Machine Learning for Data Science (CS4786)
Lecture 11

Spectral Clustering

Mar 10, 2015

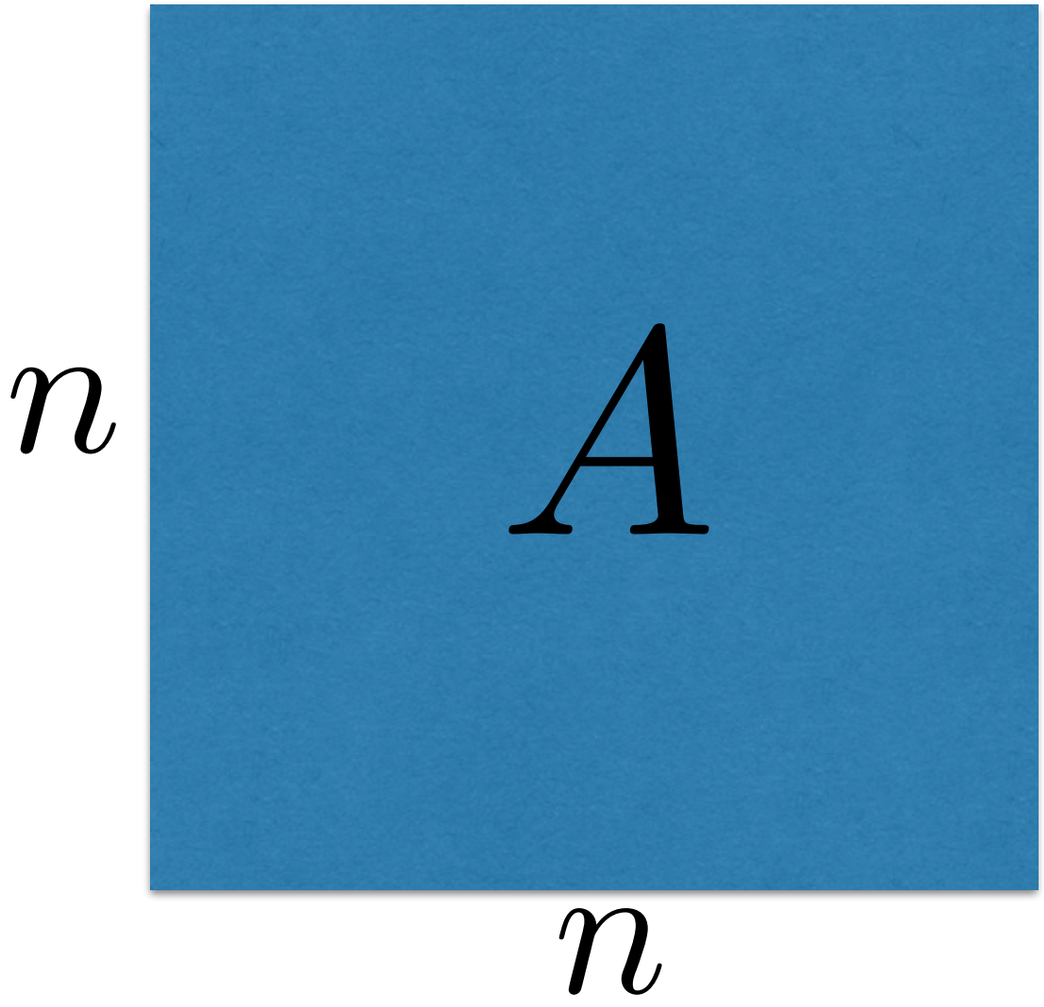
Course Webpage :

http://www.cs.cornell.edu/Courses/cs4786/2015sp/

Spectral Clustering

Input: Similarity matrix A

$A_{i,j} = A_{j,i} > 0$ indicates similarity between elements x_i and x_j



Example: $A_{i,j} = \exp(-\sigma d(x_i, x_j))$

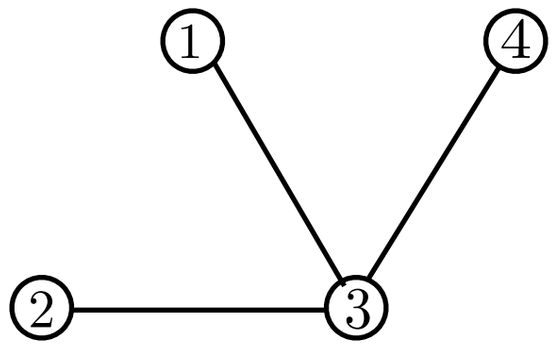
A is adjacency matrix of a graph

Spectral Clustering Algorithm (Unnormalized)

1. Given matrix A calculate diagonal matrix D s.t. $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$
2. Calculate the Laplacian matrix $L = D - A$
3. Find eigen vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of L (ascending order of eigenvalues)
4. Pick the K eigenvectors with smallest eigenvalues to get $\mathbf{y}_1, \ldots, \mathbf{y}_n \in \mathbb{R}^K$
5. Use K-means clustering algorithm on $\mathbf{y}_1, \ldots, \mathbf{y}_n$

Example

Graph Clustering



- Fact: For a connected graph, exactly one, the smallest of eigenvalues is 0, corresponding eigenvector is $\mathbf{1} = (1, \ldots, 1)^\top$

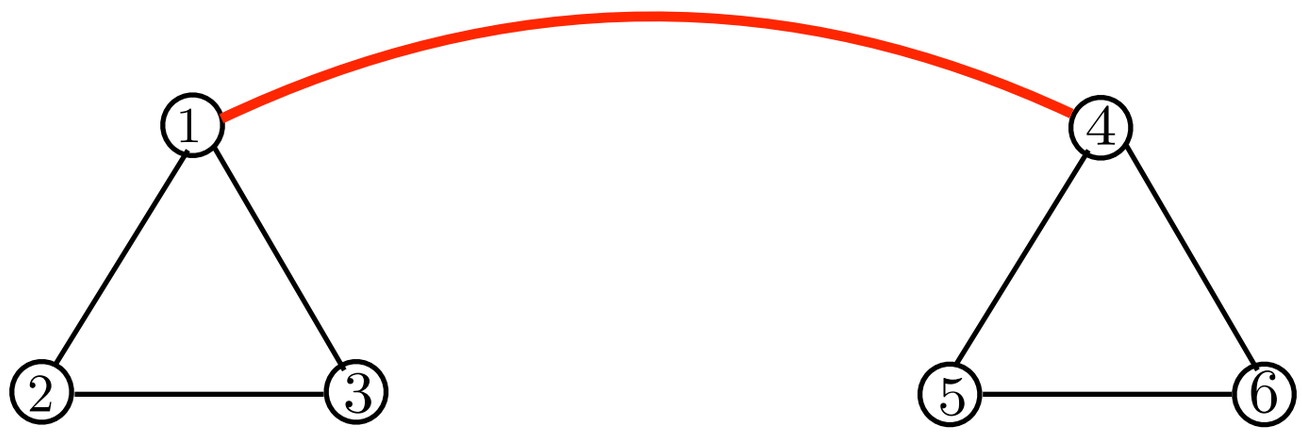
  Proof: Sum of each row of L is 0 because $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$ and $L = D - A$

Graph Clustering

- Fact: For general graph, number of 0 eigenvalues correspond to number of connected components. The corresponding eigenvectors are all 1's on the nodes of connected components

  Proof: L is block diagonal. Use connected graph result on each component.

Graph Clustering



- Fact: For general graph, number of 0 eigenvalues correspond to number of connected components. The corresponding eigenvectors are all 1's on the nodes of connected components

  Proof: L is block diagonal. Use connected graph result on each component.

Graph Clustering: Cuts

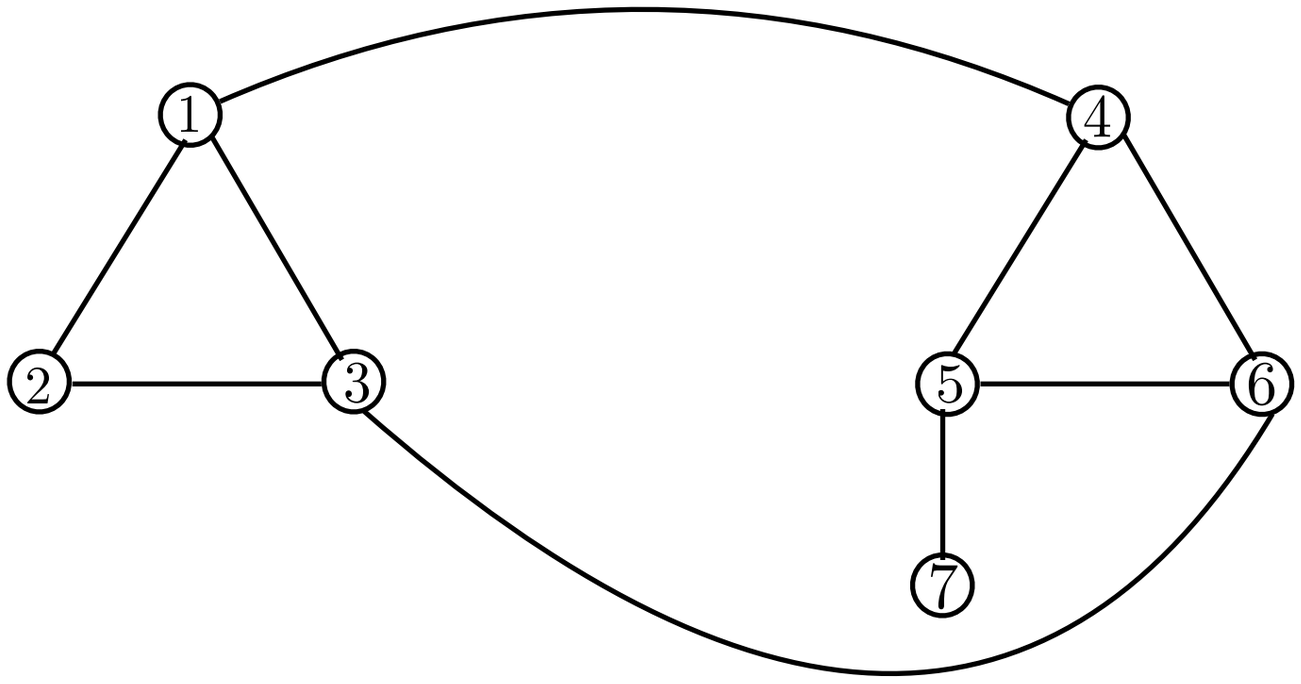
- Partition nodes so that as few edges are cut (Mincut)
- What has this got to do with the Laplacian matrix?

Normalized Cut

- Why cut is perhaps not a good measure?

Normalized Cut

- Why cut is perhaps not a good measure?



Normalized Cut

- Why cut is perhaps not a good measure?
- Normalized cut: Minimize sum of ratio of number of edges cut per cluster and number of edges within cluster

$$\text{NCUT} = \sum_j \frac{\text{CUT}(C_j)}{\text{Edges}(C_j)}$$

- Example $K = 2$

$$\text{CUT}(C_1, C_2)\left(\frac{1}{\text{Edges}(C_1)} + \frac{1}{\text{Edges}(C_2)}\right)$$

- Minimize $\text{CUT}(C_1, C_2)$ s.t. $\text{Edges}(C_1) = \text{Edges}(C_2)$

Normalized Cut

- Minimize $c^\top Lc$ s.t. $c^\top Dc = 1$, $c \perp \mathbf{1}$
- Minimizing over cluster assignments is computationally hard instead relax to real valued c's
- Solution: Find smallest eigen vectors of $\tilde{L} = I - D^{-1/2}WD^{-1/2}$
- $\mathbf{y}$'s given by these eigenvectors of the normalized Laplacian can then be clustered using K-means algorithm

Normalized Cut: Alternate view

- If we perform random walk on graph, its the partition of graph into group of vertices such that the probability of transiting from one group to another is minimized

- Transition matrix: $D^{-1}W$

- Largest eigenvalues and eigenvectors of above matrix correspond to smallest eigenvalues and eigen vectors of $\tilde{L}$