A_i, \overline{A}_i) \frac{1}{|A_i|} + cut(\overline{A}_i, A_i) \frac{1}{|A_i|}) \\
&= \frac{cut(A_i, \overline{A}_i)}{|A_i|}
\end{aligned}
$$

RatioCut

For a subset, its RatioCut is $h_i^T L h_i$, then for k subsets we have:

$$RatioCut(A_1, A_2, ...A_k) = \sum_{i=1}^{k} h_i^T L h_i = \sum_{i=1}^{k} (H^T LH)_{ii} = tr(H^T LH) \tag{2}$$

Unfortunately, this is an integer programming problem which we cannot solve efficiently. Instead, we relax the latter requirement and simply search an orthonormal matrix $H \in \mathbb{R}^{n \times k}$. By observing $H^T H = I$, we have the objective as:

$$\underbrace{arg\ min}_{H}\ tr(H^T LH)\ \ s.t.\ H^T H = I \tag{3}$$

RatioCut

Unnormalized spectral clustering

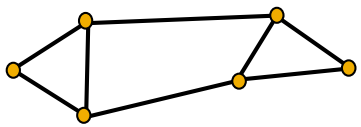
Input: Similarity matrix $S \in \mathbb{R}^{n \times n}$, number k of clusters to construct.

- Construct a similarity graph by one of the ways described in Section 2. Let W be its weighted adjacency matrix.
- Compute the unnormalized Laplacian L.
- **Compute the first k eigenvectors $u_1, \ldots, u_k$ of L.**
- Let $U \in \mathbb{R}^{n \times k}$ be the matrix containing the vectors $u_1, \ldots, u_k$ as columns.
- For $i = 1, \ldots, n$, let $y_i \in \mathbb{R}^k$ be the vector corresponding to the i-th row of U.
- Cluster the points $(y_i)_{i=1,\ldots,n}$ in $\mathbb{R}^k$ with the k-means algorithm into clusters $C_1, \ldots, C_k$.

Output: Clusters $A_1, \ldots, A_k$ with $A_i = \{j | \, y_j \in C_i\}$.

RatioCut

- **1) Pre-processing:**
  - Build Laplacian matrix L of the graph



	1	2	3	4	5	6
1	3	-1	-1	0	-1	0
2	-1	2	-1	0	0	0
3	-1	-1	3	-1	0	0
4	0	0	-1	3	-1	-1
5	-1	0	0	-1	3	-1
6	0	0	0	-1	-1	2

- **2) Decomposition:**
  - Find eigenvalues λ and eigenvectors x of the matrix L

$\lambda =$

0.0
1.0
3.0
3.0
4.0
5.0

$X =$

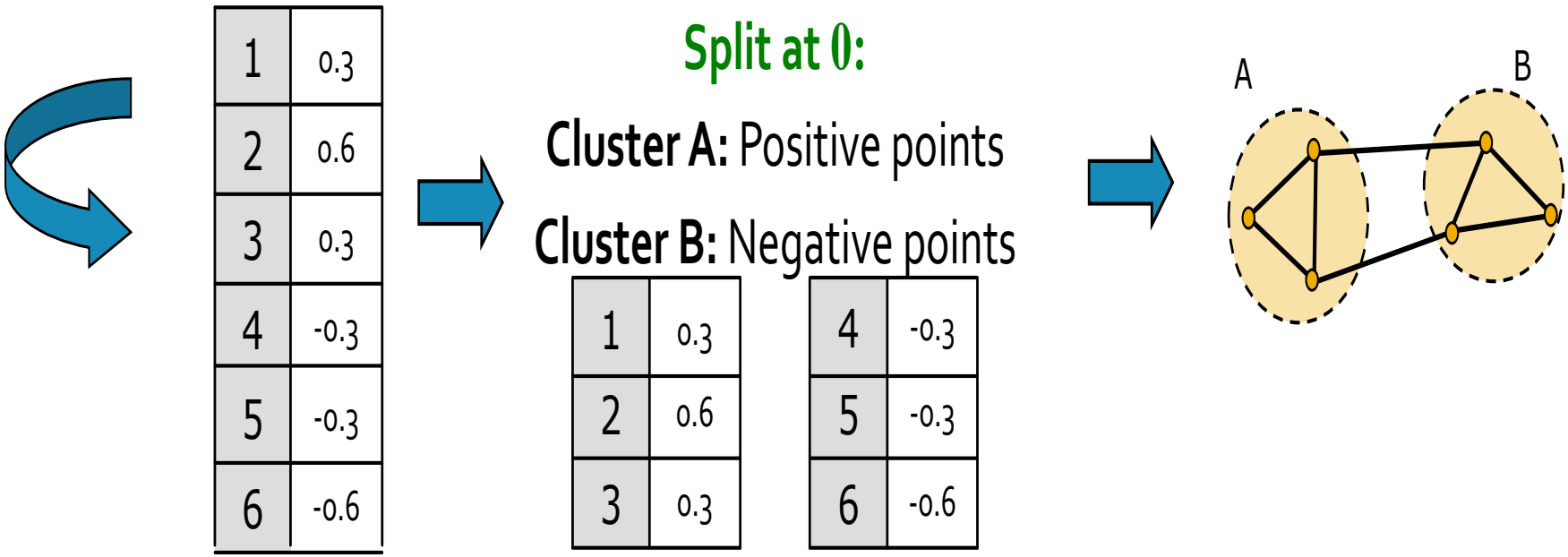
0.4	0.3	-0.5	-0.2	-0.4	-0.5
0.4	0.6	0.4	-0.4	0.4	0.0
0.4	0.3	0.1	0.6	-0.4	0.5
0.4	-0.3	0.1	0.6	0.4	-0.5
0.4	-0.3	-0.5	-0.2	0.4	0.5
0.4	-0.6	0.4	-0.4	-0.4	0.0

  - Map vertices to corresponding components of λ_2

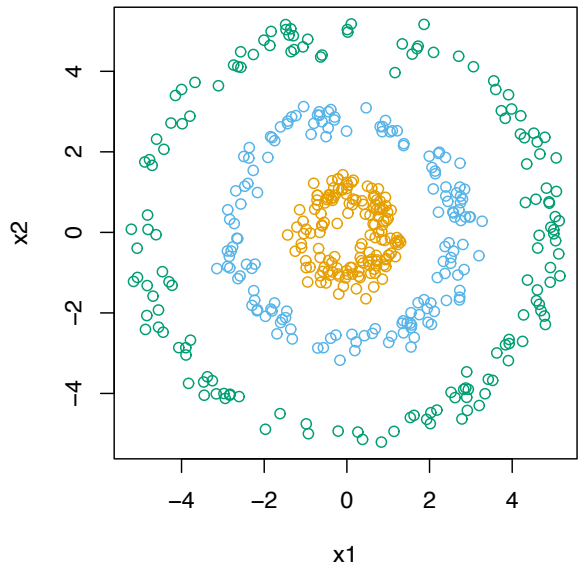
1	0.3
2	0.6
3	0.3
4	-0.3
5	-0.3
6	-0.6

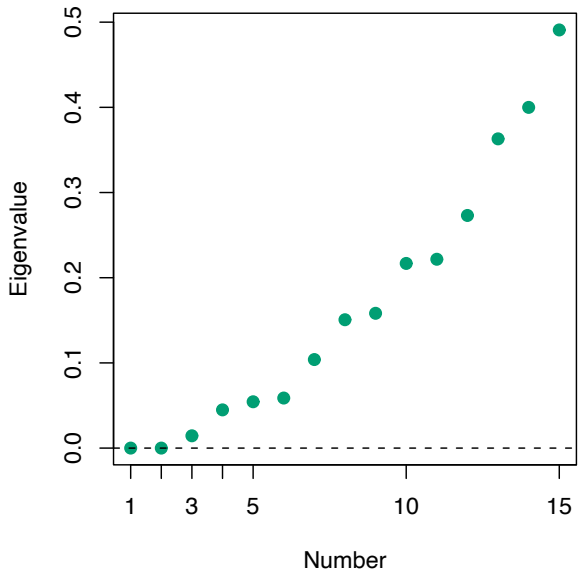
How do we now find the clusters?

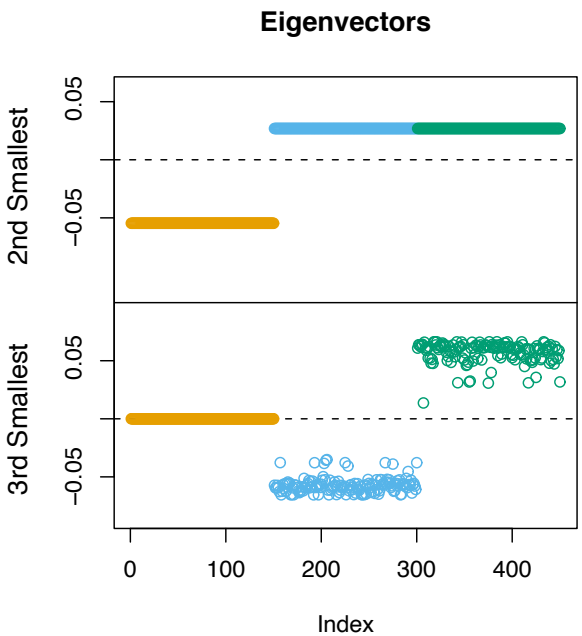
RatioCut


1 0.3
2 0.6
3 0.3
4 -0.3
5 -0.3
6 -0.6
Split at 0:
Cluster A: Positive points
Cluster B: Negative points
1 0.3
2 0.6
3 0.3
4 -0.3
5 -0.3
6 -0.6
A
B

RatioCut


x2
x1
−4
−2
0
2
4


Eigenvalue
Number
0.0
0.1
0.2
0.3
0.4
0.5
1
3
5
10
15


Eigenvectors
2nd Smallest
3rd Smallest
Index
0.05
−0.05
0
100
200
300
400

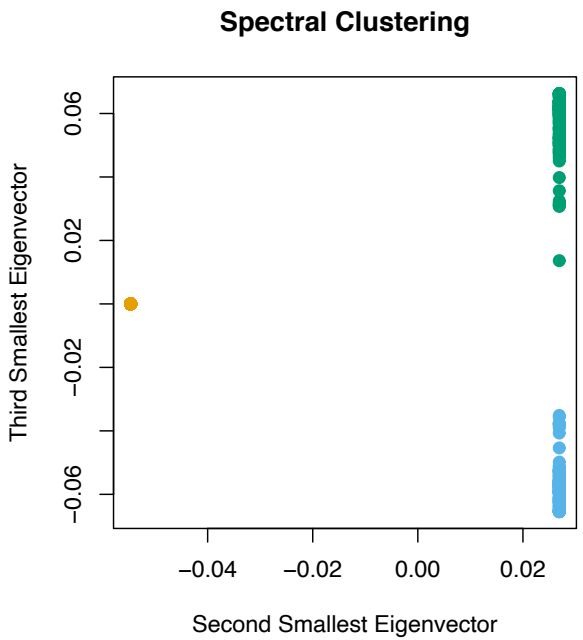
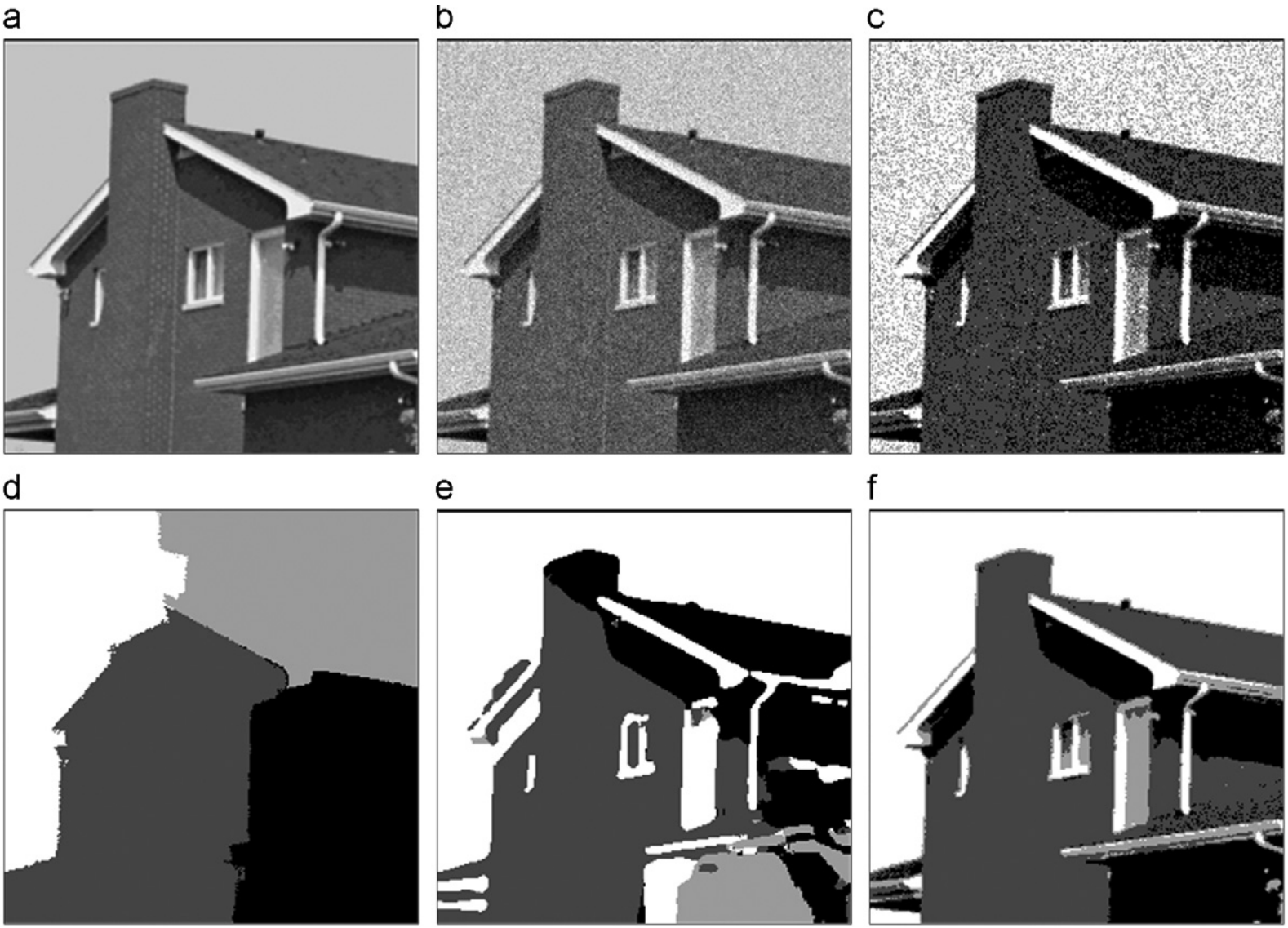

Spectral Clustering
Third Smallest Eigenvector
Second Smallest Eigenvector
0.06
0.02
−0.02
−0.06
−0.04
0.00

Image Segmentation

Image Segmentation



Clustering Results VS. K-means

k	TDT2		Reuters-21578	
	K-means	SC	K-means	SC
2	0.989	0.998	0.871	0.923
3	0.974	0.996	0.775	0.816
4	0.959	0.996	0.732	0.793
...				
9	0.852	0.984	0.553	0.625
10	0.835	0.979	0.545	0.615