Machine Learning for Data Science (CS4786) Lecture 11

Spectral clustering

Course Webpage :
http://www.cs.cornell.edu/Courses/cs4786/2016sp/

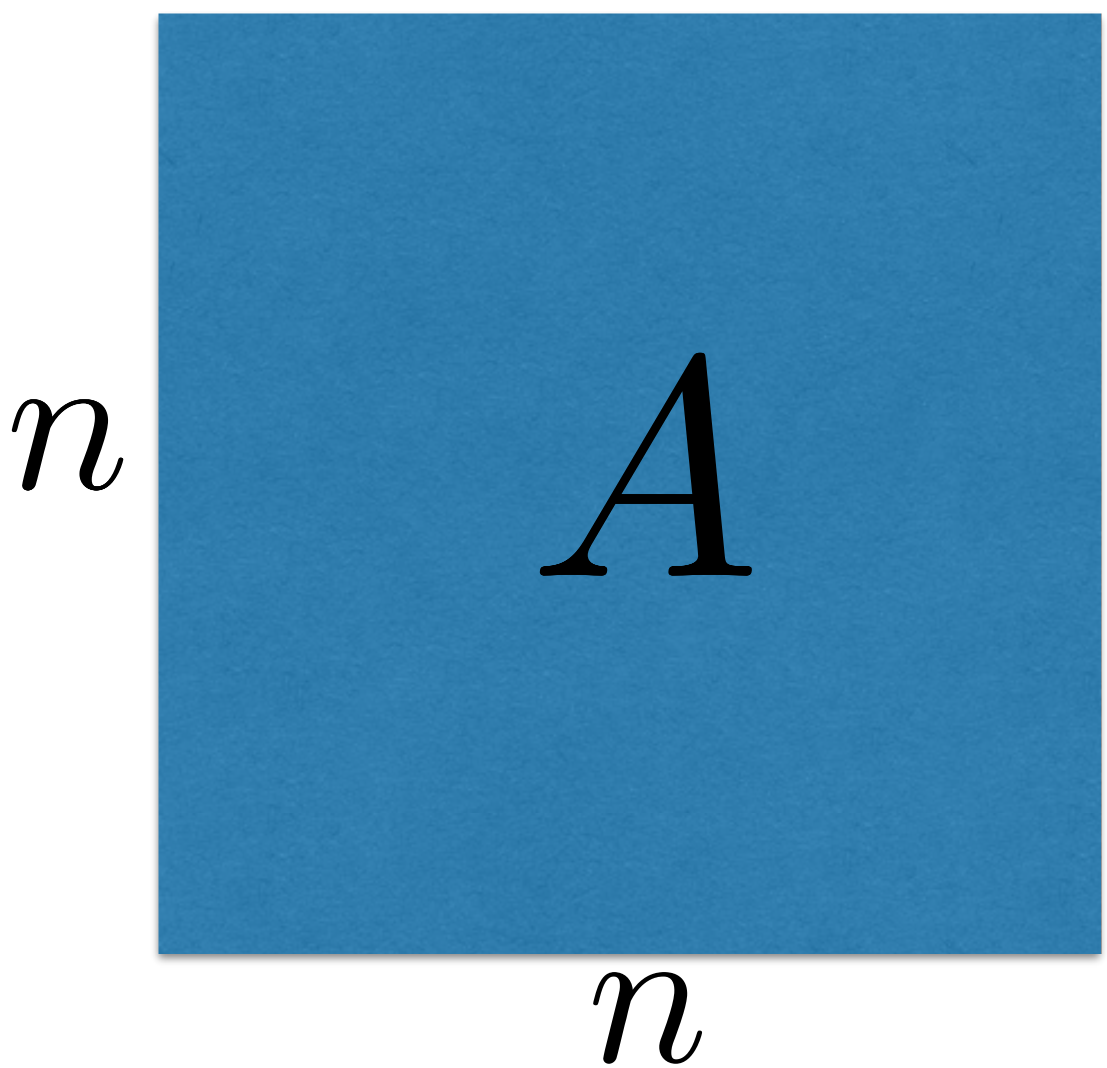
Announcement

1. Assignment $P1$ the Diagnostic assignment 1 will be posted on Tuesday, March 15th.
2. Assignment $P1$ due on Thursday March 17th.
3. Individual submissions. Do not work in groups.
4. $P1$ will be simpler than the regular assignments and no coding.

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

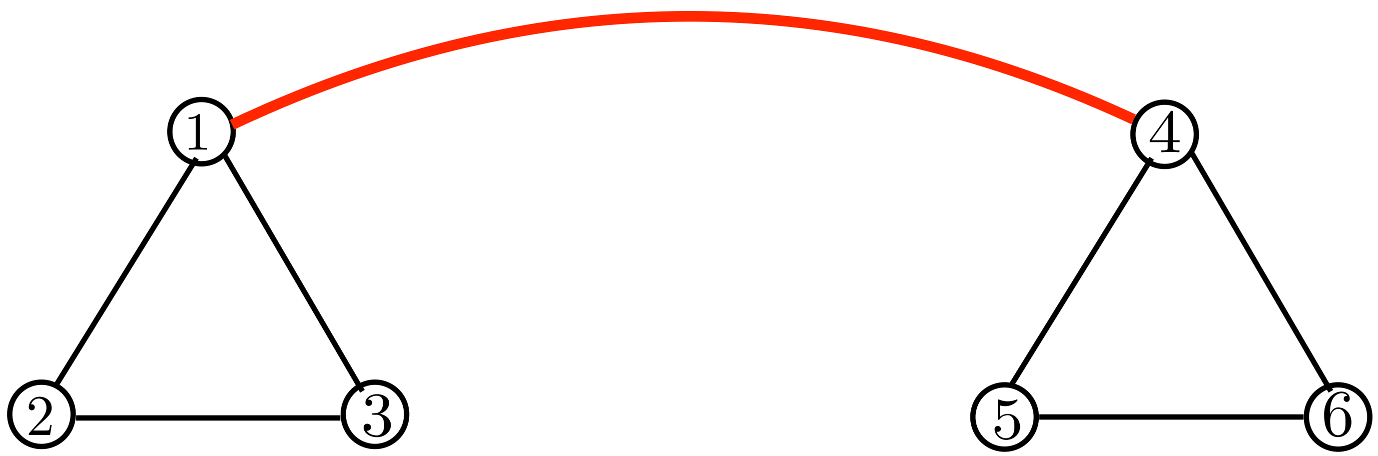
Laplacian Matrix

- Let D be diagonal matrix with $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$ (degree of vertex i)
- Laplacian matrix defined as $L = D - A$
- Fact 1: Laplacian for a connected graph has exactly one 0 eigenvalue, corresponding eigenvector is $\mathbf{1} = (1, \ldots, 1)^\top / \sqrt{n}$

  Proof: Sum of each row of L is 0 as $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$ and $L = D - A$
- Fact 2: For general graph, number of 0 eigenvalues correspond to number of connected components.

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

Graph Clustering



Graph Clustering: Cuts

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

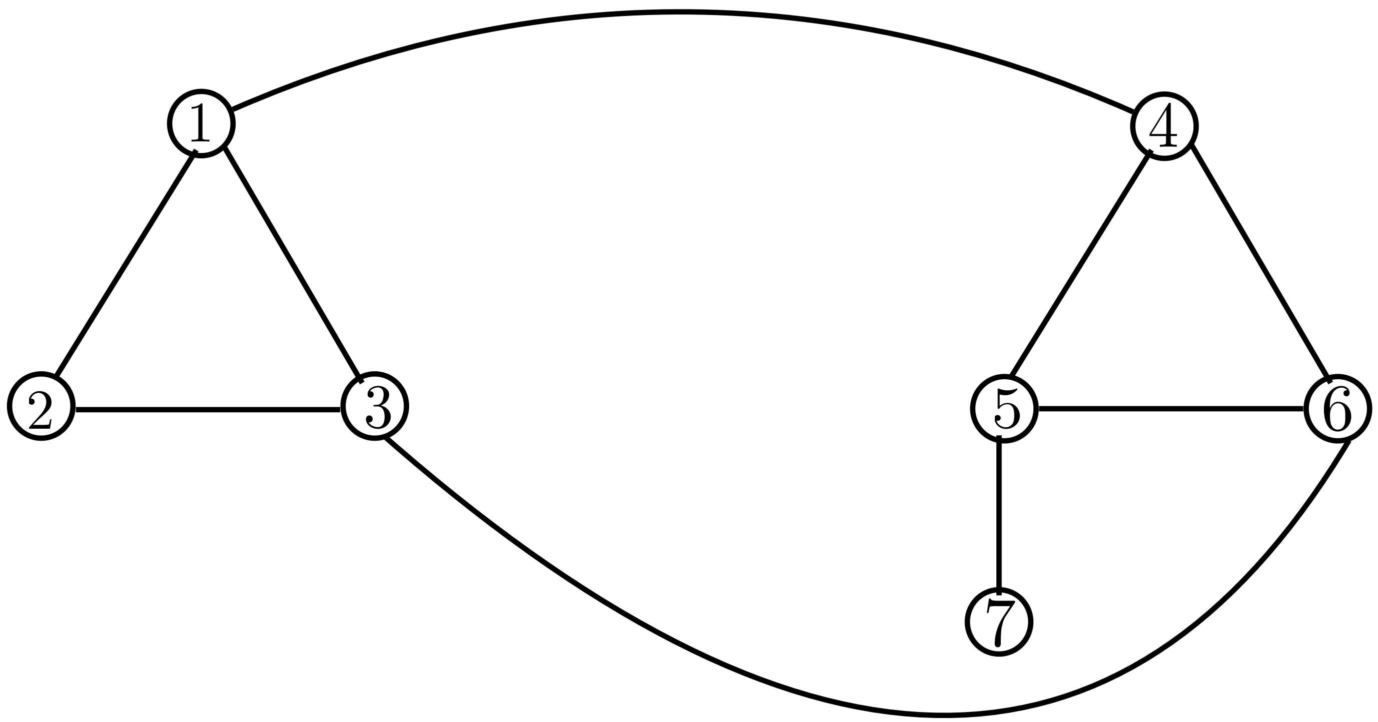
Example

Spectral Clustering (Unnormalized)

- Min-cut on a graph can be efficiently computed
- Why bother with the approximate algorithm
- Is cut even a good measure?

Normalized Cut

- Why cut is perhaps not a good measure?



Ratio Cut

- Why cut is perhaps not a good measure?
- Fixes?

Ratio Cut

- Why cut is perhaps not a good measure?
- Fixes? Perhaps Ratio Cut : $\mathrm{CUT}(C_1, C_2)\left(\frac{1}{|C_1|} + \frac{1}{|C_2|}\right)$
- Set $c_i = \begin{cases} \sqrt{\frac{|C_2|}{|C_1|}} & \text{if } i \in C_1 \\ -\sqrt{\frac{|C_1|}{|C_2|}} & \text{otherwise} \end{cases}$
- Verify that $c^\top L c = n \times \text{Ratio Cut}$ and $\|c\|_2 = \sqrt{n}$ (and $c \perp \mathbf{1}$)
- Relaxed solution is same as Unnormalized Spectral clustering

Normalized Cut

- Normalized cut: Minimize sum of ratio of number of edges cut per cluster and number of edges within cluster

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

- Example $K = 2$

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

- This is an NP hard problem!

Normalized Cut

- Normalized cut: Minimize sum of ratio of number of edges cut per cluster and number of edges within cluster

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

- Example $K = 2$

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

- This is an NP hard problem! …so relax

Normalized Cut

- First note that $\text{Edges}(C_i) = \sum_{k:x_k \in C_i} D_{k,k}$
- Set $c_i = \begin{cases} \sqrt{\frac{\text{Edges}(C_2)}{\text{Edges}(C_1)}} & \text{if } i \in C_1 \\ -\sqrt{\frac{\text{Edges}(C_1)}{\text{Edges}(C_2)}} & \text{otherwise} \end{cases}$
- Verify that $c^\top L c = |E| \times \text{NCut}$ and $c^\top D c = |E|$ (and $Dc \perp \mathbf{1}$)
- Hence we relax Minimize NCUT(C) to

$$\text{Minimize } \frac{c^\top L c}{c^\top D c} \qquad \text{s.t. } Dc \perp \mathbf{1}$$

- Solution: Find second smallest eigenvectors of $\tilde{L} = I - D^{-1/2} A D^{-1/2}$

Spectral Clustering Algorithm (Normalized)

1. Given matrix A calculate diagonal matrix D s.t. $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$
2. Calculate the normalized Laplacian matrix $\tilde{L} = I - D^{-1/2} A D^{-1/2}$
3. Find eigen vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ of $\tilde{L}$ (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 $\mathbf{y}_1, \ldots, \mathbf{y}_n$

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: $D^{-1}A$
- Largest eigenvalues and eigenvectors of above matrix correspond to smallest eigenvalues and eigenvectors of $D^{-1}L = I - D^{-1}A$
- For K-nearest neighbor graph (K-regular), same as normalized Laplacian