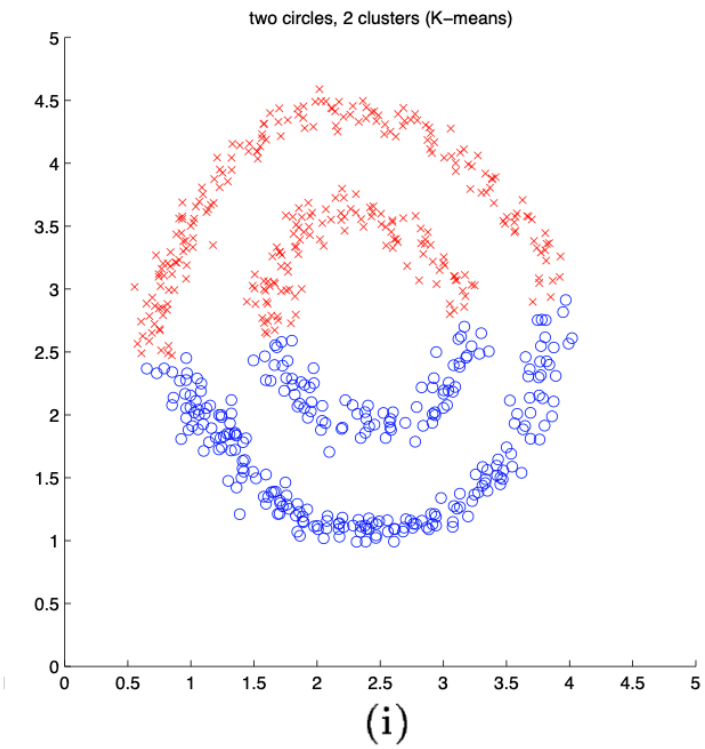
TFIP-AI – Advanced Machine Learning

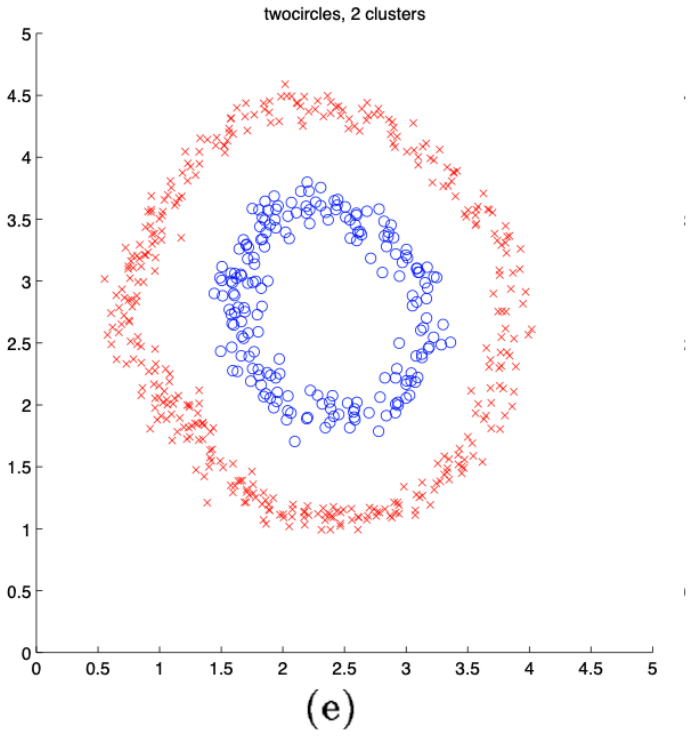
Unit 2 Spectral Clustering

Spectral clustering

K-means

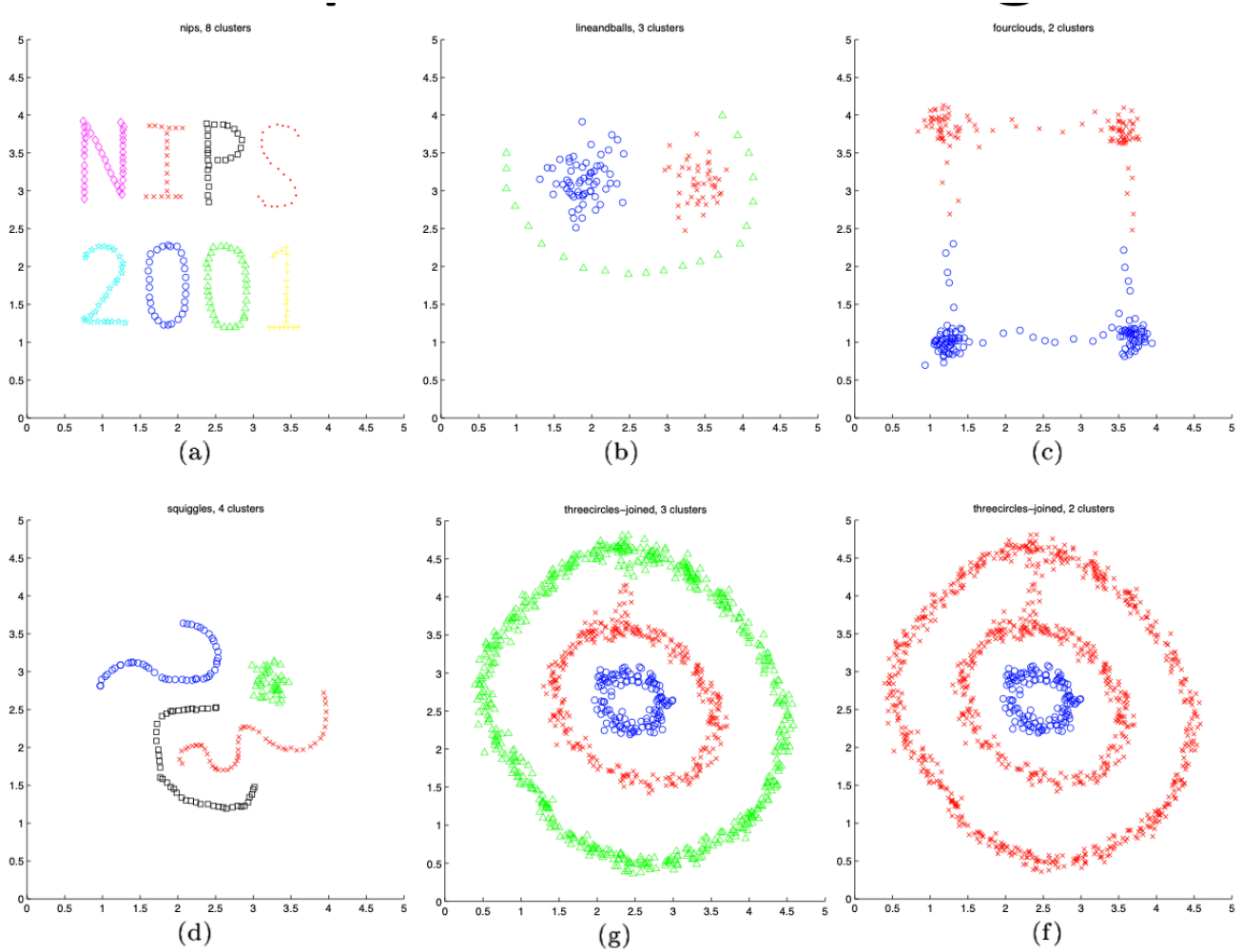


Spectral clustering



[Shi & Malik ‘00; Ng, Jordan, Weiss NIPS ‘01]

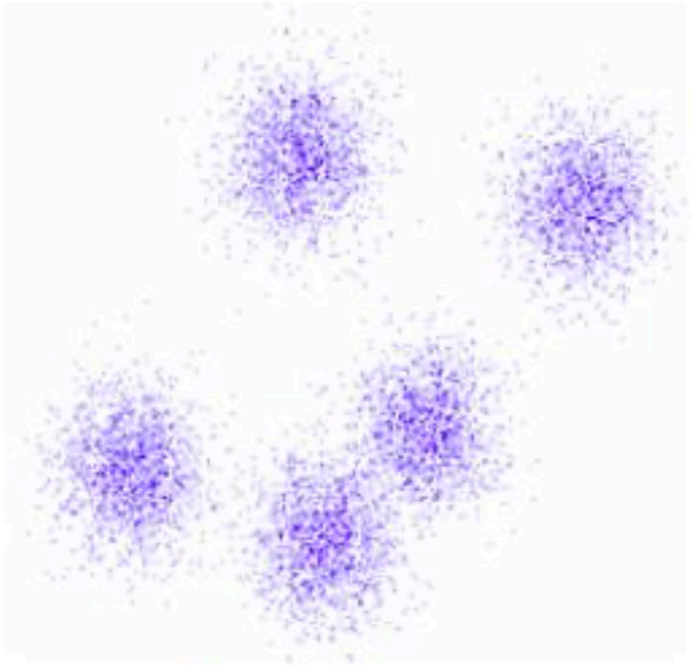
Spectral clustering cont…



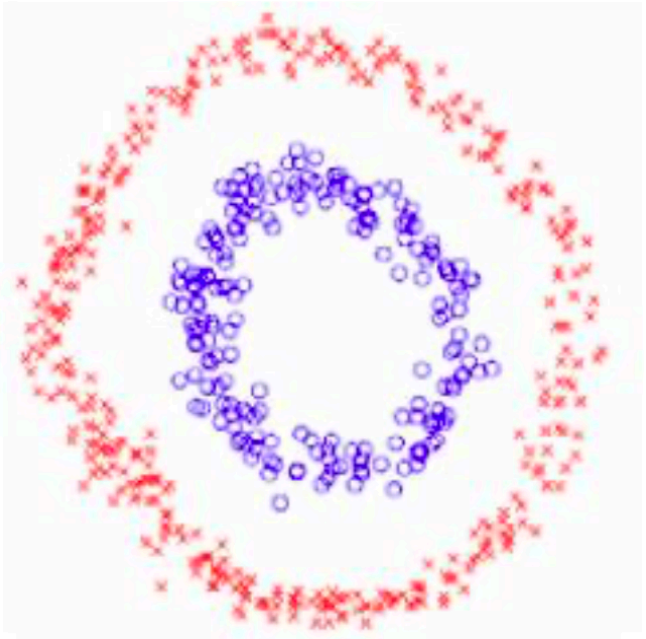
[Figures from Ng, Jordan, Weiss NIPS ‘01]

Data Clustering

- Two different criteria
  - Compactness, e.g., k-means, mixture models
  - Connectivity, e.g., spectral clustering



Compactness



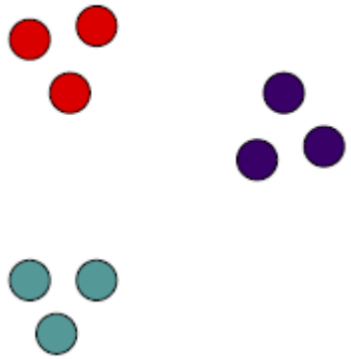
Connectivity

Graph Clustering

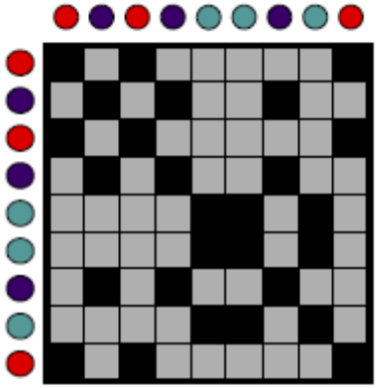
Goal: Given data points $X_1, \ldots, X_n$ and similarities $w(X_i, X_j)$, partition the data into groups so that points in a group are similar and points in different groups are dissimilar.

Similarity Graph: G(V,E,W)

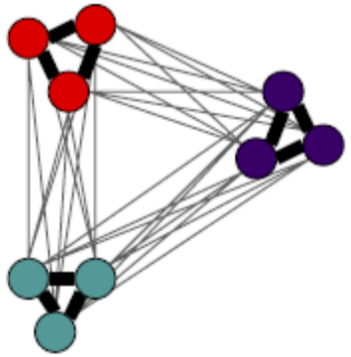
V – Vertices (Data points)
E – Edge if similarity > 0
W - Edge weights (similarities)



Data



Similarities



Similarity graph

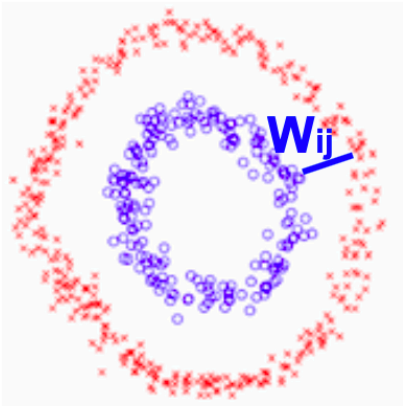
Partition the graph so that edges within a group have large weights and edges across groups have small weights.

Similarity graph construction

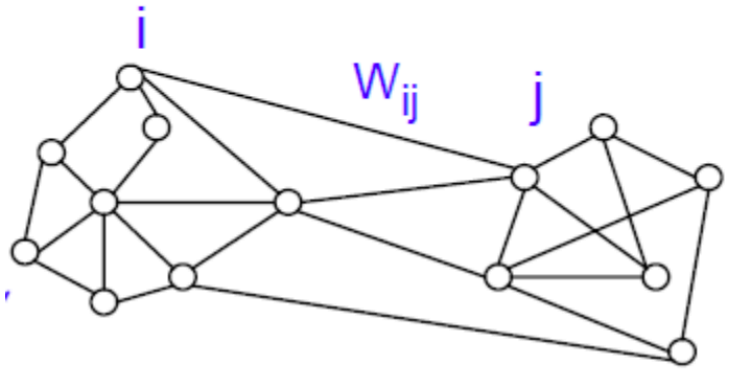
Similarity Graphs: Model local neighborhood relations between data points

E.g. Gaussian kernel similarity function

$$W_{ij} = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}$$ → Controls size of neighborhood



Data clustering

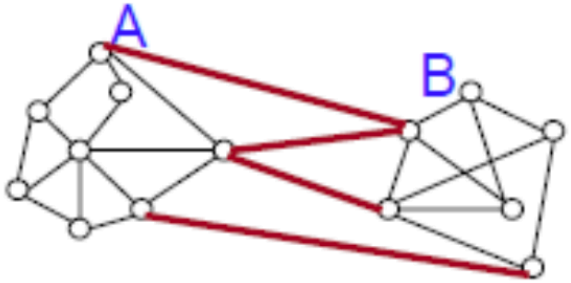


$G = \{V,E\}$

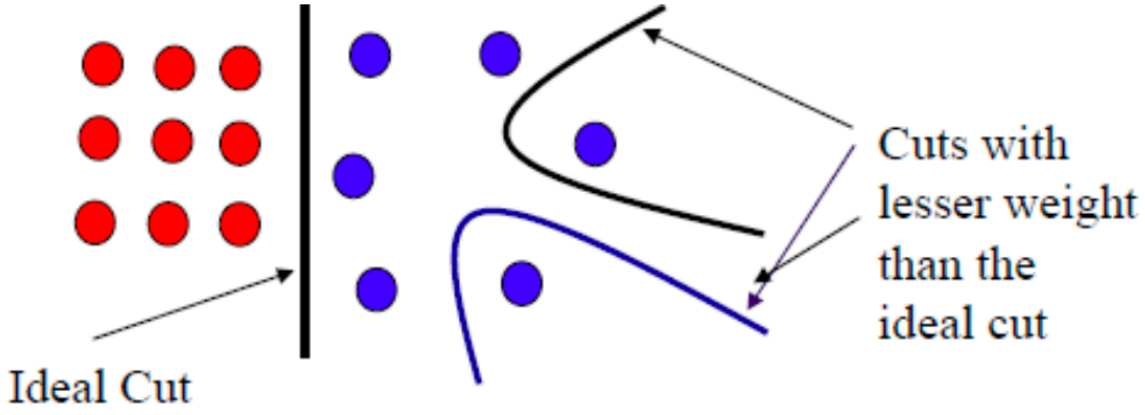
Partitioning a graph into two clusters

Min-cut: Partition graph into two sets A and B such that weight of edges connecting vertices in A to vertices in B is minimum.

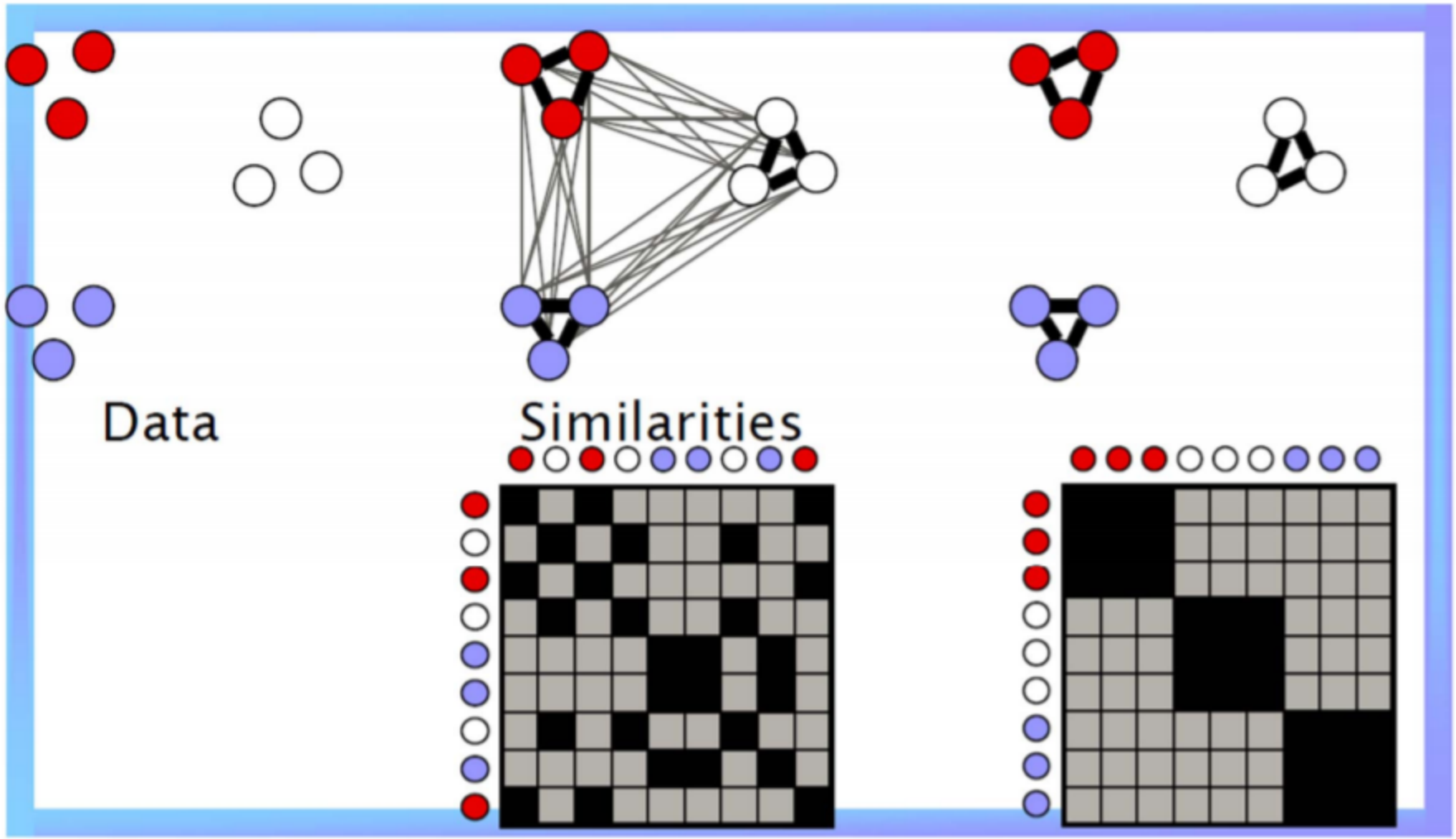
$$\text{cut}(A, B) := \sum_{i \in A, j \in B} w_{ij}$$



- Easy to solve O(VE) algorithm
- Not satisfactory partition – often isolates vertices



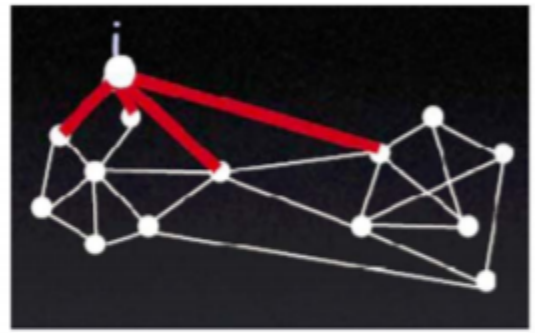
Graph partitioning

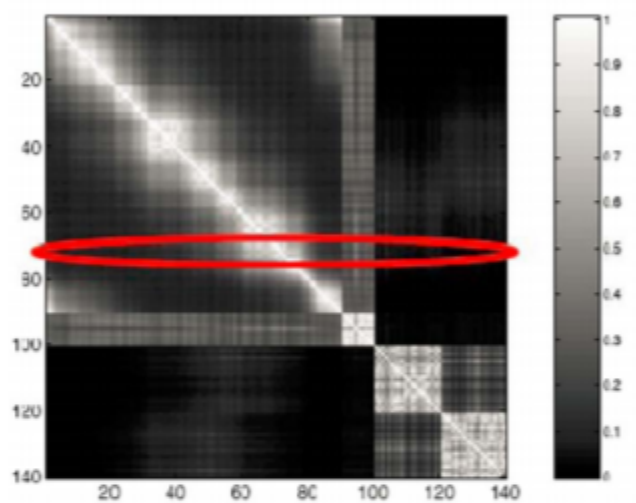


Graph Terminologies

- Degree of nodes

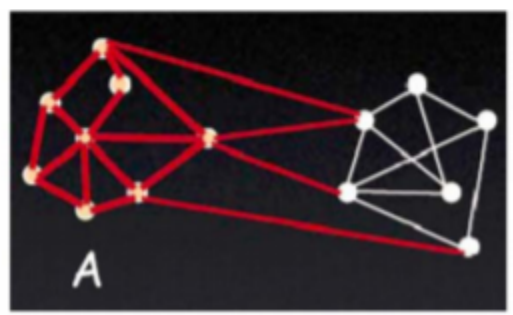
$$d_i = \sum_j w_{i,j}$$

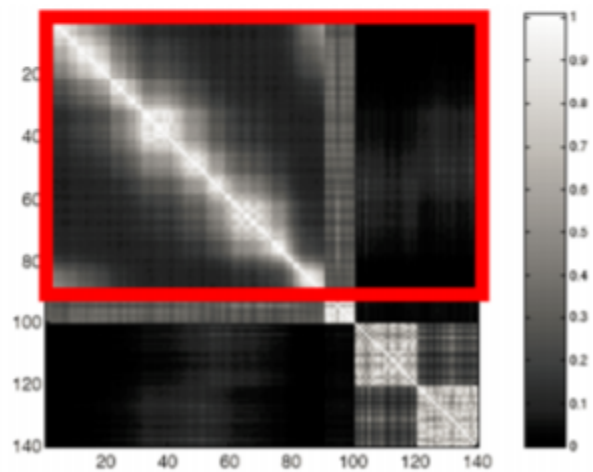




- Volume of a set

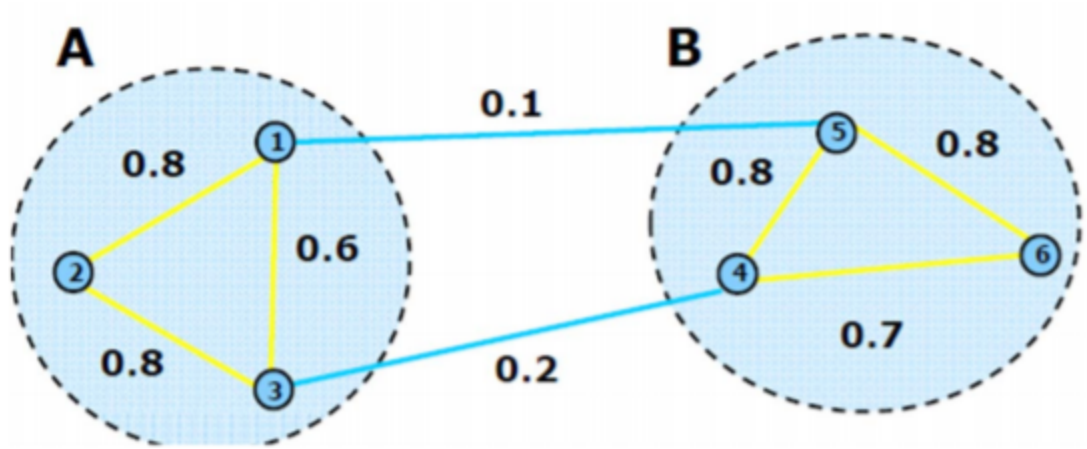
$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$





Graph Cut

- Consider a partition of the graph into two parts A and B



- ***Cut(A, B)***: sum of the weights of the set of edges that connect the two groups

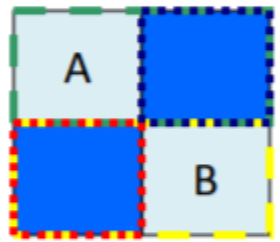
$$cut(A,B) = \sum_{i \in A, j \in B} w_{ij} = \boldsymbol{0.3}$$

- An intuitive goal is find the partition that minimizes the cut

Normalized Cut

- Consider the connectivity between groups relative to the volume of each group

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



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

Minimized when Vol(A) and Vol(B) are equal.
Thus encourage balanced cut

Solving NCut

- How to minimize $Ncut$?

$$\text{Let } W \text{ be the similarity matrix, } W(i,j) = W_{i,j};$$
$$\text{Let D be the diag. matrix, } D(i,i) = \sum_j W(i,j);$$
$$\text{Let } x \text{ be a vector in } \{1,-1\}^N, x(i) = 1 \Leftrightarrow i \in A.$$

- Relax the optimization problem into the continuous domain by solving generalized eigenvalue system:

$$\min_y y^T(D-W)y \text{ subject to } y^T Dy = 1$$

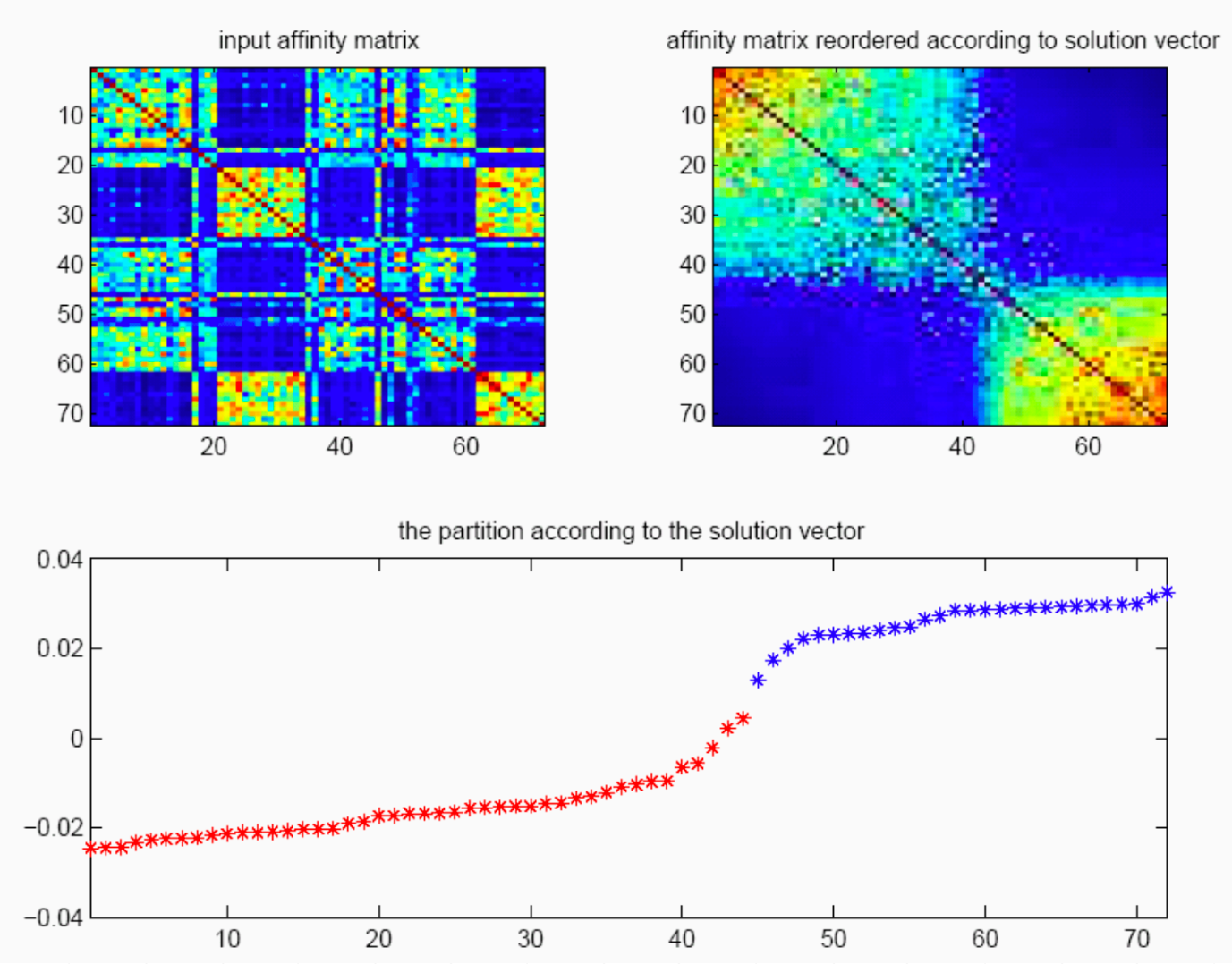
- Which gives: $(D-W)y = \lambda Dy$
- Note that $(D-W)1 = 0$, so the first eigenvector is $y_0 = 1$ with eigenvalue 0.
- The second smallest eigenvector is the real valued solution to this problem!!

2-way Normalized Cuts

1. Compute the affinity matrix W, compute the degree matrix (D), D is diagonal and $D(i,i) = \sum_{j \in V} W(i,j)$
2. Solve $(D - W)y = \lambda D y$, where $D - W$ is called the Laplacian matrix
3. Use the eigenvector with the second smallest eigen-value to bipartition the graph into two parts.

Example

Xing et al 2001



Spectral clustering for segmentation