Clustering

Enikő Székely

Computer Science Department
University of Montpellier II

March 22, 2012

Outline

Outline

Data mining

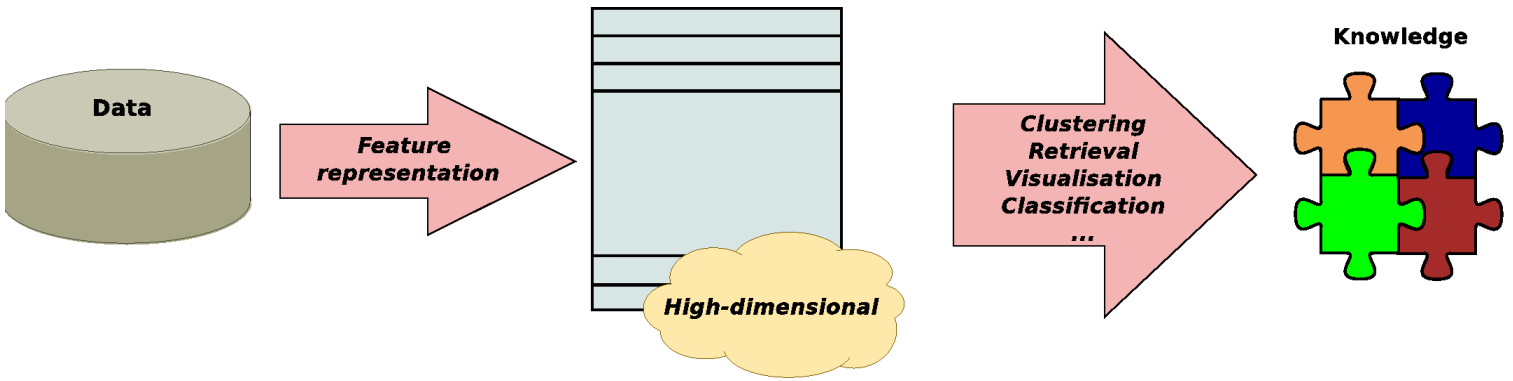


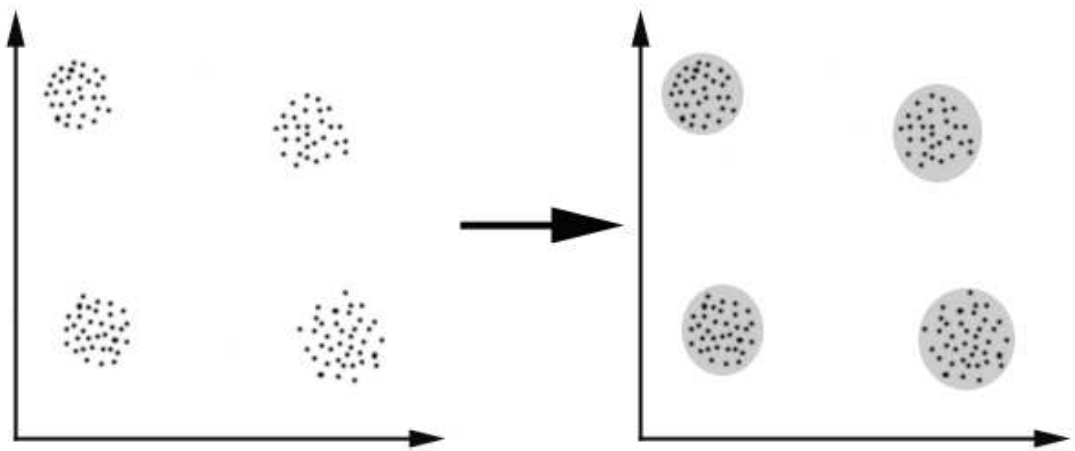
Figure: The process of data mining.

$\mathbf{X} \subset \mathbb{R}^{N \times D}$ - N data points represented in a D-dimensional space (feature/attribute representation).
$\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N\}$

Clustering

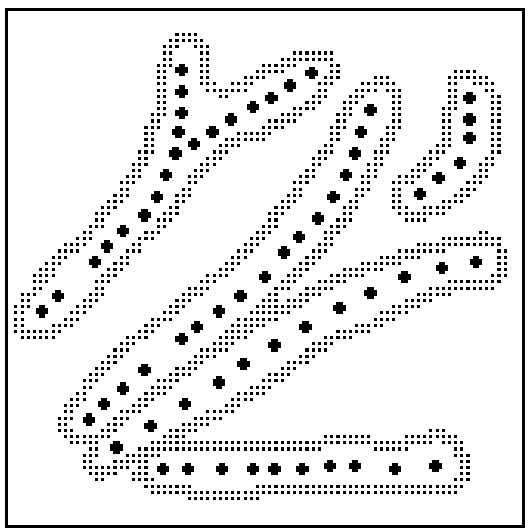
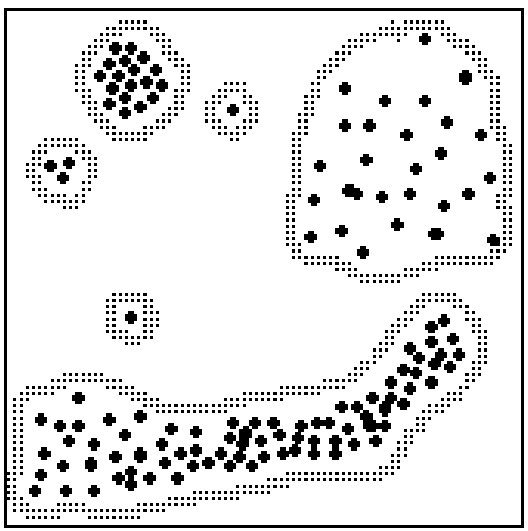
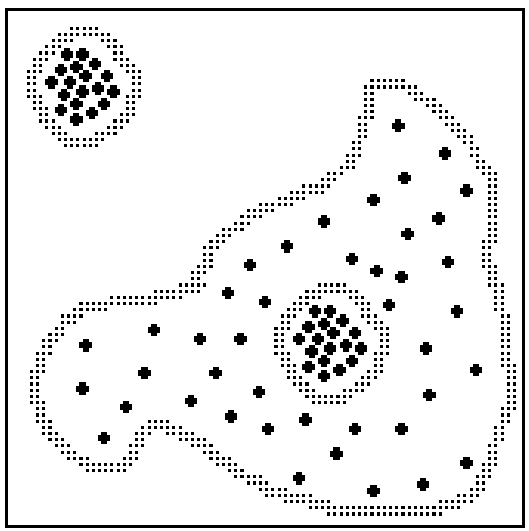
Clustering is the process of separating data into groups called *clusters* so that:

- similar items belong to the same cluster;
- dissimilar items belong to different clusters.



Applications of clustering

- Text mining
- Medicine
- Information retrieval
- Marketing / consumer insights
- Environmental studies (climatology, meteorology)
- ...

Supervised vs. unsupervised learning

Supervised learning: the labels / classes of the data are known.

- goal: predict the label of a new testing data item using information derived from the training data
- *classification*
- eg. handwriting recognition, document classification, medical imaging etc.

Unsupervised learning: no a priori information about the data is known

- goal: discover new patterns that were not known in advance
- *clustering*
- eg. social network analysis, medical imaging, image segmentation etc.

Similarity/dissimilarity measures

Definition

A **similarity/dissimilarity measure** *is a measure used to define how similar/dissimilar two data points/items/documents are.*

Measures:

- distances (Euclidean, Minkowski, Mahalanobis etc.)
- cosine similarity
- tf-idf (term frequency-inverse document frequency)
- ...

Distances

Definition

A distance function or a metric is a measure of the distance between two points.

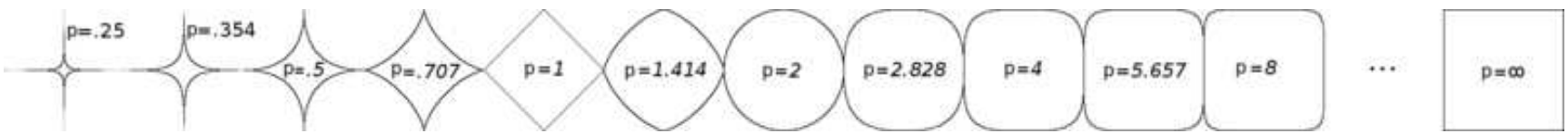
A function is called a metric if it satisfies the following four conditions:

1. $d_{ij} \geq 0$ (positiveness)
2. $d_{ij} = 0 \Leftrightarrow \mathbf{x}_i = \mathbf{x}_j$ (identity)
3. $d_{ij} = d_{ji}$ (symmetry)
4. $d_{ih} \leq d_{ij} + d_{jh}$ (triangle inequality)

Minkowski distances

$$L_p(\mathbf{x}_i, \mathbf{x}_j) = \left[\sum_{l=1}^{D} |x_i^l - x_j^l|^p\right]^{\frac{1}{p}}, p = 1, 2, .. \quad (1)$$

p	**Name**	**Distance**
1	Manhattan	$\sum_{l=1}^{D} \|x_i^l - x_j^l\|$
2	Euclidean	$\left[\sum_{l=1}^{D} \|x_i^l - x_j^l\|^2\right]^{\frac{1}{2}}$
∞	Chebyshev	$\lim_{p\to\infty} \left[\sum_{l=1}^{D} \|x_i^l - x_j^l\|^p\right]^{\frac{1}{p}} = \max_l \|x_i^l - x_j^l\|$



Outline

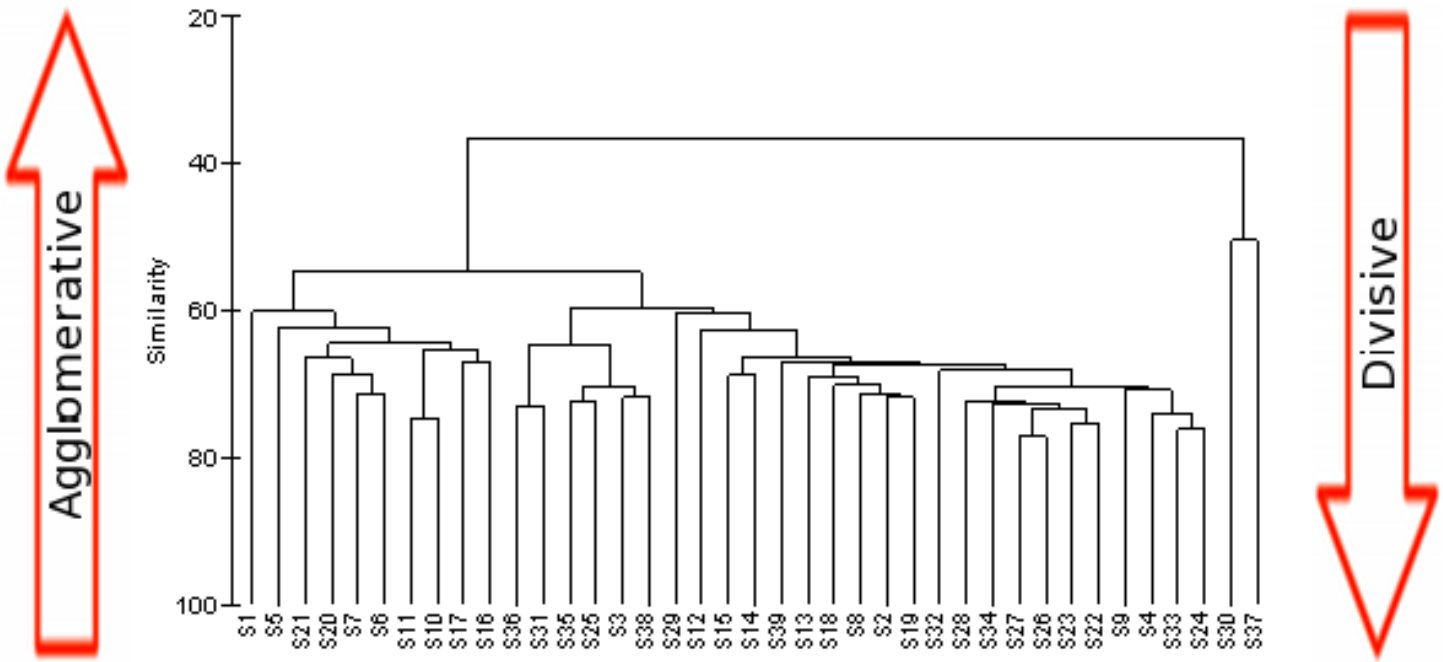
Clustering methods

Two basic types of clustering methods:

1. Hierarchical
   - create a hierarchy of clusters called a dendogram
   - either merges small clusters into bigger ones or splits big clusters into smaller ones
   - requires only a similarity matrix as input
   - various clusterings can be obtained depending on the cutoff point in the tree
2. Partitional
   - partition the data into different clusters by doing either a hard or a soft assignment
   - often requires the number of clusters K as input

Hierarchical clustering

1. Agglomerative (bottom-up)
2. Divisive (top-down)



Agglomerative hierarchical clustering

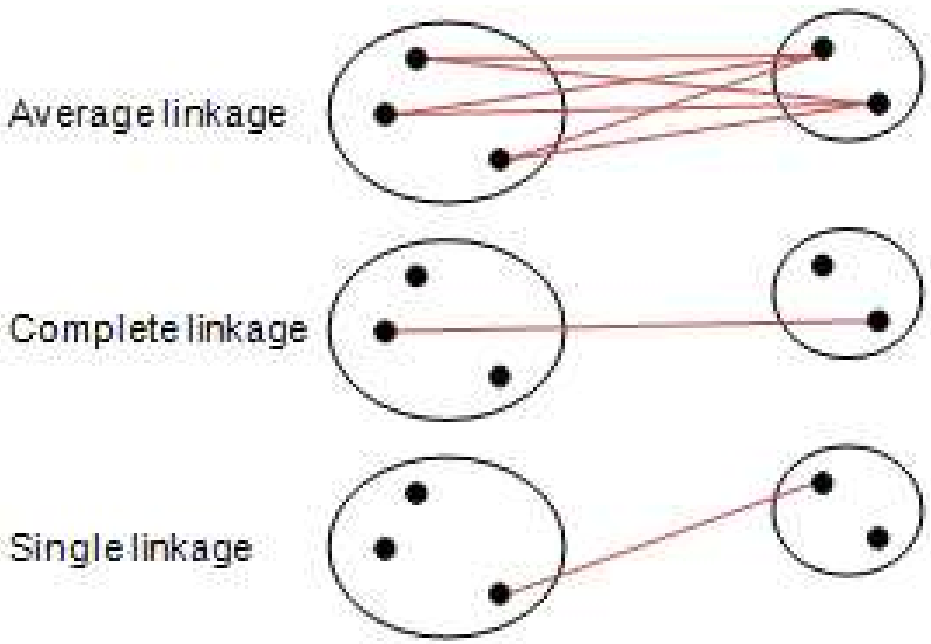
Algorithm:

1. Place each data point into its own singleton group
2. Repeat: iteratively merge the two closest clusters
3. Until: all data are merged into a single cluster

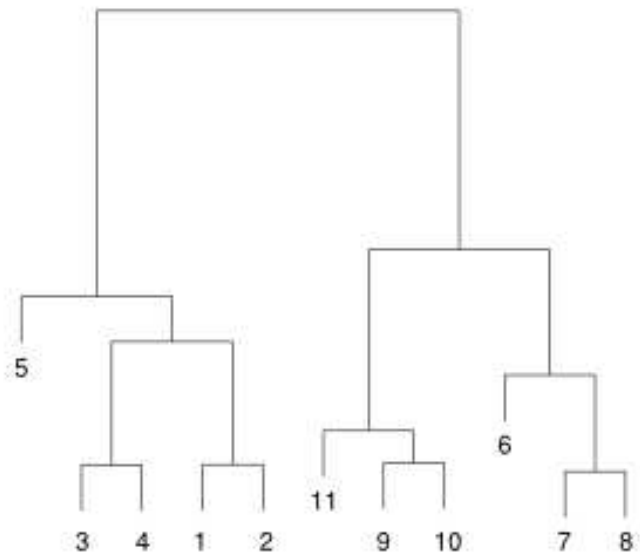
The difference is given by merging criteria (linkage).

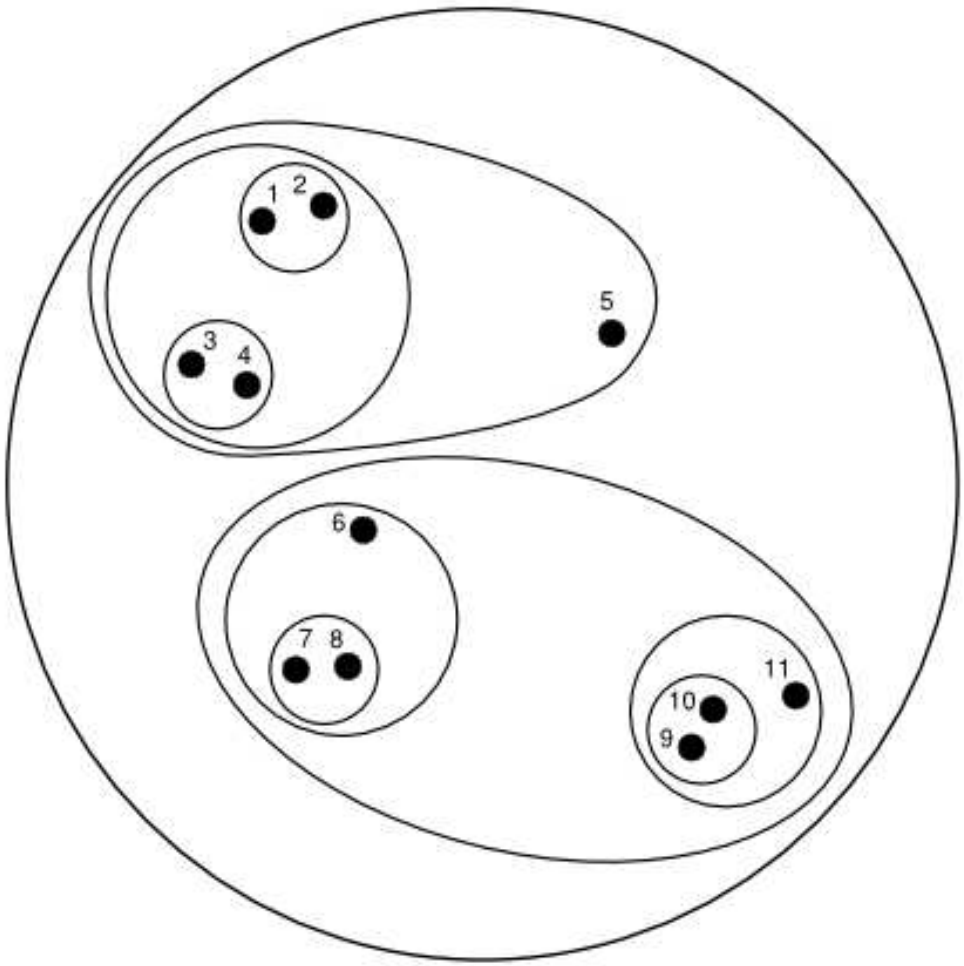
Linkage criteria

- Single linkage (minimum distance)
- Complete linkage (maximum distance)
- Average linkage (average intercluster distance)



Example of hierarchical clustering

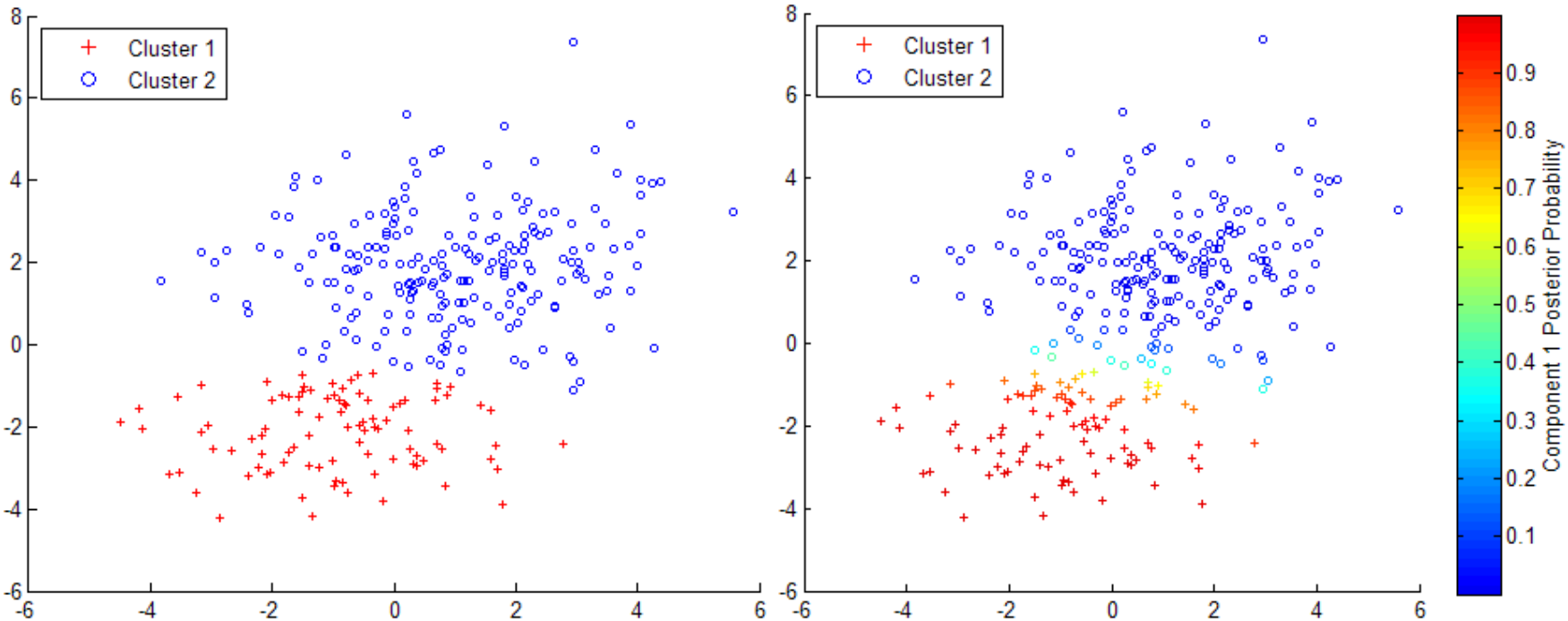




Partitional clustering

Two types of assignement:

- *hard* clustering : each point is assigned to one and only cluster
- *soft* clustering : each point is assigned different probabilities of belonging to each of the clusters



K-means algorithm (1)

Objective function: search for a partition than minimizes the sum of within-cluster distances.

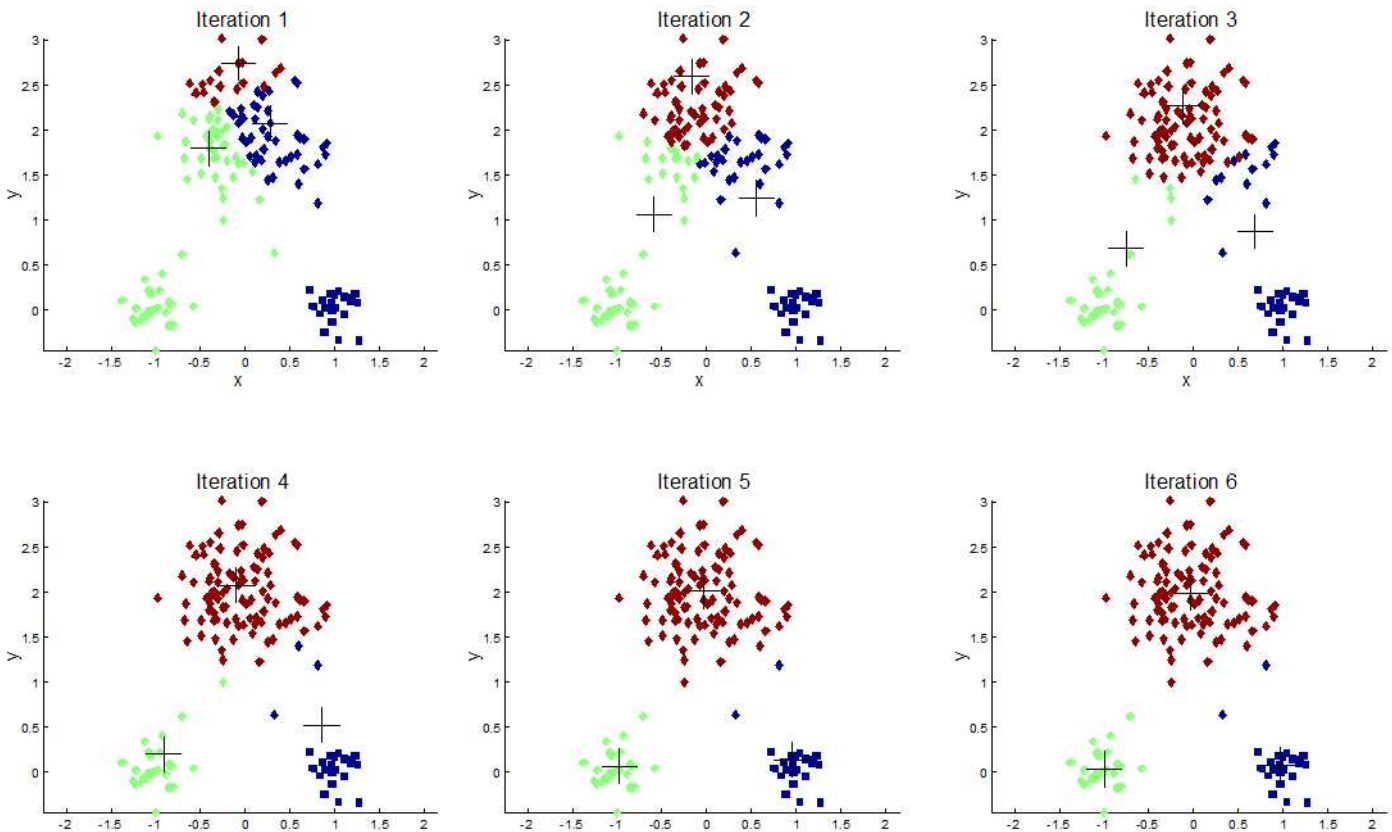
$$J = \sum_{c=1}^{K} \sum_{\mathbf{x}_i \in C_c} (\mathbf{x}_i - \boldsymbol{\mu}_c)^2 \tag{2}$$

where:

- K - the number of clusters (given);
- C_c - the subset of points $\mathbf{x}_i$ that belong to cluster c;
- $\boldsymbol{\mu}_c$ - the centroid of cluster c.

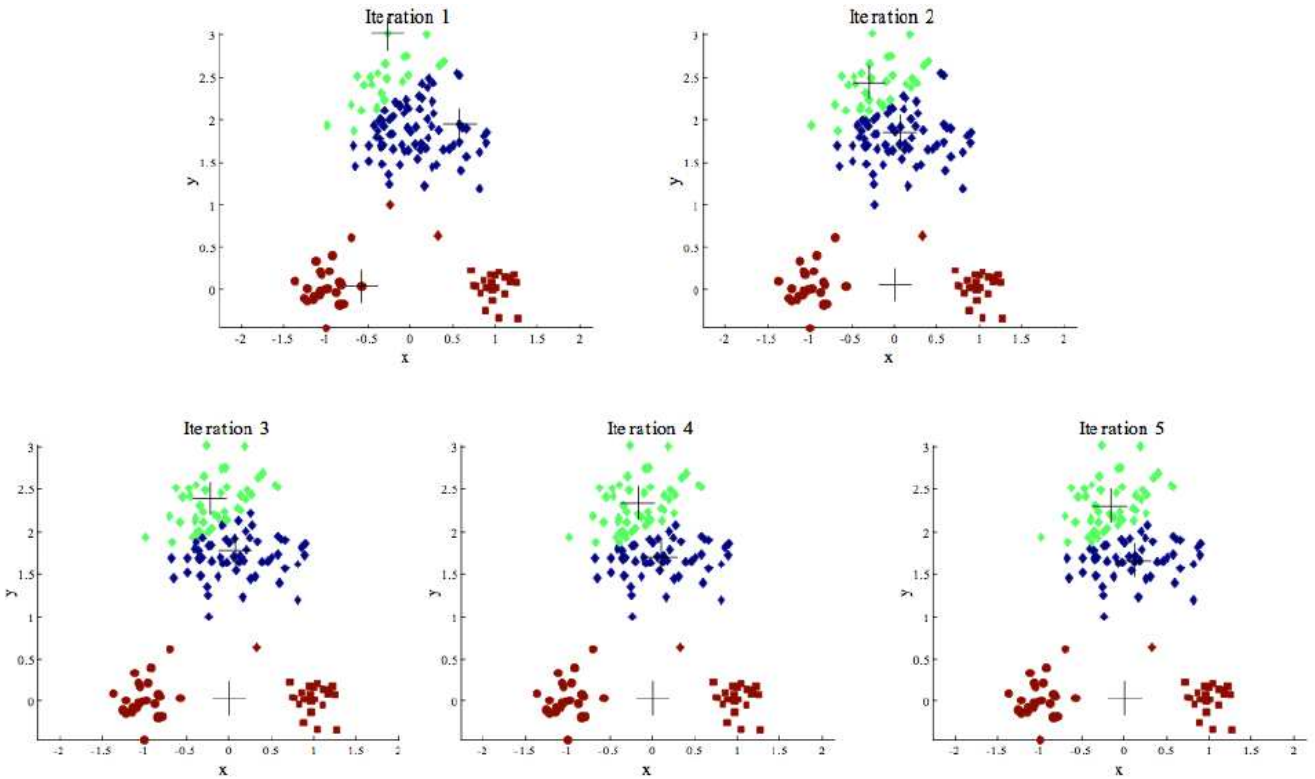
K-means algorithm (2)

1. Randomly choose the coordinates of the K centroids.
2. Assign each point to the closest centroid.
3. Recalculate the centroids, based on the assignment of points from step 2.
4. Repeat steps 2 and 3 until convergence.



K-means algorithm (3)

Choice of the initial centers!



Perform multiple K-means with different intializations.

How many clusters?

How many clusters?

Six Clusters

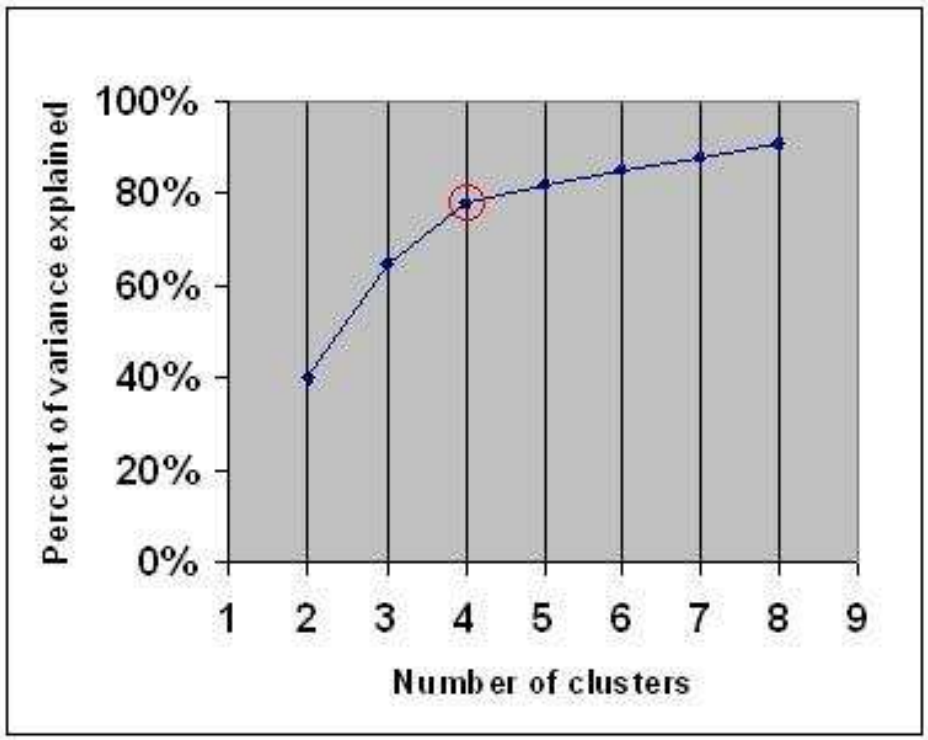
Two Clusters

Four Clusters

How to choose the number of clusters?

Penalize complexity: choose the number of clusters so that adding one more cluster does not improve the model significantly.

- Elbow criterion



- Information criterion (BIC-Bayesian information criterion, AIC-Akaike information criterion)

Mixture models

Let $f : \mathcal{F} \longrightarrow \mathbb{R}$ be the density of the data points in the space.

A **mixture model** is a probabilistic model that assumes the data is generated from a mixture of K components (sub-populations).

$$f(\mathbf{x}_i) = \sum_{c=1}^{K} \pi_c \phi(\mathbf{x}_i, \boldsymbol{\theta}_c) \tag{3}$$

where $\sum_{c=1}^{K} \pi_c = 1$

Gaussian mixture model (1)

The most common mixture model is the Gaussian Mixture Model (GMM), a weighted sum of Gaussians components:

$$p(\mathbf{x}_i) = \sum_{c=1}^{K} \pi_c \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \tag{4}$$

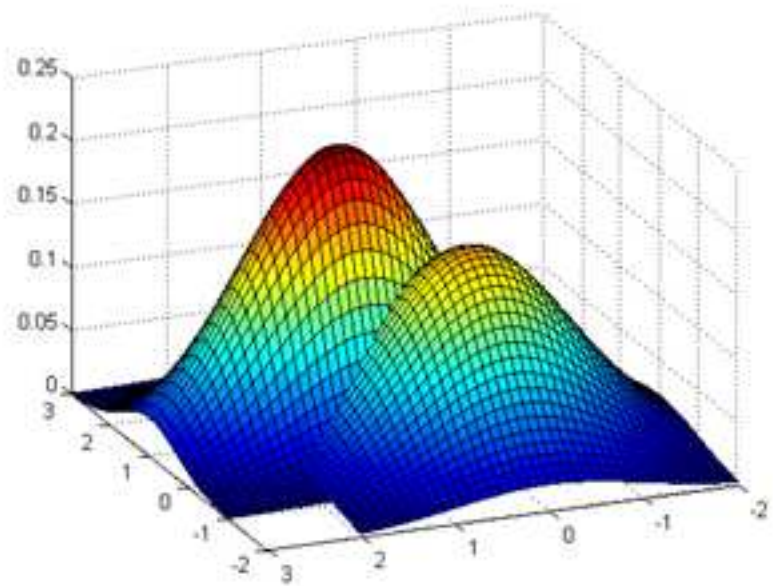
where a multivariate D-dimensional Gaussian distribution is:

$$\mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) = \frac{1}{(2\pi)^{\frac{D}{2}} \mid \boldsymbol{\Sigma}_c \mid^{\frac{1}{2}}} \exp\{-\frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_c)^T \boldsymbol{\Sigma}_c (\mathbf{x}_i - \boldsymbol{\mu}_c)\} \tag{5}$$

Gaussian mixture model (2)

For a Gaussian mixture, the parameters to estimate:

$$\theta = (\pi_c, \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c), c = 1..K \tag{6}$$



Gaussian mixture model (3)

Find the parameters that maximize the likelihood of the GMM knowing the data $\mathbf{X}$:

$$p(\mathbf{X} \mid \theta) = \prod_{i=1}^{N} p(\mathbf{x}_i \mid \theta) \tag{7}$$

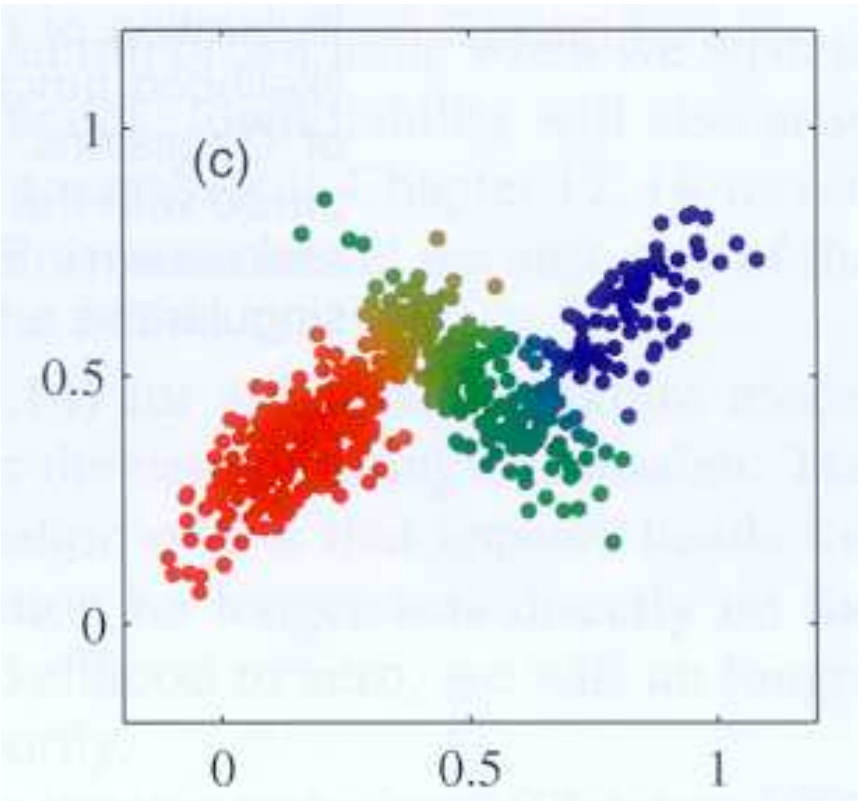
$$\hat{\theta} = \text{argmax } p(\mathbf{X} \mid \theta) \tag{8}$$

Maximize the log-likelihood:

$$\log p(\mathbf{X} \mid \theta) = \sum_{i=1}^{N} \log p(\mathbf{x}_i \mid \theta) = \sum_{i=1}^{N} \log \sum_{c=1}^{k} \pi_c \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c) \tag{9}$$

Gaussian mixture model (4)

Soft clustering assignement.



Expectation-Maximization (EM) (1)

1. Initialization step: $\theta^0 = (\pi_c^0, \boldsymbol{\mu}_c^0, \boldsymbol{\Sigma}_c^0)$
2. **Expectation** (E) step: compute the responsibilities γ_{ic}

$$\gamma_{ic} = p(c|\mathbf{x}_i) = \frac{\pi_c \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)}{\sum_j \pi_{j=1}^K \mathcal{N}(\mathbf{x}_i \mid \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \tag{10}$$

3. **Maximization** (M) step: using the responsibilities estimated in the Expectation step, recompute the parameters of the model:

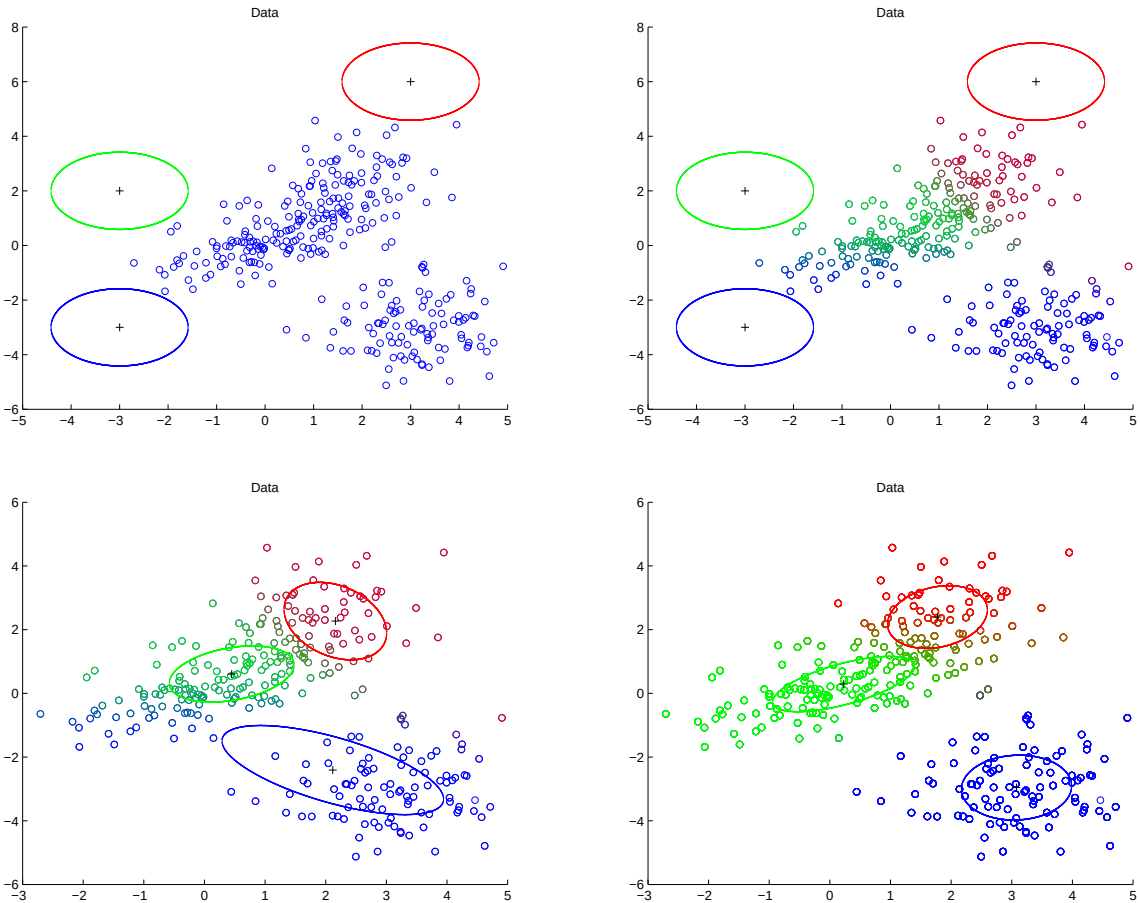
$$\pi_c = \frac{\sum_i \gamma_{ic}}{N} \tag{11}$$

$$\boldsymbol{\mu}_c = \frac{\sum_i \gamma_{ic} \mathbf{x}_i}{\sum_i \gamma_{ic}} \tag{12}$$

$$\boldsymbol{\Sigma}_c = \frac{1}{\sum_i \gamma_{ic}} \sum_i \gamma_{ic} (\mathbf{x}_i - \boldsymbol{\mu}_c)^T (\mathbf{x}_i - \boldsymbol{\mu}_c) \tag{13}$$

4. Repeat steps 2 and 3 until convergence (the log-likelihood does not change significantly).

Expectation-Maximization (EM) (2)

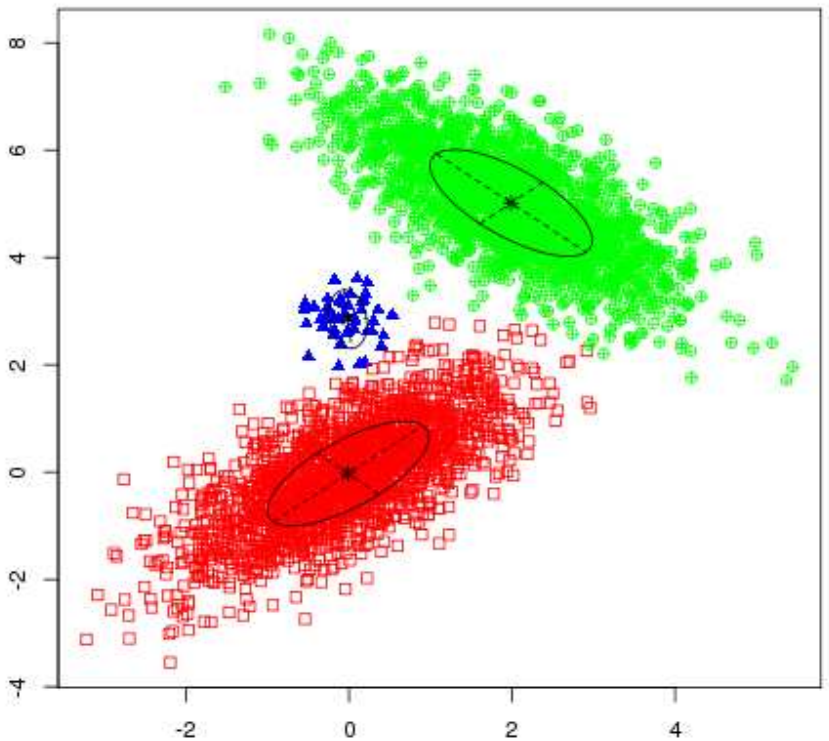


Choice of the covariance matrices

Choice of the covariance matrices:

- Spherical:
  - $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \cdots = \boldsymbol{\Sigma}_K = \sigma^2 \mathbf{I}$ (same spherical covariance matrix for all components)
  - $\boldsymbol{\Sigma}_c = \sigma_c^2 \mathbf{I}$ (different spherical covariance matrices)
- Diagonal:
  - $\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \cdots = \boldsymbol{\Sigma}_K =$ (same diagonal covariance matrix for all components).
  - $\boldsymbol{\Sigma}_c$ (different diagonal covariance matrices)
- Full covariance.

Choice of the model

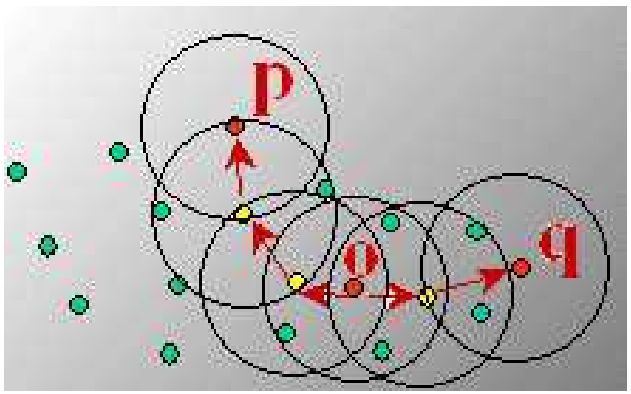


Mclust package in R.

Density-based clustering (1)

DBSCAN (Density-based spatial clustering of applications with noise):

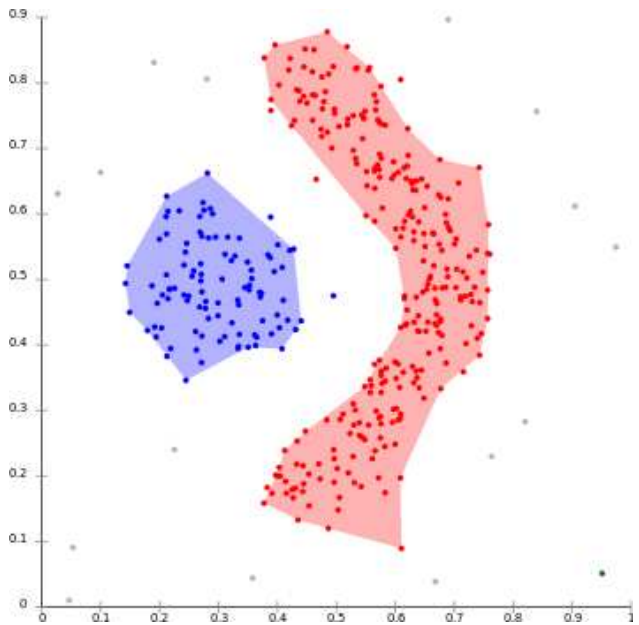
- point q is **density-reachable** from point p if there is a sequence of points where each point p_{i+1} is *directly density-reachable* from point p_i
- a point p_{i+1} is **directly density-reachable** from point p_i if it is in the ϵ-neighbourhood of p_i and if p_i is surrounded by at least *minPts* points



Density-based clustering (2)

DBSCAN:

- allows arbitrarily shaped clusters
- does not require the number of clusters in advance
- it is able to detect noise
- requires two parameters (ϵ,*minPts*)



Spectral clustering

Graph-based method: $G = (V, E)$.
Adjacency matrix: $W = (w_{ij}), i, j = 1..N$:

- $w_{ij} = 1$ if points $\mathbf{x}_i$ and $\mathbf{x}_j$ are connected in the neighbourhood graph G.
- $w_{ij} = 0$ otherwise.

Degree of a vertex:

$$d_{ij} = \sum_{i=1}^{N} w_{ij} \tag{14}$$

Laplacian matrix:

$$L = D - W \tag{15}$$

Normalized-cut algorithm

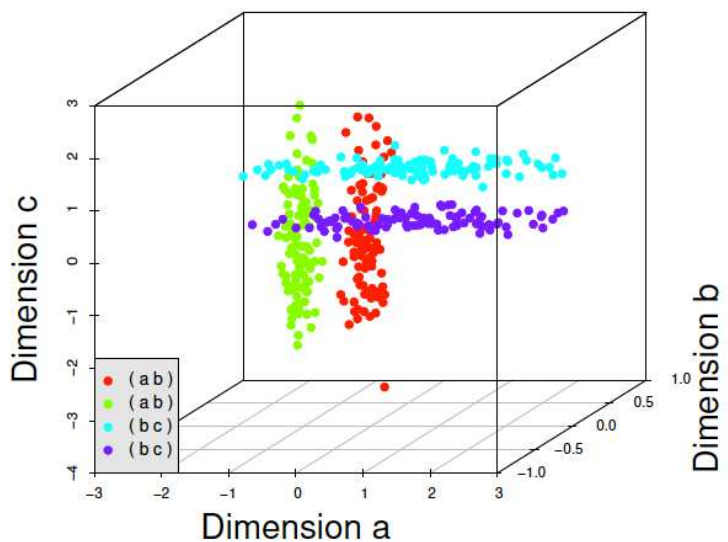
Algorithm:

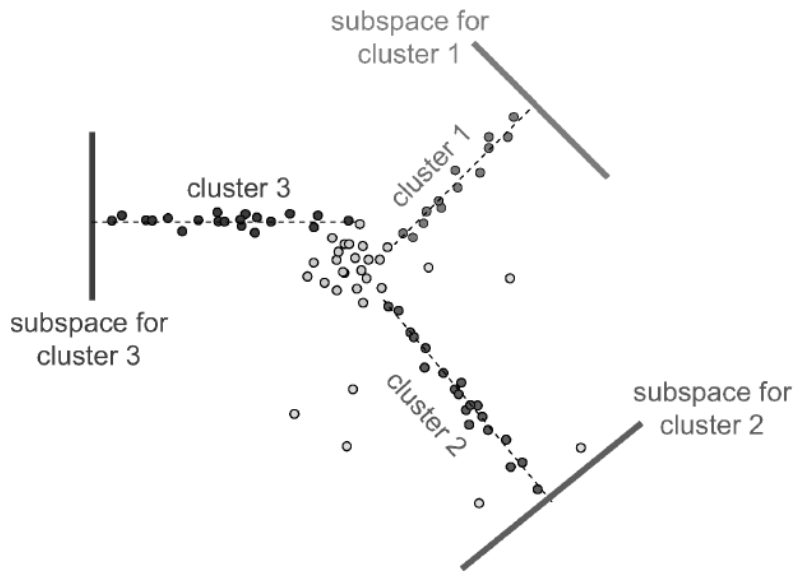
1. Compute the similarity matrix.
2. Compute the Laplacian matrix.
3. Compute the first k eigenvectors u_1,u_2,..,u_k of L.
4. Let $U = (u_1, u_2, .., u_k)$ be the matrix with the columns given by u_j, $j = 1..k$.
5. Let y_i be the vectors corresponding to the rows of U.
6. Cluster the points y_i using K-means.

The number of connected components is equal to the number of 0 eigenvalues.

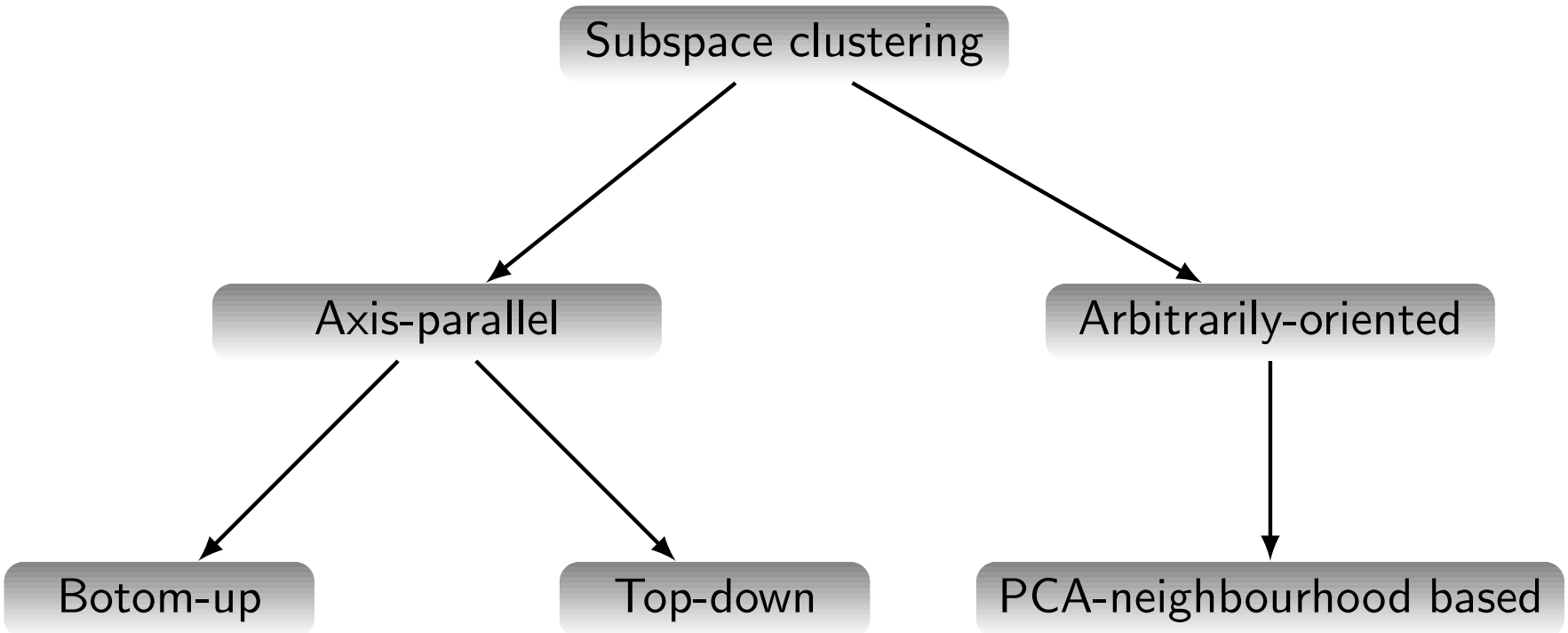
Subspace clustering

Subspace clustering attempts to find clusters in different subspaces of the original feature space.





Summary: taxonomy of subspace clustering methods



Subspace clustering

Axis-parallel:

- bottom-up:
  - start with 1-dimensional subspaces;
  - increase the dimensionality of the subspaces until no higher-dimensional dense regions are found;
- top-down:
  - start with D-dimensional space and perform clustering;
  - once clusters identified, search for the subspace of each cluster.

Axis-parallel clustering: Bottom-up

CLIQUE [Agrawal et al., 1998]:

1. Identification of subspaces that contain clusters
   - create histogram for each dimension and divide them into static grids
   - select bins with density higher than a threshold (at least n points)
   - generate higher-dimensional units using only the lower-dimensional units that are dense
   - make use of the downward closure property of density to discard units that are not dense
2. Finding clusters
   - input: a set of dense units, all in the same d-dimensional space
   - combine adjacent dense units
   - the new dense units correspond to clusters (similar to finding connected components in a graph)
3. Generating minimal cluster descriptions
   - input: disjoint sets (clusters) of connected d-dimensional units in the same subspace
   - maximal number of regions (rectangles) required to cover the cluster

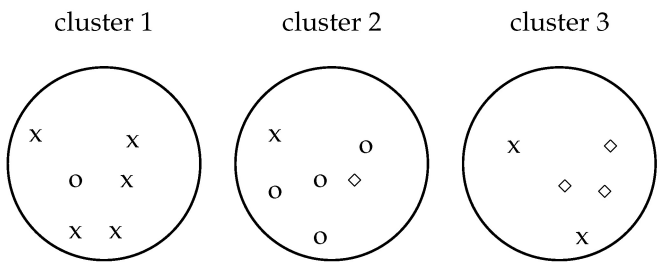
Outline

Clustering evaluation (1)

1. Cluster purity

$$\text{purity}(\Omega, \mathbb{C}) = \frac{1}{N} \sum_{c} \max_{m} |\omega_c \cap c_m| \tag{16}$$

where $\Omega =$ the set of classes and $\mathbb{C}$ is the set of clusters. Each cluster c_m is assigned the label of the most frequent class ω_c in that cluster and the accuracy is measured by counting the number of elements that are assigned to the correct class.



► **Figure 16.4** Purity as an external evaluation criterion for cluster quality. Majority class and number of members of the majority class for the three clusters are: x, 5 (cluster 1); o, 4 (cluster 2); and ⋄, 3 (cluster 3). Purity is $(1/17) \times (5 + 4 + 3) \approx 0.71$.

Clustering evaluation (2)

2. Confusion matrix (contingency table)

Eval	Same cluster	Different clusters
Same class	*TP*	*FN*
Different classes	*FP*	*TN*

where:

- *TP* = true positive
- *TN* = true negative
- *FP* = false positive
- *FN* = false negative

Clustering evaluation (3)

3. Rand Index

$$RI = \frac{TP + TN}{TP + TN + FP + FN} \tag{17}$$

4. F-measure

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

where: β=parameter that controls the balance between precision and recall

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

Clustering evaluation (4)

Internal measures:

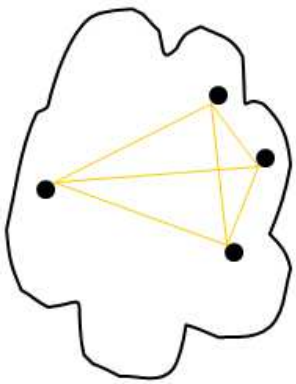
- Cohesion: measures how closely related are objects in a cluster (within sum of squares).

$$WSS = \sum_{c=1}^{K} \sum_{\mathbf{x}_i \in C_c} d(\mathbf{x}_i, \boldsymbol{\mu}_c)^2 \tag{18}$$

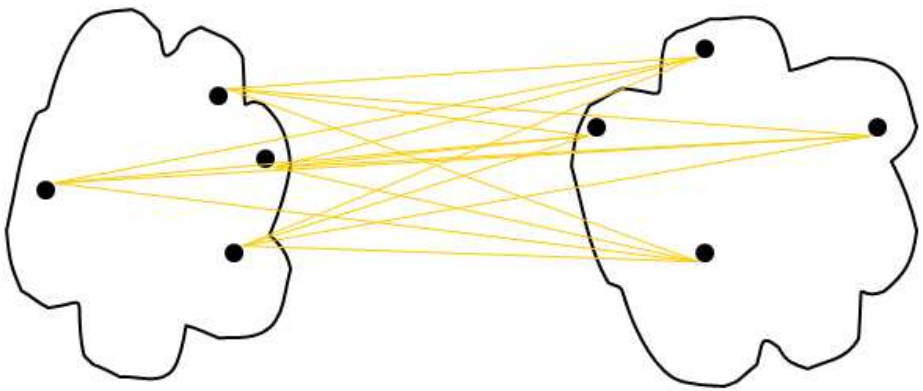
- Separation: measures how well-separated clusters are (between sum of squares).

$$BSS = \sum_{c=1}^{K} N_c d(\boldsymbol{\mu}, \boldsymbol{\mu}_c)^2 \tag{19}$$

Cluster evaluation (5)



cohesion



separation

Which clustering is best adapted?