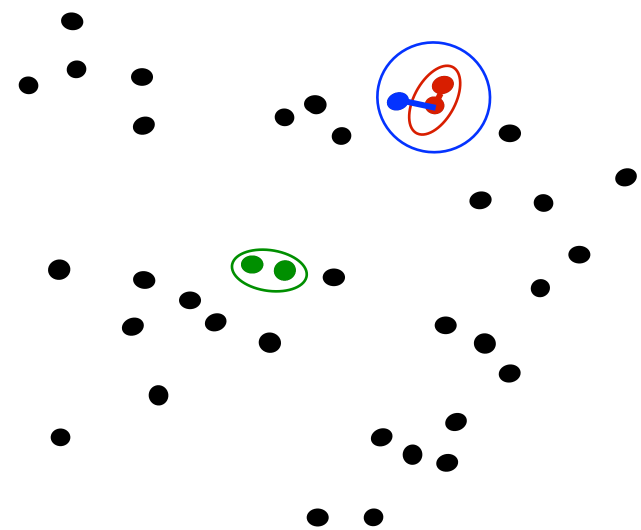
Hierarchical & Spectral clustering
Lecture 13

David Sontag
New York University

Slides adapted from Luke Zettlemoyer, Vibhav Gogate, Carlos Guestrin, Andrew Moore, Dan Klein

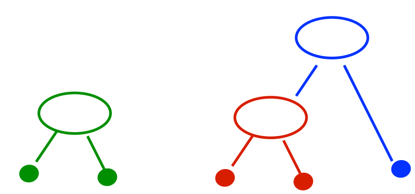
Agglomerative Clustering

- Agglomerative clustering:
  - First merge very similar instances
  - Incrementally build larger clusters out of smaller clusters



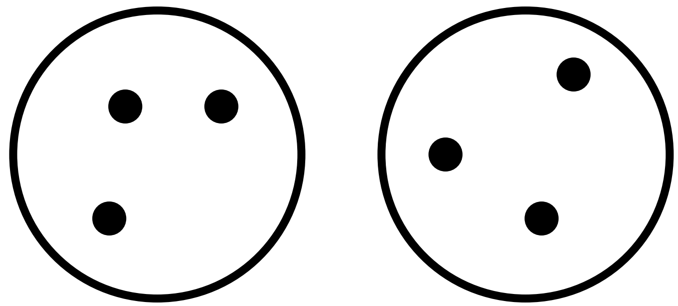
- Algorithm:
  - Maintain a set of clusters
  - Initially, each instance in its own cluster
  - Repeat:
    - Pick the two closest clusters
    - Merge them into a new cluster
    - Stop when there's only one cluster left

- Produces not one clustering, but a family of clusterings represented by a dendrogram



Agglomerative Clustering

- How should we define “closest” for clusters with multiple elements?

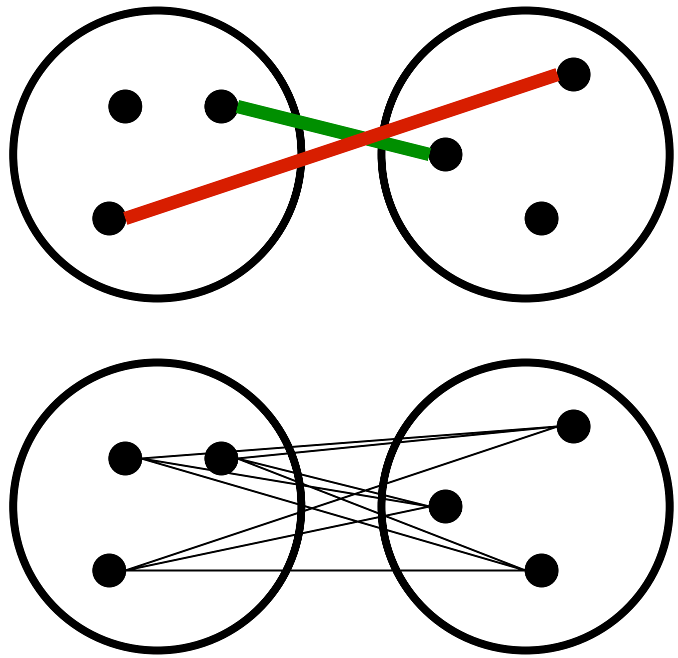


Agglomerative Clustering

- How should we define "closest" for clusters with multiple elements?

- Many options:
  - Closest pair (single-link clustering)
  - Farthest pair (complete-link clustering)
  - Average of all pairs

- Different choices create different clustering behaviors

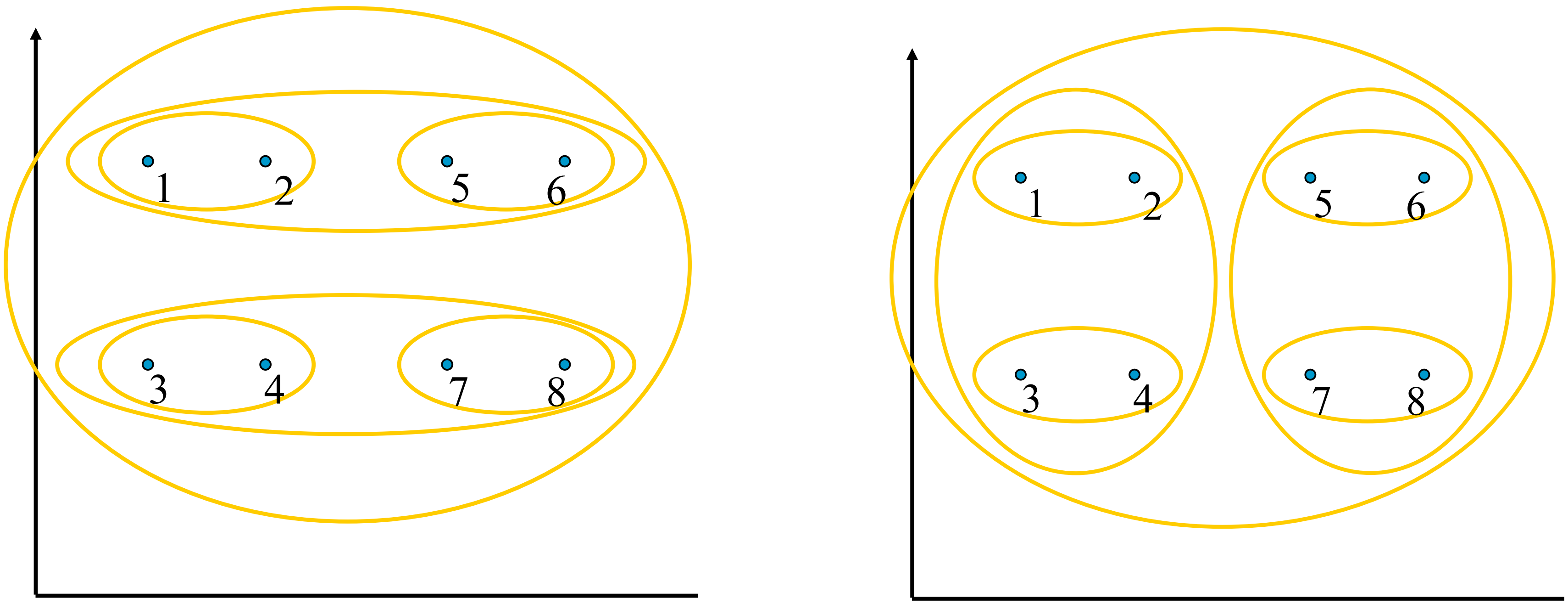


Agglomerative Clustering

- How should we define “closest” for clusters with multiple elements?

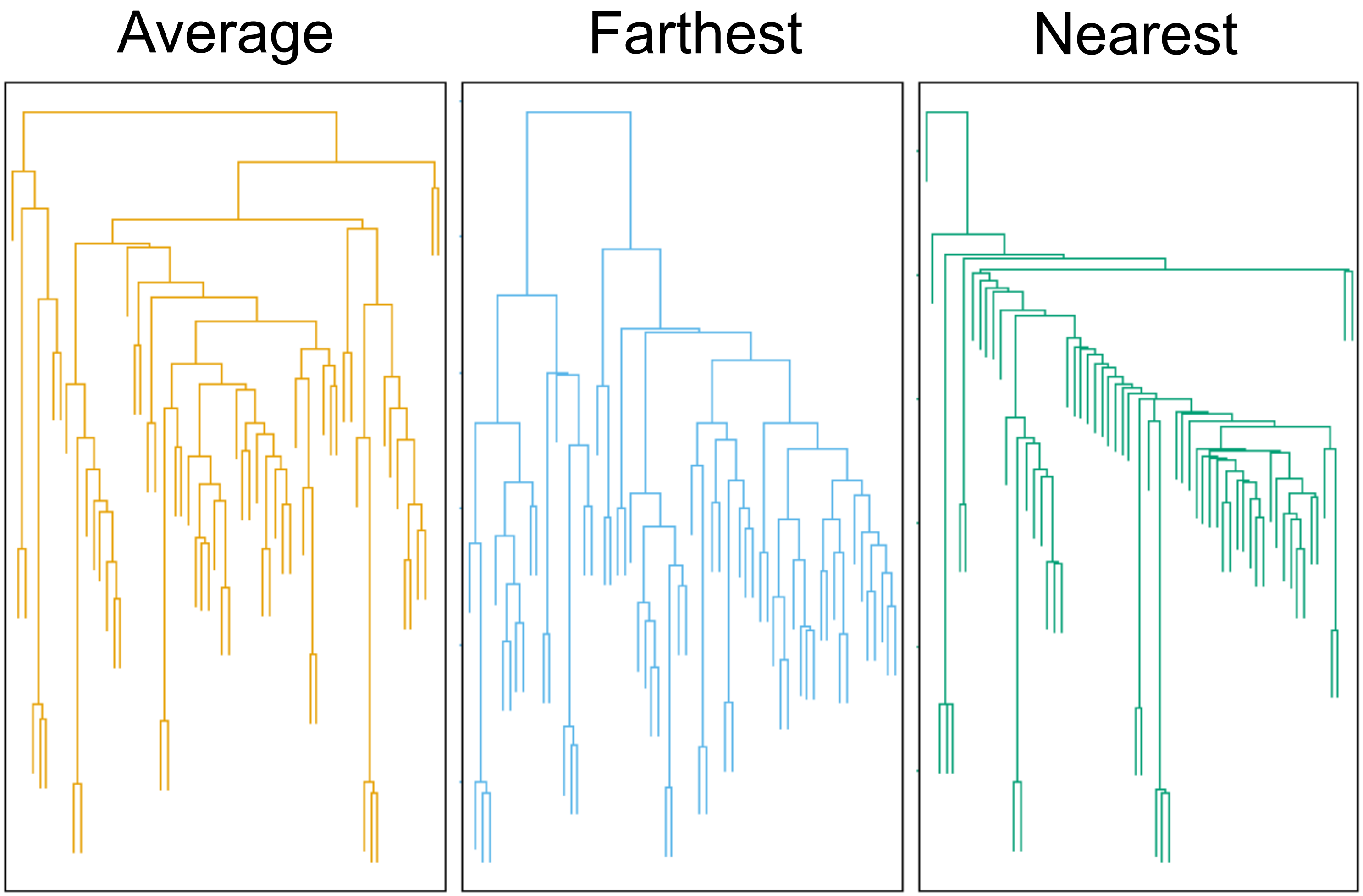
Closest pair
(single-link clustering)

Farthest pair
(complete-link clustering)



[Pictures from Thorsten Joachims]

Clustering Behavior

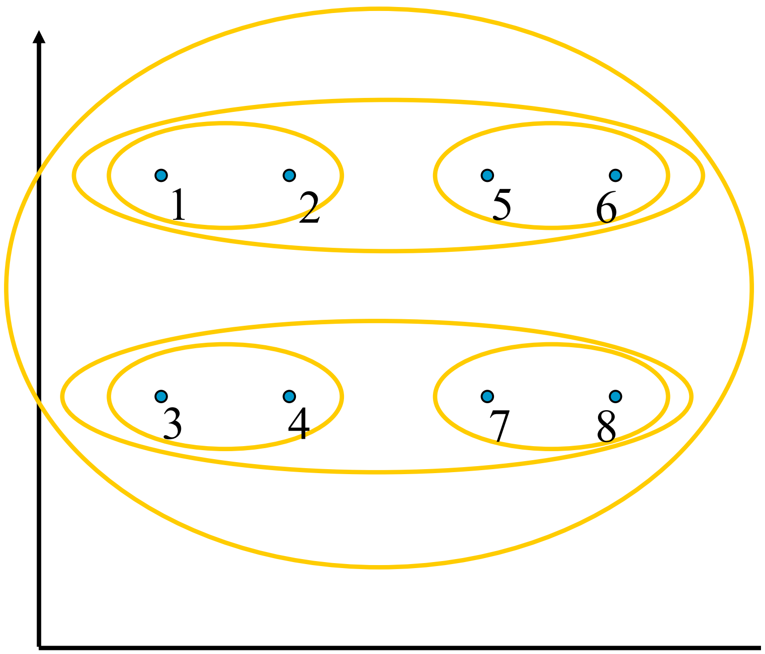


Mouse tumor data from [Hastie *et al.*]

Agglomerative Clustering

When can this be expected to work?

Closest pair
(single-link clustering)



Strong separation property:
All points are more similar to points in their own cluster than to any points in any other cluster

Then, the true clustering corresponds to some **pruning** of the tree obtained by single-link clustering!

Slightly weaker (stability) conditions are solved by average-link clustering

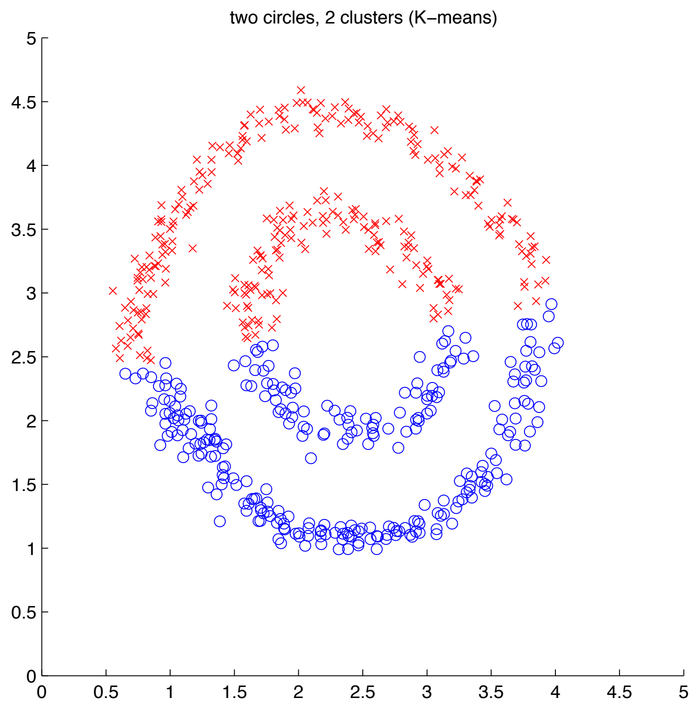
(Balcan et al., 2008)

Spectral Clustering

Slides adapted from James Hays, Alan Fern, and Tommi Jaakkola

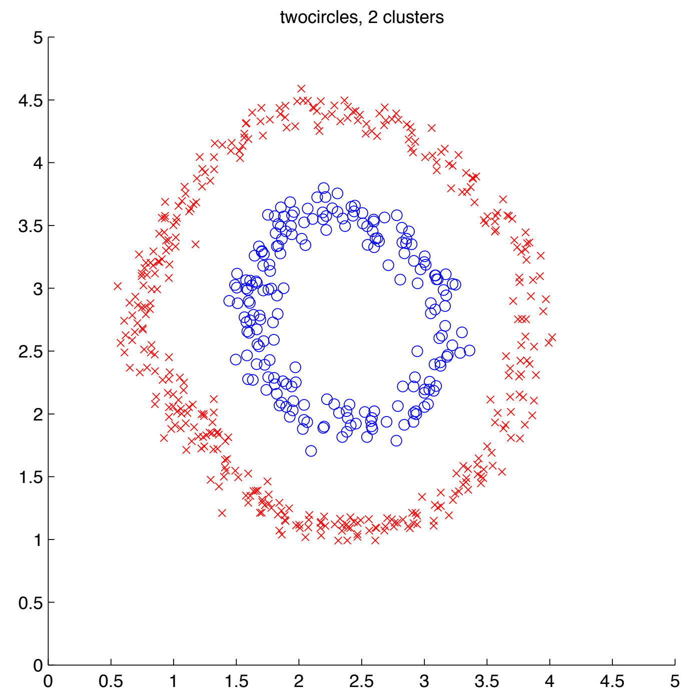
Spectral clustering

K-means



(i)

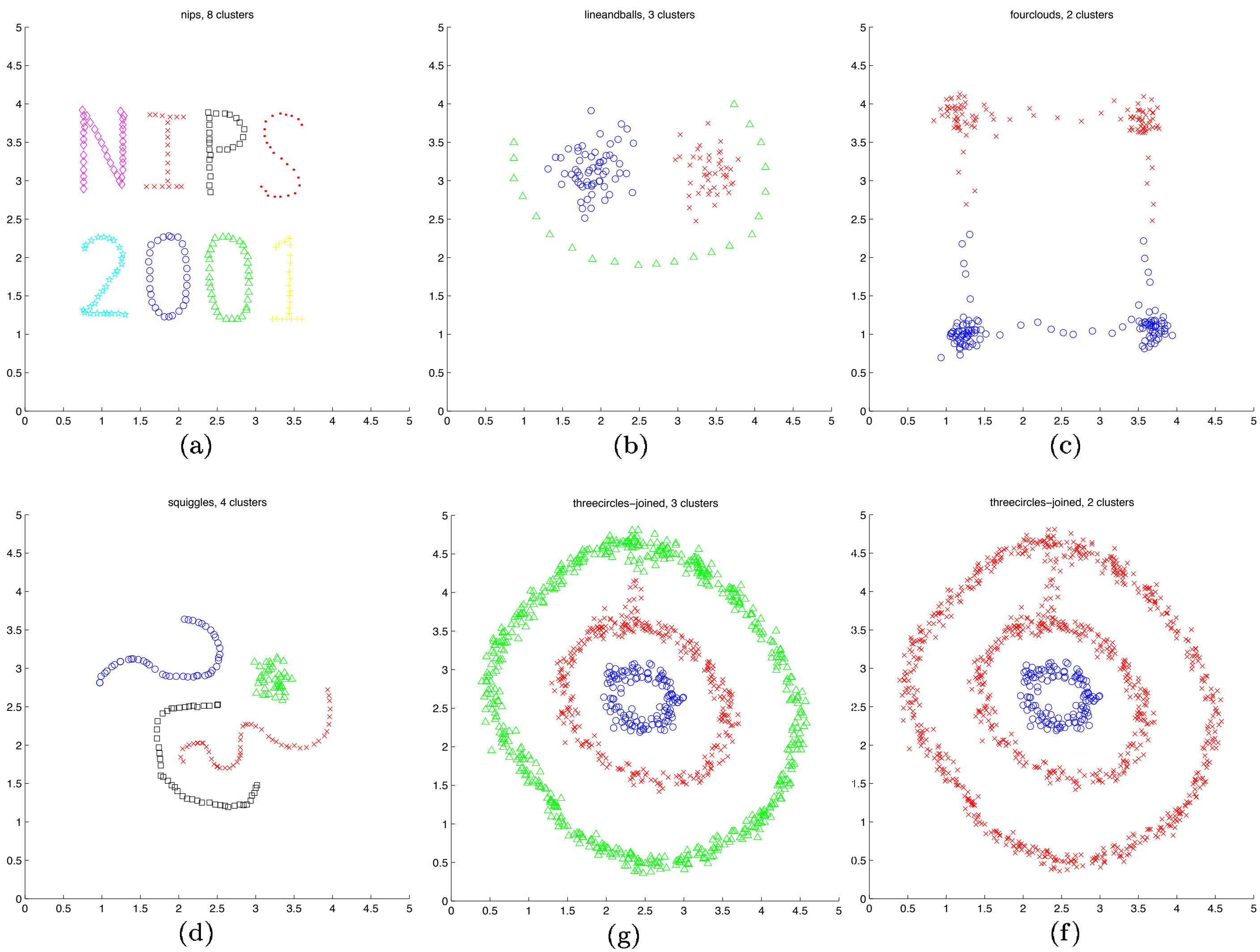
Spectral clustering



(e)

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

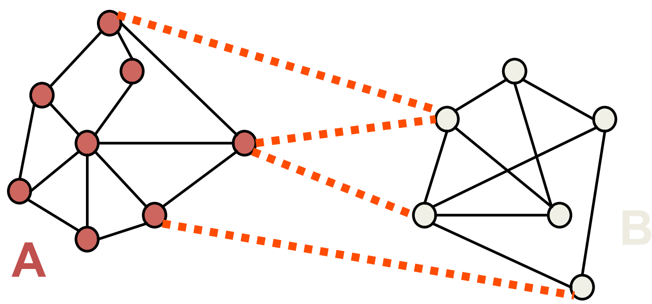
Spectral clustering

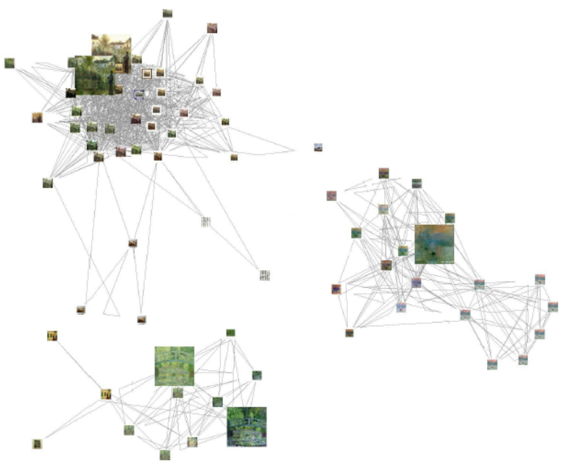


[Figures from Ng, Jordan, Weiss NIPS ‘01]

Spectral clustering

Group points based on links in a graph





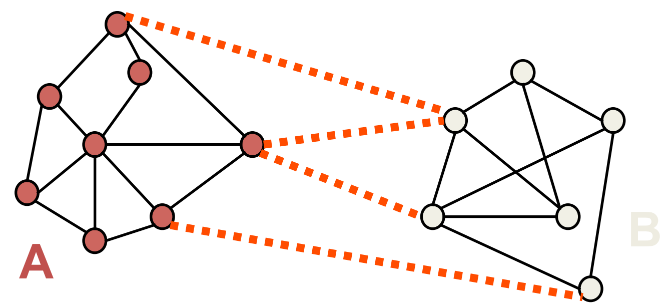
[Slide from James Hays]

How to Create the Graph ?

- It is common to use a Gaussian Kernel to compute similarity between objects

$$W(i,j) = \exp \frac{-\left|x_i - x_j\right|^2}{\sigma^2}$$

- One could create
  - A fully connected graph
  - K-nearest neighbor graph (each node is only connected to its K-nearest neighbors)



[Slide from Alan Fern]

Can we use minimum cut for clustering?

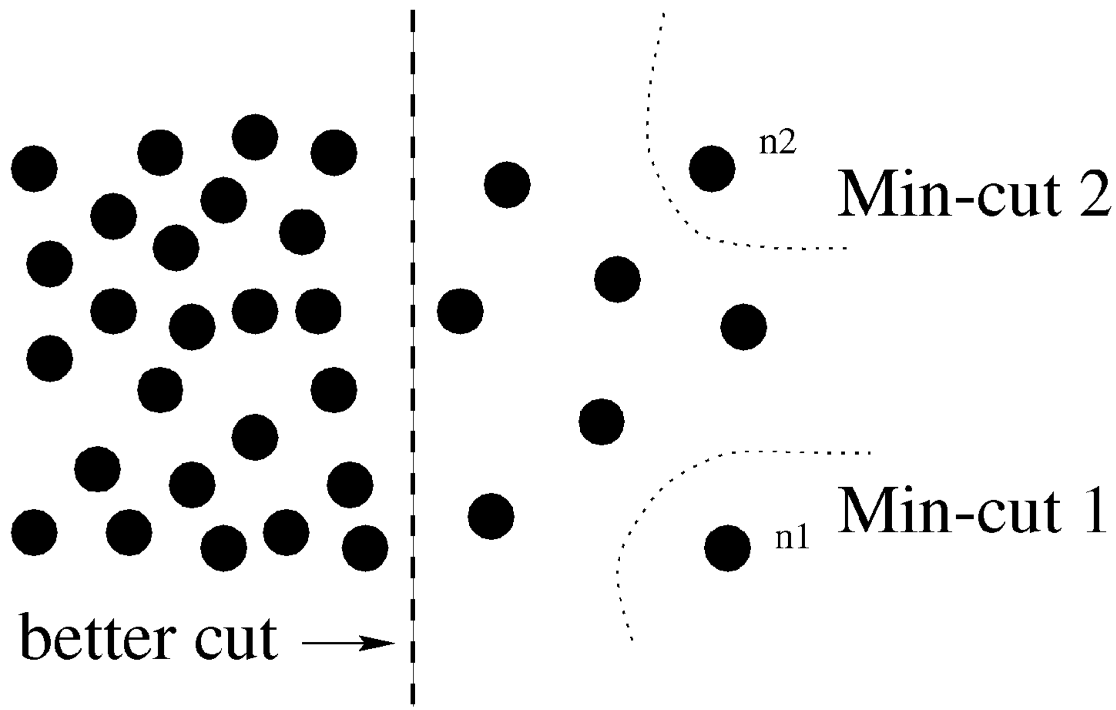
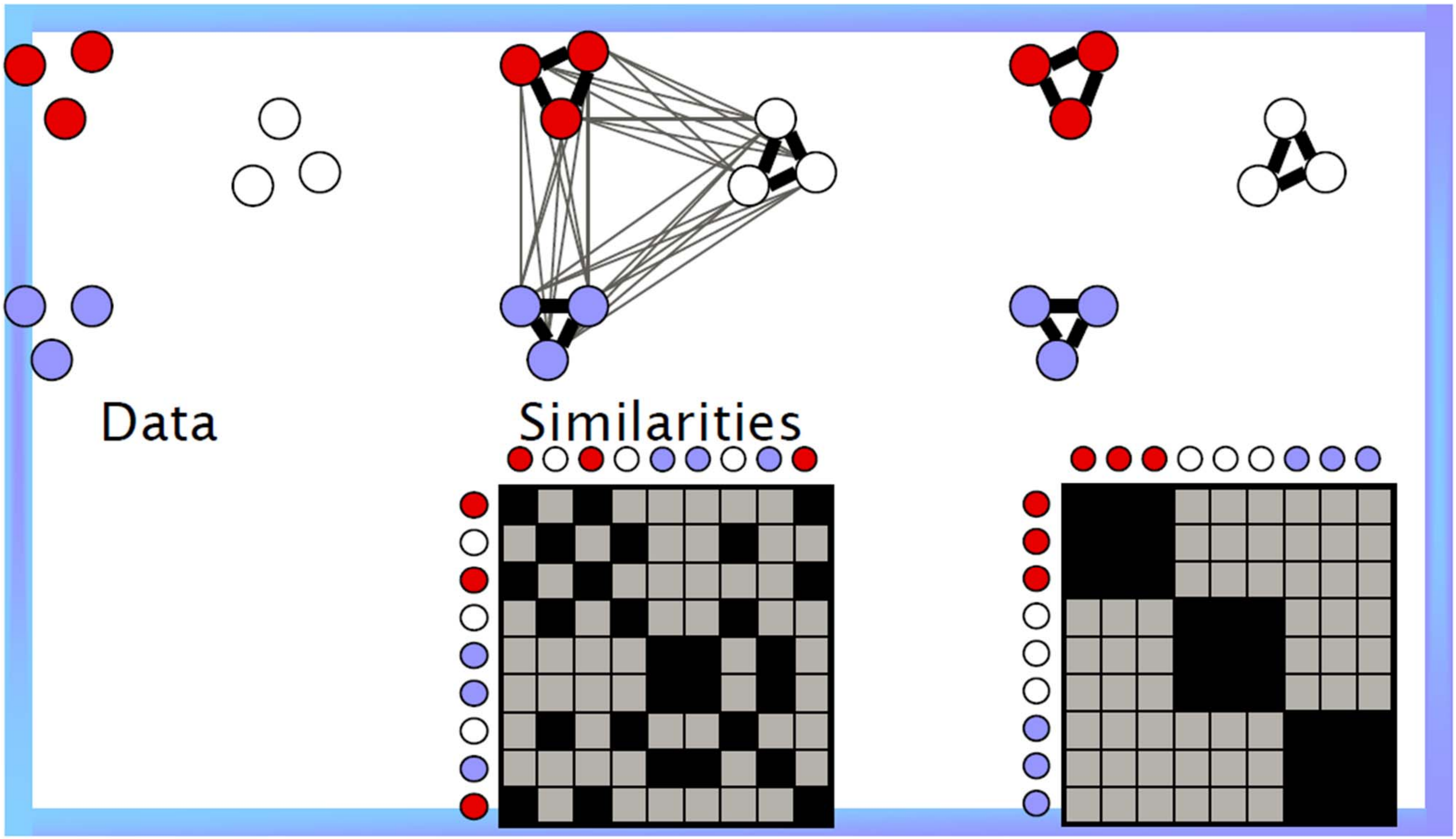


Fig. 1. A case where minimum cut gives a bad partition.

[Shi & Malik '00]

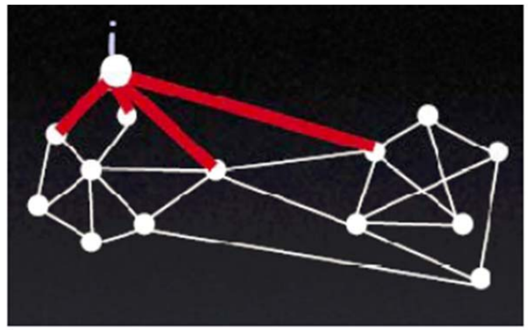
Graph partitioning

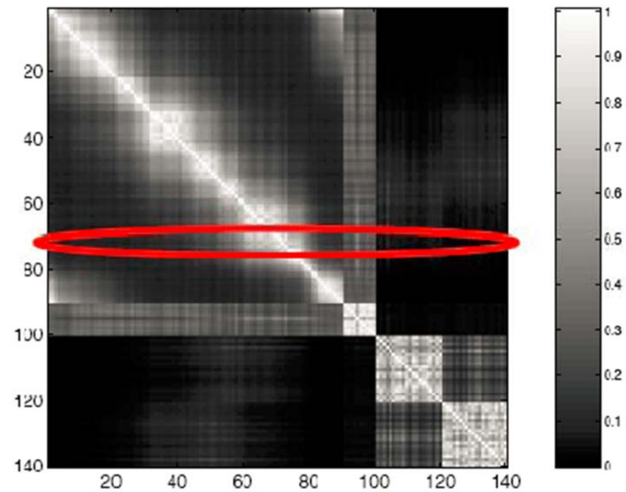


Graph Terminologies

- Degree of nodes

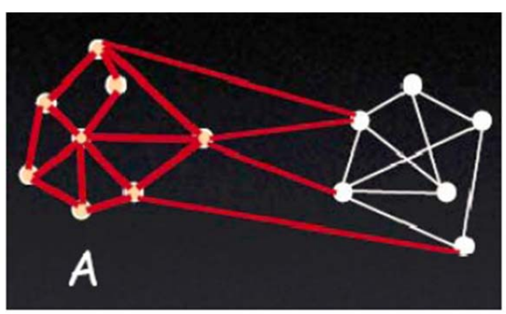
$$d_i = \sum_j w_{i,j}$$

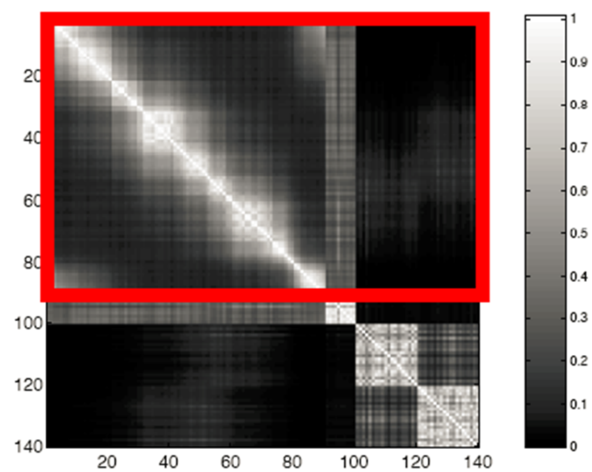




- Volume of a set

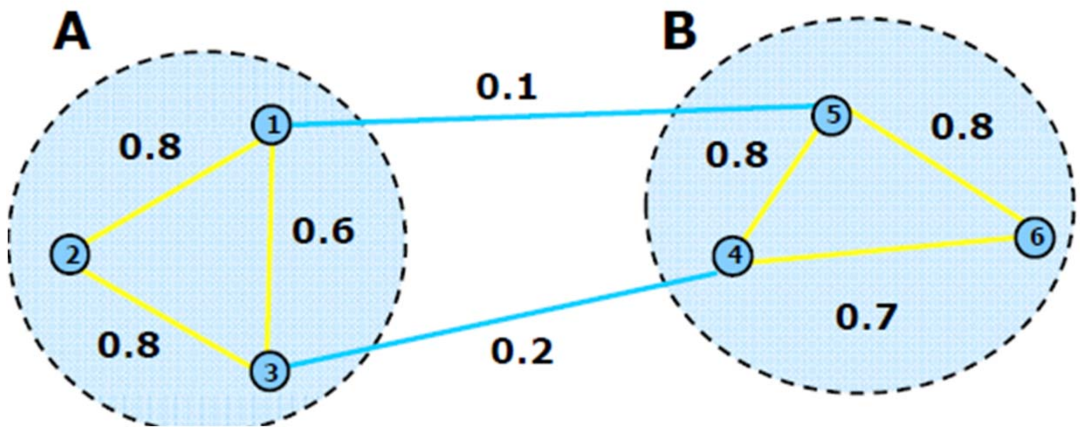
$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$





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

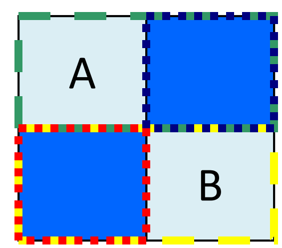
$$cut(A,B) = \sum_{i \in A, j \in B} w_{ij} = \boldsymbol{0.3}$$

- 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*?

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

- With some simplifications, we can show:

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

Rayleigh quotient

Subject to: $y^T D1 = 0$ ($\boldsymbol{y}$ *takes discrete values*)

NP-Hard!

Solving NCut

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

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

- Which gives: $(D - W)y = \lambda D y$
- 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!!

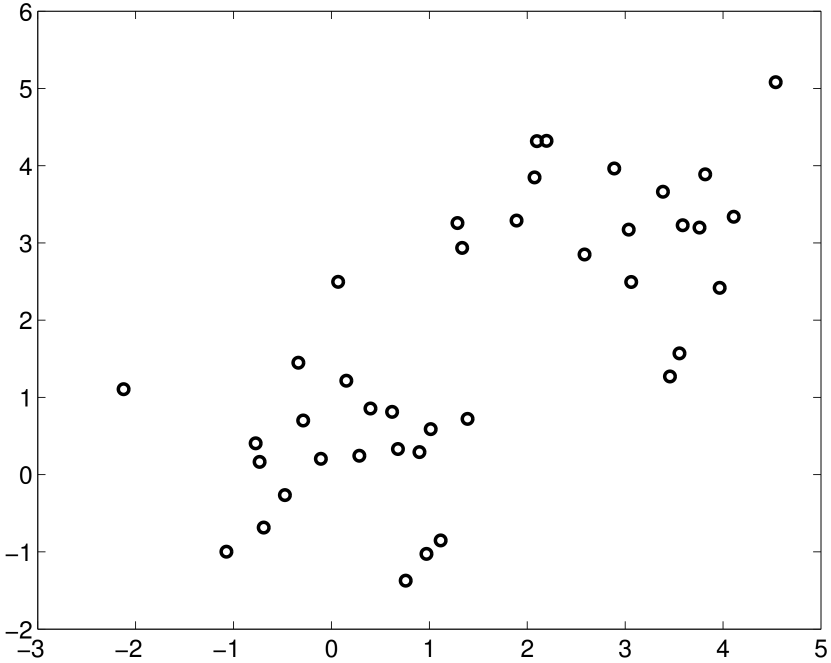
2-way Normalized Cuts

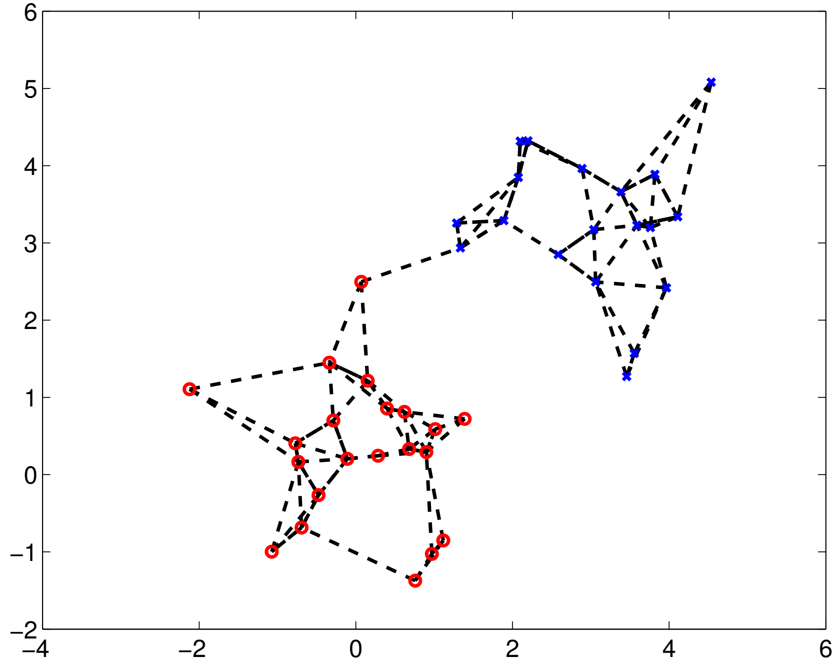
1. Compute the affinity matrix W, compute the degree matrix (D), D is diagonal and $D(i,i) = \sum_{j \in V} W(i,j)$
2. Solve $(D - W)y = \lambda Dy$, where $D - W$ is called the Laplacian matrix
3. Use the eigenvector with the second smallest eigen-value to bipartition the graph into two parts.

Creating Bi-partition Using 2nd Eigenvector

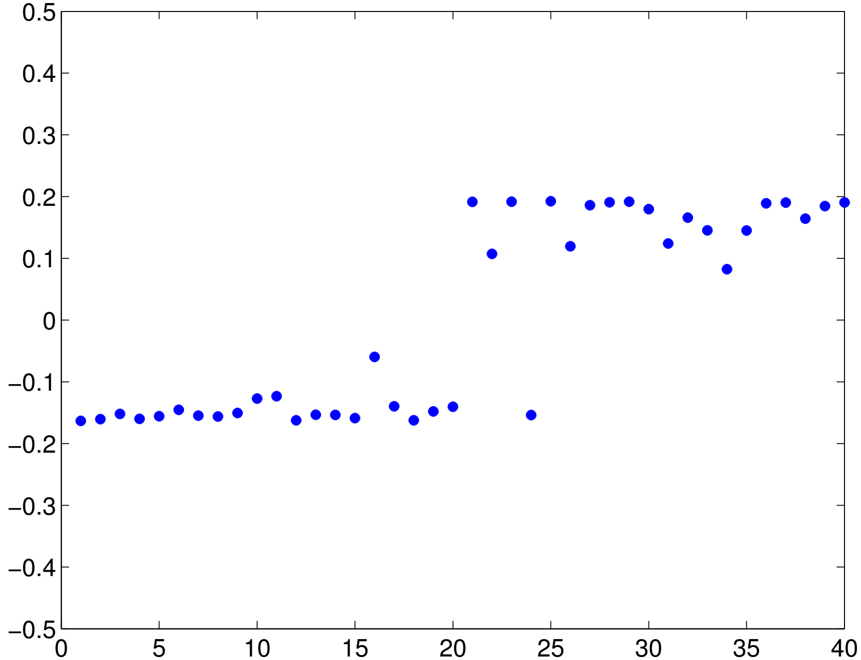
- Sometimes there is not a clear threshold to split based on the second vector since it takes continuous values
- How to choose the splitting point?
  a) Pick a constant value (0, or 0.5).
  b) Pick the median value as splitting point.
  c) Look for the splitting point that has the minimum $Ncut$ value:
     1. Choose n possible splitting points.
     2. Compute $Ncut$ value.
     3. Pick minimum.

Spectral clustering: example





Spectral clustering: example cont'd



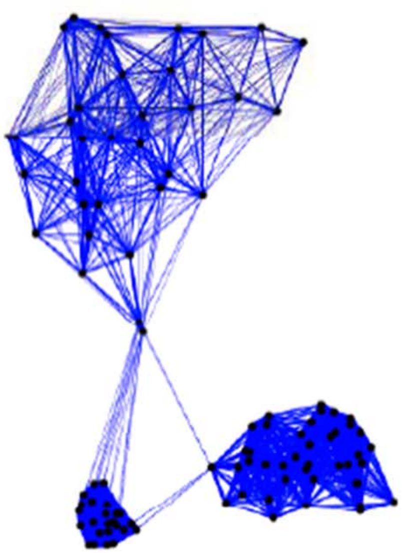
Components of the eigenvector corresponding to the second largest eigenvalue

K-way Partition?

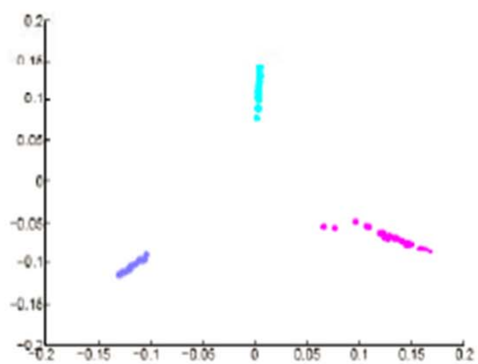
- Recursive bi-partitioning (Hagen et al.,'91)
  - Recursively apply bi-partitioning algorithm in a hierarchical divisive manner.
  - Disadvantages: Inefficient, unstable
- Cluster multiple eigenvectors
  - Build a reduced space from multiple eigenvectors.
  - Commonly used in recent papers
  - A preferable approach… its like doing dimension reduction then k-means

Beyond bi-partition

Graph, 20-NN



$\mathbf{Z}$



Clustering