Machine Learning

Lecture 15: Clustering

杨思蓓

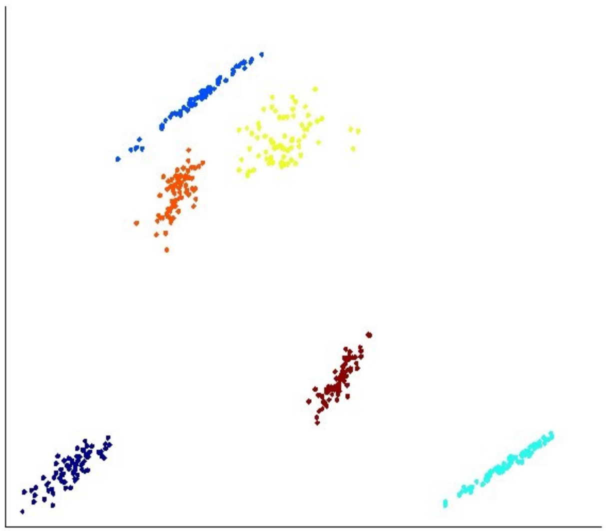
SIST

Email: yangsb@shanghaitech.edu.cn

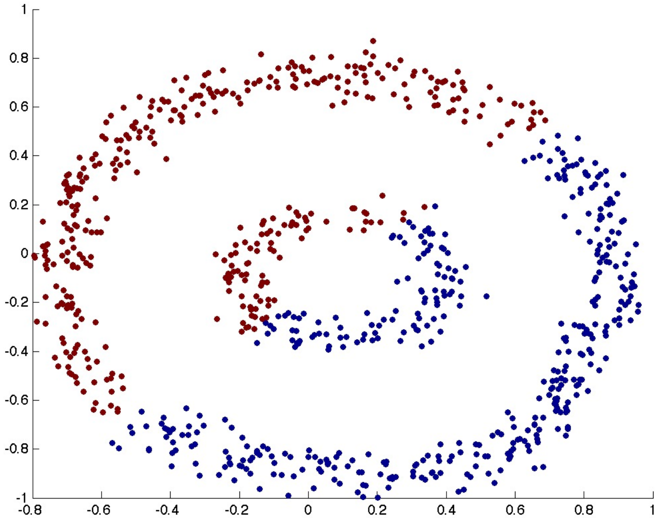
Algorithms

- Partitioning approach:
  - Construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors
  - Typical methods: **k-means**, **k-medoids**
- Model-based:
  - A model is hypothesized for each of the clusters and tries to find the best fit of that model to each other
  - Typical methods: **GMM**
- Dimensionality reduction approach
  - First dimensionality reduction, then clustering
  - Typical methods: **Spectral clustering**, Ncut

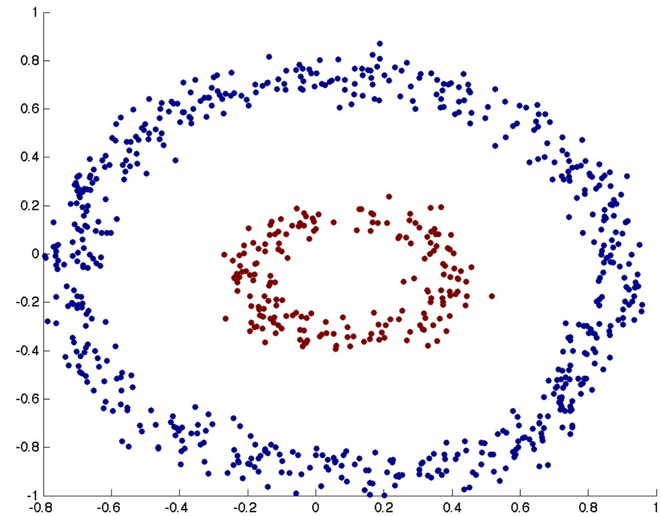
Good clustering – we know it when we see it



An Example

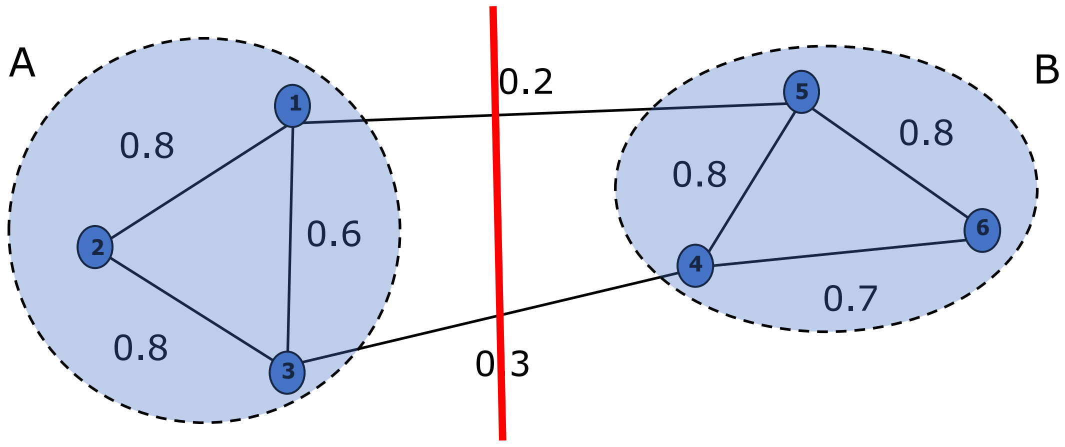


1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1



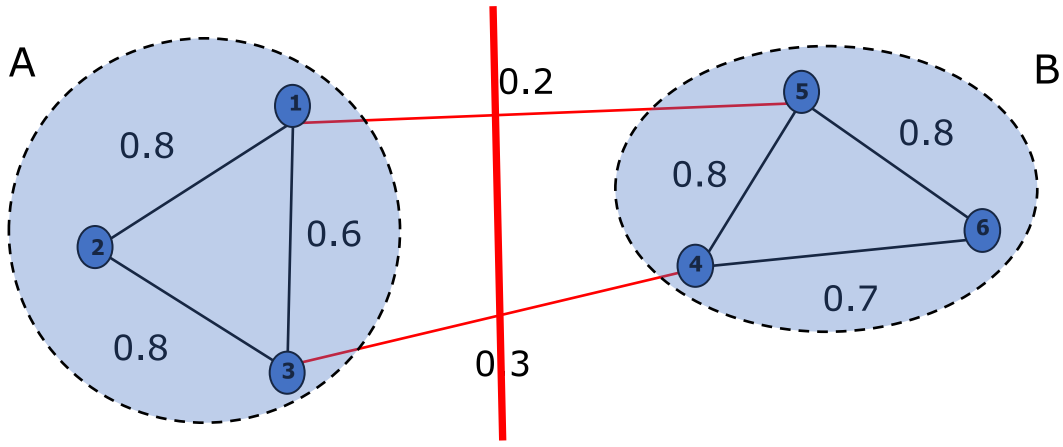
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

Spectral Clustering



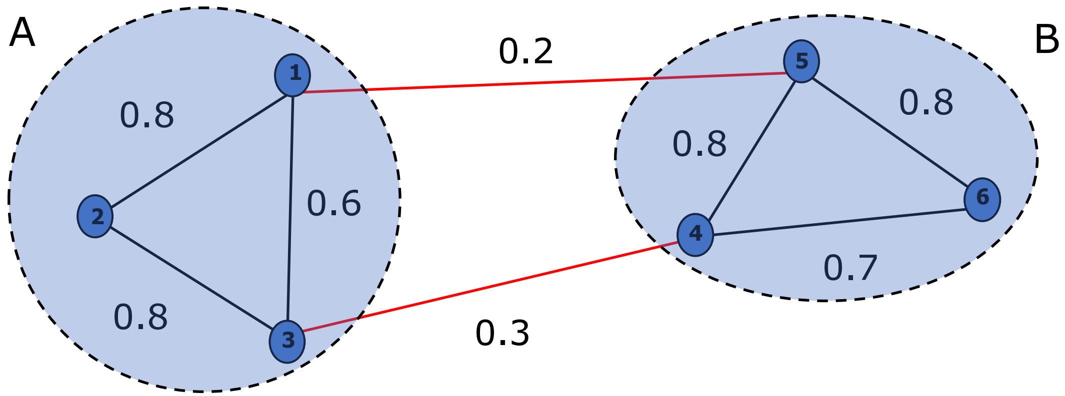
- Represent data points as the vertices V of a graph G.
  - All pairs of vertices are connected by an edge E.
  - Edges have weights W. Large weights mean that the adjacent vertices are very similar; small weights imply dissimilarity.
- Clustering can be viewed as partitioning a similarity graph
  - Divide vertices into two disjoint groups (A,B)

Clustering Objectives



- Traditional definition of a “good” clustering:
  - Points assigned to same cluster should be highly similar.
  - Points assigned to different clusters should be highly dissimilar.
- Apply these objectives to our graph representation
  - Minimize weight of between-group connections

Graph Cuts



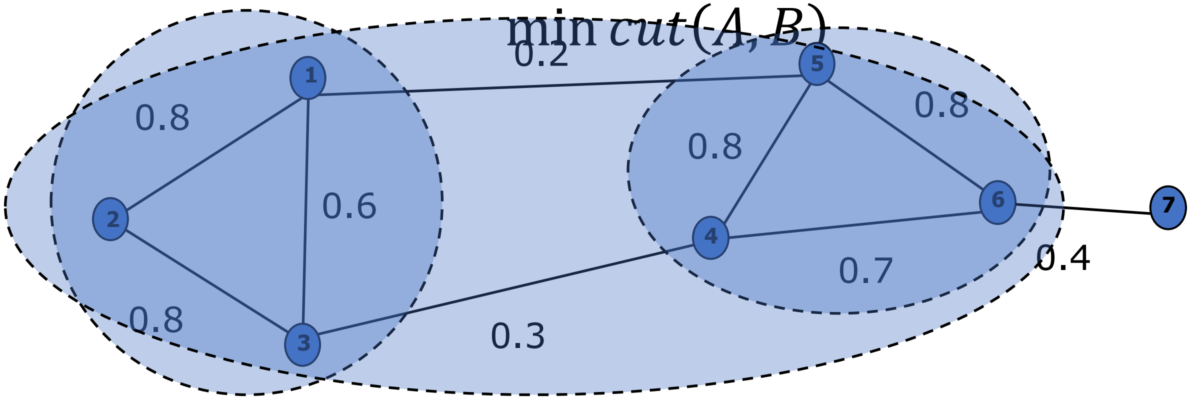
- Express partitioning objectives as a function of the “edge cut” of the partition.
  - ***Cut***: Set of edges with only one vertex in a group.we wants to find the minimal cut beetween groups. The groups that has the minimal cut would be the partition

$$cut(A,B) = \sum_{i \in A, j \in B} w_{ij}$$

Graph Cut Criteria

- Criterion: Minimum-cut
  - Minimize the weights of connections between groups

$$\min cut(A, B)$$



- Problem:
  - Only considers the inter-cluster connections
  - Does not consider the intra-cluster density

► Maximize the weights of connections within groups

$$\max\big(assoc(A, A) + assoc(B, B)\big)$$

- $assoc(A, A) = \sum_{i \in A, j \in A} w_{ij}$

Graph Cut Criteria

- Criterion: Normalized-cut (Shi & Malik,'97)
  - Consider the connectivity between groups relative to the density of each group.

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

    - Normalize the association between groups.

$$assoc(A,V) = \sum_{i \in A, j \in V} w_{ij}$$

- Produces more balanced partitions

$$\min Ncut(A,B)$$

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}$$

Graph Cut Criteria

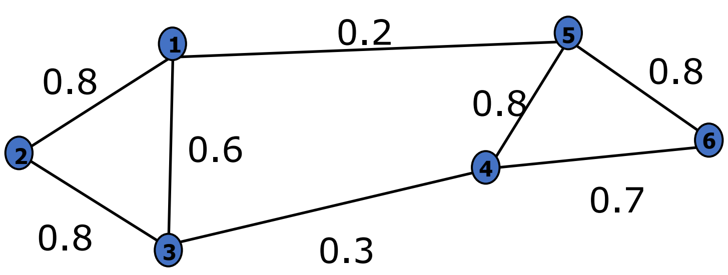
$$cut(A,B) = assoc(A,V) - assoc(A,A)$$

$$cut(A,B) = assoc(B,V) - assoc(B,B)$$

- $Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$
- $= \frac{assoc(A,V) - assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,V) - assoc(B,B)}{assoc(B,V)}$
- $= 2 - \left(\frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}\right) = 2 - Nassoc(A,B)$

Matrix Representation

- Adjacency matrix (W)
  - $n \times n$ matrix
  - w_{ij}: edge weight between vertex x_i and x_j
  - Symmetric matrix



	x_1	x_2	x_3	x_4	x_5	x_6
x_1	0	0.8	0.6	0	0.2	0
x_2	0.8	0	0.8	0	0	0
x_3	0.6	0.8	0	0.3	0	0
x_4	0	0	0.3	0	0.8	0.7
x_5	0.2	0	0	0.8	0	0.8
x_6	0	0	0	0.7	0.8	0

Objective Function of Ncut

$$\boldsymbol{x} \in [1,-1]^n, x_i = \begin{cases} 1 & i \in A \\ -1 & i \in B \end{cases} \qquad d_i = \sum_j w_{ij}$$

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)}$$

$$= \frac{\sum_{x_i>0,x_j<0} -w_{ij}x_i x_j}{\sum_{x_i>0} d_i} + \frac{\sum_{x_i<0,x_j>0} -w_{ij}x_i x_j}{\sum_{x_i<0} d_i}$$

$$W \in R^{n\times n} \qquad D \in R^{n\times n} \qquad \boldsymbol{x} \in [1,-1]^n \qquad \mathbf{1} \in [1]^n \qquad k = \frac{\sum_{x_i>0} d_i}{\sum_i d_i}$$

$$4\ Ncut(A,B) = \frac{(\mathbf{1}+\boldsymbol{x})^T(D-W)(\mathbf{1}+\boldsymbol{x})}{k\mathbf{1}^T D\mathbf{1}} + \frac{(\mathbf{1}-\boldsymbol{x})^T(D-W)(\mathbf{1}-\boldsymbol{x})}{(1-k)\mathbf{1}^T D\mathbf{1}}$$

$$b = \frac{k}{1-k}$$

$$= \frac{[(\mathbf{1}+\boldsymbol{x}) - b(1-\boldsymbol{x})]^T(D-W)[(\mathbf{1}+\boldsymbol{x}) - b(1-\boldsymbol{x})]}{b\mathbf{1}^T D\mathbf{1}}$$

Objective Function of Ncut

$$\boldsymbol{y} = (\mathbf{1} + \boldsymbol{x}) - b(1 - \boldsymbol{x}) \qquad k = \frac{\sum_{x_i>0} d_i}{\sum_i d_i} \qquad b = \frac{k}{1-k} = \frac{\sum_{x_i>0} d_i}{\sum_{x_i<0} d_i}$$

$$\boldsymbol{y}^T D \mathbf{1} = 2\sum_{x_i>0} d_i - 2b \sum_{x_i<0} d_i = 0$$

$$\boldsymbol{y}^T D \boldsymbol{y} = 4\sum_{x_i>0} d_i + 4b^2 \sum_{x_i<0} d_i = 4\left(b\sum_{x_i<0} d_i + b^2 \sum_{x_i<0} d_i\right)$$

$$= 4b\left(\sum_{x_i<0} d_i + b\sum_{x_i<0} d_i\right) = 4b\mathbf{1}^T D \mathbf{1}$$

$$\min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = \min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T (D-W)\boldsymbol{y}}{\boldsymbol{y}^T D \boldsymbol{y}}$$

$$s.t. \quad \boldsymbol{y} \in [2-2b, -2b]^n, \quad \boldsymbol{y}^T D \mathbf{1} = 0$$

- NP-hard!

Rayleigh quotient

- Relaxation:

$$\min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(D-W)\boldsymbol{y}}{\boldsymbol{y}^T D\boldsymbol{y}}, \boldsymbol{y} \in \mathcal{R}^n, \boldsymbol{y}^T D\mathbf{1} = 0$$

- $L \equiv D - W$

$$\min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T L\boldsymbol{y}}{\boldsymbol{y}^T D\boldsymbol{y}}, \boldsymbol{y} \in \mathcal{R}^n, \boldsymbol{y}^T D\mathbf{1} = 0$$

Rayleigh quotient

$$\max_x \frac{x^T A x}{x^T B x} \Rightarrow \max_x \; x^T A x \quad s.t. \; x^T B x = 1$$

Lagrangian Function

$$L(x) = \; x^T A x + \lambda(x^T B x - 1)$$

Taking the derivative with respect to x

$$\frac{\partial L(x)}{\partial x} = 0 \Rightarrow (A + A^T)x + \lambda(B + B\text{^}T)x = 0$$

If A and B are symmetric

$$Ax = \kappa \, Bx, \kappa = -\lambda \leftarrow \text{General Eigen Decomposition}$$

Generalized Eigen-problem

$$\min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(D-W)\boldsymbol{y}}{\boldsymbol{y}^T D\boldsymbol{y}}, \boldsymbol{y} \in \mathcal{R}^n, \boldsymbol{y}^T D\mathbf{1} = 0$$

$$(D-W)\boldsymbol{y} = \lambda D\boldsymbol{y}$$

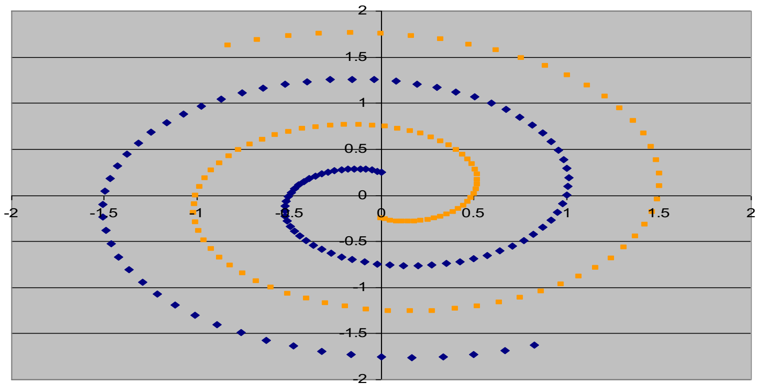
- Eigenvector corresponding to the smallest eigenvalue.
- Vector $\mathbf{1}$ is the eigenvector corresponding to the eigenvalue 0.

$$(D-W)\boldsymbol{y} = \lambda D^{\frac{1}{2}}D^{\frac{1}{2}}\boldsymbol{y}$$
$$D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}D^{\frac{1}{2}}\boldsymbol{y} = \lambda D^{\frac{1}{2}}\boldsymbol{y}$$
$$D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}\boldsymbol{z} = \lambda \boldsymbol{z}$$

- $\boldsymbol{z}_1^T\boldsymbol{z}_2 = 0 \;\Rightarrow\; \left(D^{\frac{1}{2}}\boldsymbol{y}_1\right)^T\left(D^{\frac{1}{2}}\boldsymbol{y}_2\right) = 0 \;\Rightarrow\; \boldsymbol{y}_1^T D\boldsymbol{y}_2 = 0$

- The eigenvector corresponding to the 2nd small eigenvalue.

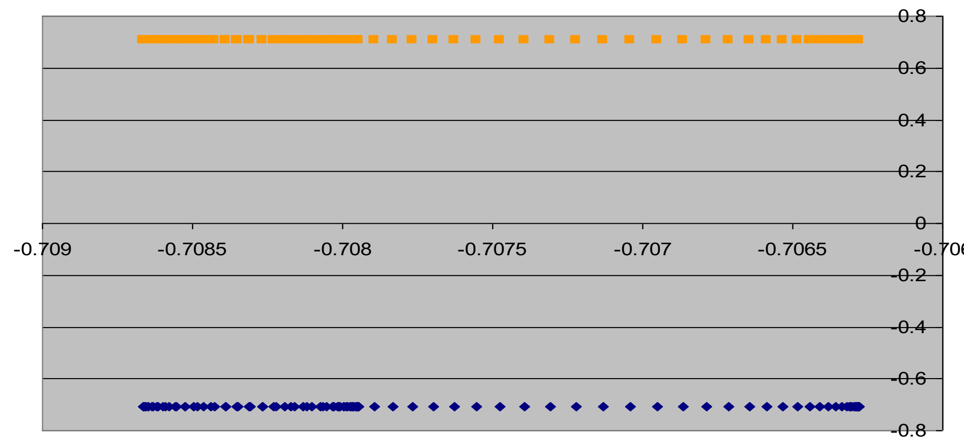
Spectral Clustering Example – 2 Spirals



Dataset exhibits complex cluster shapes

$\Rightarrow$ K-means performs very poorly in this space due bias toward dense spherical clusters.

In the embedded space given by two leading eigenvectors, clusters are trivial to separate.



K > 2

- Perform Ncut recursively.

- Use more than one eigenvectors.
  - Suppose $\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_k$ are the first k eigenvectors corresponding to the smallest eigenvalues, let
  $$Y = [\boldsymbol{y}_1, \boldsymbol{y}_2, \dots, \boldsymbol{y}_k] \in R^{n \times k}$$
  - Each row vector of Y is a k dimensional representation of the original data point.
  - Performing kmeans.

Spectral Clustering Algorithm

1. Graph construction
   - Heat kernel $w_{ij} = \exp\left\{-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}{2\sigma^2}\right\}$
   - k-nearest neighbor graph
2. Eigen-problem
   - Compute eigenvalues and eigenvectors of the matrix L
   - Map each point to a lower-dimensional representation based on one or more eigenvectors.
3. Conventional clustering schemes, e.g. K-Means
   - Assign points to two or more clusters, based on the new representation.