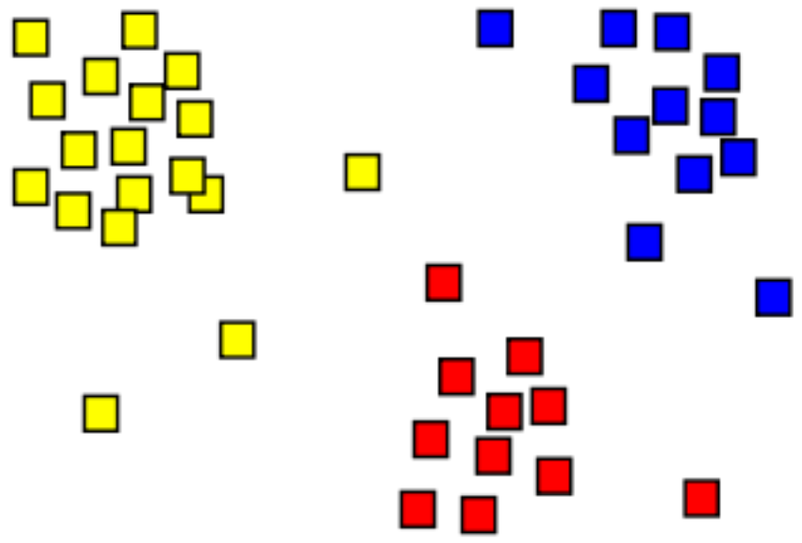
Clustering

Clustering

- Unsupervised learning task
- Basic principle: clustering by similarity between data points
- Many applications
  - User profiling
  - Explorative data analysis
  - Genome research
  - Image segmentation
  - Search engine



What principles in your mind if you want to do clustering?

- Many families of algorithms
  - K-means
  - Hierarchical clustering
  - Spectral clustering
  - Density-based clustering
  - Others…

K-means

- The most popular clustering method

- The idea: consider a cluster as a group of data points whose inter-point distances are small compared with the distances to the points outside the cluster.

- Formalizing the simple idea

Minimize:

$$J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2$$

number of clusters

cluster indicator, 0-1

cluster centroid

- Alternatively updating the two variables: cluster indicators and cluster centroids

  - E step (expectation): Update cluster indicators

$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg\min_j \|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

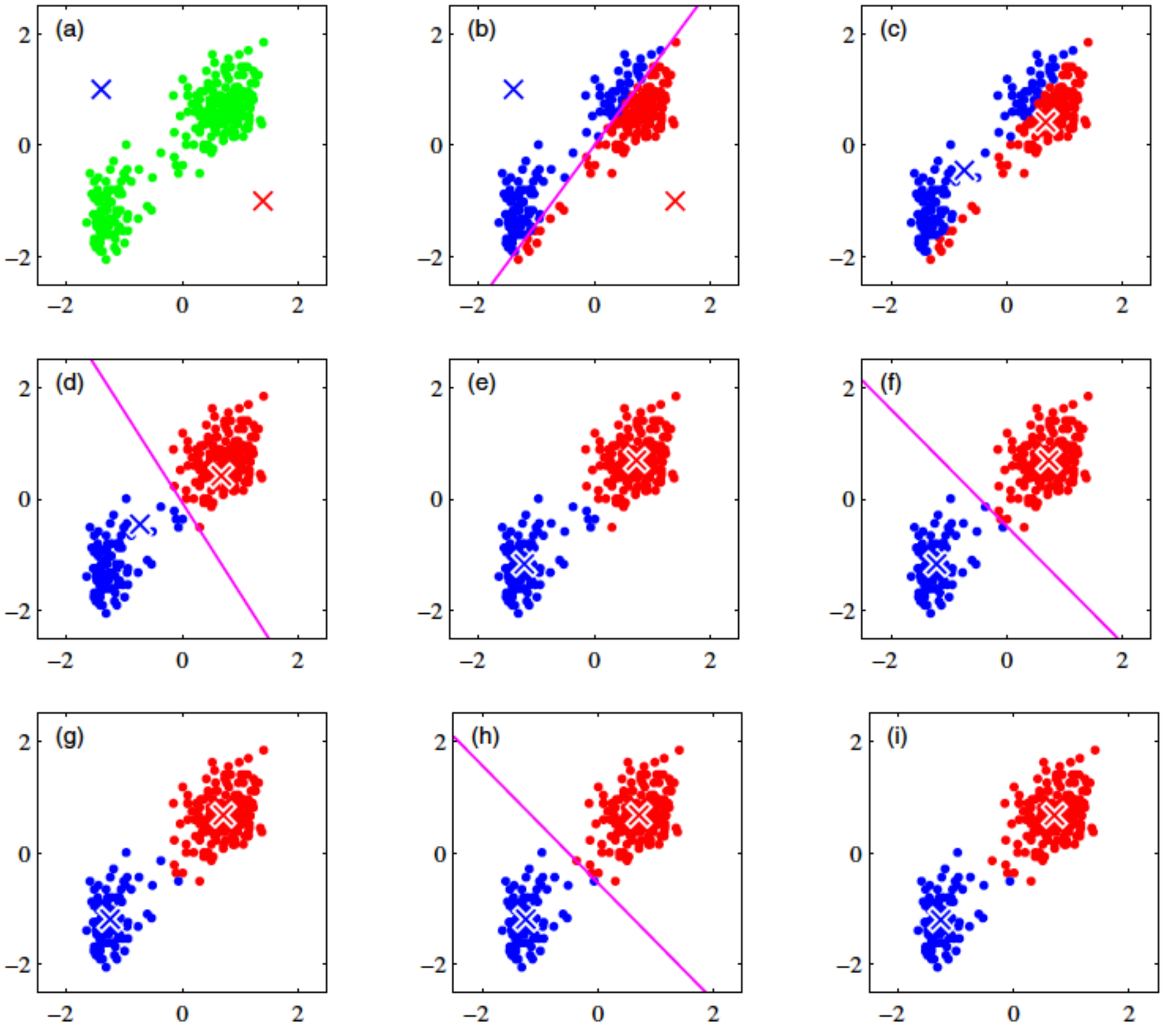
  - M step (maximization): Update cluster centroids

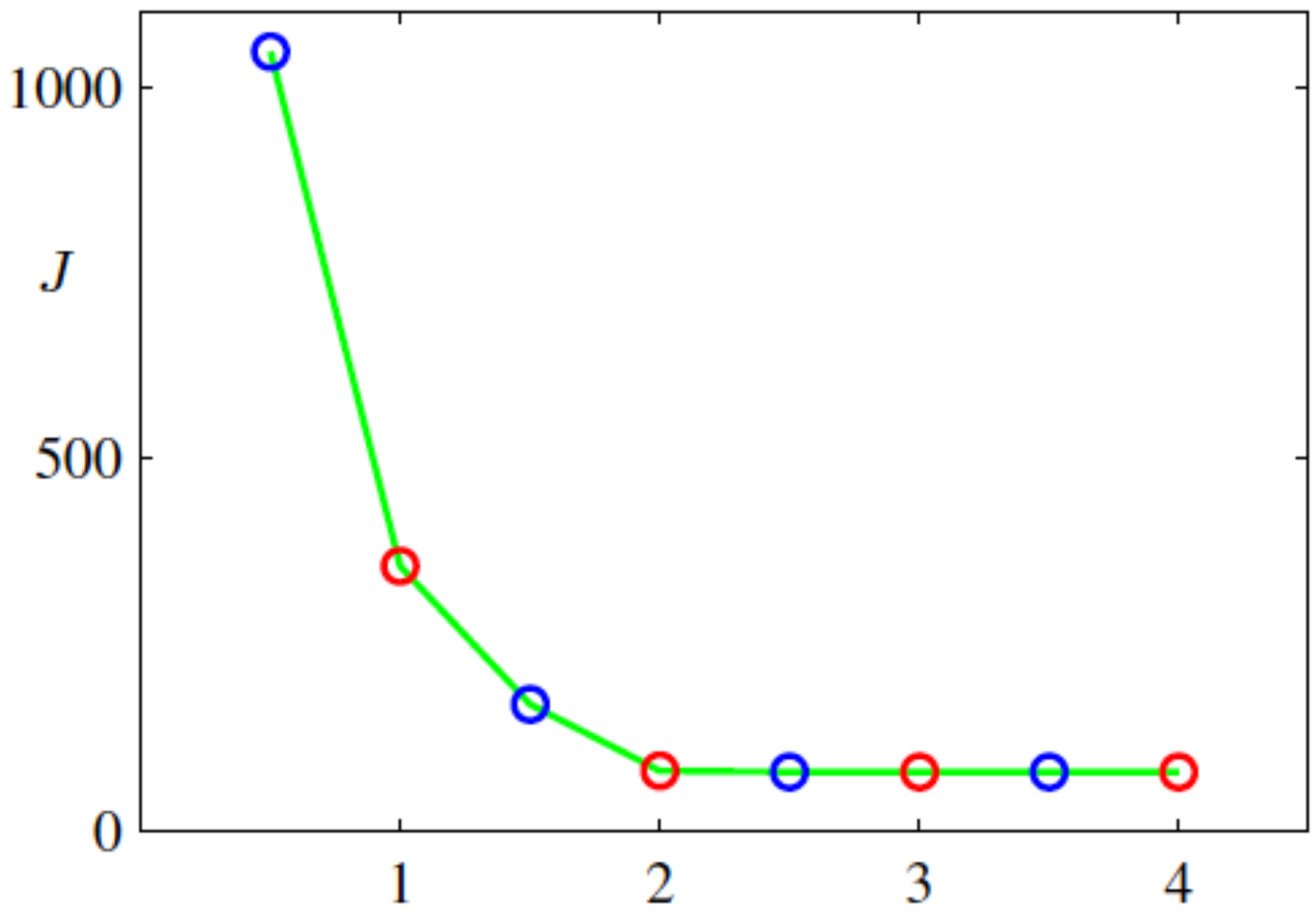
$$2\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \boldsymbol{\mu}_k) = 0$$

$$\boldsymbol{\mu}_k = \frac{\sum_n r_{nk}\mathbf{x}_n}{\sum_n r_{nk}}$$

mean of data points assigned to cluster k

- The two steps **converge to local minima** since each iteration lowers the objective function.


(a)
(b)
(c)
(d)
(e)
(f)
(g)
(h)
(i)
2
0
−2
−2
0
2

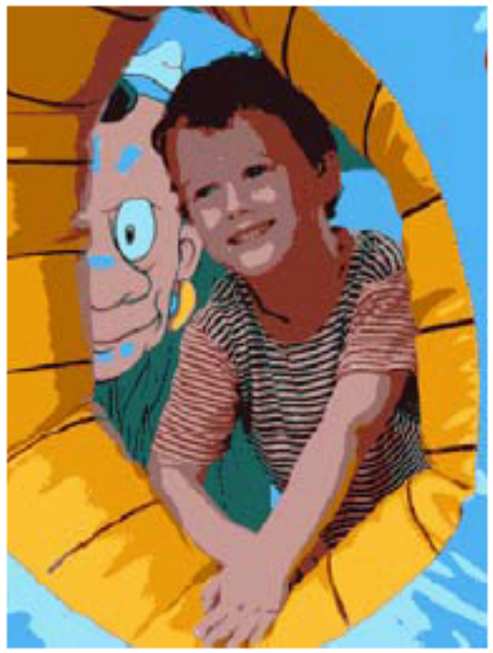


Objective function w.r.t. iteration. Blue dots: E step. Red dots: M step.

$K = 2$

$K = 3$

$K = 10$



Original image

Bishop (2006)

- Several issues of K-means

1. Initialization of centroids

   - Select the best one from multiple random initialization

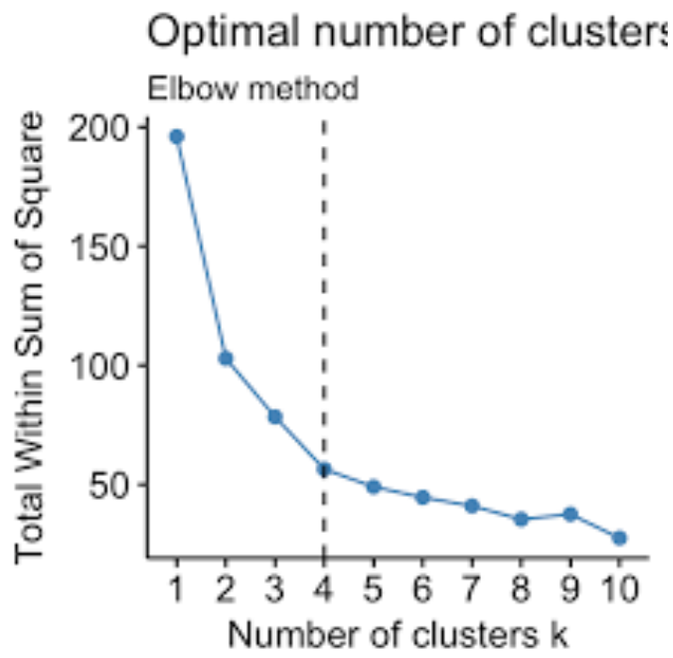
   - Initialize from hierarchical clustering

2. Selection of K value

   - Heuristic method: elbow approach

   - Information-theoretic approach (X-means)

     - Maximize Bayesian Information Criterion, BIC scoring



$$BIC(\mathcal{M}) = \hat{L}(\mathcal{D}|\mathcal{M}) - \frac{p}{2}\log n$$

para of the model

- 3. Distance measure

  - Euclidean distance

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{\sum_{i=1}^{d} (x_{1i} - x_{2i})^2}$$

  - Manhattan distance

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sum_{d} |x_{1i} - x_{2i}|$$

  - Cosine distance

$$\cos(\boldsymbol{x}_1, \boldsymbol{x}_2) = \frac{\boldsymbol{x}_1^{\mathrm{T}} \boldsymbol{x}_2}{\|\boldsymbol{x}_1\|_2 \|\boldsymbol{x}_2\|_2} = \frac{\sum_{i=1}^{d} x_{1i} x_{2i}}{\sqrt{\sum_{i=1}^{d} x_{1i}^2} \sqrt{\sum_{i=1}^{d} x_{2i}^2}}$$

  - Mahalanobis distance

$$d(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sqrt{(\boldsymbol{x}_1 - \boldsymbol{x}_2)^{\mathrm{T}} \boldsymbol{\Sigma}^{-1} (\boldsymbol{x}_1 - \boldsymbol{x}_2)}$$

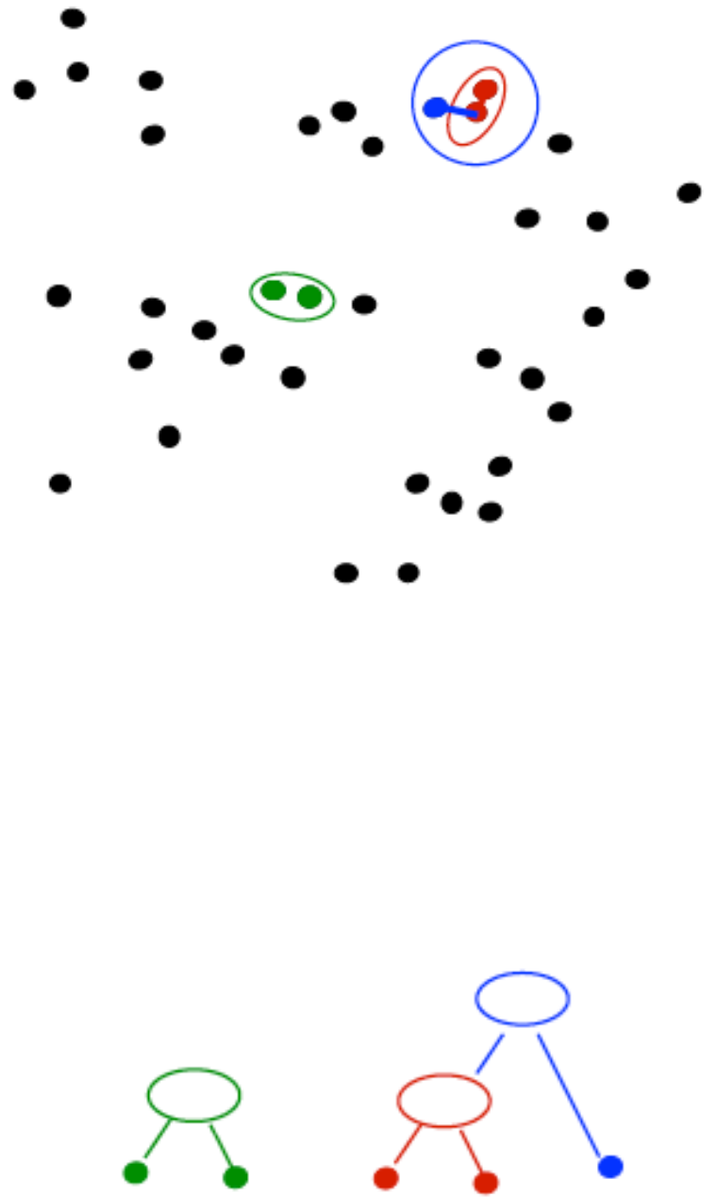
  - Hamming distance

$$\mathrm{Hamming}(\boldsymbol{x}_1, \boldsymbol{x}_2) = d - \sum_{i=1}^{d} \mathrm{I}(x_{1i}, x_{2i})$$

- 4. Large-scale issues (Exercise)
  - How to speed up K-means on large-scale datasets with a large K?
    - Think about which key factors introduces computational complexity in the algorithms.

Hierarchical Clustering

- **Agglomerative clustering**
  - First merge very similar instances
  - Incrementally build larger clusters based on small ones
- Algorithm
  - 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 results, but a family of clusterings represented by a dendrogram



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

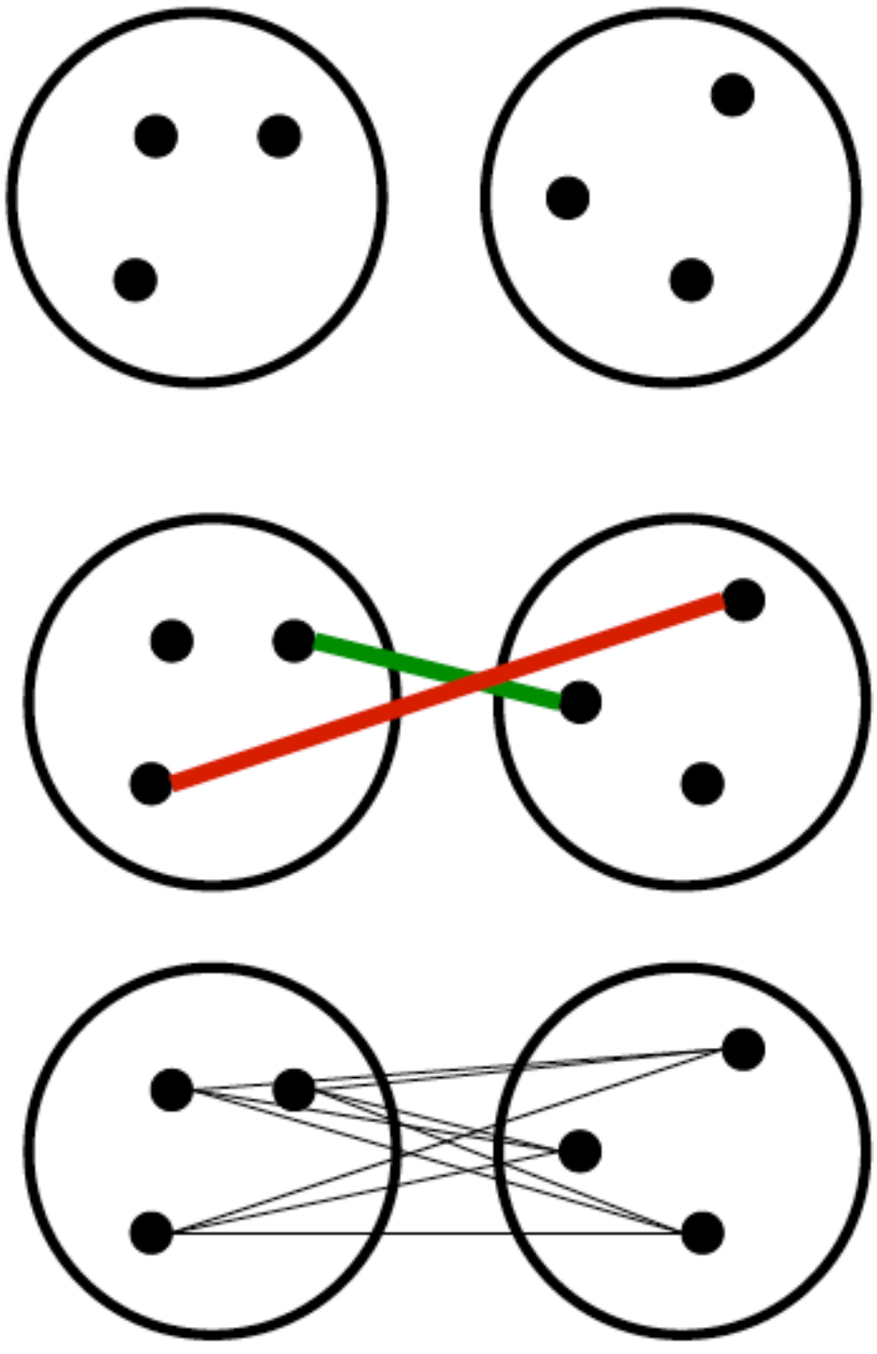
- Many options:

  - Closest pair
    (single-link clustering)

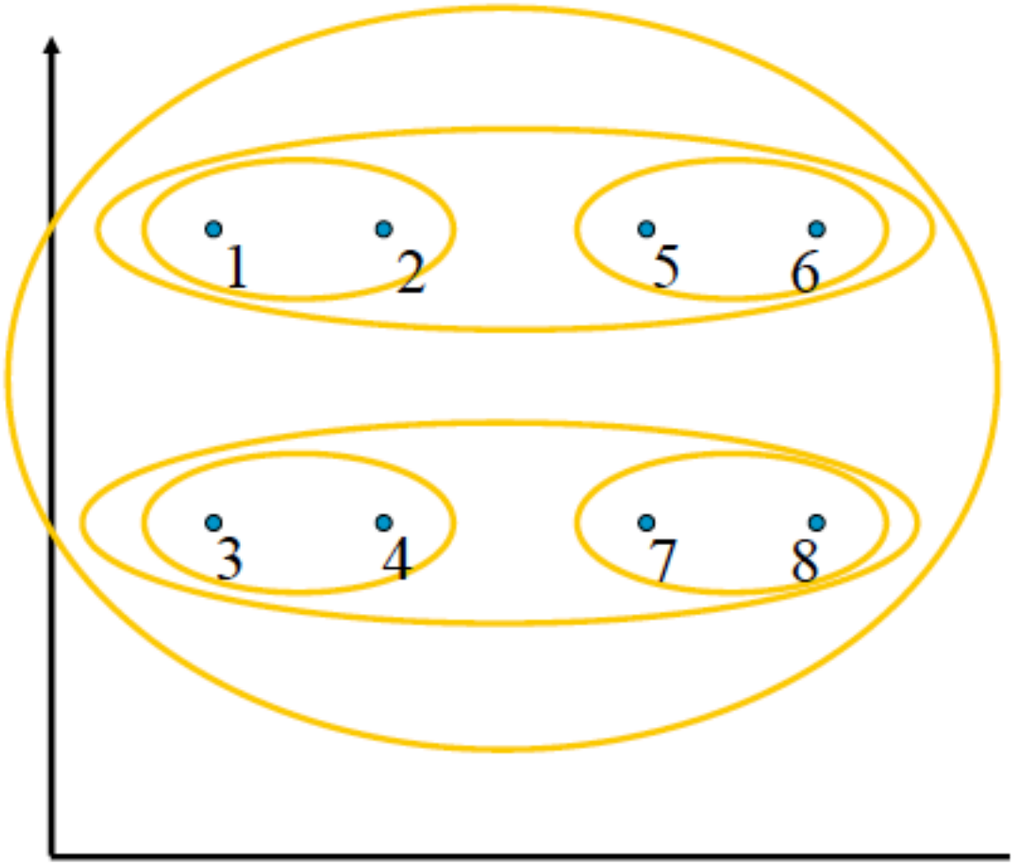
  - Farthest pair
    (complete-link clustering)

  - Average of all pairs

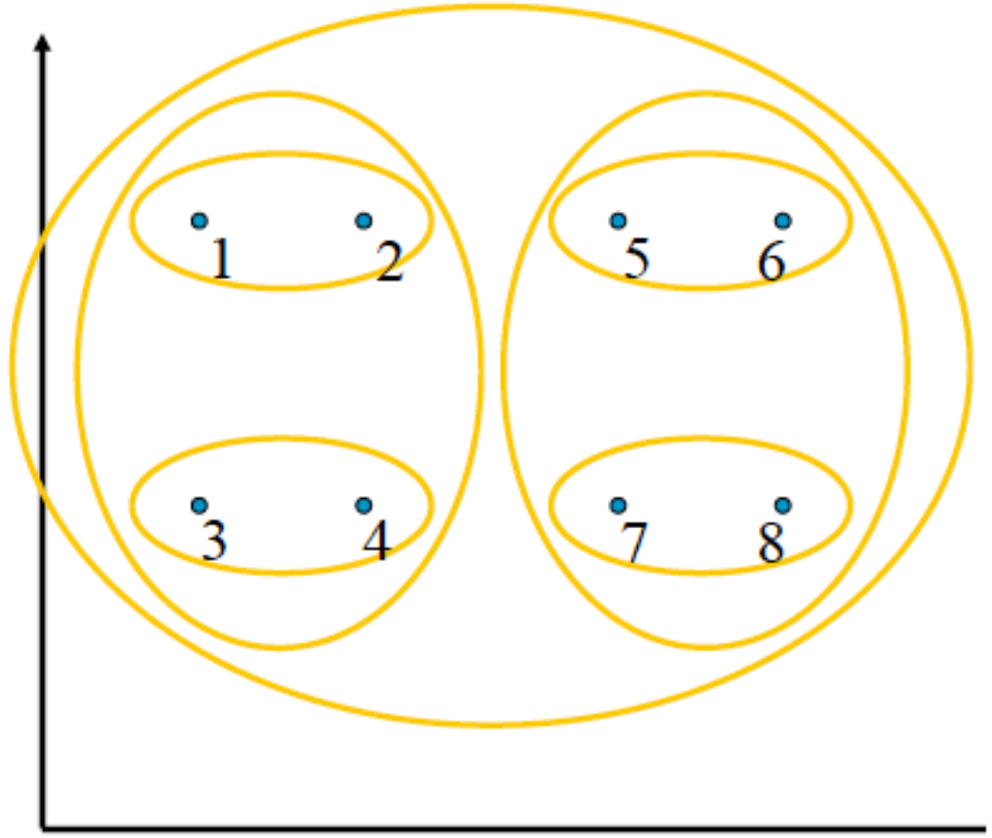
- Different choices produce different clustering behaviors



Closest pair
(single-link clustering)



Farthest pair
(complete-link clustering)

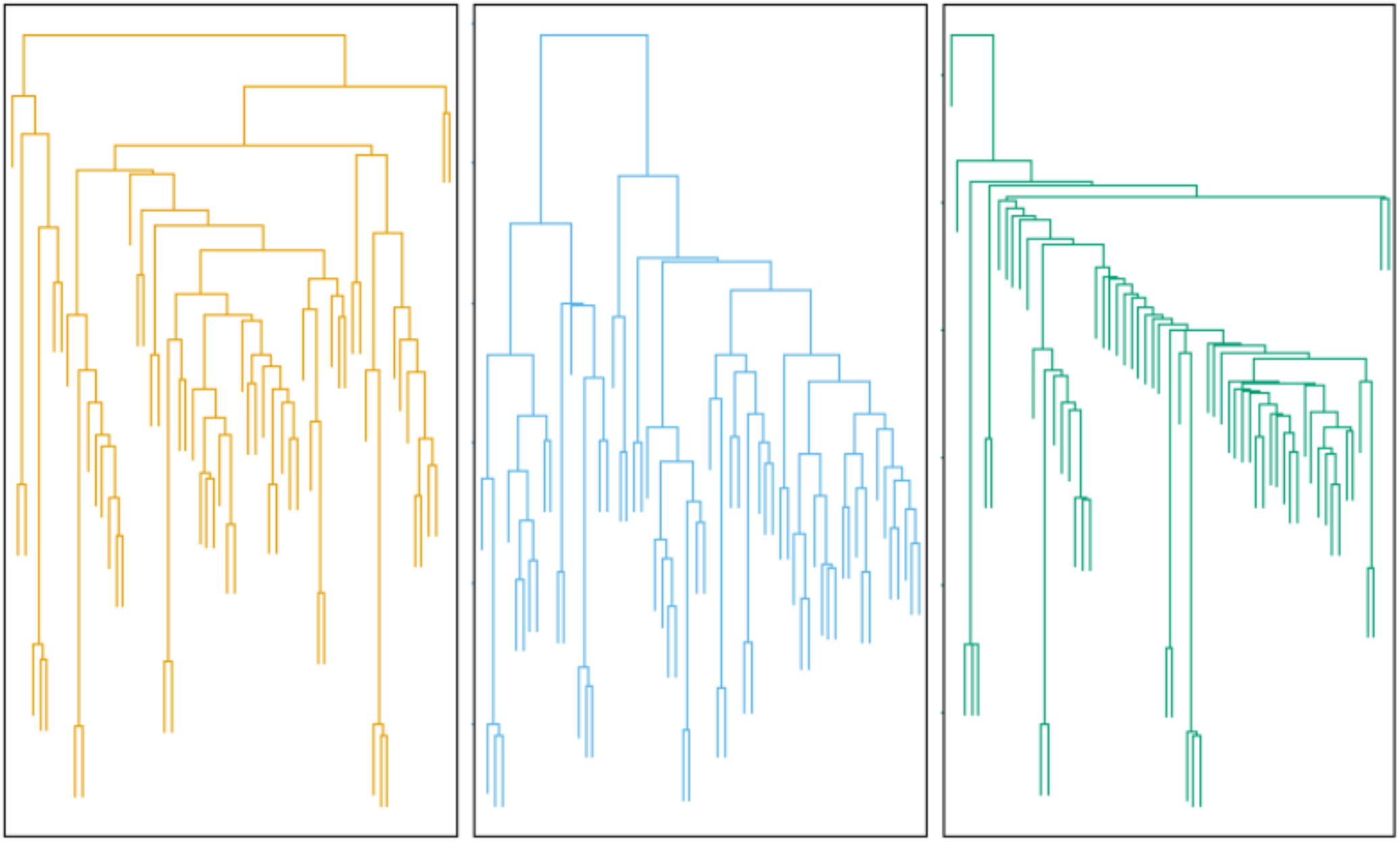


[Pictures from Thorsten Joachims]

Average

Farthest

Nearest

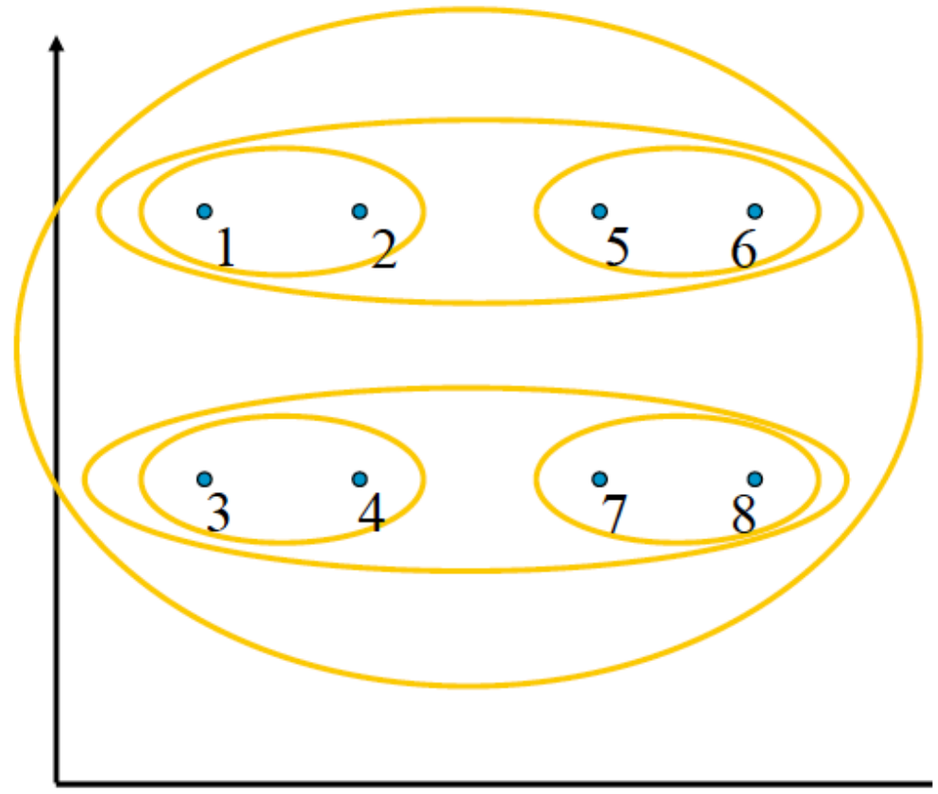


Mouse tumor data from [Hastie *et al.*]

When can this be expected to work ?

- 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

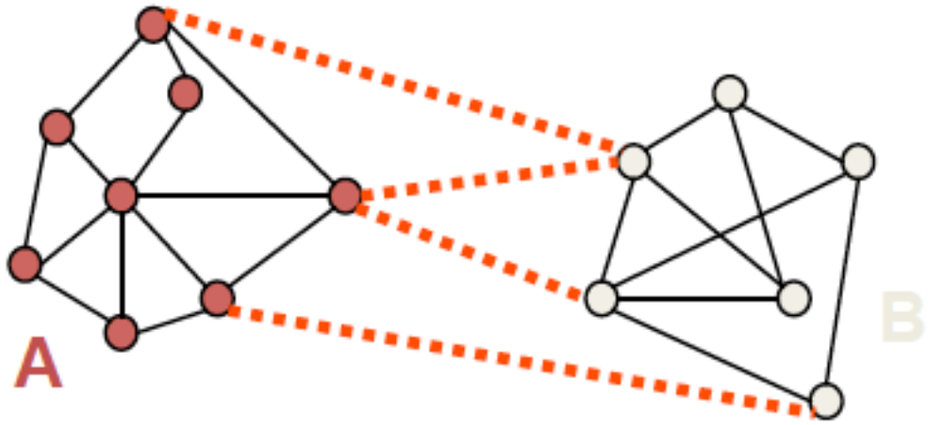
Closest pair
(single-link clustering)

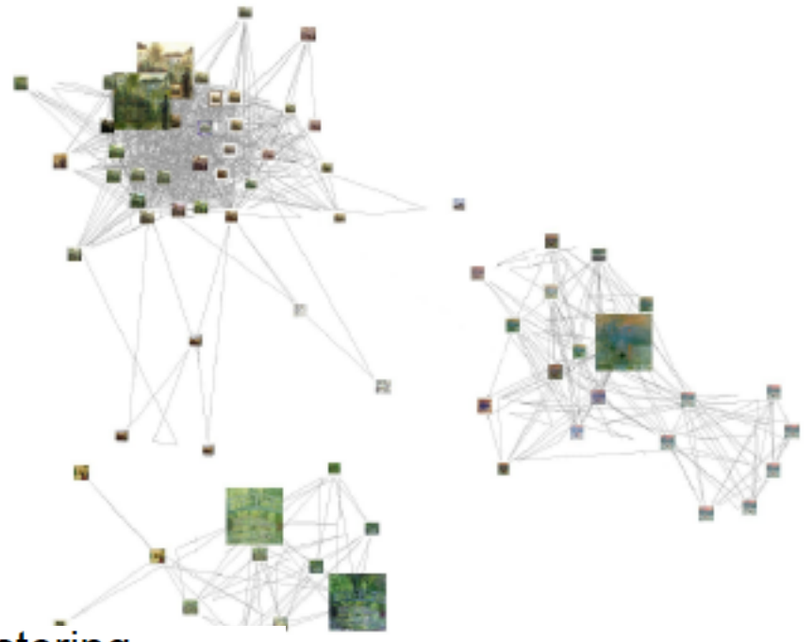


Balcan et. al. 2018

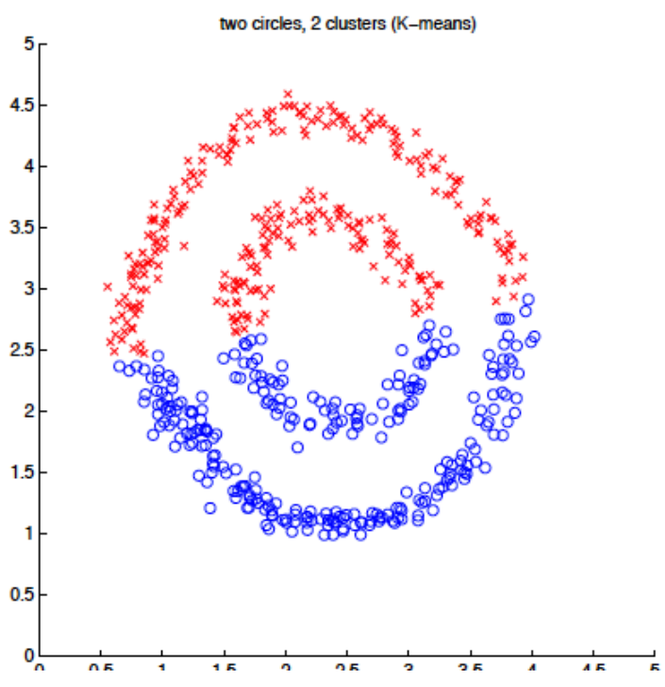
Spectral Clustering

- **Grouping points based on links in a graph**

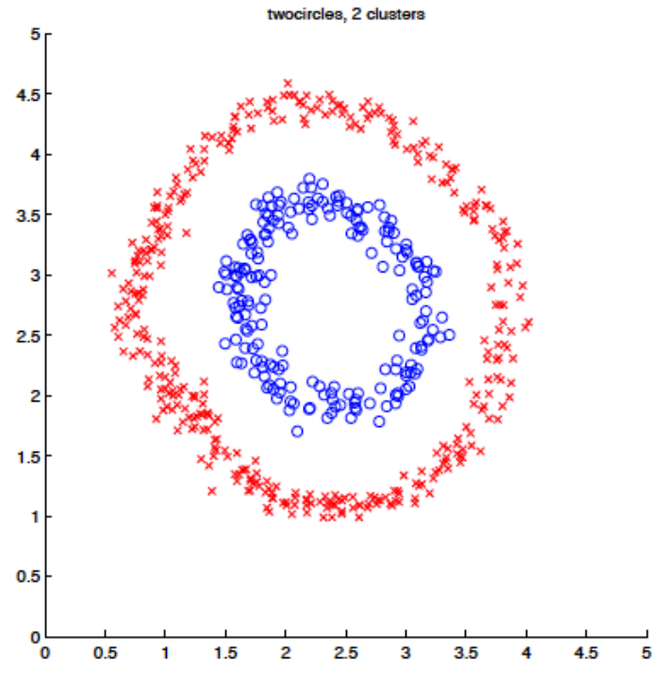




K-means



Spectral clustering



- How to construct the graph?
  - Typically using Gaussian kernel

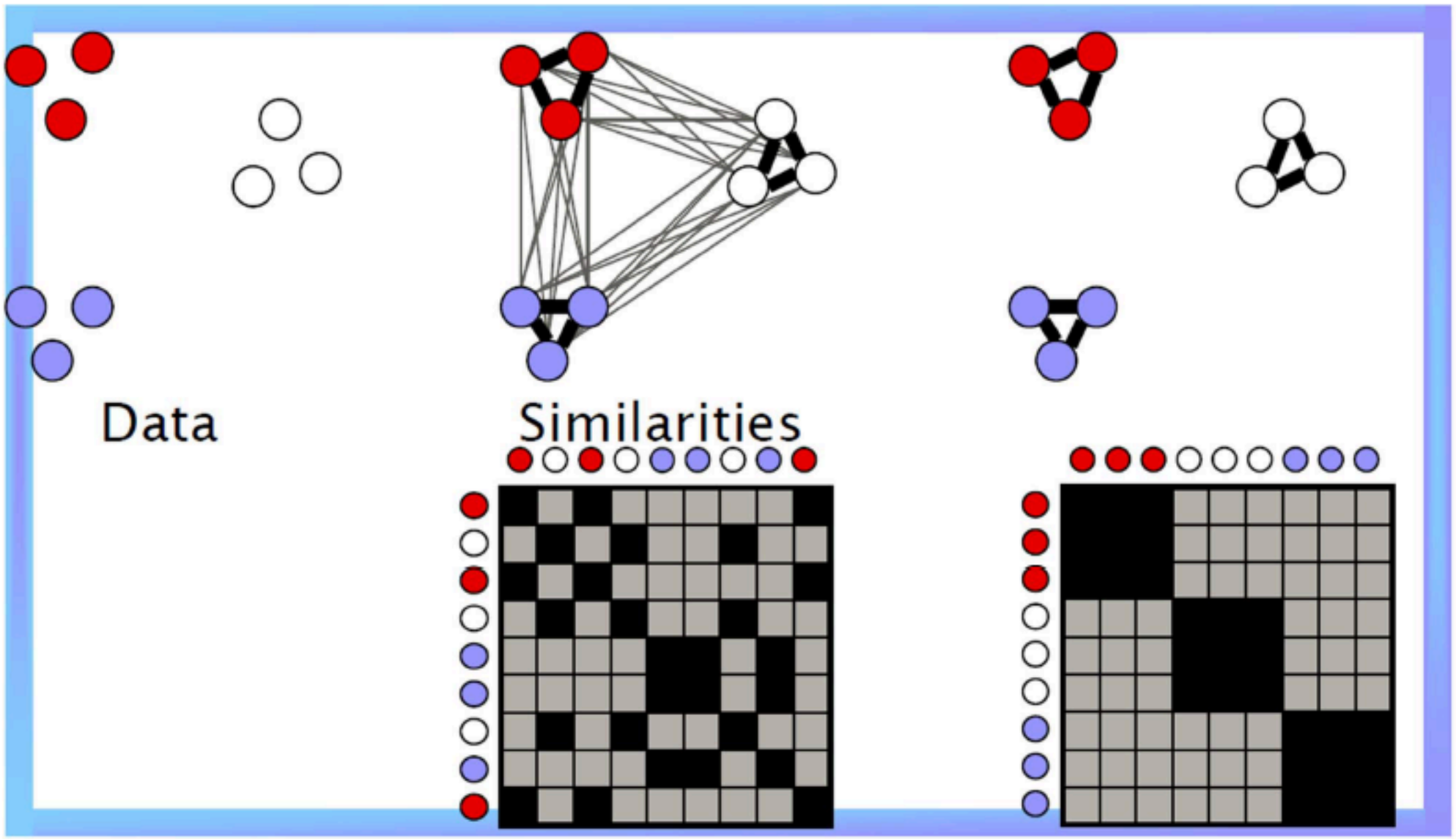
$$W(i,j) = \exp\frac{-\left|x_i - x_j\right|^2}{\sigma^2}$$

  - Different types of graphs
    - A fully connected graph
    - KNN graph (each node is only connected to its k-nearest neighbours)

Spectral clustering for segmentation

[Slide from James Hays]

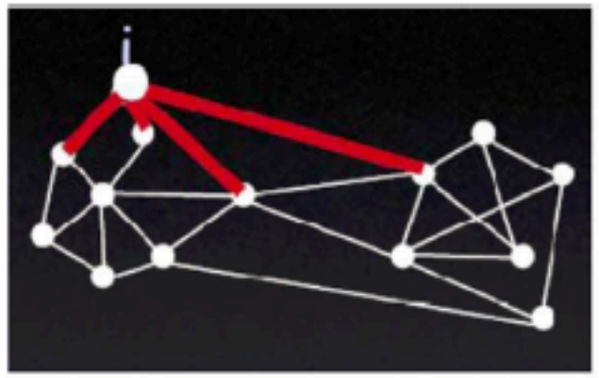
- **Graph partitioning**

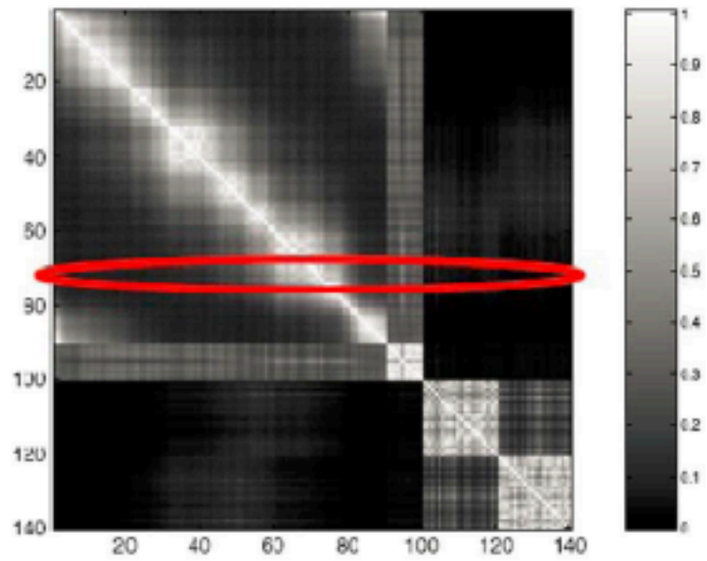


- **Graph terminology**

  - Degree of nodes

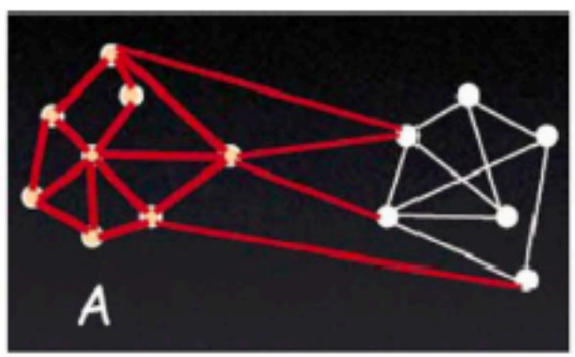
$$d_i = \sum_j w_{i,j}$$

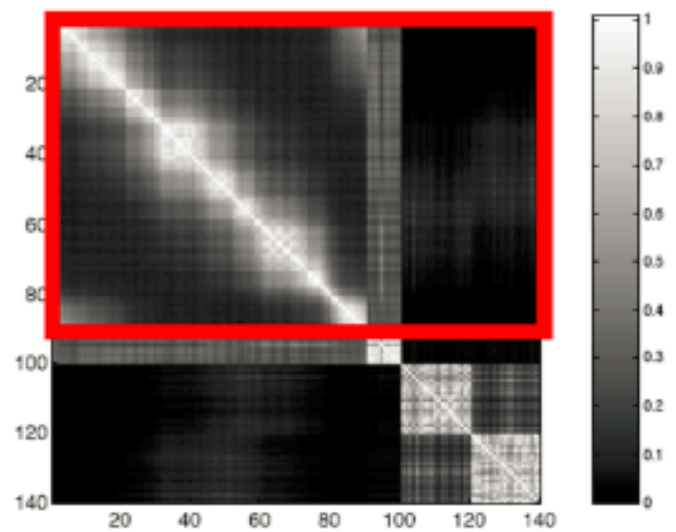




  - Volume of a set

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

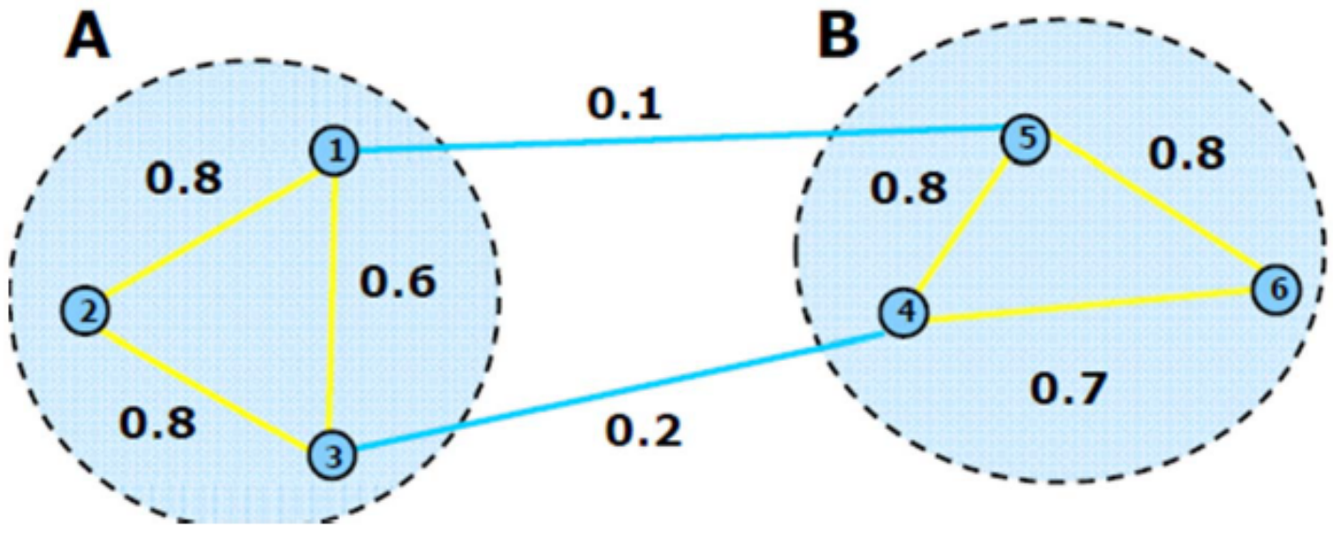




- **Graph Cut**
  - Consider “cut” the graph into two parts
  - ***Cut(A, B)***: sum of the weights of the set of edges that connect the two groups

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

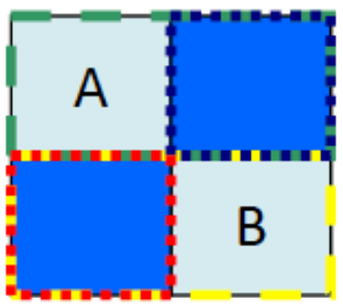
  - An intuitive goal is to minimize the cut for clustering



- **Normalized Cut**

  - The idea: 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

- **Normalized Cut**

  - After some simplifications, we have

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

$y = \{-1, 1\}^N$

Rayleigh quotient

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

NP-hard problem

- **Solving Ncut problem**

  - Relax the original NP-hard problem into continuous domain, and formulate it as a **generalized eigenvalue problem:**

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

$$\downarrow$$

$$(D-W)y = \lambda Dy$$

  - Obviously, the smallest eigenvalue is zero with eigenvector as the vector with all ones elements

  - **The 2nd smallest eigenvector** is the real valued solution to this problem!

- **The algorithm for Ncut:**

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.

- Create 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?

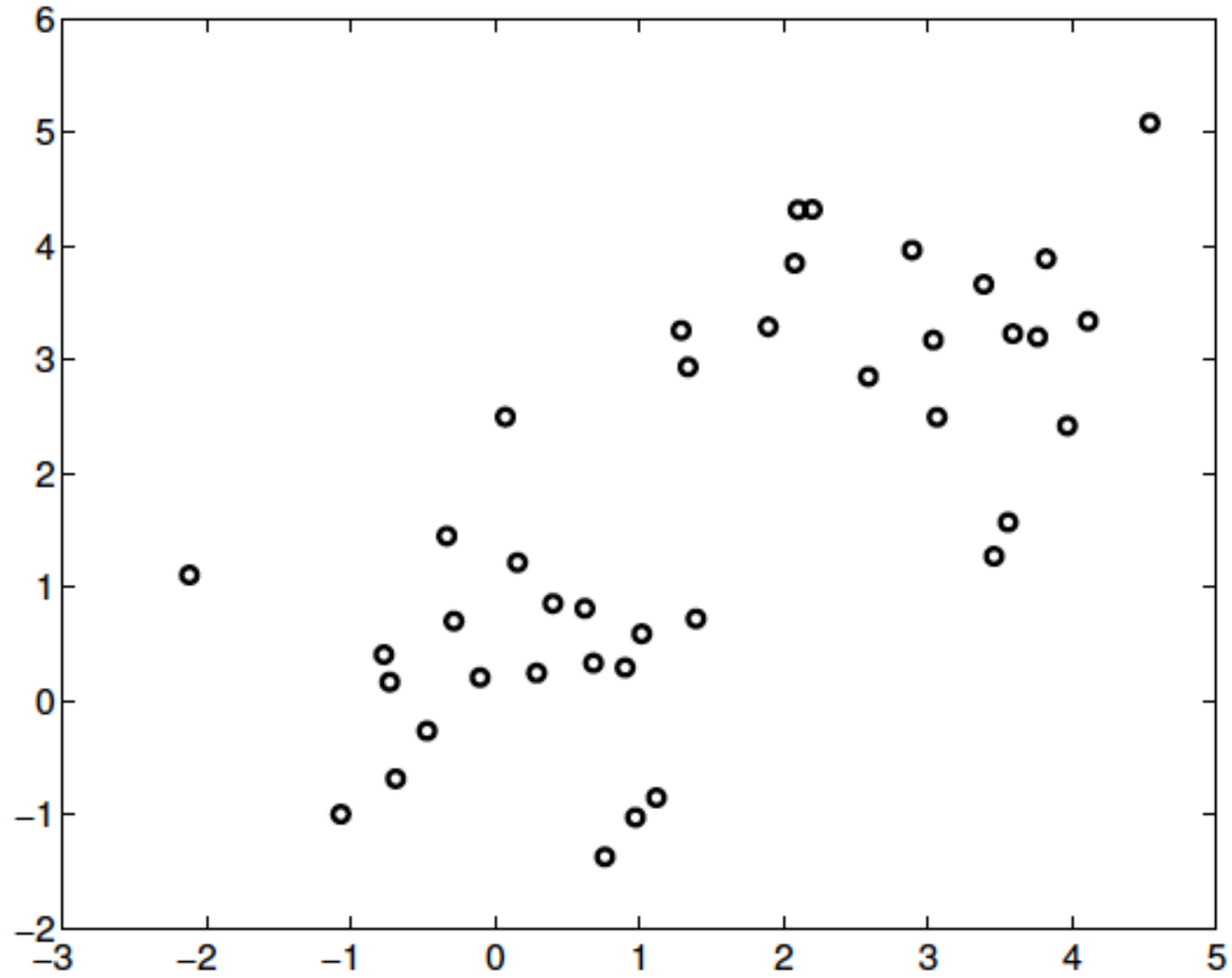
    a) Pick a constant value (0, or 0.5).

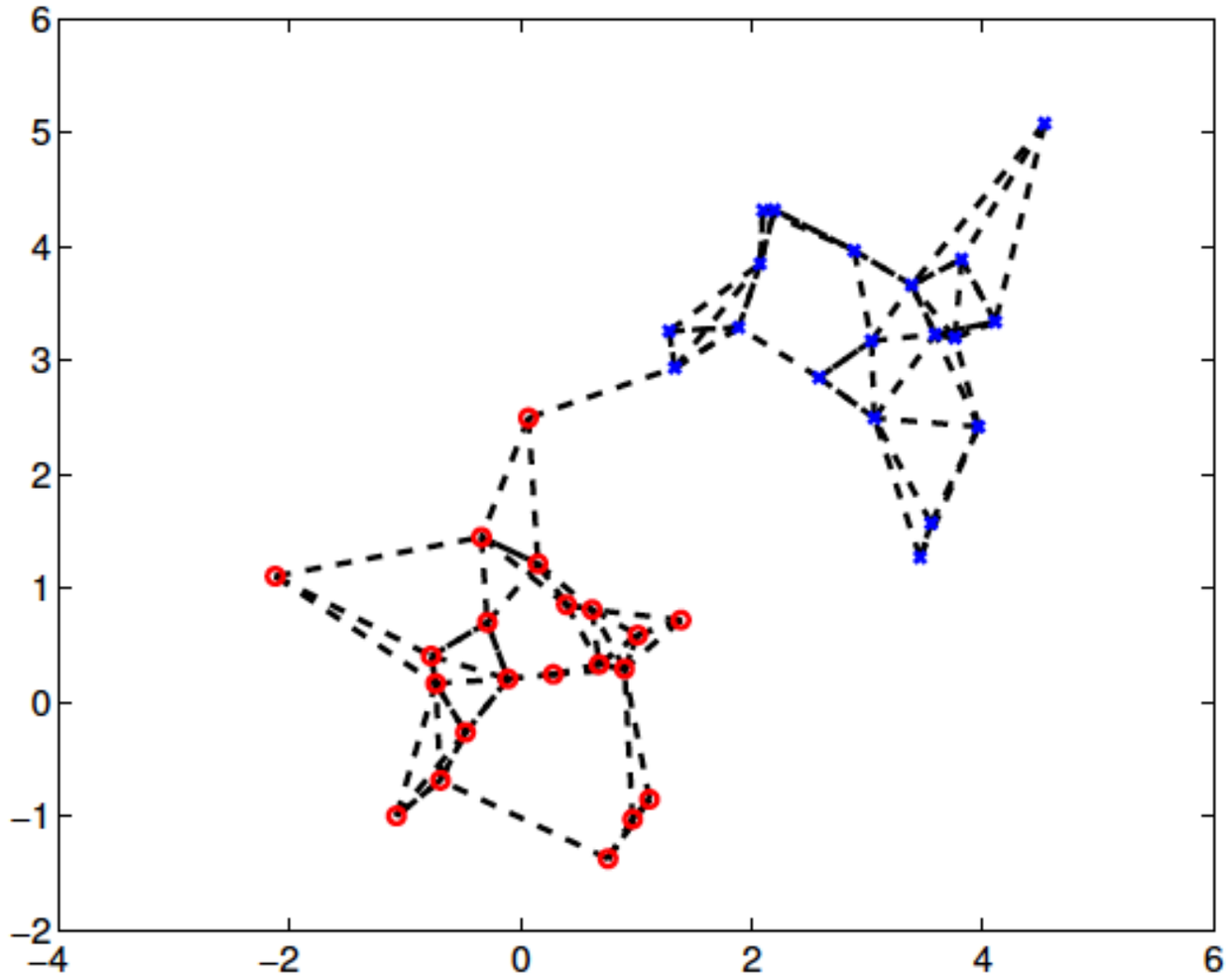
    b) Pick the median value as splitting point.

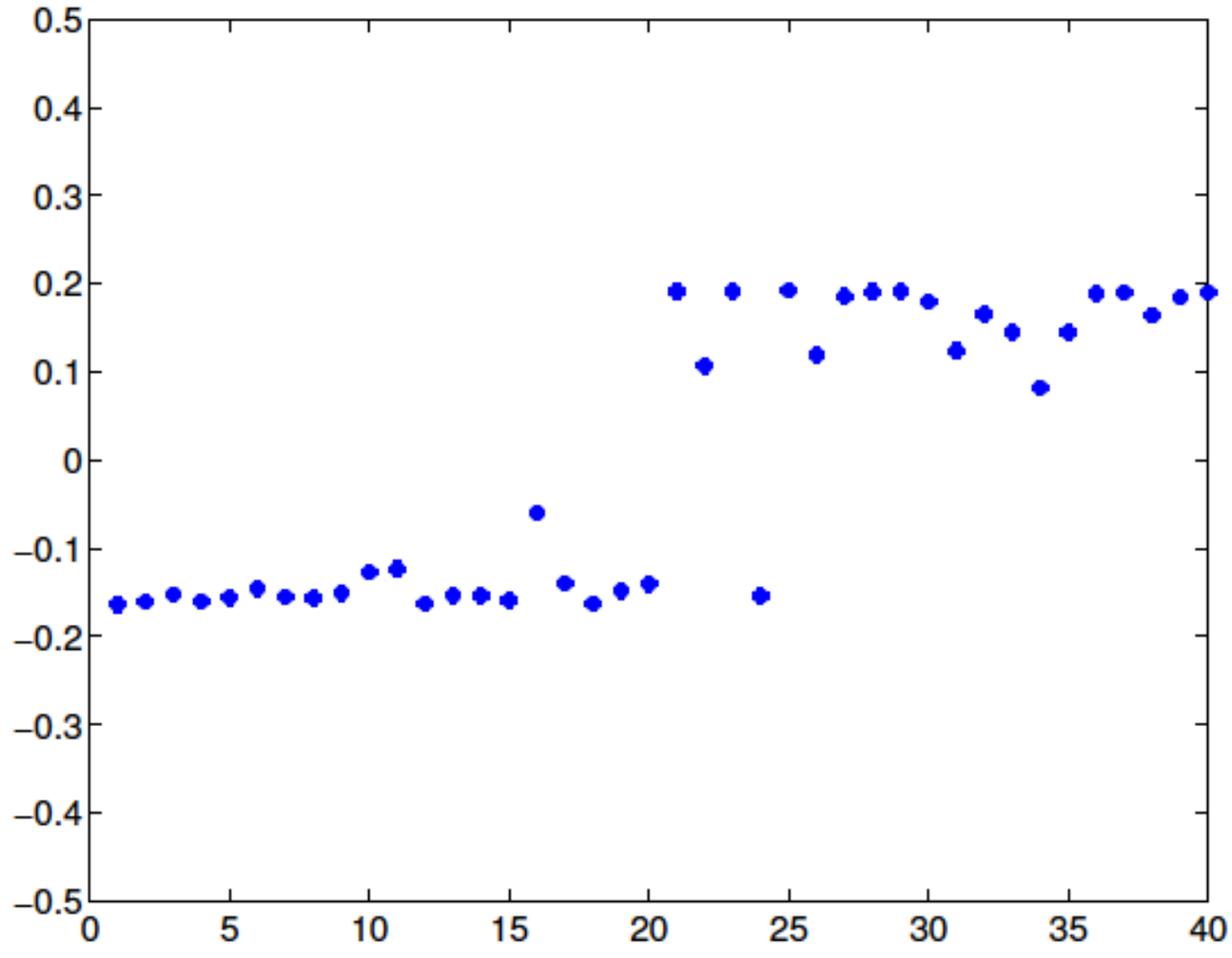
    c) Look for the splitting point that has the minimum $Ncut$ value:

       1. Choose n possible splitting points.
       2. Compute $Ncut$ value.
       3. Pick minimum.

Spectral clustering: an example





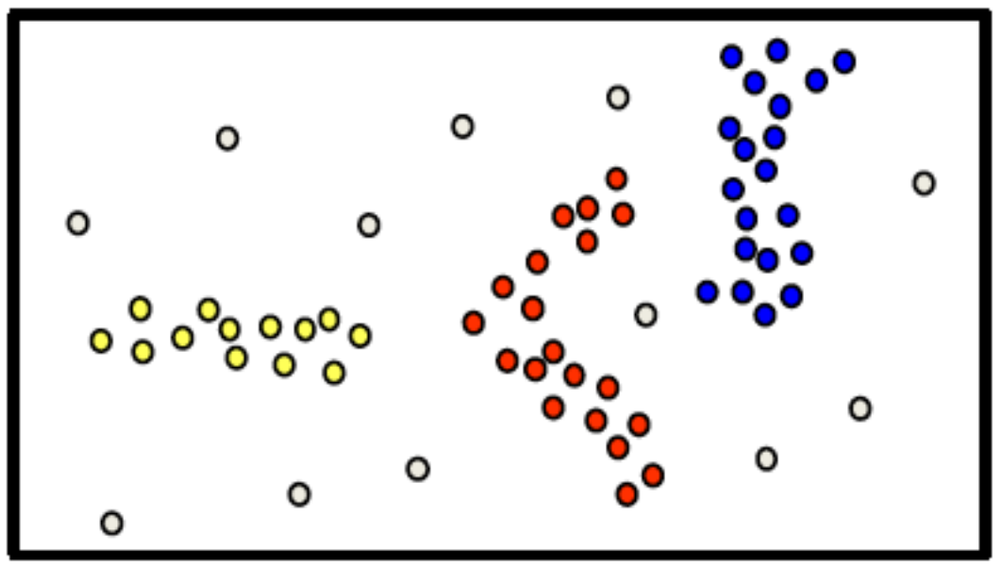


Components of the eigenvector corresponding to the second largest eigenvalue

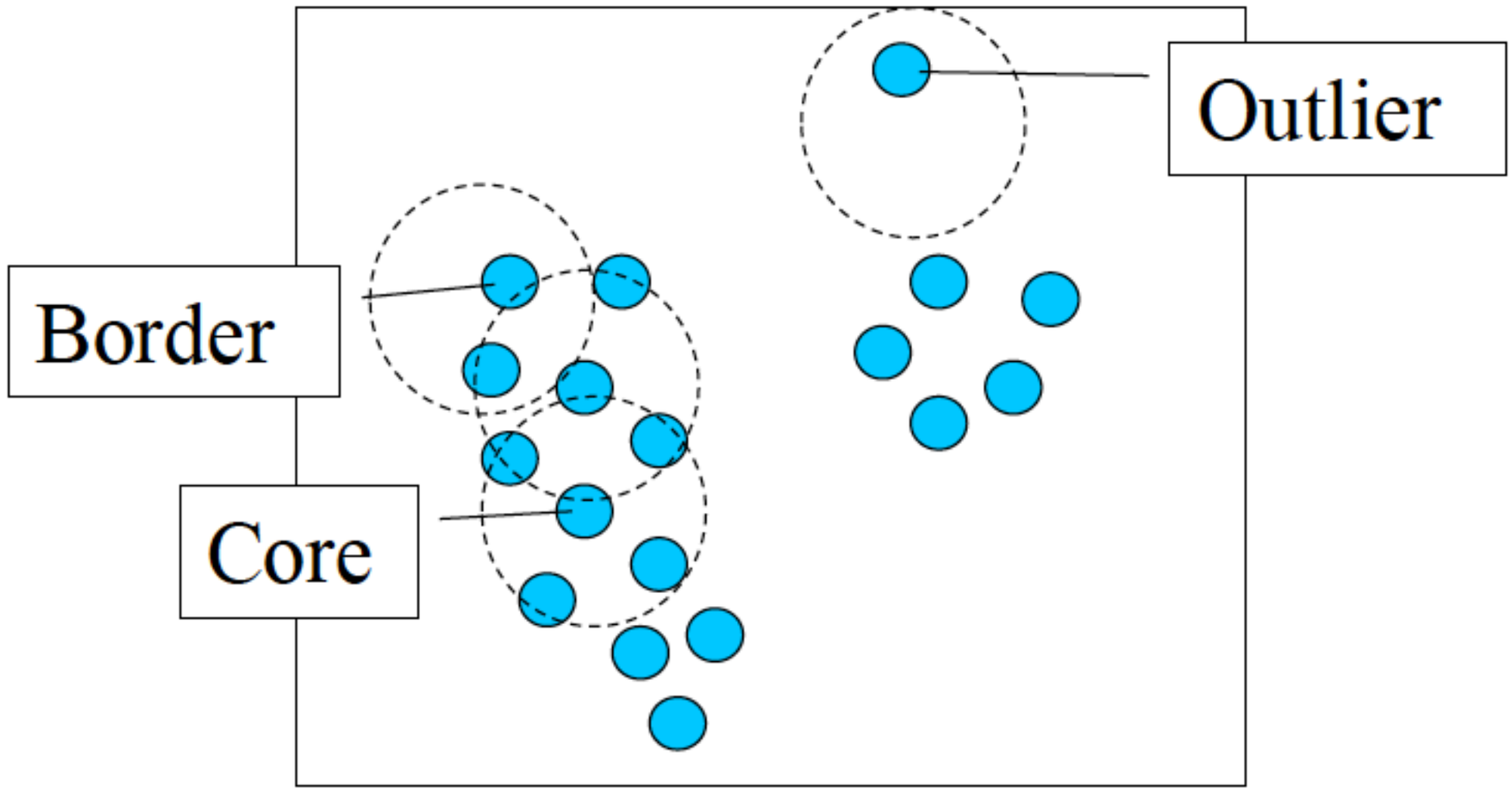
- **K-partitioning**
  - Recursive bi-partitioning (Hagen et al.,'91)
    - Recursively apply bi-partitioning algorithm in a hierarchical divisive manner.
    - Disadvantages: Inefficient, unstable
  - Cluster multiple eigenvectors
    - Build a reduced space from multiple eigenvectors.
    - Commonly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

Density-based clustering

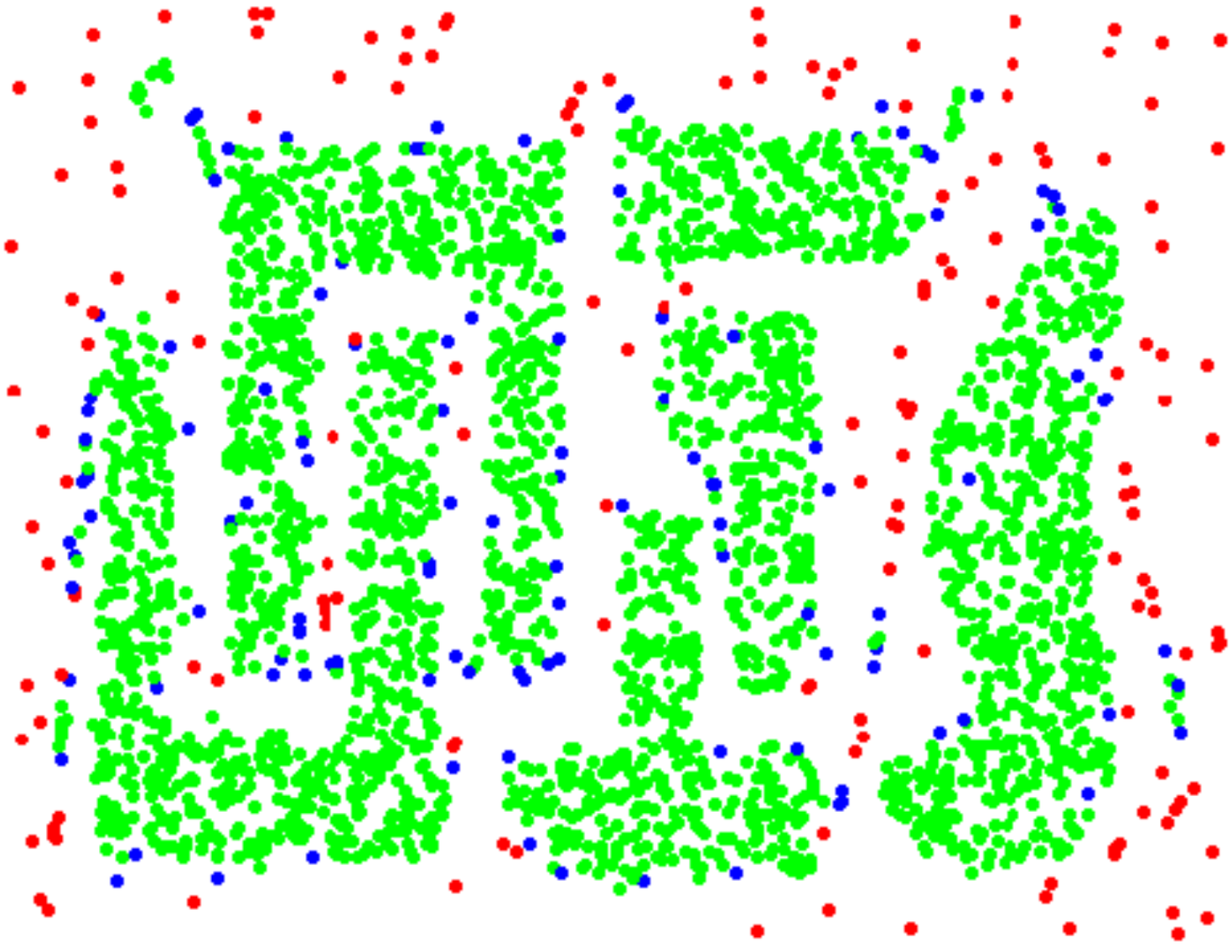
- The idea
  - Clusters are dense regions separated by low density areas
  - A cluster is defined as a maximal set of density-connected points
- Able to discover clusters of arbitrary shape



- DBSCAN (Density-Based Spatial Clustering of Application with Noise)
  - Some important definitions
    - $\epsilon - \text{neighborhood}$ objects within a radius of epsilon from an object. $N_\varepsilon(p): \{q \mid d(p,q) \le \varepsilon\}$
      - Core object: has more than *MinPts* with epsilon radius.
      - Border object: has fewer than MinPts within epsilon, but in the neighborhood of a core object.
      - Outlier/noise: not a core object or a border object



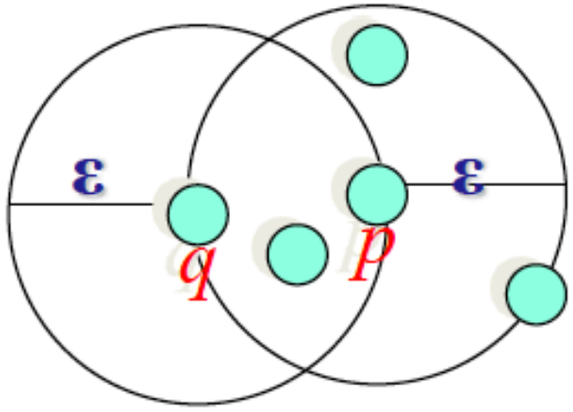
$\varepsilon = 1\text{unit}, \text{MinPts} = 5$



epsilon = 10, MinPts = 4

- **Directly density-reachable**

  - An object q is directly density-reachable from object p if p is a core object and q is in p's epsilon-neigbhorhood:



MinPts = 4

- q is directly density-reachable from p
- p is not directly density-reachable from q
- Density-reachability is asymmetric

- Density-reachable

  - Density-reachable:

    → A point p is *density-reachable* from a point q wrt. Eps, MinPts if there is a chain of points $p_1, \ldots, p_n$, with $p_1 = q, p_n = p, s.t. p_{i+1}$ is directly density reachable from p_i

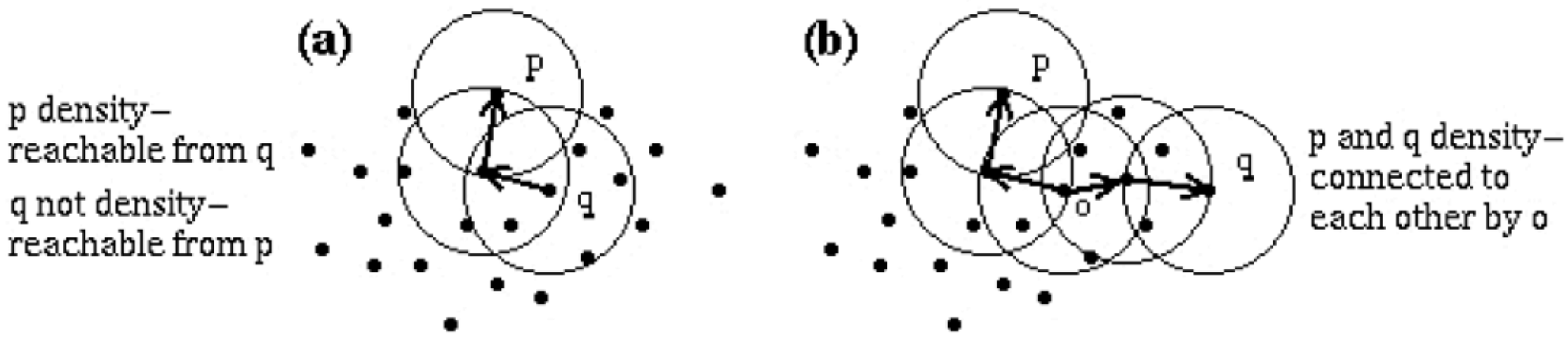
  - transitive but not symmetric

- Density-connected

Density-connected:

→ A point p is *density-connected* from a point q wrt. Eps, MinPts if there is a point *o* s.t. p and q are density-reachable from *o* wrt. Eps and MinPts

→ not symmetric



- DBSCAN

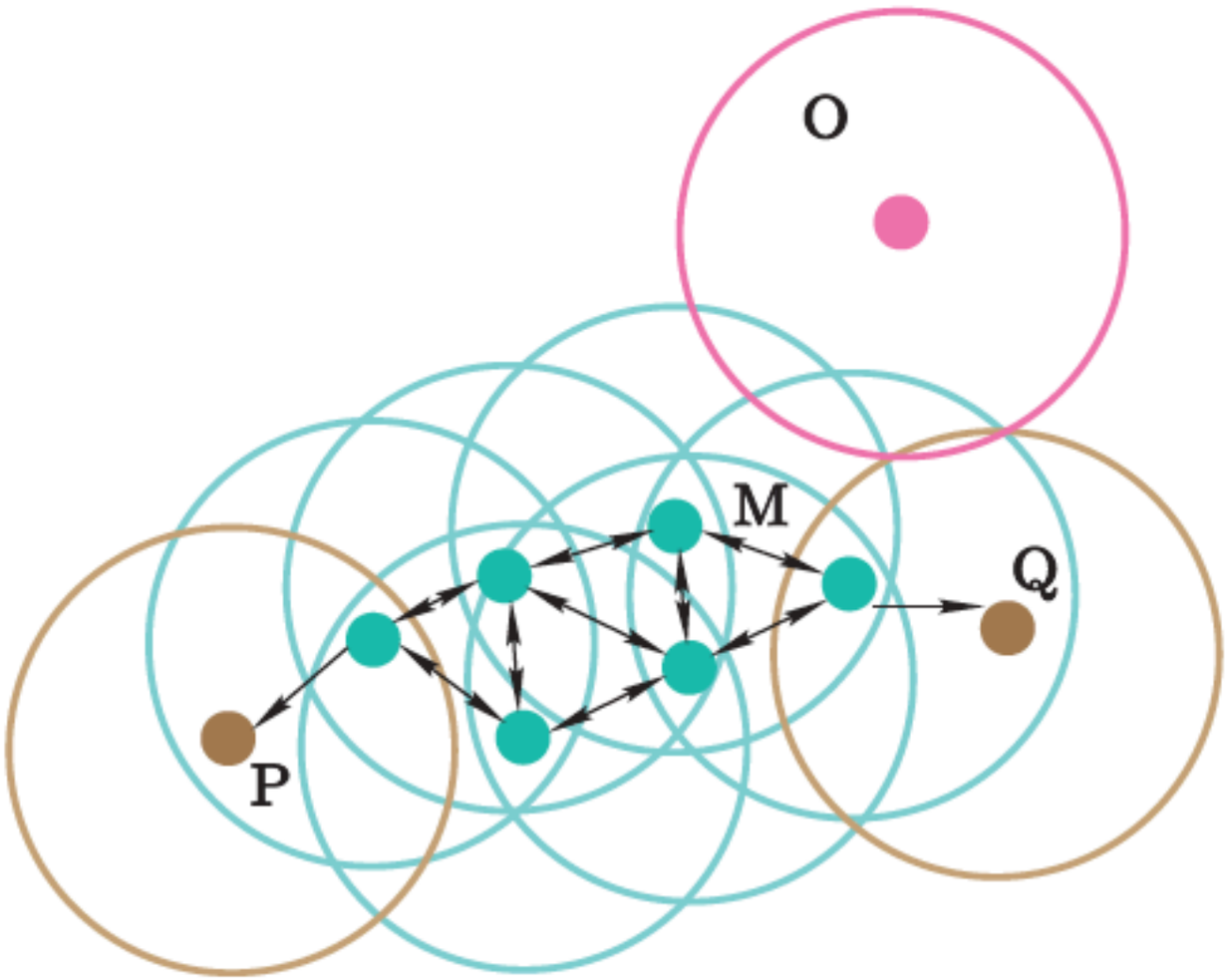
Main principle:

a cluster is defined as a maximal set of density-connected points.

- Discovers clusters of arbitrary shapes (spherical, elongated, linear), and noise
- Works with spatial datasets:
  - $\rightarrow$ geomarketing, tomography, satellite images
- Requires only two parameters (no prior knowledge of the number of clusters)

- The Algorithm

1. Randomly select a point *p*
2. Retrieve all points density-reachable from *p* wrt. *Eps* and *MinPts*
3. If *p* is a core point, a cluster is formed
4. If *p* is a border point, then no points are density-reachable from $p \rightarrow$ visit the next data point
5. Continue the process until all points have been processed


O
M
Q
P

Pros:

- ✓ discovers clusters of arbitrary shapes
- ✓ handles noise
- ✓ needs density parameters as termination condition

Cons:

- *X* cannot handle varying densities
- *X* sensitive to parameters → hard to determine the correct set of parameters
- *X* sampling affects density measures

Read the original DBSCAN paper for how to determine the hyper-parameters (exercise):

Ester, Martin; Kriegel, Hans-Peter; Sander, Jörg; Xu, Xiaowei (1996). *A density-based algorithm for discovering clusters in large spatial databases with noise*. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press. pp. 226–231.

Additional reading:

Rodriguez, Alex, and Alessandro Laio. "Clustering by fast search and find of density peaks." *Science* 344.6191 (2014): 1492-1496.

Evaluation of clustering results

- **Purity**: characterizing the percentage of correctly clustered data points

$$n_i = \sum_{j=1}^{C} n_{ij}$$

number of points assigned to cluster i but belong to true class j

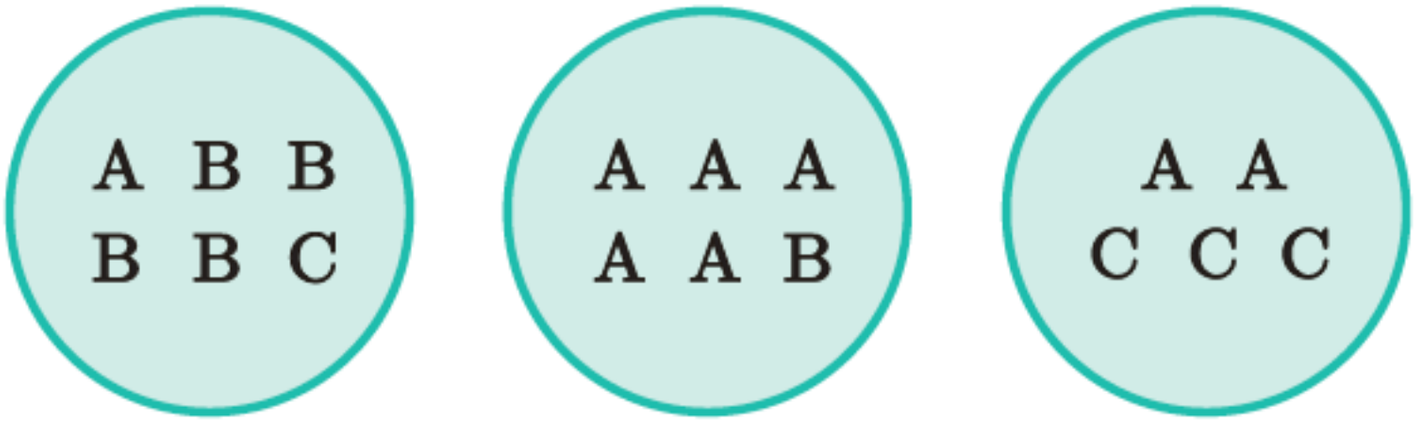
$$p_{ij} = \frac{n_{ij}}{n_i}$$

purity for cluster i

$$p_i \triangleq \max_j p_{ij}$$

$$\text{purity} = \sum_i \frac{n_i}{n} p_i$$

[0,1], the larger, the better



$$\text{purity} = \frac{6}{17} \times \frac{4}{6} + \frac{6}{17} \times \frac{5}{6} + \frac{5}{17} \times \frac{3}{5} = 0.71$$

- **Rand index**: measure the similarity between two clusterings

$$U = \{u_1, \ldots, u_R\} \qquad V = \{v_1, v_2, \cdots, v_C\}$$

$$\mathrm{RI} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{FP} + \mathrm{FN} + \mathrm{TN}}$$

  - Compute rand index (Exercise)

- **Mutual information**

$$p_{UV}(i,j) = \frac{|u_i \cap v_j|}{n} \qquad p_U(i) = \frac{|u_i|}{n} \qquad p_V(j) = \frac{|v_j|}{n}$$

$$\mathrm{MI}(U,V) = \sum_{i=1}^{R}\sum_{j=1}^{C} p_{UV}(i,j) \ln \frac{p_{UV}(i,j)}{p_U(i)p_V(j)} \qquad \mathrm{NMI}(U,V) = \frac{\mathrm{MI}(U,V)}{(H(U)+H(V))/2}$$

Exercise

- Implement K-means and DBSCAN from scratch
- How to speed up K-means with large-scale data and big K?
- Optional readings
  - Tutorials on spectral clustering
    https://arxiv.org/abs/0711.0189
  - A more recent density-based clustering
    Rodriguez, Alex, and Alessandro Laio. "Clustering by fast search and find of density peaks." *Science* 344.6191 (2014): 1492-1496.