Spectral Clustering
Introduction

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Illustrate the key idea of spectral clustering

Objective

Define basic graph notations useful for spectral clustering

Revisiting k-means & Mixture Models

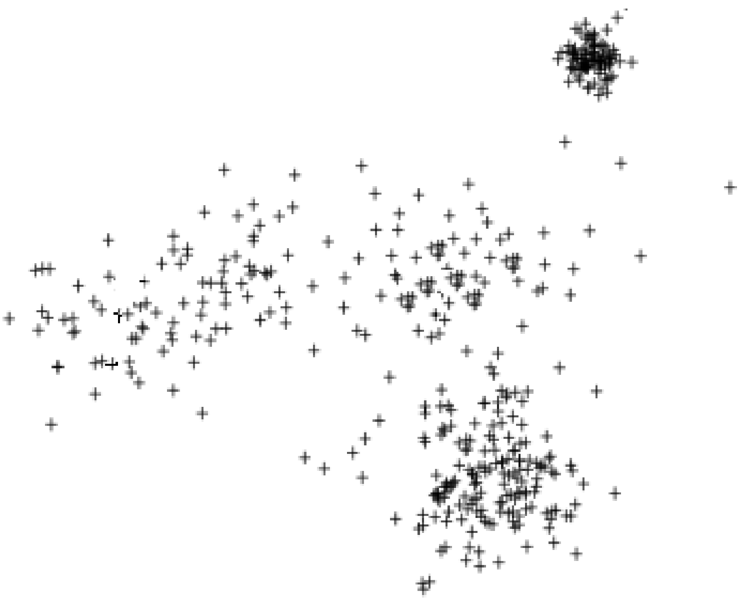
- **K-means use “hard” membership while mixture models allow “soft” membership**

- **Both use feature/vector representation of the data as input ➔ E.g., Euclidean distance is one natural (dis)similarity measure.**
  - What if the input data is NOT represented in feature/vector, format?
    - E.g., graph data.
    - E.g., objects with only pair-wise similarities (like individuals on a social network ➔ community detection)

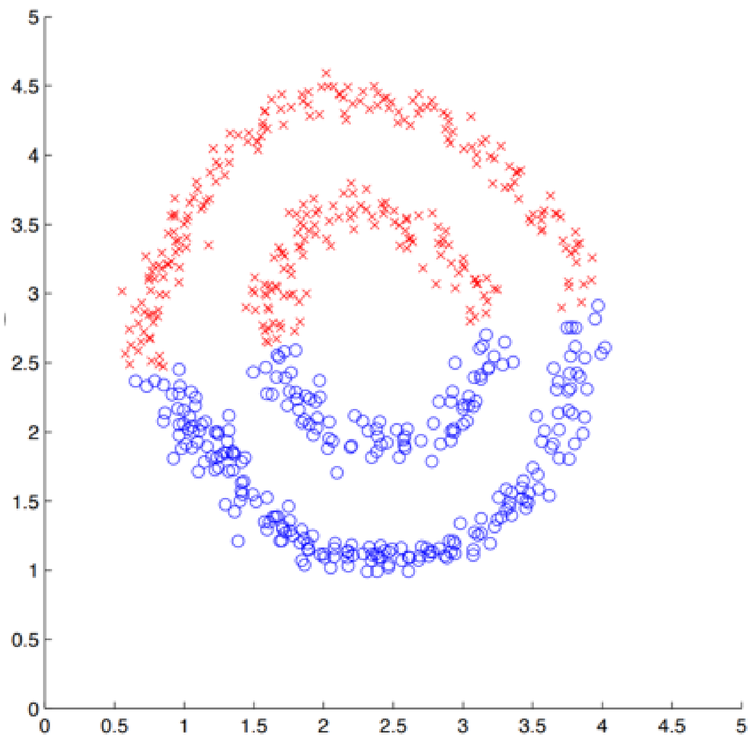
Revisiting k-means & Mixture Models

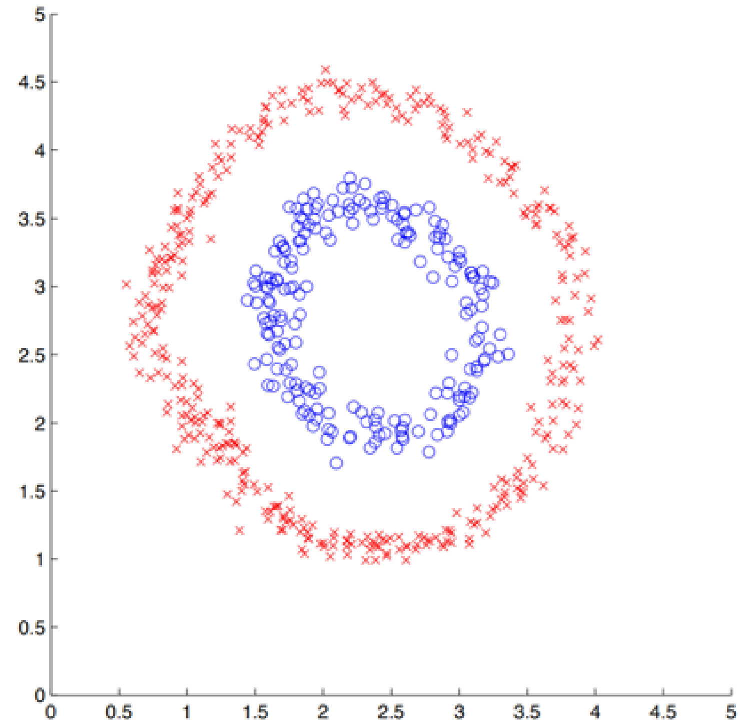
- In both k-means and mixture models, we look for compact clustering structures.

- In some cases, connected-component structures may be more desirable.



Example





Source: Ng, A.Y., Michael I.J., and Yair, W. "On spectral clustering: Analysis and an algorithm." *Advances in neural information processing systems*. 2002.

Spectral Clustering

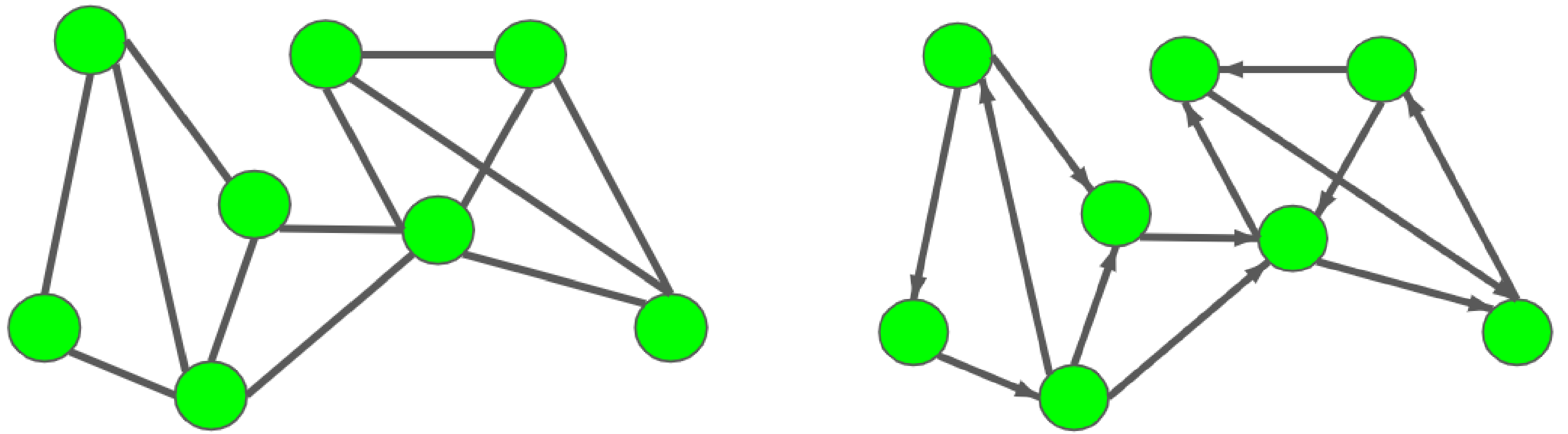
- **A family of methods for finding such similarity-based clusters**
  - “Spectral”: for using the eigenvalues (spectrum) of the *similarity matrix* of the data.
  - Graph clustering, similarity-based clustering

- **The objects to be clustered are not in a vector space.**
  - The primary feature is the similarity between objects.
  - For any pair of objects *i* and *j*, we have a value *s*(*i*,*j*) measuring their similarity; all such values form the similarity matrix.

➔ **Graphs** are intuitive for representing/visualizing such data.

Graph Representation

- **Definition: A graph *G* = (*V*, *E*) is defined by *V*, a set of N *vertices*, and *E*, a set of *edges*.**



Undirected graph Directed graph

- **In spectral clustering, we consider undirected graphs.**

Graph Representation (1/4)

Adjacency matrix W of undirected graph

- $\mathrm{N} \times N$ symmetric binary matrix
- The row and columns are indexed by the vertices and the entries represent the edges of the graph

$$\begin{cases} w_{i,j} = 0 \quad if\ vertices\ i,j\ are\ not\ connected \\ w_{i,j} = 1 \quad \quad if\ vertices\ i,j\ are\ connected \end{cases}$$

- Simple graph = zero diagonal

Graph Representation (2/4)

- **Weighted adjacency matrix (sometimes called affinity matrix)**
  - Allow values other than 0 or 1
  - Each edge is weighted by pairwise similarity

$$\begin{cases} w_{i,j} = 0 & if\ i,j\ are\ not\ connected \\ w_{i,j} = s(i,j) & if\ i,j\ are\ connected \end{cases}$$

- **$w_{i,j}$ may be defined through some kernel functions.**

Graph Representation (3/4)

Degree matrix D of undirected graph

- $N \times N$ diagonal matrix that contains information about the degree of each vertex.
- Degree $d(v_i)$ of a vertex v_i: # of edges incident to the vertex.
  - Extended to sum of weights from edges incident to the vertex.
- So, we have:

$$\mathbf{D} = \begin{bmatrix} d(v_1) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & d(v_N) \end{bmatrix}$$

Graph Representation (4/4)

Laplacian matrix L of undirected graph

- $\mathbf{L} = \mathbf{D} - \mathbf{W}$ (Degree-Affinity) (Unnormalized)
- $\mathbf{L}$ is symmetric and positive semi-definite
- N non-negative real-valued eigenvalues
- The smallest eigen-value is 0, the corresponding eigenvector is the 1-vector (all elements being 1).
- The smallest non-zero eigenvalue of **L** is called the spectral gap.

Spectral Clustering

A Graph Cut Formulation

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Define the graph partition formulation

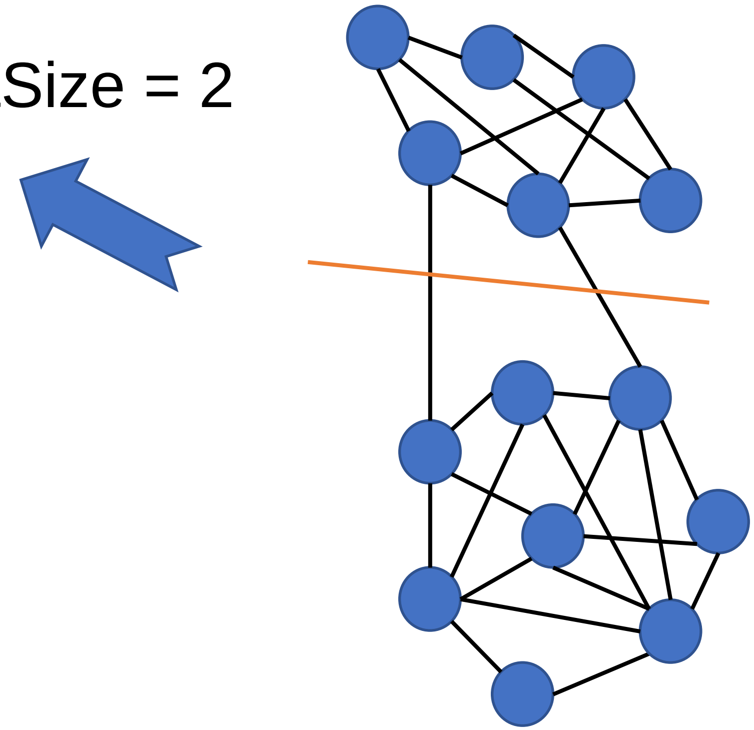
Objective

Learn how to solve simple graph partition

Clustering as Graph Partition/Cut

- Find a partition of a graph such that the edges between different groups have a very low weight while the edges within a group have high weight.
- E.g., minimum cut
- More general, consider weighted edges.

CutSize = 2



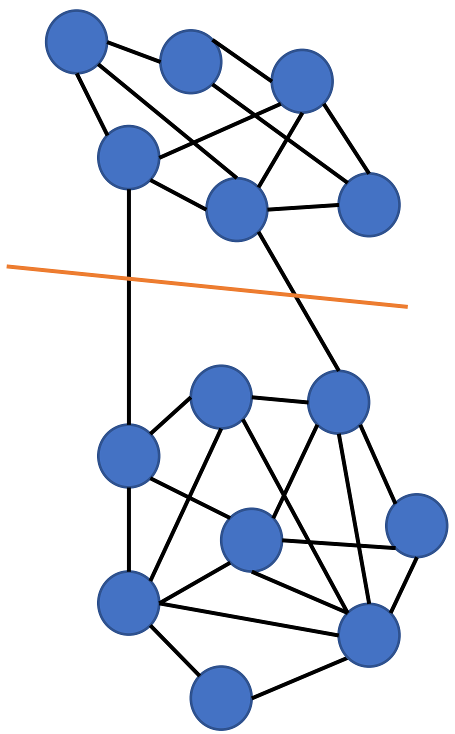
2-way Spectral Graph Partitioning

- **Weighted adjacency matrix W**
  - $w_{i,j}$: the weight between two vertices *i* and *j*
- **(Cluster) Membership vector q**

$$q_i = \begin{cases} 1 & i \in Cluster\ A \\ -1 & i \in Cluster\ B \end{cases}$$

$$\mathbf{q} = \underset{\mathbf{q} \in [-1,\ \ 1]^n}{\operatorname{argmin}}\ CutSize\,,$$

$$CutSize = \mathrm{J} = \frac{1}{4}\sum_{i,j}(q_i - q_j)^2 w_{i,j}$$



Solving the Optimization Problem

$$\mathbf{q} = \underset{\mathbf{q}\in[-1,\quad 1]^n}{\operatorname{argmin}} \frac{1}{4}\sum_{i,j}(q_i - q_j)^2 w_{i,j}\ ,$$

- Directly solving the above problem requires combinatorial search → exponential complexity
- How to reduce the computational complexity?

Relaxation Approach

- **Key difficulty: $\mathbf{q_i}$ has to be either -1,1.**
  - Relax q_i to be any real number.
  - Impose Constraint: $\sum_{i=1}^{n} q_i^2 = n$

$$J = \frac{1}{4}\sum_{i,j}(q_i - q_j)^2 w_{i,j} = \frac{1}{4}\sum_{i,j}(q_i^2 - 2q_i q_j + q_j^2) w_{i,j}$$

$$= \frac{1}{4}\sum_i 2q_i^2\left(\sum_j w_{i,j}\right) - \frac{1}{4}\sum_{i,j} 2q_i q_j w_{i,j}$$

$$= \frac{1}{2}\sum_i q_i^2 d_i - \frac{1}{2}\sum_{i,j} q_i (d_i \delta_{i,j} - w_{i,j}) q_j$$

where $d_i = \sum_j w_{i,j}$ and $D \equiv [d_i \delta_{i,j}]$

$$\Rightarrow \; J = \frac{1}{2}\mathbf{q}^T(\mathbf{D} - \mathbf{W})\mathbf{q}$$

Relaxation Approach (cont'd)

The final problem formulation:

$$\mathbf{q} = \underset{\mathbf{q}}{\operatorname{argmin}}\ J = \underset{\mathbf{q}}{\operatorname{argmin}}\, \mathbf{q}^T(\mathbf{D} - \mathbf{W})\mathbf{q}\ ,$$

subject to $\sum_{i=1}^{n} q_i^2 = n$

Solution: the second minimum eigenvector for D-W

$$(\mathbf{D} - \mathbf{W})\mathbf{q} = \lambda_2\ \mathbf{q}$$

Graph Laplacian

- $L = D - W$
- **L is semi-positive definitive matrix.**
  - For any $\mathbf{x}$, we have $\mathbf{x}^T \mathbf{L} \mathbf{x} \geq 0$. (Why?)
- **Minimum eigenvalue $\lambda_1 = 0$ (what is the eigenvector?)**

$$0 = \lambda_1 < \lambda_2 < \lambda_3 \ldots < \lambda_k$$

- **The eigenvector that corresponds to the second minimum eigenvalue λ_2 gives the best bipartite graph partition.**

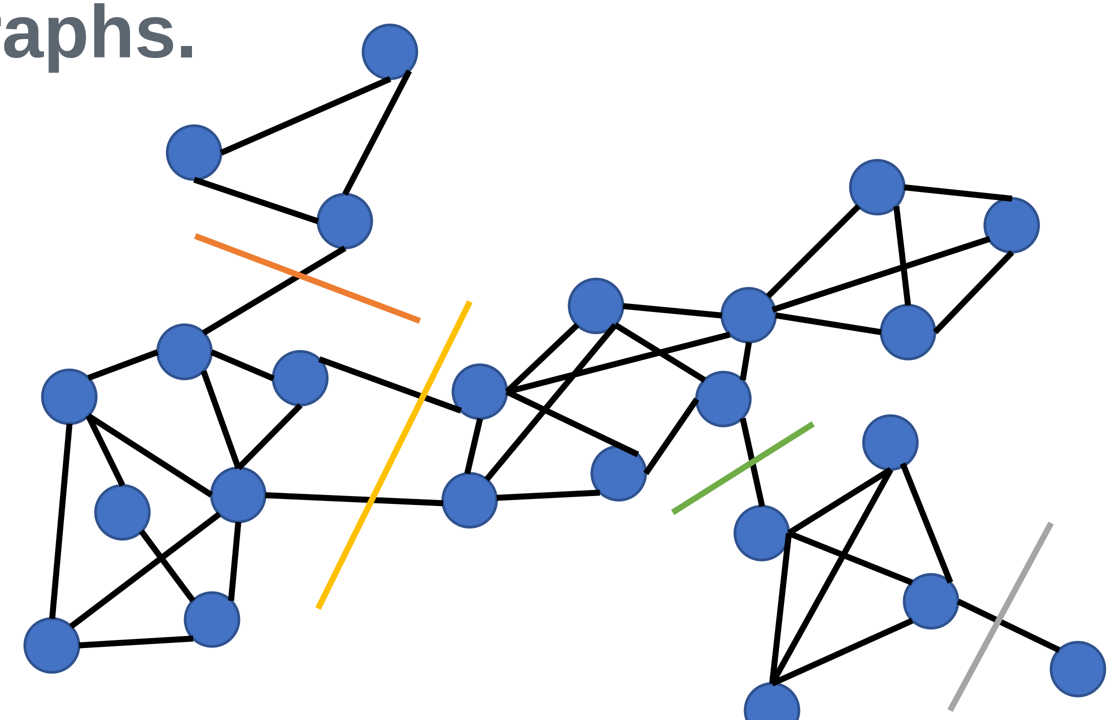
Recovering the Partitions

- Due to the relaxation, q can be any number (not just -1 and 1)
- How to construct the partition based on the eigenvector?
- A simple strategy :

$$A = \{i | q_i < 0\}, \quad B = \{i | q_i \geq 0\}$$

One Obvious Drawback

- **Minimum cut does not balance the size of bipartite graphs.**



- **How should we consider other factors like the sizes of the partitions?**

Spectral Clustering

Going Beyond MinCut

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

Objective

Discuss several graph cut approaches

Objective

Illustrate the algorithm through an example

MinCut

- In MinCut, we used the following objective function:

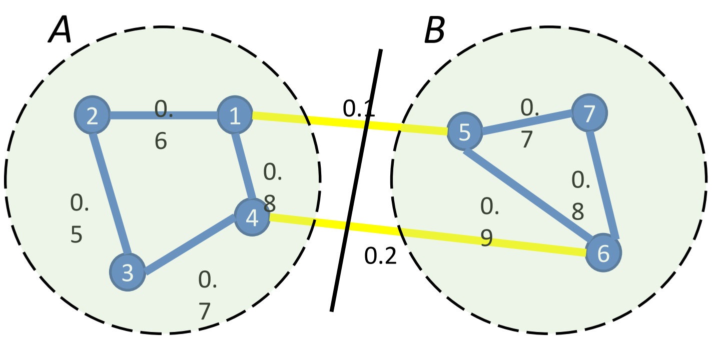
$$J_{MinCut} = Cut(A, B)$$

- We noted one drawback of MinCut: the sizes of the partitions are not considered.

- A few extensions exist.

Characterizing Graph Cut

- $Cut(A,B) = \sum_{i\in A, j\in B} w_{ij}$ $e.g., Cut(A,B) = 0.3$
- $Cut(A,A) = \sum_{i\in A, j\in A} w_{ij}$ $e.g., Cut(A,A) = 2.6$
- $Cut(B,B) = \sum_{i\in B, j\in B} w_{ij}$ $e.g., Cut(B,B) = 2.4$
- $Vol(A) = \sum_{i\in A} \sum_{i=1}^{n} w_{ij}$ $e.g., Vol(A) = 5.5$
- $Vol(B) = \sum_{i\in B} \sum_{i=1}^{n} w_{ij}$ $e.g., Vol(B) = 5.1$
- $|A| = 4, |B| = 3$



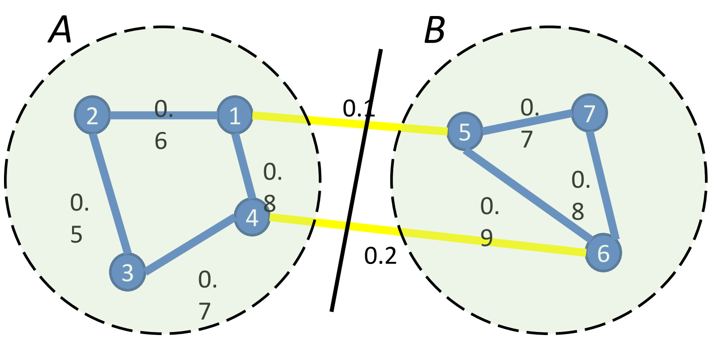
The Ratio Cut Method

The Objective function:

$$J_{RatioCut}(A,B) = Cut(A,B)\left(\frac{1}{|A|} + \frac{1}{|B|}\right)$$

Attempts to produce balanced clusters.

Example: $J_{RatioCut}(A,B) = \frac{7}{40}$



The Ratio Cut Method (cont'd)

- Similar to MinCut, the solution can be found by the following generalized eigenvalue problem:

$$(\mathbf{D} - \mathbf{W})\mathbf{q} = \lambda \mathbf{D}\mathbf{q}$$

$$\mathbf{L}\mathbf{q} = \lambda \mathbf{D}\mathbf{q}$$

Normalized Cut (NCut)

- In Ratio Cut, the balance of the partitions is defined based on the number of vertices.
- We may consider the “size” of a set based on weights of its edges ➔ Ncut
- The objective function is:

$$J_{NCut}(A,B) = Cut(A,B)\left(\frac{1}{Vol(A)} + \frac{1}{Vol(B)}\right)$$

Example: $J_{NCut}(A,B) = 0.1134$

Additional Considerations

- **In clustering, we should also consider within-cluster connections.**
- **A good partition should consider**
  - Inter-cluster connections, and
  - Intra-cluster connections.

MinMaxCut

- 1st constrant: inter-connection should be minimized: $MinCut(A, B)$

- 2nd constraint: intra-connection should be maximized : $MaxCut(A, A)$ and $MaxCut(B, B)$

- These requirements may be simultaneously satisfied by minimizing the objective function:

$$J_{MinMaxCut}(A, B) = Cut(A, B)\left(\frac{1}{Cut(A,A)} + \frac{1}{Cut(B,B)}\right)$$

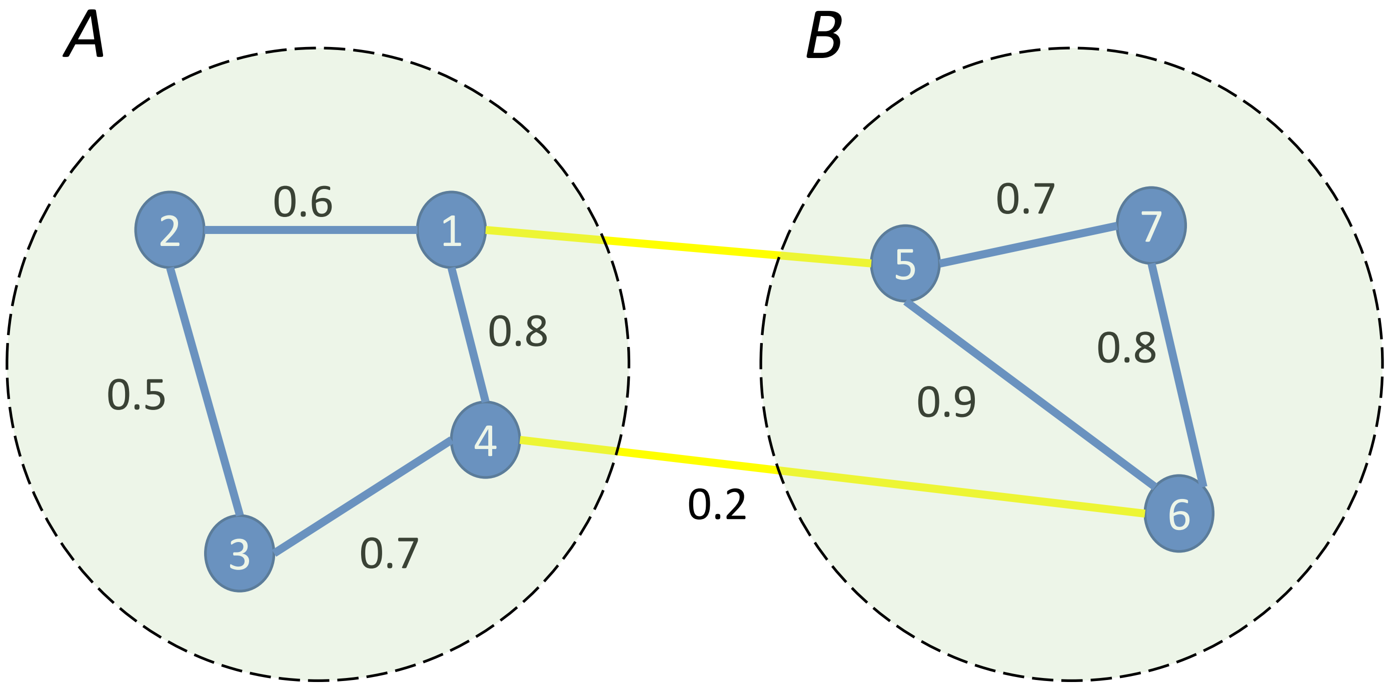
Example: $J_{MinMaxCut}(A, B) = 0.240$

Normalized and MinMaxCut Methods

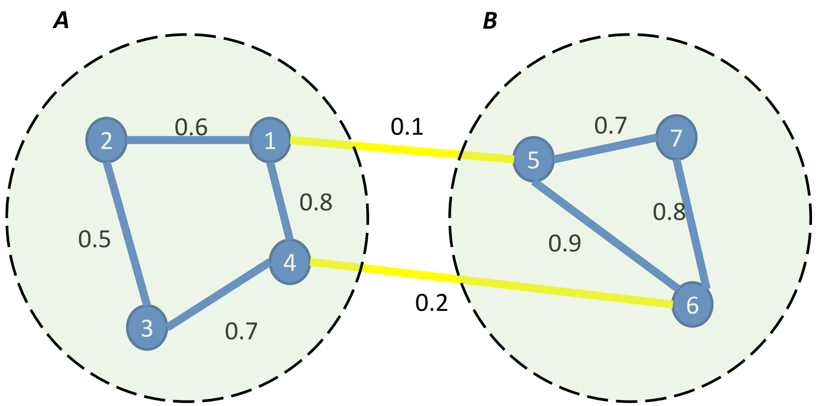
- Similar to before, we may relax the indicator vector q to real values.

- For both NCut and MinMaxCut, the solution may be found by solving generalized eigenvalue problems.

An Illustrative Example

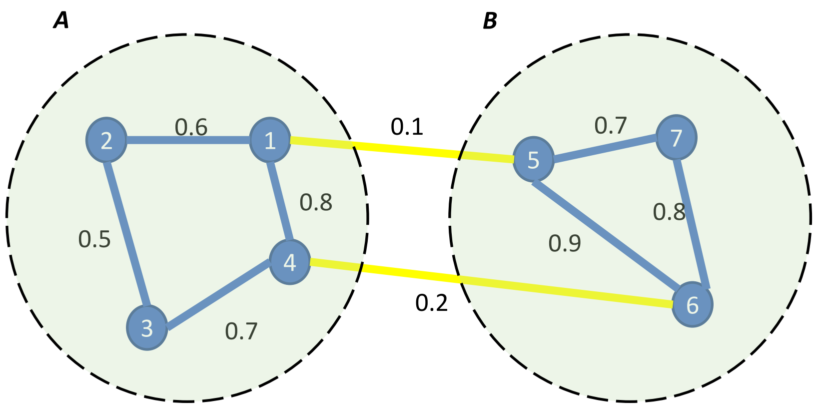


Graph and Similarity Matrix



	x1	x2	x3	x4	x5	x6	x7
x1	0	0.6	0	0.8	0.1	0	0
x2	0.6	0	0.5	0	0	0	0
x3	0	0.5	0	0.7	0	0	0
x4	0.8	0	0.7	0	0	0.2	0
x5	0.1	0	0	0	0	0.9	0.7
x6	0	0	0	0.2	0.9	0	0.8
x7	0	0	0	0	0.7	0.8	0

Graph and Laplacian Matrix

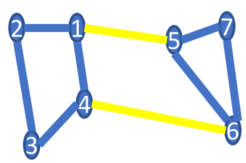


	x1	x2	x3	x4	x5	x6	x7
x1	1.5	-0.6	0	-0.8	-0.1	0	0
x2	-0.6	1.1	-0.5	0	0	0	0
x3	0	-0.5	1.2	-0.7	0	0	0
x4	-0.8	0	-0.7	1.7	0	-0.2	0
x5	-0.1	0	0	0	1.7	-0.9	-0.7
x6	0	0	0	-0.2	-0.9	1.9	-0.8
x7	0	0	0	0	-0.7	-0.8	1.5

Solve Eigen Problem

Pre-processing

- Build Laplacian matrix L of the graph.



Λ =

0
0.1588
1.2705
1.3692
2.2751
2.6238
2.9027

X =

0.378	-0.2962	0.3027	-0.6041	0.0429	0.3638	-0.4226
0.378	-0.3805	0.6392	0.4487	-0.0125	-0.233	0.2192
0.378	-0.3608	-0.5812	0.4834	-0.0221	0.2736	-0.2832
0.378	-0.2649	-0.398	-0.4373	0.0429	-0.3899	0.5323
0.378	0.4298	0.0443	0.0159	-0.6004	0.4291	0.3544
0.378	0.406	-0.0317	0.0012	-0.2174	-0.6116	-0.5196
0.378	0.4665	0.0247	0.0923	0.7667	0.1681	0.1195

Find

- Eigenvalues Λ and eigenvectors **x** of matrix **L**.
- Map vertices to the corresponding components of the 2nd eigenvector.

x1	-0.2962
x2	-0.3805
x3	-0.3608
x4	-0.2649
x5	0.4298
x6	0.406
x7	0.4665

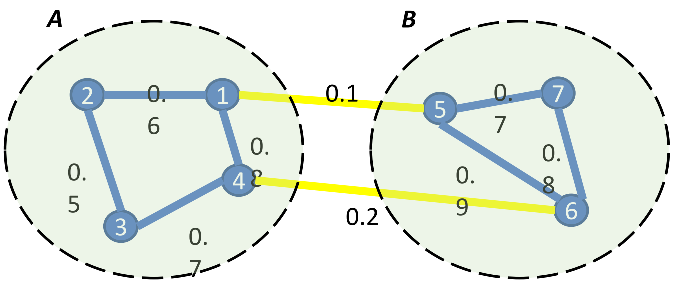
Spectral Clustering

x1	-0.2962
x2	-0.3805
x3	-0.3608
x4	-0.2649
x5	0.4298
x6	0.406
x7	0.4665

Split at value 0
Cluster A: Negative points
Cluster B: Positive Points

x1	-0.2962
x2	-0.3805
x3	-0.3608
x4	-0.2649

X5	0.4298
X6	0.406
X7	0.4665



Spectral Clustering

Practical Considerations in Implementation

Ira A. Fulton Schools of Engineering
Arizona State University

Objective

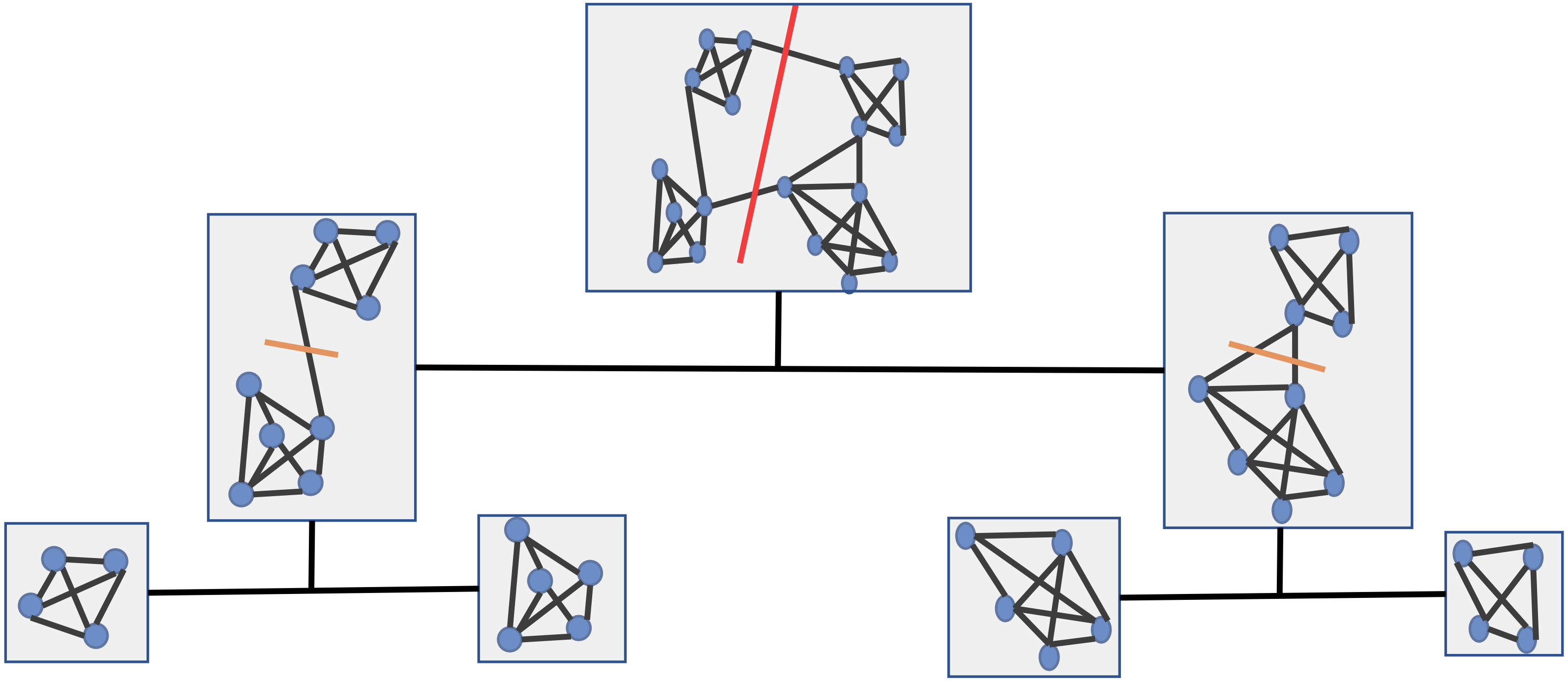
Objective

Discuss several practical implementation issues

Recursive Bi-partitioning

- Recursively apply bi-partitioning algorithm in a hierarchical divisive manner.

- Disadvantages: inefficient, stability issues.



K-way Graph Cuts

Generalizing the 2-way objective functions :

$$J_{RatioCut}(A_1, \dots, A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \overline{A_i})}{|A_i|}$$

$$J_{NCut}(A_1, \dots, A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \overline{A_i})}{Vol(A_i)}$$

$$J_{MinMaxCut}(A_1, \dots, A_k) = \sum_{i=1}^{k} \frac{Cut(A_i, \overline{A_i})}{Cut(A_i, A_i)}$$

Implementation Considerations (1/4)

- **Preprocessing: spectral clustering methods can be interpreted as tools for analysis of the block structure of the similarity matrix.**
  - Building such matrices may certainly ameliorate the results.

- **When building graphs from real data**
  - Calculation of the similarity matrix is not evident.
  - Choosing the similarity function can highly affect the results of the following steps.
  - A Gaussian kernel is often chosen, but other similarities like cosine similarity might be proper for specific applications.

Implementation Considerations (2/4)

Graph and similarity matrix construction: Laplacian matrices are generally chosen to be positive and semi-definite thus their eigenvalues will be non-negatives.

- A few variants

Unnormalized	$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$
symmetric	$\boldsymbol{L}_{Sy} = \boldsymbol{D}^{-1/2}\boldsymbol{L}\boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$
Asymmetric	$\boldsymbol{L}_{As} = \boldsymbol{D}^{-1}\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-1}\boldsymbol{W}$

Implementation Considerations (3/4)

Computing the eigenvectors.

- Efficient methods exist for sparse matrices.

Different ways of building the similarity graphs

- ε-neighborhood graph.
- k-nearest neighbor graph.
- fully connected graph.

Implementation Considerations (4/4)

Choosing *k*:

- Similar to k-means, there are many heuristics to use.
- The eigengap heuristic: to choose a k such that first k eigenvalues are very small but the (k+1)th one is relatively large.

Clustering: simple algorithms other than k-means can be used in the last stage, such as simple linkage, k-lines, elongated k-means, mixture model, etc.

Recap: Pros and Cons of Spectral Clustering

Advantages:

- Does not make strong assumptions on the forms of the clusters.
- Easy to implement, and can be implemented efficiently even for large data sets as long as the similarity graph is sparse.
- Good clustering results.
- Reasonably fast for sparse data sets of several thousand elements.

Disadvantages:

- May be sensitive to choice of parameters for neighborhood graph.
- Computationally expensive for large datasets.