Machine Learning for Data Science (CS4786)
Lecture 14

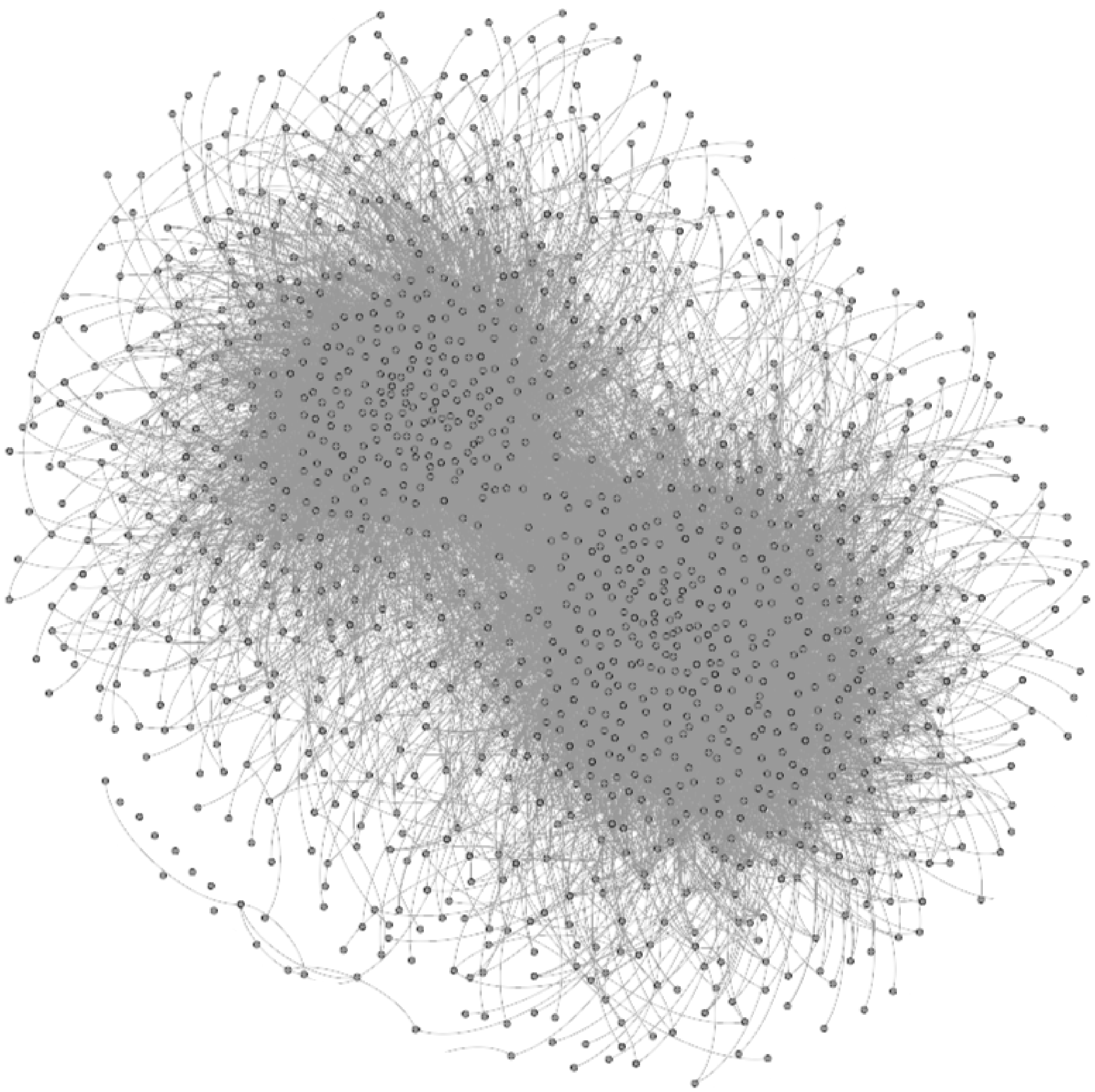
Spectral Clustering

Course Webpage :
http://www.cs.cornell.edu/Courses/cs4786/2017fa/

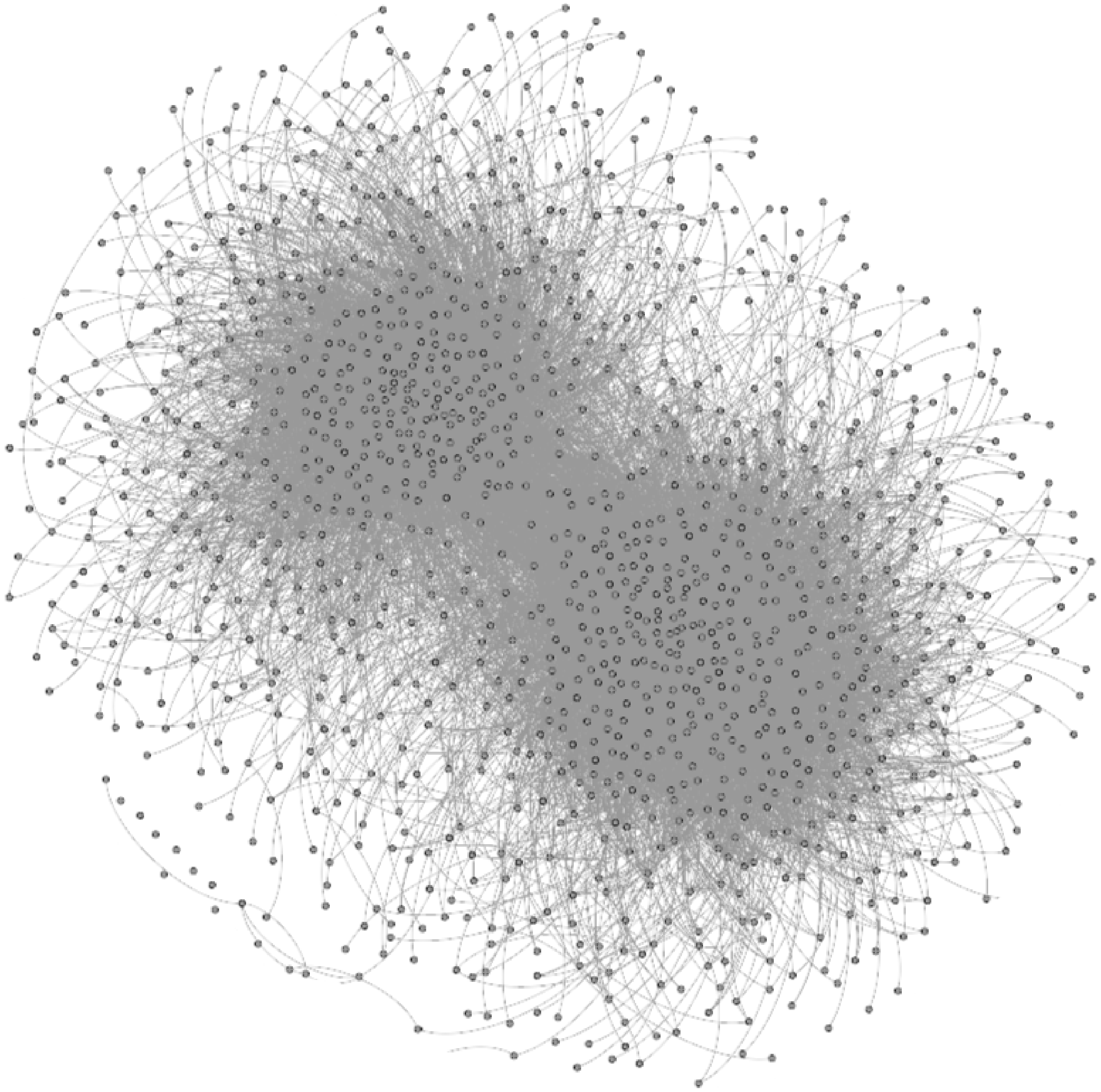
Announcement

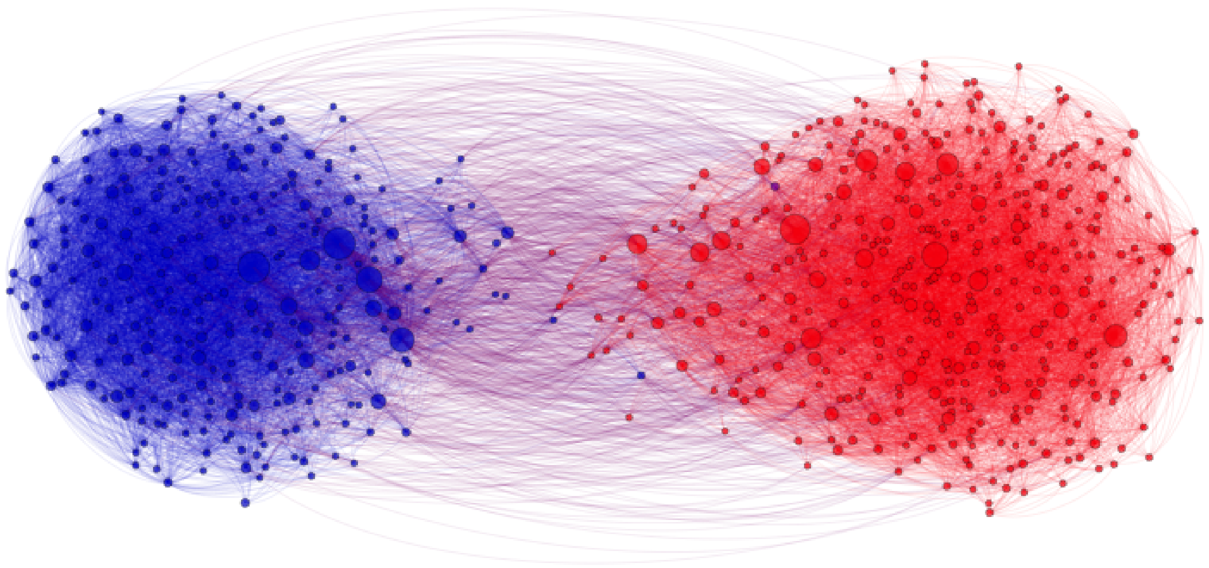
- Competition I data is out:
- https://confluence.cornell.edu/x/f3zHF

Spectral Clustering

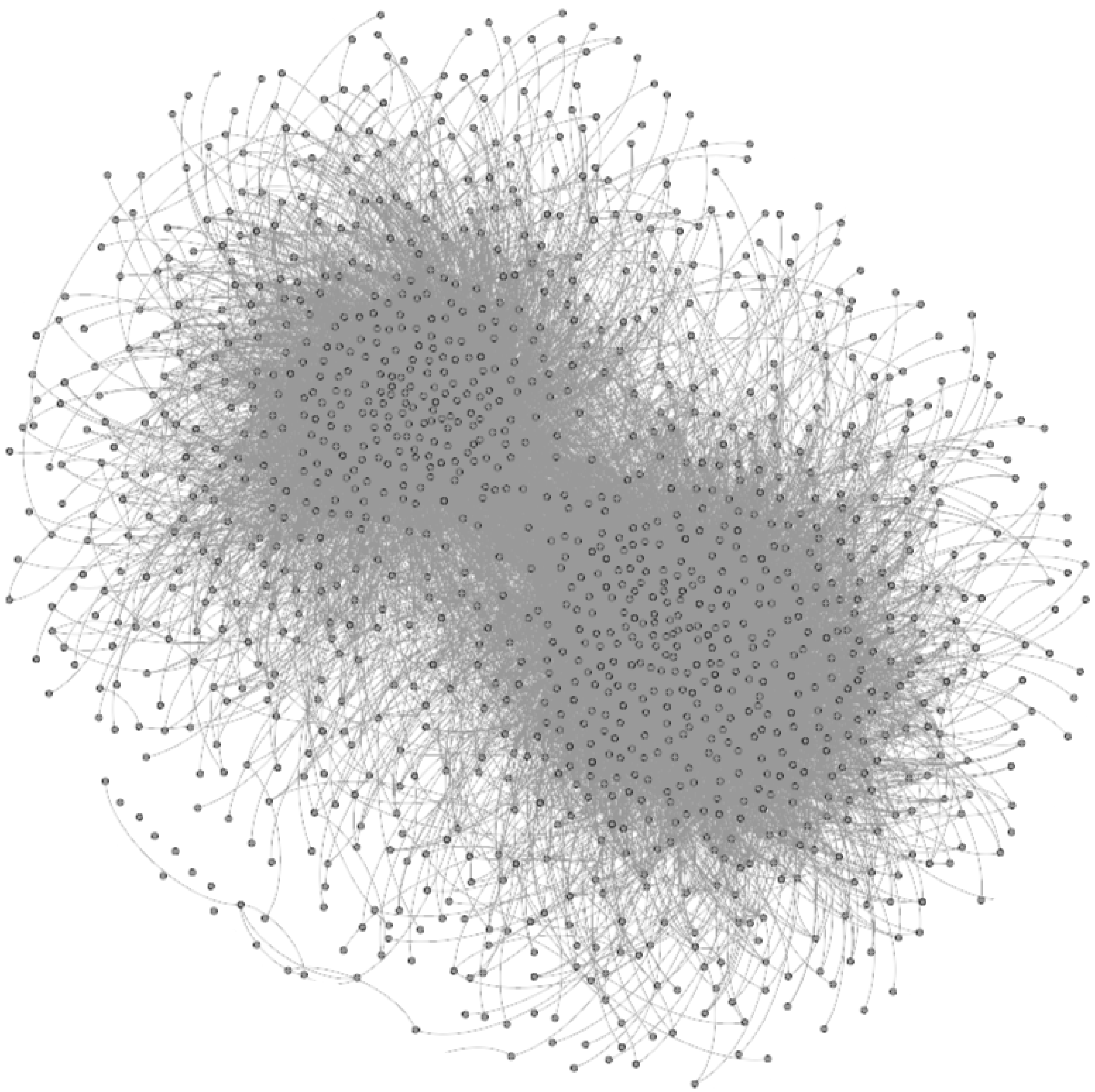


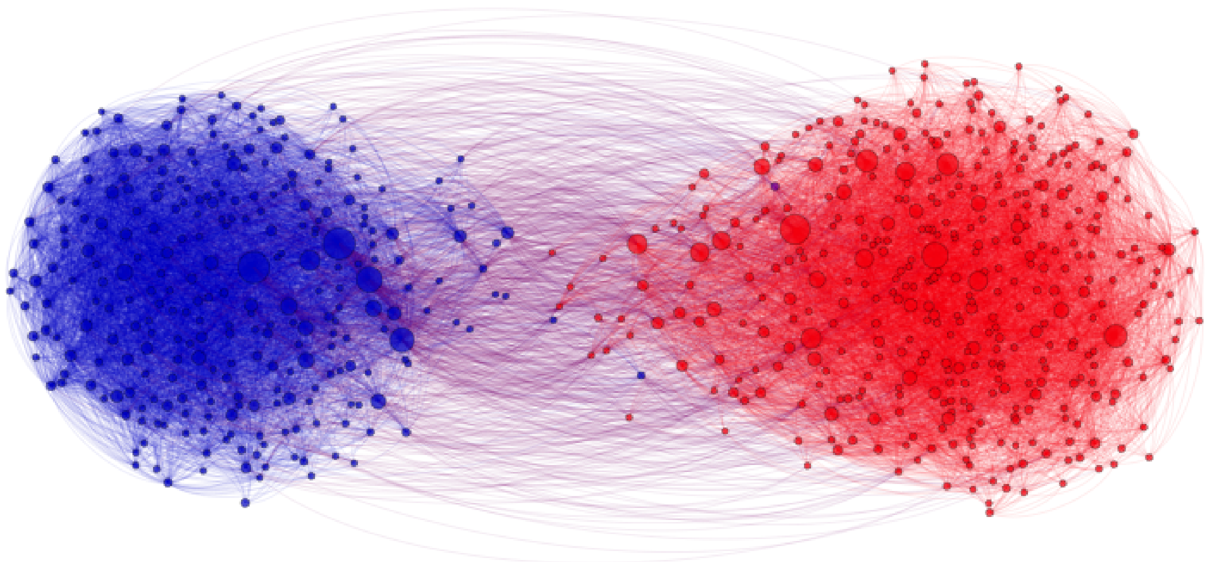
Spectral Clustering





Spectral Clustering

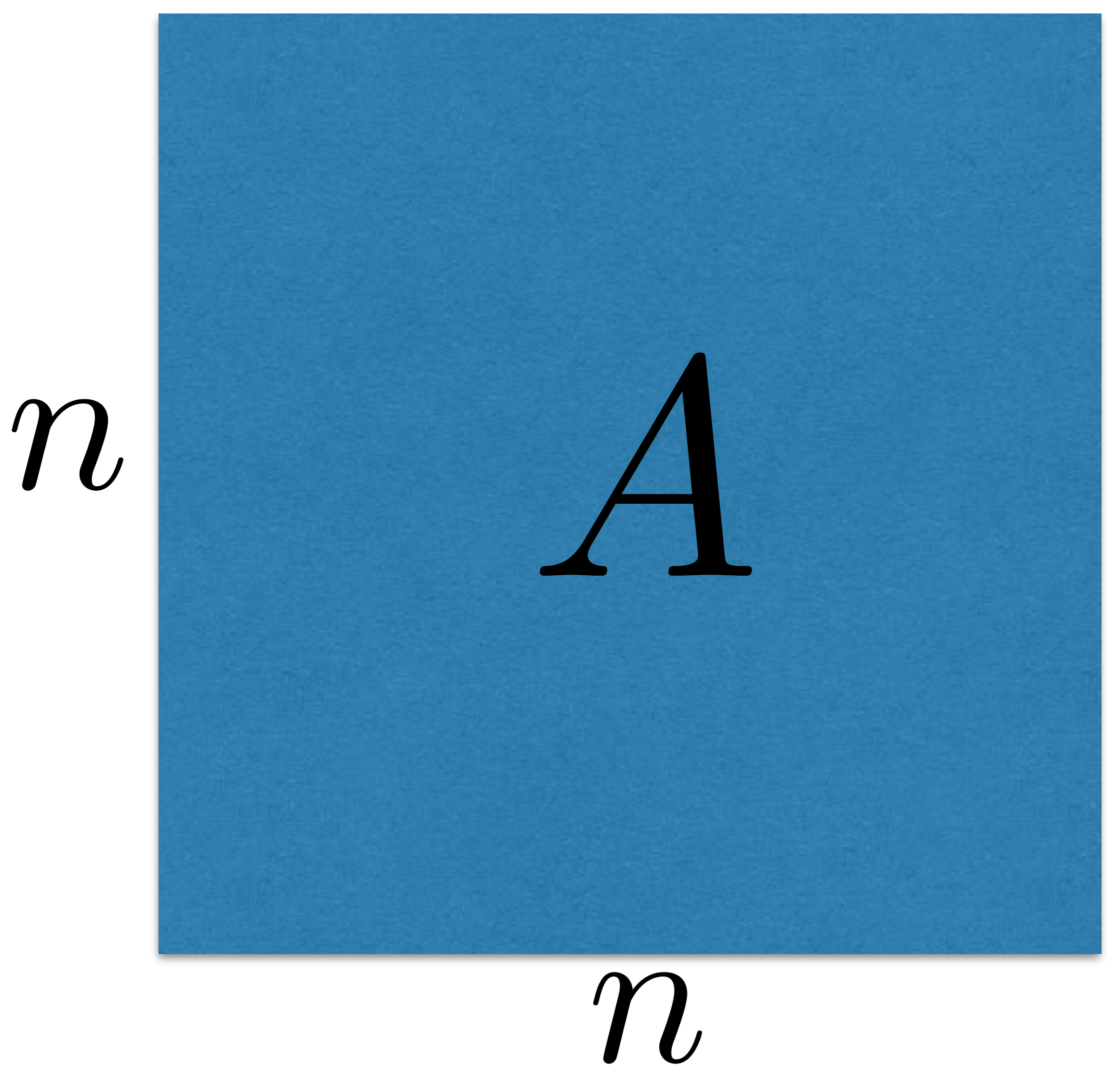




- Cluster nodes in a graph.
- Analysis of social network data.

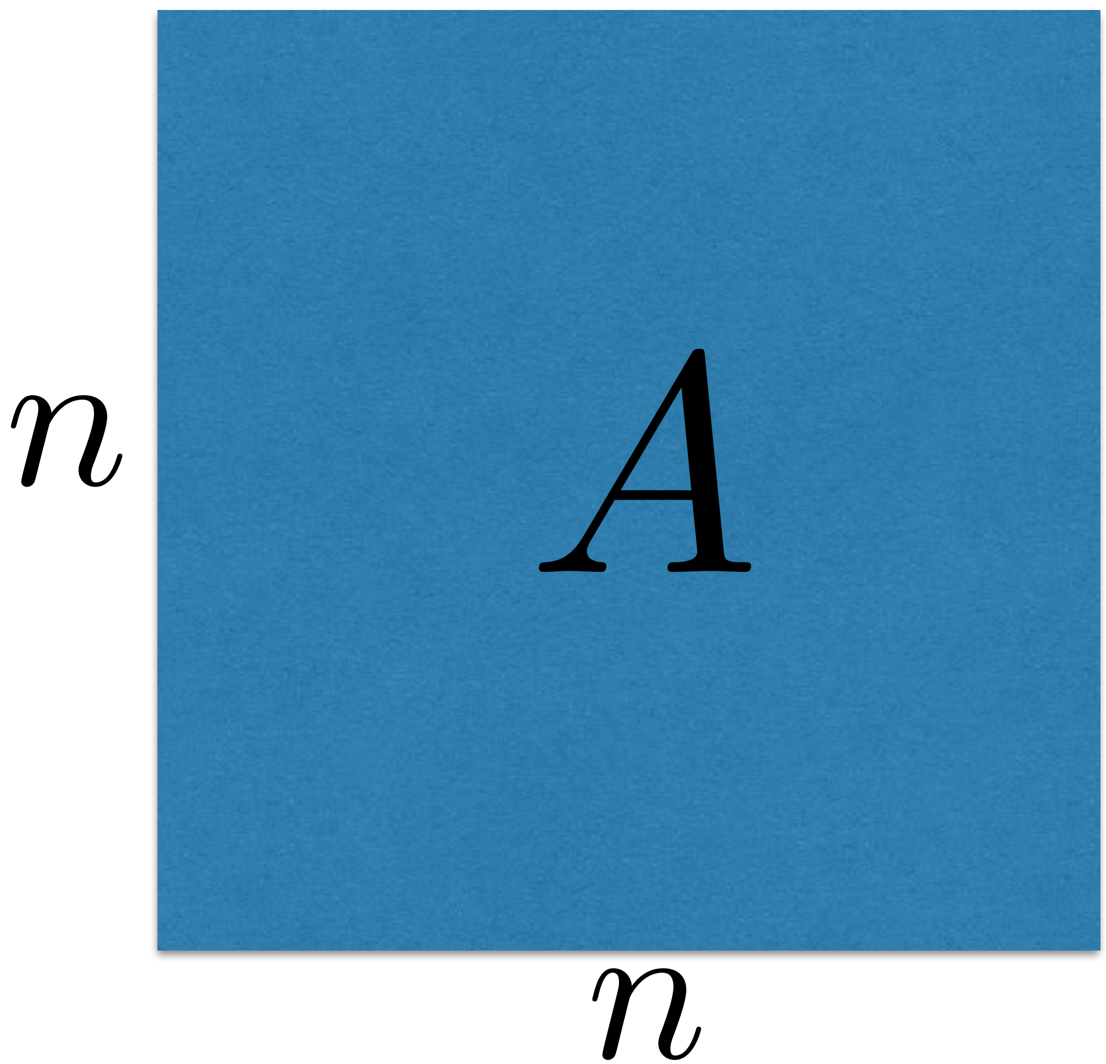
Spectral Clustering

$$A_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}$$

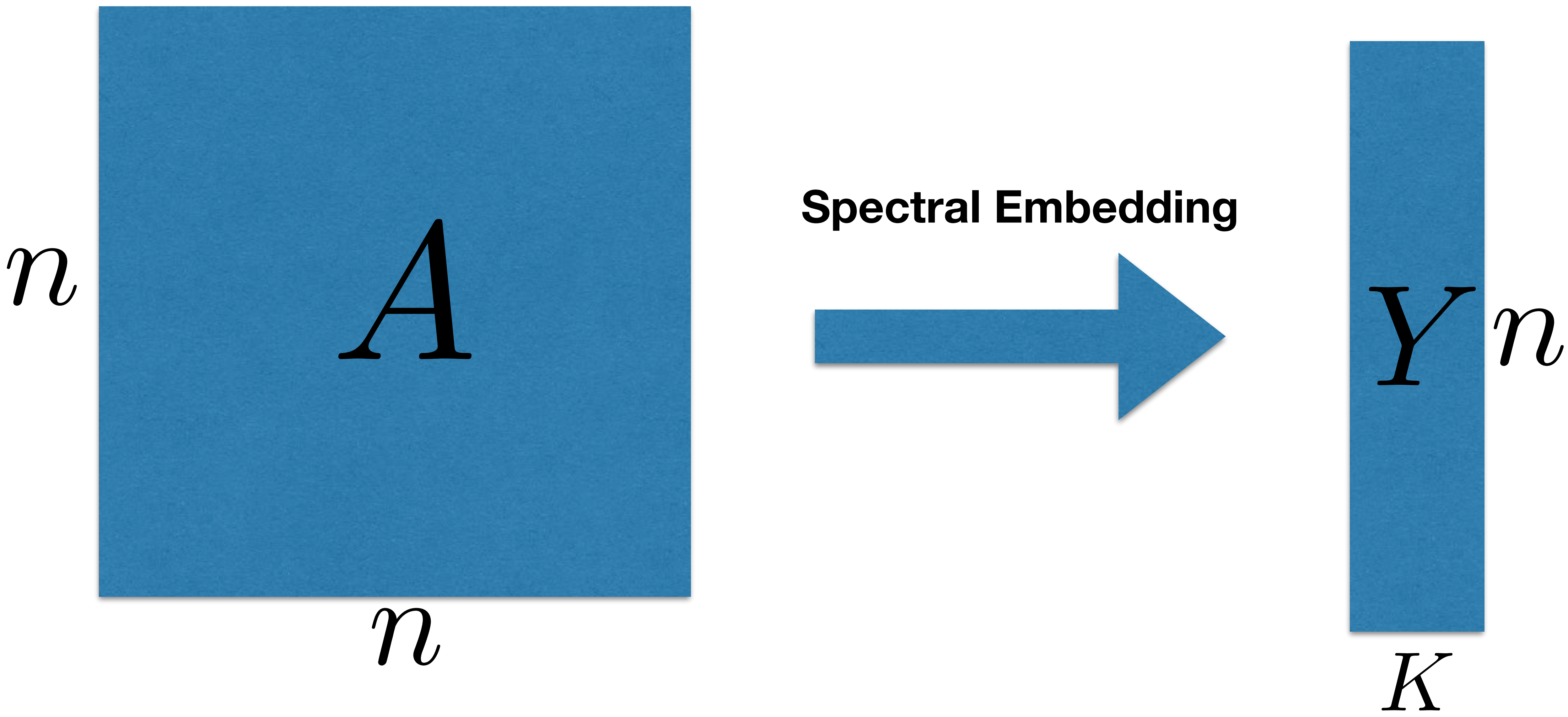


A is adjacency matrix of a graph

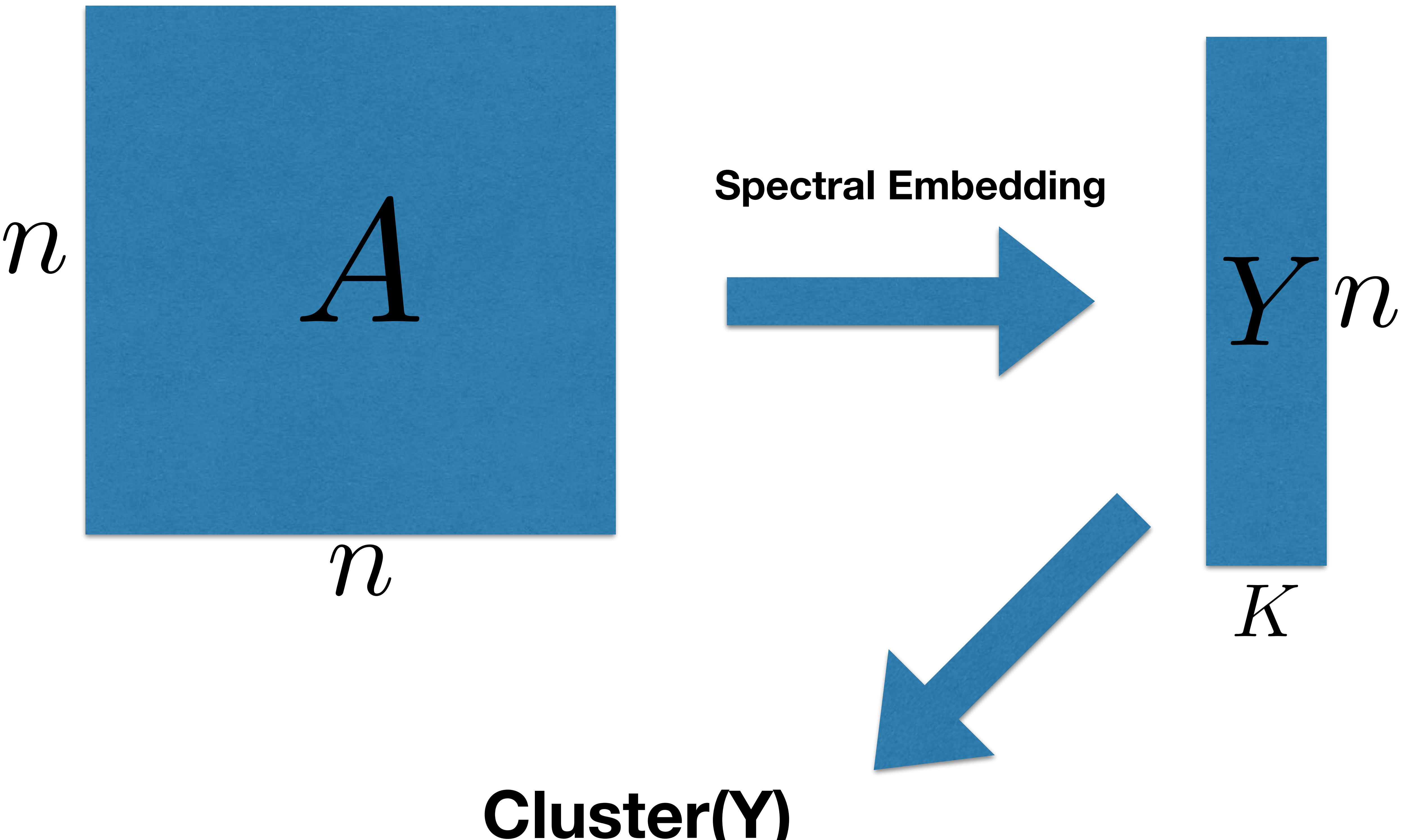
Steps



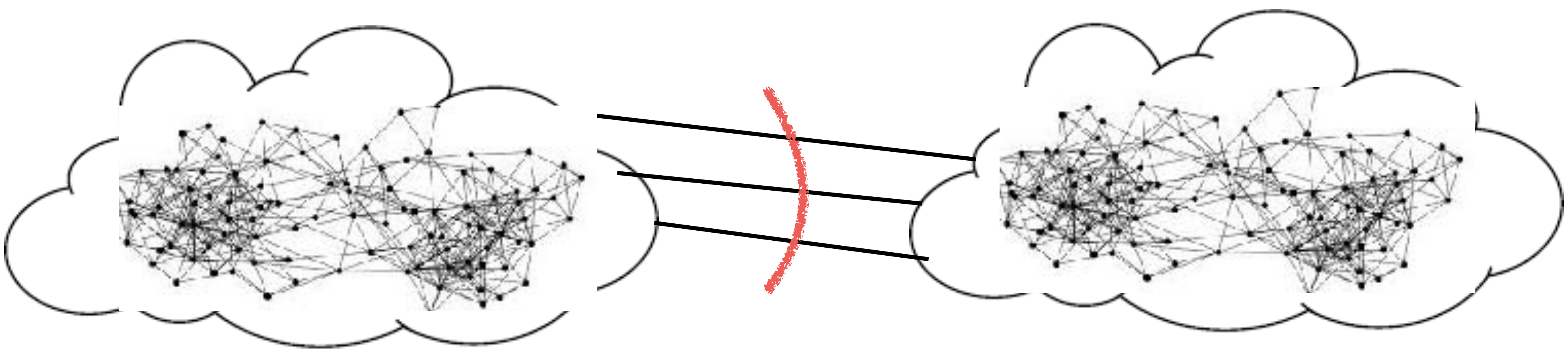
Steps



Steps



What is the Embedding?

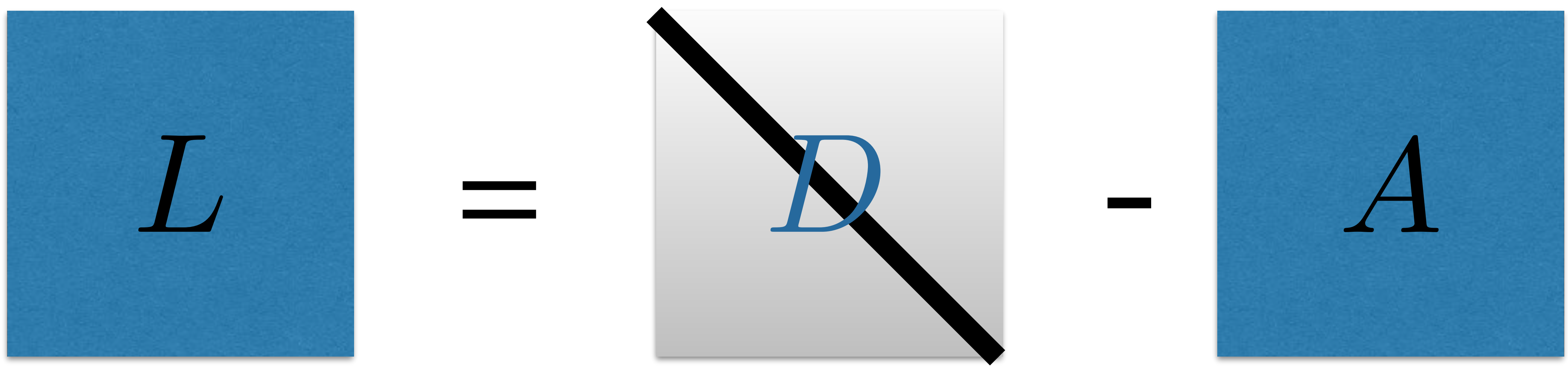


- Map each node in V to R^K
- Nodes linked to each other are close
- Disconnected groups of nodes are far from each other

Spectral Clustering

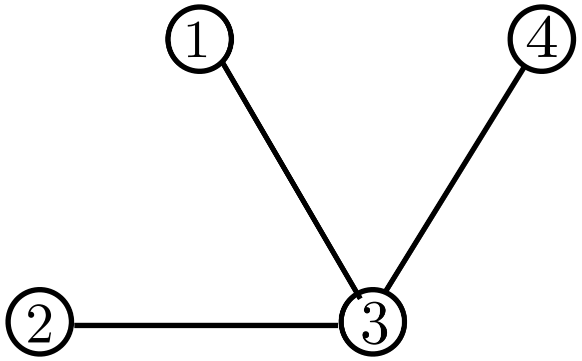
$$L = D - A$$

Spectral Clustering



$$D_{i,i} = \sum_{j=1}^{n} A_{i,j}$$

Graph Clustering



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

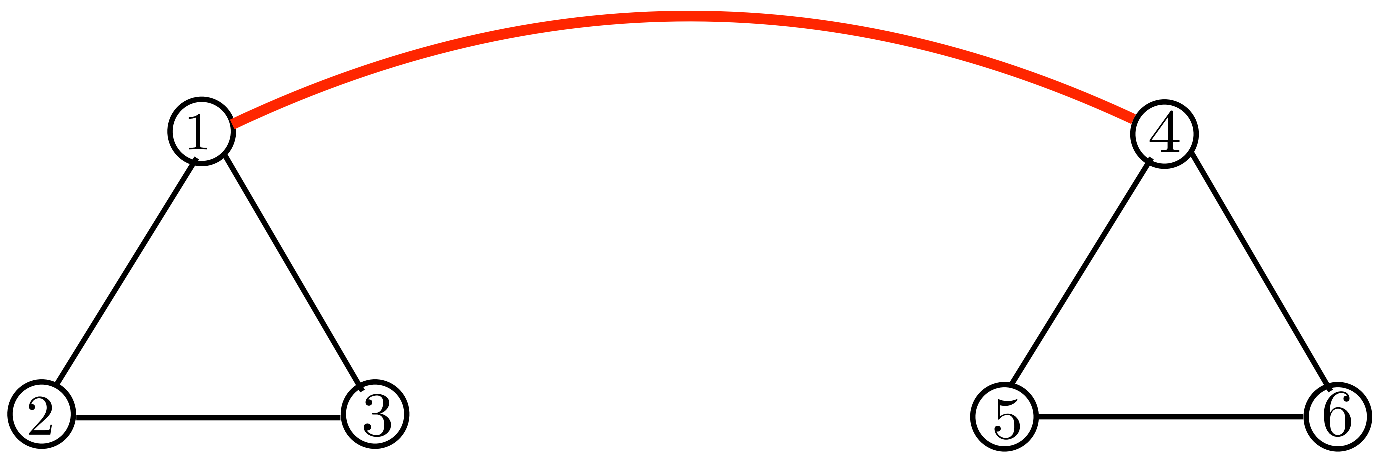
  Proof: Sum of each row of L is 0 because $D_{i,i} = \sum_{j=1}^{n} A_{i,j}$ and $L = D - A$

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.

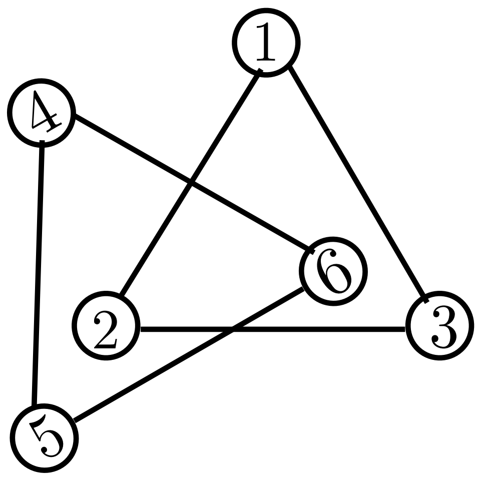
GRAPH CLUSTERING



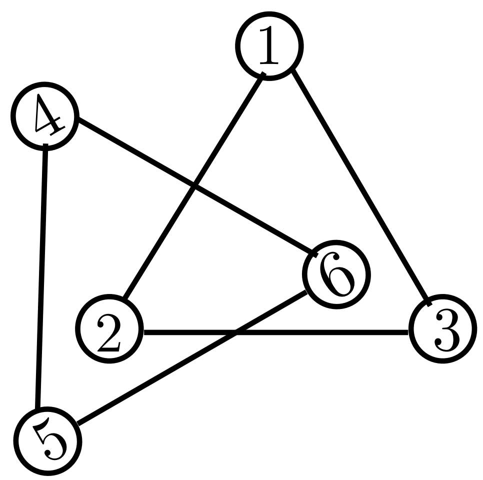
- Fact: For general graph, number of 0 eigenvalues correspond to number of connected components. The corresponding eigenvectors are all 1's on the nodes of connected components

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

Examples

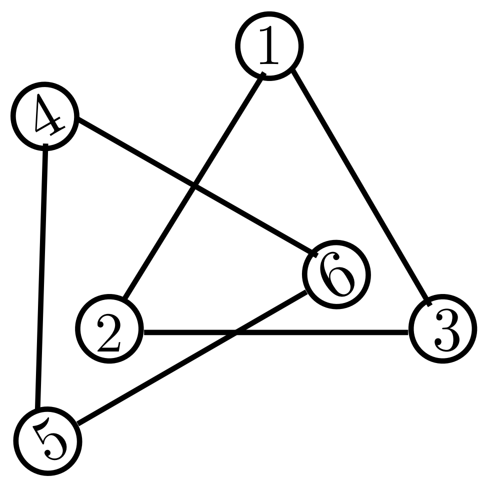


Examples



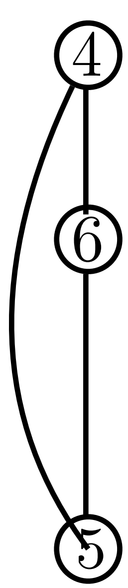
1D

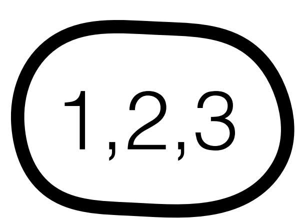
Examples



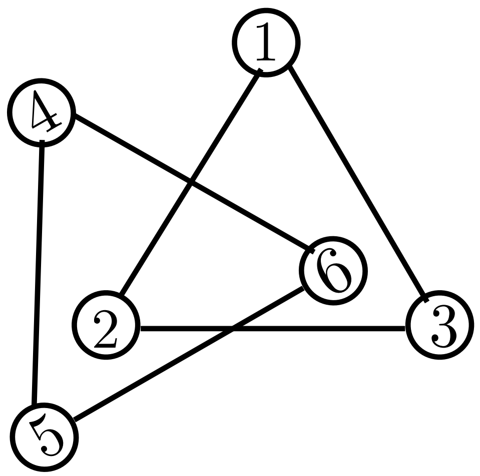
1D

2D





Examples

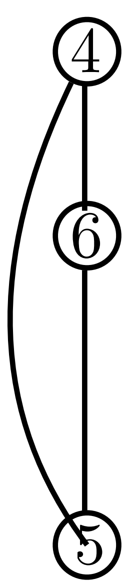


1D

1,2,3

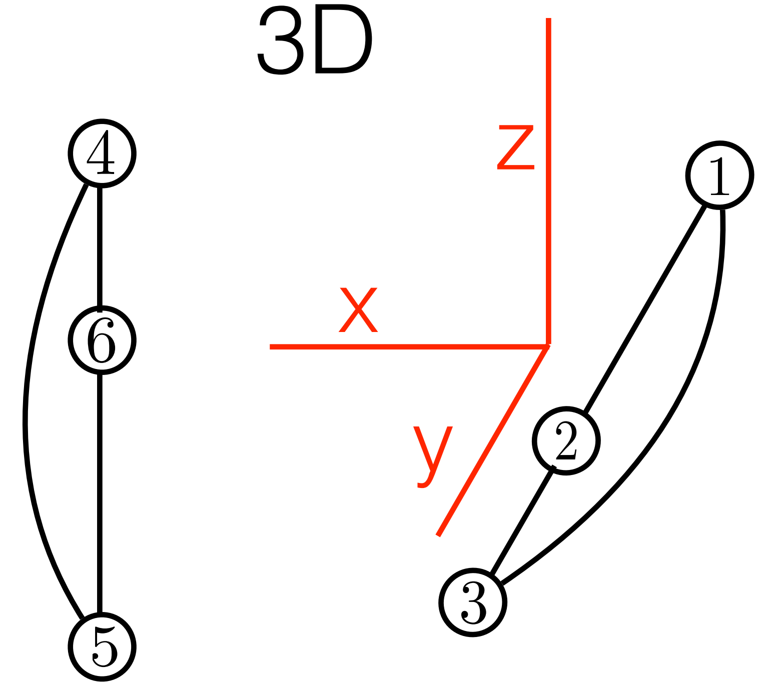
4,5,6

2D

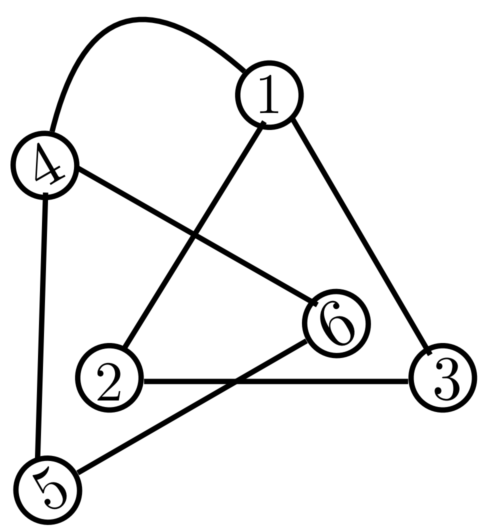


1,2,3

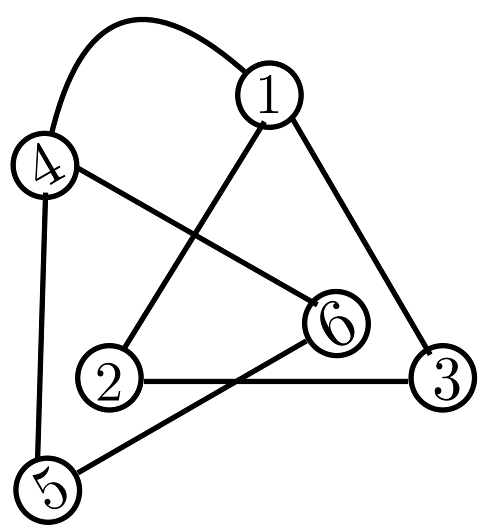
3D

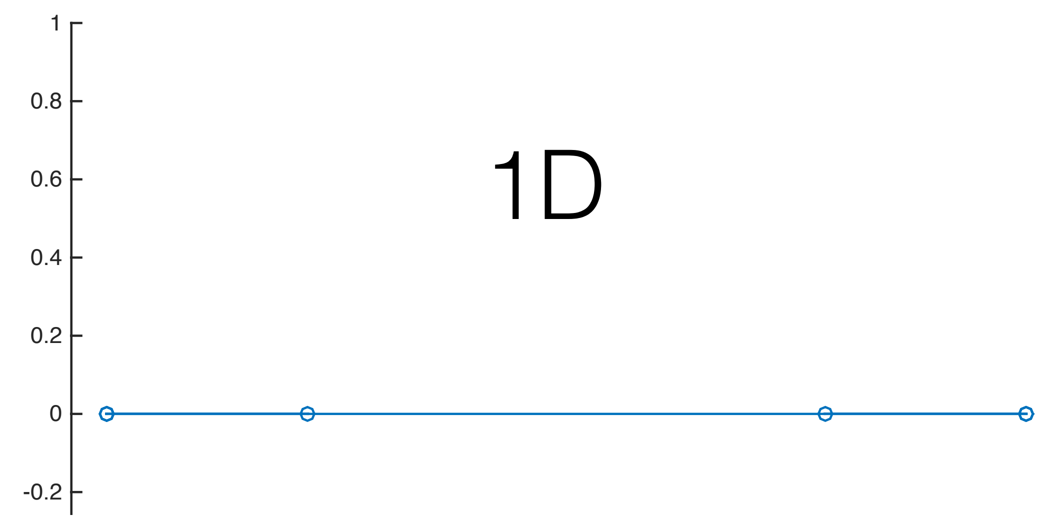


Examples

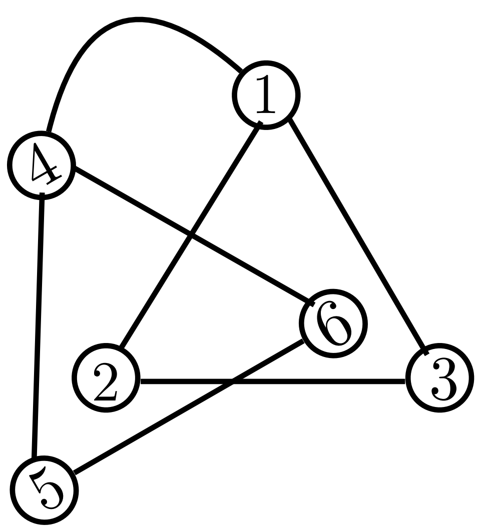


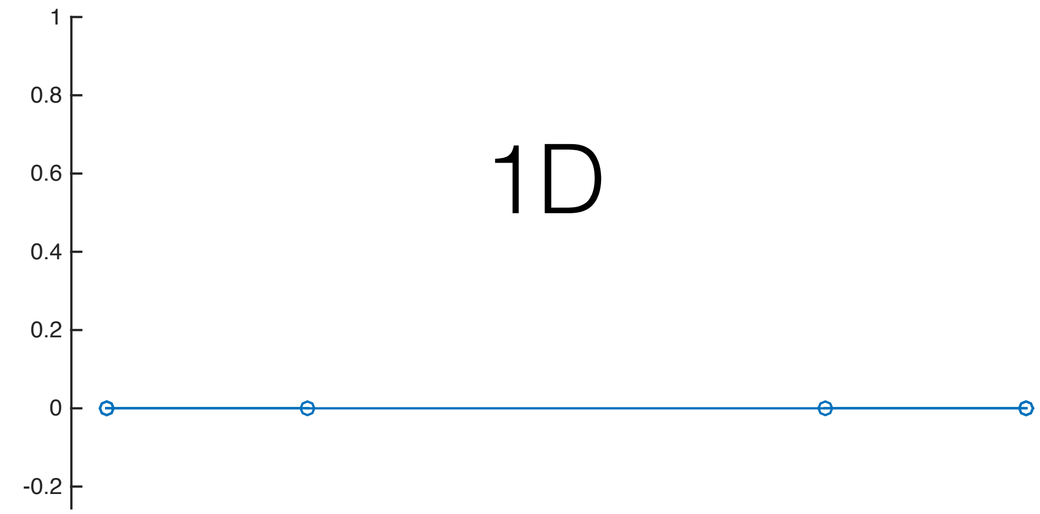
Examples

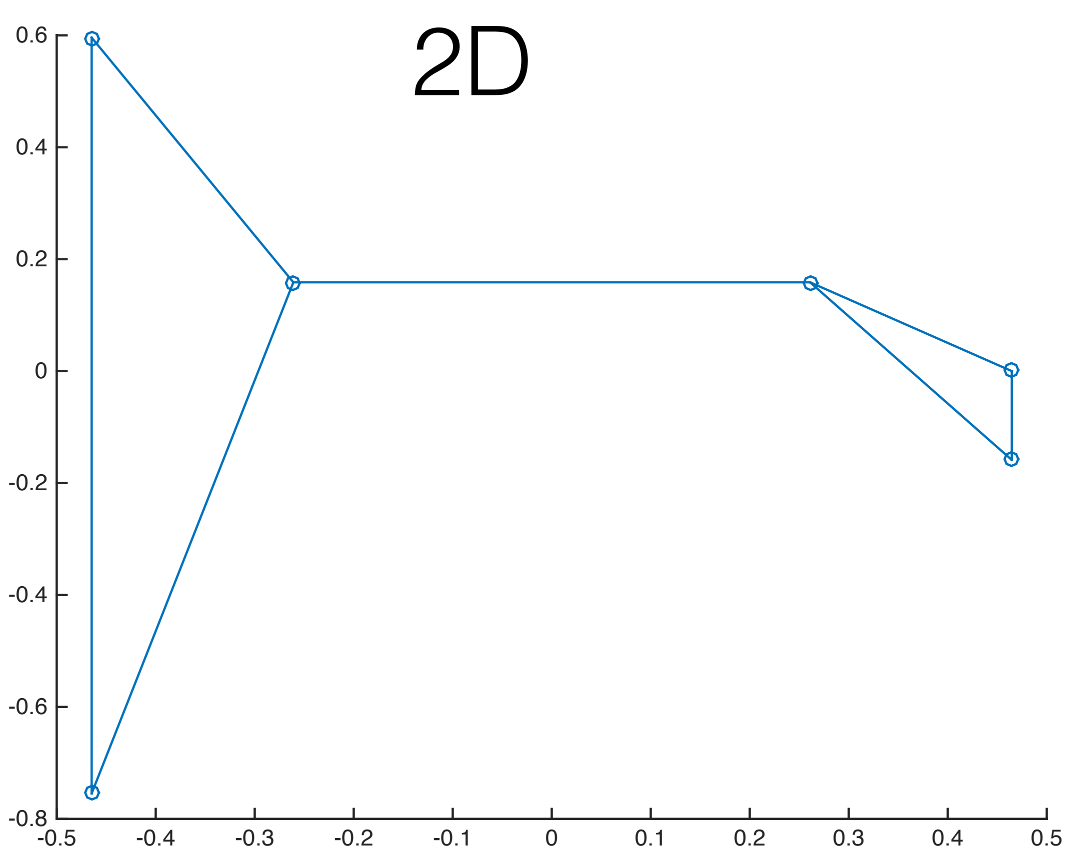




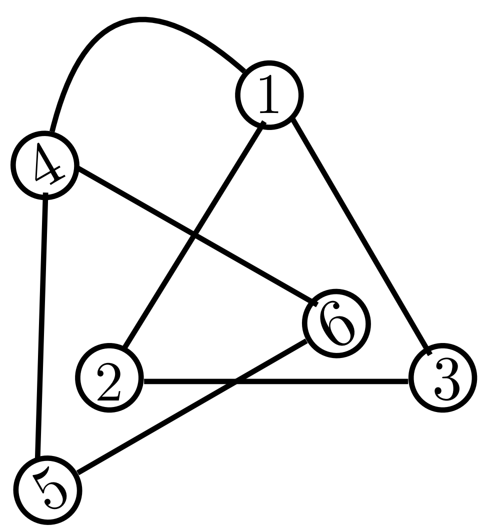
Examples

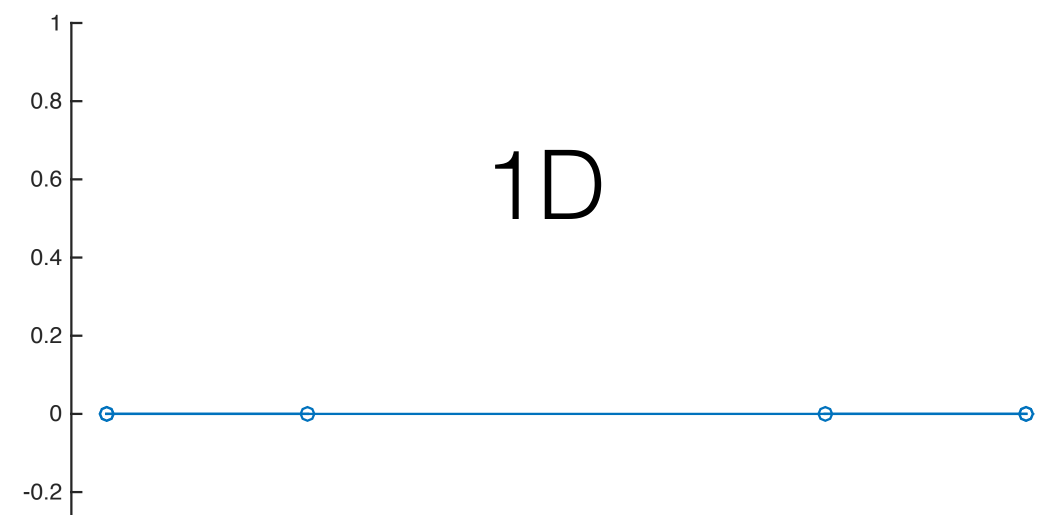

1
2
3
4
5
6

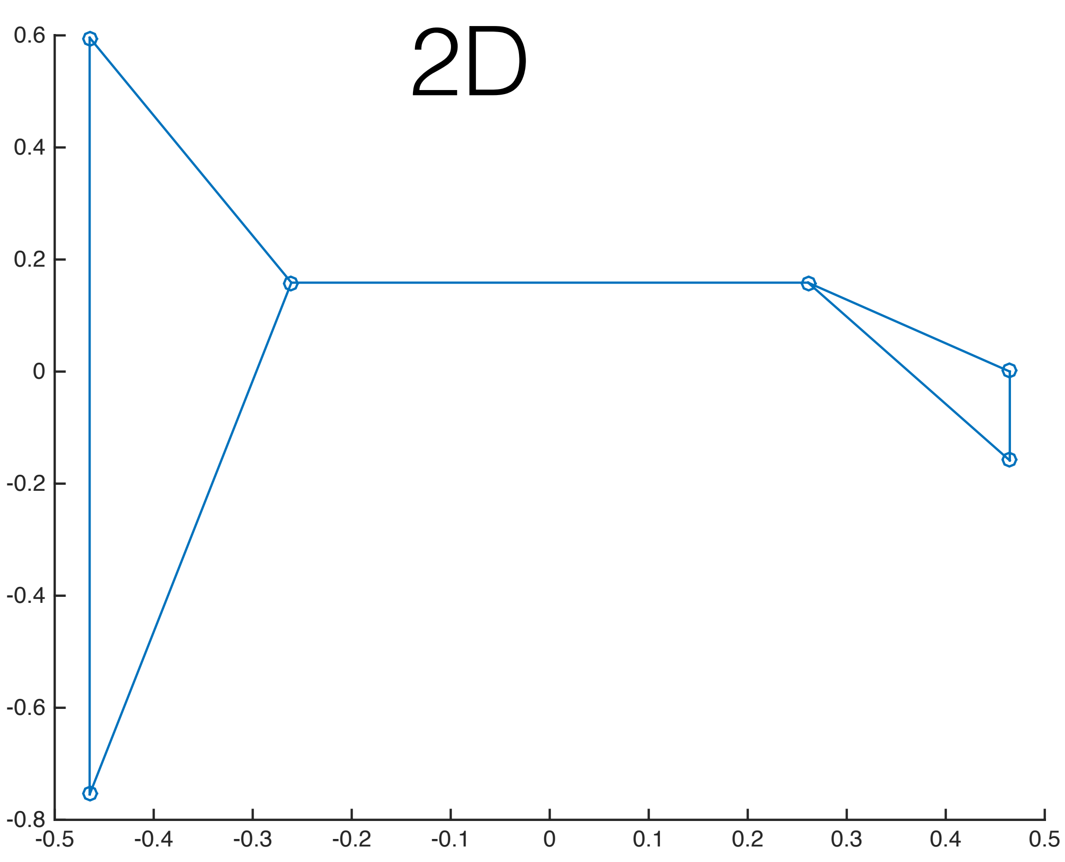

1D
1
0.8
0.6
0.4
0.2
0
-0.2

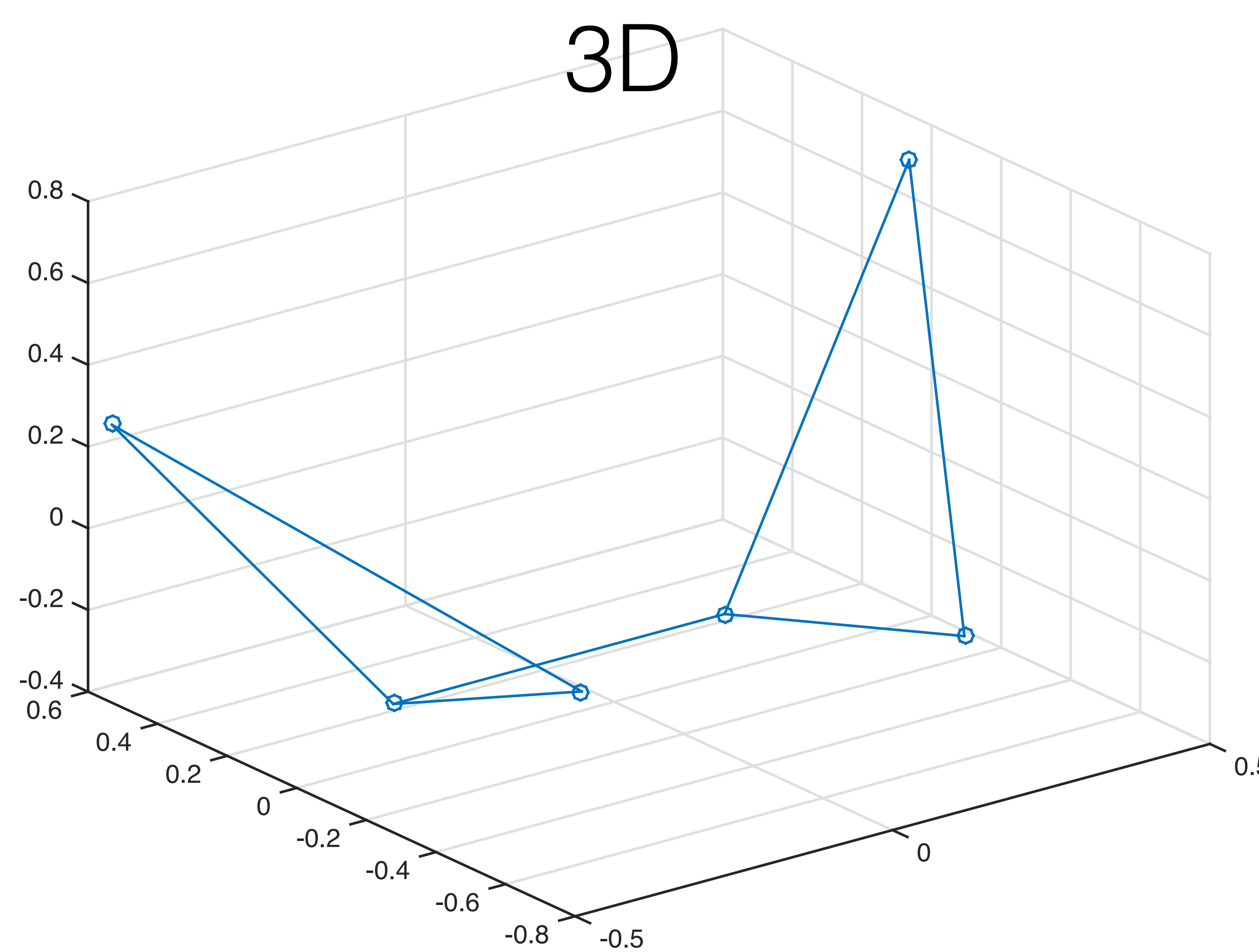

2D
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5

Examples


1
2
3
4
5
6


1D
1
0.8
0.6
0.4
0.2
0
-0.2


2D
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-0.5 -0.4 -0.3 -0.2 -0.1 0 0.1 0.2 0.3 0.4 0.5


3D
0.8
0.6
0.4
0.2
0
-0.2
-0.4
0.6 0.4 0.2 0 -0.2 -0.4 -0.6 -0.8
-0.5 0 0.5

More Examples

Spectral Embedding

- Nodes linked to each other are close in embedded space
- What has this got to do with Laplacian matrix?

Cuts and Laplacian

K = 1

$$\mathrm{Obj}(c) = \frac{1}{2} \sum_{(i,j) \in E} (c_i - c_j)^2$$

Cuts and Laplacian

K = 1

$$\mathrm{Obj}(c) = \frac{1}{2} \sum_{(i,j) \in E} (c_i - c_j)^2$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} (c_i - c_j)^2$$

Cuts and Laplacian

K = 1

$$\mathrm{Obj}(c) = \frac{1}{2} \sum_{(i,j) \in E} (c_i - c_j)^2$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} (c_i - c_j)^2$$

$$= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} (c_i^2 + c_j^2 - 2 c_i c_j)$$

Cuts and Laplacian

K = 1

$$\begin{aligned}
\mathrm{Obj}(c) &= \frac{1}{2} \sum_{(i,j)\in E} (c_i - c_j)^2 \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} (c_i - c_j)^2 \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} (c_i^2 + c_j^2 - 2 c_i c_j) \\
&= \frac{1}{2} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} A_{i,j} \right) c_i^2 + \frac{1}{2} \sum_{j=1}^{n} \left(\sum_{i=1}^{n} A_{i,j} \right) c_j^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} c_i c_j
\end{aligned}$$

Cuts and Laplacian

K = 1

$$
\begin{aligned}
\mathrm{Obj}(c) &= \frac{1}{2} \sum_{(i,j)\in E} (c_i - c_j)^2 \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} (c_i - c_j)^2 \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} (c_i^2 + c_j^2 - 2 c_i c_j) \\
&= \frac{1}{2} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} A_{i,j} \right) c_i^2 + \frac{1}{2} \sum_{j=1}^{n} \left(\sum_{i=1}^{n} A_{i,j} \right) c_j^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} c_i c_j \\
&= \frac{1}{2} \sum_{i=1}^{n} D_{i,i} c_i^2 + \frac{1}{2} \sum_{j=1}^{n} D_{j,j} c_j^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} c_i c_j
\end{aligned}
$$

Cuts and Laplacian

K = 1

$$
\begin{aligned}
\mathrm{Obj}(c) &= \frac{1}{2} \sum_{(i,j)\in E} (c_i - c_j)^2 \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j}(c_i - c_j)^2 \\
&= \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j}(c_i^2 + c_j^2 - 2c_i c_j) \\
&= \frac{1}{2} \sum_{i=1}^{n} \left(\sum_{j=1}^{n} A_{i,j} \right) c_i^2 + \frac{1}{2} \sum_{j=1}^{n} \left(\sum_{i=1}^{n} A_{i,j} \right) c_j^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} c_i c_j \\
&= \frac{1}{2} \sum_{i=1}^{n} D_{i,i} c_i^2 + \frac{1}{2} \sum_{j=1}^{n} D_{j,j} c_j^2 - \sum_{i=1}^{n} \sum_{j=1}^{n} A_{i,j} c_i c_j \\
&= c^\top D c - c^\top A c = c^\top L c
\end{aligned}
$$

Spectral Clustering, K = 1

Hence to find the solution we need to solve for

$$\text{Minimize } c^\top L c \quad \text{s.t.} \quad \|c\| = 1$$

Spectral Clustering, K = 1

Hence to find the solution we need to solve for

$$\text{Minimize } c^\top L c \quad \text{s.t.} \quad \|c\| = 1$$

Hence solution c to above is an Eigen vector, first smallest one is the all 1's vector (for connected graph), second smallest one is our solution

To get clustering assignment we simply threshold at 0

SPECTRAL CLUSTERING, K >1

- Solution obtained by considering the second smallest up to K^{th} smallest eigenvectors

$$\mathrm{Obj}(c) = \sum_{k=1}^{K} {c^k}^\top L c^k$$

c^k's are orthogonal to each other and the all ones vector

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 $\mathbf{y}_1, \ldots, \mathbf{y}_n$

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 $\mathbf{y}_1, \ldots, \mathbf{y}_n$

$\mathbf{y}_1, \ldots, \mathbf{y}_n$ are called spectral embedding

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 $\mathbf{y}_1, \ldots, \mathbf{y}_n$

$\mathbf{y}_1, \ldots, \mathbf{y}_n$ are called spectral embedding

Embeds the n nodes into K-1 dimensional vectors

- Unnormalized Spectral clustering aims to cluster based on minimizing cut

- Unnormalized Spectral clustering aims to cluster based on minimizing cut
    - cut: Number of edges that need to be deleted to have no links between the cluster and other nodes outside

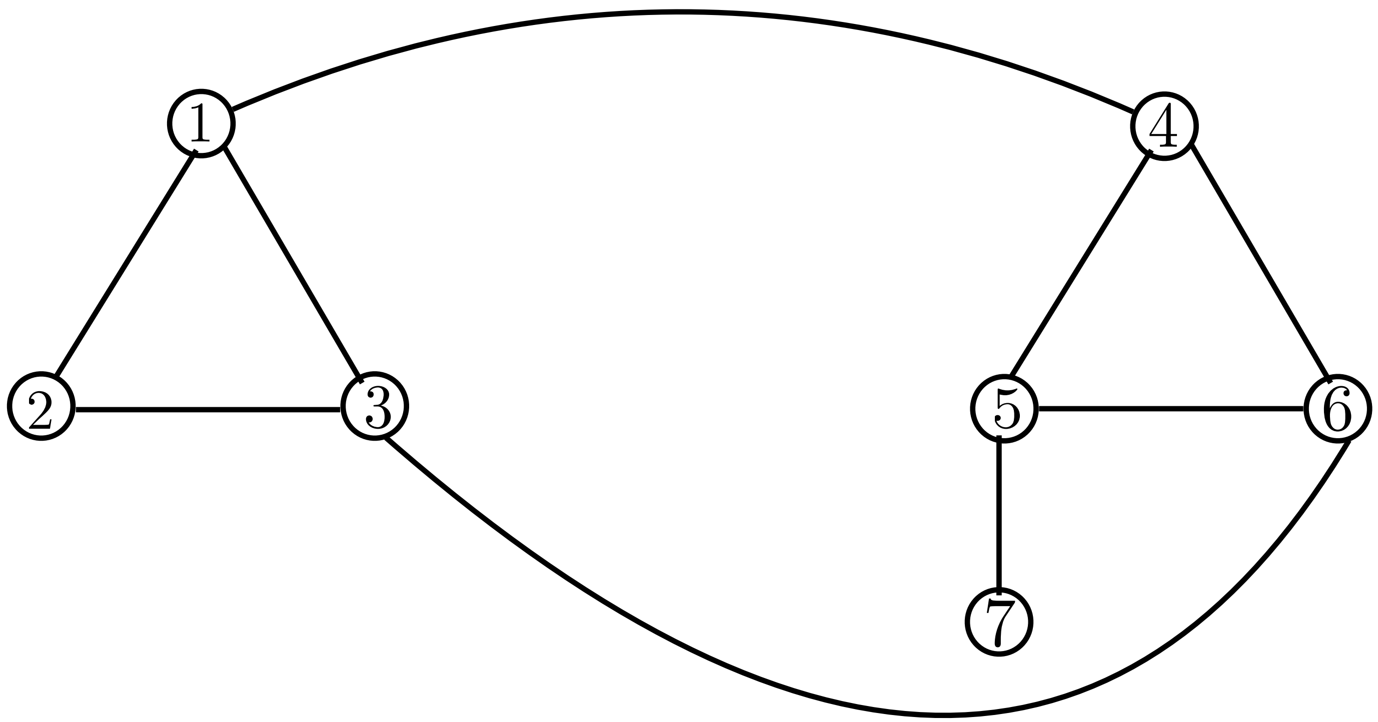
- Unnormalized Spectral clustering aims to cluster based on minimizing cut
  - cut: Number of edges that need to be deleted to have no links between the cluster and other nodes outside
  - But is cut the right metric?

Normalized Cut

- Why cut is perhaps not a good measure?

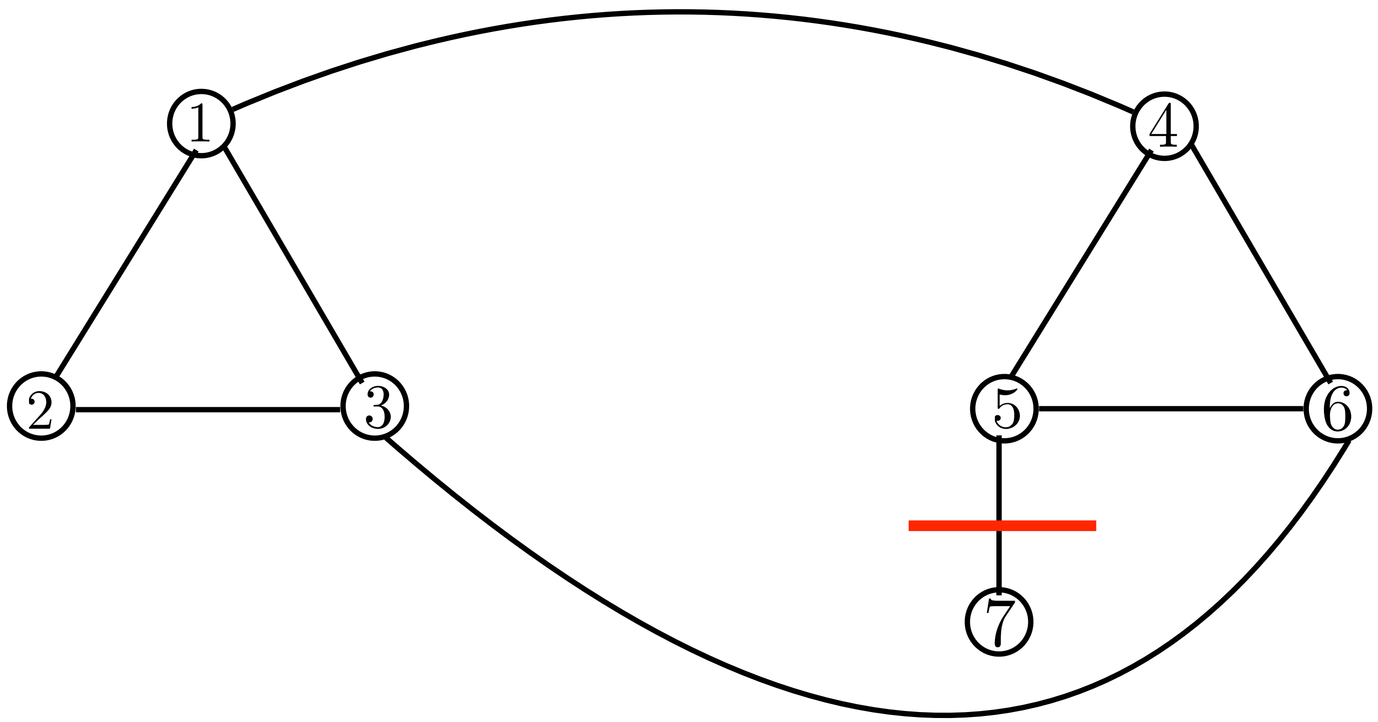
Normalized Cut

- Why cut is perhaps not a good measure?



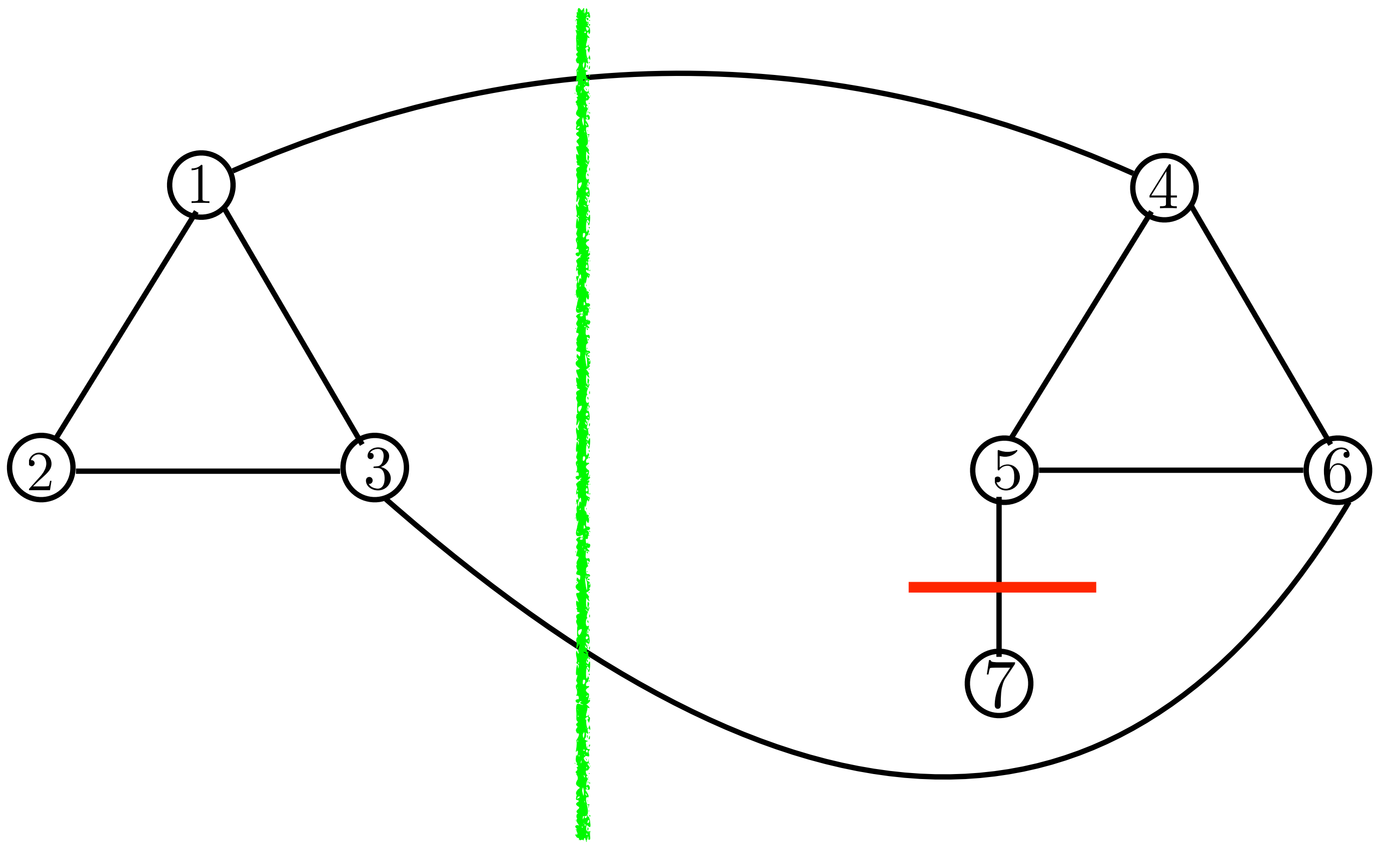
NORMALIZED CUT

- Why cut is perhaps not a good measure?



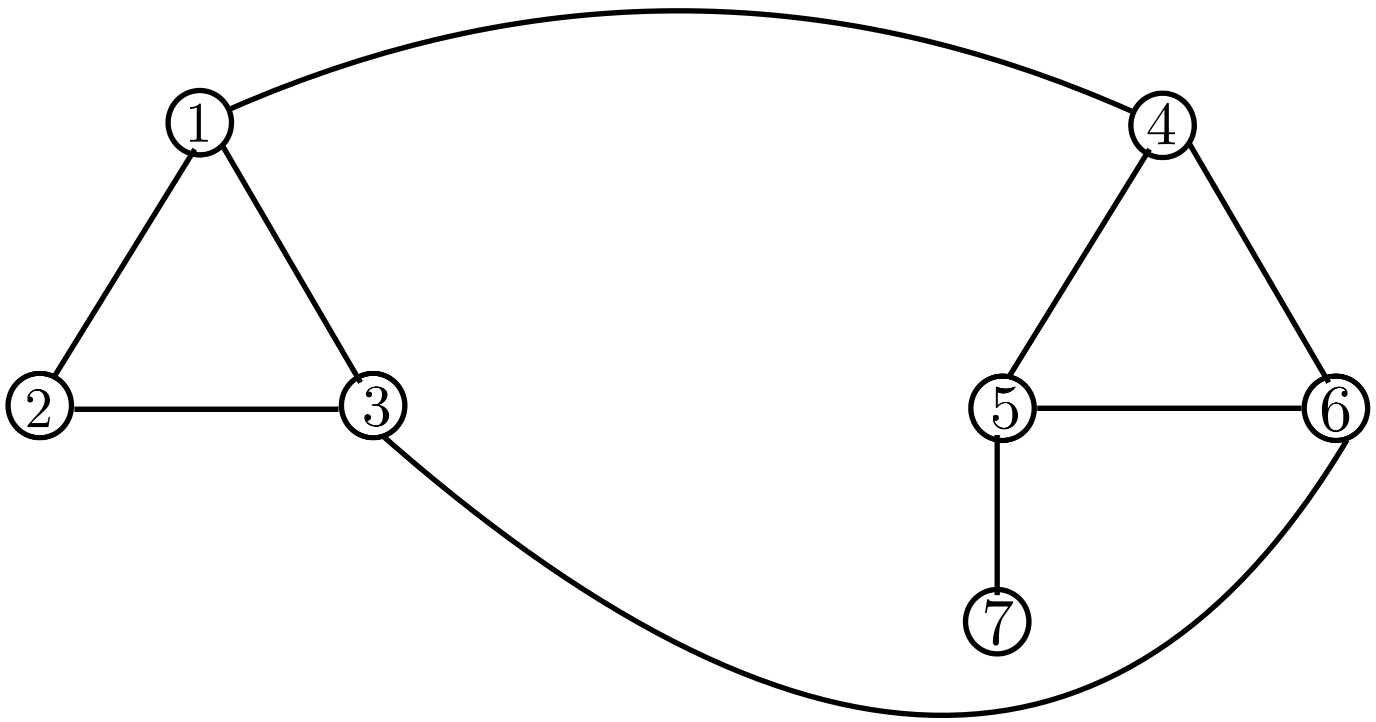
Normalized Cut

- Why cut is perhaps not a good measure?



Ratio Cut

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



Normalized Cut

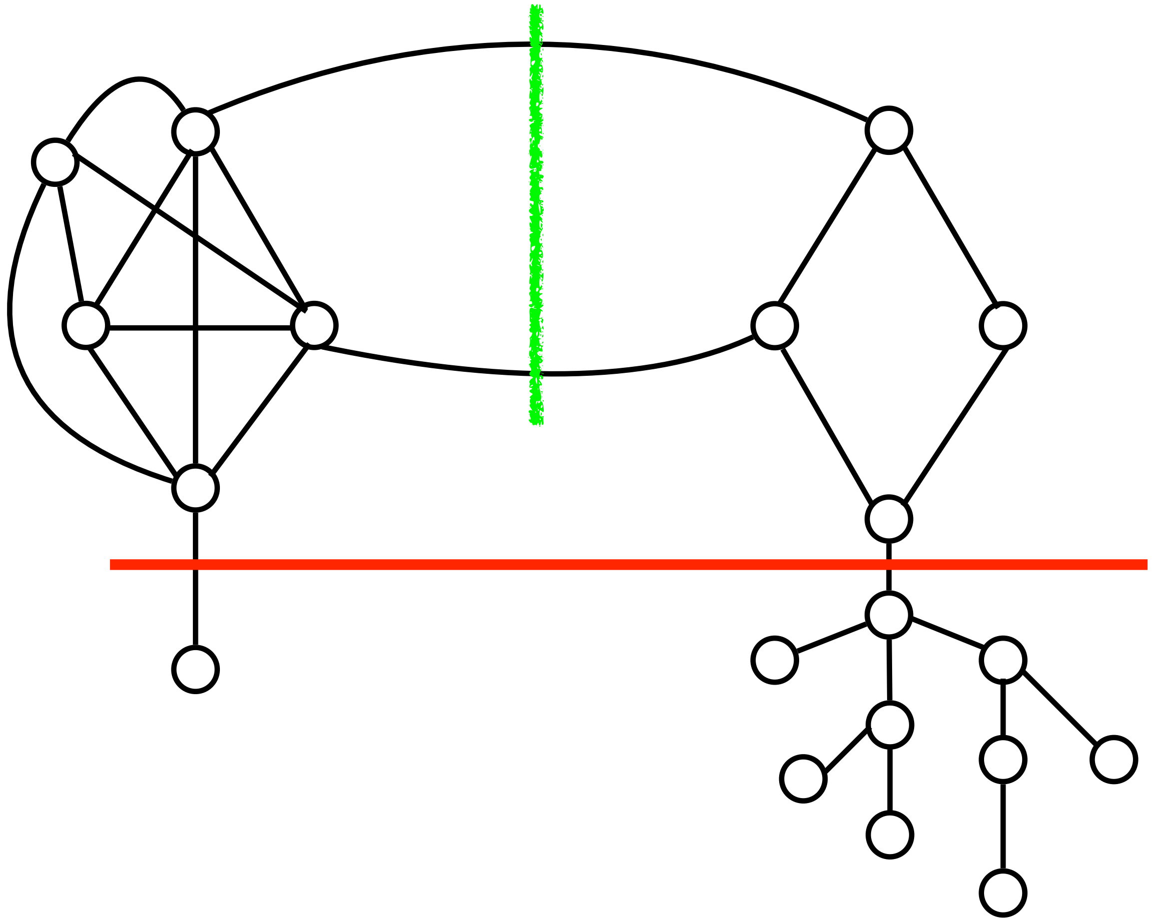
- Normalized cut: Minimize sum of ratio of number of edges cut per cluster and number of edges within cluster

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

Normalized Cut

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

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



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 $K = 2$

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

NORMALIZED CUT

- 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 $K = 2$

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

- This is an NP hard problem! …so relax

Normalized Spectral Clustering

- As before, we want to minimize $\sum_{(i,j)\in E}(c_i - c_j)^2 = c^\top L c$

Normalized Spectral Clustering

- As before, we want to minimize $\sum_{(i,j)\in E}(c_i - c_j)^2 = c^\top L c$
- But we also want to weight the values of c_i's based on degree. We want high degree nodes to have larger c magnitude

Normalized Spectral Clustering

- As before, we want to minimize $\sum_{(i,j)\in E}(c_i - c_j)^2 = c^\top Lc$
- But we also want to weight the values of c_i's based on degree. We want high degree nodes to have larger c magnitude
- That is we want to simultaneously maximize $\sum_{i=1}^{n} c_i^2 D_{i,i}^2 = c^\top Dc$

Normalized Spectral Clustering

- As before, we want to minimize $\sum_{(i,j)\in E}(c_i - c_j)^2 = c^\top L c$
- But we also want to weight the values of c_i's based on degree. We want high degree nodes to have larger c magnitude
- That is we want to simultaneously maximize $\sum_{i=1}^n c_i^2 D_{i,i}^2 = c^\top D c$
- Find c so as to:

$$\text{minimize } \frac{c^\top L c}{c^\top D c}$$

$$\equiv \text{minimize } c^\top L c \text{ subject to } c^\top D c = 1$$

Normalized Spectral Clustering

- As before, we want to minimize $\sum_{(i,j)\in E}(c_i - c_j)^2 = c^\top L c$
- But we also want to weight the values of c_i's based on degree. We want high degree nodes to have larger c magnitude
- That is we want to simultaneously maximize $\sum_{i=1}^{n} c_i^2 D_{i,i}^2 = c^\top D c$
- Find c so as to:

$$\text{minimize} \quad \frac{c^\top L c}{c^\top D c}$$

$$\equiv \text{minimize} \quad c^\top L c \text{ subject to } c^\top D c = 1$$

$$\equiv \text{minimize} \quad u^\top D^{-1/2} L D^{-1/2} u \text{ subject to } \|u\| = 1$$

$$\text{Minimize } c^\top \tilde{L} c \text{ s.t. } c \perp \mathbf{1}$$

$$\text{Minimize } c^\top \tilde{L} c \text{ s.t. } c \perp \mathbf{1}$$

Approximately Minimize normalized cut!

Spectral Clustering

$$\text{Minimize } c^\top \tilde{L} c \text{ s.t. } c \perp \mathbf{1}$$

Approximately Minimize normalized cut!

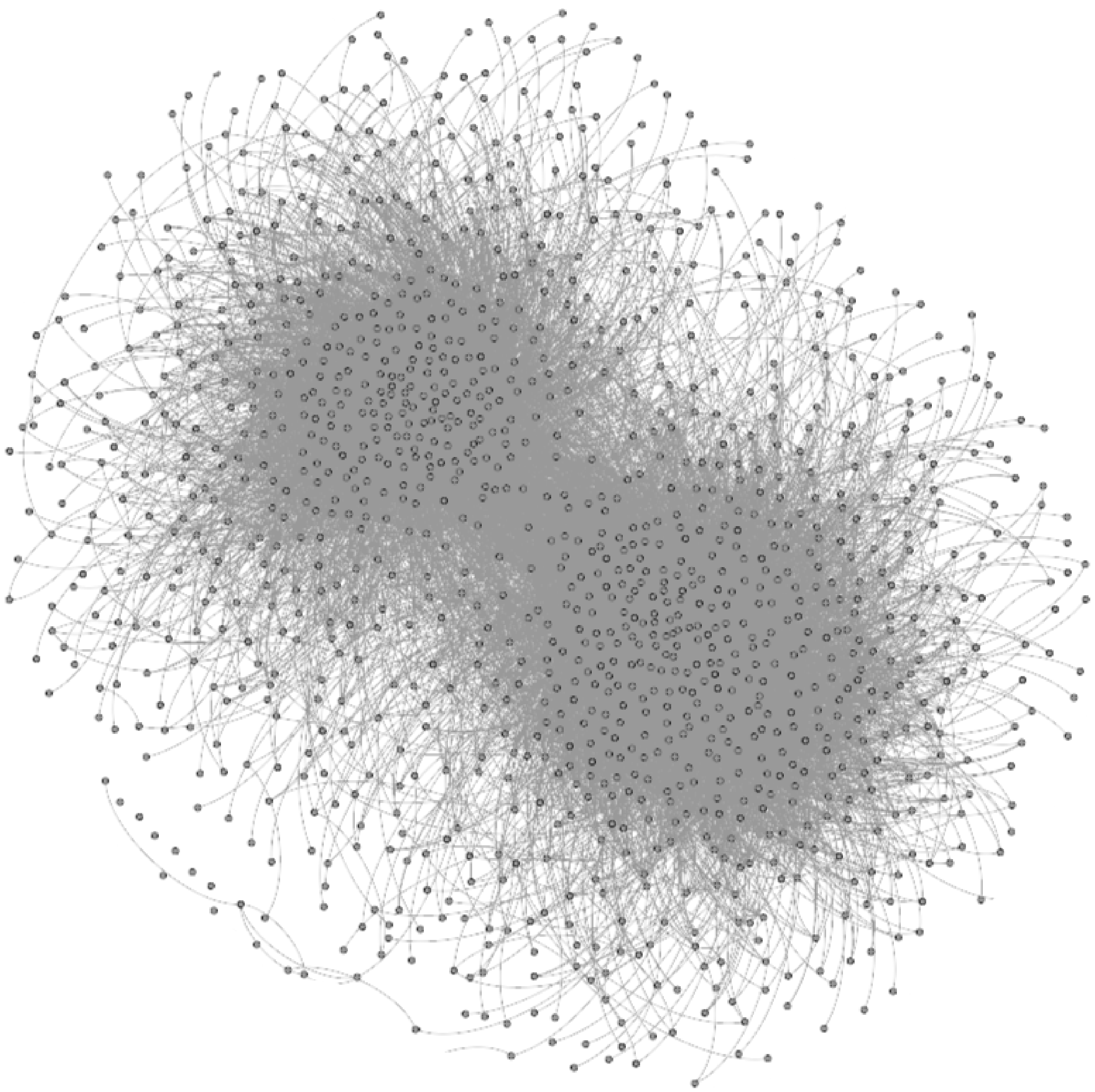
- Solution: Find second smallest eigenvectors of $\tilde{L} = I - D^{-1/2} A D^{-1/2}$

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 $\mathbf{y}_1, \ldots, \mathbf{y}_n$

Demo

Spectral Clustering



Spectral Clustering